HAL
open science

Computation of the L∞-norm of finite-dimensional linear systems

Grace Younes

To cite this version:

Grace Younes. Computation of the L∞-norm of finite-dimensional linear systems. Robotics [cs.RO]. Sorbonne Université, 2022. English. NNT : 2022SORUS125 . tel-03863650

HAL Id: tel-03863650

https://theses.hal.science/tel-03863650v1

Submitted on 21 Nov 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

École doctorale de sciences mathématiques de Paris centre

Thèse de doctorat

Discipline : Mathématiques

présentée par

Grace Younes

Computation of the L_∞-norm of finite-dimensional linear systems

---

dirigée par Fabrice Rouillier – Alban Quadrat

Soutenue le 20/01/2022 devant le jury composé de :

M. Olivier Bachelier	Université de Poitiers	rapporteur
Mme. Catherine Bonnet	Inria Saclay-Île-de-France	examinatrice
Mme. Mioara Joldes	LAAS-CNRS	examinatrice
M. Laureano Gonzalez-Vega	Universidad de Cantabria	rapporteur
M. Alban Quadrat	Inria Paris-Sorbonne Université	co-directeur
M. Fabrice Rouillier	Inria Paris-Sorbonne Université	directeur
Mme. Annick Valibouze	Sorbonne Université-LIP6	examinatrice

Institut de mathématiques de Jussieu-
Paris Rive gauche. UMR 7586.
Boîte courrier 247
4 place Jussieu
75 252 Paris Cedex 05

Université de Paris.
École doctorale de sciences
mathématiques de Paris centre.
Boîte courrier 290
4 place Jussieu
75 252 Paris Cedex 05

It's not about achieving the goal.

It's about who you have to become in order to achieve the goal.

The juice is in the growth.

Tony Robbins.

Acknowledgments

First of all, I would like to thank my thesis supervisors, Fabrice Rouillier and Alban Quadrat. Without them, the work would not have been complete. I thank them for their support and their wise advice during my thesis. Their in-depth knowledge enabled me to carry out this work. In addition, I thank them for their patience, good humor during each meeting, for knowing how to make the work environment so pleasant, even during Covid lockdown, and for being there with me to guide me and listen to me. I learned a lot from them. They have helped me deepen my work so that I can be proud of the work done today. I am deeply grateful to them.

I would like also to thank Yacine Bouzidi for collaborating with us during the first year of my Ph.D. I appreciate the moral and technical support he showed me, and all the ideas we discussed together thanks to which I got to know more about computational mathematics and the subject in general.

I express my sincere thanks to Olivier BACHELIER and Laureano GONZÁLEZ-VEGA for agreeing to be reporters and to devote time to examining the manuscript. They did me the honor of studying it carefully and I had the pleasure to read their reports and comments on the work. I am very honored to also thank Catherine BONNET, Mioara JOLDES, and Annick VALIBOUZE for having accepted the invitation to be among the jury of my thesis.

I would then like to thank all the staff of Inria Paris, in particular the members of the OURAGAN team Antonin Guilloux, Elias Tsigaridas, Josué Tonelli-Cueto, Pierre-Vincent Koseleff... for their scientific knowledge and their friendly exchanges during these three years, and with whom we did all the (online) seminars and we kept a pleasant and enjoyable interaction even during the Covid lockdown. A big thank you also to the members of the IMJ-PRG and to all the staff of Sorbonne University, where I was well received and surrounded during these three years, in particular to Elisha Falbel who was my tutor during this stay.

I express my thanks to my present and past colleagues in IMJ-PRG, Mathieu, Thibault, Jean-Michel, Thomas, Thomas, Haowen, Anna, Anna, Eva, Linyuan, Sylvain, Sudarshan,

Thiago, Nelson, Léo, Raphaël, Adrien, Nathanaël, Yicheng, Chenyu, Jacques, Arnaud, Gabriel... Your special personalities made the stay at Sorbonne University (15-16-5ème) very particular, enjoyable and funny. I wish you a very good continuation in your future life and career. Also thanks to my special office-mate Mahya, who was the first person I met on my arrival to IMJ-PRG. Thank you for your help, for the beautiful times we have spent inside and outside the office, and for all the jokes and loud laughs. Also to Christina who was the only one with whom I could discuss a little about formal calculus subjects, and to my genius office-mate Hernan, thank you for your time to answer my questions and for trying to provide help as much as you could. To Christophe, thank you for all the help and the encouragements. And to Perla, thank you for all the jokes and the lebanese-y chit-chats we shared during coffee and lunch breaks. Thank you also for your funny stories and your academic and fitness encouragements.

I want also to thank all my friends in France who was there for me whenever I needed them. Thanks to Tonia for all her support and for the enjoyable times we spent together, and to Wafa for her help and with whom I spent great times in France. Thanks to Cynthia, Issa, Mirella, Thurayia, and Dia for the beautiful gatherings in Paris. Thanks to Marianne with whom I always shared the news of my Ph.D. and was always supportive and encouraging and full of wise advise and to Elie for his help and support. I also thank my childhood friends in Lebanon, Patricia, Catherine, Pamela, Maria, Melissa and Jennyfer for the warm times we spent during my holidays in Lebanon.

My biggest gratitude is towards my lovely family. To the memory of my father Elias, thank you for all the hard work you did for me to be here today. Your memory will always encourage me through hard times. To my beautiful mother Rouba, thank you for your unconditional love and your tireless efforts and support in every path I take to achieve my dreams and goals. Without you I would not be here today. To my beautiful sisters Reine and Eliane, thank you for loving me and for showing me always how proud you are of me. Also thanks to my little adorable Adib for just existing in my world. Thanks to my grandma Samia for her love and prayers. Thanks to my precious uncles (khalo) Ghassan, Anwar and Hadi for always advising me, supporting me, loving me and treating me like a rockstar. I appreciate and thank God for your presence in my life. Thanks to my uncles Tony, Georges, Raymond and Pierre and my sweet untie Nada for their support and advice. To my favourite person Georges, thank you for your presence, motivation, support, and for cheering me up during every stressful moment in my thesis and my stay in France. Thank you also for always believing in my potential and pushing me forward.

Thank you God for everything.

Résumé

Dans cette thèse, nous étudions le problème du calcul de la norme L^∞ des systèmes de dimension finie, linéaires et invariants dans le temps. Ce problème est ramené au calcul de la y-projection maximale des solutions réelles $(x,\, y)$ d'un système d'équations polynomiales bivariées $\Sigma = \left\{P = 0, \frac{\partial P}{\partial x} = 0\right\}$, où $P \in \mathbb{Z}[x, y]$. Nous utilisons alors des méthodes classiques de calcul formel pour résoudre ce problème. En particulier, nous étudions alternativement une méthode basée sur des représentations univariées rationnelles, une méthode basée sur la séparation des racines, et enfin une méthode basée sur la variation de signes des coefficients dominants d'une suite signée de sous-résultants (suite de Sturm-Habicht) et l'identification d'un intervalle isolant pour la y-projection maximale des solutions réelles de Σ. Nous calculons ensuite la complexité binaire dans le pire des cas de chacune des méthodes proposées et nous comparons leur comportement théorique. Enfin, nous implémentons chacune des méthodes sous `Maple` et nous comparons leur comportement pratique (complexité moyenne). Une généralisation des algorithmes précédents au cas de polynômes P dépendants aussi de paramètres $\alpha = [\alpha_1, \ldots, \alpha_d] \in \mathbb{R}^d$ est finalement proposée. Pour cela, nous résolvons le problème en utilisant la notion de Décomposition Cylindrique Algébrique, classique en géométrie algébrique.

Mots-clés

Calcul de la norme L^∞, Systèmes polynomiaux, Solutions réelles maximales, Calcul symbolique, Calcul de complexité, Implémentation, Théorie du contrôle.

Computation of the L^∞-norm of finite-dimensional linear systems

In this dissertation, we study the computation of the L^∞-norm of finite-dimensional linear time-invariant systems. This problem is first reduced to the computation of the maximal y-projection of the real solutions $(x,\, y)$ of a bivariate polynomial equations system $\Sigma = \left\{P = 0, \frac{\partial P}{\partial x} = 0\right\}$, where $P \in \mathbb{Z}[x, y]$. Then, we use standard computer algebra methods to solve this problem. In particular, we alternatively study a method based on rational univariate representations, a method based on root separation, and finally a method first based on the sign variation of the leading coefficients of a signed subresultant sequence (Sturm-Habicht) and on the identification of an isolating interval for the maximal y-projection of the real solutions of Σ. We then compute the worst-case bit complexity of each method and compare their theoretical behavior. We also implement each method in `Maple` and compare their practical behavior (average complexity). A generalization of the above algorithms is finally proposed to the case where the polynomial P also depends on a set of parameters $\alpha = [\alpha_1, \ldots, \alpha_d] \in \mathbb{R}^d$. To do that, we solve the problem using the notion of the Cylindrical Algebraic Decomposition, well-known in algebraic geometry.

Keywords

L^∞-norm computation, Polynomial systems, Maximal real roots, Symbolic computation, Complexity computation, Implementation, Control theory.

Contents

Introduction **11**
0.1 Existing methods . . . 13
0.1.1 Numerical approach . . . 13
0.1.2 Symbolic-numeric approach . . . 14
0.2 Proposed methods . . . 16
0.2.1 Non-parametric case . . . 17
0.2.2 Parametric case . . . 19

1 Problem Motivation From Control Theory **21**
1.1 Definition of a linear time-invariant system . . . 21
1.2 Stability of LTI systems . . . 25
1.3 Maximum energy gain and L^∞-norm . . . 30
1.4 A real algebraic geometric reformulation . . . 37
1.5 Existing computational methods . . . 44
1.6 Robust control theory in a nutshell . . . 47

2 Prerequisite in Computer Algebra **55**
2.1 Notations . . . 55
2.2 Greatest common divisor . . . 58
2.3 Resultant of two polynomials . . . 61
2.4 Subresultant sequence . . . 64
2.5 Sturm-Habicht Sequence . . . 70
2.5.1 Sturm-Habicht sequence and real roots of polynomials . . . 72
2.6 Univariate polynomials and Root isolation . . . 76
2.6.1 Subdivision-based algorithms for root isolation . . . 78
2.7 Solving bivariate algebraic systems . . . 81
2.7.1 Rational Univariate Representation – RUR . . . 84

2.8 Systems depending on parameters 89
2.8.1 Discriminant variety 90
2.8.2 Cylindrical Algebraic Decomposition 92

3 L^∞-norm Computation 97
3.1 Non-parametric case problem 97
3.1.1 RUR method 98
3.1.2 Root separation method 107
3.1.3 Sturm-Habicht method 115
3.2 Parametric case 123

4 Application 133
4.1 Implementation and experiments 133
4.2 Some numerical method drawbacks 135
4.3 Practical examples 138

5 Conclusion 149

Introduction

My research program within OURAGAN team (Inria Paris) focuses on the development of new algorithms for the study of the maximal y-projection of the real solutions of a bivariate polynomial system describing the critical points of a bivariate polynomial depending or not on parameters.

Through its various applications in control theory (robust and optimal control) [107, 45, 44], robotics [85, 68], and signal processing [75, 71], etc., the study of the real solutions of a bivariate polynomial system depending on parameters is a classic problem, abundantly studied in computer algebra (see, e.g., [28, 63, 43, 65] and the references therein).

An example of an application worth mentioning is the recent work carried out by Guillaume Rance in his PhD thesis [77], in collaboration with the *Safran Electronics & Defense* company. The objective of this work was to compute the H^∞ controllers for gyro-stabilized systems depending on parameters. An important result of this work was to show that the problem is reduced to that of the study of maximum real solutions of some algebraic systems. To solve this last problem, the choice was made to use classical techniques of symbolic computation such as *Rational Univariate Representation*, *discriminant variety* or the *cylindrical algebraic decomposition*. In few words, the idea is to calculate an algebraic variety in the parameter space that decomposes the latter into cells. Within the obtained cells, the branches of the solutions of the system remain regular. This guarantees that the maximum solution does not change within the same cell which allows, for instance, the engineers to follow it easily by means of numerical methods.

In the continuation of this work, the problem that interests us is a *certified computation* of the L^∞*-norm of finite-dimensional linear time-invariant dynamical control systems.*

An interesting thing about *dynamical systems* is that they can be represented by ordinary differential equations. This is due to the property saying that the manner the system is changing at any given time is a function of its current *state*. For any dynamical system, if we take a look at how the energy changes by analyzing the relationship between the states and their derivatives, we can make conclusion about some physical properties of the sys-

tem, *stability* for instance. If the energy is being dissipated over time, then the system is stable, and the faster the energy is dissipated the more stable the system is. However, if the energy is growing unbounded over time, then the system is said to be unstable. This notion of stability is a deep-rooted property of the system due to the connection between the states of the system and their derivatives. Moreover, the way the system moves can also be influenced by external forces being added or removed over time. Hence, the evolution of a dynamical system is a function of the current state as well as any *external inputs*. With this being said, a *state-space representation* [20, 107], precisely introduced in the next chapter, is simply a restructure of the high order differential equations into a set of first order differential equations that focus on the relationship between derivatives, current states, and external inputs, which makes the system easier to analyze.

State-space representation is based on the state vector x that is the vector of all state variables. The variation of the state vector is a linear combination of the current state plus a linear combination of the external inputs. The way that every state changes in terms of the input of the system can leave us with a set of first order differential equations, which will be considered here as a system of linear equations, that we can package into a matrix form. This allows to apply linear algebra methods and have access to some useful and important mathematical tools for studying dynamical systems (see, e.g., [20, 107] and the references therein).

Furthermore, the control theory in control systems engineering deals with the control of continuously operating dynamical systems engineered processes and machines. This special characteristic allows the control systems to play an important role in the development and advancement of modern technology and civilization. The aim is to develop a control model for controlling dynamical systems using a control action in an optimum manner ensuring control stability. This criterion was highly studied and many projects aimed its establishment since 19th century. In addition, a controller with requisite corrective behaviour is required along with other aspects such as *controllability* and *observability* (see, e.g., [20, 107]).

To arrange controllers to achieve these requirements, in modern control theory, H^∞ *methods* are used [106, 56, 99, 39, 102, 107, 33]. To use these methods, a control designer expresses the control problem as a mathematical optimization problem and then finds the controller that solves this optimization. After initiating this theory [106], G. Zames formulated a basic feedback problem as an optimization problem with an operator norm, in particular, an H^∞-norm. The H^∞-norm has become popular in control theory since the concept of *robustness*, which plays a fundamental role in control theory, can easily be refor-

mulated in the *frequency domain* using this particular norm. But, to apply H^∞ techniques successfully, an important level of mathematical understanding is needed.

This norm can either be computed numerically via, e.g., *bisection algorithms* for the search of imaginary eigenvalues of *Hamiltonian matrices* [19, 22] or symbolically via, e.g., the maximal real root γ of a univariate polynomial $n(\omega, \gamma)$ depending on parameters ω [24] or as the maximal γ-projection of a real curve bounded on the γ-direction as followed in this dissertation. In this case, if the dynamical system, represented by its *transfer matrix* F, depends on a parameter set α, namely, on unfixed values, then the L^∞-norm of F is a function of those parameters α. Hence, the parameter space has to be decomposed into different "cells" above each an expression of the searched function γ can be identified.

0.1 Existing methods

In this section, we discuss the existing numerical and symbolic methods for the computation of the L^∞-norm of finite-dimensional linear time-invariant systems. Then, in the next section, we briefly introduce the proposed methods that compute the L^∞-norm when the transfer matrix does not depend on parameters. This case is then a stepping stone for generalising the study to the parametric dependency situation.

0.1.1 Numerical approach

Contrary to the standard L^2-norm, no tractable formula is known for the characterization of the L^∞-norm of finite-dimensional systems (i.e., systems defined either by linear ordinary differential equations or by linear recurrence equations) [39, 107]. Hence, the standard methods for the L^∞-norm computation are numerical (e.g., search for imaginary eigenvalues computation of Hamiltonian matrices, bisection algorithms).

For instance, in the late 90's, few algorithms demonstrating fast convergence of iterative approaches and exploiting the properties of the singular values of the transfer matrices have been developed.

S. Boyd, V. Balakrishnan and P. Kabamba established in [19] a correspondence between the singular values of a transfer matrix evaluated along the imaginary axis and the imaginary eigenvalues of a related Hamiltonian matrix. Their proof, based on a simple linear algebraic approach, uses a more intuitive explanation based on quadratic optimal control problem. Their result gave way to a simple bisection algorithm to compute the L^∞-norm of a transfer matrix.

Similarly, based on the relation between the singular values of the transfer function matrix and the eigenvalues of a related Hamiltonian matrix, N. A. Bruinsmaa and M. Steinbucha developed in [22] a fast algorithm to compute the L^∞-norm of a transfer function matrix with guaranteed accuracy.

Recently, the methods reported in [60] and [8] compute the L^∞-norm via localizing the common roots of two or three polynomials. In their paper [60], M. Kano and M. C. Smith first reduce the problem to the localization of the real solutions of a bivariate polynomial and then use Sturm chain tests to guarantee the accuracy of their algorithm.

In [8], using techniques involving structured linearization of the *Bezoutian matrices*, M. N. Belur and C. Praagman addressed the computation of the L^∞-norm by directly computing the isolated common zeros of two bivariate polynomials. In their paper, using numerical experiments on random transfer functions, the proposed method to L^∞-norm computation is then compared with that of N. A. Bruinsmaa and M. Steinbucha [22].

In [12], P. Benner, V. Sima and M. Voigt constructed algorithms for the computation of the L^∞-norm of transfer functions related to descriptor systems, both in the continuous and discrete-time context. This was done by computing the eigenvalues of certain structured matrix pencils by transforming them to skew-Hamiltonian/Hamiltonian matrix pencils, using the original data. To increase robustness and efficiency of their method, they further applied a structure-preserving algorithm to compute the desired eigenvalues.

However, all these methods are numeric and are devoted to linear systems free of parameters. Putting apart the case of parameters dependency, it is worth mentioning that with numerical methods, the result is usually obtained within a short time but with a slight error up to a precise accuracy. In contrast, when using symbolic methods, the result usually takes more time to be computed but is *certified* to be exact.

0.1.2 Symbolic-numeric approach

Contrary to numeric algorithms, symbolic algorithms are *certified*. By certified algorithms, we mean "guaranteed methods", that is to say, a method that, for any input, computes a result without ambiguities after a finite number of steps.

In particular, the algorithms that can structurally enter in an indefinite loop or even return a false result are excluded from this category of algorithms. However, in the case where such an algorithm is not able to return a solution of a given problem, it can simply return an answer explaining this situation.

This does not preclude having algorithms dedicated to certain constraints on the original problem as long as they are verifiable by at least one other guaranteed algorithm. Methods

depending on "generic" options, making the algorithm probabilistic, even if the probability of the error is low, are excluded from this range of algorithms, unless another algorithm can prior be used to verify the "genericity" of the considered option.

By considering the L^∞-norm computation of transfer matrices depending on parameters, we can find few methods based on symbolic computation such as [2, 45, 61]. Symbolic approaches for solving parametric optimization problems have some advantages over their numerical counterparts: for instance, using symbolic approaches, non-convex feasible regions would not represent a theoretical concern. On the other hand, symbolic approaches do not suffer from the size of the feasible parameter regions, even when unbounded. In fact, the symbolic methods divide the parameter space into connected components according to singularities, which are a natural measure of the complexity of the solving process. Paper [38] is an example on the crucial difficulties that approximation methods used in parametric optimization by the numerical approaches can face.

Following the work done in [60] by M. Kano and M. C. Smith, where they developed a validated numerical algorithm for the L^∞-norm computation, in [24], C. Chen, M. Moreno Mazza and Y. Xie provided an equivalent study using the theory of *border polynomials,* which makes the presentation of their solution simpler. In fact, they reduced the problem of computing the L^∞-norm of finite-dimensional time-invariant linear systems to the computation of the supremum of the real roots of a univariate polynomial. Then, using *real comprehensive triangular decomposition* and *cylindrical algebraic decomposition*, they generalised their non-parametric approach to the parametric case, i.e., when the transfer matrix depends on parameters. This approach decomposes the space of parameter values into connected open sets, named cells, where the number of real solutions of the system does not change when the parameters vary within the same cell.

Similarly to the work done in [24] and following the study in [60], we first study the computation of the L^∞-norm of finite-dimensional linear time-invariant systems that do not depend on parameters. Then, for the cases depending on parameters, we basically extend our approach using a cylindrical algebraic decomposition. To do that, in the parameter free case, we first further study the properties of the bivariate polynomial $P(x, y)$ that characterizes when the maximal singular value of the transfer matrix is larger than or equal to y. Then, for the study of the L^∞-norm, using the properties of this polynomial, we can propose three different certified symbolic-numeric algorithms that compute the maximal y-projection of the real solutions $(x,\, y)$ of the bivariate polynomial zero-dimensional system $\Sigma = \{P = 0, \frac{\partial P}{\partial x} = 0\}$, where $P \in \mathbb{Z}[x, y]$. We then study the efficiency of the three proposed methods by computing the *asymptotic theoretical complexity* of each of them. This

represents the number of binary operations of an algorithm. Note that *binary complexity* is different form the *arithmetic complexity* that only represents the number of arithmetic operations, i.e., each operation is assumed to have a unit cost whatever the sizes of the operands are.

It is however worthwhile mentioning that the asymptotic theoretical complexity is usually an upper bound in the worst case scenario, which therefore does not necessarily measure the average behavior of the methods, and rarely allows to obtain an objective comparisons on their effective speed of execution. Thus, for studying the efficiency of the algorithms, we rely on one more indicator, namely the time of execution of the implemented methods. Hence, we have implemented the proposed methods using `Maple` tools, and we have compared their average behaviours for some random matrices.

In general, another important indicator of efficiency of an implemented algorithm is the consequent memory occupation. In fact, it is common to see effective methods in terms of computation or arithmetic, or even binary complexity, but without practical advantages due to a prohibitive memory occupation. But, in our study, for each algorithm, we shall only be interested in the first two indicators that are the asymptotic worst case bit complexity and the practical speed.

0.2 Proposed methods

Based on standard computer algebra concepts, methods and implementations, we aim at developing new methods for the study the computation of the L^∞-norm of finite-dimensional linear time-invariant systems:

1. When the system does not depend on parameters, then the problem of L^∞-norm computation is reduced to the computation of the maximal y-projection of the real solutions $(x,\, y)$ of the *zero-dimensional* system of bivariate polynomial equations $\Sigma = \left\{P = 0,\ \frac{\partial P}{\partial x} = 0\right\}$, where $P \in \mathbb{Z}[x, y]$.

2. When the system depends on a set of parameters $\alpha = [\alpha_1, \ldots, \alpha_d] \in \mathbb{R}^d$, we study the system $\Sigma = \left\{P = 0,\ \frac{\partial P}{\partial x} = 0\right\}$, where $P \in \mathbb{Z}[\alpha][x, y]$. The problem of L^∞-norm computation is then reduced to partitioning the parameters space into cells, where above each cell, we can represent the maximal y-projections of the real solutions $(x,\, y)$ of Σ as a real function of α.

In the next paragraph, we briefly introduce the proposed methods, which will be further explained in Section 3.1 and Section 3.2.

0.2.1 Non-parametric case

Given two coprime polynomials P and Q (in our case $Q = \frac{\partial P}{\partial x}$) in $\mathbb{Z}[x, y]$ of degrees bounded by d and of coefficients bitsize bounded by τ, we propose two different approaches for computing the maximal y-projection of the real solutions of Σ. The first approach computes a linear separating form and the second uses the real root counting of a univariate polynomial with algebraic coefficients.

Separating linear form

Two methods used in this dissertation require putting the system in *a local generic position*, i.e., require to finding a separating linear form $y + a\,x$ that defines a shear of the coordinate system $(x,\, y)$, i.e., $(x,\, y) \longmapsto (x,\, t - a\,x)$, so that no two distinct solutions of the sheared polynomial equations system (*sheared system* for short), defined by

$$\Sigma_a = \big\{P(x, t - a\,x) = 0,\, Q(x, t - a\,x) = 0\big\},$$

are horizontally aligned. This approach has long been used in the computer algebra literature. For instance, as shown in [16, 17], a separating linear form $y + a\,x$ with $a \in \{0, \ldots, 2\,d^4\}$ can be computed. As studied in [17], we can then use a *Rational Univariate Representation* (RUR) for the sheared system Σ_a followed by the computation of *isolating boxes* for its real solutions. We simply apply this approach (i.e., the so-called *RUR method*) to the polynomial system associated with the L^∞-norm computation problem and then choose the maximal y-projection of the real solutions of the system. We mention that this value is represented by its *isolating interval* with respect to the univariate polynomial embodying the y-projection of the system solutions, that is the *resultant polynomial* $\mathrm{Res}(P, Q, x)$. The complexity analysis shows that this algorithm performs $\tilde{\mathcal{O}}_B\big(d_y\, d_x^3\, (d_y^2 + d_x\, d_x + d_x\, \tau)\big)$ bit operations in the worst case, where

$$d_x = \max\big(\deg_x(P), \deg_x(Q)\big),\;\; d_y = \max\big(\deg_y(P), \deg_y(Q)\big),$$

and τ is the maximal coefficient bitsize of the polynomials P and Q. We develop this method in Subsection 3.1.1.

Alternatively, we can also localize the maximal y-projection of the real solutions of the polynomial equations system Σ by only applying a linear separating form on the system Σ and without computing isolating boxes for the whole system solutions. The linear separating

form $t = y + s\,x$ proposed in [25] preserves the order of the solutions of the sheared system

$$\Sigma_s = \big\{P(x, , t - s\,x) = 0,\, Q(x, t - s\,x) = 0\big\}$$

with respect to the y-projection of the solutions of the original system Σ. In other words, with this linear separating form $t = y + s\,x$, we obtain:

$$t_1 = y_1 + s\,x_1 < t_2 = y_2 + s\,x_2 \implies y_1 \leq y_2.$$

Thus, the projection of the solutions of Σ_s onto the new separating axis t can be done so that we can simply choose the y-projection corresponding to the maximal t-projection of the real solutions of Σ_s. The drawback of this method lies on the growth of the size of the coefficients of the sheared system for the linear separating form $t = y + s\,x$ due to the large size of s. The complexity analysis shows that this algorithm performs $\tilde{\mathcal{O}}_B\big(d_x^3\, d_y^4\, \tau\, \big(d_x^2 + d_x\, d_y + d_y^2\big)\big)$ bit operations in the worst case. This method is developed in Subsection 3.1.2.

Real roots counting

The third method, developed in Subsection 3.1.3, localizes the maximal y-projection of the system real solutions − denoted by $\bar{y}$ − by first isolating the real roots of the univariate resultant polynomial $\mathrm{Res}(P, \frac{\partial P}{\partial x}, x)$ and then verifying the existence of at least one real root of the *greatest common divisor* $\gcd\big(P(x,\bar{y}), \frac{\partial P}{\partial y}(x,\bar{y})\big) \in \mathbb{R}[x]$ of $P(x,\bar{y})$ and $\frac{\partial P}{\partial x}(x,\bar{y})$. However, the polynomial P, corresponding to our modeled problem, defines a plane real algebraic curve bounded in the y-direction by the value that we are aiming at computing. Thus, a resulting key point is that the number of real roots of $\gcd\big(P(x,\bar{y}), \frac{\partial P}{\partial x}(x,\bar{y})\big)$ is equal to the number of real roots of $P(x,\bar{y})$. Hence, we can simply compute the *Sturm-Habicht sequence* [53] of $P(x,\bar{y})$ for counting the number of its real roots without any consequent overhead. Since the gcd polynomial has a larger size than the polynomial P, then this key point leads to a better complexity in the worst case. We mention that the Sturm-Habicht sequence corresponding to $P(x,\bar{y}) \in \mathbb{R}[x]$ is a *signed subresultant sequence* of the polynomials $P(x,\bar{y}) \in \mathbb{R}[x]$ and its derivative with respect to x. Being already computed to obtain $\mathrm{Res}(P, \frac{\partial P}{\partial x}, x)$, the practical and theoretical complexities are mainly carried by the complexity of evaluating the leading coefficients with respect to x of the subresultant polynomials (called *the principal subresultant coefficients*) over the real value $\bar{y}$. Additionally, if the real plane algebraic curve $P(x,y) = 0$ has no real *isolated singular points*, then the complexity can be further improved since, in this case, the evaluation is done over a rational number instead of an algebraic number. It is worthwhile mentioning that this improvement is theoretically

slight. In fact, for evaluating over an algebraic number, say $\bar{y}$, we are technically evaluating over two rational numbers, that are mainly the endpoints of the isolating interval of the algebraic value $\bar{y}$ as a real root of the resultant polynomial $\mathrm{Res}(P, \frac{\partial P}{\partial x}, x)$. The complexity analysis shows that the proposed algorithm performs $\tilde{\mathcal{O}}_B\big(d_x^4\, d_y^2\, (d_y+\tau)\big)$ bit operations in the worst case and $\tilde{\mathcal{O}}_B(d_x^4\, d_y^2\, \tau)$, when the plane curve $P(x, y) = 0$ has no real isolated singular points.

0.2.2 Parametric case

Given a set of parameters $\alpha = [\alpha_1, \ldots, \alpha_d] \in \mathbb{R}^d$ and a *well-behaved* system

$$\Sigma = \left\{ P = 0\,,\ \frac{\partial P}{\partial x} = 0 \right\},$$

where $P \in \mathbb{Z}[\alpha][x, y]$, we aim at representing the "maximal" y-projection of the real solutions of Σ as a function of the parameters α.

In applications, the structure of the solution set is dependent on the parameters variation, i.e., for a precise parameter values, the system has real solutions, and more generally, for a precise parameter values, the system has a constant number of real solutions. Thus, to solve the well-behaved system Σ, it is crucial to choose a finite number of representative "good" parameter values that cover all possible cases. In contrast, the "bad" parameter values are mainly represented by the so-called *discriminant variety* − proposed by D. Lazard and F. Rouillier [63] as a generalization of the well-known discriminant of a univariate polynomial − which defines the parameters leading to non-generic solutions of the system.

In Section 3.2, we propose an algorithm that, using a *cylindrical algebraic decomposition*, decomposes the parameter space, mainly the space of "good" parameter values, into a finite disjoint union of connected open sets (called *cells*) such that the system has a constant number of real solutions and the order of the y-projection of the solutions does not change when a parameter value varies within the same cell. We are then able to represent the searched value as a function of the parameters over a given cell.

Chapter 1

Problem Motivation From Control Theory

In this chapter, we first present definitions and reminders on *finite-dimensional continuous-time linear time-invariant control systems*, i.e., control systems defined by linear differential equations with constant coefficients [20, 39, 101, 107]. Then, we show that the *maximum gain* of a continuous-time linear time-invariant control system is equal to the L^∞-norm of its transfer matrix [39, 101, 107]. The computation of the L^∞-norm of a transfer matrix can be reduced to the study of the critical points of a real plane algebraic curve [60]. Finally, we give a short overview of robust control theory where the L^∞-norm plays a fundamental role.

1.1 Definition of a linear time-invariant system

Definition 1.1. A *state-space representation* of a finite-dimensional linear time-invariant (LTI) dynamical system is given by

$$\begin{cases} \dot{x}(t) = A\,x(t) + B\,u(t), \\ y(t) = C\,x(t) + D\,u(t), \\ x(t_0) = x_0, \end{cases} \tag{1.1}$$

where $\dot{x}(t) = \frac{dx(t)}{dt}$ denotes the time derivative of the *state vector* x with respect to the continuous-time variable t, u is the *input*, y the *output*, $t_0 \in \mathbb{R}$ is the *initial instant* and x_0 the *initial state*, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$ and $D \in \mathbb{R}^{p\times m}$.

This state-space representation is referred to as (A, B, C, D).

A control system with single-input ($m = 1$) and single-output ($p = 1$) is called a *single-input and single-output* (SISO) system. Otherwise, it is called a *multiple-input and multiple-output* (MIMO) system (if $m = 1$ and $p > 1$ (resp., $m > 1$ and $p = 1$), it is sometimes also called SIMO (resp., MISO)).

Without loss of generality, we can assume that $t_0 = 0$ and $u = 0$ for $t < 0$.

By the standard Cauchy theorem, given a regular vector-valued function u, (1.1) has a unique regular state x and a unique regular output y. More precisely, integrating (1.1) by means of the *variation of constants method*, we obtain:

$$\begin{cases} x(t) = e^{A\,t}\,x_0 + \displaystyle\int_0^t e^{A\,(t-\tau)}\,B\,u(\tau)\,d\tau, \\ y(t) = C\,e^{A\,t}\,x_0 + C\displaystyle\int_0^t e^{A\,(t-\tau)}\,B\,u(\tau)\,d\tau + D\,u(t). \end{cases} \tag{1.2}$$

Through the choice of the input u, the behavior of the state x, and thus of the input y can be influenced to achieve desired goals (e.g., reaching a given state $x_T \in \mathbb{R}^n$ at given time $t = T$ (*controllability*), design a feedback law between x and u (*state feedback*) or between y and u (*output feedback*) which achieves a certain goal such that stabilizing an unstable system (*stabilization problems*), minimizing a given energy integral (*optimal control*), etc.). The study of systems of the form (1.1) and their extensions − including differential time-delay, partial differential equations, recurrence relations, etc. − is at the heart of an engineering science called *automatic control theory* and at a mathematical theory called *control theory*. See, e.g., [20, 39, 101, 107] and the references therein.

Example 1. To better illustrate the state-space representation, consider the example of a mass connected to a spring of spring constant k and to a damper of damping coefficient b. Let $f(t)$ be the external force applied to the mass m on a certain time $t > 0$, that leads to a variation in its position $z(t)$.

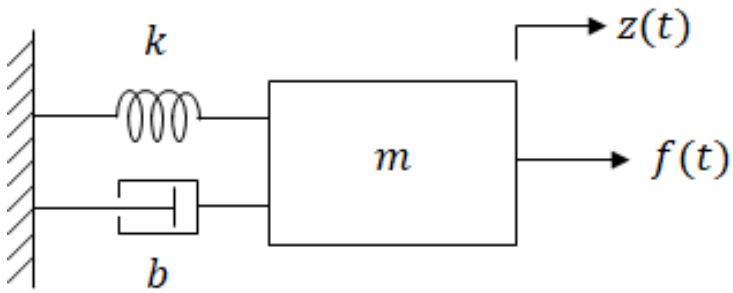

Figure 1.1: Spring mass damper system

Newton's second law shows that z satisfies the ordinary differential equation:

$$m\,\frac{d^2 z(t)}{dt^2} = f(t) - b\,\frac{dz(t)}{dt} - k\,z(t). \tag{1.3}$$

The state variables $x_1(t)$ and $x_2(t)$ can be chosen to be the position $z(t)$ and the velocity $\dot{z}(t) = \dfrac{dz(t)}{dt}$ of the mass.

By rewriting (1.3) accordingly, we obtain $m\,\dot{x}_2(t) = u(t) - b\,x_2(t) - k\,x_1(t)$, where $\dot{x}_1(t) = x_2(t)$. We can then rearrange equations to obtain the following linear system of ordinary differential equations:

$$\begin{cases} \dot{x}_1(t) = \dot{z}(t) = x_2(t), \\ \dot{x}_2(t) = -\dfrac{k}{m}\,x_1(t) - \dfrac{b}{m}\,x_2(t) + \dfrac{1}{m}\,u(t), \\ y(t) = x_1(t). \end{cases}$$

Finally, in matrix-vector, we obtain the following linear system:

$$\begin{cases} \begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\dfrac{k}{m} & -\dfrac{b}{m} \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + \begin{pmatrix} 0 \\ \dfrac{1}{m} \end{pmatrix} u(t), \\ y(t) = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}. \end{cases}$$

Hence, the state-space representation of this spring mass damper system is defined by $(A,\, B,\, C,\, D)$, where:

$$A = \begin{pmatrix} 0 & 1 \\ -\dfrac{k}{m} & -\dfrac{b}{m} \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ \dfrac{1}{m} \end{pmatrix}, \quad C = \begin{pmatrix} 1 & 0 \end{pmatrix}, \quad D = 0. \tag{1.4}$$

Let $\mathscr{L}(f)$ denote the *Laplace transform* of a real-valued function f with support on $\mathbb{R}_+ := \{t \in \mathbb{R} \mid t > 0\}$ defined by

$$\mathscr{L}(f)(s) = \int_0^{+\infty} e^{-s\,t}\, f(t)\, dt,$$

where $s \in \mathbb{C}$ (the *Laplace variable*) is such that $\mathrm{Re}(s) > \alpha$ for a certain abscissa α. Note that the Laplace transform is well-defined for $f \in L^1(\mathbb{R}_+)$, where $L^1(\mathbb{R}_+)$ denotes the *Banach space* of all the *Lebesgue measurable functions* on $\mathbb{R}_+$ for which the absolute value is *Lebesgue*

integrable, i.e.:

$$\| f \|_1 := \int_0^{+\infty} |f(t)|\, dt < +\infty.$$

More generally, the Laplace transform can be defined for *tempered distributions*. For instance, if H denotes the *Heaviside distribution*, i.e., $H(t) = 1$ for $t > 0$ and 0 for $t < 0$, then we have $\mathscr{L}(H) = s^{-1}$, and if δ is the *Dirac distribution*, i.e., $\delta = \dot{H}$ in the sense of the distribution theory, then we have $\mathscr{L}(\delta) = 1$.

Using the standard identity $\mathscr{L}(\dot{x})(s) = s\,\mathscr{L}(x)(s) - x(0)$ for regular functions x (obtained by an integration by parts) and assuming that $x_0 = x(0) = 0$, the relation between an input vector-valued function u and the corresponding output vector-valued function y defined by (1.1) takes the form of

$$Y(s) = G(s)\, U(s),$$

where $U = \mathscr{L}(u)$ and $Y = \mathscr{L}(y)$ are the Laplace transforms of u and y, and:

$$G(s) = C\,(s\, I_n - A)^{-1}\, B + D. \tag{1.5}$$

The matrix G is called the *transfer matrix* of (1.1). If $p = m = 1$, i.e., (1.1) is SISO, then the function G is called the *transfer function* of (1.1).

For a square matrix $M \in \mathbb{R}[s]^{n\times n}$, we denote the *determinant* of M by $\det(M)$ and by $\mathrm{adj}(M)$ the *adjugate matrix* of M defined as the transpose of the cofactor matrix of M. Hence, we have

$$(s\, I_n - A)^{-1} = \frac{1}{\det(s\, I_n - A)}\, \mathrm{adj}(s\, I_n - A), \tag{1.6}$$

where $s\, I_n - A$ and $\mathrm{adj}(s\, I_n - A)$ both belong to $\mathbb{R}[s]^{n\times n}$, which shows that:

$$G(s) = \frac{1}{\det(s\, I_n - A)}\, C\, \mathrm{adj}(s\, I_n - A)\, B + D \in \mathbb{R}(s)^{p\times m}. \tag{1.7}$$

Hence, the entries of the matrix G of (1.1) are real rational functions of s. In the control theory literature, G is simply called a *(real) rational matrix*.

Using (1.7) and the fact that the polynomial matrix $\mathrm{adj}(s\, I_n - A)$ is defined by the $(n-1)\times(n-1)$-minors of $s\, I_n - A$, we can easily check that all the rational function entries of G are *proper* in the sense that the degree of their numerator is less than or equal to the degree of their denominator. Note that the degrees of the denominators of the entries of G can be strictly less than n due to simplifications occurring in (1.7). Hence, we have $G(\infty) := \lim_{|s|\to+\infty} G(s) = D$. This definition can be extended to general matrices with

rational function entries.

Definition 1.2. A matrix $T \in \mathbb{R}(s)^{p\times m}$ is *proper* (resp., *strictly proper*) if $T(\infty)$ is finite, i.e., there exists $T_\infty \in \mathbb{R}^{p\times m}$ such that $T(\infty) = T_\infty$ (resp., $T(\infty) = 0$).

Example 2. Let us compute the transfer function G of the linear system defined in Example 1. Using the matrices defined in (1.4), we first get

$$s\,I_2 - A = \begin{pmatrix} s & -1 \\ \dfrac{k}{m} & s + \dfrac{b}{m} \end{pmatrix} \;\Rightarrow\; (s\,I_2 - A)^{-1} = \frac{1}{m\,s^2 + b\,s + k} \begin{pmatrix} m\,s + b & m \\ -k & m\,s \end{pmatrix},$$

which yields the following transfer function:

$$G(s) = \frac{1}{m\,s^2 + b\,s + k} \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} m\,s + b & m \\ -k & m\,s \end{pmatrix} \begin{pmatrix} 0 \\ \dfrac{1}{m} \end{pmatrix} = \frac{1}{m\,s^2 + b\,s + k}.$$

G is strictly proper since $G(\infty) = 0$. Finally, if we now observe both x_1 and x_2, i.e., the displacement and the velocity of the spring, then we have $y = C\,x$, where $C = I_2$ and $x = (x_1 \quad x_2)^T$, which yields the strictly proper transfer matrix:

$$G'(s) = \begin{pmatrix} \dfrac{1}{m\,s^2 + b\,s + k} \\ \dfrac{s}{m\,s^2 + b\,s + k} \end{pmatrix}.$$

1.2 Stability of LTI systems

We first study the asymptotic behavior of the solutions of $\dot{x}(t) = A\,x(t)$, i.e., of (1.1) where $u = 0$. Hence, setting $u = 0$ for $t \geq 0$ in (1.2), we get that the solution $x(t) = e^{A\,t}\,x_0$ tends to 0 for t tending to $+\infty$, i.e., $\lim_{t\to+\infty} x(t) = 0$, if and only if all the eigenvalues of $A \in \mathbb{R}^{n\times n}$ have strictly negative real parts. Hence, we can introduce the following definition.

Definition 1.3. A matrix $A \in \mathbb{R}^{n\times n}$ is said to be *(asymptotically) stable* if all of its eigenvalues have strictly negative real part, i.e., all the eigenvalues of A belong to the open left half-plane $\mathbb{C}_- := \{s \in \mathbb{C} \mid \mathrm{Re}(s) < 0\}$.

Using (1.7), the denominators of the entries of G are factors of the polynomial $\det(s\,I_n - A)$, whose zeros of $\det(s\,I_n - A)$ are the eigenvalues of A. Hence, the poles of the entries of G are included in the eigenvalues of A. In particular, if A is stable, then all the poles of the entries of G have strictly negative real part.

Definition 1.4. A matrix $T \in \mathbb{R}(s)^{p\times m}$ is said to be *stable* if all the entries of T have poles in $\mathbb{C}_- := \{s \in \mathbb{C} \mid \mathrm{Re}(s) < 0\}$.

Example 3. We consider again Examples 1 and 2. We can check that $\det(s\,I_2 - A) = s^2 + b\,s/m + k/m$. When the damping coefficient $b > 0$ is small, i.e., when the discriminant $\Delta := (b^2 - 4\,k\,m)/m^2$ of $\det(s\,I_2 - A)$ is strictly negative, the two complex solutions of $\det(s\,I_2 - A)$ are defined by

$$s_\pm = \frac{-b \pm \delta\, i}{2\,m},$$

where $\delta = \sqrt{4\,k\,m - b^2}$. When $\Delta = 0$, $s_\pm = -b/2\,m$. When $b > 0$ is large, i.e., when the discriminant $\Delta = (b^2 - 4\,k\,m)/m^2$ of $\det(s\,I_2 - A)$ is strictly positive, the two real solutions of $\det(s\,I_2 - A)$ are defined by $s_\pm = (-b \pm \sqrt{\Delta})/2$. In this case, i.e., when $\Delta > 0$, we can easily check that the real solutions are both strictly negative by noticing the positive sign of their product k/m. Consequently, in the three cases $\Delta < 0$, $\Delta = 0$, and $\Delta > 0$, the solutions have strictly negative real parts, i.e., $s_\pm \in \mathbb{C}_-$, for all $b > 0$. Thus, the linear system $\dot{x}(t) = A\,x(t)$, where A is defined by (1.4), is asymptotically stable for all $b > 0$.

We now study the stability of the solutions of the linear system (1.1), i.e., when u is not reduced to 0. To do that, we first state a few remarks. The transfer matrix G – given by (1.5) – is the Laplace transform of the *input-output operator*, which maps an input u to an output y, defined by:

$$y(t) = C \int_0^t e^{A\,(t-\tau)}\, B\, u(\tau)\, d\tau + D\, u(t). \tag{1.8}$$

If $\mathscr{L}^{-1}$ denotes the *inverse Laplace transform* and if we set

$$K := \mathscr{L}^{-1}(G) = C\, e^{A\,t}\, B\, H(t) + D\,\delta, \tag{1.9}$$

where H (resp., δ) is the Heaviside (resp., Dirac) distribution, then we can rewrite the above input-output operator as the *convolution operator* $y = K \star u$, namely:

$$\begin{aligned} y(t) &= \int_0^{+\infty} C\, e^{A\,(t-\tau)}\, B\, H(t-\tau)\, u(\tau)\, d\tau + D\,\delta \star u(t) \\ &= \int_0^t C\, e^{A\,(t-\tau)}\, B\, u(\tau)\, d\tau + D\, u(t). \end{aligned} \tag{1.10}$$

Indeed, Laplace transform maps convolutions to products and δ is a unit for the convolution product, i.e., $\delta \star u = u$. To simplify the notations, we shall sometimes identify the above convolution operator with its kernel matrix K.

Example 4. We consider again Examples 1, 2 and 3. Let us compute the kernel matrix K of the convolution operator associated to the system defining Example 1. Using (1.9), we have to compute $e^{A\,t}$. By *Cayley–Hamilton theorem*, we have $e^{A\,t} = c_0(t)\,I_2 + c_1(t)\,A$. Using the notations of Example 3, the characteristic polynomial of A is $s^2 + b\,s/m + k/m = (\lambda - s_+)(\lambda - s_-)$. If P denotes the matrix formed by the eigenvectors respectively associated with the eigenvalues s_+ and s_- (i.e., $(1 \quad s_+)^T$, resp., $(1 \quad s_-)^T$), then we have:

$$\begin{aligned}\begin{pmatrix} e^{s_+\,t} & 0 \\ 0 & e^{s_-\,t} \end{pmatrix} &= P^{-1}\,e^{A\,t}\,P = P^{-1}\,\left(c_0(t)\,I_2 + c_1(t)\,A\right)\,P \\ &= c_0(t)\,I_2 + c_1(t)\begin{pmatrix} s_+ & 0 \\ 0 & s_- \end{pmatrix}.\end{aligned}$$

In other words, c_0 and c_1 satisfy the following linear system:

$$\begin{cases} e^{s_+t} = c_0(t) + c_1(t)\,s_+, \\ e^{s_-t} = c_0(t) + c_1(t)\,s_-. \end{cases}$$

Solving the above linear system, we then obtain:

$$\begin{cases} c_0(t) = \dfrac{s_-\,e^{s_+\,t} - s_+\,e^{s_-\,t}}{s_- - s_+}, \\ c_1(t) = \dfrac{e^{s_-\,t} - e^{s_+\,t}}{s_- - s_+}. \end{cases}$$

Hence, we obtain:

$$K(t) = C\,e^{A\,t}\,B\,H(t) = \frac{c_1(t)}{m}\,H(t).$$

Alternatively, we can write:

$$\begin{aligned}G(s) &= \frac{1}{m\,s^2 + b\,s + k} = \frac{1}{m\,(\lambda - s_+)(\lambda - s_-)} = \frac{1}{m\,(s_- - s_+)}\left(\frac{1}{s - s_-} - \frac{1}{s - s_+}\right) \\ &= \mathscr{L}^{-1}\left(\frac{1}{m\,(s_- - s_+)}\,\left(e^{s_-\,t} - e^{s_+\,t}\right)\,H(t)\right).\end{aligned}$$

Since $G = \mathscr{L}(K)$, we find again that $K(t) = c_1\,H(t)/m$

If A is stable, then the entries of the matrix K, defined by (1.9), belong to $\mathcal{A} := L^1(\mathbb{R}_+) \oplus \mathbb{R}\,\delta := \{f + \lambda\,\delta \mid f \in L^1(\mathbb{R}_+),\ \lambda \in \mathbb{R}\}$. $\mathcal{A}$ is a *Banach algebra (Wiener-Laplace*

algebra) for the sum and the convolution $\star$, and for the norm:

$$\| f + \lambda\, \delta \|_{\mathcal{A}} := \| f \|_1 + |\lambda|.$$

For more details, see Section 6.4.1 of [101] and the references therein.

To simplify the account, we first suppose that $p = m = 1$, i.e., (1.1) is SISO, so that $K \in \mathcal{A}$. If $L^\infty(\mathbb{R}_+)$ denotes the Banach space formed by the Lebesgue measurable real-valued functions on $\mathbb{R}_+$ that are essentially bounded, i.e.,

$$\| f \|_\infty := \text{ess.sup}_{t \in \mathbb{R}_+} |f(t)] < +\infty,$$

then the input-output operator (1.8) or, equivalently, (1.9) clearly satisfies:

$$\| y \|_\infty \leq \| K \|_{\mathcal{A}} \| u \|_\infty .$$

See, e.g., Theorem 30 (page 298) of [101]. Hence, if A is stable, then a bounded input u yields a bounded output y. We then say that (1.9) is *bounded input bounded output* (BIBO). Moreover, we get

$$\| K \|_{\mathcal{L}(L^\infty, L^\infty)} := \sup_{0 \neq u \in L^\infty(\mathbb{R}_+)} \frac{\| y \|_\infty}{\| u \|_\infty} \leq \| K \|_{\mathcal{A}},$$

which shows that the *norm of the input-output operator* for the ∞-norm is bounded by $\| K \|_{\mathcal{A}}$. In fact, the equality can be proved (see, e.g., Point 3 on page 301 of [101]). Hence, $\| K \|_{\mathcal{A}}$ corresponds to the *maximum gain* of the input-output operator (1.8) for the ∞-norm, i.e., its "maximum amplification". It is an important information about the system (1.1). Its computation requires an integral calculation, which might be difficult.

Example 5. Let us consider the kernel matrix K defined in Example 4. Let us study if K is Lebesgue integrable when $\Delta < 0$ (see Example 3). Using the notations of Example 3, we have

$$\begin{aligned}
\| K \|_1 &= \int_0^{+\infty} \left| \frac{e^{s_- t} - e^{s_+ t}}{m\,(s_- - s_+)} \right| dt = \int_0^{+\infty} \left| \frac{e^{\frac{(-b-\delta\, i)}{2\, m}\, t} - e^{\frac{(-b+\delta\, i)}{2\, m}\, t}}{\delta\, i} \right| dt \\
&= \frac{1}{\delta} \int_0^{+\infty} \left| e^{\frac{-b\, t}{2\, m}} \left(\cos\left(\frac{\delta\, t}{2m}\right) - i\, \sin\left(\frac{\delta\, t}{2\, m}\right) - \cos\left(\frac{\delta\, t}{2m}\right) - i\, \sin\left(\frac{\delta\, t}{2\, m}\right) \right) \right| dt \\
&= \frac{2}{\delta} \int_0^{+\infty} e^{\frac{-b\, t}{2\, m}} \left| \sin\left(\frac{\delta\, t}{2\, m}\right) \right| dt \\
&\leq \frac{2}{\delta} \int_0^{+\infty} e^{\frac{-b\, t}{2\, m}}\, dt = \frac{2}{\delta} \left[\frac{e^{\frac{-b\, t}{2\, m}}}{\frac{-b}{2\, m}} \right]_0^{+\infty} = \frac{4\, m}{\delta\, b},
\end{aligned}$$

since $b > 0$ ((1.4) is stable by Example 3), which shows that $K \in \mathcal{A}$ and:

$$\| K \|_{\mathcal{L}(L^\infty, L^\infty)} \leq \| K \|_{\mathcal{A}} = \| K \|_1 \leq 4 \frac{m}{\delta\, b}.$$

Let us now consider the *Hilbert space* $L^2(\mathbb{R}_+)$ defined by the Lebesgue measurable real-valued functions on $\mathbb{R}_+$ that are bounded for the 2-norm:

$$\| f \|_2 := \sqrt{\int_0^{+\infty} |f(t)|^2 \, dt}.$$

If A is stable, i.e., $K \in \mathcal{A}$, then it can be proved that $y = K \star u \in L^2(\mathbb{R}_+)$ for all $u \in L^2(\mathbb{R}_+)$. Hence, an input u with a finite energy yields an output y with a finite energy. Moreover, we have:

$$\forall\, u \in L^2(\mathbb{R}_+), \quad \| y \|_2 \leq \| K \|_{\mathcal{A}} \| u \|_2 \,. \tag{1.11}$$

Note that a similar result holds for the Banach space $L^p(\mathbb{R}_+)$ for $1 \leq p \leq \infty$ (which shows that $L^p(\mathbb{R}_+)$ is an $\mathcal{A}$-*module for* $1 \leq p \leq \infty$). See, e.g., Theorem 30 on page 298 of [101]. These results yield the following definition.

Definition 1.5. An LTI system (1.2) of state-space representation (A, B, C, D) is *(asymptotically/input-output) stable* if A is stable.

Example 6. The poles of G (resp., G') defined in Example 2 are defined by the two zeros of the polynomial $\det(s\, I_2 - A)$, whose real parts are strictly negative (see Example 3). Hence, the control linear system (1.4) is stable.

Since $L^2(\mathbb{R}_+)$ plays an important role in practice, it is important to compute the operator norm of the input-output operator (1.9) for the 2-norm, namely:

$$\| K \|_{\mathcal{L}(L^2, L^2)} := \sup_{0 \neq u \in L^2(\mathbb{R}_+)} \frac{\| y \|_2}{\| u \|_2}. \tag{1.12}$$

This operator norm corresponds to the *maximum energy gain* of (1.9) or, equivalently, of the linear system (1.2). Note that the identity (1.11) shows that:

$$\| K \|_{\mathcal{L}(L^2, L^2)} \leq \| K \|_{\mathcal{A}} \,. \tag{1.13}$$

The computation of the norm (1.12) has many applications in control theory, especially in the so-called *robust control theory* and more particularly in the H^∞ *control* [39, 107, 33].

The goal of this dissertation to investigate this question for general finite-dimensional linear time-invariant systems.

Remark 1.1. More general classes of linear control systems (e.g., differential time-delay, partial differential equations) can be defined by means of more general classes of transfer matrices G (e.g., non rational transfer matrices) and of convolution kernels K. We shall only consider here rational transfer matrices.

We state again a standard result of control theory.

Theorem 1.1. Any proper rational matrix $T \in \mathbb{R}(s)^{p\times m}$ can be *realized* by an LTI system (1.1), i.e., there exist $n \geq 1$ and $(A,\, B,\, C,\, D)$ such that:

$$T = C\,(s\,I_n - A)^{-1}\,B + D.$$

Moreover, the state-space representation $(A,\, B,\, C,\, D)$ of T can be chosen to be both *controllable* and *observable*, i.e., such that we have respectively:

$$\operatorname{rank}_{\mathbb{R}}(B \quad A\,B \quad A^2\,B \;\ldots\; A^{n-1}\,B) = n, \quad \operatorname{rank}_{\mathbb{R}}\begin{pmatrix} C \\ C\,A \\ \vdots \\ C\,A^{n-1} \end{pmatrix} = n.$$

Such a realization is called *minimal*. For a minimal realization $(A,\, B,\, C,\, D)$ of T, the poles of the entries of T are exactly the eigenvalues of the matrix A.

According to Theorem 1.1, any proper rational matrix is the transfer matrix of a certain linear time-invariant system defined by a state-space representation (1.1). This dissertation aims at computing the operator norm (1.12) for an LTI system defined by a stable and proper rational transfer matrix G. In the next section section, we shall explain how this computation is related to the L^∞-norm computation of the transfer matrix G.

1.3 Maximum energy gain and L^∞-norm

We are going to derive a more tractable characterization of the operator norm (1.12). To do that, we need to introduce a few more functional spaces.

Let $\overline{\mathbb{C}_+} := \{s \in \mathbb{C} \mid \operatorname{Re} s \geq 0\}$ be the closed right half-plane, i.e., the complement of $\mathbb{C}_-$ in $\mathbb{C}$. If $d \in \mathbb{R}[s]$, then complex zero set of d is denoted by:

$$V_{\mathbb{C}}(\langle d \rangle) := \{s \in \mathbb{C} \mid d(s) = 0\}.$$

Finally, if p_1, p_2 are two univariate polynomials with coefficients in a field, then $\gcd(p_1, p_2)$ denotes the *greatest common divisor* of p_1 and p_2 (see Section 2.2).

Using Definitions 1.2 and 1.4, we can check again that the set of SISO proper and stable transfer functions form an algebra over $\mathbb{R}$.

Definition 1.6. Let RH_∞ be the $\mathbb{R}$-algebra of proper and stable real rational functions, namely:

$$RH_\infty := \left\{ \frac{n}{d} \mid n, d \in \mathbb{R}[s],\ \gcd(n, d) = 1,\ \deg_s(n) \leq \deg_s(d), V_{\mathbb{C}}(\langle d \rangle) \cap \overline{\mathbb{C}_+} = \emptyset \right\}.$$

Example 7. For instance, $1/(s+1)$, $(s-1)/(s+1)$ and $(s-1)/(s+1)^2$ belong to RH_∞ but s, $s^2/(s+1)$, $1/(s-1)$ and $1/(s^2+1)$ do not belong to RH_∞.

A stable and proper rational transfer matrix $G \in \mathbb{R}(s)^{p \times m}$ satisfies:

$$G \in RH_\infty^{p \times m}.$$

An element g of RH_∞ is holomorphic and bounded on $\mathbb{C}_+$, i.e.:

$$\| g \|_\infty := \sup_{s \in \mathbb{C}_+} |g(s)| < +\infty.$$

Hence, RH_∞ is an $\mathbb{R}$-sub-algebra of the *Hardy algebra* $H^\infty(\mathbb{C}_+)$ of bounded holomorphic functions on $\mathbb{C}_+$ [33, 107]. Note that $H^\infty(\mathbb{C}_+)$ is a *Banach algebra*, namely, a Banach space and a $\mathbb{C}$-algebra which satisfies $\| f g \|_\infty \leq \| f \|_\infty \| g \|_\infty$ (i.e., the multiplication is continuous) (see, e.g., [33] and the references therein).

In the control theory literature, $H^\infty(\mathbb{C}_+)$ is usually denoted by $H_\infty(\mathbb{C}_+)$.

Example 8. If $\tau > 0$, then $e^{-\tau s} \in H^\infty(\mathbb{C}_+)$, $\| e^{-\tau s} \|_\infty = 1$, but $e^{-\tau s} \notin RH_\infty$. Similarly, $e^{-\tau s}/(s+1) \in H^\infty(\mathbb{C}_+)$ but it does not belong to RH_∞. Note that $e^{-\tau s}/(s+1)$ corresponds to the transfer function of the time-invariant infinite-dimensional linear system (*differential time-delay system*):

$$\dot{x}(t) + x(t) = u(t - \tau).$$

For an introduction to the theory of infinite-dimensional systems (i.e., systems defined by partial differential equations or by differential time-delay equations) and the H^∞ methods for this class of systems, see [33] and the references therein.

If $g \in H^\infty(\mathbb{C}_+)$, the *maximum modulus principle* of complex analysis yields:

$$\| g \|_\infty = \text{ess.sup}_{\omega \in \mathbb{R}} |g(i\,\omega)|.$$

If $g \in RH_\infty$, then we can consider its restriction $g_{|i\,\mathbb{R}}$ on the imaginary axis $i\,\mathbb{R} := \{i\,\omega \mid \omega \in \mathbb{R}\}$. Clearly, $g_{|i\,R} \in L^\infty(i\,\mathbb{R})$, where $L^\infty(i\,\mathbb{R})$ denotes the Banach space of essentially bounded Lebesgue measurable functions on $i\,\mathbb{R}$. More precisely, $g_{|i\,\mathbb{R}}$ belongs to the $\mathbb{R}$-subalgebra RL_∞ of $L^\infty(i\,\mathbb{R})$ of the real rational functions on the imaginary axis which are proper and have no poles on $i\,\mathbb{R}$, i.e.,

$$RL_\infty := \left\{ \frac{n(i\,\omega)}{d(i\,\omega)} \mid n, d \in \mathbb{R}[i\,\omega],\ \gcd(n,d) = 1, \deg_\omega(n) \leq \deg_\omega(d), V_{\mathbb{C}}(\langle d \rangle) \cap i\,\mathbb{R} = \emptyset \right\},$$

or simply the algebra of real rational functions with no poles on $i\,\mathbb{P}^1(\mathbb{R})$, where:

$$\mathbb{P}^1(\mathbb{R}) := \mathbb{R} \cup \{\infty\}.$$

We can now come back to the problem studied in this dissertation, namely, the characterization of the operator norm (1.12) for a stable SISO linear system (1.1), i.e., for $G = \mathscr{L}(K) \in RH_\infty$. A standard result on H^∞ asserts that:

$$\| K \|_{\mathcal{L}(L^2,L^2)} := \sup_{0 \neq u \in L^2(\mathbb{R}_+)} \frac{\| y \|_2}{\| u \|_2} = \| G \|_\infty := \sup_{s \in \mathbb{C}_+} |G(s)| = \sup_{\omega \in \mathbb{R}} |G_{|i\,\mathbb{R}}(i\,\omega)|. \tag{1.14}$$

Hence, the maximum energy gain of the convolution operator (1.10), i.e., of (1.8), is exactly the maximum of the modulus of the Laplace transform G of its kernel K, i.e., of $G = \mathscr{L}(K)$, or equivalently, the *maximum peak* of its continuous restriction $G_{|i\,\mathbb{R}}$ over all the frequencies $i\,\omega$ with $\omega \in \mathbb{P}^1(\mathbb{R})$, i.e.:

$$\sup_{\omega \in \mathbb{R}} |G_{|i\,\mathbb{R}}(i\,\omega)| = \max_{z \in i\,\mathbb{P}^1(\mathbb{R}) = i\,\mathbb{R} \cup \{i\,\infty\}} |G(z)| = \max\{\max_{z \in i\,\mathbb{R}} |G(z)|,\ |G(i\,\infty)|\}.$$

In control theory literature, the graph of the function $\omega \in \mathbb{R} \longrightarrow |G(i\,\omega)|$ is called *Bode magnitude plot* and the ∞-norm is then the peak value of this graph.

More generally, if $G \in RL_\infty$ then, as explained above, $\| G \|_\infty$ is the supremum of the

continuous function $\omega \in \mathbb{P}^1(\mathbb{R}) := \mathbb{R} \cup \{\infty\} \longmapsto |G(i\,\omega)|$, and thus, we have

$$\| G \|_\infty = \max_{\omega \in \mathbb{P}^1(\mathbb{R})} |G(i\,\omega)|,$$

i.e., $\| G \|_\infty = \max\{|G(i\,\infty)|, \gamma_{\max}\}$, where:

$$\gamma_{\max} := \max_{\omega \in \mathbb{R}} |G(i\,\omega)| = \max\left\{\gamma \geq 0 \in \mathbb{R} \mid \exists\, \omega \in \mathbb{R} : \gamma^2 = |G(i\,\omega)|^2\right\}.$$

Hence, $\gamma > \| G \|_\infty$ if and only if $\gamma > |G(i\,\infty)|$ and $\Phi_\gamma(i\,\omega) := \gamma^2 - |G(i\,\omega)|^2 \neq 0$ for all $\omega \in \mathbb{R}$. This result will be generalized to the matrix case in Proposition 1.1.

Using $|G(i\,\omega)|^2 = G(-i\,\omega)\,G(i\,\omega) \in \mathbb{R}(\omega^2)$, a first method for computing $\| G \|_\infty$ consists in first computing the zeros of the numerator of $\frac{\mathrm{d}|G(i\,\omega)|^2}{\mathrm{d}\omega}$, then evaluating $|G(i\,\omega)|$ on these zeros and finally choosing the maximal occurring value, that to say $\bar{\gamma}$, and (iii) $\| G \|_\infty = \max\{|G(i\,\infty)|, \bar{\gamma}\}$.

More explicitly, if we write G as $G = a/b$, where $a, b \in \mathbb{R}[s]$ are two *coprime polynomials*, namely, $\gcd(a, b) = 1$, $q = \deg_s(a) \leq r = \deg_s(b)$, and b does not vanish on $i\,\mathbb{R}$, then $G(i\,\infty) = 0$ if $q < r$ (i.e., if G is strictly proper) or $G(i\,\infty) = a_r/b_r$ if $q = r$ (i.e., if G is proper), where $a_r = \mathrm{Lc}_s(a)$ (resp., $b_r = \mathrm{Lc}_s(b)$) is the *leading coefficient* of the polynomial a (resp., b) in s. Moreover, we can write $|G(i\,\omega)|^2 = |a(i\,\omega)|^2/|b(i\,\omega)|^2 = N(\omega)/D(\omega)$, where $N, D \in \mathbb{R}[\omega^2]$ are coprime, i.e., $\gcd(D, N) = 1$. Since $b(i\,\omega)$ has not real roots, $D(\omega) \neq 0$ for all $\omega \in \mathbb{R}$. If we note $\mathcal{Z} := \{\omega \in \mathbb{R} \mid N'(\omega)\,D(\omega) - N(\omega)\,D'(\omega) = 0\}$, then we get:

$$\| G \|_\infty = \max\{|G(i\,\infty)|, \bar{\gamma}\}, \quad \bar{\gamma} := \max_{\omega \in \mathcal{Z}} \left\{(N(\omega)/D(\omega))^{1/2}\right\}.$$

Remark 1.2. If $\mathcal{Z} \cap V_{\mathbb{R}}(\langle D'(\omega)\rangle) = V_{\mathbb{R}}(\langle N'(\omega)\,D(\omega), D'(\omega)\rangle) = \emptyset$, note that:

$$\bar{\gamma} = \max_{\omega \in \mathcal{Z}} \left\{(N'(\omega)/D'(\omega))^{1/2}\right\}.$$

Example 9. If $G = (2\,s + 1)/(s + 1)$, then $N(\omega) = 4\,\omega^2 + 1$, $D(\omega) = \omega^2 + 1$, $\mathcal{Z} = \{0\}$, $\bar{\gamma} = (N(0)/D(0))^{1/2} = 1$, $|G(i\,\infty)| = 2$, and thus, $\| G \|_\infty = \max\{2, \bar{\gamma}\} = 2$.

Example 10. If $G = (s + 1)/(s - 1)$, then $G \in RL_\infty$ but $G \notin RH_\infty$ since G has a pole at $1 \in \mathbb{C}_+$. We have $|G(i\,\omega)|^2 = |1 + i\,\omega|^2/|1 - i\,\omega|^2 = 1$, and thus, $N = 1$, $D = 1$, $\mathcal{Z} = \mathbb{R}$, $\bar{\gamma} = 1$ and $|G(i\,\infty)| = 1$, which finally shows that $\| G \|_\infty = 1$.

Finally, let us give an example which depends on parameters.

Example 11. In Example 5, we obtained an upper bound for $\| K \|_{\mathcal{L}(L^\infty,L^\infty)}$, where K is the convolution kernel defined in Example 4 of the system defined in Example 1. Now, according to (1.14) and using $G = \mathscr{L}(K)$, we have:

$$\begin{aligned}\| K \|_{\mathcal{L}(L^2,L^2)} &= \left\| \frac{1}{m\,s^2 + b\,s + k} \right\|_\infty = \sup_{\omega\in\mathbb{R}} \left| \frac{1}{m\,(i\,\omega)^2 + b\,i\,\omega + k} \right| \\ &= \sup_{\omega\in\mathbb{R}} \frac{1}{\sqrt{(k - m\,\omega^2)^2 + (b\,\omega)^2}} = \left(\inf_{\omega\in\mathbb{R}} \sqrt{(k - m\,\omega^2)^2 + (b\,\omega)^2} \right)^{-1}.\end{aligned}$$

Hence, $N = 1$, $D = (k - m\,\omega^2)^2 + (b\,\omega)^2$ and $\mathcal{Z} = \{\omega \in \mathbb{R} \mid D'(\omega) = 0\}$. Hence, computing the extrema of D, we get that $D'(\omega) = 2\,\omega\,(2\,m^2\,\omega^2 - 2\,m\,k + b^2)$, which implies that $D'(\omega) = 0$ if and only if $\omega_0 = 0$ or $\omega_\pm = \pm\,\dfrac{1}{m}\sqrt{\dfrac{2\,k\,m - b^2}{2}}$ if $0 < b \leq \sqrt{2\,k\,m}$, and thus:

$$\sqrt{D(0)} = k, \quad \sqrt{D(\omega_\pm)} = \frac{b\,\sqrt{4\,k\,m - b^2}}{2\,m} = \frac{b\,\delta}{2\,m}.$$

If $0 < b < \sqrt{2\,k\,m}$, we have the following variations of the function D

ω	$-\infty$		ω_-		0		ω_+		$+\infty$
$D'(\omega)$		$-$	0	$+$	0	$-$	0	$+$	
$\sqrt{D(\omega)}$	$+\infty$	$\searrow$	$\dfrac{b\,\delta}{2\,m}$	$\nearrow$	k	$\searrow$	$\dfrac{b\,\delta}{2\,m}$	$\nearrow$	$+\infty$

which yields:

$$\| K \|_{\mathcal{L}(L^2,L^2)} = \| G \|_\infty = \frac{2\,m}{b\,\sqrt{4\,k\,m - b^2}}. \tag{1.15}$$

If $b = \sqrt{2\,k\,m}$, then $D'(\omega) = 4\,m^2\,\omega^3$, which yields $\omega_\pm = \omega_0 = 0$ and:

$$\| K \|_{\mathcal{L}(L^2,L^2)} = \| G \|_\infty = \frac{1}{k}.$$

If $\sqrt{2\,k\,m} < b$, then $\omega_\pm$ are complex numbers, which yields:

$$\| K \|_{\mathcal{L}(L^2,L^2)} = \| G \|_\infty = \frac{1}{k}.$$

Using closed-form expression (1.15) for $\| G \|_\infty$ with respect to the system parameters

m, k and $0 < b < \sqrt{2\,k\,m}$, we can notice that

$$\| G \|_\infty = \sqrt{\frac{m}{k}}\, b^{-1} + \frac{1}{8\,k\,\sqrt{m\,k}}\, b + \mathcal{O}(b^3),$$

which shows the behaviour of the operator gain $\| K \|_{\mathcal{L}(L^2,L^2)}$ with respect to a small damping coefficient b. This result is coherent with the fact for $b = 0$, the transfer function $G_0 = 1/(m\,s^2 + k)$ has then two poles $\pm\sqrt{\frac{k}{m}}\, i$ on the imaginary axis, showing that the system (1.4) is then unstable, $\| G_0 \|_\infty = +\infty$ and $G_0 \notin RL_\infty$. Finally, we can check that this gain becomes singular on the algebraic set $\Delta = b^2 - 4\,m\,k = 0$ in the parameter space $\{m, k, b\}$.

A rich interplay between mathematical system theory, operator theory and complex analysis has provided a more tractable characterization of the maximum energy gain of a stable SISO linear system (1.1) in terms of the maximum modulus on the imaginary axis of the proper and stable transfer function $G(s) = C\,(s\,I_n - A)^{-1}\,B + D$ of (1.1). This result advocates for the study of the computation of the ∞-norm of a rational functions with no poles on $i\,\mathbb{P}^1(\mathbb{R})$.

Example 12. Let us consider the following linear time-invariant system

$$\begin{cases} \dot{x}(t) = -x(t) - u(t), \\ y(t) = x(t) + u(t), \end{cases} \tag{1.16}$$

i.e., $A = -1$, $B = -1$, $C = 1$ and $D = 1$. Clearly, A is stable, and thus, so is (1.16). Let $X = \mathscr{L}(x)$ (resp., $U = \mathscr{L}(u)$, $Y = \mathscr{L}(y)$) be the Laplace transform of x (resp., u, y). Then, the transfer function of (1.16) is defined by $Y(s) = X(s) + U(s)$, where $X(s) = -U(s)/(s+1)$, i.e., $G = s/(s+1) \in RH_\infty$. We have $\| G \|_\infty = \sup_{\omega\in\mathbb{R}} |\omega|/\sqrt{1+\omega^2} = 1$. The convolution kernel K of (1.16) is defined by $K(t) = -e^{-t}\,H(t) + \delta$, which satisfies $\| K \|_{\mathcal{A}} = \| e^{-t}\,H(t) \|_1 + 1 = 2$. Hence, this example shows that the inequality (1.13) can be strict, i.e.:

$$\| K \|_{\mathcal{L}(L^2,L^2)} = \| G \|_\infty < \| K \|_{\mathcal{A}}\,.$$

To extend the above results to MIMO systems, we generalize the different functional spaces to consider matrices. We first state a few standard notations.

Definition 1.7. Let $M \in \mathbb{C}^{m\times n}$ be a $m \times n$ matrix with complex entries.

1. The *complex conjugate transpose/Hermitian transpose* of M is the matrix $M^\star \in \mathbb{C}^{n\times m}$

obtained by transposing of the complex conjugate $\overline{M}$, i.e.:

$$M^\star = \overline{M}^T.$$

2. The largest singular value of M, denoted by $\overline{\sigma}(M)$, is the square root of the *largest eigenvalue* of the positive semi-definite matrix $M^\star M \in \mathbb{C}^{n\times n}$.

If $x \in \mathbb{C}^n$ and $\| x \|_2 = \sqrt{x^\star x} = \sqrt{\sum_{i=1}^n |x_i|^2}$, then we state again that:

$$\bar{\sigma}(M) = \sup_{0\neq x\in\mathbb{C}^m} \frac{\| M x \|_2}{\| x \|_2}. \tag{1.17}$$

We extend the Banach spaces L^2, L^∞ and H^∞ for matrix-valued functions.

Definition 1.8. 1. Let $L^2(\mathbb{R}_+)$ be the Hilbert space of Lebesgue measurable matrix-valued functions defined on $\mathbb{R}_+$ bounded for the 2-norm, namely,

$$\| F \|_2 := \sqrt{\int_0^{+\infty} \mathrm{Tr}(F^\star(t)\, F(t))\, dt},$$

where Tr denotes the standard *trace* of the corresponding matrix.

2. Let $L^\infty(i\,\mathbb{R})$ be the Banach space of Lebesgue measurable matrix-valued functions defined on $\mathbb{R}$ essentially bounded, namely:

$$\| F \|_\infty = \text{ess.sup}_{\omega\in\mathbb{R}}\, \overline{\sigma}(F(i\,\omega)).$$

3. Let $H^\infty(\mathbb{C}_+)$ be the Hardy algebra of holomorphic functions defined on $\mathbb{C}_+$ bounded for the ∞-norm:

$$\| F \|_\infty = \sup_{s\in\mathbb{C}_+} \overline{\sigma}(F(s)).$$

Now, if we consider a stable MIMO linear system (1.1) defining a transfer matrix $G \in RH_\infty^{p\times m}$, then $G \in H^\infty(\mathbb{C}_+)$, where $H^\infty(\mathbb{C}_+)$ is defined in Definition 1.8, and $\| G \|_\infty = \sup_{s\in\mathbb{C}_+} \overline{\sigma}(G(s))$. Again, the restriction $G_{|i\,\mathbb{R}}$ of G to the imaginary axis $i\,\mathbb{R}$ belongs to $L^\infty(i\,\mathbb{R}_+)$ defined in Definition 1.8, and $\| G_{|i\,\mathbb{R}} \|_\infty = \text{ess.sup}_{\omega\in\mathbb{R}}\, \overline{\sigma}(G_{|i\,\mathbb{R}}(i\,\omega))$. Similarly, a standard result shows that:

$$\| G \|_\infty = \| G_{|i\,\mathbb{R}} \|_\infty .$$

Considering the input-output operator (1.8) or, equivalently, the convolution operator (1.10), where the kernel $K = \mathscr{L}^{-1}(G) \in \mathcal{A}^{p\times m}$ is defined by (1.9), then a standard result

on H^∞ shows that

$$\| K \|_{\mathcal{L}(L^2,L^2)} := \sup_{0 \neq u \in L^2(\mathbb{R}_+)} \frac{\| y \|_2}{\| u \|_2} = \| G \|_\infty = \| G_{|i\,\mathbb{R}} \|_\infty, \tag{1.18}$$

where, by Definition 1.8, we have:

$$\| u \|_2 = \sqrt{\int_0^{+\infty} \sum_{i=1}^{m} |u_i(t)|^2 \, dt}, \quad \| y \|_2 = \sqrt{\int_0^{+\infty} \sum_{i=1}^{p} |y_i(t)|^2 \, dt}.$$

See, e.g., Theorem 4.4 of [107]. Hence, the computation of the maximum energy gain of a stable MIMO linear system (1.1) is equivalent to computing the L^∞-norm of its proper and stable transfer matrix G, i.e., of $G \in RH_\infty^{p \times m}$.

1.4 A real algebraic geometric reformulation

Using computer algebra methods (*symbolic-numeric methods*), in this dissertation, we study the problem of the (certified) computation of the norm $\| G_{|i\,\mathbb{R}} \|_\infty$ for $G \in RH_\infty^{p \times m}$, or, more generally, the computation of $\| F \|_\infty$ for $F \in RL_\infty^{p \times m}$. This last problem can be reduced to the study of the extremal real zeros of a certain polynomial system. To see that, we use the following characterization of $\| F \|_\infty$.

Proposition 1.1 ([107, 60])**.** Let $F \in RL_\infty^{p \times m}$, $\gamma > 0$ and:

$$\Phi_\gamma(i\,\omega) = \gamma^2 \, I_m - F^T(-i\,\omega) \, F(i\,\omega).$$

Then, $\gamma > \| F \|_\infty$ if and only if $\gamma > \overline{\sigma}\,(F(i\,\infty))$ and:

$$\forall\, \omega \in \mathbb{R}, \quad \det(\Phi_\gamma(i\,\omega)) \neq 0.$$

The sketch of the proof of Proposition 1.1 is the following: using (1.17), $\| F \|_\infty < \gamma$ if and only if

$$\begin{aligned}
&\sup_{\omega \in \mathbb{R}} \sup_{x \in \mathbb{C}^m} \frac{\| F(i\,\omega)\, x \|_2}{\| x \|_2} < \gamma \\
\Leftrightarrow\ & \forall\, \omega \in \mathbb{R},\ \forall\, x \in \mathbb{C}^m,\ \| F(i\,\omega)\, x \|_2 < \gamma^2 \ \| x \|_2 \\
\Leftrightarrow\ & \forall\, \omega \in \mathbb{R},\ \forall\, x \in \mathbb{C}^m,\ x^\star \, F^T(-i\,\omega)\, F(i\,\omega)\, x < \gamma^2 \, x^\star \, x \\
\Leftrightarrow\ & \forall\, \omega \in \mathbb{R},\ \forall\, x \in \mathbb{C}^m,\ x^\star \left(\gamma^2 \, I_m - F^T(-i\,\omega)\, F(i\,\omega)\right) x > 0,
\end{aligned}$$

i.e., if and only if $\Phi_\gamma(i\,\omega) = \gamma^2\, I_m - F^T(-i\,\omega)\, F(i\,\omega) > 0$ for all $\omega \in \mathbb{R}$, namely, the square hermitian matrix $\Phi_\gamma(i\,\omega)$ is positive definite. Using the continuity of the function $\omega \in \mathbb{P}^1(\mathbb{R}) \longmapsto F(i\,\omega)$, and thus, the continuity of the function $\omega \in \mathbb{P}^1(\mathbb{R}) \longmapsto \Phi_\gamma(i\,\omega)$, we obtain that $\Phi_\gamma(i\,\omega) > 0$ for all $\omega \in \mathbb{P}^1(\mathbb{R})$ if and only if $\Phi_\gamma(i\,\infty) > 0$ and $\Phi_\gamma(i\,\omega)$ is non-singular for all $\omega \in \mathbb{R}$, i.e., if and only if $\overline{\sigma}\,(F(i\,\infty)) < \gamma$ and $\det(\Phi_\gamma(i\,\omega)) \neq 0$ for all $\omega \in \mathbb{R}$, which proves the result.

Since F is a proper rational matrix, $F(i\,\infty)$ is a constant matrix and $\overline{\sigma}\,(F(i\,\infty))$ can be computed by standard linear algebra methods.

Example 13. If $F(s) = C\,(s\,I_n - A)^{-1}\,B + D$, then $F(i\,\infty) = D$. Hence, the condition $\Phi_\gamma(i\,\infty) > 0$ amounts to saying that $\gamma^2\, I_m - D^T\, D > 0$, i.e., $\overline{\sigma}(D) < \gamma$.

Hence, to compute the maximal singular value of $F(i\,\omega)$, we have to compute the maximal real value γ satisfying that a real value ω exists such that $\det(\Phi_\gamma(i\,\omega))$ vanishes. Since $\det(\Phi_\gamma(i\,\omega))$ is a real rational function of ω and γ (in fact a real rational function of ω^2 and γ^2 by the parity of Φ_γ), we can write $\det(\Phi_\gamma(i\,\omega)) = n(\omega,\gamma)/d(\omega)$, where $n(\omega,\gamma) \in \mathbb{R}[\gamma,\omega]$ and $d(\omega) \in \mathbb{R}[\omega]$ are coprime. Since F has no poles on the imaginary axis, $d(\omega)$ does not vanish on $\mathbb{R}$. Hence, to compute the L^∞-norm of F, it suffices to compute the maximal real value γ such that there exists at least one real value ω for which $n(\omega,\gamma)$ vanishes. *Hence, we are led to studying the γ-extremal points (and thus, the* critical points*) of the following real plane algebraic curve:*

$$\mathcal{C} := \{(\omega,\,\gamma) \in \mathbb{R}^2 \mid n(\omega,\gamma) = 0\}. \tag{1.19}$$

This problem belongs to the realm of *real algebraic geometry.*

Let us now state important properties on $\mathcal{C}$. Using $F(i\,\omega) \in RL_\infty^{p\times m}$ and the definition of RL_∞, we can check that $F^T(-i\,\omega) \in RL_\infty^{m\times p}$. In other words, RL_∞ is a $\mathbb{R}$-sub-algebra of the *von Neumann algebra* $L^\infty(i\,\mathbb{R})$ which is equipped with the *involution* $f \longmapsto f^\star$ defined by $f^\star(i\,\omega) = f(-i\,\omega)$ for all $f \in RL_\infty$. Therefore, we get that $\Phi_\gamma \in RL_\infty^{m\times m}$. Now, a standard result of module theory on the determinant of matrices with entries in a commutative ring asserts that $\det \Phi_\gamma = \gamma^{2m} + \sum_{k=0}^{m-1} a_{2k}\,\gamma^{2\,k}$, where $a_{2k} \in RL_\infty$ for $k = 0,\ldots,m-1$. The coefficients a_{2k} are real proper rational functions without poles on the imaginary axis $i\,\mathbb{R}$. Writing $a_{2k} = n_{2k}/d_{2k}$, where $d_{2k},\, n_{2k} \in \mathbb{R}[\omega]$ are coprime, then d_{2k} have no real roots and $\deg d_{2k} \leq \deg n_{2k}$ for $k = 0,\ldots,m-1$. Hence, we have:

$$\det \Phi_\gamma = \frac{\prod_{k=0}^{m-1} d_{2k}(\omega)\,\gamma^{2m} + \prod_{k=0}^{m-2} d_{2k}(\omega)\, n_{2(m-1)}(\omega)\,\gamma^{2(m-1)} + \ldots + \prod_{k=1}^{m-1} d_{2k}(\omega)\, n_0(\omega)}{\prod_{k=0}^{m-1} d_{2k}(\omega)}.$$

Let us note:

$$A_{2m} = \prod_{k=0}^{m-1} d_{2k}(\omega),\ A_{2(m-1)} = \prod_{k=0}^{m-2} d_{2k}(\omega)\, n_{2(m-1)}(\omega), \ldots, A_0 = \prod_{k=1}^{m-1} d_{2k}(\omega)\, n_0(\omega).$$

Since the d_{2k}'s have no real roots, then $\det \Phi_\gamma = 0$ is equivalent to the bivariate polynomial equation $\sum_{k=0}^{m} A_{2k}(\omega)\, \gamma^{2k} = 0$. Let $g \in \mathbb{R}[\omega]$ be the greatest common divisor of the A_{2k}'s and set $\bar{A}_{2k} := A_{2k}/g \in \mathbb{R}[\omega]$ for $k = 0, \ldots, m$. Then, we get:

$$n(\omega, \gamma) = \sum_{k=0}^{m} \bar{A}_{2k}(\omega)\, \gamma^{2k}, \quad d(\omega) = \bar{A}_{2m}(\omega).$$

Since the $\bar{A}_{2k}$'s has no common factor, the above polynomial n cannot be divided by a pure polynomial of ω. Hence, the real plane algebraic curve $\mathcal{C}$ does not contain vertical lines $\omega = \omega_\star \in \mathbb{R}$. Moreover, the leading coefficient of $n(\omega, \gamma)$ seen as polynomial in γ is $\bar{A}_{2m}(\omega)$, whose roots are among those of $A_{2m}(\omega)$. Since A_{2m} has no real roots so has $\bar{A}_{2m}$. Hence, the real plane algebraic curve $\mathcal{C}$ has no vertical asymptotes.

Using the fact that $\deg d_{2k} \leq \deg n_{2k}$ for $k = 0, \ldots, m$, $\deg A_{2k} \leq \deg A_{2m}$ for $k = 0, \ldots, m-1$, and thus, $\deg \bar{A}_{2k} \leq \deg \bar{A}_{2m}$ for $k = 0, \ldots, m-1$. Let J be the set formed by the indices j's such that $\deg \bar{A}_{2j}$ is equal to $\deg \bar{A}_{2m}$, that is to say, $J = \{j \in [\![0, \ldots, m]\!] \mid \deg \bar{A}_{2j} = \deg \bar{A}_{2m}\}$. Then, the leading coefficient of the polynomial $n(\omega, \gamma)$ seen as a polynomial in ω is $\sum_{j \in J} C_{2j}\, \gamma^{2j}$, where C_{2j} is the leading coefficient of $\bar{A}_{2j}$. Hence, $n(\omega, \gamma) = \left(\sum_{j \in J} C_{2j}\, \gamma^{2j}\right) \omega^{2l} + r(\gamma, \omega)$, where $2\,l = \deg \bar{A}_{2m}$ and the terms in $r(\gamma, \omega)$ have degrees in ω strictly less than $2\,l$. Thus, $\mathcal{C}$ has horizontal asymptotes for the real solutions of the polynomial:

$$\sum_{j \in J} C_{2j}\, \gamma^{2j} = 0. \tag{1.20}$$

In particular, we note that $\mathcal{C}$ is bounded in the direction of γ, i.e., no real branch in $\gamma(\omega)$ of $\mathcal{C}$ goes to infinity when ω tends to infinity. This shows again that $\| F \|_\infty$ is bounded, i.e., finite. Moreover, the real plane algebraic curve $\mathcal{C}$ is then unbounded in the direction of ω for the real γ satisfying (1.20) since the degree in ω of the polynomial $n(\omega, \gamma)$ then drops. Since by construction, ω can tend to ∞, (1.20) has then at least one real root. Using $\det \Phi_\gamma(\omega) = 0$ for all $\omega \in \mathbb{R}$, we get $\lim_{\omega \to +\infty} \det \Phi_\gamma(\omega) = 0$, i.e., $\det\big(\gamma^2\, I_m - F^T(i\,\infty)\, F(i\,\infty)\big) = 0$, which shows that the horizontal asymptotes in the direction of γ are the singular values of $F(i\,\infty)$. For instance, if $F \in RL_\infty$ is strictly proper, then $F(i\,\infty) = 0$ and $\gamma = 0$ is the only horizontal asymptote of the real plane algebraic curve $\mathcal{C}$.

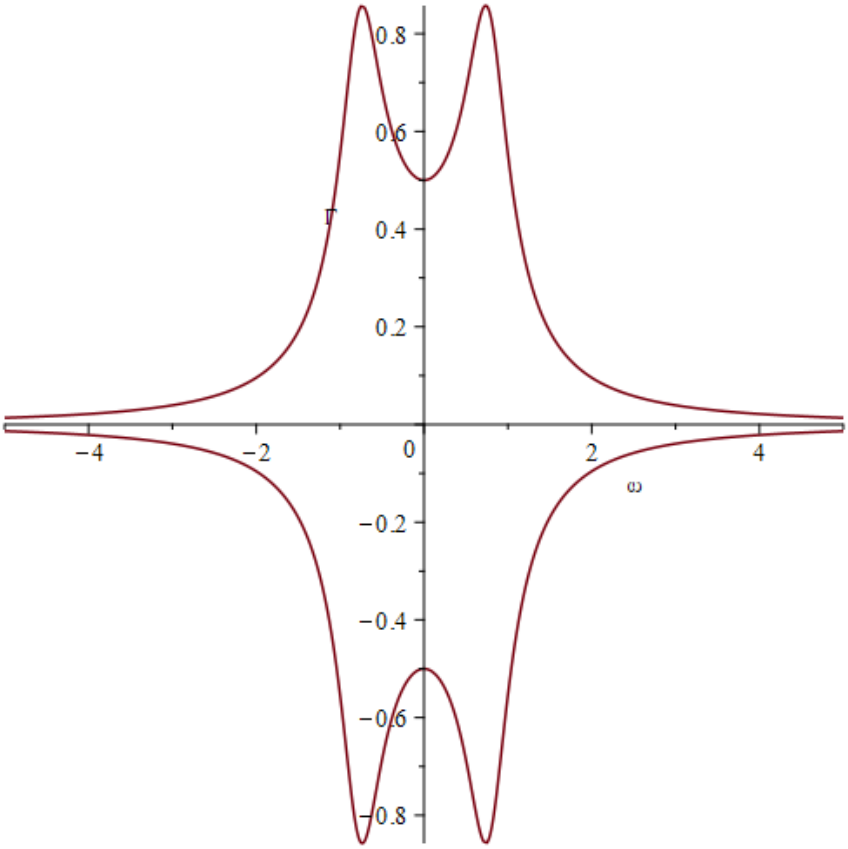

Figure 1.2: Plot of $\mathcal{C}$ for $m = 3$, $k = 2$ and $b = 3/2$, where ω/γ is in the horizontal/vertical axis.

Example 14. Let us consider again Examples 1 and 2. We have:

$$\begin{aligned}\Phi_\gamma(i\,\omega) &= \gamma^2 - \frac{1}{(m\,(-i\,\omega)^2 + b\,(-i\,\omega) + k)}\,\frac{1}{(m\,(i\,\omega)^2 + b\,(i\,\omega) + k)} \\ &= \frac{((k - m\,\omega^2)^2 + (b\,\omega)^2)\,\gamma^2 - 1}{(k - m\,\omega^2)^2 + (b\,\omega)^2}.\end{aligned}$$

Hence, we have $A_2 = (k - m\,\omega^2)^2 + (b\,\omega)^2$, $A_0 = -1$, $g = 1$, $\bar{A}_2 = A_2$, $\bar{A}_0 = A_0$, which shows that $n(\omega, \gamma) = ((k - m\,\omega^2)^2 + (b\,\omega)^2)\,\gamma^2 - 1 = b^2\,\gamma^2\,\omega^4 + \ldots$. Since $G = 1/(m\,s^2 + b\,s + k)$ has no poles on the imaginary axis, $\deg_\gamma n(\omega, \gamma) = 2$ for all $\omega \in \mathbb{R}$ and $\deg_\omega n(\omega, \gamma) = 4$ if $\gamma \neq 0$ and $b \neq 0$, and $\deg_\omega n(\omega, \gamma) = 0$ if $\gamma = 0$. Hence, for all numeric values $b \neq 0$, $m \neq 0$ and $k \neq 0$, the real algebraic curve $\mathcal{C} = \{(\omega,\, \gamma) \in \mathbb{R}^2 \mid ((k - m\,\omega^2)^2 + (b\,\omega)^2)\,\gamma^2 - 1 = 0\}$ is bounded in the direction of γ and unbounded in the direction of ω with a horizontal asymptote at $G(i\,\infty) = 0$. In Figure 1.2, we can visualize the curve $\mathcal{C}$ for the particular numerical values $m = 3$, $k = 2$ and $b = 3/2$.

Example 15. Let us consider the following transfer matrix:

$$F = \begin{pmatrix} \dfrac{10\,s}{s+1} & 1 \\ 0 & \dfrac{5\,s}{s+1} \end{pmatrix} \in RH_\infty^{2\times 2}.$$

We can easily check that:

$$n(\omega,\gamma) = (\gamma^4 - 126\,\gamma^2 + 2500)\,\omega^4 + (2\,\gamma^4 - 127\,\gamma^2)\,\omega^2 + \gamma^4 - \gamma^2.$$

We have

$$F(i\,\infty) = \begin{pmatrix} 10 & 1 \\ 0 & 5 \end{pmatrix},$$

whose singular values are the positive real γ satisfying $\gamma^4 - 126\,\gamma^2 + 2500 = 0$, namely, $\left(\sqrt{226} + \sqrt{26}\right)/2$ and $\left(\sqrt{226} - \sqrt{26}\right)/2$.

A reformulation of Proposition 1.1 for $F \in RL_\infty$ is the following result.

Corollary 1.1. Let $F = a/b \in RL_\infty$, where $a,\, b \in \mathbb{R}[i\,\omega]$ are coprime polynomials, i.e., $\gcd(a,\, b) = 1$, $A_2 = |b(i\,\omega)|^2$, $A_0 = |a(i\,\omega)|^2$, $g = \gcd(A_2, A_0)$, $\bar{A}_2 = A_2/g$, $\bar{A}_0 = A_0/g$ and $n(\omega,\gamma) = \bar{A}_2(\omega)\,\gamma^2 - \bar{A}_0(\omega)$. If $\mathrm{Lc}_\omega(n)$ denotes the leading coefficient of n seen as a polynomial in ω,

$$V_{\mathbb{R}}\left(\left\langle n, \frac{\partial n}{\partial \omega}\right\rangle\right) := \left\{(\omega,\, \gamma) \in \mathbb{R}^2 \mid n(\omega,\gamma) = 0,\ \frac{\partial n(\omega,\gamma)}{\partial \omega} = 0\right\},$$

and $\pi_\gamma : \mathbb{R}^2 \longrightarrow \mathbb{R}$ the projection onto the γ-axis, i.e., $\pi_\gamma(\omega,\, \gamma) = \gamma$, then:

$$\parallel F \parallel_\infty = \max\left\{\pi_\gamma\left(V_{\mathbb{R}}\left(\left\langle n, \frac{\partial n}{\partial \omega}\right\rangle\right)\right) \cup V_{\mathbb{R}}\left(\langle \mathrm{Lc}_\omega(n)\rangle\right)\right\}. \tag{1.21}$$

As explained above, the computation of $\parallel F \parallel_\infty$ is connected to the study of the *critical points* of the real algebraic curve $\mathcal{C} = \{(\omega,\, \gamma) \in \mathbb{R}^2 \mid n(\omega,\gamma) = 0\}$ (namely, the common zeros of the polynomial system formed by $n(\omega,\gamma) = 0$ and $\partial n(\omega,\gamma)/\partial\omega = 0$), where the bivariate polynomial n satisfies the properties:

1. $n(\omega,\gamma) = \bar{A}_{2m}(\omega)\,\gamma^{2\,m} + \bar{A}_{2(m-1)}(\omega)\,\gamma^{2(m-1)} + \ldots + \bar{A}_0(\omega)$, where the greatest common divisor of the $\bar{A}_{2k}$'s is 1, $\deg(\bar{A}_i) \leq \deg(\bar{A}_{2m})$, $i = 0, \ldots, 2\,(m-1)$ and $\bar{A}_{2m}$ has no real roots ω. The real plane algebraic curve $\mathcal{C}$ has no vertical lines nor vertical asymptotes and is bounded in the direction γ.

2. $n(\omega,\gamma) = B_{2l}(\gamma)\,\omega^{2\,l} + B_{2(l-1)}(\gamma)\,\omega^{2(l-1)} + \ldots + B_0(\gamma)$, where the roots of B_{2l} are the singular values of the real matrix $F(i\,\infty)$ and their opposite. The real plane algebraic curve $\mathcal{C}$ is unbounded in the direction ω.

Remark 1.3. Note that l can be reduced to 0 in the above Point 2. For instance, if we consider the *all-pass system* defined by $F(s) = (1 - a\,s)/(1 + a\,s)$, where $a \in \mathbb{R}$, then we have $\Phi_\gamma(i\,\omega) = \gamma^2 - F(-i\,\omega)\,F(i\,\omega) = \gamma^2 - 1$, which yields $l = 0$.

Remark 1.4. Note that the bivariate polynomial n does not need to be *square-free*, namely, divisors of n can be a square of a non-constant polynomial. For instance, if we consider the following transfer matrix

$$F = \begin{pmatrix} \dfrac{1}{s+1} & 0 \\ 0 & \dfrac{1}{s+1} \end{pmatrix} \in RH_\infty^{2\times 2},$$

then we can easily show that $n(\omega,\gamma) = (-(\omega^2+1)\,\gamma^2 + 1)^2$. The study of the real plane algebraic curve $\mathcal{C}$ can then be done by means of the *square-free part* $\bar{n}$ of n, namely, $\bar{n} = -(\omega^2+1)\,\gamma^2 + 1$.

Remark 1.4 shows that it is advantageous to study the critical points of the real plane algebraic curve $\mathcal{C}$ by considering the square-free part $\bar{n}$ of n.

To finish this section, we give an equivalent formulation of the above characterization of the L^∞-norm of LTI systems. This equivalent formulation is intensively used in control theory and for the numerical computation of L^∞-norm as we shall explain in Section 1.5.

Let $G \in \mathbb{R}(s)^{p\times m}$ be such that $G_{|i\,\mathbb{R}} \in RL_\infty^{p\times m}$. Moreover, let us consider $\Phi_\gamma(s) = \gamma^2\,I_m - G^T(-s)\,G(s)$. Above, we showed that $\parallel G \parallel_\infty < \gamma$ if and only if $\Phi_\gamma(i\,\infty) > 0$ and $\Phi_\gamma(i\,\omega)$ is non-singular for all $\omega \in \mathbb{R}$. Let us now suppose that $G(s) = C\,(s\,I_n - A)^{-1}\,B + D$ and set $R = \gamma^2\,I_m - D^T\,D$, where $\gamma > 0$ is not a singular value of D. Now, by elementary algebraic calculations, it can be proved that

$$\Phi_\gamma(s)^{-1} = C'\,(s\,I_{2n} - H_\gamma)^{-1}\,B' + D',$$

where:

$$H_\gamma = \begin{pmatrix} A + B\,R^{-1}\,D^T\,C & B\,R^{-1}\,B^T \\ -C^T\,(I_p + D\,R^{-1}\,D^T)\,C & -\left(A + B\,R^{-1}\,D^T\,C^T\right)^T \end{pmatrix}, \quad D' = R^{-1},$$

$$B' = \begin{pmatrix} B\,R^{-1} \\ -C^T\,D\,R^{-1} \end{pmatrix}, \quad C' = \begin{pmatrix} R^{-1}\,D^T\,C & R^{-1}\,B^T \end{pmatrix}.$$

For more details, see [22]. Instead of studying the real zeros of $\Phi_\gamma(i\,\omega)$, we can study the real poles of the matrix $\Phi_\gamma(i\,\omega)^{-1} = C'\,(i\,\omega\,I_{2n} - H_\gamma)^{-1}\,B' + D'$. Using the identity (1.7), the poles of $\Phi_\gamma(i\,\omega)^{-1}$ are among the imaginary eigenvalues of H_γ. If no pole-zero cancellations occur in the entries of the matrix $\Phi_\gamma(i\,\omega)^{-1}$, then the poles of $\Phi_\gamma(i\,\omega)^{-1}$ are exactly the imaginary eigenvalues of H_γ. It can easily be proved that no such pole-zero cancellations can happen (i.e., if $i\,\omega$ is an eigenvalue of H_γ, then $i\,\omega$ is not an *uncontrollable mode*, as well as not an *unobservable mode*) [19, 22]. Hence, the poles of $\Phi_\gamma(i\,\omega)^{-1}$ are exactly the eigenvalue $i\,\omega$ of H_γ. In other words, the curve $\mathcal{C}$ defined by (1.19) can simply be defined as the characteristic polynomial of the above matrix H_γ.

We get the following characterization of an upper bound of the L_∞-norm.

Theorem 1.2 ([19, 22, 107])**.** Let $G = C\,(s\,I_n - A)^{-1}\,B + D$ be a transfer matrix, where A has no eigenvalues on the imaginary axis, i.e., such that $G \in RL_\infty^{p\times m}$. Let $\gamma > 0$ and $R = \gamma^2\,I_m - D^T\,D$. Then, the following assertions are equivalent:

1. $\|\,G\,\|_\infty < \gamma$,
2. $\overline{\sigma}(D) < \gamma$ and the matrix H_γ has no imaginary eigenvalues.

Example 16. Consider again Example 1. Then, $D = 0$, $R = \gamma^2$ and:

$$H_\gamma = \begin{pmatrix} A & B\,\gamma^{-2}\,B^T \\ -C^T\,C & -A^T \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -\dfrac{k}{m} & -\dfrac{b}{m} & 0 & \dfrac{1}{m^2\,\gamma^2} \\ -1 & 0 & 0 & \dfrac{k}{m} \\ 0 & 0 & -1 & \dfrac{b}{m} \end{pmatrix}.$$

Thus, the characteristic polynomial of H_γ is given by:

$$p(\lambda,\gamma) = \frac{((m\,\gamma)^2\,\lambda^4 + (2\,k\,m - b^2)\,\gamma^2\,\lambda^2 + (k\,\gamma)^2 - 1)}{(m\,\gamma)^2}.$$

Finally, if we set $\lambda = i\,\omega$ into p and consider the numerator n of the corresponding result, then we obtain $n(\omega,\gamma) = ((k - m\,\omega)^2 + (b\,\omega)^2)\,\gamma^2 - 1$ and we find again the real plane algebraic curve $\mathcal{C}$ defined in Example 14.

Remark 1.5. In the literature, we can also see Theorem 1.2 with the matrix

$$H'_\gamma = \begin{pmatrix} A - B\,R'^{-1}\,D^T\,C & -\gamma\,B\,R'^{-1}\,B^T \\ \gamma\,C^T\,S'^{-1}\,C & -\left(A - B\,R'^{-1}\,D^T\,C^T\right)^T \end{pmatrix} \tag{1.22}$$

instead of H_γ, where $R' = D^T D - \gamma^2 I_m$ and $S' = D D^T - \gamma^2 I_p$. See, e.g., [19, 22]. To explain the equivalence, we first note that $R = -R'$. Then, that all $U \in \mathbb{C}^{q\times r}$ and $V \in \mathbb{C}^{r\times q}$, we have

$$\big(I_q + U\,(I_r - V\,U)^{-1}\,V\big)\;(I_q - U\,V) = I_q - U\,V + U\,(I_r - V\,U)^{-1}\,(I_r - V\,U)\,V = I_q,$$

which proves the standard identity $(I_q - U\,V)^{-1} = I_q + U\,(I_r - V\,U)^{-1}\,V$. Setting $U = D/\gamma$ and $V = D^T/\gamma$, we then get $I_p - D\,R'^{-1}\,D^T = -\gamma^2\,S'^{-1}$. Finally, we can easily check that H'_γ is similar to H_γ for the following invertible transform:

$$T = \begin{pmatrix} I_n & 0 \\ 0 & \gamma\,I_n \end{pmatrix}.$$

The matrix H'_γ is called a *Hamiltonian matrix* since it satisfies the identity

$$J^{-1}\,H'_\gamma\,J = -H'^{\,T}_\gamma,$$

where J is the standard matrix defined by:

$$J = \begin{pmatrix} 0 & I_n \\ -I_n & 0 \end{pmatrix}.$$

According to Theorem 1.2, the search for $\|\,G\,\|_\infty$ can be reduced to the problem of finding the maximal $\gamma > 0$ for which the characteristic polynomial of the matrix H_γ or, equivalently, of the Hamiltonian matrix H'_γ, has no imaginary eigenvalues. Since $\det(i\,\omega\,I_{2n} - H_\gamma) = \det\big(i\,\omega\,I_{2n} - H'_\gamma\big) = 0$ is equivalent to $n(\omega,\gamma) = 0$, we are also led to the study of extremal points, for the projection onto the γ-axis, of the real plane algebraic curve $\mathcal{C}$ defined by (1.19).

1.5 Existing computational methods

Until the end of the eighties, it does not seem that much attention has been paid to the problem of computing the L^∞-norm for LTI systems. For SISO systems, as explained in Section 1.3, this problem corresponds to the peak value of the function $\omega \in \mathbb{R} \longmapsto |G(i\,\omega)|$, i.e., the peak value of the so-called *Bode magnitude plot*. Equivalently, this norm corresponds to the farthest point from the origin (for the distance defined by the modulus) of the complex implicit rational curve $\omega \in \mathbb{R} \longmapsto G(i\,\omega) = \mathrm{Re}(G(i\,\omega)) + i\;\mathrm{Im}(G(i\,\omega)) \in \mathbb{C}$, called the *Nyquist*

plot. The maximum of $|G(i\,\omega)|$ can also be computed by means of studying the real roots of the univariate polynomial $d|G(i\,\omega)|^2/d\omega = 0$ as shown in Example 11. Graphical methods were mainly used to get an approximation of $\| G \|_\infty$.

For MIMO systems, $\| G \|_\infty$ was usually studied by searching the maximum of $\{\overline{\sigma}(G(i\,\omega_k))\}_{k=1,\ldots,N}$, where $\{\omega_k\}_{k=1,\ldots,N}$ is a fine grid for the frequency axis. The graph of $\{(\omega_k, \overline{\sigma}(G(i\,\omega_k)))\}_{k=1,\ldots,N}$ is usually referred to as a *singular-value (SV) plot*. The main drawbacks of this method are the following:

- determining the range and the spacing of the frequencies to be checked (especially, when A has eigenvalues with small real part, such as for lightly damped mechanical structures (see Example 11)),
- the computation of many singular values of matrices (one at each frequency point ω_k),
- no accuracy bound was obtained,
- no certification of the result was given, etc.

At the end of the eighties, alternative numerical methods for the computation of the L^∞-norm of LTI systems were developed in [19, 84] based on Theorem 1.2 and Remark 1.5, i.e., based on the search for imaginary eigenvalues of the Hamiltonian matrix H'_γ defined by (1.22). To achieve that, *bisection algorithms* for determining γ were developed. The original bisection method, developed in [19], consists in finding an upper bound γ_u and a lower bound γ_l for $\| G \|_\infty$ (using, e.g., *Hankel singular values*), then setting $\gamma = (\gamma_u + \gamma_l)/2$, testing if $\| G \|_\infty < \gamma$ by checking the existence of imaginary eigenvalues of H_γ or H'_γ (using, e.g., a direct computation or *Sturm/Routh test*), and setting $\gamma_l = \gamma$ if such imaginary eigenvalues exist, or $\gamma_u = \gamma$ else. The algorithm stops when $(\gamma_u - \gamma_l)/\gamma_l$ is less than a specified level ε and returns:

$$\| G \|_\infty \approx \frac{\gamma_u + \gamma_l}{2}.$$

The error $| \| G \|_\infty - (\gamma_u + \gamma_l)/2|$ is then less than or equal to $\varepsilon \| G \|_\infty$, i.e., the result is guaranteed within a relative accuracy of ε. As stated in [46], numerical problems can occur in the bisection method

1. for transfer matrices G (resp., state-space system (A, B, C, D)) with imaginary poles (resp., imaginary eigenvalues of A) of multiple multiplicity (see, e.g., Remark 1.4),
2. due to a poor scaling or balancing state-space realization (A, B, C, D) of the transfer matrix G,

3. to an *ill-conditioned eigenvalue problem* for the associated Hamiltonian matrix H'_γ,

4. for *all-pass* or *high-pass response systems*, etc.

Different important improvements of the original bisection method have been developed in [18, 22, 46, 9]. In particular, the quadratic convergence of this method was proved in [18]. Variants of this method were implemented in different `Matlab` toolboxes such as the `normhinf` function of *Robust Control Toolbox*, the `hinfnorm` function of *μ-Analysis and Synthesis Toolbox*, and the `norminf` function of the *LMI Control Toolbox*, in `Scilab` (`hinfnorm` command), or in the `Maple` library *DynamicSystems* (`NormHinf` command).

Numerical algorithms based on *Linear Matrix Inequalities* (LMI) were also developed in [57] for transfer matrices defined by *left* or *right coprime factorizations*. In particular, no state-space representation of the transfer matrix is needed. The results obtained in [57] were implemented in the `hinfnorm` command of the `Matlab` toolbox *Polynomial Toolbox*.

The computation of the L^∞-norm of *descriptor LTI systems*, namely

$$\begin{cases} E\,\dot{x}(t) = A\,x(t) + B\,u(t), \\ y(t) = C\,x(t) + D\,u(t), \end{cases}$$

where $E \in \mathbb{R}^{n\times n}$ is a singular matrix, or equivalently to the class of transfer matrices of the form $G(s) = C\,(E\,s - A)^{-1}\,B + D$, has been developed in [11, 10, 48, 95]. In particular, numerical improvements in the computation of L^∞-norm of LTI systems were obtained. The algorithms developed in [10] are implemented in the library `SLICOT` [5, 9] based on the `BLAS` and `LAPACK` libraries (`Fortran 77`).

The L^∞-norm computation of LTI systems depending affinely on a set of free parameters was studied in [100] using an equivalent *semi-infinite convex optimization reformulation* that can be numerically studied by a relaxation approach over a finite set of frequency values and *semi-definite programming methods*.

To our knowledge, the first *symbolic-numeric* study of the computation of L^∞-norm for LTI systems was developed in [60]. The real plane algebraic curve $\mathcal{C}$ defined by (1.19) and its study appear in [60]. A validated numerical computation of the L^∞-norm is proposed based on the study of $\mathcal{C}$ and *Sturm test*. Moreover, a complexity analysis of the proposed algorithm is given. The study of $\mathcal{C}$ was continued in [8]. A purely numerical algorithm, implemented in `Matlab`, was developed based on *Bezoutian matrices*. In [24], the problem of the L^∞-norm computation is explicitly stated as the problem of finding the supremum $\gamma > 0$ such that there exists a real ω satisfying $n(\omega, \gamma) = 0$, i.e., as the problem of finding the γ-extremal point of $\mathcal{C}$. Using *border polynomials* and *triangular decomposition methods*,

the computation of the L^∞-norm is then studied for a polynomial n which depends on free *parameters* (e.g., the system parameters m, k and b in Example 11). The results of [24] were implemented in `Maple` using the `RegularChains` library.

1.6 Robust control theory in a nutshell

Let us briefly explain why the L^∞-norm computation plays a fundamental role in control theory. A mathematical model of a linear system describing a physical phenomenon is incomplete in the sense that it can only be a rough approximation of the original phenomenon. Indeed, some dynamics have been (intentionally or unintentionally) neglected, some system parameters are not well-known or not well-estimated, etc. For more details, see, e.g., [39, 102, 107, 33].

To handle this important issue in the control theory, *robust control theory* has been developed in the eighties based on the ideas that a mathematical model should be close to the real system for a certain topology, the properties of a system should be as robust as possible to small perturbations (*robust margins*), and the design of *stabilizing controllers* should be robust to certain families of perturbations of the system (e.g., additive/multiplicative/inverse additive/inverse multiplicative perturbations). Since the robustness competes with the performance, a compromise between these two opposite objectives has to be chosen wisely while designing a *stabilizing controller*, namely, designing a linear system − called *controller* − which *stabilizes* the new system obtained by adding the controller to the system in a feedback loop (see the stability definitions given in Section 1.2). A stabilizing controller is said to *robustly stabilizes* the system if not only it stabilizes the system but all the systems in a neighborhood of the system for a certain topology. In H_∞*-control theory* − the most popular approach in robust control theory − the norm for measuring the distance is the L^∞-norm. The computation of the maximal radius of the "ball of systems", centered at the system, that are stabilized by the controller is an important issue [39, 107, 33].

To be more precise, let us briefly mathematically state the different problems introduced above. Let $P \in \mathbb{R}(s)^{q\times r}$ be a proper rational transfer matrix defining an approximation of a real system. Then, the controller defined by a proper rational transfer matrix $C \in \mathbb{R}(s)^{r\times q}$ is said to *stabilize* P if:

$$\Pi(P,C) := \begin{pmatrix} (I_q - P\,C)^{-1} & -(I_q - P\,C)^{-1}\,P \\ C\,(I_q - P\,C)^{-1} & -C\,(I_q - P\,C)^{-1}\,P \end{pmatrix} \in RH_\infty^{(q+r)\times(q+r)}. \tag{1.23}$$

To understand the definition of *stabilizability*, we first note that P is not necessarily an element of $RH_\infty^{q\times r}$, which means that the operator $u \longmapsto y = K \star u$, where $K = \mathscr{L}(P)^{-1}$, is not necessarily $L^2 - L^2$-stable (i.e., $u \in L^2(\mathbb{R}_+)^r$ does not necessarily yield $y \in L^2(\mathbb{R}_+)^q$) since $\| K \|_{\mathcal{L}(L^2,L^2)} = \| P \|_\infty$ is not necessarily bounded (see (1.14)). We also note that $P \in \mathbb{R}(s)^{q\times r}$ and $C \in \mathbb{R}(s)^{r\times q}$ yields $\Pi(P,C) \in \mathbb{R}(s)^{(q+r)\times(q+r)}$. As above, if $\Pi(P,C)$ is considered as the transfer matrix of a new system defined by $(u_1 \quad u_2)^T \longmapsto (e_1 \quad y_1)^T = \widetilde{K} \star (u_1 \quad u_2)^T$, where $\widetilde{K} = \mathscr{L}(\Pi(P,C))^{-1}$, then this system is $L^2 - L^2$-stable if and only if $\Pi(P,C) \in RH_\infty^{(q+r)\times(q+r)}$. In particular, $\| \Pi(P,C) \|_\infty < +\infty$ is a necessary condition for the $L^2 - L^2$-stability. Finally, if we consider the closed-loop system, defined in Figure 1.3, obtained by adding the controller C in feedback to the system P, then, at the two interconnections, we have the following equations

$$\left\{ \begin{array}{l} e_1 = u_1 + y_2 = u_1 + P\, e_2, \\ e_2 = u_2 + y_1 = u_2 + C\, e_1, \end{array} \right. \Leftrightarrow \left(\begin{array}{c} u_1 \\ u_2 \end{array} \right) = \left(\begin{array}{cc} I_q & -P \\ -C & I_r \end{array} \right) \left(\begin{array}{c} e_1 \\ e_2 \end{array} \right),$$

which yields

$$\left(\begin{array}{c} e_1 \\ e_2 \end{array} \right) = H(P,C) \left(\begin{array}{c} u_1 \\ u_2 \end{array} \right),$$

where:

$$H(P,C) = \left(\begin{array}{cc} I_q & -P \\ -C & I_r \end{array} \right)^{-1} = \left(\begin{array}{cc} (I_q - P\,C)^{-1} & (I_q - P\,C)^{-1}\,P \\ C\,(I_q - P\,C)^{-1} & I_r + C\,(I_q - P\,C)^{-1}\,P \end{array} \right).$$

Using (1.23) and Figure 1.3, we can then easily check that:

$$\Pi(P,C) \left(\begin{array}{c} u_1 \\ u_2 \end{array} \right) = \left(\begin{array}{c} e_1 \\ C\,e_1 \end{array} \right) = \left(\begin{array}{c} e_1 \\ y_1 \end{array} \right).$$

Hence, if C stabilizes P, then the operator which maps $(u_1 \quad u_2)^T$ to $(e_1 \quad y_1)^T$ is $L^2 - L^2$-stable and $\| \Pi(P,C) \|_\infty$ is its maximum energy gain. More generally, we can check that all the transfer matrices between two signals appearing in Figure 1.3 can be formed by means of the four block matrices defined in $\Pi(P,C)$, which shows that all the operators relating two signals appearing in Figure 1.3 are then $L^2 - L^2$-stable and their maximum energy gains are equal to the L^∞-norm of their corresponding transfer matrices. These results show the importance of the computation of L^∞-norm in control theory and robust control.

As explained above, the transfer matrix P is only a mathematical model of a physical system, and thus, it is a rough approximation. To take into account this uncertainty on the

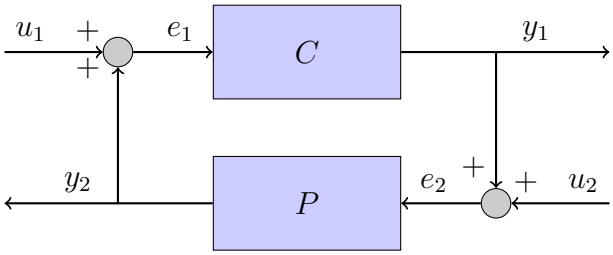

Figure 1.3: Closed-loop system

model, the H_∞-control theory aims at determining a controller C that not only stabilizes P but also all the systems defined by transfer matrices in a neighborhood of P. More precisely, if we consider the system defined by the following transfer matrix

$$P' = (I_q + \Delta_1)^{-1} (P + \Delta_2) = (P + \Delta_3) (I_r + \Delta_4)^{-1}, \tag{1.24}$$

where the matrices Δ_i's are the following matrices

$$\left\{ \begin{array}{l} (\Delta_1 \quad - \Delta_2) = (I_q \quad - P)\, V, \\ \begin{pmatrix} \Delta_3 \\ \Delta_4 \end{pmatrix} = W \begin{pmatrix} P \\ I_r \end{pmatrix}, \end{array} \right.$$

and the matrices V and W are given by

$$V = \begin{pmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{pmatrix}, \quad W = \begin{pmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{pmatrix},$$
$$\left\{ \begin{array}{l} V_{11} \in RH_\infty^{q\times q},\ V_{12} \in RH_\infty^{q\times r},\ V_{21} \in RH_\infty^{r\times q},\ V_{22} \in RH_\infty^{r\times r}, \\ W_{11} \in RH_\infty^{q\times q},\ W_{12} \in RH_\infty^{q\times r},\ W_{21} \in RH_\infty^{r\times q},\ W_{22} \in RH_\infty^{r\times r}, \end{array} \right.$$

then we obtain the following general *model of perturbations* of P

$$P' = (I_q + V_{11} - P\, V_{21})^{-1} (P\, (I_r + V_{22}) - V_{12}) = ((I_q + W_{11})\, P + W_{12})\, (I_r + W_{22} + W_{21}\, P)^{-1}, \tag{1.25}$$

where the matrices Δ_i's are considered to be arbitrary [76]. In particular, we can find again the following standard perturbation models at once [39, 102, 107, 33]:

1. If $V_{12} = 0$, $V_{21} = 0$ and $V_{22} = 0$, then $P' = (I_q + V_{11})^{-1} P$ is an *inverse additive perturbation* of P.

2. If $V_{11} = 0$, $V_{21} = 0$ and $V_{22} = 0$, then $P' = P - V_{12}$ is an *additive perturbation* of P.

3. If $V_{11} = 0$, $V_{12} = 0$ and $V_{22} = 0$, then $P' = (I_q - P\,V_{21})^{-1}\,P$ is an *inverse multiplicative perturbation* of P.

4. If $V_{11} = 0$, $V_{12} = 0$ and $V_{21} = 0$, then $P' = P\,(I_r + V_{22})$ is a *multiplicative perturbation* of P.

Coprime perturbations [107, 102, 33] can also be covered by the class of perturbations defined by (1.24) or (1.25). For more details, see [76].

The robust stabilization problem then aims at determining a controller $C \in \mathbb{R}(s)^{r\times q}$ that stabilizes the larger class of systems P'. Let us suppose that C stabilizes P. If we note

$$\Pi'(P,C) = \begin{pmatrix} -P\,(I_r - C\,P)^{-1}\,C & P\,(I_r - C\,P)^{-1} \\ (I_r - C\,P)^{-1}\,C & (I_r - C\,P)^{-1} \end{pmatrix} \in RH_\infty^{(q+r)\times(q+r)},$$

then we can first easily check again that $\Pi(P,C)$ and $\Pi'(P,C)$ are *two complementary idempotents* of the non-commutative ring $RH_\infty^{(q+r)\times(q+r)}$, namely:

$$\Pi(P,C)^2 = \Pi(P,C), \quad \Pi'(P,C)^2 = \Pi'(P,C), \quad \Pi(P,C) + \Pi'(P,C) = I_{q+r}.$$

Moreover, it can be proved that C stabilizes P' defined by (1.24) or, equivalently, by (1.25), for all $V \in RH_\infty^{(q+r)\times(q+r)}$ and $W \in RH_\infty^{(q+r)\times(q+r)}$ satisfying:

$$\parallel V \parallel_\infty < \parallel \Pi(P,C) \parallel_\infty^{-1}, \quad \parallel W \parallel_\infty < \parallel \Pi'(P,C) \parallel_\infty^{-1}.$$

For more details, see [76, 102] and the references therein. Hence, the computation of the L^∞-norm of the two idempotents $\Pi(P,C)$ and $\Pi'(P,C)$ (it can be shown that $\parallel \Pi(P,C) \parallel_\infty = \parallel \Pi'(P,C) \parallel_\infty$ [102]) yields the maximum radius of the "ball" of linear systems

$$\begin{aligned} B_C(P) := \{\, & P' = (I_q + \Delta_1)^{-1}\,(P + \Delta_2) = (P + \Delta_3)\,(I_r + \Delta_4)^{-1} \mid \\ & (\Delta_1 \quad -\Delta_2) = (I_q \quad -P)\,V,\ (\Delta_3^T \quad \Delta_4^T)^T = W\,(P^T \quad I_r^T)^T \\ & \parallel V \parallel_\infty < \parallel \Pi(P,C) \parallel_\infty^{-1}, \quad \parallel W \parallel_\infty < \parallel \Pi(P,C) \parallel_\infty^{-1} \} \end{aligned}$$

that are stabilized by the stabilizing controller C. If we denote by $\mathrm{Stab}(P)$ the set of all the stabilizing controllers of P, then the robust stabilization problem aims at determining the stabilizing controllers which minimize the L^∞-norm $\parallel \Pi(P,C) \parallel_\infty$, i.e., it aims at determining:

$$\mathrm{argmin}_{C\in\mathrm{Stab}(P)} \parallel \Pi(P,C) \parallel_\infty . \tag{1.26}$$

These controllers thus maximize the above radius of robustness. The set $\mathrm{Stab}(P)$ can be

explicitly parametrized by means of the so-called *Youla-Kučera parametrization*, which is affine in an arbitrary matrix parameter $Q \in RH_\infty^{r\times q}$. For instance, see [39, 107]. Hence, the nonlinear optimization problem (1.26) can be transformed into an affine optimization problem in Q, and thus, into a *convex optimization problem*. But, since $Q \in RH_\infty^{r\times q}$ and $RH_\infty^{r\times q}$ is an infinite-dimensional $\mathbb{R}$-vector space, the optimization problem (1.26) is infinite-dimensional, which makes nontrivial the search for its solution. Nevertheless, using the so-called *Nehari theorem* – which characterizes the distance of $f \in L^\infty(i\,\mathbb{R})$ from $H^\infty(\mathbb{C}_+)$ in $L^\infty(i\,\mathbb{R})$, namely, $\mathrm{dist}(f, H^\infty) := \inf_{g\in H^\infty(\mathbb{C}_+)} \| f - g \|_\infty$ – the *maximum stability margin*

$$b_{\mathrm{opt}}(P) = \left(\inf_{C\in \mathrm{Stab}(P)} \| \Pi(P, C) \|_\infty \right)^{-1}$$

can be characterized as the operator norm of a certain *Hankel operator*, i.e., as a *Hankel norm* (see, e.g., [33, 107]). For finite-dimensional LTI systems, $b_{\mathrm{opt}}(P)$ can nicely be characterized as follows: Let (A, B, C, D) be a controllable and observable realization of the transfer matrix P (see Theorem 1.1) and let X (resp., Y) be the unique positive definite solution of the *Riccati equation* $X\,A + A^T\,X - X\,B\,B^T\,X + C^T\,C = 0$ (resp., $Y\,A^T + A\,Y - Y\,C^T\,C\,Y + B\,B^T = 0$), then we have

$$b_{\mathrm{opt}}(P) = \sqrt{1 + \lambda_{\max}(Y\,X)}, \tag{1.27}$$

where $\lambda_{\max}$ denotes the maximal eigenvalue. The problem of explicitly characterizing the robust controllers achieving this bound is more involved. Hence, in the robust control theory, the following problem is usually preferred [102, 107, 33]: Given $b_{\mathrm{opt}}(P) > b > 1$, find a stabilizing controller C of P which satisfies:

$$\| \Pi(P, C) \|_\infty \leq b. \tag{1.28}$$

This problem receives a tractable solution [51, 107]. Finally, let us state the explicit characterization of a particular solution. See [51] for the general solution. Let X and Y be the unique definite positive solutions of the above Riccati equations, then $C_b(s) = C_b\,(s\,I_n - A_b)^{-1}\,B_b$ stabilizes P and satisfies (1.28), where:

$$\begin{cases} Z_b = (I_n + (X\,Y - b^2\,I_n))^{-T}, \\ A_b = A - B\,B^T\,X + b^2\,Z_b\,Y\,C^T\,C, \\ B_b = -b^2\,Z_b\,Y\,C^T, \\ C_b = B^T\,X. \end{cases} \tag{1.29}$$

Example 17. We consider again Example 1 and the corresponding transfer function $P =$

$c_0/(s^2 + a_1\, s + a_0)$, where $c_0 = 1/m$, $a_1 = b/m$ and $a_0 = k/m$. Using new algebraic methods developed in [78, 77], we can check again that

$$X = \begin{pmatrix} \beta_0\, \beta_1 - a_1\, a_0 & \beta_0 - a_0 \\ \beta_0 - a_0 & \beta_1 - a_1 \end{pmatrix},$$

where β_0 and β_1 satisfy the following system of polynomial equations

$$\begin{cases} \beta_0^2 = a_0^2 + c_0^2, \\ \beta_1^2 = 2\, \beta_0 + a_1^2 - 2\, a_0, \end{cases}$$

satisfies the Riccati equation $X\, A + A^T\, X - X\, B\, B^T\, X + C^T\, C = 0$, where the matrices A, B and C are defined by:

$$A = \begin{pmatrix} 0 & 1 \\ -a_0 & -a_1 \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad C = (c_0 \quad 0).$$

Using a result of [78, 77], X is the unique positive definite solution of the above Riccati equation if and only if:

$$\beta_0 = \sqrt{a_0^2 + c_0^2}, \quad \beta_1 = \sqrt{2\, \sqrt{a_0^2 + c_0^2} + a_1^2 - 2\, a_0}. \tag{1.30}$$

Moreover, we can check that $Y = Q\, X\, Q$, where Q is defined by

$$Q = \frac{1}{c_0} \begin{pmatrix} 0 & 1 \\ 1 & -a_1 \end{pmatrix}$$

namely,

$$Y = \frac{1}{c_0^2} \begin{pmatrix} \beta_1 - a_1 & \beta_0 - a_1\, \beta_1 + a_1^2 - a_0 \\ \beta_0 - a_1\, \beta_1 + a_1^2 - a_0 & \beta_0\, \beta_1 - 2\, a_0\, \beta_0 + a_1^2\, \beta_1 - a_1^3 + a_0\, a_1 \end{pmatrix},$$

satisfies the Riccati equation $Y\, A^T + A\, Y - Y\, C^T\, C\, Y + B\, B^T = 0$. For more details, see [78, 77]. Moreover, Y is the unique positive definite solution of the above Riccati equation if and only if (β_0, β_1) satisfies (1.30) [78, 77]. Then, using (1.27), we obtain $b_{\text{opt}}(P) = \sqrt{1 + \lambda_{\max}(Y\, X)}$, where $\lambda_{\max}(Y\, X)$ is the maximal real eigenvalue of $Y\, X$. To

simplify the notations, if we note

$$Z = Y\,X = \begin{pmatrix} z_{11} & z_{12} \\ z_{21} & z_{22} \end{pmatrix},$$

then the characteristic polynomial of Z is defined by

$$p_Z(\lambda) = \lambda^2 - \mathrm{trace}(Z)\,\lambda + \det(Z) = \lambda^2 - (z_{11} + z_{22})\,\lambda + (z_{11}\,z_{22} - z_{12}\,z_{21}),$$

and the maximal real eigenvalue of Z is defined by:

$$\lambda_{\max}(Y\,X) = \frac{1}{2}\,(\mathrm{trace}(Z) + \sqrt{\mathrm{trace}(Z)^2 - 4\,\det(Z)}).$$

We then obtain the following formula for $b_{\mathrm{opt}}(P)$

$$b_{\mathrm{opt}}(P) = \sqrt{1 + \frac{1}{2}\,(\mathrm{trace}(Z) + \sqrt{\mathrm{trace}(Z)^2 - 4\,\det(Z)})},$$

which, using (1.30), can be made explicit in terms of c_0, a_0 and a_1. It can be shown that $b_{\mathrm{opt}}(P)$ depends only on the two parameters $P(0) = c_0/a_0 = 1/k$ and $\rho = a_1/\sqrt{a_0} = b/\sqrt{k\,m}$. Hence, $b_{\mathrm{opt}}(P)$ can be plotted as a function of k and $b/\sqrt{k\,m}$. For more details, see [79, 77]. Finally, for $1 < b < b_{\mathrm{opt}}(P)$, we can then use (1.29) to obtain an explicit stabilizing controller C of P satisfying (1.28).

For more explicit examples with system parameters, see [78, 79, 80, 77].

Chapter 2

Prerequisite in Computer Algebra

2.1 Notations

We introduce some notations that are used in the rest of this dissertation.

We denote by $\mathbb{D}$ an *integral domain* (namely, a commutative ring with no non-trivial zero-divisors), typically $\mathbb{Z}[x]$, $\mathbb{Z}[x, y]$ or $\mathbb{Z}$, and by $\mathbb{F}_{\mathbb{D}}$ its *fraction field* (namely, $\mathbb{F}_{\mathbb{D}} = \{d_1/d_2 \mid 0 \neq d_2, d_1 \in \mathbb{D}\}$). We also denote by $\Bbbk$ an arbitrary field, typically $\mathbb{Q}$ or $\mathbb{R}$. We recall that any field $\mathbb{K}_1$ containing $\Bbbk$ is an *extension field* of $\Bbbk$. An element x of $\mathbb{K}_1$ is said to be *algebraic* over $\Bbbk$ if there exists a polynomial P with coefficients in $\Bbbk$ such that $P(x) = 0$. An extension field $\mathbb{K}_1$ of $\Bbbk$ is said to be *algebraic* if all of its elements are algebraic over $\Bbbk$. A field $\mathbb{K}$ is said to be *algebraically closed* if it does not have any non-trivial algebraic extension. $\mathbb{K}$ is an *algebraic closure* of $\Bbbk$ if $\mathbb{K}$ is an algebraic extension of $\Bbbk$ which is *algebraically closed*. In the sequel, we denote by $\mathbb{K}$ the algebraic closure of $\Bbbk$ [94, 83]. Finally, let $\mathbb{C}_+ := \{s \in \mathbb{C} \mid \mathrm{Re}(s) > 0\}$ be the *open right-half plane* of $\mathbb{C}$.

Let $P \in \Bbbk[x, y]$ and $v \in \{x, y\}$. Then, $\mathrm{Lc}_v(P)$ is the *leading coefficient* of P with respect to the variable v and $\deg_v(P)$ denotes the *degree* of P in v. Hence, we can write $P = \mathrm{Lc}_v(P)\, v^{\deg_v(P)} + \sum_{i=0}^{\deg_v(P)-1} a_i\, v^i$, where $\mathrm{Lc}_v(P)$ and the a_i's are univariate polynomials in the variable $u \in \{x, y\} \backslash \{v\}$. We also denote by $\deg(P)$ the *total degree* of P, namely, if $P = \sum_{0 \leq i \leq n_1, 0 \leq j \leq n_2} a_{ij}\, x^i\, y^j$, where $a_{ij} \in \Bbbk$ and $a_{ij} \neq 0$ for at least one pair $(i,\, j)$ such that $i + j = n_1 + n_2$, then $\deg(P) = n_1 + n_2$. Moreover, let $\pi_x : \mathbb{R}^2 \longrightarrow \mathbb{R}$ be the projection map from the real plane $\mathbb{R}^2$ onto the x-axis, i.e., $\pi_x(x, y) = x$ for all $(x,\, y) \in \mathbb{R}^2$. For $P,\ Q \in \Bbbk[x, y]$, let $\langle P,\ Q\rangle$ be the ideal of $\Bbbk[x, y]$ generated by P and Q. Let I be an ideal of $\Bbbk[x, y]$, the notation $V_{\mathbb{K}}(I)$ refers to the *affine algebraic set* over $\Bbbk$ associated with I, i.e., $V_{\mathbb{K}}(I) := \{(x,\, y) \in \mathbb{K}^2 \mid \forall\ R \in I :\ R(x, y) = 0\}$. For $\Bbbk = \mathbb{C}$, we simply denote $V_{\mathbb{K}}(I)$ by $V(I)$.

Bitsize The bitsize of an integer n is the number of bits needed to represent it, that is $\lfloor \log_2 n \rfloor + 1$ (the notation $\log_2$ refers to the logarithm in base 2 and $\lfloor m \rfloor$ stands for the greatest integer less than or equal to m). If p is a rational number, then its bitsize is given as the maximum bitsize of its numerator and denominator. The bitsize of a polynomial with integer or rational coefficients is the maximum bitsize of its coefficients. We refer to τ_a as the bitsize of a polynomial, a rational or a integer a. For a polynomial $P \in \mathbb{D}[x]$, we define the size of P, denoted by size(P), to be the pair $(\deg_x(P), \tau_P)$.

Complexity We use the classical $\mathcal{O}$ for denoting asymptotic bounds. Recall that $f(n) = \mathcal{O}(g(n))$ for $n \to +\infty$ if there exist two positive constants N and C such that $|f(n)| \leq C\,|g(n)|$ for all $n > N$. When $\mathbb{D} = \mathbb{Z}$, to obtain a more relevant measure of complexity, we consider the growth of the coefficients in the cost of the operations by considering the *bit complexity* of the algorithm, that is the number of bit operations performed by the algorithm. We denote the bit complexity by $\mathcal{O}_B$. Finally, we denote by $\tilde{\mathcal{O}}$ the complexities where polylogarithmic factors are omitted. More precisely, $f(n) = \tilde{\mathcal{O}}(g(n))$ means that there exists $k \geq 0$ such that $f(n) = \mathcal{O}(g(n)\ \log_2^k(\max(|g(n)|, 2)))$. Note that, unless specified otherwise, the stated complexities are worst-case bit complexities.

We give an example of the bit complexities of some basic operations over polynomials, such as adding, multiplying, and dividing two univariate polynomials $f,\ g \in \mathbb{Z}[x]$. For more details on the corresponding algorithms, along with a proof of complexity, the reader is referred to [103].

Example 18. We consider two univariate polynomials $f,\ g \in \mathbb{Z}[x]$, with degrees bounded by d and coefficients of bitsize bounded by τ. Then, the sum $f + g$ can be computed in $\tilde{\mathcal{O}}_B(d\,\tau)$ bit operations and has coefficients of bitsize at most $\tau + 1$. The product $f\,g$ can be computed in $\tilde{\mathcal{O}}_B(d\,\tau)$ bit operations and has coefficients of bitsize in $\mathcal{O}(\tau + \log d)$. Finally, for the *Euclidean Division*, the computation of $q,\ r \in \mathbb{Z}[x]$ such that $f = q\,g + r$ with $\deg(r) < \deg(g)$ can be done using $\tilde{\mathcal{O}}_B(d^2\,\tau)$ bit operations. The polynomials q and r have degrees at most d and coefficients of bitsize bounded by $\mathcal{O}(d\,\tau)$.

Sign variation Let $\mathbf{D}$ be an *ordered domain*, namely, an integral domain with a *total order* $\leq$ satisfying the following two properties:

- $d_1 \leq d_2$ yields $d_1 + d_3 \leq d_2 + d_3$ all $d_1,\ d_2,\ d_3 \in \mathbf{D}$
- $0 \leq d_1$ and $0 \leq d_2$ imply $0 \leq d_1\,d_2$ for all $d_1,\ d_2 \in \mathbf{D}$.

Let $\mathbf{R}$ be a *real closed field* containing $\mathbf{D}$, namely, an ordered field that has the *intermediate value property*: if a polynomial $P \in \mathbf{R}[x]$ changes sign on an interval, i.e., $P(a) < 0 < P(b)$ for some $a, b \in \mathbf{R}$, then P has a zero in the interval, i.e., $P(c) = 0$ for some $a < c < b$. More properties about real closed fields can be found in [6]. The prototypical example of a real closed field is the field of real numbers $\mathbb{R}$.

Most of the discussions and methods presented in the following sections aim at studying the notion of variations of signs in a sequence of scalars, particularly in the sequence of polynomial coefficients. We thus state the following definition.

Definition 2.1. We define the *sign* $\operatorname{sign}(d)$ of an element $d \in \mathbf{R}$ as follows:

$$\operatorname{sign}(d) = \begin{cases} 0 & \text{if} \quad d = 0, \\ 1 & \text{if} \quad d \neq 0, \ d \geq 0, \\ -1 & \text{if} \quad d \neq 0, \ -d \geq 0. \end{cases}$$

Let $d = (d_0, d_1, \ldots, d_n)$ be a finite sequence of non zero elements in $\mathbf{R}$. The number of sign variations in d, denoted by $\mathbf{V}(d)$, is defined by induction over an integer k by:

$$\begin{cases} \mathbf{V}(d_1) & = & 0, \\ \mathbf{V}(d_1, \ldots, d_k) & = & \begin{cases} \mathbf{V}(d_1, \ldots, d_{k-1}) + 1 & \text{if} \quad \operatorname{sign}(d_{k-1}\, d_k) = -1, \\ \mathbf{V}(d_1, \ldots, d_{k-1}) & \text{else.} \end{cases} \end{cases}$$

We can generalise Definition 2.1 to any finite sequence of real numbers, where at least one element is not equal to 0, by eliminating the zeros occurring in the set. Let $(d_0, \ldots, d_n)$ be a finite sequence of real numbers. We write $\mathbf{V}(d_0, \ldots, d_n)$ for the number of sign variations in the set. In contrast, let $\mathbf{P}(d_0, \ldots, d_n)$ be the number of *sign permanences* in the sequence $(d_0, \ldots, d_n)$, namely, the number of consecutive signs $\{+, +\}$ or $\{-, -\}$.

Example 19. If we consider $a = (1, -1, 2, 0, 0, 3, 4, -5, -2, 0, 3)$, then we get:

$$\begin{cases} \mathbf{V}(a) & = & \mathbf{V}(1, 0, 0, -2, 3, 4, 0, -2, -3) & = & \mathbf{V}(1, -2, 3, 4, -2, -3) & = & 3, \\ \mathbf{P}(a) & = & \mathbf{P}(1, 0, 0, -2, 3, 4, 0, -2, -3) & = & \mathbf{P}(1, -2, 3, 4, -2, -3) & = & 2. \end{cases}$$

Definition 2.2. Let $P = \sum_{i=0}^{n} a_i x^i \in \mathbb{R}[x]$, then we set $\mathbf{V}(P) = \mathbf{V}(a_1, \ldots, a_n)$.

Critical points Let $\mathcal{C}$ be a plane curve defined by a polynomial $P(x, y)$, i.e., $\mathcal{C} = \{(x,\, y) \in \mathbb{C}^2 \mid P(x, y) = 0\}$, and $t_p = \left(\frac{\partial P}{\partial y}, -\frac{\partial P}{\partial x}\right)^T$ denotes the tangent vector of $\mathcal{C}$ at p. Then, the point $p \in \mathcal{C}$ is called:

1. An *x-critical point* if $t_p\,(1,\ 0)^T = \frac{\partial P(p)}{\partial y} = 0$.

2. A *y-critical point* if $t_p\,(0,\ 1)^T = -\frac{\partial P(p)}{\partial x} = 0$.

3. A *singular point* if $t_p = (0,\ 0)^T$, i.e., $\frac{\partial P(p)}{\partial x} = \frac{\partial P(p)}{\partial y} = 0$.

4. A singular point q is called *isolated* if there is a real neighborhood of q that does not contain other points on the curve. Equivalently, no real tangents of the curve exists at the singular point. If p is a singular point, then it is isolated if:

$$\frac{\partial^2 P(p)}{\partial x^2}\,\frac{\partial^2 P(p)}{\partial y^2} - \frac{\partial^2 P(p)}{\partial x\,\partial y} > 0.$$

   For instance, the curve defined by $y^2 - x^4 + 4\,x^2 = 0$ (or $y^3 + y^2 + x^2 = 0$) has an isolated singular point at $(0,\ 0)$.

For more details, see, e.g., [14, 89].

A critical point possessing real coordinates is called a *real critical point*. Finally, we shall simply call a *real point* any point on the curve possessing a real coordinate $(x,\ y)$. All four types of real critical points are shown in Figure 2.1.

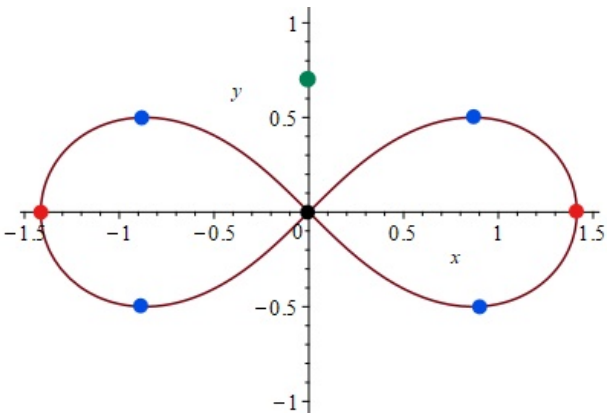

Figure 2.1: Blue dots represent y-critical points, red dots represent x-critical points, the black dot represents a singular point, and the green dot represents an isolated singular point.

2.2 Greatest common divisor

We recall the definition of the *greatest common divisor* (gcd) of two univariate polynomials and give some of its important properties that are used throughout this work. We also give

complexity results on the size and the computation of the gcd of two univariate polynomials with integer coefficients.

Definition 2.3. Let P and Q be two polynomials in $\mathbb{D}[x]$. A *greatest common divisor* of P and Q, denoted by $\gcd(P, Q)$, is a polynomial that divides both P and Q and is a multiple of every common divisor of P and Q.

Note that a gcd of two polynomials P and Q is unique up to an invertible element in $\mathbb{D}$.

When $\mathbb{D}[x]$ is an *Euclidean ring*, a greatest common divisor of P and Q can be computed by applying successive Euclidean divisions. We recall that an Euclidean ring $\mathbb{A}$ is an integral domain where there exists an *Euclidean division*, i.e., an Euclidean function over $\mathbb{A}$, defined as a map ν from $\mathbb{A}\backslash\{0\}$ to $\mathbb{N}$, exists and verifies the following properties:

1. For all $a \in \mathbb{A}$, $b \in \mathbb{A} \setminus \{0\}$, there exist q, $r \in \mathbb{A}$ such that $a = b\,q + r$, where either $r = 0$ or $\nu(r) < \nu(b)$.

2. For all a, $b \in \mathbb{A} \setminus \{0\}$, $\nu(b) \leq \nu(a\,b)$.

Some examples of Euclidean domains include any field $\Bbbk$ where ν can be defined by $\nu(x) = 1$ for all $x \in \Bbbk\backslash\{0\}$, the ring of integers $\mathbb{Z}$ where ν can be defined by $\nu(n) = |n|$ for all $n \in \mathbb{Z}$, and the ring $\Bbbk[x]$ of polynomials in x over a field $\Bbbk$ where for each nonzero polynomial P, $\nu(P)$ is defined by the degree of P.

In the rest of this section, we suppose that $\mathbb{D}[x]$ is an Euclidean domain.

Following the notation stated above, we denote r by $a\,\mathtt{rem}\,b$. With this notation, we recall a classic algorithm, known as the *Euclidean Algorithm*, which computes the gcd of two polynomials f and $g \neq 0$ in $\mathbb{D}[x]$.

In other words, the resultant sequence $\{r_0, \ldots, r_k\}$ of remainders in the Euclidean Algorithm – called the *Euclidean remainder sequence* of P and Q – is such that the last non-vanishing element r_k of the sequence is the gcd of P and Q in $\mathbb{D}[x]$ up to multiplication by a non-zero element in $\mathbb{D}$. The proof of exactness of this algorithm is mainly based on the following equalities:

$$\gcd(P, Q) = \gcd(r_2, Q) = \gcd(r_3, r_2) = \cdots = \gcd(r_{i-1}, r_i). \tag{2.1}$$

More details can be found in [31].

Algorithm 1 Classical Euclidean Algorithm

Input: Two univariate polynomials P and Q in $\mathbb{D}[x]$
Output: A greatest common divisor of P and Q in $\mathbb{D}[x]$

1. $r_0 := P$, $r_1 := Q$;
2. `while` $r_i \neq 0$ `do`
   (a) $r_{i+1} := r_{i-1}\,\mathtt{rem}\,r_i$;
   (b) $i := i + 1$;
3. `return` r_{i-1};

Example 20. Let $P = r_0 = x^4 - 1$ and $Q = r_1 = x^6 - 1$ in $\mathbb{R}[x]$. Applying Euclidean Algorithm yields:

$$\begin{aligned} x^4 - 1 &= 0\,(x^6 - 1) + x^4 - 1, \\ x^6 - 1 &= x^2\,(x^4 - 1) + x^2 - 1, \\ x^4 - 1 &= (x^2 + 1)\,(x^2 - 1) + 0. \end{aligned}$$

Following (2.1), we have

$$\begin{aligned} \gcd(x^4 - 1,\, x^6 - 1) &= \gcd(x^6 - 1,\, x^4 - 1) = \gcd(x^2 - 1,\, x^4 - 1) = \gcd(x^2 - 1,\, 0) \\ &= x^2 - 1, \end{aligned}$$

which is the last non-vanishing remainder in the Euclidean remainder sequence:

$$\{x^4 - 1,\, x^6 - 1,\, x^4 - 1,\, x^2 - 1,\, 0\}.$$

Theorem 2.1 (Bézout's identity). [7]. Let P, $Q \in \mathbb{D}[X]$, where $\deg(P) = p$ and $\deg(Q) = q$. Let $G = \gcd(P, Q)$. If $\deg(G) = g$, then there exist U, $V \in \mathbb{D}[x]$ with $\deg(U) < q - g$ and $\deg(V) < p - g$ such that $UP + VQ = G$.

Lemma 2.1. Let P, $Q \in \mathbb{D}[X]$, where $\deg(P) = p$ and $\deg(Q) = q$, and $G = \gcd(P, Q)$. Then, $\deg(G) \geq 1$, i.e., $G \notin \mathbb{D}$, if and only if there exist U, $V \in \mathbb{D}$ such that $\deg(U) < p$, $\deg(V) < q$ and $UP + VQ = 0$.

Having stated again the notion of the greatest common divisor, the gcd-*free part* of a polynomial P with respect to Q is defined as $P/\gcd(P, Q)$. In particular, when $Q = \frac{dP}{dx}$, the gcd-free part of P with respect to Q is the *square-free part* of P, denoted by $\bar{P}$, that is the divisor of P of maximum degree that has no square factors, provided that the *characteristic*

of the coefficients ring is zero or sufficiently large (e.g., larger than the degree of P).

Theorem 2.2. [7, Corollary 10.12 & Remark 10.19] Let $P, Q \in \mathbb{Z}[x]$, where $\max\big(\deg(P), \deg(Q)\big) \leq d$ and $\max(\tau_P, \tau_Q) \leq \tau$. Let $G = \gcd(P, Q) \in \mathbb{Z}[x]$. Then, G can be computed in $\tilde{\mathcal{O}}_B(d^2\,\tau)$ bit operations and its coefficients are of bitsize bounded by $\mathcal{O}(d + \tau)$. The same bounds hold for the computation of the gcd-free part of P with respect to Q along with its bitsize.

The following corollary is a refinement of Theorem 2.2 in the case of two polynomials with different degrees and bitsizes.

Corollary 2.1. [67, Corollary 5.2] Let P and Q be two polynomials in $\mathbb{Z}[x]$ with $\text{size}(P) = (p,\, \tau_p)$ and $\text{size}(Q) = (q,\, \tau_q)$. Then, a gcd of P and Q of bitsize $\mathcal{O}\big(\min(p + \tau_P,\, q + \tau_Q)\big)$ can be computed in $\tilde{\mathcal{O}}_B\big(\max(p,\, q)(p\,\tau_Q + q\,\tau_P)\big)$ bit operations. A gcd-free part of P with respect to Q, of bitsize $\mathcal{O}(p + \tau_P)$, can be computed in the same bit complexity.

2.3 Resultant of two polynomials

Let p, q be two positive integers and $P = a_p\,x^p + \cdots + a_0$, $Q = b_q\,x^q + \cdots + b_0$ be two polynomials of $\mathbb{D}[x]$ of degrees p and q respectively. Suppose for instance that $0 < p < q$, then the following matrix is called the *Sylvester matrix* of P and Q with respect to x.

$$L = \begin{pmatrix} a_p & a_{p-1} & \cdots & a_0 & & & & \\ & a_p & a_{p-1} & \cdots & a_0 & & & \\ & & \ddots & \ddots & & \ddots & & \\ & & & \ddots & \ddots & & \ddots & \\ & & & & a_p & a_{p-1} & \cdots & a_0 \\ b_q & b_{q-1} & \cdots & \cdots & \cdots & b_0 & & \\ & b_q & b_{q-1} & \cdots & \cdots & \cdots & b_0 & \\ & & b_q & b_{q-1} & \cdots & \cdots & \cdots & b_0 \end{pmatrix} \begin{matrix} \left.\vphantom{\begin{matrix}a\\a\\a\\a\\a\end{matrix}}\right\} q \\ \left.\vphantom{\begin{matrix}a\\a\\a\\a\end{matrix}}\right\} p \end{matrix}$$

Note that L is a $(p + q) \times (p + q)$-matrix. Its determinant is called the *(Sylvester) resultant* of P and Q with respect to x, denoted by $\text{Res}(P, Q, x)$. The resultant $\text{Res}(P, Q, x)$ of P and Q is an integer polynomial in the coefficients of P and Q. See, e.g., [32, Chapter 3].

Let $\mathbb{D}[x]_{<i}$ be the ring of polynomials in $\mathbb{D}[x]$ of degrees strictly less than i. The above

matrix L is the transpose of the matrix associated to the linear map

$$\begin{array}{ccc} \mathbb{D}[x]_{<q} \times \mathbb{D}[x]_{<p} & \longrightarrow & \mathbb{D}[x]_{<p+q} \\ (U, V) & \longmapsto & UP + VQ, \end{array}$$

with respect to the *standard basis* $\{x^i\}_{i=0,\ldots,q-1}$ (resp., $\{x^j\}_{j=0,\ldots,p-1}$, $\{x^k\}_{i=0,\ldots,p+q-1}$) of $\mathbb{D}[x]_{<q}$ (resp., of $\mathbb{D}[x]_{<p}$, $\mathbb{D}[x]_{<p+q}$).

Theorem 2.3 (**Common factor property**). [32, Chapter 3] If P, $Q \in \mathbb{D}[x]$, then P and Q have a non-trivial common factor in $\mathbb{D}[x]$ if and only if $\mathrm{Res}(P, Q, x) = 0$.

Proof. According to Lemma 2.1, P and Q have a non-trivial common factor in $\mathbb{D}[x]$ if and only if there exist two polynomials U, $V \in \mathbb{D}[x]$, with $\deg(U) < q$ and $\deg(V) < p$, such that $UP + VQ = 0$. Thus, the proof follows from the definition of the resultant as the determinant of the Sylvester matrix L. □

Theorem 2.4 (**Elimination property**). [32, Chapter 3] If P, $Q \in \mathbb{D}[x]$, then there exist polynomials A, $B \in \mathbb{D}[x]$ such that $A\,P + B\,Q = \mathrm{Res}(P, Q, x)$. The coefficients of A and B are integer polynomials in the coefficients of P and Q.

Example 21. We consider the polynomials $P = x^4 - 1$ and $Q = x^6 - 1$ of Example 20. Then, we have:

$$\mathrm{Res}\big(x^4 - 1, x^6 - 1, x\big) = \det \begin{pmatrix} 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix} = 0.$$

According to Theorem 2.3, P and Q have a non-trivial common factor in $\mathbb{R}[x]$.

Example 22. Let $P = x^3 + x - 1$, and $Q = 2\,x^2 + 3\,x + 7$. Then, we have:

$$\mathrm{Res}\big(x^3 + x - 1, 2\,x^2 + 3\,x + 7, x\big) = \det \begin{pmatrix} 1 & 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & 1 & -1 \\ 2 & 3 & 7 & 0 & 0 \\ 0 & 2 & 3 & 7 & 0 \\ 0 & 0 & 2 & 3 & 7 \end{pmatrix} = 159.$$

By Theorem 2.3, P, $Q \in \mathbb{R}[x]$ have no non-trivial common factor in $\mathbb{R}[x]$.

In particular, let us consider P and Q to be two bivariate polynomials with integer coefficients, i.e., $P,\ Q \in \mathbb{Z}[x, y]$. In this case, for instance, one can see these polynomials as univariate polynomials in x with coefficients in $\mathbb{Z}[y]$. Set $\deg_x(P) = p$ and $\deg_x(Q) = q$. Thus, we have $L \in \mathbb{Z}[y]^{(p+q)\times(p+q)}$ and $\mathrm{Res}(P, Q, x) = \det(L) \in \mathbb{Z}[y]$. In this case, we say that $\mathrm{Res}(P, Q, x)$ is the *resultant polynomial* of P and Q with respect to the variable x. The following proposition shows that the resultant polynomial of P and Q with respect to a variable, to say x, denoted by $\mathrm{Res}(P, Q, x)$, can embody the y-coordinates of the solutions of the bivariate polynomial system $\{P = 0,\ Q = 0\}$.

Proposition 2.1. Let P and Q be two polynomials in $\mathbb{Z}[x, y]$ and $\mathrm{Res}(P, Q, v)$ be their resultant with respect to the variable $v \in \{x, y\}$. Let $\mathrm{Lc}_v(P) = a_p$ and $\mathrm{Lc}_v(Q) = b_q$, where $a_p, b_q, \in \mathbb{Z}[u]$, $u \in \{x, y\} \setminus \{v\}$. Moreover, let $\alpha \in \mathbb{C}$. Then, the following two statements are equivalent:

1. $\mathrm{Res}(P, Q, v)(\alpha) = 0$.
2. $a_p(\alpha) = b_q(\alpha) = 0$ or there exists $\beta \in \mathbb{C}$ such that $(\alpha,\ \beta) \in V\big(\langle P,\ Q\rangle\big)$.

Example 23. We consider the polynomials $P = x\,y - 1$ and $Q = x^2 + y^2 - 4$. In this situation, the polynomials depend on two variables, but if we regard P and Q as univariate polynomials in x whose coefficients are polynomials in y, then we can compute the resultant with respect to x to obtain:

$$\mathrm{Res}\big(x\,y - 1, x^2 + y^2 - 4, x\big) = \det \begin{pmatrix} y & -1 & 0 \\ 0 & y & -1 \\ 1 & 0 & y^2 - 4 \end{pmatrix} = y^4 - 4\,y^2 + 1.$$

By Theorem 2.4, there are polynomials A, $B \in \mathbb{R}[x, y]$ satisfying:

$$A\,P + B\,Q = y^4 - 4\,y^2 + 1.$$

This means that the polynomial $y^4 - 4\,y^2 + 1$ vanishes at any common solution of $\{P = 0,\, Q = 0\}$. Furthermore, while considering the y-projection of the solutions of $\{P = 0, Q = 0\}$, we see that $\mathrm{Lc}_x(P) = y$ and $\mathrm{Lc}_x(Q) = 1$ do not both vanish at any real root of $y^4 - 4\,y^2 + 1$. Hence, Proposition 2.1 implies that the solutions of the polynomial $y^4 - 4\,y^2 + 1 = 0$ correspond to the y-projection of the solutions of the bivariate polynomial system $\{P = 0,\, Q = 0\}$.

Example 24. If we consider $P = x\,y - 1$ and $Q = x^2\,y - 2$, then we have:

$$\mathrm{Res}\big(x\,y - 1, x^2\,y - 2, x\big) = \det \begin{pmatrix} y & -1 & 0 \\ 0 & y & -1 \\ y & 0 & -2 \end{pmatrix} = -y\,(2\,y - 1).$$

In this case, $\mathrm{Res}(P, Q, x)$ vanishes for $y = 0$ and $y = 1/2$. However, $y = 0$ cannot be a y-projection of a solution of $\{P = 0,\, Q = 0\}$ since $P(x, 0) = -1$ and $Q(x, 0) = -2$. But we can see that $y = 0$ is a common zero of $\mathrm{Lc}_x(P) = y$ and $\mathrm{Lc}_x(Q) = y$. Whilst $y = 1/2$ is a y-coordinate of a solution of the polynomial system $\{P = 0,\, Q = 0\} = \{(2,\, 1/2)\}$.

Finally, let us introduce the concept of *discriminant* of a polynomial.

Definition 2.4. [6] Let $P = a_p\,x^p + \cdots + a_0 \in \mathbb{D}[x]$. Then, the *discriminant* of P is defined by:

$$\mathrm{discrim}(P) = (-1)^{\frac{p\,(p-1)}{2}}\,\mathrm{Lc}_x(P)^{-1}\,\mathrm{Res}(P, P', x).$$

Hence, up to a sign and the leading coefficient of P, the discriminant of P is equal to $\mathrm{Res}(P, P', x)$. Using Theorem 2.3, we get that P and P' has a common factor if and only if $\mathrm{discrim}(P) = 0$. Hence, P has a multiple factor, namely, P is divisible by Q^2 where $\deg_x(Q) > 0$, if and only if $\mathrm{discrim}(P) = 0$.

In the next section, we shall generalise the concept of resultant to the so-called *subresultant sequence*.

2.4 Subresultant sequence

In this section, we introduce the concept of the *subresultant sequence* of two univariate polynomials P and Q in the variable x with coefficients in $\mathbb{D}$. Up to a factor in $\mathbb{D}$, each subresultant is a polynomial in $\mathbb{D}[x]$ equal to a remainder of a variant of the classical Euclidean algorithm associated with P and Q. Such factors are mainly determinants of structured

sub-matrices of the *Sylvester matrix* of P and Q. The least non-vanishing subresultant of P and Q is their resultant which is, by structure, closely related to $\gcd(P,\, Q)$.

Again, let $P = a_p\, x^p + \cdots + a_0$, $Q = b_q\, x^q + \cdots + b_0$ be two polynomials of $\mathbb{D}[x]$ of degrees p and q respectively and let $\lambda = \min(p,\ q)$. For any $0 \leq i < \lambda$, let L_i be the sub-matrix of L formed by removing the bottom i rows that include the coefficients of P and the bottom i rows that include the coefficients of Q. Note that L_i is a $(p+q-2\,i) \times (p+q)$ matrix. For $j = 0, \ldots, i$, let $L_{i,j}$ be the submatrix of L_i consisting of the first $p+q-2\,i-1$ columns and the $(p+q-2\,i+j)^{\text{th}}$ columns. The following polynomial is called the ith *subresultant* of P and Q:

$$\mathrm{Sres}_i(P, Q, x) = \sum_{j=0}^{i} \det(L_{i,j})\, x^{i-j}.$$

Example 25. Let $P = x^3 + 2\,x^2 + 1$ and $Q = x^4 + x + 1$. Hence, $p = \deg(P) = 3$ and $q = \deg(Q) = 4$, and thus, $\lambda = \min(3, 4) = 3$. The Sylvester matrix of P and Q with respect to x is then defined by:

$$L = \begin{pmatrix} 1 & 2 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 2 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 2 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

Moreover, we have

$$L_1 = \begin{pmatrix} 1 & 2 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 2 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \end{pmatrix},$$

and thus:

$$\det L_{1,1} = \det \begin{pmatrix} 1 & 2 & 0 & 1 & 0 \\ 0 & 1 & 2 & 0 & 0 \\ 0 & 0 & 1 & 2 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix} = -8,\ \det L_{1,0} = \det \begin{pmatrix} 1 & 2 & 0 & 1 & 0 \\ 0 & 1 & 2 & 0 & 1 \\ 0 & 0 & 1 & 2 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix} = -12.$$

We also have

$$L_2 = \begin{pmatrix} 1 & 2 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 2 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix},$$

which yields

$$\det L_{2,2} = \det \begin{pmatrix} 1 & 2 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} = 3, \quad \det L_{2,1} = \det \begin{pmatrix} 1 & 2 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} = 0,$$

and:

$$\det L_{2,0} = \det \begin{pmatrix} 1 & 2 & 0 \\ 0 & 1 & 2 \\ 1 & 0 & 0 \end{pmatrix} = 4.$$

Hence, we have:

- $\mathrm{Sres}_0(P, Q, x) = \mathrm{Res}(P, Q, x) = \det(L) = 43$,
- $\mathrm{Sres}_1(P, Q, x) = \det L_{1,0}\, x + \det L_{1,1} = -12\, x - 8$,
- $\mathrm{Sres}_2(P, Q, x) = \det L_{2,0}\, x^2 + \det L_{2,1} x + \det L_{2,2} = 4\, x^2 + 3$.

Let *the principal subresultant coefficient* of Sres_i be defined by:

$$\mathrm{sres}_i(P, Q, x) = \mathrm{Lc}_x\big(\mathrm{Sres}_i(P, Q, x)\big).$$

Note that for $i = 0$, L_i coincides with L and thus $\mathrm{Sres}_0 = \mathrm{sres}_0 = \mathrm{Res}(P, Q, x)$.

We extend the definition of subresultants and principal subresultant coefficients as follows.

- If $p \geq q$, then we set:

$$\begin{cases} \mathrm{Sres}_{\lambda+1}(P, Q, x) = P, \\ \mathrm{Sres}_{\lambda}(P, Q, x) = Q, \\ \mathrm{sres}_{\lambda+1}(P, Q, x) = a_p, \\ \mathrm{sres}_{\lambda}(P, Q, x) = b_q. \end{cases}$$

- If $p < q$, then we set:

$$\begin{cases} \mathrm{Sres}_{\lambda+1}(P, Q, x) = Q, \\ \mathrm{Sres}_{\lambda}(P, Q, x) = P, \\ \mathrm{sres}_{\lambda+1}(P, Q, x) = b_q, \\ \mathrm{sres}_{\lambda}(P, Q, x) = a_p. \end{cases}$$

The following theorem explains the *specialization property* of subresultants sequence. This property is very useful for our proposed methods for the computation of the L^∞-norm. Let $\phi : \mathbb{D} \longrightarrow \mathbb{D}'$ be a *ring homomorphism* (namely, $\phi(1) = 1$, $\phi(d_1 + d_2) = \phi(d_1) + \phi(d_2)$ and $\phi(d_1\, d_2) = \phi(d_1)\, \phi(d_2)$ for all d_1, $d_2 \in \mathbb{D}$) that induces a ring homomorphism from $\mathbb{D}[x]$ to $\mathbb{D}'[x]$ defined by mapping $d \in \mathbb{D}$ to $\phi(d)$ and x to x, also simply denoted by ϕ.

Theorem 2.5. Let P and Q be two polynomials of $\mathbb{D}[x]$. If $\phi(\mathrm{Lc}_x(P)) \neq 0$ and $\phi(\mathrm{Lc}_x(Q)) \neq 0$, then we have:

$$\phi\big(\mathrm{Sres}_j(P, Q, x)\big) = \mathrm{Sres}_j\big(\phi(P), \phi(Q), x\big).$$

Proof. The proof immediately follows from the following identity

$$\phi\big(\det\big(\mathrm{Sres}_k(P, Q, x)\big)\big) = \det\big(\mathrm{Sres}_k\big(\phi(P), \phi(Q), x\big)\big)$$

which comes from *Leibniz formula* for determinants (i.e., the standard formula $\det\big((a_{i,j})_{1\leq i,\, j\leq n}\big) = \sum_{\sigma\in S_n} \epsilon(\sigma) \prod_{i=1}^{n} a_{i,\sigma(i)}$, where $\epsilon(\sigma)$ denotes the *signature* of the *permutation* σ which belongs to the *symmetric group* S_n) and the definition of ring homomorphisms. The conditions $\phi\big(\mathrm{Lc}_x(P)\big) \neq 0$ and $\phi\big(\mathrm{Lc}_x(Q)\big) \neq 0$ are used to ensure that the matrix dimensions do not change. □

Example 26. We consider the following bivariate polynomials

$$P(x, y) = x^4 - (y + 2)\, x^3 + (2\, y + 1)\, x^2 - (y + 2)\, x + 2\, y \in \mathbb{Z}[x, y]$$

and $Q(x, y) = \dfrac{\partial P(x, y)}{\partial x}$ and the ring homomorphism $\phi : \mathbb{Z}[x, y] \longrightarrow \mathbb{Z}[x, y]$ defined by $\phi(P(x, y)) = P(x, \alpha)$, where $\alpha \in \mathbb{Q}$. Then, we have:

- $\mathrm{Sres}_0(\phi(P), \phi(Q), x) = \mathrm{Sres}_0(P, Q, x)(x, \alpha) = -100\, (\alpha - 2)^2\, (\alpha^2 + 1)^2$,
- $\mathrm{Sres}_1(\phi(P), \phi(Q), x) = \mathrm{Sres}_1(P, Q, x)(x, \alpha)$
  $= (-2\, \alpha^4 + 40\, \alpha^3 - 74\, \alpha^2 + 40\, \alpha + 128)\, x - 4\, (\alpha + 2)\, (4\, \alpha^3 - 13\, \alpha^2 + 24\, \alpha - 3)$,
- $\mathrm{Sres}_2(\phi(P), \phi(Q), x) = \mathrm{Sres}_2(P, Q, x)(x, \alpha)$
  $= (-3\, \alpha^2 + 4\, \alpha - 4)\, x^2 + 2\, (\alpha + 2)\, (2\, \alpha - 5)\, x - \alpha^2 + 28\, \alpha - 4$,

- $\mathrm{Sres}_3(\phi(P), \phi(Q), x) = \mathrm{Sres}_3(P, Q, x)(x, \alpha) = Q(x, \alpha)$
  $= 4\,x^3 - (3\,\alpha + 6)\,x^2 + (4\,\alpha + 2)\,x - \alpha - 2,$

- $\mathrm{Sres}_4(\phi(P), \phi(Q), x) = \mathrm{Sres}_4(P, Q, x)(x, \alpha) = P(x, \alpha)$
  $= x^4 - (\alpha + 2)\,x^3 + (2\,\alpha + 1)\,x^2 - (\alpha + 2)\,x + 2\,\alpha.$

Hereafter, when there is no ambiguity, we simply denote $\mathrm{Sres}_j(P, Q, x)$ and $\mathrm{sres}_j(P, Q, x)$ by Sres_j and sres_j respectively. Moreover, we set

$$\mathrm{sres}_{i,j} = \mathrm{Lc}_{x^j}(\mathrm{Sres}_i)$$

in the sense that $\mathrm{sres}_{i,j}$ is the coefficient of the Sres_i for the power x^j.

Corollary 2.2. [7] Let P and Q be two polynomials of $\mathbb{D}[x]$. If r_{j-1} and r_j are two consecutive polynomials in the Euclidean remainder sequence of P and Q of degree d_{j-1} and degree d_j respectively, then $\mathrm{Sres}_{d_{j-1}-1}(P, Q, x)$ and $\mathrm{Sres}_{d_j}(P, Q, x)$ are proportional to r_j.

In particular, one can determine the degree of $\gcd(P, Q)$ by recognizing the vanishing principle subresultants.

Proposition 2.2. [7, 41] Let P and Q be two polynomials of $\mathbb{D}[x]$ where $\mathbb{D}$ is an integral domain, then

- if the polynomials P and Q have a common divisor of degree k, then $\mathrm{sres}_i(P, Q, x) = 0$ for $i = 0, \ldots, k-1$,

- if $\mathrm{sres}_i(P, Q, x) = 0$ for $i = 0, \ldots, k-1$ and $\mathrm{sres}_k \neq 0$, then $\gcd(P, Q) = \mathrm{Sres}_k(P, Q, x)$ over $\mathbb{F}_{\mathbb{D}}$.

Example 27. We consider again Example 20. By doing the computation of the subresultant sequence of P and Q with respect to x, we obtain:

- $\mathrm{Sres}_0(P, Q, x) = 0,$

- $\mathrm{Sres}_1(P, Q, x) = 0,$

- $\mathrm{Sres}_2(P, Q, x) = x^2 - 1,$

- $\mathrm{Sres}_3(P, Q, x) = -x^2 + 1.$

By Proposition 2.2 we see again that $\gcd(P, Q) = \mathrm{Sres}_2(P, Q, x) = x^2 - 1$.

To sum up, up to multiplicative factors in $\mathbb{D}$, the subresultants of P and Q are either null or equal to polynomials figuring in the Euclidean remainder sequence of P and Q.

We finally state a useful property of subresultants that is a consequence of the specialization property. We consider two positive integers $p \geq q$ and two bivariate polynomials with integer coefficients P, $Q \in \mathbb{Z}[x, y]$ with $\deg_y(P) = p$ and $\deg_y(Q) = q$. As discussed in Proposition 2.1, the resultant polynomial $\mathrm{sres}_0(P, Q, y) = \mathrm{Res}(P, Q, y)$ embodies the x-coordinates of the solutions of the polynomial system $\{P,\ Q\}$ or points where their leading coefficients vanish.

Following the definition of the subresultants sequence and taking advantage of the specialization property, we can compute the gcd polynomial of P and Q over these x-coordinates for which the leading coefficients of P and Q do not both vanish.

Theorem 2.6. [41] Let P, $Q \in \mathbb{Z}[x, y]$ and let us note $a_p = \mathrm{Lc}_y(P)$ and $b_q = \mathrm{Lc}_y(Q)$, where $a_p, b_q \in \mathbb{Z}[x]$. For any $x_i \in \mathbb{R}$ such that $a_p(x_i) \neq 0$ and $b_q(x_i) \neq 0$, we can consider the polynomials $P_i = P(x_i, y)$ and $Q_i = Q(x_i, y)$ in $\mathbb{R}[y]$. Then, the least non-vanishing $\mathrm{Sres}_k(P_i, Q_i, y)$ for k increasing is equal to $\gcd(P_i, Q_i)$, up to a non-zero element in the fraction field $\mathbb{Z}(x_i)$ of $\mathbb{Z}[x_i]$.

For more details about the subresultant sequence, see [41].

Based on the above results, it is possible to efficiently compute the subresultant polynomials using a variant of the classical Euclidean algorithm [7, Algorithm 8.21]. This algorithm performs successive divisions and returns the sequence of the intermediate remainders. The next proposition gives bit complexity results concerning the size and the computation of the subresultant polynomials − with respect to one precise variable − of two polynomials with polynomial coefficients over $\mathbb{Z}$.

Proposition 2.3. [7, Proposition 8.46]. Let $P,\ Q \in \mathbb{Z}[x_1, \ldots, x_n][y]$ be two polynomials with a maximum coefficient bitsize τ, their degrees in y bounded by d_y and their degrees in the other variables x_i, $i \in \{1, \ldots, n\}$, bounded by d. Then, we have:

- The coefficients of $\mathrm{Sres}_i(P, Q, y)$ have coefficients bitsize $\tilde{\mathcal{O}}(d_y\, \tau)$.
- The degree in x_j of $\mathrm{Sres}_i(P, Q, y)$ is at most $2\, d\, (d_y - i)$.
- Any subresultant $\mathrm{Sres}_i(P, Q, y)$ can be computed in $\tilde{\mathcal{O}}(d^n\, d_y^{n+1})$ arithmetic operations and in $\tilde{\mathcal{O}}_B(d^n\, d_y^{n+2}\, \tau)$ bit operations.

In the next section, we define the *Sturm-Habicht sequence* which is equal to the subresultant sequence up to a sign. This sequence sometimes called *signed subresultant sequence*.

This sequence plays in important role in Chapter 3.1.3 as a powerful tool for the real root counting due to its stability under specializations and its controlled coefficients growth. Finally, it is worth mentioning that the relation between the Sturm-Habicht sequence and the subresultant sequence is similar to the relation between the Sturm sequence and the Euclidean remainder sequence.

2.5 Sturm-Habicht Sequence

Let $\mathbf{D}$ be an ordered domain and $\mathbf{R}$ a real closed field containing $\mathbf{D}$. We first introduce the *real root counting problem*:

Given $P \in \boldsymbol{D}[x]$, compute the number of roots of P in $\boldsymbol{R}$.

In [53], the authors studied an algorithm for the following problem:
Given P, $Q \in \boldsymbol{D}[x]$, compute:

$$\operatorname{card}\big(\{\alpha \in \boldsymbol{R} \mid P(\alpha) = 0,\, Q(\alpha) > 0\}\big) - \operatorname{card}\big(\{\alpha \in \boldsymbol{R} \mid P(\alpha) = 0,\ Q(\alpha) < 0\}\big). \tag{2.2}$$

Thus, when $Q = 1$, a solution of (2.2) gives a solution to the real root counting problem.

The first known algorithm solving the real root counting problem is due to C. Sturm [97], where he obtained the number of real roots in terms of the *Sturm sequence*, which is the Euclidean remainder sequence for P and its derivative up to sign changes. Then, in [98], Sylvester generalized Sturm's method to problem (2.2). Finally, in [58], C. Hermite developed a general theory giving the number of real roots as the signature of a *Hankel matrix*. However, many drawbacks are present in using Sturm sequence for computing (2.2). First of all, the computation of the Sturm sequence has bad numerical behaviours. Indeed, using exact arithmetic for a polynomial with integer coefficients, the polynomials in the Sturm sequence have rational coefficients whose size grows exponentially in the degree of the polynomials. Moreover, when the polynomial depends on parameters, the computation has no good specialization properties. Contrary to the polynomials in the Euclidean remainder sequence, the subresultants are stable under specialization and the size of their coefficients is well-controlled. Thus, by taking care of the signs, subresultants were found to be an advantageous tool for real root counting problem.

The technique defined below, used in Chapter 3.1.3, was developed in [53]. It is based on the *Sturm-Habicht sequence* introduced by W. Habicht in [55].

Definition 2.5. [53] Let $P, Q \in \mathbb{D}[x]$, $p = \deg(P)$, $q = \deg(Q)$, $v = p + q - 1$ and $\delta_k = (-1)^{\frac{k(k+1)}{2}}$ for all $k \in \mathbb{N}$. Then, *the Sturm-Habicht sequence* associated to P and Q is

defined by the list of polynomials $\big\{\mathrm{StHa}_j(P,Q)\big\}_{j=0,\ldots,v+1}$, where:

$$\begin{cases} \mathrm{StHa}_{v+1}(P,Q) = P, \\ \mathrm{StHa}_v(P,Q) = \dfrac{dP}{dx}\,Q, \\ \mathrm{StHa}_j(P,Q) = \delta_{v-j}\,\mathrm{Sres}_j\left(P, \dfrac{dP}{dx}\,Q,\ x\right),\ j \in \{0,\ldots,v-1\}. \end{cases}$$

The principal j^{th} Sturm-Habicht coefficient is then defined by:

$$\mathrm{stha}_j(P,Q) = \mathrm{Lc}_x\big(\mathrm{StHa}_j(P,Q)\big),\ j \in \{0,\ldots,v-1\}.$$

Example 28. We consider the univariate polynomial

$$P = x^6 - 3\,x^5 + 7\,x^4 - 15\,x^3 + 14\,x^2 - 12\,x + 8 \in \mathbb{Z}[x]$$

and $Q = 1$. Then, $v = 6 - 1 = 5$ and the Sturm-Habicht sequence $\big\{\mathrm{StHa}_j(P,Q)\big\}_{j=0,\ldots,6}$ associated to P and Q is defined by:

1. $\mathrm{StHa}_6(P,1) = P = x^6 - 3\,x^5 + 7\,x^4 - 15\,x^3 + 14\,x^2 - 12\,x + 8$,
   - $\mathrm{stha}_6(P,1) = 1$,
2. $\mathrm{StHa}_5(P,1) = \frac{dP}{dx} = 6\,x^5 - 15\,x^4 + 28\,x^3 - 45\,x^2 + 28\,x - 12$,
   - $\mathrm{stha}_5(P,1) = 6$,
3. $\mathrm{StHa}_4(P,1) = \delta_1\,\mathrm{Sres}_4\left(P, \frac{dP}{dx}, x\right) = (-1)\big(39\,x^4 - 186\,x^3 + 201\,x^2 - 276\,x + 252\big)$,
   - $\mathrm{stha}_4(P,1) = -39$,
4. $\mathrm{StHa}_3(P,1) = \delta_2\,\mathrm{Sres}_3\left(P, \frac{dP}{dx}, x\right) = (-1)\big(2620\,x^3 - 3072\,x^2 + 3616\,x - 4224\big)$,
   - $\mathrm{stha}_3(P,1) = -2620$,
5. $\mathrm{StHa}_2(P,1) = \delta_3\,\mathrm{Sres}_2\left(P, \frac{dP}{dx}, x\right) = (+1)\big(-78064\,x^2 - 88128\,x + 116672\big)$,
   - $\mathrm{stha}_2(P,1) = -78064$,
6. $\mathrm{StHa}_1(P,1) = \delta_4\,\mathrm{Sres}_1\left(P, \frac{dP}{dx}, x\right) = (+1)\big(12729600\,x - 11750400\big)$,
   - $\mathrm{stha}_1(P,1) = 12729600$,

7. $\mathrm{StHa}_0(P,1) = \mathrm{stha}_0(P,1) = \delta_5\,\mathrm{sres}_0\left(P, \frac{dP}{dx}, x\right) = 829440000.$

Example 29. We consider the bivariate polynomial

$$P(x,y) = x^4 - (y+2)\,x^3 + (2\,y+1)\,x^2 - (y+2)\,x + 2\,y \in \mathbb{Z}[x,y]$$

studied in Example 26 and the constant polynomial $Q = 1$. Then, $v = 4 - 1 = 3$ and the Sturm-Habicht sequence $\left\{\mathrm{StHa}_j(P,Q)\right\}_{j=0,\ldots,4}$ associated to P and Q is defined by:

1. $\mathrm{StHa}_4(P,1) = P = x^4 - (y+2)\,x^3 + (2\,y+1)\,x^2 - (y+2)\,x + 2\,y,$
   - $\mathrm{stha}_4(P,1) = 1,$
2. $\mathrm{StHa}_3(P,1) = \frac{dP}{dx} = 4\,x^3 - (3\,y+6)\,x^2 + (4\,y+2)\,x - y - 2,$
   - $\mathrm{stha}_3(P,1) = 4,$
3. $\mathrm{StHa}_2(P,1) = \delta_1\,\mathrm{Sres}_2(P, \frac{\partial P}{\partial x}, x) = (-1)\big((-3\,y^2 + 4\,y - 4)\,x^2 + 2\,(y+2)\,(2\,y-5)\,x - y^2 + 28\,y - 4\big),$
   - $\mathrm{stha}_2(P,1) = 3\,y^2 - 4\,y + 4,$
4. $\mathrm{StHa}_1(P,1) = \delta_2\,\mathrm{Sres}_1(P, \frac{\partial P}{\partial x}, x) = (-1)\big((-2\,y^4 + 40\,y^3 - 74\,y^2 + 40\,y + 128)\,x - 4\,(y+2)\,(4\,y^3 - 13\,y^2 + 24\,y - 3)\big),$
   - $\mathrm{stha}_1(P,1) = 2\,y^4 - 40\,y^3 + 74\,y^2 - 40\,y - 128,$
5. $\mathrm{StHa}_0(P,1) = \mathrm{stha}_0(P,1) = \delta_3\,\mathrm{sres}_0(P, \frac{\partial P}{\partial x}, x) = -100\,(y-2)^2\,(y^2+1)^2.$

2.5.1 Sturm-Habicht sequence and real roots of polynomials

Let $\mathbf{D}$ be an ordered domain and $\mathbf{R}$ its real closure. To establish the relation between the number of real zeros (zeros in $\mathbf{R}$) of a polynomial $P \in \mathbf{D}[x]$ and the polynomials in the Sturm-Habicht sequence of P and Q, where $Q \in \mathbf{D}[x]$, for every $\epsilon \in \{-1, 0, +1\}$, we first introduce the following integer:

$$c_\epsilon(P,Q) = \mathrm{card}\big(\{\alpha \in \mathbf{R} \mid P(\alpha) = 0,\ \mathrm{sign}\big(Q(\alpha)\big) = \epsilon\}\big).$$

We now study how to compute the integer $c_+(P,\ Q) - c_-(P,\ Q)$ knowing only the principal Sturm-Habicht coefficients associated to P and Q. For doing so, we define a *sign variation function*, called **SignVar**, applied on a finite sequence of elements in $\mathbf{R}$. Such

function computes the difference between the sign permanences and sign variations of that sequence, while taking into consideration the zeros occurring in that sequence.

Definition 2.6. [53] Let $a_0, a_1, \ldots, a_n$ be elements in $\mathbf{R}$, where $a_0 \neq 0$, with the following distribution of zeros

$$\{a_0, a_1, \ldots, a_n\} = \{a_0, \ldots, a_{i_1}, \overbrace{0, 0, \ldots, 0}^{k_1}, a_{i_1+k_1+1}, \ldots, a_{i_2}, \overbrace{0, 0, \ldots, 0}^{k_2},$$
$$a_{i_2+k_2+1}, \ldots, a_{i_3}, \overbrace{0, 0, \ldots, 0}^{k_3}, a_{i_{t-1}+k_{t-1}+1}, \ldots, a_{i_t}, \overbrace{0, 0, \ldots, 0}^{k_t}\},$$

where all the a_i's that have been written are not 0. Let i_0 and k_0 be such that $i_0 + k_0 + 1 = 0$ and note

$$\mathbf{S} = \sum_{s=1}^{t} \Big(\mathbf{P}\big(a_{i_{s-1}+k_{s-1}+1}, \ldots, a_{i_s}\big) - \mathbf{V}\big(a_{i_{s-1}+k_{s-1}+1}, \ldots, a_{i_s}\big) \Big),$$

where $\mathbf{P}$ and $\mathbf{V}$ are defined in Section 2.1. Then, we can define **SignVar** by

$$\mathbf{SignVar}\big(a_0, a_1, \ldots, a_n\big) = \mathbf{S} + \sum_{s=1}^{t-1} \epsilon_{i_s},$$

where:

$$\epsilon_{i_s} = \begin{cases} 0 & \text{if } k_s \text{ is odd,} \\ (-1)^{\frac{k_s}{2}} \operatorname{sign}\left(\dfrac{a_{i_s} + k_s + 1}{a_{i_s}}\right) & \text{if } k_s \text{ is even.} \end{cases}$$

Theorem 2.7. [53] If P and Q are polynomials in $\mathbb{D}[x]$ with $p = \deg(P)$, then:

$$\mathbf{SignVar}\big(\mathrm{stha}_p(P, Q), \ldots, \mathrm{stha}_0(P, Q)\big) = c_+(P, Q) - c_-(P, Q).$$

In particular, $\mathbf{SignVar}\big(\mathrm{stha}_p(P, 1), \ldots, \mathrm{stha}_0(P, 1)\big) = c_+(P, 1)$ gives the number of real zeros of P.

Remark 2.1. In sign variation functions such as $\mathbf{P}$, $\mathbf{V}$ or **SignVar**, the number occurring in the given sequence can be simply represented by its sign. For instance, $\mathbf{V}(130, -199, 0, 6) = \mathbf{V}(1, -1, 0, 1) = 2$ and thus, instead of writing $\mathbf{V}(130, -199, 0, 6)$, we can simply write $\mathbf{V}(1, -1, 0, 1)$.

Remark 2.2. When 0 does not belong to a finite sequence a, then $\mathbf{SignVar}(a)$ is simply equal to $\mathbf{P}(a) - \mathbf{V}(a)$ as it will be illustrated in Example 30.

Example 30. We consider again the univariate polynomial

$$P = x^6 - 3\,x^5 + 7\,x^4 - 15\,x^3 + 14\,x^2 - 12\,x + 8 \in \mathbb{Z}[x]$$

studied in Example 28. We aim at computing the number of its real roots by applying Theorem 2.7 to P and $Q = 1$. In this case, the number of real roots of P is given by $\mathbf{SignVar}\big(\mathrm{stha}_6(P,1),\ldots,\mathrm{stha}_0(P,1)\big)$ which is, by the Definition 2.5 and based on the computation done in Example 28, equal to:

$$\begin{aligned}
& \mathbf{SignVar}\big(\mathrm{sres}_6(P,P',x),\ldots,\delta_5\,\mathrm{sres}_0(P,P',x)\big)\\
= \;& \mathbf{SignVar}\big(1,\ 6,\ -39,\ -2620,\ -78064,\ 12729600,\ 829440000\big)\\
= \;& \mathbf{P}\big(1,\ 1,\ -1,\ -1,\ -1,\ 1,\ 1\big) - \mathbf{V}\big(1,\ 1,\ -1,\ -1,\ -1,\ 1,\ 1\big) = 4 - 2 = 2.
\end{aligned}$$

Hence, P admits only 2 real roots.

Example 31. We consider the following bivariate polynomial

$$P = x^4 - (y+2)\,x^3 + (2\,y+1)\,x^2 - (y+2)\,x + 2\,y \in \mathbb{Z}[x,y]$$

studied in Example 29. We can use the specialization property of the subresultant sequence regarding to the ring homomorphism $\phi : \mathbb{Z}[x,y] \longrightarrow \mathbb{Z}[x,y]$ defined by $\phi(P(x,y)) = P(x,\alpha)$, where $\alpha \in \mathbb{Q}$, and compute the number of real solutions of $\{P(x,y) = 0, y - \alpha = 0\}$ for any $\alpha \in \mathbb{Q}$ by computing $\mathrm{Sres}(P, \frac{\partial P}{\partial x}, x)$. By Definition 2.5 and using the computations done in Examples 26 and 29, we have:

$$\begin{aligned}
& \Big(\mathrm{stha}_4(P(x,\alpha),\ 1),\ldots,\mathrm{stha}_0(P(x,\alpha),\ 1)\Big)\\
= \;& \Big(\mathrm{sres}_4(P(x,\alpha),\ P'(x,\alpha),x),\ldots,\delta_3\,\mathrm{sres}_0(P(x,\alpha),\ P'(x,\alpha),x)\Big)\\
= \;& \Big(\mathrm{sres}_4(P,\ \tfrac{\partial P}{\partial x},x)(x,\alpha),\ldots,\delta_3\,\mathrm{sres}_0(P,\ \tfrac{\partial P}{\partial x},x)(x,\alpha)\Big)\\
= \;& \big(1, 4,\ 3\,\alpha^2 - 4\,\alpha + 4,\ 2\,\alpha^4 - 40\,\alpha^3 + 74\,\alpha^2 - 40\,\alpha - 128,\\
& \quad -100\,(\alpha-2)^2\,(\alpha^2+1)^2\big).
\end{aligned}$$

For instance, if $\alpha = 2$, then we get

$$\big(\mathrm{stha}_4(P(x,2),\ 1),\ldots,\mathrm{stha}_0(P(x,2),\ 1)\big) = (1,4,8,-200,0)$$

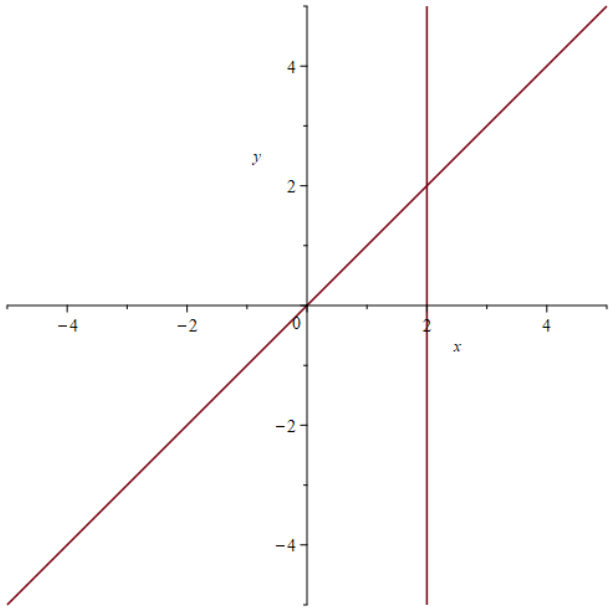

Figure 2.2: Real curve of $P(x, y) = x^4 - (y + 2)\, x^3 + (2\, y + 1)\, x^2 - (y + 2)\, x + 2\, y$

and thus, the number of real roots of $P(x, 2)$ is equal to:

$$\mathbf{SignVar}(1, 4, 8, -200, 0) = 1.$$

We can see that $\mathrm{Res}\big(P, \frac{\partial P}{\partial x}, x\big)$ admits only one real root $y = 2$. By noticing that $\mathrm{Lc}_x(P) = \mathrm{Lc}_x(\frac{\partial P}{\partial x}) = 1$, we can say that $P(x, 2)$ and $\frac{\partial P}{\partial x}(x, 2)$ has a non-trivial gcd in $\mathbb{Q}[x]$, which means that 2 is a real y-coordinate of a y-critical point of the curve defined by $P(x, y)$.

Now, if we consider $\alpha = 3$, then we get

$$\big(\mathrm{stha}_4(P(x, 3),\ 1), \ldots, \mathrm{stha}_0(P(x, 3),\ 1)\big) = (1, 4, 19, -500, -1000)$$

and thus, the number of real roots of $P(x, 3)$ is equal to:

$$\mathbf{SignVar}(1, 4, 19, -500, -1000) = 2.$$

This means that a real point with a y-coordinate equal to 3 (> 2) belongs to the curve defined by $P(x, y)$. Thus, in this particular case, since $y = 2$ is the maximal real y-projection of the y-critical points of the curve, we can simply conclude that the curve $P(x, y) = 0$ cannot be bounded in the y-direction. We can see the curve of $P(x, y) = 0$ in Figure 2.2.

2.6 Univariate polynomials and Root isolation

Given a square-free polynomial $P \in \mathbb{Q}[x]$ (see Section 2.2) and an interval containing only one of the real roots of P, the polynomial have opposite signs when evaluated over the interval end-points.

Remark 2.3. If P is not square-free, then the above remark is not true. For instance, if we consider $P = x^2$ and the interval $[-1, 1]$ which contains the only real root 0 of P, then $P(-1) = 1$ and $P(1) = 1$ have a constant sign. But the square-free part $\bar{P}$ of P, i.e., $\bar{P} = P/\gcd(P, P') = x$, satisfies that $\bar{P}(-1) = -1$ and $\bar{P}(1) = 1$ have opposite signs.

One of the oldest algorithms for root-finding, due to L. Kronecker, is mainly based on the following basic method:

1. Define an interval $[a, b]$ which contains all the real roots of $P \in \mathbb{Q}[x]$.
2. Compute the minimal distance between two real roots of P, say μ, and split the interval $[a, b]$ into two intervals of length less than μ.
3. Compute the sign of P at each end-point of the obtained intervals (we consider P to be square-free) to detect the existence of one real root.

Such a procedure is called a *binary search*. It is a root-finding algorithm that yield a simple and robust method for producing real roots. However, in general, it cannot certify having found all real roots and, theoretically, its complexity is exponential with respect to the polynomial degree.

Assuming that the real roots of the polynomial can be bounded, the root isolation algorithm can start with a well-defined interval containing all real roots of the given polynomial. Then, by providing a procedure for counting the real roots of a polynomial in an interval, without having to compute them, the root-finding algorithm can be extended into efficient algorithms for isolating all real roots of a polynomial.

We recall bounds on univariate polynomial roots and their separation that can both be computed directly by means of the polynomial coefficients.

Proposition 2.4. Let $P = \sum_{i=0}^{n} a_i\, x^i \in \mathbb{Q}[x]$ be a *monic*, namely, $a_n = 1$, and let suppose that $a_0 \neq 0$. If α is a complex root of P, then we have:

$$|\alpha| \;<\; 1 + \max_{i=0,\ldots,n} |a_i|.$$

Proof. Set $A = \max_{i=0,\ldots,n} |a_i|$. Then, for all $x \in \mathbb{R}$ verifying $|x| \geq A+1$, we get:

$$|P(x)| \geq |x|^n - A\,(|x|^{n-1} + \ldots + 1) = |x|^n - A\,\frac{(|x|^n - 1)}{|x| - 1}.$$

Since $|x| \geq A+1$, then we get $1 \geq A/(|x|-1)$, and thus, $|x|^n \geq A\,|x|^n/(|x|-1)$, which implies $|P(x)| \geq A/(\,|x|-1\,) > 0$. □

It is worthwhile mentioning that there exist many other bounds for the complex/real roots of a polynomial. For more details, see, e.g., [70].

The other important bound is the *root separation* of a polynomial.

Definition 2.7. The *root separation* of a polynomial $P \in \mathbb{Z}[x]$, denoted by $\mathrm{sep}(P)$, is the minimal distance between two distinct roots, i.e., it is the minimum of the absolute values of the difference of two distinct roots of P.

It constitutes an inevitable measure for the real root-finding algorithms. It is also an important measure for guaranteeing the convergence of root isolation algorithms. In fact, it allows bounding the number of interval divisions that are needed for isolating all the real roots.

Proposition 2.5. [69, 86] Let $P = \sum_{i=0}^{n} a_i\,x^i \in \mathbb{Q}[x]$ be a square-free polynomial such that $a_n = 1$ and $a_0 \neq 0$. We denote its roots by $\alpha_1, \ldots, \alpha_n$, where $\alpha_i \neq 0$ for all $0 \leq i \leq n$. Let $\mathrm{sep}(P)$ be the root separation of P. Then, we have

$$\mathrm{sep}(P) \;\geq\; \sqrt{\frac{3}{n^n+2}} \times \frac{1}{\mathrm{M}(P)^{n-1}},$$

where $\mathrm{M}(P) = |a_n| \prod_{k=1}^{n} \max(1, |a_k|)$ is called the *Mahler measure* of P.

Nevertheless, the bound for the root separation given in Proposition 2.5 is not simply computed in general. We can however compute a lower bound provided in the following corollary.

Corollary 2.3. [69, 86] Let $P = \sum_{i=0}^{n} a_i\,x^i \in \mathbb{Q}[x]$ be a square-free polynomial such that $a_n = 1$ and $a_0 \neq 0$. We denote its roots by $\alpha_1, \ldots, \alpha_n$, where $\alpha_i \neq 0$ for all $0 \leq i \leq n$. Let $\mathrm{sep}(P)$ be the root separation of P. Then, we have

$$\mathrm{sep}(P) \;\geq\; \sqrt{\frac{3}{n^n+2}} \times \frac{1}{\parallel P \parallel_2^{n-1}},$$

where $\parallel P \parallel_2 = \sqrt{\sum_{i=0}^{n} a_i^2}$.

2.6.1 Subdivision-based algorithms for root isolation

An important procedure in computer algebra is the *root isolation* of univariate polynomials. Such an algorithm takes as an input a univariate polynomial

$$P = \sum_{i=0}^{n} a_i\, x^i \;\in\; \mathbb{R}[x],\; a_n \neq 0,\; n \geq 2,$$

and returns as an output a sequence of pairwise disjoint intervals $(I_1, \ldots, I_r)$ such that r is the number of distinct real roots of P and each interval I_i, $1 \leq i \leq r$, contains exactly one real root of P. This procedure is called real *root isolation* and the intervals $I_1, \ldots, I_r$ are called *isolating intervals* for the real roots of P. The most well-known and frequently used algorithms are the *subdivision algorithms*, either based on Sturm sequences or on *Descartes' rule of signs*, differing by their ways of counting real roots in a given interval. These algorithms can fix the exponential complexity problem occurring in the original binary search algorithm. In fact, considering a univariate polynomial P in x that is square-free (if it's not the case,then we replace P by $\bar{P} = P/\gcd(P, P')$, where $P' = \frac{dP}{dx}$), *subdivision-based algorithms* start with computing a bound on the absolute value of the real roots of the polynomial, say A, then execute the bisection process after performing the change of variable $x \longmapsto A\,x$ in P. This approach allows one to start with the interval $[0,\, 1]$ containing all the real roots, and reduces the exponential theoretical complexity with respect to the polynomial degree into a polynomial complexity.

Real root counting with Sturm sequence

The *Sturm sequence* of a univariate polynomial $P \in \mathbb{R}[x]$ is defined as the sequence of polynomials $\{f_0, \ldots, f_s\}$ defined by

$$f_0 = P,\; f_1 = \frac{dP}{dx},\; f_{i+1} = -\mathtt{rem}(f_{i-1},\, f_i),\; i \geq 1,$$

where $\mathtt{rem}(P_{i-1}, P_i)$ denotes the remainder of the Euclidean division of P_{i-1} by P_i. The length of the Sturm sequence is at most $\deg(P)$ [6]. Sturm sequences have been generalised as follows.

Definition 2.8 (Sturm Sequence). [6, 86] Let $P \in \mathbb{Q}[x]$ and $[a,\, b]$ be an interval of $\mathbb{R}$. A *Sturm Sequence* of P over $[a,\, b]$ is defined as a set of univariate polynomials over $\mathbb{Q}$, denoted by $\{f_0, \ldots, f_s\}$, such that:

- $f_0 = P$.

- f_s has no real roots in $[a,\, b]$.
- For $0 < i < s$, if for $\alpha \in [a,\, b]$, $f_i(\alpha) = 0$, then $f_{i-1}(\alpha)\, f_{i+1}(\alpha) < 0$.
- If for $\alpha \in [a,\, b]$, we have $f_0(\alpha) = 0$, then

$$\begin{cases} (f_0\, f_1)(\alpha - \epsilon) < 0, \\ (f_0\, f_1)(\alpha + \epsilon) > 0, \end{cases}$$

  for all ϵ sufficiently small real number.

Proposition 2.6. [97] Let $P \in \mathbb{Q}[x]$ and $\{f_0, \ldots, f_s\}$ be its Sturm sequence over an interval $[a,\, b]$. Then, the number of real roots of P in $]a, b]$ is equal to:

$$\mathbf{V}\big(f_0(a), \ldots, f_s(a)\big) \,-\, \mathbf{V}\big(f_0(b), \ldots, f_s(b)\big).$$

The above result can be extended to unbounded intervals by defining the sign at $+\infty$ (resp., $-\infty$) of a polynomial as the sign of its leading coefficient (resp., as the sign of its leading coefficient when the polynomial is of even degree and the opposite sign of its leading coefficient when it is of odd degree).

Corollary 2.4. [6] Let $P \in \mathbb{Q}[x]$ and $\{f_0, \ldots, f_s\}$ be its Sturm sequence over $\mathbb{R}$. Then, the number of real roots of P in $\mathbb{R}$ is equal to:

$$\mathbf{V}\big(f_0(-\infty), \ldots, f_s(-\infty)\big) \,-\, \mathbf{V}\big(f_0(+\infty), \ldots, f_s(+\infty)\big).$$

Example 32. We consider the univariate polynomial

$$P = x^6 - 3\,x^5 + 7\,x^4 - 15\,x^3 + 14\,x^2 - 12\,x + 8 \in \mathbb{Z}[x]$$

studied in Examples 28 and 30 and we compute again the number of real roots of P using Sturm sequence. In this case, $f_0 = P$, $f_1 = \frac{dP}{dx}$ and

- $f_2 = -\mathtt{rem}(f_0, f_1) = -\dfrac{13}{12}\,x^4 + \dfrac{31}{6}\,x^3 - \dfrac{67}{12}\,x^2 + \dfrac{23}{3}\,x - 7,$
- $f_3 = -\mathtt{rem}(f_1, f_2) = -\dfrac{10480}{169}\,x^3 + \dfrac{12288}{169}\,x^2 - \dfrac{14464}{169}\,x + \dfrac{16896}{169},$
- $f_4 = -\mathtt{rem}(f_2, f_3) = -\dfrac{824551}{1716100}\,x^2 - \dfrac{232713}{429025}\,x + \dfrac{308087}{429025},$

- $f_5 = -\mathtt{rem}(f_3, f_4) = \dfrac{6177960000}{18203549}\,x - \dfrac{74135520000}{236646137}$,
- $f_6 = -\mathtt{rem}(f_4, f_5) = \dfrac{82369}{429025}$.

Hence, we have:

$$\begin{aligned} & \mathbf{V}\big(f_0(-\infty), \ldots, f_6(-\infty)\big) - \mathbf{V}\big(f_0(+\infty), \ldots, f_6(+\infty)\big) \\ = \; & \mathbf{V}\big(+,-,-,+,-,-,+\big) - \mathbf{V}\big(+,+,-,-,-,+,+\big) \\ = \; & 4 - 2 = 2. \end{aligned}$$

Thus, we find again that P admits only 2 real roots.

Real root counting with Descartes' rule of signs

Theorem 2.8 (Descartes' rule of signs). [6, 34] Let $P = \sum_{i=0}^{n} a_i\, x^i \in \mathbb{R}[x]$ with exactly p positive real roots counted with their multiplicities. Let $v = \mathbf{V}(P)$. Then, we have:

$$v \geq p, \quad v \equiv p \,(\mathrm{mod}\, 2).$$

If all roots of P are real, then we have $v = p$.

In particular, if $\mathbf{V}(P) = 0$ then P has no positive real root, and if $\mathbf{V}(P) = 1$, then P has exactly one positive real root.

Example 33. Considering again $P = x^6 - 3\,x^5 + 7\,x^4 - 15\,x^3 + 14\,x^2 - 12\,x + 8$ studied in Example 28, we see that $\mathbf{V}(P) = \mathbf{V}(+,-,+,-,+,-,+) = 6$, which means that P has at most 6 positive real roots and that the number of positive real roots is even. To find the number of negative roots, we change the signs of the coefficients of the terms with odd exponents, i.e., apply Descartes' rule of signs to the polynomial $P(-x)$, to obtain the following polynomial:

$$P(-x) = x^6 + 3\,x^5 + 7\,x^4 + 15\,x^3 + 14\,x^2 + 12\,x + 8.$$

In this case, $\mathbf{V}(P(-x)) = \mathbf{V}(+,+,+,+,+,+,+) = 0$. This means that $P(-x)$ has no positive real roots, and thus, P has no negative real roots.

Note that the proof of Descartes' rule of signs uses *Rolle's Theorem* in an inductive process [104].

Descartes' rule of signs, as stated above, is concerned with the number of roots in the open interval $[0, +\infty]$. The discussion can be generalized to arbitrary open intervals.

This generalization is mainly employed to bound the number of real roots of a univariate polynomial P in a given open interval [6].

On the complexity of root isolation

Theorem 2.9. [7, 105] Let $P \in \mathbb{R}[x]$, with $\text{size}(P) = (d, \tau)$, and α be a root of P. Then, we have that $\max(1, |\alpha|)$ is in the order $2^{\mathcal{O}(\tau)}$, and $\prod_{\alpha \in V(P)} \max(1, |\alpha|)$ is in the order $2^{\mathcal{O}(\tau)}$.

For a complex root z of a polynomial $P \in \mathbb{Z}[x]$, we can define the separation of z with respect to P, denoted $\text{sep}(z, P)$, to be the minimal distance between z and any distinct root z' of P. The separation of P (see Definition 2.7) can thus be defined as:

$$\text{sep}(P) = \min_{\{z \in \mathbb{C} \,|\, P(z)=0\}} \text{sep}(z, P).$$

Lemma 2.2. [91] Let $P \in \mathbb{Z}[x]$ be a square-free polynomial with $\text{size}(P) = (d, \tau)$. Then, the size of

$$\prod_{\{z \in \mathbb{C} \,|\, P(z)=0\}} \min\big(1, \text{sep}(z, P)\big)$$

is in the order of $2^{-\tilde{O}(d\,\tau)}$.

Theorem 2.10 (real roots isolation). [73] Let $P \in \mathbb{Z}[x]$ be such that $\text{size}(P) = (d, \tau)$. Then, we can compute isolating intervals of all the real roots of P and refine them up to a width less than 2^{-L} with a worst case bit complexity in:

$$\tilde{\mathcal{O}}_B(d^3 + d^2\,\tau + d\,L).$$

2.7 Solving bivariate algebraic systems

In some contexts, solving a system refers to obtaining a formal representation of its solutions that allows to concluding some useful information on these solutions. Such a representation can be, for instance, a *Gröbner basis*, a *univariate parameterization* [87, 81, 50, 1] or a *triangular representation* [4, 62, 52, 66, 26, 66, 64].

Let I be an ideal of $\mathbb{K}[x_1, \ldots, x_n]$ with a *monomial order* $\prec$ (namely, $\prec$ is a *total order* on the set of monomials $\{x^\alpha := x_1^{\alpha_1} \ldots x_n^{\alpha_n} \mid \alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n\}$, simply identified with $\mathbb{N}^n$, i.e., all the elements of $\mathbb{N}^n$ are comparable to each other, $\prec$ is *compatible with multiplication* of $\mathbb{K}[x_1, \ldots, x_n]$, i.e, if $\alpha \prec \beta$, then $\alpha + \gamma \prec \beta + \gamma$ for all $\alpha, \beta, \gamma \in \mathbb{N}^n$, and $\prec$ is a *well-ordering*, i.e, any nonempty subset of $\mathbb{N}^n$ has a smaller element for $\prec$) [54]. We

recall that a *Göbner basis* of I is a finite set of polynomials $G = \{Q_1, ..., Q_t\}$ generating I with the property that for every nonzero $P \in I$, $\mathrm{Lt}(P)$ is divisible by $\mathrm{Lt}(Q_i)$ for some i, where $\mathrm{Lt}(P)$ denotes the *leading monomial* of P with respect to $\prec$ [40].

If $\mathbb{K}$ is an algebraically closed field of characteristic 0, then information on the affine algebraic set $V_{\mathbb{K}}(I) = \{x \in \mathbb{K}^n \mid \forall\ P \in I :\ P(x) = 0\}$ can be read on a Gröbner basis of I for certain monomial orders (e.g., the dimension of $V_{\mathbb{K}}(I)$).

In some other contexts, the main focus is about the computation of numerical approximations of the solutions. These numerical approximations are usually given as a set of isolating axis-parallel boxes such that every real solution of the system lies in a unique box. *Isolating boxes* for the solutions can be computed directly from the input system using numerical approximation methods, such as *Newton's method*, or they can be obtained from a formal representation of the solutions. However, considering the theoretical complexity, it is not shown that computing a Gröbner basis or a triangular system of a system can always simplify the numerical isolation of its solutions. This observation also explains why it is not an easy task to precisely define what a formal solution of a system is. However, we can compute or represent the real solutions of a bivariate polynomial system in $\mathbb{Z}[x, y]$ through various methods. These methods are split into two categories: numerical and symbolic methods, where we note that most of the symbolic methods are able to solve only generic systems.

Numerical methods consider the polynomials as real-valued functions. They are usually based on local computation such as the famous Newton (-Raphson) method [82]. The undoubted advantage of numerical methods lies in their practical efficiency due essentially to the use of approximate computation. However, the major drawback of such methods manifests in the absence of convenient assumptions on the input system or additional algebraic computations, where these methods fail to guarantee correctness and convergence. For instance, Newton techniques cannot be applied in the case of systems with singular solutions (solutions with multiplicities). One important representative of this set of methods is the *subdivision-based methods* [74, 47, 93] that follow the same techniques as Section 2.6, up to an adaptation to isolating boxes instead of intervals. In fact, they usually start with a precise box, then split it recursively into width-smaller boxes, eliminating the ones that do not contain a zero of the system, and ending up with a union of boxes that contains all solutions of the system and surely contained in the given precise box. In the numerical methods, the verification for the existence and the uniqueness of a solution in a given box is done using numerical predicates based on interval computation techniques such as *interval evaluation*, *Newton interval method*, *Krawzcyk operator* [90], etc.

Symbolic approaches include the classical elimination-based methods such as resultant computation and isolation, *triangular decomposition*, or *rational univariate representation-based methods*. Contrary to numerical methods, these methods are complete and exact. However, a major drawback for symbolic methods lies in their high cost. In fact, as soon as the degree and/or the number of variables of a polynomial system increases, these methods become slow and impractical. Nevertheless, they still constitute a suitable choice for solving bivariate polynomial systems since many studies were driven in order to reduce as much as possible the impact of such drawback, providing improvements in the theoretical and practical complexities [17, 16, 64]. Symbolic methods, which can be viewed as generalizations of the classical *Gauss elimination method*, usually consist in the following two steps:

1. A symbolic step, called a *projection step*, computes a formal representation of the system solutions using algebraic properties and polynomial combinations.

2. A numerical step, called the *lifting step*, computes numerical approximations of the system solutions based on the knowledge of its formal representation.

Finally, many types of algebraic methods for solving bivariate polynomial systems differ by their symbolic step. The formal representation can be given as, e.g., a *resultant representation* [92, 25, 35, 13] or as a *univariate representation*.

In brief, the methods based on the resultant representation first project the solutions along several directions by only computing different resultant polynomials. This allows one to obtain a set of candidate solutions. Then, they identify the winning candidates to be the system solutions. The best known complexity for such methods is proved to be $\tilde{\mathcal{O}}_B(d^8 + d^7\tau)$ bit operations, where d and τ respectively denote the bound on the degrees and the coefficient bitsizes of the input polynomials. This complexity refers to the algorithm proposed by Berberich et al. [13] provided in [42].

Another way to obtain a symbolic representation of the system solutions is a *triangular representation* [52, 17]. In the case of bivariate polynomial system with integer coefficients, a triangular representation of the system is of the form $\big\{\big(U_i(x), V_i(x,y)\big)\big\}_{i\in\mathcal{I}}$, where the U_i's and V_i's are polynomial systems with integrer coefficients, and the sets of solutions the $\big(U_i(x), V_i(x,y)\big)$'s are disjoint and are exactly those of the polynomial system. Many complexity bounds were obtained in the case of bivariate polynomial systems with integer coefficients. The best-known complexity $\tilde{\mathcal{O}}_B(d^6 + d^5\tau)$ was recently obtained in [17], where d and τ are respectively a bound for the degrees and for coefficient bitsize of the input polynomials.

As for the *univariate representation* of the solutions, it is a one-to-one correspondence between the system solutions and the roots of a univariate polynomial. The computation of any univariate representation consists of two major steps. The first one consists in computing a separating polynomial for the solutions, namely, a polynomial taking different values when evaluated at the distinct (complex) solutions of the system. A second step consists in computing the polynomials defining the univariate representation of the system. There exist many of algorithms for computing univariate representations but the one that has the best-known worst-case bit complexity, namely, as $\tilde{\mathcal{O}}_B(d^6 + d^5\tau)$ bit operations, is represented in [17].

The main advantage of the univariate representation of the solutions is that it can easily turn many queries on the system into queries on univariate polynomials. Solving a system through this representation mainly consists in isolating the resulting univariate polynomial and then evaluating the image of the obtained isolating intervals through the maps of the rational representation. In the next section, we give more details about this approach, and apply it to one of our proposed methods for the L^∞-norm computation in Section 3.1.1.

2.7.1 Rational Univariate Representation – RUR

As explained in the above section, one important approach for solving a system of polynomials with a finite number of complex solutions is to compute a rational parametrization of its solutions [87, 88, 16, 17]. One particular type of rational parameterization called *Rational Univariate Representation*, RUR for short, was first studied in [87] and then improved in [17] for bivariate polynomial systems. Its constituting parts are mainly the computation of a *separating linear form*, a *triangular decomposition*, *rational univariate representations* and finally of *isolating boxes* of the solutions.

It has been recently proved in [17] that all of the constituting steps for solving a bivariate polynomial system $\{P, Q\} \subset \mathbb{Z}[x, y]$ are computed in a worst-case bit complexity $\tilde{\mathcal{O}}_B(d^6 + d^5\tau)$, where P and Q are of degrees bounded by d and have coefficients of bitsize bounded by τ.

We briefly give details about each step.

Separating linear form

A separating linear form of bivariate polynomial systems in $\mathbb{Z}[x, y]$ is a linear combination of the variables x and y that takes different values when evaluated at distinct complex solutions of the system. In other words, a separating linear form defines a shear of the coordinate

system $\{x,\, y\}$ that puts the algebraic system in a *generic position*, where no two distinct solutions are vertically aligned.

We mention that the computation of such linear forms is at the core of most algorithms that solve algebraic systems by computing rational parameterizations of the solutions.

Let P and Q be two coprime polynomials in $\mathbb{Z}[x, y]$ of degree bounded by d and of bitsize bounded by τ. The best known worst-case bit complexity for computing a separating linear form $x + ay$ for the bivariate system $\{P,\ Q\}$ is $\tilde{\mathcal{O}}_B(d^6 + d^5\tau)$, achieved in [17], where $a \in \{0, \ldots, 2\,d^4\}$. This separating linear form is based on the one presented in [15] but by improving its worst-case bit complexity by a factor d. This improvement is achieved by taking advantage of the fact that computing a separating linear form for a system $\{P,\, Q\}$ is essentially equivalent (in terms of asymptotic bit complexity) to computing a separating linear form for the critical points of a curve.

Triangular decomposition

For computing a triangular decomposition of a bivariate polynomial system $\{P,\, Q\} \in \mathbb{Z}[x, y]$, a classical algorithm, using subresultant sequences, was first given by L. Gonzalez-Vega and M. El Kahoui in the context of the computation of the topology of curves [52]. This result is a direct consequence of the specialization property of subresultants and of the so-called *gap structure theorem*. Hence, with the same hypotheses of Theorem 2.6, the gap structure theorem induces a decomposition of the system $\{P,\, Q\}$ into triangular subsystems $\{U_i(x), \mathrm{Sres}_i(P, Q, y)(x,\ y)\}_{i \in I}$, where the product of the U_i's is the (square-free part of the) resultant of P and Q with respect to y.

For a bivariate polynomial system of total polynomial degrees bounded by d and of coefficients of bitsize bounded by τ, the worst-case bit complexity for the computation of a triangular decomposition has been improved from $\tilde{\mathcal{O}}_B(d^{16} + d^{14}\,\tau^2)$ [52] to $\tilde{\mathcal{O}}_B(d^6 + d^5\,\tau)$ obtained recently in [17]. This improvement is due to the amortized bounds on the degrees and of the bitsizes of the resultant polynomial factors and is the best-known worst-case bit complexity for the computation of a triangular decomposition.

Rational Univariate Representation

Definition 2.9. [32] Let $\bar{\mathbb{Q}}$ denote the algebraic closure of $\mathbb{Q}$ and I an ideal of $\mathbb{Q}[x, y]$. To each zero $(\alpha,\, \beta) \in V_{\bar{\mathbb{Q}}}(I)$, we can define the *local ring* $\left(\bar{\mathbb{Q}}[x, y]/I\right)_{(\alpha,\, \beta)}$ obtained by *localizing* the ring $\bar{\mathbb{Q}}[x, y]/I$ at the *maximal ideal* $\langle x - \alpha,\, y - \beta\rangle$ of $\bar{\mathbb{Q}}[x, y]/I$, namely:

$$\left(\bar{\mathbb{Q}}[x, y]/I\right)_{(\alpha,\, \beta)} = \left\{\frac{n}{d} \mid d,\, n \in \bar{\mathbb{Q}}[x, y]/I,\, d(\alpha, \beta) \neq 0\right\}.$$

When this local ring is finite dimensional as $\mathbb{Q}$-vector space, then we say that $(\alpha,\, \beta)$ is an *isolated zero* of I and its dimension is called the *multiplicity* of $(\alpha,\, \beta)$ as a zero of I.

Definition 2.10. [87, Definition 3.3] Let $I \subset \mathbb{Q}[x, y]$ be a *zero-dimensional ideal*, namely, $\mathbb{Q}[x, y]/I$ is a finite-dimensional $\mathbb{Q}$-vector space,

$$V_{\mathbb{C}}(I) = \{\sigma \in \mathbb{C}^2 \mid \forall\, P \in I :\ P(\sigma) = 0\}$$

its associated affine algebraic set, and $(x,\, y) \longmapsto x + s\, y$ a linear form where s in $\mathbb{Q}$. The *RUR-candidate* of I associated to $x + s\, y$ (or simply, to s), denoted $\mathrm{RUR}_{I,s}$, is the following set of four univariate polynomials in $\mathbb{C}[t]$

$$\begin{aligned} f_{I,s}(t) &= \prod_{\sigma \in V_{\mathbb{C}}(I)} (t - x(\sigma) - a\, y(\sigma))^{\mu_I(\sigma)} \\ f_{I,s,v}(t) &= \sum_{\sigma \in V_{\mathbb{C}}(I)} \mu_I(\sigma)\, v(\sigma) \prod_{\{\lambda \in V_{\mathbb{C}}(I) \,|\, \lambda \neq \sigma\}} (t - x(\lambda) - s\, y(\lambda)),\ v \in \{1, x, y\} \end{aligned}$$

where $\mu_I(\sigma)$ denotes the multiplicity of σ in I for all σ in $V_{\mathbb{C}}(I)$, namely, the dimension of the local ring $\big(\mathbb{Q}[x, y]/I\big)_{\sigma}$. If $(x,\, y) \longmapsto x + s\, y$ is injective on $V_{\mathbb{C}}(I)$, then we say that the linear form $x + s\, y$ *separates* $V_{\mathbb{C}}(I)$ (or is separating for I) and $\mathrm{RUR}_{I,s}$ is called a *rational univariate representation* (RUR) (of I associated to s).

The following proposition states the fundamental properties of RURs, which are all straightforward from the definition except for the fact that the RUR polynomials have rational coefficients [87, Theorem 3.1].

Proposition 2.7. [87, Theorem 3.1]. If $I \subset \mathbb{Q}[x, y]$ is a zero-dimensional ideal and s in $\mathbb{Q}$, the four polynomials of the RUR-candidate $\mathrm{RUR}_{I,s}$ have rational coeficients. Furthermore, if $x + s\, y$ separates $V_{\mathbb{C}}(I)$, the following mapping between $V_{\mathbb{C}}(I)$ and $V_{\mathbb{C}}(f_{I,s}) = \{\gamma \in \mathbb{C} \mid f_{I,s}(\gamma) = 0\}$

$$\begin{array}{rcl} V_{\mathbb{C}}(I) & \longrightarrow & V_{\mathbb{C}}(f_{I,s}) \\ (\alpha,\, \beta) & \longmapsto & \alpha + s\, \beta, \\ \left(\dfrac{f_{I,s,x}}{f_{I,s,1}}(\gamma),\, \dfrac{f_{I,s,y}}{f_{I,s,1}}(\gamma)\right) & \longleftarrow\!\shortmid & \gamma \end{array}$$

is a bijection which preserves the real roots and the multiplicities.

In the sequel, we shall simply note f_d for $f_{I,s,1}$, f_x for $f_{I,s,x}$, f_y for $f_{I,s,y}$, and f_t for $f_{I,s}$. We now give the definition of a *RUR decomposition of an ideal* [17].

Definition 2.11. Let $I \subset \mathbb{Q}[x, y]$ be a zero-dimensional ideal, its associated affine algebraic set $V_{\mathbb{C}}(I) = \{\sigma \in \mathbb{C}^2 \mid \forall\, P \in I : P(\sigma) = 0\}$ and $(x, y) \longmapsto x + s\,y$ a linear form where s in $\mathbb{Q}$. A *RUR-candidate decomposition* of I is a sequence of RUR-candidates, associated to $x + s\,y$, of ideals $I_i \supseteq I$ such that $V_{\mathbb{C}}(I)$ is the disjoint union of the varieties $V_{\mathbb{C}}(I_i)$. If $x + s\,y$ separates $V_{\mathbb{C}}(I_i)$ for all i, then the RUR-candidate decomposition is a *R decomposition of the ideal* I.

In [17, Algorithm 6], the authors compute a RUR decomposition of a zero-dimensional bivariate system $\{P, Q\}$ by first computing a separating linear form $x + s\,y$. They use this separating form to shear the system in generic position and then compute the radical of a triangular decomposition of this system. Finally, using a multimodular approach, they compute a RUR of each of the resulting radical systems and return these RURs after a shear back. This computation is proved to be done with $\tilde{\mathcal{O}}_B(d^6 + d^5\tau)$ bit operations in the worst-case, where the degrees of P and Q are bounded by d and their coefficients are of bitsize bounded by τ [17, Proposition 42].

Isolating boxes

A RUR is naturally designed to compute isolating boxes of polynomial system solutions using univariate isolation and interval evaluation. For defining such boxes, let L be an arbitrary positive integer. Then, $\tilde{x} \in \mathbb{Q} + i\mathbb{Q}$ is defined to be an *L-bit approximation* of x if $\tilde{x}$ is of the form $\tilde{x} = (a + i\,b)\,2^{-L-2}$, where $a, b \in \mathbb{Z}$, and $|x - \tilde{x}| < 2^{-L}$. An L-bit approximation $\tilde{x} = (a + i\,b)\,2^{-L-2}$ of some point $x \in \mathbb{C}$ naturally defines the following box

$$B(\tilde{x}) = \frac{[a-4, a+4]}{2^{L+2}} + i\,\frac{[b-4, b+4]}{2^{L+2}} \subset \mathbb{C}$$

that contains x of width 2^{-L+1}.

In [16, §5.1 & Proposition 35], an algorithm of worst-case bit complexity $\tilde{\mathcal{O}}_B(d^8 + d^7\tau)$ isolates the real solutions of a system $\{P, Q\}$ of two bivariate polynomials of degrees bounded by d and of bitsize bounded by τ. In [17, §7.2], the authors presented a modified algorithm that isolates all the complex solutions. Using several amortized bounds for the roots of polynomials, they developed an algorithm that, applied to a RUR decomposition of a system $\{P, Q\}$, isolates all the complex solutions in $\tilde{\mathcal{O}}_B(d^6 + d^5\tau)$ [17, Theorem 59].

Example 34. We consider again $P = x^4 - (y+2)\,x^3 + (2\,y+1)\,x^2 - (y+2)\,x + 2\,y$ studied in Examples 29 and 31 and the system $\Sigma = \left\{P = 0,\ \frac{\partial P}{\partial x} = 0\right\}$ defining the y-critical points of the curve defined by $P(x, y) = 0$ (see Section 2.1). The rational univariate representation

of Σ is then given by:

$$\begin{cases} x = t, \\ y = \dfrac{2\,(t^2 - t + 3)}{(3\,t - 1)\,(t - 1)}, \\ f_t = (t - 2)^2\,(t^2 + 1)^2. \end{cases}$$

Hence, the real solutions $(x,\, y)$ of Σ are obtained by evaluating the rational univariate functions x and y at the real roots of the univariate polynomial f_t. We can easily notice that $t = 2$ is the only real root of f_t. Thus, the only real solution of Σ is defined by $(x,\, y) = (2,\, 2)$, and thus, the curve defined by $P(x, y) = 0$ admits only one real y-critical point of coordinates $(2,\, 2)$.

Example 35. Let $P = x^2\,y^2 - 2$, $Q = x + y + 2$ and $\Sigma = \{P = 0, Q = 0\}$. The rational univariate representation of Σ is then given by:

$$\begin{cases} x = -\dfrac{t^3 + 4\,t^2 + 4\,t + 2}{t\,(t^2 + 3\,t + 2)}, \\ y = -\dfrac{t^3 + 2\,t^2 - 2}{t\,(t^2 + 3\,t + 2)}, \\ f_t = t^4 + 4\,t^3 + 4\,t^2 - 2. \end{cases}$$

Hence, the real solutions $(x,\, y)$ of Σ are obtained by evaluating the rational univariate functions x and y at the real roots of the univariate polynomial f_t. In this case, f_t admits 2 real roots t_1 and t_2 given by isolating intervals:

$$\begin{cases} t_1 = \left[-\dfrac{6029928309892131977397}{2361183241434822606848}, -\dfrac{1205985661978426395476 7}{4722366482869645213696}\right], \\ t_2 = \left[\dfrac{311747032886141}{562949953421312}, \dfrac{1246988131544591}{2251799813685248}\right]. \end{cases}$$

The isolating boxes of the real solutions (x_1, y_1) and (x_2, y_2) of the system Σ are then given by

$$\Big[x_1 = [a_1, a_2],\; y_1 = [b_1, b_2]\Big],\; \Big[x_2 = [c_1, c_2],\; y_2 = [d_1, d_2]\Big],$$

where:

$$\begin{cases} a_1 = \dfrac{311747032886133}{562949953421312},\ a_2 = \dfrac{155873516443073}{281474976710656}, \\ b_1 = -\dfrac{359411734932195}{140737488355328},\ b_2 = -\dfrac{89852933733047}{35184372088832}, \\ c_1 = -\dfrac{179705867466097}{70368744177664},\ c_2 = -\dfrac{179705867466095}{70368744177664}, \\ d_1 = \dfrac{311747032886143}{562949953421312},\ d_2 = \dfrac{155873516443073}{281474976710656}. \end{cases}$$

2.8 Systems depending on parameters

Many problems in natural sciences and engineering sciences can be reduced to solving a parametric polynomial system of the form

$$\begin{cases} p_1(u_1, \ldots, u_d, x_1, \ldots, x_n) & = & 0, \\ \qquad\qquad \vdots & & \\ p_n(u_1, \ldots, u_d, x_1, \ldots, x_n) & = & 0, \\ f_1(u_1, \ldots, u_d, x_1, \ldots, x_n) & > & 0, \\ \qquad\qquad \vdots & & \\ f_n(u_1, \ldots, u_d, x_1, \ldots, x_n) & > & 0, \end{cases} \tag{2.3}$$

where $p_i, f_i \in \mathbb{Q}[u_1, \ldots, u_d, x_1, \ldots, x_n]$ for $i = 1, \ldots, n$, $U = \{u_1, \ldots, u_d\}$ is a set of *parameters*, and $X = \{x_1, \ldots, x_n\}$ is a set of *indeterminates*.

In this section and hereafter, we shall only consider systems that are so-called *well-behaved systems*. They are systems which contain as many equations as indeterminates, are generically zero-dimensional, i.e., for almost all complex parameter values at most finitely many complex solutions exist, and *generically radical*, i.e., for almost all complex parameter values, there are no solutions of multiplicity greater than 1 (in particular, the input equations are square-free).

In applications, questions that often arise concern the structure of the solution space in terms of the parameters, such as, e.g., determining the parameter values for which real solutions exist or, more generally, determining the parameter values for which the system have a given number of real solutions.

For naturally answering such questions, an idea would be to randomly pick real values for the parameters and then solve the corresponding non-parametric system. Such a procedure is simple and can be repeated as often as wished for. Yet, there is no guarantee that parameter values with the desired properties can be found (even if they exist). There-

fore, to solve the well-behaved system (2.3), it is significant to choose a finite number of representative "good" parameter values that cover all possible cases.

A remarkable idea was proposed by D. Lazard and F. Rouillier [63] and is based on the concept of *discriminant variety*. This method can be outlined as follows. First, the set of solutions of the system, considered as equations in both the parameters and the indeterminates, is projected onto the parameter space. Then, the topological closure $\bar{S}$ of the resulting projection, which will usually be equal to the whole parameter space $\mathbb{R}^d$, is divided into two parts: the discriminant variety W and its complement $\bar{S}\backslash W$. The discriminant variety W of the system can be understood as the set of "bad" parameter values leading to non-generic solutions of the system, for instance, infinitely many solutions, solutions at infinity, or solutions of multiplicity greater than 1. It is a generalization of the well-known discriminant of a univariate polynomial. The complement of W, $\bar{S}\backslash W$, can be expressed as a finite disjoint union of connected open sets, usually called cells, such that the number of the system real solutions does not change when the parameters vary within the same cell. For a well-behaved system, the discriminant variety is of dimension less that d, and it characterizes the boundaries between these cells. Hence, the number of solutions of the system only changes on the boundary or when crossing a boundary. For each open cell in the parameter space, we can choose a sample point, evaluate the original system at the sample point, and then solve the resulting non-parametric system. In this way, we can, for instance, determine the number of real solutions of the system that is constant for parameter values chosen from the same open cell.

Since the geometry of the connected open cells in the complement of the discriminant variety may be quite complicated, it is required to further subdivide them by the so-called *Cylindrical Algebraic Decomposition* or CAD for short.

2.8.1 Discriminant variety

Let us consider the *basic semi-algebraic set* defined by

$$\mathcal{S} = \{x \in \mathbb{R}^n \mid p_1(x) = 0, \ldots, p_s(x) = 0,\ f_1(x) > 0, \ldots, f_s(x) > 0\}$$

and the *basic constructible set* defined by

$$\mathcal{C} = \{x \in \mathbb{C}^n \mid p_1(x) = 0, \ldots, p_s(x) = 0, f_1(x) \neq 0, \ldots, f_s(x) \neq 0\},$$

where p_i, f_j are polynomials with rational coefficients for $i = 1, \ldots, s$. Moreover, let $[U, X] := [u_1, \ldots, u_d, x_{d+1}, \ldots, x_n]$ be the set of unknowns or variables, while $U = [u_1, \ldots, u_d]$ is the set of parameters and $X = [x_{d+1}, \ldots, x_n]$ the set of the indeterminate. We denote by $\Pi_U : \mathbb{C}^n \longrightarrow \mathbb{C}^d$ the canonical projection onto the parameter space $(u_1, \ldots, u_d, x_{d+1}, \ldots, x_n) \longmapsto (u_1, \ldots, u_d)$. Finally, for any set $\mathcal{V} \subset \mathbb{C}^n$, we shall denote by $\overline{\mathcal{V}}$ the $\mathbb{C}$-*Zariski closure* of $\mathcal{V}$, namely, the smallest affine algebraic set containing $\mathcal{V}$.

Definition 2.12 (Covering space, covering map)**.** [63] Given a connected open set $U \subset \mathbb{C}^s$, a *covering space* of U is a topological space C together with a continuous surjective map $\Pi : C \longrightarrow U$ such that there exist open sets $C_i \subset C$, $i \in \{1, \ldots, m\}$, satisfying:

- $\Pi^{-1}(U) = C_1 \cup \ldots \cup C_m$,
- $\Pi_{|C_i} : C_i \longrightarrow U$ is a homeomorphism,
- $C_i \cap C_j = \emptyset$ for all i, $j \in \{1, \ldots, m\}$.

Then, Π is called the *covering map* and (C, Π) is an *analytic covering* of U.

Definition 2.13. [63] Let E be a subset of the parameters space. A parametric system $\mathcal{S}$ defining a constructible set $\mathcal{C}$ is said to be *geometrically regular* over E if for all open sets $\mathcal{U} \subset E$, $(\Pi_U^{-1}(\mathcal{U}) \cap \mathcal{C},\ \Pi_U)$ is an analytic covering of $\mathcal{U}$.

We can now introduce the concept of *discriminant variety*.

Definition 2.14. [63] A *discriminant variety* of the parametric system $\mathcal{C}$ with respect to Π_U is a variety W in the parameters space such that $\mathcal{C}$ is geometrically regular over $\mathbb{C}^s \backslash W$.

Definition 2.15. [63] The minimal discriminant variety of $\mathcal{C}$ with respect to Π_U is the intersection of all the discriminant varieties of $\mathcal{C}$ with respect to Π_U.

Example 36. Consider the semi-algebraic system

$$\mathcal{C} = \big\{a\,x^2 + b - 1 = 0,\ b\,z + y = 0,\ c\,z + y = 0,\ c > 0\big\},$$

where $\{a, b, c\}$ is the set of parameters and $\{x, y, z\}$ is the set of indeterminates. Then, the discriminant variety is defined by:

$$\mathcal{D} = \big\{(a, b, c) \mid a = 0 \text{ or } b = 1 \text{ or } b = c \text{ or } c = 0\big\}.$$

In fact, the case $a = 0$ corresponds to a vanishing leading coefficient of the first equation which can be interpreted as "solution at ∞". If $b = 1$, then the first equation has a real root $x = 0$ of multiplicity 2. If $b = c$, the second and third equations coincide and therefore the system $\mathcal{C}$ becomes underdetermined and has infinitely many solutions. Finally, the case $c = 0$ corresponds to a boundary case for the inequality $c > 0$. Thus, for every choice of the parameter values outside the discriminant variety $\mathcal{D}$, the system $\mathcal{C}$ has finitely many solutions, all of multiplicity 1.

A `Maple` package for solving parametric polynomial systems was introduced in [49] under the name `Parametric`, and is mainly based on techniques such as Gröbner bases, polynomial real root finding, and cylindrical algebraic decomposition. In particular, the command `DiscriminantVariety` of this package computes the discriminant variety of a polynomial system depending on parameters and verifying the required conditions.

2.8.2 Cylindrical Algebraic Decomposition

In this paragraph, we briefly explain the concept of an *open Cylindrical Algebraic Decomposition* or a *generic CAD* [37]. It is a well-known concept in computational real algebraic geometry, first proposed by G. E. Collins [27]. It has been followed by many substantial improvements including adjacency and clustering techniques [3], improved projection methods [59, 72, 23, 21], partially built CADs [29], improved stack construction [30], efficient projection orders [36], etc.

A CAD of $\mathbb{R}^d$ is a representation of $\mathbb{R}^d$ as a finite union of disjoint cells in a "cylindrical fashion" such that the canonical projection onto the first $d-1$ coordinates is a CAD of $\mathbb{R}^{d-1}$. A CAD of $\mathbb{R} = \mathbb{R}^1$ is just a representation of $\mathbb{R}$ as a finite disjoint union of points and open intervals.

Finally, for a finite set F of polynomials in d variables, a CAD of $\mathbb{R}^d$ is said to be *F-invariant* if each of the polynomials in F has a constant sign on each cell of the decomposition.

Let W be the discriminant variety of the parametric system $\mathcal{C}$ with respect to Π_U. A CAD allows us to obtain a practical description of the connected components of $\Pi_U(\mathcal{S}) \setminus (W \cap \mathbb{R}^d)$. This CAD is mainly a cylindrical decomposition of $\mathbb{R}^d$ associated to a family of polynomials $\mathcal{P}$ of $\mathbb{R}[x_{d+1}, \ldots, x_n]$. This cylindrical decomposition decomposes $\mathbb{R}^d$ into cells over which all of the polynomials of $\mathcal{P}$ are of a constant sign. These cells can be defined by a sample point and a *semi-algebraic function*, namely, a function $f : T_1 \to T_2$, where $T_1 \subset \mathbf{R}^k$ and $T_1 \subset \mathbf{R}^\ell$ are two semi-algebraic sets (see Section 2.8.1), such that its

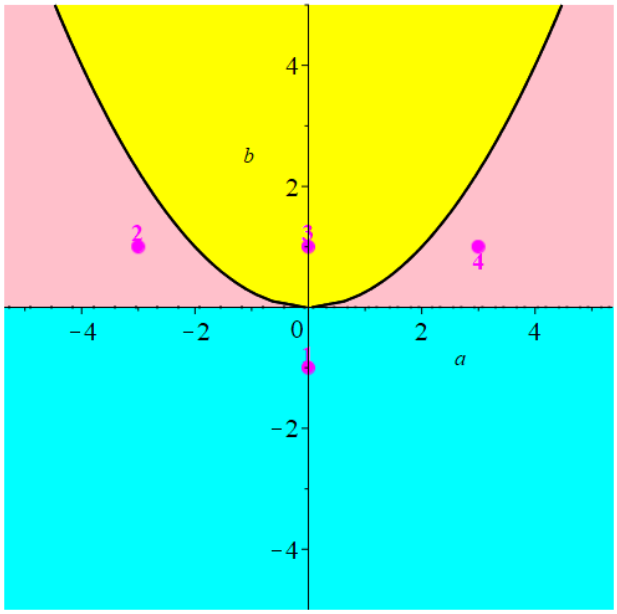

Figure 2.3: Plot of the 4 regions of the parameter space for which the univariate equation $P = x^2 + a\,x + b$ has exactly two solutions in the pink (regions 2 and 4) and cyan (region 1) cells below the parabola, and no solution in the yellow cell (region 3) above the parabola.

graph Graph(f) is a semi-algebraic subset of $\mathbf{R}^{k+\ell}$ [6].

Example 37. We consider the univariate polynomial $P = x^2 + a\,x + b$ whose coefficients depend on the parameters set $\{a, b\}$. The discriminant variety is thus the curve $a^2 - 4\,b = 0$. Using CAD, the parameter space can then be decomposed into 4 cells (see Figure 2.3) in which P has a constant number of solutions that can be computed using a sample point from each cell.

Example 38. In this example, we present a cylindrical algebraic decomposition of $\mathbb{R}^3$ adapted to the unit sphere, illustrated in Figure 2.4. Note that the projection of the sphere on the $(x,\, y)$ plane is the unit disk. The intersection of the sphere and the cylinder above the open unit disk consists of two hemispheres. The intersection of the sphere and the cylinder above the unit circle consists of a circle. The intersection of the sphere and the cylinder above the complement of the unit disk is empty. Note also that the projection of the unit circle on the line is the interval $[-1,\, 1]$.

The decomposition of $\mathbb{R}$ consists of five cells, called cells of level 1, corresponding to

the points -1 and 1 and the three intervals they define, namely:

$$\begin{cases} S_1 =]-\infty,\, -1[, \\ S_2 = \{-1\}, \\ S_3 =]-1,\, 1[, \\ S_4 = \{1\}, \\ S_5 =]1,\, +\infty[. \end{cases}$$

Above S_i for $i = 1, 5$, there are no semi-algebraic functions and only one cell $S_{i,1} = S_i \times \mathbb{R}$.

Above S_i, for $i = 2, 4$, there is only one semi-algebraic function associating to -1 and 1, the constant value 0, and there are three cells:

$$\begin{cases} S_{i,1} = S_i \times]-\infty,\, 0[, \\ S_{i,2} = S_i \times \{0\}, \\ S_{i,3} = S_i \times]0,\, +\infty[. \end{cases}$$

Above S_3, there are two semi-algebraic functions $f_{3,1}$ and $f_{3,2}$ associating to $x \in S_3$ defined by $f_{3,1}(x) = -\sqrt{1-x^2}$ and $f_{3,2}(x) = \sqrt{1-x^2}$. There are 5 cells above S_3, the graphs of $f_{3,1}$ and $f_{3,2}$ and the bands they define, namely:

$$\begin{cases} S_1 = \big\{(x,y) \mid -1 < x < 1,\, y < f_{3,1}(x)\big\}, \\ S_2 = \big\{(x,y) \mid -1 < x < 1,\, y = f_{3,1}(x)\big\}, \\ S_3 = \big\{(x,y) \mid -1 < x < 1,\, f_{3,1}(x) < y < f_{3,2}(x)\big\}, \\ S_4 = \big\{(x,y) \mid -1 < x < 1,\, y = f_{3,2}(x)\big\}, \\ S_5 = \big\{(x,y) \mid -1 < x < 1,\, f_{3,2}(x) < y\big\}. \end{cases}$$

Above $S_{i,j}$ for $(i,\, j) \in \big\{(1,\, 1),\, (2,\, 1),\, (2,\, 3),\, (3,\, 1),\, (3,\, 5),\, (4,\, 1),\, (4, 3),\, (5, 1)\big\}$, there are no semi-algebraic functions, and only one cell:

$$S_{i,j,1} = S_{i,j} \times \mathbb{R}.$$

Above $S_{i,j}$ for $(i,\, j) \in \big\{(2,\, 2),\, (3,\, 2),\, (3,\, 4),\, (4,\, 2)\big\}$, there is only one semi-algebraic function, the constant function 0, and three cells:

$$\begin{cases} S_{i,j,1} = S_{i,j} \times]-\infty,\, 0[, \\ S_{i,j,2} = S_{i,j} \times \{0\}, \\ S_{i,j,3} = S_{i,j} \times]0,\, +\infty[. \end{cases}$$

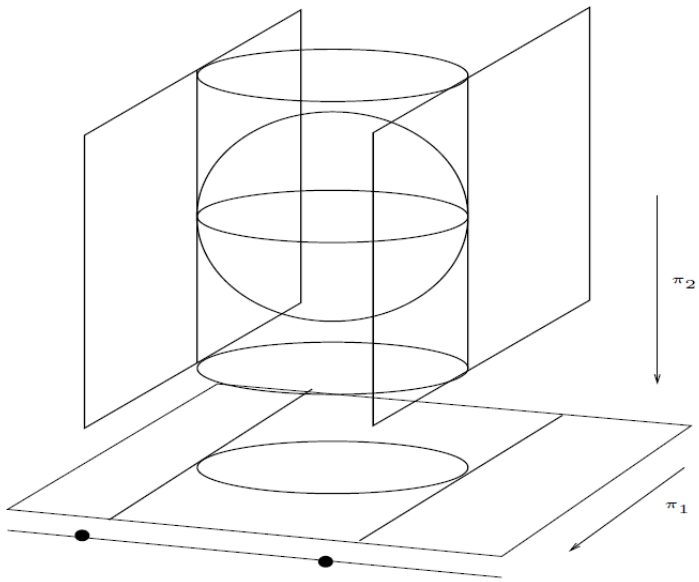

Figure 2.4: A cylindrical decomposition adapted to the sphere in $\mathbb{R}^3$

Above $S_{3,3}$, there are two semi-algebraic functions $f_{3,3,1}$ and $f_{3,3,2}$ associating to $(x,\, y) \in S_{3,3}$ defined by

$$f_{3,3,1}(x,y) = -\sqrt{1-x^2-y^2},$$
$$f_{3,3,2}(x,y) = \sqrt{1-x^2-y^2},$$

and five cells defined by:

$$\begin{cases} S_{3,3,1} = \big\{(x,\, y,\, z) \,|\, (x,\, y) \in S_{3,3},\, z < f_{3,3,1}(x,y)\big\} \\ S_{3,3,2} = \big\{(x,\, y,\, z) \,|\, (x,\, y) \in S_{3,3},\, z = f_{3,3,1}(x,y)\big\}, \\ S_{3,3,3} = \big\{(x,\, y,\, z) \,|\, (x,\, y) \in S_{3,3},\, f_{3,3,1}(x,y) < z < f_{3,3,2}(x,y)\big\}, \\ S_{3,3,4} = \big\{(x,\, y,\, z) \,|\, (x,y) \in S_{3,3},\, z = f_{3,3,2}(x,y)\big\}, \\ S_{3,3,5} = \big\{(x,\, y,\, z) \,|\, (x,y) \in S_{3,3},\, f_{3,3,2}(x,y) < z\big\}. \end{cases}$$

If we want to characterize the cells of $\Pi_U(\mathcal{S}) \setminus (W \cap \mathbb{R}^d)$ obtained with a CAD, a naive method is proposed in [86, § 5.6.7].

In the case of well-behaved systems, we have $\overline{\Pi_U(\mathcal{C})} = \mathbb{C}^d$. In this case, the useful cells of a CAD are those of a higher dimension. In Collins' algorithm [27] or any of its variants, the construction of cells of lower dimension, such as those represented by a sample point and a semi-algebraic function, requires the resolution of univariate polynomial equations with algebraic coefficients. This is not the case for the cells of higher dimension where all the polynomials we are dealing with are of rational coefficients (mainly, the polynomials

figuring in the projection and lifting phases explained below).

The algorithm of a CAD is then decomposed into two steps: A projection phase followed by a recovery phase. In the kth step of the projection, we suppose that we have a set $\mathcal{P}_k \subset \mathbb{Q}[u_k, \ldots, u_d]$. The $(k+1)^{\text{th}}$ step of projection consists in constructing $\mathcal{P}_{k+1} = \text{Proj}(\mathcal{P}_k)$ which is the smallest subset of $\mathbb{Q}[u_{k+1}, \ldots, u_d]$ such that:

- For $p \in \mathcal{P}_k$, $\deg_{u_k}(p) = d \geq 2$, $\text{Proj}(P_k)$ contains $\text{discrim}(p)$ (p seen as univariate in u_k),
- For p, $q \in \mathcal{P}_k$, $\text{Proj}(P_k)$ contains $\text{Res}(p, q, u_k)$,
- For $p \in \mathcal{P}_k$ such that $\deg_{u_k}(p) \geq 1$ and $\text{Lc}_{u_k}(p)$ is non constant, then $\text{Proj}(P_k)$ contains $\text{Lc}_{u_k}(p)$,
- For $p \in \mathcal{P}_k$ such that $\deg_{u_k}(p) = 0$ and p is non constant, then $\text{Proj}(P_k)$ contains p.

The recovery step in the CAD is also recursive starting with $\mathcal{P}_k = \mathcal{P}_d$:

- Isolate and sort all the real roots of the polynomials in $\mathcal{P}_k$, consisting in univariate polynomials in u_k with rational coefficients.
- Choose a point in every interval between two real roots of polynomials in the set $\mathcal{P}_k$,
- Substitute u_k with the chosen point in the previous step in the set $\mathcal{P}_{k-1}, \ldots, \mathcal{P}_1$ and then recover $\mathcal{P}_{k-1}$ with the same lifting steps.

For more details about the Cylindrical Algebraic Decomposition, see [6] and the references therein.

Chapter 3

L^∞-norm Computation

Based on the standard computer algebra concepts and methods − stated again in Chapter 2 − in this chapter, we shall explain our proposed methods for the computation of the L^∞-norm of LTI systems (see Chapter 1) based on the problem modeling given in Section 1.4. We shall consider two different cases. The first case (the non-parametric case) is when the LTI system does not depend on parameters, i.e., when all the entries of a state-space representation (A, B, C, D) of the control system or, equivalently, when all the coefficients of the rational function entries of its transfer matrix G, are fixed to numerical values, supposed thereafter to be rational. As explained in Section 1.4, the problem of L^∞-norm computation is then reduced to the computation of the maximal x-projection of the real solutions (x, y) of a bivariate polynomial equations system $\Sigma = \left\{P = 0, \frac{\partial P}{\partial x} = 0\right\}$ for a certain polynomial $P \in \mathbb{Z}[x, y]$. The second case (the parametric case) is when the control system depends on a set of parameters $\alpha = (\alpha_1, \ldots, \alpha_d) \in \mathbb{R}^d$ (e.g., an unfixed mass, spring constant or damping coefficient), in which case we have to study the polynomial system $\Sigma = \left\{P = 0, \frac{\partial P}{\partial x} = 0\right\}$ for a certain polynomial $P \in \mathbb{Z}[\alpha][x, y]$. In this case, for computing the L^∞-norm of the corresponding transfer matrix G, the parameters space has to be partitioned into cells in such a way that, above each, we can represent the maximal x-projections of the real solutions (x, y) of Σ as a real function of the parameters α.

3.1 Non-parametric case problem

Let $F \in RL_\infty^{p\times m}$ and $\Phi_\gamma(i\,\omega) = \gamma^2\, I_m - F^T(-i\,\omega)\, F(i\,\omega)$ for $\omega \in \mathbb{R}$. Let $n(\omega, \gamma)$ and $d(\omega)$ be two coprime polynomials over $\mathbb{R}[\omega, \gamma]$, i.e., $\gcd(n, d) = 1$, satisfying:

$$\det(\Phi_\gamma(i\,\omega)) = \frac{n(\omega, \gamma)}{d(\omega)}. \tag{3.1}$$

Let us also note $\bar{n}$ the square-free part of n. Then, according to the results developed in Chapter 1 (see Proposition 1.1 and the results after), the problem of computing $\| F \|_\infty$ can be reduced to the computation of the maximal γ-projection of the real solutions of the system of bivariate polynomial equations:

$$\Sigma = \left\{ \bar{n}(\omega, \gamma) = 0, \frac{\partial \bar{n}(\omega, \gamma)}{\partial \omega} = 0 \right\}. \tag{3.2}$$

Without loss of generality, we shall suppose that n is square-free in $\mathbb{Z}[\omega, \gamma]$ and, in the sequel, we simply denote $\bar{n}$ by n.

In the next sections, using standard computer algebra methods (see Chapter 2), three different *symbolic-numeric algorithms* for the computation of the maximal γ-projection of the real solutions of Σ will be proposed. We alternatively study a method based on *rational univariate representation*, a method based on *root separation*, and finally, a method based on *real root counting*. The latter is based on the sign variation of the leading coefficients of the signed subresultant sequence. These three algorithms identify an isolating interval of the maximal γ-projection of the real solutions of Σ. These three different methods then give us three different methods to compute the L^∞-norm of $F \in RL_\infty^{p \times m}$. The worst case bit complexity of each algorithm is analyzed and their theoretical complexities are finally compared to their practical complexities.

3.1.1 RUR method

In this section, we state a straightforward algorithm which computes the maximal γ-projection of the real solutions of (3.2) based on a *Rational Univariate Representation method* defined in Subsection 2.7.1. This algorithm consists in first computing a rational parametrization (RUR) of the solutions of (3.2), then isolating the roots of a univariate polynomial p defining the associated field extension, and finally using the intervals obtained to compute isolating boxes for the solutions of (3.2). After performing interval refinements, we can then select the solution of (3.2) with the maximal γ-projection.

We recall that if P, $Q \in \mathbb{Q}[x, y]$ are two coprime polynomials, i.e., $\gcd(P, Q) = 1$, then the computation of the RUR of $V_{\mathbb{K}}(\langle P, Q \rangle)$, where $\mathbb{K} = \mathbb{R}$, $\mathbb{C}$, consists in finding $s \in \mathbb{N}$ such that $t := x + s\,y$ separates the $\mathbb{K}$-zeros of $\{P, Q\}$ and four polynomials f_d, f_x, f_y, $f_t \in \mathbb{Q}[T]$

which define the following 1-1 correspondence between $V_{\mathbb{K}}(\langle P,\, Q\rangle)$ and $V_{\mathbb{K}}(\langle f_t\rangle)$:

$$\begin{array}{rcl} V_{\mathbb{K}}(\langle P,\, Q\rangle) & \longrightarrow & V_{\mathbb{K}}(\langle f_t\rangle) \\ (x,\, y) & \longmapsto & t = x + s\, y, \\ .\left(\dfrac{f_x(t)}{f_d(t)},\, \dfrac{f_y(t)}{f_d(t)}\right) & \longleftarrow\!\shortmid & t \end{array}$$

Using the RUR of $V_{\mathbb{K}}(\langle P,\, Q\rangle)$, we can transform the study of problems on $V_{\mathbb{K}}(\langle P,\, Q\rangle)$ into corresponding problems on $V_{\mathbb{K}}(\langle f_t\rangle)$. See Subsection 2.7.1.

Given two coprime polynomials $P,\, Q \in \mathbb{Z}[x, y]$ of degree bounded by d and coefficient bitsize bounded by τ, we recall that an algorithm for computing a linear separating form, a RUR decomposition and isolating boxes of the system solutions can be obtained in the worst case bit complexity $\tilde{\mathcal{O}}_B(d^6 + d^5\,\tau)$ (see Subsection 2.7.1). The solutions of the polynomial system $\{P = 0,\, Q = 0\}$ are then represented by isolating boxes.

For the L^∞-norm computation, the polynomials in $\mathbb{Z}[\omega, \gamma]$ defining (3.2) are coprime. Hence, to compute $\parallel F \parallel_\infty$, we first use the RUR method to obtain isolating boxes for the real solutions $(\omega,\, \gamma)$ of Σ, choose the maximal γ-projection γ_1, then compute an isolating interval γ_2 for the maximal real root of the univariate polynomial $\mathrm{Lc}_\omega(n)$, and finally get $\parallel F \parallel_\infty = \max\{\gamma_1, \gamma_2\}$. We sum up the different steps of the corresponding algorithm in Algorithm 2.

Algorithm 2 RUR method

Input: A zero-dimensional polynomial system $\{n,\, \frac{\partial n}{\partial \omega}\} \subset \mathbb{Z}[\omega,\, \gamma]$.
Output: An isolating interval of $\max\left\{\pi_\gamma\left(V_{\mathbb{R}}\left(\langle n,\, \frac{\partial n}{\partial \omega}\rangle\right)\right) \cup V_{\mathbb{R}}\left(\mathrm{Lc}_\omega(n)\right)\right\}$.

1. Apply the RUR function (`Isolate`) for solving $\Sigma = \{n = 0, \frac{\partial n}{\partial \omega} = 0\}$ and denote $[a_i,\, b_i] \times [c_i,\, d_i]$ the isolating boxes of the obtained real solutions.
2. For $\omega \in [a_i,\, b_i] > 0$, if $\exists\, i,\, j$ such that $[c_i,\, d_i] \cap [c_j,\, d_j] \neq \emptyset$,
   - compute the root separation of $R_\gamma := \mathrm{Res}\left(n, \frac{\partial n}{\partial \omega}, \omega\right)$ (using for instance Corollary 2.3), and denote it L,
   - isolate the RUR polynomials and compute the isolating boxes of the solutions of Σ up to the precision L,
3. let γ_1 be the maximal γ-projection of the real solutions of Σ and γ_2 be the maximal real root of $\mathrm{Lc}_\omega(n)$.
4. **return** the isolating interval of $\max\{\gamma_1, \gamma_2\}$.

In Step 1 of Algorithm 2, we obtain isolating boxes $[a_i,\, b_i] \times [c_i,\, d_i]$ of the real solutions $(\omega_i,\, \gamma_i)$ of Σ. To compare the real values $\gamma \in [c_i\,,\, d_i]$, we need to make sure that when two intervals $[c_i,\, d_i]$ and $[c_j,\, d_j]$ intersect, they both contain only 1, and the same, real γ-projection of the system's real solutions. For doing so, in Step 2, we apply a straightforward strategy consisting in computing isolating boxes in a way that each interval $[c_i,\, d_i]$ is included in, or intersects at most, one isolating interval of the roots of R_γ, since R_γ is a polynomial embodying the γ-projection of the system's solutions. We mention that we can simply look only at the solutions of positive ω-projection since Σ is symmetric with respect to the γ-axis. In the next paragraph, we further discuss this important step of Algorithm 2.

For a zero-dimensional polynomial system $\{P = 0,\, Q = 0\}$, in [17, Section 7], Bouzidi et al. provide a fine algorithm for computing disjoint isolating boxes for the solutions $\sigma \in \mathbb{C}^2$. In this algorithm for the RUR decomposition of $\{P, Q\}$, a RUR decomposition RUR_i of an ideal $\tilde{T}_i$ – coming from the triangular decomposition $\{T_i\}_{i\in\mathcal{I}}$ of the ideal $\langle P, Q\rangle$ – verifying certain conditions, is used. Then, for a given L, they first compute L-bit approximations $\tilde{\sigma}_{i,j} = (\tilde{x}_{i,j},\, \tilde{y}_{i,j})$ of the solutions $\sigma_{i,j} = (x_{i,j},\, y_{i,j})$, $1 \le j \le d_i = \deg(f_{i,t})$, of each factor $\mathrm{RUR}_i = (f_{i,t}, f_{i,d}, f_{i,x}, f_{i,y})$ in the RUR decomposition $(\mathrm{RUR}_i)_{i\in\mathcal{I}}$ of $\{P,\, Q\}$. The computation of isolating boxes of the system $\{P = 0,\, Q = 0\}$ solutions is achieved by first computing sufficiently small isolating disks for the roots $\alpha_{i,j}$ of the univariate polynomial $f_{i,t}$ in RUR_i and then evaluating the fractions $\frac{f_{i,x}}{f_{i,d}}$ and $\frac{f_{i,y}}{f_{i,d}}$ at the roots $\alpha_{i,j}$ to an absolute error less than 2^{-L}. From the corresponding L-bit approximations $\tilde{x}_{i,j}$ and $\tilde{y}_{i,j}$, they derive boxes (as defined in Paragraph 2.7.1) $B_{i,j} = B(\tilde{\sigma}_{i,j}) = B(\tilde{x}_{i,j}) \times B(\tilde{y}_{i,j}) \subset \mathbb{C}^2$ of width 2^{-L+1} containing all the solutions of RUR_i. If for all i and j, the boxes $B_{i,j}$ do not overlap, then they are already isolating for the solutions of $\{P = 0,\, Q = 0\}$. Otherwise, L must be increased until the boxes do not overlap.

We recall that $R_x = \mathrm{Res}(P, Q, y)$ (resp., $R_y = \mathrm{Res}(P, Q, x)$) embodies the x-projection (resp., y-projection) of the system $\{P = 0,\, Q = 0\}$ solutions. Thus, having non-overlapping isolating boxes of the system solutions means that for all $\tilde{\sigma}_k = (\tilde{x}_k,\, \tilde{y}_k)$, corresponding to L-bit approximations of the system solutions $\sigma_k = (x_k,\, y_k)$, either $B(\tilde{x}_{k_1})$ and $B(\tilde{x}_{k_2})$ or $B(\tilde{y}_{k_1})$ and $B(\tilde{y}_{k_2})$ do not overlap.

Thus, to further guarantee that each $B(\tilde{x}_k)$ contains only one x-projection of the system $\{P = 0,\, Q = 0\}$ solutions, we consider $L = \mathrm{sep}(R_x)$ where $\mathrm{sep}(R_x)$ denotes the separation bound of R_x.

Lemma 3.1. [17, Lemma 57] Let $g = \dfrac{g_1}{g_2}$, with $g_1,\, g_2 \in \mathbb{Z}[x]$ polynomials of degree at most $d_g = \mathcal{O}(d)$ with coefficients of bitsize at most τ_g. Suppose that g_2 does not vanish at any of the roots $\alpha_1, \ldots, \alpha_d$ of f where $\mathrm{size}(f) = (d\,,\, \tau)$. Then, for any given positive integer L, we

can compute L-bit approximations of all values $y_i = g(\alpha_i)$ using a number of bit operations bounded by

$$\tilde{\mathcal{O}}_B(d^3 + d^2\,(\tau + \tau_g) + d\,L).$$

For the complexity analysis of the computation of the L^∞-norm of the matrix $F \in RL_\infty^{p\times m}$, we first need to obtain bounds on the degrees the polynomial n, defining Σ (see (3.2)), as well as its coefficient bitsize in terms of F.

Lemma 3.2. Let $F \in RL_\infty^{p\times m}$, where $F_{i,j} := \dfrac{P_{i,j}}{Q_{i,j}}$, $P_{i,j}$, $Q_{i,j} \in \mathbb{Z}[i\,\omega]$ are coprime polynomials for $1 \le i \le m$ and $1 \le j \le p$, and let τ_F be the maximal coefficient bitsize of $P_{i,j}$ and $Q_{i,j}$. Moreover, let $n \in \mathbb{Z}[\omega,\,\gamma]$ be the numerator of $\det(\Phi_\gamma(\omega))$ where $\Phi_\gamma(\omega) = \gamma^2\,I_m - F^T(-i\,\omega)\,F(i\,\omega)$ and $d_\gamma := \deg_\gamma(n) = 2\,m$, $d_\omega := \deg_\omega(n)$, and τ_n the coefficients bitsize of n. If we note

$$\alpha := \max\big\{p,\,m\big\}, \quad N := \max\big\{\,\deg_\omega(Q_{i,j}),\ 1 \le i \le m,\ 1 \le j \le p\big\},$$

then we have:

$$d_\gamma = \mathcal{O}(\alpha), \quad d_\omega = \mathcal{O}(\alpha^2\,N), \quad \tau_n = \tilde{\mathcal{O}}(\alpha^2\,\tau_F).$$

Proof. We have:

$$F(i\,\omega) = \left(\frac{P_{i,j}(i\,\omega)}{Q_{i,j}(i\,\omega)}\right)_{i,j} = (a_{i,j})_{i,j}, \quad F^T(-i\,\omega) = \left(\frac{P_{j,i}(-\,i\,\omega)}{Q_{j,i}(-\,i\,\omega)}\right)_{i,j} = (b_{i,j})_{i,j}.$$

Then, we get:

$$F^T(-i\,\omega)\,F(i\,\omega) = \left(\sum_{k=1}^{p} b_{i,k}\,a_{k,j}\right)_{i,j} = \left(\sum_{k=1}^{p} \frac{P_{k,i}(-i\,\omega)\,P_{k,j}(i\,\omega)}{Q_{k,i}(-i\,\omega)Q_{k,j}(i\,\omega)}\right)_{i,j} = (c_{i,j})_{i,j}.$$

Since F is a proper matrix, then $\max\big(\deg_\omega(P_{i,j})\big) \le N = \max\big(\deg_\omega(Q_{i,j})\big)$. Thus, $\deg_\omega\big(P_{k,i}(-i\,\omega)\,P_{k,j}(i\,\omega)\big) \le \deg_\omega\big(Q_{k,i}(-i\,\omega)\,Q_{k,j}(i\,\omega)\big) \le 2\,N$. Now, we denote the matrix $\Phi_\gamma(\omega) := \gamma^2\,I_m - F^T(-i\,\omega)\,F(i\,\omega)$ by $(A_{i,j})_{i,j}$, where:

$$A_{i,i} = \gamma^2 - \sum_{k=1}^{p} \frac{P_{k,i}(-i\,\omega)\,P_{k,i}(i\,\omega)}{Q_{k,i}(-i\,\omega)Q_{k,i}(i\,\omega)}, \quad A_{i,j} = -\sum_{k=1}^{p} \frac{P_{k,i}(-i\,\omega)\,P_{k,j}(i\,\omega)}{Q_{k,i}(-i\,\omega)Q_{k,j}(i\,\omega)}.$$

Thus, by the Leibniz formula for the determinant, we can write

$$\det\big(\Phi_\gamma(\omega)\big) = \prod_{i=1}^{m} A_{i,i} + \sum_{\sigma\in S_v\setminus\mathrm{Id}} \epsilon(\sigma) \prod_{i=1}^{m} A_{i,\sigma(i)},$$

where S_v denotes the set of permutations σ of the set $\{1, 2, \ldots, m\}$ and ϵ is the signature of a permutation. Then, it is clear that degree of the numerator $n(\gamma, \omega)$ of $\det\big(\Phi_\gamma(\omega)\big)$ is in the order of $\mathcal{O}(2\,m) = \mathcal{O}(\alpha)$ with respect to the variable γ.

To compute the degree of $n(\gamma, \omega)$ with respect to ω, we first compute the denominator of

$$\prod_{i=1}^{m} A_{i,\sigma(i)} = \prod_{i=1}^{m} \sum_{k=1}^{p} \frac{P_{k,i}(-i\,\omega)\, P_{k,\sigma(i)}(i\,\omega)}{Q_{k,i}(-i\,\omega) Q_{k,\sigma(i)}(i\,\omega)},$$

for $\sigma \in S_v$. In fact, by multiplying all denominators, we obtain the denominator $\prod_{i=1}^{m} \prod_{k=1}^{p} Q_{k,i}(-i\,\omega)\, Q_{k,\sigma(i)}(i\,\omega)$ that can be simply written as $\prod_{k=1}^{p} \prod_{m=1}^{m} Q_{k,m}(-i\,\omega)\, Q_{k,m}(i\,\omega)$ by changing of index, since σ is a permutation over the index i. Hence, $n(\gamma, \omega)$ can be written as

$$n(\gamma, \omega) = \prod_{k=1}^{m} \prod_{m=1}^{p} Q_{k,m}(-i\,\omega)\, Q_{k,m}(i\,\omega)\, \gamma^{2\,m} + \sum_{i=0}^{m-1} C_{2\,i}\, \gamma^{2\,i},$$

such that $C_{2\,i} \in \mathbb{Z}[\omega^2]$, and $\deg_\omega(C_{2\,i}) < 2\,m\,p\,N$ for $i \in \{0, \ldots, m-1\}$. Hence, we can conclude that the degree of $n(\gamma, \omega)$ in ω is $2\,m\,p\,N = \mathcal{O}(\alpha^2\,N)$ and its bitsize is $\tau_n = \tilde{\mathcal{O}}(m\,p\,\tau_F) = \tilde{\mathcal{O}}(\alpha^2\,\tau_F)$. □

Theorem 3.1. With the notations of Lemma 3.2, the complexity of Algorithm 2 for the computation of $\| F \|_\infty$, where $F \in RL_\infty^{p \times m}$, is given by:

$$\tilde{\mathcal{O}}_B\big(d_\gamma\, d_\omega^3\, (d_\gamma^2 + d_\gamma\, d_\omega + d_\omega\, \tau_n)\big) = \tilde{\mathcal{O}}_B\big(\alpha^9\, N^4\, (\alpha + \tau_n)\big).$$

Proof. According to [17] and as explained in Subsection 2.7.1, the complexity of the resolution of a zero-dimensional bivariate polynomial system using the RUR method comes from:

1. The computation of the triangular decomposition of the system obtained after shearing the original system using a separating linear form $t = \gamma + s\,\omega$.

2. The root isolation of the univariate polynomial in t defining the associated field extension.

3. The computation of isolating boxes for the solutions of the system.

In the present case, the degrees in ω and γ are not of the same order (see Lemma 3.2). Hence, the results of [17] must be adapted.

First, we determine the size and the degree of the sheared system up to the method used in [17] with respect to the separating linear form $t = \gamma + s\,\omega$: with the notations of Lemma 3.2, the degree with respect to the variable ω of the sheared system is $\tilde{\mathcal{O}}(d_\gamma + d_\omega)$ and d_γ with respect to the variable $t = \gamma + s\,\omega$. Moreover, the size of the sheared system is $\tilde{\mathcal{O}}(\tau_n + d_\gamma)$.

Then, from [64], the complexity of the computation of a triangular decomposition of a polynomial system $\{P = 0,\, Q = 0\}$ over $\mathbb{Z}[x, y]$, where

$$d_x := \max\big(\deg_x(P),\, \deg_x(Q)\big),\; d_y := \max\big(\deg_y(P),\, \deg_y(Q)\big),$$

and the polynomials are of coefficient bitsize bounded by $\tilde{\tau}$, costs:

$$\tilde{\mathcal{O}}_B\big(d_x^3\, d_y^3 + (d_x^3\, d_y^2 + d_x^4\, d_y)\, \tilde{\tau}\big).$$

The triangular decomposition of the sheared system can be computed by considering $P = n(\omega, t)$ and $Q = \frac{\partial n}{\partial \omega}(\omega, t)$. Thus, the complexity of the computation of the triangular decomposition of the sheared system in $\mathbb{Z}[t, \omega]$ is given by:

$$\begin{aligned}&\tilde{\mathcal{O}}_B\big((d_\gamma + d_\omega)^3\, d_\gamma^3 + ((d_\gamma + d_\omega)^3\, d_\gamma^2 + (d_\gamma + d_\omega)^4\, d_\gamma)\, (\tau_n + d_\gamma)\big)\\ &= \tilde{\mathcal{O}}_B\big((d_\gamma + d_\omega)^3\, d_\gamma\, \big(d_\gamma^2 + d_\gamma\, d_\omega + d_\omega\, \tau_n + d_\gamma\, \tau_n)\big)\,\big).\end{aligned}$$

Now, using Lemma 3.2, we have $d_\gamma = \mathcal{O}(\alpha)$, $d_\omega = \mathcal{O}(\alpha^2\, N)$, which shows that we can assume that d_ω is larger than d_γ. Hence, we obtain $\tilde{\mathcal{O}}_B\big((d_\gamma + d_\omega)^3\big) = \tilde{\mathcal{O}}_B\big(d_\omega^3\big)$ and $\tilde{\mathcal{O}}_B\big(d_\omega\, \tau_n + d_\gamma\, \tau_n\big) = \tilde{\mathcal{O}}_B\big(d_\omega\, \tau_n\big)$, which shows that:

$$\tilde{\mathcal{O}}_B\big((d_\gamma + d_\omega)^3\, d_\gamma\, \big(d_\gamma^2 + d_\gamma\, d_\omega + d_\omega\, \tau_n + d_\gamma\, \tau_n)\big) = \tilde{\mathcal{O}}_B\big(d_\omega^3\, d_\gamma\, (d_\gamma^2 + d_\gamma\, d_\omega + d_\omega\, \tau_n)\big).$$

Moreover, using again $d_\gamma = \mathcal{O}(\alpha)$ and $d_\omega = \mathcal{O}(\alpha^2\, N)$ by Lemma 3.2, we obtain:

$$\begin{aligned}\tilde{\mathcal{O}}_B\big(d_\omega^3\, d_\gamma\, (d_\gamma^2 + d_\gamma\, d_\omega + d_\omega\, \tau_n)\big) &= \tilde{\mathcal{O}}_B\big(\alpha^7\, N^3\big(\alpha^2 + \alpha^3\, N + \alpha^2\, N\, \tau_n\big)\big)\\ &= \tilde{\mathcal{O}}_B\big(\alpha^9\, N^4\, (\alpha + \tau_n)\big).\end{aligned}$$

Now, using [17], the RUR decomposition corresponding to this triangular decomposition yields the RUR polynomials $(f_t,\, f_d,\, f_x,\, f_y)$ of degrees

$$\mathcal{O}\big((d_\gamma + d_\omega)\, d_\gamma\big) = \mathcal{O}(\alpha^3\, N),$$

with coefficients of bitsize $\tilde{\mathcal{O}}\big((d_\gamma+d_\omega)\,(d_\gamma+\tau_n)\big) = \tilde{\mathcal{O}}\big(\alpha^2\,N\,(\alpha+\tau_n)\big)$. Then the computation of isolating intervals of the roots of f_t can be done in

$$\tilde{\mathcal{O}}_B\big((\alpha^3\,N)^3 + (\alpha^3\,N)^2\,(\alpha^2\,N\,(\alpha+\tau_n))\big) = \tilde{\mathcal{O}}_B\big(\alpha^8\,N^3\,(\alpha+\tau_n)\big)$$

bit operations, using Theorem 2.10.

Now, let $R_\gamma = \mathrm{Res}\left(n, \frac{\partial n}{\partial \omega}, \omega\right)$ where $\mathrm{size}(R_\gamma) = (d_\gamma\,d_\omega\,,\,d_\omega\,\tau_n)$ using Lemma 2.3. Thus, $\mathrm{sep}(R_\gamma) = \tilde{\mathcal{O}}(d_\gamma\,d_\omega^2\,\tau_n)$ using Lemma 2.2 and denote $L = \mathrm{sep}(R_\gamma)$. Hence, according to Lemma 3.1 and Lemma 3.2, we can compute L-bit approximations of $\dfrac{f_x}{f_d}$ and $\dfrac{f_y}{f_d}$ at the real roots of f_t with a worst case bit complexity

$$\begin{aligned}&\tilde{\mathcal{O}}_B\big((\alpha^3\,N)^3 + (\alpha^3\,N)^2\,(\alpha^2\,N\,(\alpha+\tau_n)) \;+\; (\alpha^3\,N)\,(\alpha^5\,N^2)\,\tau_n\big)\\ &= \tilde{\mathcal{O}}_B\big(\alpha^8\,N^3\,(\alpha+\tau_n)\big).\end{aligned}$$

Then, based on the discussion of the previous paragraph and up to a permutation of the roles of x and y, we obtain isolating boxes of the form $[a_i,\,b_i]\times[c_i,\,d_i]$ of length in the order $\mathcal{O}(L)$ and thus each interval $[c_i,\,d_i]$ contains only 1 real γ-projection of the real solutions of the system since R_γ encodes the γ-projection of the system's solution. Thus, we can compare the real values $\gamma_i \in [c_i,\,d_i]$.

Finally, the overall worst case bit complexity of Algorithm 2 is then:

$$\tilde{\mathcal{O}}_B\big(d_\gamma\,d_\omega^3\,(d_\gamma^2 + d_\gamma\,d_\omega + d_\omega\,\tau_n)\big) = \tilde{\mathcal{O}}_B\big(\alpha^9\,N^4\,(\alpha+\tau_n)\big).$$

□

We illustrate Algorithm 2 with two examples using the `Maple` commands for the computation of the RUR decomposition and root isolation. It is worthwhile to mention that the RUR decomposition used in these examples is not the same as the RUR decomposition used for the complexity analysis which is not implemented in `Maple` yet but is under construction. The one existing already in `Maple` is implemented in `C` for general zero dimensional systems and is mainly based on Gröbner basis computation.

Example 39. We consider the following transfer matrix:

$$G = \begin{pmatrix} \dfrac{1}{s+1} & \dfrac{1}{s+1} \\ 0 & \dfrac{1}{s+1} \end{pmatrix} \in RH_\infty^{2\times 2}.$$

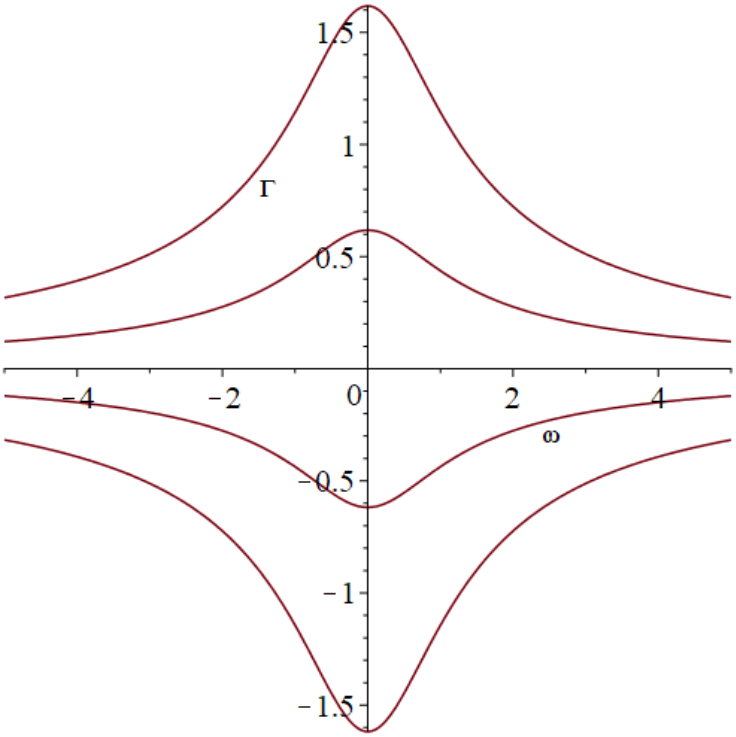

Figure 3.1: Plot of $n(\omega,\,\gamma)=0$, where ω/γ is in the horizontal/vertical axis.

Let $\Phi_\gamma(s)=\gamma^2\,I_2-G^T(-s)\,G(s)$ and $\det(\Phi_\gamma(i\,\omega))=\frac{n(\omega,\gamma)}{d(\omega)}$. We have to study the real solutions of $\Sigma=\left\{n=0,\frac{\partial n}{\partial\omega}=0\right\}$, where the polynomial n is defined by

$$n(\omega,\,\gamma)=\gamma^4\,\omega^4+\gamma^2\,(2\,\gamma^2-3)\,\omega^2+(\gamma^2+\gamma-1)(\gamma^2-\gamma-1)$$

and the corresponding curve $\mathcal{C}$ is shown in Figure 3.1.

We first have that $V_{\mathbb{R}}(\mathrm{Lc}_\omega(n))=\{0\}$. Then, applying the RUR method to $\Sigma=\left\{n(\omega,\gamma)=0,\frac{\partial n(\omega,\gamma)}{\partial\omega}=0\right\}$, we obtain the following RUR of Σ:

$$\begin{cases} p=(t^2+t-1)(t^2-t-1),\\ \gamma=\dfrac{3\,t^2-2}{t\,(2\,t^2-3)},\\ \omega=0. \end{cases}$$

The real solutions $(\omega,\,\gamma)$ of Σ are then defined by:

$$\left(0,\,-\frac{\sqrt{5}-1}{2}\right),\ \left(0,\,-\frac{\sqrt{5}+1}{2}\right),\ \left(0,\,\frac{\sqrt{5}-1}{2}\right),\ \left(0,\,\frac{\sqrt{5}+1}{2}\right).$$

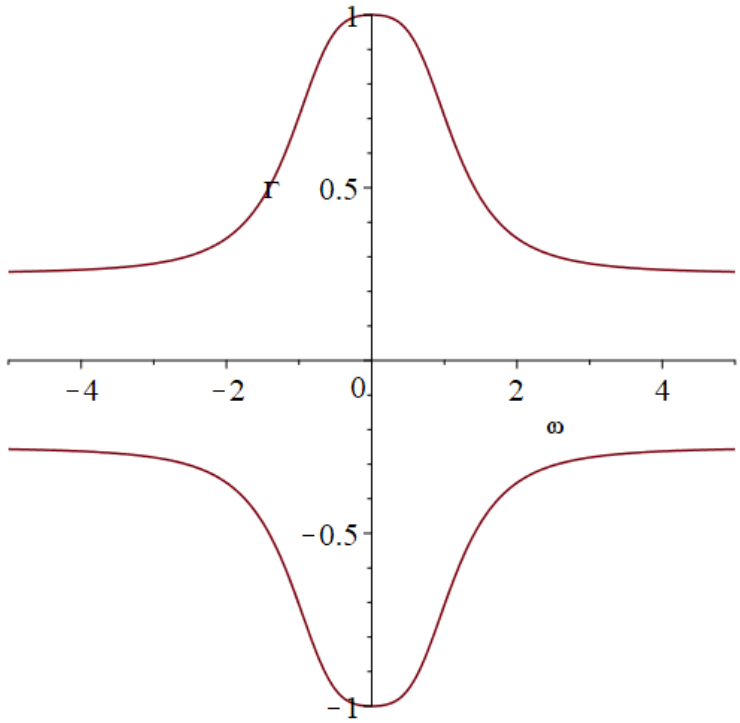

Figure 3.2: Plot of $n(\omega,\,\gamma)=0$, where ω/γ is in the horizontal/vertical axis.

We can then pick their maximal γ-projection to obtain $\frac{\sqrt{5}+1}{2}$, which yields:

$$\parallel G \parallel_\infty = \max\left\{0, \frac{\sqrt{5}+1}{2}\right\} = \frac{\sqrt{5}+1}{2}.$$

Example 40. We consider the transfer function defined in Example 46 with the particular numerical values $\omega_0 = 2$, $\omega_1 = 1$, and $\xi = 3/4$, i.e.:

$$G = \frac{s^2+3\,s+4}{2\left(2\,s^2+3\,s+2\right)}.$$

We first have $G(i\,\infty) = 1/4$. Doing the computations, we then obtain

$$n(\omega,\,\gamma) = (4\,\gamma-1)(4\,\gamma+1)\,\omega^4 + (2\,\gamma-1)(2\,\gamma+1)\,\omega^2 + 16\,(\gamma-1)(\gamma+1),$$

where the curve of $\mathcal{C}$ is shown in Figure 3.2.

We can check again that $G(i\,\infty) = 1/4$ is the maximal real root of $\mathrm{Lc}_\omega(n)$. By applying

the RUR method to $\Sigma = \left\{ n(\omega,\gamma) = 0, \frac{\partial n(\omega,\gamma)}{\partial \omega} = 0 \right\}$, we obtain the following RUR of Σ

$$\begin{cases} p = (t-1)(t+1)\ p_t, \\ \gamma = \dfrac{39424\,t^8 + 2874688\,t^6 + 54625488\,t^4 - 118479684\,t^2 + 69632373}{t\,(62720\,t^8 + 3856384\,t^6 + 51759648\,t^4 - 106855392\,t^2 + 59868929)}, \\ \omega = \dfrac{-64\,(t-1)\,(t+1)\,(15680\,t^6 + 599760\,t^4 - 1051140\,t^2 + 839423)}{t\,(62720\,t^8 + 3856384\,t^6 + 51759648\,t^4 - 106855392\,t^2 + 59868929)}. \end{cases}$$

where $p_t = 12544\ t^8 + 976640\ t^6 + 18229856\ t^4 - 35197840\ t^2 + 24671089$.

The real roots of p are then $\{t = -1,\ t = 1\}$. Thus, the real solutions $(\omega,\ \gamma)$ of Σ are defined by:

$$\Big(0,\ -1\Big), \Big(0,\ 1\Big).$$

Picking the maximal γ-projection and comparing it with $|G(i\,\infty)|$, we finally get:

$$\parallel G \parallel_\infty = \max\left\{\frac{1}{4},\ 1\right\} = 1.$$

3.1.2 Root separation method

In this section, we localize the maximal γ-projection of the real solutions of the system $\Sigma = \left\{ n(\omega,\gamma) = 0, \frac{\partial n(\omega,\gamma)}{\partial \omega} = 0 \right\}$ by only shearing the system Σ using a special linear separating used in [25]. Using this special linear separating form $t = \gamma + s\,\omega$, for two solutions $(\omega_1,\ \gamma_1)$ and $(\omega_2,\ \gamma_2)$ of Σ, we then have:

$$t_1 = \gamma_1 + s\,\omega_1 \ < \ t_2 = \gamma_2 + s\,\omega_2 \ \Longrightarrow \ \gamma_1 \ \leq \ \gamma_2. \tag{3.3}$$

The problem of computing the maximal γ-projection of the real solutions of Σ is then reduced to the computation of the maximal real solution of a univariate polynomial in t.

Let $P, Q \in \mathbb{Z}[x,y]$ be two coprime polynomials, i.e., $\gcd(P,Q) = 1$, and $R_y = \mathrm{Res}(P,Q,x) \in \mathbb{Z}[y]$ their resultant with respect to x. Let $y_1 \leq \ldots \leq y_m$ be the real roots of R_y with their isolating intervals $[c_1,\ d_1], \ldots, [c_m,\ d_m]$. Moreover, let us define the real numbers δ, M and s as follows:

$$\begin{cases} \delta < \frac{1}{2}\min_{i=1,\ldots,m-1}(y_{i+1} - y_i), \\ M > \max\{x \mid (x,\ y) \in V_{\mathbb{R}}(\langle P,\ Q\rangle)\}, \\ 0 < s < \dfrac{\delta}{M}. \end{cases} \tag{3.4}$$

Remark 3.1. Using Proposition 2.4, M can be taken as $M = 1 + \max|a_i|$, where the a_i's are the coefficients of the univariate polynomial $R_x = \mathrm{Res}(P, Q, y)$ since R_x embodies the x-projection of the solutions of $\{P = 0,\, Q = 0\}$. $2\,\delta$ can be chosen as the root separation bound of R_y (defined in Proposition 2.5 and Corollary 2.3).

We can use general root bounds for zero-dimensional systems to estimate δ and M. See, e.g., [105].

Note that δ can simply be considered to be equal to:

$$\frac{1}{2} \min_{i=1,\ldots,m-1} \{c_{i+1} - d_i\}\,.$$

For M, note that the computation of the resultant R_x can be avoided by using the concept of *sleeve functions* studied in [26] and [25, Lemma 3.3].

With the notations (3.4), let us consider an invertible linear map (a *shear*) defined by:

$$\begin{array}{rccl} \Psi_s : & \mathbb{R}^2 & \longrightarrow & \mathbb{R}^2 \\ & (x,\, y) & \longmapsto & (x,\, t) = (x,\, y + s\,x). \end{array}$$

Let us also note

$$\Psi_s(P) = P(x, t - s\,x),\ \ \Psi_s(Q) = Q(x, t - s\,x),\ \ R_t = \mathrm{Res}(\Psi_s(P), \Psi_s(Q), x),$$

and let $t_1 \leq \ldots \leq t_{m'} = t_{\max}$ be the real roots of R_t.

If $(x_\star,\, y_\star)$ is a solution of $\{P(x, y) = 0,\, Q(x, y) = 0\}$, then $(x_\star,\, y_\star + s\,x_\star)$ is a solution of $\{\Psi_s(P)(x, t) = 0,\, \Psi_s(Q)(x, t) = 0\}$. See Figure 3.3.

To get a one-to-one correspondence between the zeros of $\{P,\, Q\}$ and the roots of R_t, $\mathrm{Lc}_x(\Psi_s(P))$ and $\mathrm{Lc}_x(\Psi_s(Q))$ must not both vanish (since the values where both $\mathrm{Lc}_x(\Psi_s(P))$ and $\mathrm{Lc}_x(\Psi_s(Q))$ vanish represent the common horizontal asymptotes of the curves defined by $\Psi_s(P)$ and $\Psi_s(Q)$). It is always possible to choose s such that this condition is satisfied as shown in [25]. In what follows, we shall consider that s always satisfies this condition.

Proposition 3.1. With the above notations, let y_m be a real root of R_y with an isolating interval $[c_m,\, d_m]$ and $t_{\max}$ the maximal real root of R_t. If $t_{\max} \in [c_m - \delta,\, d_m + \delta]$, then the maximal y-projection of $V_{\mathbb{R}}(\langle P,\, Q\rangle)$ is equal to y_m.

Proof. For each real root y_i of R_y with an isolating interval $[c_i,\, d_i]$, let us denote by $P_{i,j} = (x_{i,j},\, y_i)$ the real solutions of $\{P = 0,\, Q = 0\}$. Then, we have

$$\Psi_s(P_{i,j}) = (x_{i,j},\, y_i + s\,x_{i,j}),$$

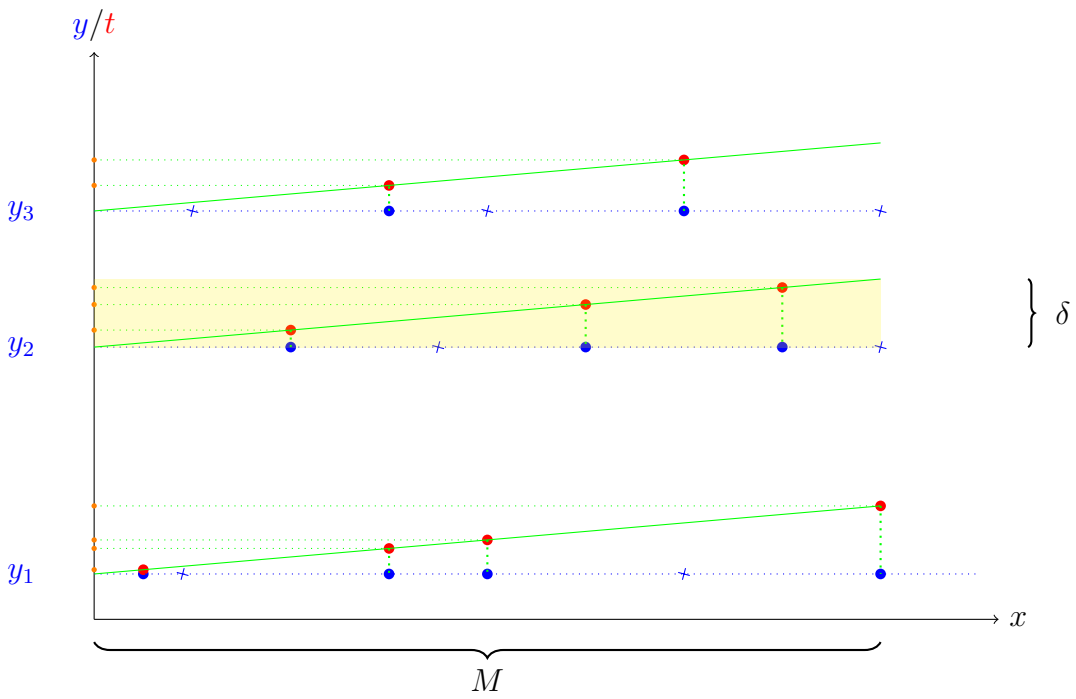

Figure 3.3: The blue dots represent the real solutions of $\Sigma = \{P = 0, Q = 0\}$ and the blue crosses are the complex ones; the red dots are the solutions of the system $\Psi_s(\Sigma) := \{\Psi_s(P) = 0, \Psi_s(Q) = 0\}$; the orange dots on the vertical axis are the roots of the univariate polynomial R_t.

where $y_i + s\,x_{i,j}$ is the second coordinate of a real solution of the polynomial system $\{\Psi_s(P) = 0, \Psi_s(Q) = 0\}$. Note that $y_i + s\,x_{i,j} = y_i + s\,x_{i,k}$ yields $x_{i,j} = x_{i,k}$. Hence, for a fixed i, the t-projection of the $\Psi_s(P_{ij})$'s are different. Furthermore, we have $|y_i + s\,x_{i,j} - y_i| = |s\,x_{i,j}| < \dfrac{\delta}{M}\,M = \delta$, which yields:

$$\Psi_s(P_{i,j}) \in I_i := [-M,\, M] \times [y_i - \delta,\, y_i + \delta].$$

Hence, for a fixed i, the t-projection of the $\Psi_s(P_{ij})$'s belong to the same interval I_i. In addition, since $\delta < \frac{1}{2}(y_{i+1} - y_i)$, the I_i's are disjoint for different i. Thus, the linear form $(x,\, y) \mapsto (x,\, y + s\,x)$ is separating and verifies Property (3.3). Hence, the system $\{\Psi_s(P) = 0,\, \Psi_s(Q) = 0\}$ is in a *generic position* in the sense that no two solutions are horizontally aligned.

Moreover, a real solution $(x,\, y)$ of $\{P = 0,\, Q = 0\}$ is mapped by Ψ_s to $(x,\, \eta)$, where $\eta \in [y - \delta,\, y + \delta]$. Finally, since $y_i \in [c_i,\, d_i]$, the real roots of R_t associated with y_i are in the interval $[c_i - \delta,\, d_i + \delta]$ and since the above separating form is a one-to-one mapping between the real solutions of Σ and the real roots of R_t, if $t_{\max} \in [c_m - \delta,\, d_m + \delta]$, then the maximal y-projection of the real roots of Σ has then the isolating interval $[c_m,\, d_m]$. □

Algorithm 3 Root separation method

Input: A zero-dimensional polynomial system $\{P,\, Q\} \subset \mathbb{Z}[x, y]$, where $Q = \frac{\partial P}{\partial x}$.
Output: An isolating interval of $\max\left\{\pi_y\big(V_{\mathbb{R}}\big(\langle P, Q\rangle\big)\big) \cup V_{\mathbb{R}}\big(\langle \mathrm{Lc}_x(P)\rangle\big)\right\}$

1. Isolate $R_y = \mathrm{Res}(P, Q, x)$ and let $\big\{[c_1,\, d_1], \ldots, [c_m,\, d_m]\big\}$ be the isolating intervals of the real roots $\big\{y_1, \ldots, y_m\big\}$ of R_y.
2. Compute M, $\delta = \frac{1}{4}\min_{i=1,\ldots,m-1} |c_{i+1} - d_i|$ and s up to the required conditions.
3. Expand $\big\{\Psi_s(P), \Psi_s(Q)\big\}$ and compute $R_t = \mathrm{Res}\left(\Psi_s(P), \Psi_s(Q), x\right)$.
4. Isolate R_t up to an accuracy less than δ and set $[p_t,\, q_t]$ to be the isolating interval of its maximal real root $t_{\max}$.
5. For j from 1 to m do:
   - if $[p_t,\, q_t] \subset [c_j - \delta,\, d_j + \delta]$, then $Y_1 = y_j$.
6. Let Y_2 be the maximal real root of $\mathrm{Lc}_x(P)$.
7. **Return** the isolating interval of $\max\big\{Y_1, Y_2\big\}$.

Lemma 3.3. Let $P \in \mathbb{Z}[x, y]$, $d_x = \deg_x(P)$, $d_y = \deg_y(P)$ and τ be the maximal coefficient bitsize of P. The sheared polynomial $P(x,\, t - s\,x)$ then satisfies

$$\deg_x\big(P(x,\, t - s\,x)\big) = d_x + d_y,\ \deg_t\big(P(x,\, t - s\,x)\big) = d_y,$$

and it can be expanded in $\tilde{\mathcal{O}}_B\big(d_y\, d_x^2\, (\tau + d_y\, (1 + \tau_s)\big)$ bit operations. The maximal bitsize of the coefficients of $P(x,\, t - s\,x)$ is equal to $\tilde{\mathcal{O}}\big(\tau + d_y\, (1 + \tau_s)\big)$, where τ_s denotes the bitsize of s.

Proof. The proof is a direct consequence of [16, Lemma 7] by taking into account the bitsize τ_s of s. Let us write $P = \sum_{i=0}^{d_x} a_i\, x^i$ where $a_i = \sum_{j=0}^{d_y} b_{ij}\, y^j$. The expansion of the substitution of y by $t - s\,x$ in P needs the computation of the successive powers $(t - s\,x)^j$ for $j \in \{1, \ldots, d_y\}$. The binomial formula

$$(t - s\,x)^{d_y} \;=\; \sum_{j=0}^{d_y} \binom{d_y}{j} (s\,x)^{d_y - j}\, t^j$$

first yields:

$$\deg_x\big(P(x, t - s\,x)\big) \;=\; d_x \;+\; d_y, \quad \deg_t\big(P(x, t - s\,x)\big) \;=\; d_y.$$

It also shows that each polynomial $(t - s\,x)^j$ is the sum of $j + 1$ monomials with coefficients of bitsize in $\mathcal{O}(j\ \log j + d_y\, \tau_s)$. Using the recursion formula $(t - s\,x)^j = (t - s\,x)^{j-1}\, (t - s\,x)$, given the polynomial $(t - s\,x)^{j-1}$, the computation of $(t - s\,x)^j$ requires $2\,j$ multiplications of coefficients having coefficient bitsize in $\mathcal{O}(j\ \log j + d_y\, \tau_s)$, which can be done in $\tilde{\mathcal{O}}_B(j^2\ \log j + j\, d_y\, \tau_s)$ bit operations. The worst case bit complexity for the computation of all the powers $(t - s\,x)^j$'s is then in:

$$\tilde{\mathcal{O}}_B\big(d_y^3\, (\log d_y + \tau_s)\big).$$

The second step is to multiply x^i by $(t - s\,x)^{j_i}$ for $i \in \{1, \ldots, d_x\}$. Each polynomial multiplication can be done with $\mathcal{O}(d_x\, d_y)$ multiplications of integers of bitsize in $\mathcal{O}(\tau)$ or $\mathcal{O}(j \log j + d_y\, \tau_s)$. Thus, this operation can be done in $\tilde{\mathcal{O}}_B\big(d_x\, d_y\big(\tau + d_y\, (1 + \tau_s)\big)\big)$ bit operations and yields polynomials of coefficients bitsize $\tilde{\mathcal{O}}\big(\tau + d_y(1 + \tau_s)\big)$. After operating d_x multiplications, the overall worst case bit complexity is then in:

$$\tilde{\mathcal{O}}_B(d_x^2\, d_y\, (\tau + d_y\, (1 + \tau_s))).$$

□

Theorem 3.2. Let us consider a zero-dimensional system $\{P, Q\} \subset \mathbb{Z}[x, y]$, where $d_x = \max(\deg_x(P), \deg_x(Q))$, $d_y = \max(\deg_y(P), \deg_y(Q))$ and τ the maximal coefficient bitsize of P and Q. Then, using Algorithm 3, we can compute an isolating interval for the maximal y-projection of the real solutions of the polynomial system $\{P = 0, Q = 0\}$ in $\tilde{\mathcal{O}}_B\big(d_x^3\, d_y^4\, \tau\, \big(d_x^2 + d_x\, d_y + d_y^2\big)\big)$ bit operations.

Proof. Based on Proposition 3.1, Algorithm 3 outputs an isolating interval for the maximal y-projection of the real solutions of $\{P, Q\}$.

As for the complexity computation, note $d = \deg(R_y) = d_y\, d_x$. Using Proposition 2.3, let $\tilde{\tau} = \tilde{\mathcal{O}}(d_x\, \tau)$ be the coefficients bitsize of R_y. Then, according to Theorem 2.10, Step 1 of Algorithm 3 is of worst case bit complexity:

$$\tilde{\mathcal{O}}_B(d^3 + d^2\, \tilde{\tau}) = \tilde{\mathcal{O}}_B(d_y^2\, d_x^3\, (d_y + \tau)).$$

In Step 2, using Lemma 2.2, we get $\delta = 2^{-\tilde{\mathcal{O}}(d_y\, d_x^2\, \tau)}$. Moreover, using Theorem 2.9 for $R_x = \mathrm{Res}(P, Q, y)$ of size$(R_y) = (d_x\, d_y, d_y\, \tau)$ (by Proposition 2.3), using Remark 3.1, we then obtain that $M = 2^{\mathcal{O}(d_y\, \tau)}$. Hence, the bitsize of s is then equal to $\tilde{\mathcal{O}}(d_y\, d_x^2\, \tau)$. Consequently, as shown in Lemma 3.3, $\deg_x(\Psi_s(P)) = d_x + d_y$, $\deg_t(\Psi_s(Q)) = d_y$ and the maximal bitsize of the sheared system is $\tilde{\mathcal{O}}(d_y^2\, d_x^2\, \tau)$ and the worst case bit complexity of Step 3 is then $\tilde{\mathcal{O}}_B(d_y\, (d_y + d_x)^3\, d_y^2\, d_x^2\, \tau) = \tilde{\mathcal{O}}_B(d_y^3\, d_x^2\, (d_y + d_x)^3\, \tau)$ by Proposition 2.3.

In step 4, we isolate the resultant R_t of the sheared system. Considering the size and the degree of the sheared polynomials given above, the size and degree of the resultant of the sheared system are $\tilde{\mathcal{O}}(d_y^2\, d_x^3\, \tau)$ and $\tilde{\mathcal{O}}(d_y\, (d_x + d_y))$ respectively. Then, knowing the complexity of the isolation from Theorem 2.10, we can say that the worst case bit complexity of Step 4 is equal to:

$$\tilde{\mathcal{O}}_B\Big(\big(d_y\, (d_x + d_y)\big)^3 + \big(d_y\, (d_x + d_y)\big)^2\, \big(d_y^2\, d_x^3\, \tau\big)\Big) = \tilde{\mathcal{O}}_B\Big(d_x^3\, d_y^4\, \tau\, \big(d_x^2 + d_x\, d_y + d_y^2\big)\Big).$$

Finally, in Step 5, we simply compare two rational numbers. The maximal coefficient bitsize of these rationals is in $\tilde{\mathcal{O}}(d_y^3\, d_x^3\, (d_x + d_y)\, \tau)$ and the computation in this step is done in $\tilde{\mathcal{O}}_B(d_x^3\, d_y^3\, (d_x + d_y)\, \tau)$ bit operations. Hence, the overall bit complexity of Algorithm 3 is given by $\tilde{\mathcal{O}}_B\big(d_x^3\, d_y^4\, \tau\, \big(d_x^2 + d_x\, d_y + d_y^2\big)\big)$. □

Corollary 3.1. With the notations of Lemma 3.2 and considering $d_x = d_\omega$ and $d_y = d_\gamma$, the worst case bit complexity for the computation of $\| F \|_\infty$ with the root separation method

(Algorithm 3) applied on $\Sigma = \left\{ n(\omega,\gamma) = 0, \frac{\partial n(\omega,\gamma)}{\partial \omega} = 0 \right\}$ is given by:

$$\tilde{\mathcal{O}}_B\big(d_\omega^5\, d_\gamma^4\, \tau_n\big) \;=\; \tilde{\mathcal{O}}_B\big(\alpha^{14}\, N^5\, \tau_n\big).$$

Example 41. We again consider the transfer matrix defined in Example 39 and we follow the root separation method that consists in directly focusing on the maximal γ-projection of the real solutions of the system. In this case, we first compute $\mathrm{Res}\left(n, \frac{\partial n}{\partial \omega}, \omega\right)$ and denote by $R = \gamma\,(\gamma^2+\gamma-1)\,(\gamma^2-\gamma-1) \in \mathbb{Z}[\gamma]$ its square-free part. Then, the maximal real root of R has the following isolating interval:

$$[a,\, b] = \left[\frac{56929509912547}{35184372088832}, \frac{113859019825121}{70368744177664}\right].$$

Following the root separation method, we obtain:

$$\begin{cases} s = \dfrac{12060328540887}{281474976710656}, \\ \delta = \dfrac{43490275647441}{140737488355328}. \end{cases}$$

We have $R_t = \mathrm{Res}\left(\Psi_s(n), \Psi_s(\frac{\partial n}{\partial \omega}), \omega\right) = \alpha\,(t^2+t-1)(t^2-t-1)$, where α is a rational of size around 3000 bits. We denote by $t_{\max}$ the maximal real root of R_t. An isolating interval of $t_{\max}$ is then given by:

$$[c,\, d] = \left[\frac{113859019825095}{70368744177664}, \frac{56929509912561}{35184372088832}\right].$$

In this case, $[c,\, d] \subset [a-\delta,\, b+\delta]$, which, after comparing $[a,\, b]$ with the isolating interval of the maximal real root of $\mathrm{Lc}_\omega(n)$, shows that $\| G \|_\infty$ is equal to the maximal real root of R has the isolating interval $[a,\, b]$.

Example 42. We consider again the transfer matrix defined in Example 40. Similarly to the previous example, we start by computing $\mathrm{Res}(n, \frac{\partial n}{\partial \omega}, \omega)$ and then its square-free part:

$$R = (\gamma-1)\,(4\,\gamma+1)\,(4\,\gamma-1)\,(\gamma+1)\,(112\,\gamma^4 - 120\,\gamma^2 + 7) \in \mathbb{Z}[\gamma].$$

R has 8 distinct real roots. But, since $n \in \mathbb{Z}[\omega^2, \gamma^2]$, $\Sigma = \{n = 0, \frac{\partial n}{\partial \omega} = 0\}$ is symmetric with respect to the γ-axis and the ω-axis. Hence, to compute an isolating interval for the maximal γ-projection of the real solutions of Σ, it suffices to look at the set of the isolating intervals of the 4 largest (positive) real roots of R. This latter set, denoted by L_R, is ordered

as follows

$$L_R = \{\, [a_1,\, b_1],\, [a_2,\, b_2],\, [a_3,\, b_3],\, [a_4,\, b_4] \,\}\,,$$

where $[a_4,\, b_4]$ refers to the isolating interval of the maximal real root of R.

The maximal real root of R has the following isolating interval:

$$[a_4,\, b_4] = \left[\frac{593098324447812476929}{590295810358705651712}, \frac{1186196648895625602425}{1180591620717411303424}\right].$$

Moreover, by following the root separation method, we obtain

$$\begin{cases} s = \dfrac{16356461765629375240721289194171 75}{4980610507814138795424615819973140283520749775070691328}, \\ \delta = \dfrac{85121814626263}{144115188075855872}, \end{cases}$$

and

$$R_t = \mathrm{Res}\left(\Psi_s(n), \Psi_s\left(\frac{\partial n}{\partial \omega}\right), \omega\right) = (t-1)(t+1)(\alpha_1\, t^8 + \alpha_2\, t^6 + \alpha_3\, t^4 + \alpha_4\, t^2 + \alpha_5),$$

where α_i are integers of size bounded by 1500 bits. We denote by $t_{\max}$ the maximal real root of R_t. An isolating interval of $t_{\max}$ is then given by:

$$[c,\, d] \;=\; [1,\, 1]\,.$$

In this case, $[c,\, d] \not\subset [a_4 - \delta,\, b_4 + \delta]$, which shows that $\| G \|_\infty$ is not equal to the maximal real root of R of isolating interval $[a_4,\, b_4]$. Hence, we pass to $[a_3,\, b_3] = [1,\, 1]$. For this isolating interval, it is clear that $[c,\, d] \subset [a_3 - \delta,\, b_3 + \delta]$, which, after comparing $\gamma = 1$ with $\gamma = 1/4$, shows that $\| G \|_\infty$ is equal to the real root of R with the isolating interval $[a_3,\, b_3]$, that is $\gamma = 1 = \| G \|_\infty$.

From the results obtained in this section, we can conclude that trying to only concentrate on the solution with the maximal γ-projection, after putting the system in a generic position, costs much more than simply computing isolating boxes for all the real solutions (see Section 3.1.1), due to the large size of the separating bound that must be used. Hence, in the next section, using a different strategy than shearing the system, we shall try to find the maximal γ-projection of the solutions of Σ without computing isolating boxes for all of its real solutions.

3.1.3 Sturm-Habicht method

In this section, as in Section 3.1.2, we shall concentrate only on the maximal γ-projection $\bar{\gamma}$ of the real solutions of the following polynomial system:

$$\Sigma = \left\{ n(\omega, \gamma) = 0, \frac{\partial n(\omega, \gamma)}{\partial \omega} = 0 \right\}.$$

But instead of shearing the system Σ, we shall verify the existence of a real root of Σ over $\bar{\gamma}$, i.e., for $\gamma = \bar{\gamma}$, by studying the sign variation of the leading coefficients of subresulant polynomials over $\bar{\gamma}$ (see Sections 2.4 and 2.5).

As stated in Chapter 1, we aim at computing

$$\bar{\gamma} = \max \left\{ \pi_\gamma \left(V_{\mathbb{R}} \left(\left\langle n, \frac{\partial n}{\partial \omega} \right\rangle \right) \right) \cup V_{\mathbb{R}} \left(\langle \mathrm{Lc}_\omega(n) \rangle \right) \right\},$$

where n is the square-free part of the numerator of $\det \left(\gamma^2 I_m - F^T(-i\,\omega)\, F(i\,\omega) \right)$. Hence, $\bar{\gamma}$ is either the maximal real root of $\mathrm{Lc}_\omega(n)$ or an algebraic value over which the polynomial $\gcd\left(n(\omega, \bar{\gamma}), \frac{\partial n}{\partial \omega}(\omega, \bar{\gamma})\right)$ in ω has at least one real root. We recall that $\gcd\left(n(\omega, \bar{\gamma}), \frac{\partial n}{\partial \omega}(\omega, \bar{\gamma})\right)$ is proportional to the first subresultant polynomial $\mathrm{Sres}_i\left(n, \frac{\partial n}{\partial \omega}, \omega\right)$ (for i increasing) that does not identically vanish for $\gamma = \bar{\gamma}$ (see Theorem 2.6). If $\bar{\gamma}$ is not a real root of $\mathrm{Lc}_\omega(n)$, then we can compute the Sturm-Habicht sequence of the univariate polynomial $n(\omega, \bar{\gamma}) \in \mathbb{R}[\omega]$ to check the existence of a real root for $\gcd\left(n(\omega, \bar{\gamma}), \frac{\partial n}{\partial \omega}(\omega, \bar{\gamma})\right)$.

In what follows, we shall need the next results.

The first lemma provides a bound on the complexity for the evaluation of a univariate polynomial at a given rational point.

Lemma 3.4 (Lemma 6 of [16])**.** Let a be a rational of bitsize τ_a. The evaluation at a of a univariate polynomial P of degree d and rational coefficients of bitsize τ can be done in $\tilde{\mathcal{O}}_B\left(d\,(\tau + \tau_a)\right)$ bit operations and $P(a)$ has bitsize in $\mathcal{O}(\tau + d\,\tau_a)$.

Lemma 3.5. Let $P \in \mathbb{Z}[x, y]$ and $\bar{y}$ be a real root of $\mathrm{Res}\left(P, \frac{\partial P}{\partial x}, x\right)$. Moreover, let us note $G = \gcd\left(P(x, \bar{y}), \frac{\partial P}{\partial x}(x, \bar{y})\right) \in \mathbb{R}[x]$. If the y-projection of the points of the real plane algebraic curve $\{(x,\, y) \in \mathbb{R}^2 \mid P(x, y) = 0\}$ is bounded by $\bar{y}$, then we have $V_{\mathbb{R}}(\langle P(x, \bar{y}) \rangle) = V_{\mathbb{R}}(\langle g(x, \bar{y}) \rangle)$.

Proof. $V_{\mathbb{R}}(\langle G(x) \rangle)$ is clearly a subset of $V_{\mathbb{R}}(\langle P(x,\, \bar{y}) \rangle)$. Now, if we have $V_{\mathbb{R}}(\langle G(x) \rangle) \subsetneq V_{\mathbb{R}}(\langle P(x,\, \bar{y}) \rangle)$, then there exists $x_0 \in \mathbb{R}$ such that $P(x_0,\, \bar{y}) = 0$ and $G(x_0) \neq 0$. This is equivalent to saying that $P(x_0,\, \bar{y}) = 0$ and $\frac{\partial P}{\partial x}(x_0,\, \bar{y}) \neq 0$. Hence, based on the *implicit function theorem*, there exists a real function φ of class C^p $(p > 0)$, defined on an open

interval $V \subset \mathbb{R}$, containing $\bar{y}$, and an open neighborhood Ω of $(x_0, \bar{y})$ in $\mathbb{R}^2$ such that $\{(x, y) \in \Omega \mid P(x, y) = 0\}$ is equivalent to $\{y \in V \mid x = \varphi(y)\}$. This cannot be true since the y-projection of the points of the curve $\{(x, y) \in \mathbb{R}^2 \mid P(x, y) = 0\}$ is bounded by $\bar{y}$, and thus, an open interval containing $\bar{y}$, such as V, does not exist. Consequently, we obtain $V_{\mathbb{R}}(\langle P(x, \bar{y})\rangle) = V_{\mathbb{R}}(\langle G(x)\rangle)$.

□

Algorithm 4 Sturm-Habicht method

Input: A bivariate polynomial $P \in \mathbb{Z}[x, y]$ – seen as univariate in x – such that the $\{(x, y) \in \mathbb{R}^2 \mid P(x, y) = 0\}$ is bounded in the y-direction.
Output: Isolating interval of $\max\left\{\pi_y\left(V_{\mathbb{R}}\left(\left\langle P, \frac{\partial P}{\partial x}\right\rangle\right)\right) \cup V_{\mathbb{R}}\left(\langle \mathrm{Lc}_x(P)\rangle\right)\right\}$.

1. Compute $\{\mathrm{Sres}_j(P, \frac{\partial P}{\partial x}, x)\}_{j=0,\ldots,\deg_x(P)}$.
2. Compute $y_1 < \ldots < y_m$ the real roots of sres_0.
3. For i from 1 to m do:
   - if $y_{1-i+m} \in V_{\mathbb{R}}(\langle \mathrm{Lc}_x(P)\rangle)$ then **return** the isolating interval of y_{1-i+m};
   - **elif SignVar**$(\{\mathrm{sign}(\mathrm{stha}_{d_x}(y_{1-i+m})), \ldots, \mathrm{sign}(\mathrm{stha}_1(y_{1-i+m}))\}) > 0$, then **return** the isolating interval of y_{1-i+m}.

Lemma 3.6. Let $P \in \mathbb{Z}[x, y]$, $d_x = \deg_x(P)$, $d_y = \deg_y(P)$ and τ be the maximal coefficients bitsize of P. Let $\{\mathrm{StHa}_j\big(P(x, y), 1\big)\}_{j=0,\ldots,d_x}$ be the Sturm-Habicht sequence and y_j a real root of $\mathrm{sres}_0\big(P, \frac{\partial P}{\partial x}, x\big)$. Then, $\big\{\mathrm{sign}\big(\mathrm{stha}_k(y_j)\big)\big\}_{k=d_x,\ldots,1}$ can be computed in $\tilde{\mathcal{O}}_B\big(d_y^2\, d_x^4\, (d_y + \tau)\big)$ bit operations.

Proof. We denote by sres_0 (resp., sres_i) $\mathrm{sres}_0\big(P, \frac{\partial P}{\partial x}, x\big)$ (resp., $\mathrm{sres}_i\big(P, \frac{\partial P}{\partial x}, x\big)$), where $\mathrm{sres}_i \in \mathbb{Z}[y]$. We first recall that $\mathrm{stha}_i(y_j) = \delta_{d_x-1-i}\, \mathrm{sres}_i(y_j)$. See Definition 2.5. Based on Proposition 2.3, sres_i is of degree $d_x\, d_y$ and of coefficient bitsize $d_x\, \tau$. Thus, the square-free part of sres_0 is of coefficient bitsize $\mathcal{O}\big(d_x\, (d_y + \tau)\big)$ and, based on Theorem 2.2, can be computed in $\tilde{\mathcal{O}}_B(d_y^2\, d_x^3\, \tau)$. Hereafter in the proof, we consider sres_0 to be square-free and we denote its degree by D and its coefficient bitsize by $\mathcal{T}$. Similarly, the degree of sres_i is denoted by D' and its coefficient bitsize by $\mathcal{T}'$. Let $h_i = \gcd(\mathrm{sres}_0, \mathrm{sres}_i)$. We can compute h_i in $\tilde{\mathcal{O}}_B(D^2\, \mathcal{T})$ bit operations, where h_i is of degree $\mathcal{O}(D)$ and of coefficient bitsize $\tilde{\mathcal{O}}(D' + \mathcal{T}')$ based on Theorem 2.2. The roots of the polynomial $S_i = \mathrm{sres}_0\, \mathrm{sres}_i$ are the roots of sres_0 and sres_i. Thus, its separation bound provides a bound on how close are the (non-common)

roots of sres_0 and the roots of sres_i. The isolating intervals of the real roots of S_i are isolating intervals for the real roots of sres_0, for the real roots of sres_i and for the real roots h_i. Such interval has endpoints of bitsize s_i. Moreover, by the aggregate version of the separation bound [96, Corollary 3], we have $\sum_i s_i = \tilde{O}(D\,\mathcal{T})$. If we refine an isolating interval $I_j = [a_j,\, b_j]$ of a real root y_j of the polynomial sres_0, up to the accuracy $\tilde{O}(D\,\mathcal{T})$, then we are sure that no other real root of sres_0 or sres_i or h_i exists in the obtained refined interval. In other words, we are certain that y_j is the only root of sres_0 and sres_i that is contained in the obtained refined interval. Based on Theorem 2.10, this can be done in $\tilde{\mathcal{O}}_B(D^3 + D^2\,\mathcal{T})$. Next, we evaluate h_i at the endpoint of the refined interval. By Lemma 3.4, each evaluation costs $\tilde{\mathcal{O}}_B\big(D\,(D' + \mathcal{T}' + D\,\mathcal{T})\big)$ bit operations. Then, we get two cases for the value of sres_i at y_j: If y_j is a root of h_i, and thus a common root of sres_0 and sres_i, then the two evaluations will have different signs and $\mathrm{sres}_i(y_j) = 0$. If y_j is not a root of h_i, and thus not a root of sres_i, then the two evaluations have the same sign. Since there is no root of sres_i in the refined interval $[a_j,\, b_j]$, sres_i has a constant sign at this interval. Hence, it suffices to evaluate sres_i at one of the endpoints to obtain the $\mathrm{sign}\big(\mathrm{sres}_i(y_j)\big)$. Finally, to obtain the list $\big\{\mathrm{sign}\big(\mathrm{stha}_{d_x}(y_j)\big), \ldots, \mathrm{sign}\big(\mathrm{stha}_1(y_j)\big)\big\}$, we proceed the evaluation of sres_i over an endpoint of the refined interval of y_j for $i \in \{1, \ldots, d_x\}$. Each evaluation costs $\tilde{\mathcal{O}}_B\big(D'\,(\mathcal{T}' + D\mathcal{T})\big)$, where $D\,\mathcal{T}$ is the bit size of the isolating interval endpoints. Therefore, the d_x evaluations over all the principle subresultants cost $\tilde{\mathcal{O}}_B\big(d_x\,D'\,(\mathcal{T}' + D\mathcal{T})\big)$. Thus, the overall cost for obtaining the list $\big\{\mathrm{sign}\big(\mathrm{stha}_{d_x}(y_j)\big), \ldots, \mathrm{sign}\big(\mathrm{stha}_1(y_j)\big)\big\}$ is in the order $\tilde{\mathcal{O}}_B\big(d_y^2\,d_x^4\,(d_y + \tau)\big)$. □

Theorem 3.3. Let $P \in \mathbb{Z}[x, y]$ be such that $d_x = \deg_x(P)$, $d_y = \deg_y(P)$ and of maximal coefficient bitsize τ. Then, we can compute an isolating interval of the maximal y-projection of the real solutions of $\{P = 0, \frac{\partial P}{\partial x} = 0\}$ (Algorithm 4) in $\tilde{\mathcal{O}}_B\big(d_y^2\,d_x^4\,(d_y + \tau)\big)$ bit operations in the worst case.

Proof. The maximal y-projection of the real solutions of $\big\{P = 0, \frac{\partial P}{\partial x} = 0\big\}$ is the maximal real root of $\mathrm{sres}_0\big(P, \frac{\partial P}{\partial x}, x\big)$, say y_m, such that $\gcd\big(P(x, y_m), \frac{\partial P}{\partial y}(x, y_m)\big)$ has at least one real root. If the y-projection of the points of P is bounded by y_m, then, by Lemma 3.5, the real roots of $\gcd\big(P(x, y_m), \frac{\partial P}{\partial x}(x, y_m)\big)$ are the real roots of $P(x, y_m)$. Consequently, we can compute an isolating interval of y_m using Algorithm 4. According to Proposition 2.3, we can compute the set of principal subresultants in $\tilde{\mathcal{O}}_B(d_y\,d_x^3\,\tau)$ bit operations and each subresultant polynomial is of degree $\mathcal{O}(d_x\,d_y)$ and of coefficient bit size $\tilde{\mathcal{O}}(d_x\,\tau)$. Thus, Step 2 of Algorithm 4, which performs real root isolation of sres_0, is of complexity $\tilde{\mathcal{O}}\big((d_x\,d_y)^3 + (d_x\,d_y)^2\,d_x\,\tau\big)$ by Theorem 2.10. Using Lemma 3.6, Step 3 of Algorithm 4 can be done in $\tilde{\mathcal{O}}_B\big(d_y^2\,d_x^4\,(d_y + \tau)\big)$ operations since its first step is of complexity $\tilde{\mathcal{O}}_B(d_y^3 + d_y^2\,\tau)$. Hence, the

overall complexity of this algorithm is:

$$\tilde{\mathcal{O}}_B\big(d_y^2\, d_x^4\, (d_y + \tau)\big).$$

□

Considering the notations of Lemma 3.2, the following result is an immediate consequence of the fact that the curve $\{(\omega,\, \gamma) \in \mathbb{R}^2 \mid n(\omega,\, \gamma) = 0\}$ is bounded in the direction of γ (see Section 1.4) and Theorem 3.3.

Corollary 3.2. Based on Theorem 3.3, with the notations of Lemma 3.2, and by considering $d_x = d_\omega$, $d_y = d_\gamma$, the worst case bit complexity for the computation of $\| F \|_\infty$ with the Sturm-Habicht method (Algorithm 4) applied to the polynomial $n(\omega, \gamma)$ defined by (3.1) is given by:

$$\tilde{\mathcal{O}}_B\big(d_\omega^4\, d_\gamma^2\, (d_\gamma + \tau_n)\big) \;=\; \tilde{\mathcal{O}}_B\big(\alpha^{10}\, N^4\, (\alpha + \tau_n)\big).$$

In Algorithm 5, we suppose that there are no real *isolated singular points*, and thus, we replace the computation of signs of polynomials at real algebraic numbers by signs of polynomials at rational numbers. Syntactically, these are small modifications but the effect on the computations is consequent in practice, as well as in theory, since the evaluation of signs of polynomials at real algebraic numbers carries the theoretical worst case complexity of Algorithm 4.

Algorithm 5 Sturm-Habicht method − equidimensional

Input: A bivariate polynomial $P \in \mathbb{Z}[x, y]$ − seen as univariate in x − such that the algebraic plane curve $\{(x,\, y) \in \mathbb{R}^2 \mid P(x, y) = 0\}$ is bounded in the y-direction and has not real isolated singular points.
Output: An isolating interval of $\max\left\{\pi_y\left(V_\mathbb{R}\left(\left\langle P, \frac{\partial P}{\partial x}\right\rangle\right)\right) \cup V_\mathbb{R}\left(\langle \mathrm{Lc}_x(P)\rangle\right)\right\}$

1. Compute $\{\mathrm{StHa}_j(P, 1)\}_{j=0,\ldots,\deg_x(P)}$.
2. Let $y_1 < \ldots < y_m$ be the real roots of sres_0.
3. for i from 1 to m do:
   - if $y_{1-i+m} \in V_\mathbb{R}(\mathrm{Lc}_x)$, then **return** the isolating interval of y_{1-i+m};
   - **else** let $Y' \in \mathbb{Q}$ such that $y_{m-i} < Y' < y_{1-i+m}$;
     - if **SignVar**$\big(\{\, \mathrm{sign}(\mathrm{stha}_{d_x}(Y')), \ldots, \mathrm{sign}(\mathrm{stha}_1(Y'))\, \}\big) > 0$, then **return** the isolating interval of y_{1-i+m};

Theorem 3.4. Let $P \in \mathbb{Z}[x, y]$ be a bivariate polynomial of maximal coefficient bitsize τ and let $d_x = \deg_x(P)$ and $d_y = \deg_y(P)$. Moreover, let us suppose that $V_{\mathbb{R}}(\langle P \rangle)$ has no isolated singular points. Using Algorithm 5, an isolating interval of the maximal y-projection of the real solutions of $\{P = 0, \frac{\partial P}{\partial x} = 0\}$ can be computed in the worst case bit complexity $\tilde{\mathcal{O}}_B(d_y^2\, d_x^4\, \tau)$.

Proof. As mentioned in the proof of Theorem 3.3, we can compute the set of principal subresultants in $\tilde{\mathcal{O}}_B(d_y\, d_x^3\, \tau)$ bit operations and, by Proposition 2.3, each subresultant polynomial is of degree $\mathcal{O}(d_x\, d_y)$ and of coefficient bitsize $\tilde{\mathcal{O}}(d_x\, \tau)$. Thus, Step 2 of Algorithm 5, which performs the real root isolation of sres_0, is of complexity $\tilde{\mathcal{O}}\big((d_x\, d_y)^3 + (d_x\, d_y)^2\, d_x\, \tau\big)$ by Theorem 2.10. Steps 3 and 4 are of same bit complexity: in these steps, we perform $\mathcal{O}(d_x)$ evaluations of the principal subresultant polynomials over a rational number which is between two real roots of sres_0. This rational number is of worst possible coefficient bitsize $\tilde{\mathcal{O}}_B(d_x^2\, d_y\, \tau)$, which is equal to the separating bound of sres_0. According to Lemma 3.4, the d_x evaluations are done in $\tilde{\mathcal{O}}_B\left(d_x\left(d_x\, d_y\, (d_x\, \tau + d_y\, d_x^2\, \tau)\right)\right) = \tilde{\mathcal{O}}_B(d_y^2\, d_x^4\, \tau)$. Hence, the overall cost is given by $\tilde{\mathcal{O}}_B(d_y^2\, d_x^4\, \tau)$. □

Corollary 3.3. Based on Theorem 3.4, $\| F \|_\infty$ can be computed by the Sturm-Habicht method (Algorithm 5) in the worst case bit complexity $\tilde{\mathcal{O}}_B(\alpha^{10}\, N^4\, \tau_n)$.

From the above complexity analysis, we can conclude that RUR method and the Sturm-Habicht method have comparable theoretical complexities since, we usually have $\alpha \ll N$.

The next proposition proves that the curve $\mathcal{C} = \{(\omega,\, \gamma) \in \mathbb{R}^2 \mid n(\omega, \gamma) = 0\}$, associated with the L^∞-norm computation of an element $F \in RL_\infty$ (e.g., stable SISO systems), has no (isolated) singularities. Hence, Corollary 3.3 holds for $F \in RL_\infty$ and $\| F \|_\infty$ can be computed by Algorithm 5.

Proposition 3.2. The curve $\mathcal{C} = \{(\omega,\, \gamma) \in \mathbb{R}^2 \mid n(\omega, \gamma) = 0\}$, associated with the L^∞-norm computation of an element $F \in RL_\infty$ has no (isolated) singularities.

Proof. Let $F \in RL_\infty$, i.e., $F = a/b$, where $a,\, b \in \mathbb{R}[s]$, $\gcd(a, b) = 1$, $q = \deg_s(a) \leq r = \deg_s(b)$, and b does not vanish on $i\,\mathbb{R}$. See Section 1.3. In this case, $F(i\,\infty) = 0$ if $q < r$, (i.e., if F is strictly proper) or $F(i\,\infty) = a_r/b_r$ if $q = r$ (i.e., if F is proper), where $a_r = \mathrm{Lc}_s(a)$ and $b_r = \mathrm{Lc}_s(b)$. Moreover, we have:

$$\Phi_\gamma(i\,\omega) = \gamma^2 - F(-i\,\omega)\, F(i\,\omega) \;=\; \gamma^2 - |F(i\,\omega)|^2.$$

Writing $|F(i\,\omega)|^2 = |a(i\,\omega)|^2/|b(i\,\omega)|^2 = N(\omega)/D(\omega)$, where N and D are coprime polyno-

mials of $\mathbb{R}[\omega^2]$, we then have:

$$\Phi_\gamma(s) = \gamma^2 - \frac{N(\omega)}{D(\omega)} = \frac{D(\omega)\,\gamma^2 - N(\omega)}{D(\omega)}.$$

Hence, we get $n(\omega,\,\gamma) = D(\omega)\,\gamma^2 - N(\omega)$. Let us define the polynomial system defining the set of singular points of the curve $\mathcal{C} = \{(\omega,\,\gamma) \in \mathbb{R}^2 \mid n(\omega,\gamma) = 0\}$:

$$\Sigma' = \left\{ n(\omega,\gamma) = 0,\ \frac{\partial n}{\partial \omega}(\omega,\gamma) = 0,\ \frac{\partial n}{\partial \gamma}(\omega,\gamma) = 0 \right\}.$$

More precisely, we have:

$$\Sigma' = \begin{cases} n(\omega,\,\gamma) = D(\omega)\,\gamma^2 - N(\omega) = 0, \\ \dfrac{\partial n}{\partial \omega}(\omega,\,\gamma) = D'(\omega)\,\gamma^2 - N'(\omega) = 0, \\ \dfrac{\partial n}{\partial \gamma}(\omega,\,\gamma) = 2\,D(\omega)\,\gamma = 0, \end{cases}$$

Hence, $D(\omega) \neq 0$ for all $\omega \in \mathbb{R}$ implies that $(\omega,\,\gamma) \in \Sigma'$ if and only if $\gamma = 0$ and $N'(\omega) = N(\omega) = 0$. Consequently, for all $F \in RL_\infty$ such that $F \neq 0$, $n(\omega,\,\gamma)$ has no real (isolated) singular points, and the proof holds. □

Example 43. Let us consider the transfer function defined in Example 46 for the particular numerical values $\omega_0 = 2$, $\omega_1 = 1$ and $\xi = 1/2$, i.e.:

$$G = \frac{s^2 + 2\,s + 4}{4\,(s^2 + s + 1)}.$$

Then, we obtain the polynomial

$$n(\omega,\,\gamma) = (4\,\gamma - 1)\,(4\,\gamma + 1)\,\omega^4 - 4\,(2\,\gamma - 1)\,(2\,\gamma + 1)\,\omega^2 + 16\,(\gamma - 1)\,(\gamma + 1),$$

which defines the real plane algebraic curve plotted in Figure 3.4.

We can check that $G(i\,\infty) = 1/4$. This value can be again found as the maximal real root of $\mathrm{Lc}_\omega(n)$, and thus, as a real root of $R_\gamma = \mathrm{Res}\big(n, \frac{\partial n}{\partial \omega}, \omega\big)$.

Let us consider $\Sigma = \{n = 0, \frac{\partial n}{\partial \omega} = 0\}$. To obtain the γ-projection of the real solutions of Σ, we compute the square-free part R of R_γ:

$$R = (\gamma - 1)\,(\gamma + 1)\,(4\,\gamma + 1)\,(4\,\gamma - 1)\,(16\,\gamma^4 - 20\,\gamma^2 + 1) \in \mathbb{Z}[\gamma].$$

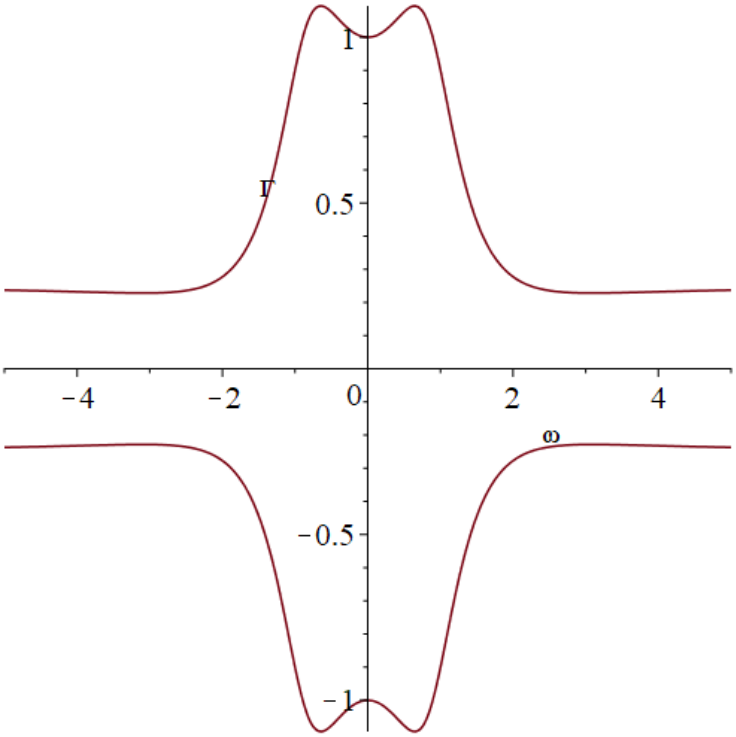

Figure 3.4: Plot of $n(\omega,\,\gamma)=0$, where ω/γ is in the horizontal/vertical axis.

Then, the maximal real root of R has the following isolating interval:

$$[a,\,b]=\left[\frac{3307773758987265766065 29}{302231454903657293676544},\frac{826943439746816441526 65}{755578637259143234191 36}\right].$$

Following Algorithm 5, it suffices to verify the existence of a real root for the univariate polynomial $n(\omega,a)\in\mathbb{Q}[\omega]$. This can be easily verified through many ways, but using Algorithm 5, we compute $L=[\mathrm{stha}_i(n,1)]_{i=1,\ldots,\deg_\omega(n)=4}$:

$$\begin{aligned}L=[&-24\,(2\,\gamma+1)\,(2\gamma-1)\,(16\,\gamma^4-20\,\gamma^2+1)\,(4\,\gamma-1)^2\,(4\,\gamma+1)^2,2\,(2\,\gamma+1)\\&(2\,\gamma-1)\,(4\,\gamma-1)^2\,(4\,\gamma+1)^2,\,4\,(4\,\gamma-1)\,(4\,\gamma+1),\,(4\,\gamma-1)(4\,\gamma+1)].\end{aligned}$$

After substituting $\gamma=a$ in L, we obtain the list of signs $L_s=[0,+,+,+]$. Then, we have $\mathbf{SignVar}(L_s)=2$ and we conclude that $n(\omega,[a,b])$ admits two real roots. Comparing a with the maximal real root of $\mathrm{Lc}_\omega(n)$, we can then say that $\|\,G\,\|_\infty$ is equal to the maximal real root of R having an isolating interval $[a,\,b]$.

Example 44. We consider again the transfer matrix defined in Example 39. As done in

Example 41, we compute $R_\gamma = \mathrm{Res}(n, \frac{\partial n}{\partial \omega}, \omega)$ and its square-free part:

$$R = \gamma\,(\gamma^2 + \gamma - 1)\,(\gamma^2 - \gamma - 1) \in \mathbb{Z}[\gamma].$$

Then, the maximal real root of R has the following isolating interval:

$$[a,\, b] = \left[\frac{56929509912547}{35184372088832}, \frac{113859019825121}{70368744177664}\right].$$

We now have to check the existence of a real root for the univariate polynomial $n(\omega, [a,\, b])$. To do that, we first compute

$$L = [\mathrm{stha}_i(n, 1)]_{i=1,\ldots,\deg_\omega(n)=4} = [-5\,\gamma^{14}\,(2\,\gamma^2 - 3), -2\,\gamma^{10}\,(2\,\gamma^2 - 3), 4\,\gamma^4, \gamma^4].$$

Then, we compute the list of signs of the elements of L over $[a,\, b]$. We obtain the list $L_s = [-, -, +, +]$. Hence, $\mathbf{SignVar}(L_s) = 1$ and we can conclude that $n(\omega, [a,\, b])$ admits one real root. Hence, after comparing $[a,\, b]$ with the isolating interval of the maximal real root of $\mathrm{Lc}_\omega(n)$, we can say that $\parallel G \parallel_\infty$ is equal to the maximal real root of R having an isolating interval $[a,\, b]$.

Example 45. We consider again the transfer matrix of Example 40. We start by computing the square-free part R of $R_\gamma = \mathrm{Res}(n, \frac{\partial n}{\partial \omega}, \omega)$ and then isolating intervals for its real roots. Since $G \in RL_\infty$, $n(\omega, \gamma)$ has no real (isolated) singular points and thus, we can use Algorithm 5 to compute $\parallel G \parallel_\infty$. Let us consider $L = [\mathrm{stha}_i(n, 1)]_{i=1,\ldots,\deg_\omega(n)=4}$, where:

$$\begin{aligned} L =&[9\,(2\,\gamma + 1)\,(2\gamma - 1)\,(112\,\gamma^4 - 120\,\gamma^2 + 7)\,(4\,\gamma - 1)^2\,(4\,\gamma + 1)^2, -2\,(2\,\gamma + 1) \\ &(2\,\gamma - 1)\,(4\,\gamma - 1)^2\,(4\,\gamma + 1)^2,\ 4\,(4\,\gamma - 1)\,(4\,\gamma + 1),\ (4\,\gamma - 1)\,(4\,\gamma + 1)]. \end{aligned}$$

With the notations used in Example 42, we substitute $\gamma = a_4$ in L for verifying the existence of a real root for the univariate polynomial $n(\omega, a_4) \in \mathbb{Q}[\omega]$ and we obtain $L_s = [0, -, +, +]$. Hence, we have $\mathbf{SignVar}(L_s) = 0$ and we can conclude that $n(\omega, [a_4,\, b_4])$ admits no real roots. Next, we substitute $\gamma = a_3$ in L and we obtain the list of signs $L_s = [-, -, +, +]$ and $\mathbf{SignVar}(L_s) = 2$. Hence, we can say that $n(\omega, [a_3,\, b_3])$ admits two real roots. After comparing $\gamma = 1$ with the isolating interval $[a_3,\, b_3]$ with the maximal real root $\gamma = 1/4$ of $\mathrm{Lc}_\omega(n)$, we conclude that $\parallel G \parallel_\infty$ is equal to $\gamma = 1$.

In the next section, we shall generalise the methods developed in Section 3.1 for the computation of L^∞-norm of $F \in RL_\infty^{p\times m}$ to the case where F also depends on parameters α. The searched value γ will then be represented as a real function of α. For a precise

parameter value $\alpha_\star$, we know that the system Σ has a constant number of real solutions. Thus, to solve our problem in presence of parameters, we shall choose a finite number of representative parameter values that cover the conditions we are searching for. This can be done by using the well-known concept in computer algebra so-called *Cylindrical Algebraic Decomposition* (CAD) (see Section 2.8.2).

3.2 Parametric case

In this section, we consider the case of a matrix which coefficients depend on a set of parameters $\alpha = (\alpha_1, \ldots, \alpha_d)$. Adapting the previous definitions, we obtain the following proposition.

Corollary 3.4. Let $\alpha = (\alpha_1, \ldots, \alpha_d)$ be a set of unknown parameters. Let $F \in RL_\infty^{p\times m}$ and $n \in \mathbb{Z}[\alpha][\omega, \gamma]$ be defined by (3.1). We consider:

$$\begin{cases} \Sigma = \big\{(\omega, \gamma, \alpha) \in \mathbb{R}^{2+d} \mid n(\omega, \gamma, \alpha) = 0, \frac{\partial n}{\partial \omega}(\omega, \gamma, \alpha) = 0\big\}, \\ \Sigma_\infty = \big\{(\gamma, \alpha) \in \mathbb{R}^{1+d} \mid \mathrm{Lc}_\omega(n) = 0\big\}. \end{cases}$$

Thus, we have:

$$\forall\, \alpha \in \mathbb{R}^d, \quad \| F \|_\infty = \max\Big(\pi_\gamma(\Sigma) \cup \Sigma_\infty\Big).$$

In other words, the norm that we aim at computing is in this case is a function of the parameters. By substituting the parameters with random rational values, we can simply compute the L^∞-norm of a transfer matrix by applying one of the proposed methods in Section 3.1. Yet, there is no guarantee that parameter values with the desired properties can be found even if they exist. Therefore, to compute the L^∞-norm based on the stated definitions, it is essential to choose a finite number of representative "good" parameter values that cover all possible cases. This set of "good" parameter values can be expressed as a finite disjoint union of connected open sets. Within the same set, the number of the system real solutions does not change when the parameters vary, and moreover, the position of the curves representing the γ-projection of the system's solutions does not change. This last detail is crucial since we are mainly interested in the maximality of γ.

Example 46. We compute the L^∞-norm of the following transfer function

$$G = \frac{\left(\frac{s}{\omega_0}\right)^2 + 2\,\xi\left(\frac{s}{\omega_0}\right) + 1}{\left(\frac{s}{\omega_1}\right)^2 + 2\,\xi\left(\frac{s}{\omega_1}\right) + 1},$$

depending on three parameters $\xi, \omega_0, \omega_1 \in \mathbb{R}_{>0}$, $\omega_0 \neq \omega_1$, where $0 < \xi \leq 1$. Hence, $\| G \|_\infty$ is a function on these parameters. This norm is explicitly computed in the following Proposition 3.3.

Proposition 3.3. Let $\omega_0,\ \omega_1 \in \mathbb{R}_{>0}$, $\omega_0 \neq \omega_1$, $0 < \xi \leq 1$ and:

$$G = \frac{\left(\frac{s}{\omega_0}\right)^2 + 2\,\xi\left(\frac{s}{\omega_0}\right) + 1}{\left(\frac{s}{\omega_1}\right)^2 + 2\,\xi\left(\frac{s}{\omega_1}\right) + 1}.$$

Set $r = \omega_1/\omega_0$, $\mu = 4\,\xi^2\,(\xi - 1)\,(\xi + 1)$ and let δ be the maximal real root of:

$$M = \mu\,\gamma^4 + \left((r^2 - 1)^2 - 2\,\mu\,r^2\right)\gamma^2 + \mu\,r^4 \in \mathbb{R}[\gamma].$$

Then, the L^∞-norm of G is given by:

$$\| G \|_\infty = \begin{cases} \max\{1, r^2\} & \text{if } \xi \geq \frac{1}{\sqrt{2}}, \\ \delta & \text{if } \xi < \frac{1}{\sqrt{2}}. \end{cases} \tag{3.5}$$

Before proving Proposition 3.3, we first give two useful lemmas.

Lemma 3.7. Let us consider $\xi, \omega_0, \omega_1 \in \mathbb{R}_{>0}$, $\omega_0 \neq \omega_1$, $0 < \xi < 1$, $r = \omega_1/\omega_0$, and $\mu = 4\,\xi^2\,(\xi - 1)\,(\xi + 1)$. Then, the following polynomial

$$M_1 = \mu\,X^2 + \left((r^2 - 1)^2 - 2\,\mu\,r^2\right)X + \mu\,r^4 \in \mathbb{R}[X]$$

has two positive real roots X_1 and X_2 verifying $0 < X_1 < 1 < X_2$ and $X_2 > r^4$.

Proof. The discriminant of M_1 is $\Delta = (r+1)^2\,(r-1)^2\left((r^2-1)^2 - 4\,\mu\,r^2\right)$. Since $0 < \xi \leq 1$, we get $-\mu \leq 0$, and thus, $\Delta > 0$, which shows that M_1 has two distinct real solutions,

denoted by X_1 and X_2 with the assumption that $X_1 < X_2$. Moreover, we have

$$X_1\,X_2 = r^4 > 0, \quad X_1 + X_2 = \frac{(r^2-1)^2 - 2\,\mu\,r^2}{-\mu} > 0,$$

which yields $X_1 > 0$ and $X_2 = r^4/X_1 > 0$. Now, if we let $x := X - 1$, then we get $M_1(X) = M_1(x+1) = m(x)$, where:

$$m(x) = \mu\,x^2 + \big((r^2-1)^2 - 8\,\xi^2\,(1-\xi^2)\,(r^2+1)\big)\,x + (2\,\xi^2-1)^2\,(r-1)^2\,(r+1)^2.$$

Clearly, the two roots of m are $X_1 - 1$ and $X_2 - 1$, and we have

$$(X_1-1)\,(X_2-1) = \frac{(2\,\xi^2-1)^2\,(r-1)^2\,(r+1)^2}{\mu} < 0,$$

which shows that $X_1 < 1$ and $X_2 > 1$ since $X_1 < X_2$. Finally, $X_1 < 1$ yields $X_2 = r^4/X_1 > r^4$, which proves the result. □

Lemma 3.8. Let ω_0, $\omega_1 \in \mathbb{R}_{>0}$, $\omega_0 \neq \omega_1$, $0 < \xi < 1$, $\xi \neq 1/\sqrt{2}$, $r = \omega_1/\omega_0$ and $\beta = 2\,\xi^2 - 1$. Then, the following polynomial

$$L = \beta\,Y^2 + \omega_0^2\,(r^2+1)\,Y + \beta\,r^2\,\omega_0^4 \in \mathbb{R}[Y]$$

has two positive real roots if and only if $0 < \xi < 1/\sqrt{2}$.

Proof. The discriminant of L is $\delta = w_0^4\,(r^2 + 2\,\beta\,r + 1)\,(r^2 - 2\,\beta\,r + 1)$. We have $\beta^2 - 1 = 4\,\xi\,(\xi - 1) < 0$, and thus, the discriminant $4\,(\beta^2 - 1)$ of the two polynomials $r^2 + 2\,\beta\,r + 1$ and $r^2 - 2\,\beta\,r + 1$ is negative, which yields $\delta > 0$ and thus, L has two distinct real roots. The product of these roots is $r^2\,\omega_0^4 > 0$ and their sum is $\omega_0^2\,(r^2+1)/(-\beta)$. Since $\beta < 0$ if and only if $0 < \xi < 1/\sqrt{2}$, we obtain that the sum is positive if and only if $0 < \xi < 1/\sqrt{2}$, which then implies that L has two positive real roots only when $0 < \xi < 1/\sqrt{2}$. □

We can tate the proof of Proposition 3.3 based on Lemmas 3.7 and 3.8.

Proof. Let N and D be two polynomials such that:

$$G(-i\,\omega)\,G(i\,\omega) = \frac{N(\omega)}{D(\omega)}.$$

Let $n(\gamma,\,\omega) = D(\omega)\,\gamma^2 - N(\omega)$, $\alpha = [r, \omega_0, \xi] \in \mathbb{R}^3$ and consider

$$\begin{cases} \Sigma = \big\{(\omega,\,\gamma) \in \mathbb{R}^2,\ n(\omega,\,\gamma) = 0,\ \frac{\partial n}{\partial \omega}(\omega,\,\gamma) = 0\big\}, \\ \Sigma_\infty = \big\{\gamma \in \mathbb{R},\ \mathrm{Lc}_\omega(n) = 0\big\}. \end{cases}$$

Doing the computation, we obtain

$$n = (\gamma^2 - r^4)\,\omega^4 + 2\,r^2\,\omega_0^2\,\beta\,(\gamma^2 - r^2)\,\omega^2 + r^4\,\omega_0^4\,(\gamma^2 - 1),$$

where $\beta = 2\,\xi^2 - 1$. The resultant R of $n(\omega, \gamma)$ and $\frac{\partial n}{\partial \omega}(\omega, \gamma)$ with respect to ω is then:

$$R = 256\,w_0^{12}\,r^{12}\,(\gamma^2 - 1)\,(\gamma^2 - r^4)^2\,M^2,$$

where $M = \mu\,\gamma^4 + ((r^2 - 1)^2 - 2\,\mu\,r^2)\,\gamma^2 + \mu\,r^4$, i.e., $M(\gamma) = M_1(\gamma^2)$ with M_1 defined in Lemma 3.7. By assumptions on the parameters, $M_1(X)$ has two positive real solutions, X_1 and X_2, and thus, M has the four real roots $\pm\sqrt{X_1}$, $\pm\sqrt{X_2}$. Thus, based on the properties of the resultant polynomial, we have:

$$\parallel G \parallel_\infty = \max\left\{\left\{1,\,r^2,\,\sqrt{X_1},\,\sqrt{X_2}\right\} \cap \Big(\pi_\gamma(\Sigma) \cup \Sigma_\infty\Big)\right\}.$$

1. For $\gamma = r^2$: $\mathrm{Lc}_\omega(n) = (\gamma^2 - r^4) = 0$, i.e, $r^2 \in \Sigma_\infty$.

2. For $\gamma = 1$, we get:

$$\begin{cases} n(\omega,\,1) = (r^2 - 1)\,\omega^2\,f_1, \\ q(\omega,\,1) = 4\,(r^2 - 1)\,\omega\,f_2, \end{cases}, \quad \begin{cases} f_1 := (r^2 + 1)\,\omega^2 + 2\,r^2\,\omega_0^2\,(2\,\xi^2 - 1), \\ f_2 := (r^2 + 1)\,\omega^2 + r^2\,\omega_0^2\,(2\,\xi^2 - 1). \end{cases}$$

   Hence, $\mathrm{Res}(f_1,\,f_2,\,\omega) = \big(\omega_0\,r^2\,(\xi^2{+}1)\,(r^2{+}1)\big)^2$ does not vanish. Thus, $\gcd(f_1, f_2) = 1$, which yields $\gcd\big(n(\omega,\,1),\,q(\omega,\,1)\big) = (r^2 - 1)\,\omega$ and proves:

$$(\omega,\,\gamma) = (0,\,1) \in \Sigma.$$

3. For γ real root of M: The point is to verify that $\gamma \in \pi_\gamma(\Sigma)$. For doing so, we start by computing $F = \mathrm{Res}(n(\omega, \gamma), \frac{\partial n}{\partial \omega}(\omega, \gamma), \gamma)$. We recall that based on the properties of resultants, $\pi_\omega(\Sigma) \subset V(F)$, where $F \in \mathbb{R}[\omega]$:

$$F = c\,\omega^2\,F_1^2, \quad \begin{cases} F_1 = \beta\,\omega^4 + \omega_0^2\,(r^2+1)\,\omega^2 + r^2\omega_0^4\,\beta, \\ c = 16\,\omega_0^{20}\,r^8\,(r^2-1)^2. \end{cases}$$

We notice that the roots of $F_1 \in \mathbb{R}[\omega]$, are the ω-coordinates of $(\omega_{i,j},\,\gamma_i) \in V_{\mathbb{R}\times\mathbb{C}}(\langle n,\,\frac{\partial n}{\partial\omega}\rangle)$, where $\gamma_i \in V_{\mathbb{R}}(M)$: In fact, we have seen that $\pi_\omega\Big(V\big(\langle n(\omega,\,\pm1),\frac{\partial n}{\partial\omega}(\omega,\,\pm1)\,\rangle\big)\Big) = 0$. By taking into consideration the power and degree of the factor ω in F and the power and degree of the factor γ^2-1 in R, and by following the properties of the resultant concerning root multiplicity (see, e.g., [32, Chapter 4]), we can say that

$$(\omega,\,\gamma) \in \Sigma,\ \gamma = \pm1 \iff \omega = 0.$$

With this being said, let $L \in \mathbb{R}[Y]$ be the polynomial obtained after substituting ω^2 by Y in F_1. Based on Lemma 3.8, L has two positive real roots if and only if $\xi < \frac{\sqrt{2}}{2}$. Thus we can conclude two cases:

- For $\xi < \frac{\sqrt{2}}{2}$, $F_1 \in \mathbb{R}[\omega]$ has four real roots. Thus,

  $$\forall\gamma_i \in V_{\mathbb{R}}(M),\ \exists\,\omega_{i,j} \in V_{\mathbb{R}}(F_1),\ \text{such that } (\omega_{i,j},\,\gamma_i) \in \Sigma.$$

  Consequently, based on Lemma 3.7 where we proved that

  $$X_1 < 1 < X_2, \quad X_2 > r^4,$$

  we can say that $\delta = \sqrt{X_2} > r^2$, and we conclude that $\|\,G\,\|_\infty = \delta$.

- For $\xi > \frac{\sqrt{2}}{2}$, F_1 has no real roots in ω. In this case, none of the real roots of M is a good candidate, and we conclude that:

  $$\|\,G\,\|_\infty = \max\{1, r^2\}.$$

4. For $\xi = \frac{\sqrt{2}}{2}$, $M = c\,(\gamma^2-1)\,(\gamma^2-r^4)$, where $c \in \mathbb{R}$. In this case, we have:

   $$\|\,G\,\|_\infty = \max\{1, r^2\}.$$

5. Similarly, for $\xi = 1$, $M = c\,\gamma\,(\gamma^2 - 1)\,(\gamma^2 - r^4)$ where $c \in \mathbb{R}$. Then, we have:

$$\| G \|_\infty = \max\{1, r^2\}.$$

□

A way for obtaining such decomposition is by computing a Cylindrical Algebraic Decomposition (CAD) of $\mathbb{R}^{d+2}$ adapted to $\{n(\omega,\, \gamma) = 0,\, \frac{\partial n}{\partial \omega}(\omega,\, \gamma) = 0\}$ and a semi-algebraic set S_p containing the inequalities verified by the parameters. But it may give us a result very huge and difficult to analyse in practice, and with lots of cells we are not interested in. In fact, we are just interested in the cells where the curves representing the γ-projection of the system solutions, when they exist, are continuous and do not intersect.

In the first step in order to obtain a decomposition easier to manipulate, we shall compute the discriminant variety R of $\big\{n(\omega,\, \gamma) = 0\big\}$ with respect to $\Pi_{\alpha,\gamma}$, where $n(\omega,\, \gamma)$ is seen as a univariate polynomial in ω. We recall that this discriminant variety is the set of parameter values leading to non-generic solutions of the system, for example, infinitely many solutions, solutions at infinity, or solutions of multiplicity greater than 1. It is simply the resultant polynomial of the polynomial $n(\omega,\, \gamma)$ and its derivative with respect to the main variable ω, i.e., $R = \mathrm{Res}\left(n(\omega,\gamma), \frac{\partial n}{\partial \omega}(\omega,\gamma), \omega\right) \in \mathbb{Q}[\alpha][\gamma]$. In this case, $R = 0$ is a sub-variety of $\mathbb{R}^{d+1}$ and the complement of $R = 0$, $\mathbb{R}^{d+1} \backslash \{R = 0\}$, can be expressed as a finite disjoint union of cells, which are connected open sets, such that the number of the system real solutions ω does not change when the parameters vary within the same cell.

In the second step, we shall consider the variable γ as the main variable in the polynomial $R \in \mathbb{Q}[\alpha,\ \gamma]$ for it is the polynomial embodying the γ-projection of the system $\big\{n(\omega,\, \gamma) = 0,\ \frac{\partial n}{\partial \omega}(\omega,\, \gamma) = 0\big\}$. In this case, the γ-projection of the system solutions is considered as a real function of α such that the position of the curves representing $\gamma(\alpha) = 0$ changes after each intersection of at least two curves in $\mathbb{R}^d$. We shall thus decompose $\mathbb{R}^d$ into cells where no changes in the position of the curves of $\gamma(\alpha)$ occur in order to be able to locate the maximal value γ over a given cell. For doing so, we can naturally propose to eliminate from the parameter space the set of "bad" parameter values leading to non-generic solutions of $R = 0$, i.e., the discriminant variety of $\{(\alpha, \gamma) \in \mathbb{R}^{d+1} \mid R = 0\} \cup S_p$ with respect to Π_α, that we denote R_2. We recall that in this case, R_2 is simply the curve of the discriminant of R with respect to the variable γ, multiplied by the leading coefficient of R with respect to γ, i.e., $\mathrm{Res}\left(R, \frac{\partial R}{\partial \gamma}, \gamma\right) \in \mathbb{Z}[\alpha]$, up to some curves related to the inequalities of S_p.

Then, using a CAD, we can decompose $C = \mathbb{R}^d \backslash R_2$ into connected cells, above each cell, the variable γ is represented as real valued functions depending continuously on the

parameters, whose graphs are disjoint.

We represent the algorithm describing our proposed method.

The proposed algorithm Let $\{C_1, \ldots, C_l\}$ be the partition of C, and let $\texttt{sample}_\texttt{i}$ be a sample point in C_i. We consider $\texttt{index}_\texttt{i}$ to be the index of

$$\gamma_{\max} = \max\left\{\pi_\gamma\left(V_{\mathbb{R}}\left(\left\langle n, \frac{\partial n}{\partial \omega}\right\rangle\right)\right) \cup V_{\mathbb{R}}\left(\langle \mathrm{Lc}_\omega(n)\rangle\right)\right\}$$

in the sorted set of the real roots of R. Thus, we can represent the searched value over a cell C_i by the couple $[C_i, \texttt{index}_\texttt{i}]$.

We denote `non_parametric` one of the proposed methods for computing $\gamma_{\max}$ in the non-parametric case. Thus, we state Algorithm 6 that, given as input a bivariate polynomial system $\Sigma = \{n, \frac{\partial n}{\partial \omega}\}$ of unknowns $[\omega, \gamma]$ and depending on the parameters set $\alpha = [\alpha_1, \ldots, \alpha_d]$ that verifies certain conditions, outputs a list of couples $[C_i, \texttt{index}_\texttt{i}]$, for $i \in \{1, \ldots, l\}$, defined in the previous paragraph.

Algorithm 6 Parametric case

Input: A zero-dimensional polynomial system $\Sigma = \left\{n(\omega, \gamma), \frac{\partial n}{\partial \omega}(\omega, \gamma)\right\} \subset \mathbb{Z}[\alpha][\omega, \gamma]$ and a semi-algebraic set S_p.
Output: A list of couples $\left\{[C_i, \texttt{index_i}], i = 1, \ldots, l\right\}$

1. Compute $R = \mathrm{Res}(n, \frac{\partial n}{\partial \omega}, \omega)$,
2. Compute R_2, the discriminant variety of $\{R = 0, S_p\}$ with respect to Π_α,
3. Using a `CAD`, compute the partition $\{C_1, \ldots, C_l\}$ of $C = \mathbb{R}^d \backslash \mathrm{R}_2$, along with sample points $\texttt{sample}_\texttt{i} \in C_i$,
4. Apply `non_parametric` on $\texttt{subs}(\texttt{sample}_\texttt{i}, \Sigma)$ and get `index_i`,
5. `return` $\left\{[C_i, \texttt{index_i}], i = 1, \ldots, l\right\}$.

Example 47. We consider the transfer function studied in Example 46

$$G = \frac{\left(\frac{s}{\omega_0}\right)^2 + 2\,\xi\left(\frac{s}{\omega_0}\right) + 1}{\left(\frac{s}{\omega_1}\right)^2 + 2\,\xi\left(\frac{s}{\omega_1}\right) + 1},$$

for $S_p = \{\omega_0 > 0,\, \omega_1 > 0,\, \omega_0 \neq \omega_1,\, 0 < \xi \leq 1\}$. We apply Algorithm 6 to the corresponding

polynomial equation system $\Sigma = \left\{n(\omega,\,\gamma) = 0, \frac{\partial n}{\partial \omega}(\omega,\,\gamma) = 0\right\}$ and S_p in order to represent the L^∞-norm of G.

Let $R = \mathrm{Res}(\mathrm{n}, \frac{\partial \mathrm{n}}{\partial \omega}, \omega) \in \mathbb{Z}[\alpha, \gamma]$, where $n(\omega,\,\gamma)$ and R are already computed in Example 46. In this case, by doing the computation using `Maple` functions such as `DiscriminantVariety` and `CellDecomposition` of the package `RootFinding[Parametric]`, the discriminant variety R_2 is given by the union of the curves defined by the following polynomials:

$$\left\{r,\,\omega_0,\,\xi,\,r-1,\,r+1,\,\xi-1,\,\xi+1,\,2\xi^2-1,\,-4r\xi^2+r^2+2r+1,\,4r\xi^2+r^2-2r+1\right\}.$$

By computing a `CAD` of $C = \mathbb{R}^d \backslash R_2$, we get the partition $\{C_1, \ldots, C_4\}$ where

$$C_1 = \left\{0 < \xi < \sqrt{2}/2\right\} \cap \left\{0 < r < 1\right\} \cap \left\{\omega_0 > 0\right\},$$

$$C_2 = \left\{0 < \xi < \sqrt{2}/2\right\} \cap \left\{r > 1\right\} \cap \left\{\omega_0 > 0\right\},$$

$$C_3 = \left\{\sqrt{2}/2 < \xi < 1\right\} \cap \left\{0 < r < 1\right\} \cap \left\{\omega_0 > 0\right\},$$

$$C_4 = \left\{\sqrt{2}/2 < \xi < 1\right\} \cap \left\{r > 1\right\} \cap \left\{\omega_0 > 0\right\},$$

and the sample points

$$\mathtt{sample}_1 = [\xi = \frac{25476206690102465}{72057594037927936}, r = 1/2, \omega_0 = 1],$$

$$\mathtt{sample}_2 = [\xi = \frac{25476206690102465}{72057594037927936}, r = 2, \omega_0 = 1],$$

$$\mathtt{sample}_3 = [\xi = \frac{30752501854533959}{36028797018963968}, r = 1/2, \omega_0 = 1],$$

$$\mathtt{sample}_4 = [\xi = \frac{30752501854533959}{36028797018963968}, r = 2, \omega_0 = 1].$$

After substituting α by $\mathtt{sample}_1$ in $n(\omega,\,\gamma)$, we obtain

$$n(\omega,\,\gamma) = (\gamma - \frac{1}{4})(\gamma + \frac{1}{4})\omega^4 + (a\gamma^2 + b)\omega^2 + \frac{1}{16}(\gamma - 1)(\gamma + 1),$$

where

$$\begin{cases} a = -\dfrac{19471113219505922191282559655338 23}{51922968585348276285304963292200 96}, \\ b = \dfrac{19471113219505922191282559655338 23}{20769187434139310514121985316880384}. \end{cases}$$

In this case, by using one of the proposed methods in Section 3.1, we obtain the isolating interval of $\gamma_{\max}$

$$\left[\frac{6100687164736347533}{4611686018427387904}, \frac{24402748658945394263}{18446744073709551616}\right],$$

which is the isolating interval of the element of index 8 in the sorted set of the real roots of $R = \mathrm{Res}(\mathrm{n}, \frac{\partial \mathrm{n}}{\partial \omega}, \omega) \in \mathbb{Z}[\gamma]$. Finally we get $[C_1, \mathtt{index}_1 = 8]$ as an element returned in the output list. In order to match this result with Example 46 and by considering the same notations, we can see that the real roots of R are ordered as $\{-\sqrt{X_2}, -1, -r^2, -\sqrt{X_1}, \sqrt{X_1}, r^2, 1, \sqrt{X_2}\}$ over C_1 and the element of index 8 is indeed the L_∞-norm of G as proven in Example 46, Proposition 3.4.

Similarly, after substituting α by $\mathtt{sample}_2$ in $n(\omega, \gamma)$, we see that $\gamma_{\max}$ is of isolating interval

$$\left[\frac{6100687164736347533}{1152921504606846976}, \frac{24402748658945394263}{4611686018427387904}\right],$$

which is also the isolating interval of the element of index 8 in the sorted set of the real roots of $R = \mathrm{Res}(\mathrm{n}, \frac{\partial \mathrm{n}}{\partial \omega}, \omega) \in \mathbb{Z}[\gamma]$. Thus the second element of the output list of Algorithm 6 is $[C_2, \mathtt{index}_2 = 8]$. With the notations of Example 46, the real roots of R are ordered as $\{-\sqrt{X_2}, -r^2, -1, -\sqrt{X_1}, \sqrt{X_1}, 1, r^2, \sqrt{X_2}\}$ over C_2, and we can see that the result matches with Proposition 3.4.

By following the same approach and substituting α by $\mathtt{sample}_3$ in $n(\omega, \gamma)$, we obtain that $\gamma_{\max}$ is of isolating interval $[1, 1]$, which is the isolating interval of the element of index 7 in the sorted set of the real roots of R. The third element of the output list of Algorithm 6 is thus $[C_3, \mathtt{index}_3 = 7]$. Moreover, with the notations of Example 46, we can verify the result of Proposition 3.4 where the real roots of R are ordered as $\{-\sqrt{X_2}, -1, -r^2, -\sqrt{X_1}, \sqrt{X_1}, r^2, 1, \sqrt{X_2}\}$ over C_3.

Finally, by substituting α by $\mathtt{sample}_4$ in $n(\omega, \gamma)$, we obtain that $\gamma_{\max}$ is of isolating interval $[4, 4]$ which is the isolating interval of the element of index 7 in the sorted set of the real roots of R. The fourth element of the output list of Algorithm 6 is thus $[C_4, \mathtt{index}_4 = 7]$. With the notations of Example 46, the real roots of R are ordered over C_4 as $\{-\sqrt{X_2}, -r^2, -1, -\sqrt{X_1}, \sqrt{X_1}, 1, r^2, \sqrt{X_2}\}$, and we can see that the result matches with Proposition 3.4.

Chapter 4

Application

4.1 Implementation and experiments

The three proposed methods for the non-parametric case can be implemented in a few lines of `Maple` but we then have to use implementations at different levels that do not give valuable information about the intrinsic efficiency. For instance, the RUR is implemented in `C` but for general zero dimensional polynomial systems: a variant for bivariate polynomials, the one used for the complexity analysis, is not part of `Maple` and is much more efficient for bivariate systems.

In order to have fair comparisons, we extract the dominating operations and compare them using exactly the same implementations. Namely, resultant computations of sheared/non sheared systems and Root Isolation carry the largest percentage of the computation time. For instance, Algorithm 5 saves time on the resultant computation since it does not perform any shear while it loses time on the root isolation.

For the three methods, the principle subresultant sequence is computed using the routine `SubResultantChain` of the `Maple` package `RegularChain`.

Isolating the real roots of univariate polynomials is another common basic block shared between the three algorithms for which we use `Isolate` provided by the `Maple routine` package `RootFinding`.

In Table 4.1, we list the main steps of the three algorithms. The check marks mean that the step makes part of the method and the double check marks indicate that this step is the bottleneck of the method. Note that *Res1* stands for the resultant of the original system and *Res2* for the resultant of the sheared system. Moreover, `Hinf_RUR` stands for the RUR method, `Hinf_Sep` stands for the separation method and `Hinf_Sres` stands for the Sturm-Habicht method. Keep in mind that the shear done in `Hinf_RUR` is different from the

	Res1 + Iso	Res2 + Iso	List of signs
`Hinf_RUR`	✓	✓✓	
`Hinf_Srep`	✓	✓✓	
`Hinf_Sres`	✓		✓✓

Table 4.1: Main steps considered in the implementation of the proposed method.

α	N	`Hinf_RUR`	`Hinf_Sep`	`Hinf_Sres`
2	2	0.2	3	0.2
	3	0.5	7	0.5
	4	2.5	25	2
	5	10	83	6
	6	37	96	10
	7	50	186	47.5
	8	133.5	353	59
	9	236	394	130

Table 4.2: Timings for L^∞-norm for random matrices valued functions with $\tau_G = 2$.

one done in `Hinf_Sep`. Finally, *Iso* means `Isolate`.

In Table 4.2, we report the average running time in CPU seconds of the marked steps listed in Table 4.1 for the three proposed algorithms ran on square matrices of size α, with entries given by random proper rational functions of degree N (degree of the denominators)[1].

These random matrices are of a fixed input coefficient bitsize $\tau = 2$, i.e., the rational functions constituting the entries of the matrices F have coefficients of magnitude $\mathcal{O}(2^\tau)$.

We finally mention that with these experiments, our goal is not to illustrate the theoretical complexity but, on the contrary, to show that on practical examples, the results are different from theory. In theory, the RUR algorithm might asymptotically be the fastest while in practice the Sturm-Habicht method performs better.

[1]The experiments were conducted on Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz 2.90 GHz, Installed RAM 8.00 GB under a Windows platform

4.2 Some numerical method drawbacks

Error in [77, Proposition 5.2]: In Example 46 we have computed the L^∞-norm of the following transfer function

$$G = \frac{\left(\frac{s}{\omega_0}\right)^2 + 2\,\xi\left(\frac{s}{\omega_0}\right) + 1}{\left(\frac{s}{\omega_1}\right)^2 + 2\,\xi\left(\frac{s}{\omega_1}\right) + 1},$$

which depends on three parameters $\xi, \omega_0, \omega_1 \in \mathbb{R}_{>0}$, $\omega_0 \neq \omega_1$ and $0 < \xi \leq 1$. This computation was done manually by using `Maple` tools and we obtained in Proposition 3.3 that the L^∞-norm of G is given by:

$$\| G \|_\infty = \begin{cases} \max\{1, r^2\} & \text{if } \xi \geq \frac{1}{\sqrt{2}}, \\ \delta & \text{if } \xi < \frac{1}{\sqrt{2}}, \end{cases}$$

where $r = \omega_1/\omega_0$, $\mu = 4\,\xi^2\,(\xi - 1)\,(\xi + 1)$ and δ is the maximal real root of:

$$M = \mu\,\gamma^4 + \left((r^2 - 1)^2 - 2\,\mu\,r^2\right)\gamma^2 + \mu\,r^4 \in \mathbb{R}[\gamma].$$

Then, we verified this result by applying Algorithm 6 in Example 47.

This norm was already computed in [77, Proposition 5.2], where the author has obtained a result with an error. In this mentioned proposition, with the notation $\lambda = (r^2 + 1)^2 - 4\,r^2\,(2\,\xi^2 - 1)^2$, the L^∞-norm of G was given by:

$$\| G \|_\infty = \begin{cases} 1 & \text{if } \xi \geq \frac{1}{\sqrt{2}},\ r < 1, \\ r^2 & \text{if } \xi \geq \frac{1}{\sqrt{2}},\ r > 1, \\ r\sqrt{\dfrac{1 - r^2 + \sqrt{\lambda}}{r^2 - 1 + \sqrt{\lambda}}} & \text{if } \xi < \frac{1}{\sqrt{2}},\ r < 1, \\ r\sqrt{\dfrac{r^2 - 1 + \sqrt{\lambda}}{1 - r^2 + \sqrt{\lambda}}} & \text{if } \xi < \frac{1}{\sqrt{2}},\ r > 1. \end{cases} \tag{4.1}$$

The error of this result can be checked by considering the numerical values:

$$\omega_0 = 1, \quad \omega_1 = 50, \quad \xi = 0.01. \tag{4.2}$$

By substituting the values of (4.2) into (4.1), we obtain:

$$\| G \|_\infty = 124975.0075.$$

However, by substituting the values of (4.2) into (3.5) of Proposition 3.3, we get $\| G \|_\infty = 124956.2680$. We can notice the difference between the results. Using one of the `Maple` implemented methods of the algorithms proposed in Section 3.1, i.e., either `Hinf_RUR` or `Hinf_Sep` or `Hinf_Sres`, we then get

$$\| G \|_\infty \in [124956.2680,\ 124956.2680],$$

that refers to the isolating interval of the searched value. We can clearly see that the result matches with the one of Proposition 3.3 and thus make sure of the error occurring in [77, Proposition 5.2].

Moreover, we can also check this result using the standard numerical method for the L^∞-norm computation based on the bisection algorithm and eigenvalue computation of Hamiltonian matrices (see Section 1.4 and Setion 1.5). This numerical method is implemented in `Maple` under the name `NormHinf` of the package `DynamicSystems`. However, the obtained result is

$$\| G \|_\infty = 124950.134953011.$$

We can also clearly notice the inaccuracy of this numerical method for the computation of the L^∞-norm.

Another drawback of the above numerical method is the instability whenever we chose parameter values that are very close to the discriminant variety of the polynomial system corresponding to the transfer function. For instance, consider again the transfer function

$$G = \frac{\left(\frac{s}{\omega_0}\right)^2 + 2\,\xi\,\left(\frac{s}{\omega_0}\right) + 1}{\left(\frac{s}{\omega_1}\right)^2 + 2\,\xi\,\left(\frac{s}{\omega_1}\right) + 1},$$

and let $\alpha = [r,\, \omega_0,\, \xi] \in \mathbb{R}^3$. The corresponding polynomial system is given by $\Sigma = \left\{n(\omega,\, \gamma),\, \frac{\partial n}{\partial \omega}(\omega,\, \gamma)\right\} \subset \mathbb{Z}[\alpha][\omega, \gamma]$, which has been already defined in Proposition 3.3. Following Example 47, the discriminant variety of the system is the union of the curves defined

by the following polynomials:

$$\left\{r,\, \omega_0,\, \xi,\, r-1,\, r+1,\, \xi-1,\, \xi+1,\, 2\,\xi^2-1,\, -4\,r\,\xi^2+r^2+2\,r+1,\, 4\,r\,\xi^2+r^2-2\,r+1\right\}.$$

Thus, if we consider the following numerical values

$$\omega_0 \;=\; 1, \quad \omega_1 \;=\; 2, \quad \xi \;=\; 10^{-10}, \tag{4.3}$$

then we get the transfer function:

$$G \;=\; \frac{100000000000s^2+2s+100000000000}{25000000000s^2+s+100000000000}.$$

Using the `Maple NormHinf` function for the system corresponding to this transfer function G, the execution is then interrupted on the step of bisection and returns an error (see Figure 4.1) since the system is considered to be unstable and of norm equals to infinity.

```
>
> sys:=TransferFunction(G):
> NormHinf(sys);
Error, (in DynamicSystems:-NormHinf) Hinf norm is infinite. 'sys' eigenvalues on imaginary axis:
{-0.2000000000e-10-2.*I, -0.2000000000e-10+2.*I}
>
```

Figure 4.1

But by substituting the values of (4.3) into (3.5) of Proposition 3.3, we get:

$$\|\, G \,\|_\infty = \; 1.500000000\, 10^{10}.$$

Using one of the algorithms proposed in Section 3.1 and implemented in `Maple`, we obtain

$$\|\, G \,\|_\infty \in \; [1.500000000\, 10^{10},\; 1.500000000\, 10^{10}],$$

that refers to the isolating interval of the searched value. The results match.

In fact, by simplifying the expression of the transfer function G, we get:

$$G \;=\; r^2\, \frac{s^2 \;+\; 2\,\omega_0\,\xi\,s \;+\; \omega_0^2}{s^2 \;+\; 2\,r\,\omega_0\,\xi\,s \;+\; r^2\,\omega_0^2}.$$

Thus, the poles of the transfer function G are the complex roots of the denominator $l \;:=$

$s^2 + 2\,r\,\omega_0\,\xi\,s + r^2\,\omega_0^2 \in \mathbb{R}[s]$. The discriminant of l is:

$$\Delta = 4\,r^2\,\omega_0^2\,(\xi - 1)\,(\xi + 1).$$

Hence, $\Delta < 0$ if and only if $\xi < 1$. In the case where $\Delta < 0$, the two complex solutions of l are defined by $l_\pm := -r\,\omega_0\,\xi \pm \delta\,i$, where $\delta = r\,\omega_0\,\sqrt{(1-\xi)\,(1+\xi)}$. Hence, when ξ tends to 0, the poles of G become close to the imaginary axis and the system tends to be unstable, which represents a problem to the numerical method.

4.3 Practical examples

Spring mass damper system: We aim in this paragraph at computing the H_∞-norm of the transfer function G computed in Example 2 that represents the spring mass damper system discussed in Example 1. Let

$$G = \frac{1}{m\,s^2 + b\,s + k},$$

where $m,\, b,\, k \in \mathbb{R}_{>0}$. We apply Algorithm 6 on the corresponding polynomial system $\left\{n(\omega,\,\gamma),\, \frac{\partial n}{\partial \omega}(\omega,\,\gamma)\right\}$ for

$$n(\omega,\,\gamma) \;=\; -\gamma^2\,m^2\,\omega^4 \;+\; (-b^2 \;+\; 2\,k\,m)\,\gamma^2\,\omega^2 \;-\; \gamma^2\,k^2 + 1,$$

by taking into consideration the inequalities $m > 0$, $b > 0$, $k > 0$. In this case, we get the output

$$\left\{\;[C_1,\,5],\;\;[C_2,\,4],\;\;[C_3,\,3]\;\right\}, \tag{4.4}$$

where

$$C_1 = \Big\{m > 0\Big\} \cap \Big\{k > 0\Big\} \cap \Big\{0 < b < \sqrt{2\,k\,m}\Big\},$$

$$C_2 = \Big\{m > 0\Big\} \cap \Big\{k > 0\Big\} \cap \Big\{\sqrt{2\,k\,m} < b < 2\,\sqrt{k\,m}\Big\},$$

$$C_3 = \Big\{m > 0\Big\} \cap \Big\{k > 0\Big\} \cap \Big\{b > 2\,\sqrt{k\,m}\Big\},$$

and the numbers 5, 4 and 3 refer to the index of the searched value $\parallel G \parallel_\infty$ in the sorted set of the real roots of the polynomial $R = \mathrm{Res}(n, \frac{\partial n}{\partial \omega}, \omega) \in \mathbb{R}[\gamma]$, whenever the parameters take values in the corresponding cell.

For a better visualization of this result, we consider a sample point in each cell C_i. For instance, let $\texttt{sample}_1 = [b = 1, k = 1, m = 1] \in C_1$. In this case, by evaluating the values of $\texttt{sample}_1$ in G, we obtain

$$G_1 = \frac{1}{s^2 + s + 1}, \quad n_1(\omega,\, \gamma) = -\gamma^2\, \omega^4 \;+\; \gamma^2\, \omega^2 \;-\; \gamma^2 + 1,$$

and the sorted set of the isolating intervals of the real roots of $\mathrm{Res}(n_1, \frac{\partial n_1}{\partial \omega}, \omega)$ using the command `Isolate` on `Maple` is given by

$$\Big\{[-a, -b],\ [-1, -1],\ [0, 0],\ [1, 1],\ [c, d]\Big\}, \tag{4.5}$$

such that when converted from `rational` to `float`, the values

$$a \approx b \approx c \approx d \approx 1.154700538.$$

Now based on (4.4), we can say that $\parallel G_1 \parallel_\infty \in [c, d]$, which is the 5th element of (4.5). We visualise the curve of $n_1(\omega,\, \gamma)$ in order to see the critical points and their maximal γ-projection in Figure 4.2.

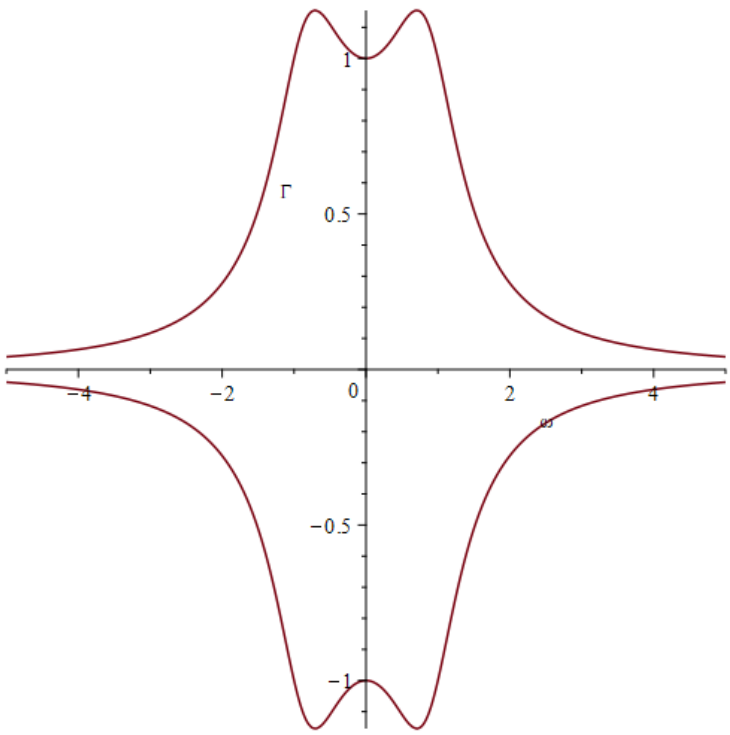

Figure 4.2: Plot of $n_1(\omega,\, \gamma) = 0$, where ω/γ is in the horizontal/vertical axis

We can also visualize in Figure 4.3 the Bode magnitude plot of G_1 where $\| G_1 \|_\infty$ equals the peak value.

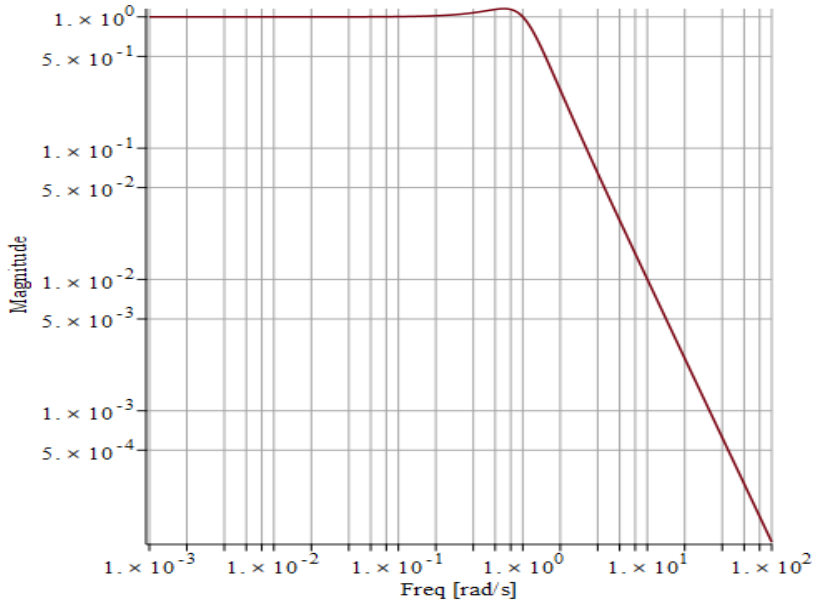

Figure 4.3: Bode plot of G_1

Let

$$\texttt{sample}_2 = [b = \frac{30752501854533959}{18014398509481984}, k = 1, m = 1] \in C_2.$$

In this case, by evaluating the values of $\texttt{sample}_2$ in G, we obtain

$$G_2 = \frac{1}{s^2 + \dfrac{30752501854533959}{18014398509481984}\, s + 1},$$

and

$$n_2(\omega,\, \gamma) = -\gamma^2\, \omega^4 \;\; + \alpha\, \gamma^2\, \omega^2 \;\; - \;\; \gamma^2 + 1,$$

for

$$\alpha = \frac{29667926299626113402489304306116 9}{324518553658426726783156020576256}.$$

The sorted set of the isolating intervals of the real roots of $\mathrm{Res}(n_2, \frac{\partial n_2}{\partial \omega}, \omega)$ using the command `Isolate` on `Maple` is given by

$$\Big\{[-a, -b],\ [-1, -1],\ [0, 0],\ [1, 1],\ [c, d]\Big\}, \tag{4.6}$$

such that when converted from `rational` to `float`, the values

$$a \approx b \approx c \approx d \approx 1.124338551.$$

Now based on (4.4), we can say that $\| G_1 \|_\infty \in [1, 1]$, which is the 4th element of (4.6). We visualise the curve of $n_2(\omega, \gamma)$ in order to see the critical points and their maximal γ-projection in Figure 4.4.

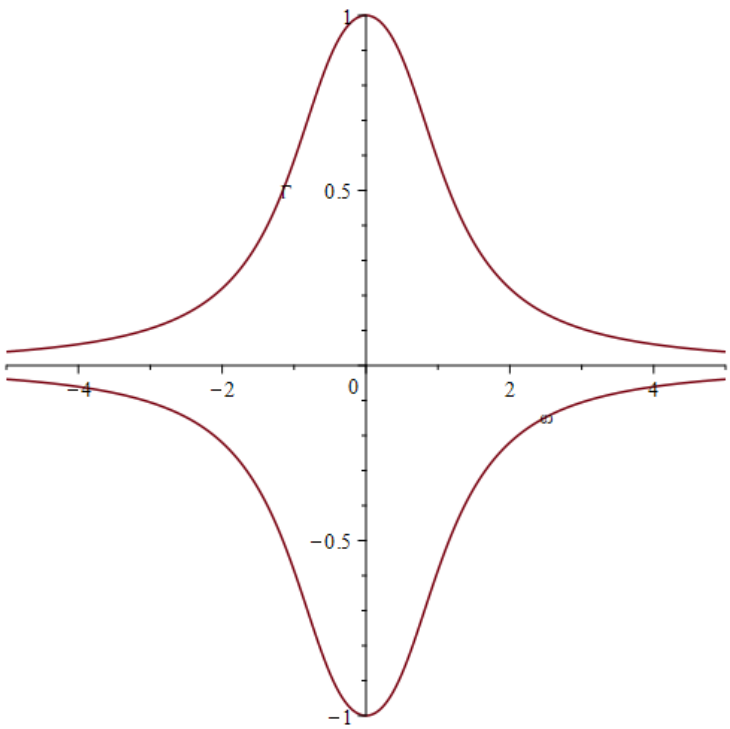

Figure 4.4: Plot of $n_2(\omega, \gamma) = 0$, where ω/γ is in the horizontal/vertical axis

We can also visualize in Figure 4.5 the Bode magnitude plot of G_2 where $\| G_2 \|_\infty$ equals the peak value.

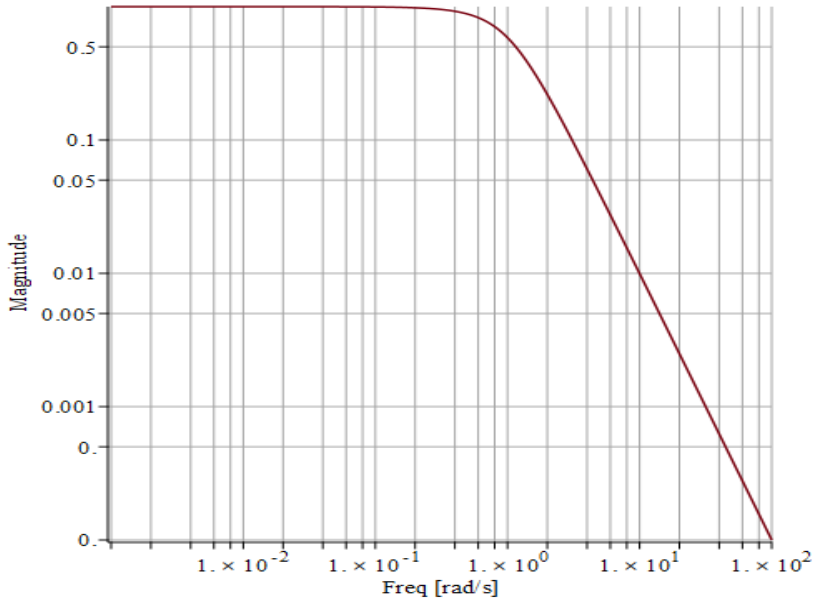

Figure 4.5: Bode plot of G_2

Similarly, let

$$\texttt{sample}_3 = [b = 3, k = 1, m = 1] \in C_3.$$

In this case, by evaluating the values of $\texttt{sample}_3$ in G, we obtain

$$G_3 = \frac{1}{s^2 + 3\,s + 1},$$

and

$$n_3(\omega,\, \gamma) = -\gamma^2\,\omega^4 \;\; - 7\,\gamma^2\,\omega^2 \;\; - \;\; \gamma^2 + 1.$$

The sorted set of the isolating intervals of the real roots of $\mathrm{Res}(n_3,\, \frac{\partial n_3}{\partial \omega}, \omega)$ using the command `Isolate` on `Maple` is given by

$$\Big\{[-1, -1],\; [0, 0],\; [1, 1]\Big\}, \tag{4.7}$$

Based on (4.4), we can say that $\parallel\, G_1\, \parallel_\infty \in\, [1, 1]$, which is the 3rd element of (4.7). We visualise the curve of $n_3(\omega,\, \gamma)$ in order to see the critical points and their maximal γ-projection in Figure 4.6.

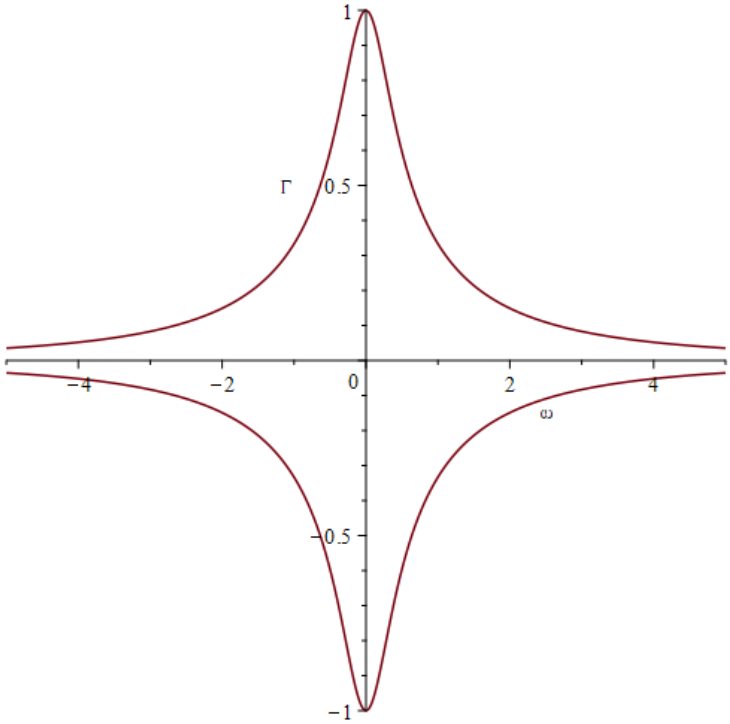

Figure 4.6: Plot of $n_3(\omega, \gamma) = 0$, where ω/γ is in the horizontal/vertical axis

We can also visualize in Figure 4.7 the Bode magnitude plot of G_3 where $\| G_3 \|_\infty$ equals the peak value.

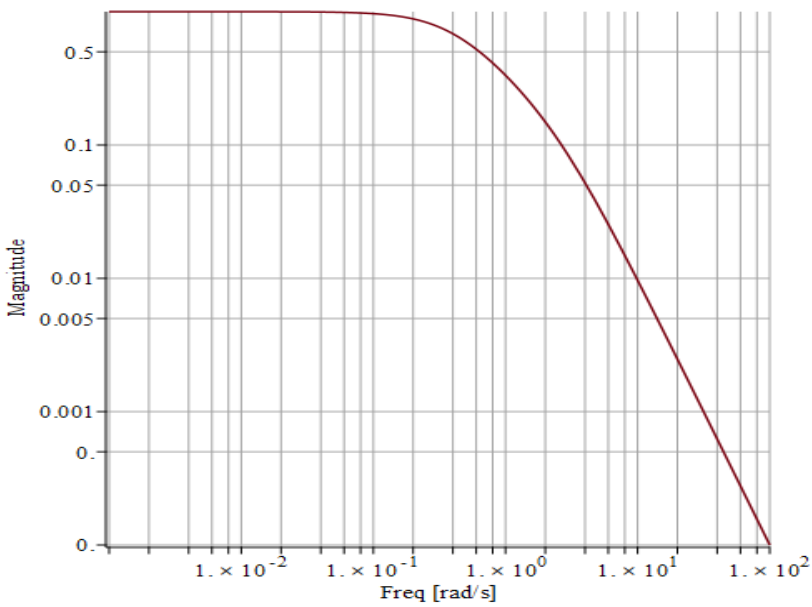

Figure 4.7: Bode plot of G_3

Example with more parameters In this example, we consider the servo-control of a mechanical axis of a sight, characterized by an inertia J. This axis is driven by a motor. R designates the resistance of the motor, L its inductance, and k_e its electrical constant. The motor current is measured along with the speed of the line of sight. This latter is measured using a gyro-meter. The transfer function of the system is given by

$$G(s) := \begin{pmatrix} \dfrac{1}{J\ s}\ K_e\, E(s)\, M(s) \\ E(s) \end{pmatrix},$$

where $E(s)$ is the electric transfer function and is given by

$$E(s) = \frac{1}{R + L\, s},$$

and $M(s)$ is the transfer function of a mechanical mode of resonance pulsation ω_d (of anti-resonance ω_n) given by

$$M(s) := \frac{1 + 2\ \xi_n\ \dfrac{s}{\omega_n} + \left(\dfrac{s}{\omega_n}\right)^2}{1 + 2\ \xi_d\ \dfrac{s}{\omega_d} + \left(\dfrac{s}{\omega_d}\right)^2}.$$

After simplifying the expression of $G(s)$ and substituting K_e/J by the parameter α, we get

$$G(s) = \begin{pmatrix} \dfrac{\alpha\ s\ \left(s^2\ +\ 2\ s\ \omega_n\ \xi_n\ +\ \omega_n^2\right)\ \omega_d^2}{\left(L\ s + R\right)\ \left(s^2\ +\ 2\ s\ \omega_d\ \xi_d\ +\ \omega_d^2\right)\ \omega_n^2} \\ \dfrac{1}{L\ s + R} \end{pmatrix}.$$

We can apply Algorithm 6 for computing or representing the L^∞-norm of the transfer matrix G as a function in the parameter set $\{\alpha,\ \omega_n,\ \xi_n,\ \omega_d,\ \xi_d,\ L,\ R\} \subset \mathbb{R}_{>0}$. However, this example is considered to be a huge one and exhausting to the implemented function for executing it in the presence of all the seven parameters. Nevertheless, for some simplified models that remain interesting in practice, it is possible for the algorithm to execute with up to five parameters in some cases. For instance, we consider in the first place the substitution $L = 0$ in the transfer matrix G. In this case, Algorithm 6 has successfully executed with the following substitutions in the matrix G:

$$\{L = 0,\ \xi_d = 0\},\ \{L = 0,\ \xi_d = 1\},\ \{L = 0,\ \omega_d = 1\},$$
$$\{L = 0,\ \xi_n = 1\} \text{ and } \{L = 0,\ \xi_n = 0\},$$

where the transfer matrix G is left with five parameters. Whereas the cases that failed to execute with five parameters are the cases with the following parameter substitutions

$$\{L = 0,\ R = 1\},\ \{L = 0, \omega_n = 1\} \text{ and } \{L = 0, \alpha = 1\}. \tag{4.8}$$

Thus, for the failed cases in (4.8), we try to execute with four parameters, by substituting one more parameter with a numerical value such as

$$R = 1 \text{ or } \alpha = 1 \text{ or } \xi_d = 0 \text{ or } \xi_d = 1 \text{ or } \omega_d = 1 \text{ or } \omega_n = 1 \text{ or } \xi_n = 0 \text{ or } \xi_n = 1.$$

All the cases were executed successfully except for the substitutions

$$\{L = 0,\ R = 1,\ \omega_n = 1\},\ \{L = 0,\ R = 1,\ \alpha = 1\} \text{ and } \{L = 0,\ \alpha = 1,\ \omega_n = 1\}.$$

For these cases, we again substitute one more parameter with a numerical value in order to obtain a system depending on three parameters. After this substitution, Algorithm 6 executed successfully except for the case

$$\{L = 0,\ \alpha = 1,\ \omega_n = 1,\ R = 1\},$$

i.e., when the system is left with the parameters ξ_d, ξ_n and ω_d. This last case can however be executed by substituting one more parameter and being left with a transfer matrix depending on two parameters only.

With this said, we consider for instance the example

$$G_5 = \begin{pmatrix} \dfrac{\alpha\ s\ \left(s^2\ +\ 2\ s\ \omega_n\ \xi_n\ +\ \omega_n^2\right)\ \omega_d^2}{R\ \left(s^2\ +\ \omega_d^2\right)\ \omega_n^2} \\ \dfrac{1}{R} \end{pmatrix}$$

depending on the five parameters $\{\alpha, \omega_n, \xi_n, \omega_d, R\} \subset \mathbb{R}_{>0}$, which is obtained after the substitution $\{L = 0,\ \xi_d = 0\}$ in G. The corresponding polynomial system $\left\{n_5(\omega,\ \gamma), \frac{\partial n_5}{\partial \omega}(\omega,\ \gamma)\right\} \subset \mathbb{Q}[R, \alpha, \omega_n, \xi_n, \omega_d][\omega, \gamma]$ is such that

$$n_5(\omega,\ \gamma) = \alpha^2 \omega_d^4\ \omega^6 - \omega_n^2 f_5\ \omega^4 + g_5\ \omega_d^2\ \omega_n^4\ \omega^2 - h_5,$$

where

$$\begin{cases} f_5 = R^2\ \omega_n^2\ \gamma^2 - 2\ \omega_d^4\ \left(2\ \xi_n^2 - 1\right)\ \alpha^2 - \omega_n^2, \\ g_5 = 2\ R^2\ \gamma^2 + \alpha^2\ \omega_d^2 - 2, \\ h_5 = \left(R\ \gamma - 1\right)\ \left(R\ \gamma + 1\right)\ \omega_d^4\ \omega_n^4, \end{cases}$$

$\deg(n_5) = 14$ and $\deg_\omega(n_5) = 6$.

In this case, the output of Algorithm 6 applied to $\left\{n_5(\omega,\ \gamma), \frac{\partial n_5}{\partial \omega}(\omega,\ \gamma)\right\}$ along with $\left\{\alpha > 0,\ \omega_n > 0,\ \xi_n > 0,\ \omega_d > 0,\ R > 0\right\}$ using `Maple` functions is

$$\left\{ \begin{array}{c} [C_1,\ 7],\ \ [C_2,\ 5],\ \ [C_3,\ 3]\ \ [C_4,\ 3],\ \ [C_5,\ 5],\ \ [C_6,\ 7]\ \ [C_7,\ 5], \\ [C_8,\ 3]\ \ [C_9,\ 3],\ \ [C_{10},\ 7],\ \ [C_{11},\ 5],\ \ [C_{12},\ 3]\ \ [C_{13},\ 6],\ \ [C_{14},\ 4] \end{array} \right\},$$

such that $\{C_1, \ldots, C_{14}\}$ is the corresponding cell decomposition of the parameter space.

We tackle another example where the transfer matrix should depend on four parameters and we consider the substitution $\{L = 0,\ \omega_n = 1,\ \xi_n = 0\}$ in G. In this case, we obtain the transfer matrix

$$G_4 = \begin{pmatrix} \dfrac{\alpha\ s\ \left(s^2 + 1\right)\ \omega_d^2}{R\ \left(s^2\ +\ 2\ \omega_d\ \xi_d\ s + \omega_d^2\right)} \\ \dfrac{1}{R} \end{pmatrix}$$

depending on five parameters $\{\alpha,\ \omega_d,\ \xi_d,\ R\} \subset \mathbb{R}_{>0}$. The corresponding polynomial system $\left\{n_4(\omega,\ \gamma), \frac{\partial n_4}{\partial \omega}(\omega,\ \gamma)\right\} \subset \mathbb{Q}[R, \alpha, \omega_d, \xi_d][\omega,\ \gamma]$ is such that

$$n_4(\omega,\ \gamma) = \alpha^2 \omega_d^4\ \omega^6 - f_4\ \omega^4 - g_4\ \omega_d^2\ \omega^2 - h_4,$$

where

$$\begin{cases} f_4 = R^2\ \gamma^2 + 2\ \omega_d^4\ \alpha^2 - 1, \\ g_4 = \left(4\,\xi_d^2 - 2\right)\ R^2\ \gamma^2 - \alpha^2\ \omega_d^2 - 4\ \xi_d^2 + 2, \\ h_4 = \left(R\ \gamma - 1\right)\ \left(R\ \gamma + 1\right)\ \omega_d^4, \end{cases}$$

$\deg(n_4) = 12$ and $\deg_\omega(n_4) = 6$.

In this case, the output of Algorithm 6 applied to $\left\{n_4(\omega,\ \gamma), \frac{\partial n_4}{\partial \omega}(\omega,\ \gamma)\right\}$ along with

$\{\alpha > 0,\ \xi_d > 0,\ \omega_d > 0,\ R > 0\}$ using `Maple` functions is

$$\left\{ \begin{array}{l} [C_1,\, 8],\ [C_2,\, 6],\ [C_3,\, 4]\ [C_4,\, 4],\ [C_5,\, 8],\ [C_6,\, 8]\ [C_7,\, 6], \\ [C_8,\, 4]\ [C_9,\, 4],\ [C_{10},\, 8],\ [C_{11},\, 6],\ [C_{12},\, 4]\ [C_{13},\, 8],\ [C_{14},\, 6] \end{array} \right\},$$

such that $\{C_1, \ldots, C_{14}\}$ is the corresponding cell decomposition of the parameter space.

We tackle an example where the transfer matrix needs to be depending on three parameters only and we consider the substitution

$$\{L = 0,\ \omega_n = 1,\ R = 1,\ \xi_d = 1\}$$

in G. In this case, we obtain the transfer matrix

$$G_3 = \begin{pmatrix} \dfrac{\alpha\ s\ \left(s^2 + 2\ \xi_n\ s + 1\right)\ \omega_d^2}{s^2\ +\ 2\ \omega_d\ s + \omega_d^2} \\ 1 \end{pmatrix}$$

depending on three parameters $\{\omega_d,\, \alpha,\, \xi_n\} \subset \mathbb{R}_{>0}$. The corresponding polynomial system $\left\{ n_3(\omega, \gamma), \frac{\partial n_3}{\partial \omega}(\omega, \gamma) \right\} \subset \mathbb{Q}[\omega_d, \alpha, \xi_n][\omega, \gamma]$ is such that

$$n_3(\omega,\ \gamma) = \alpha^2 \omega_d^4\ \omega^6 - f_3\ \omega^4 - g_3\ \omega_d^2\ \omega^2 - h_3,$$

where

$$\left\{ \begin{array}{l} f_3 = \gamma^2 - 4\big(\xi_n^2 - \frac{1}{2}\big)\ \omega_d^4\ \alpha^2 - 1, \\ g_3 = 2\ \gamma^2 - \alpha^2\ \omega_d^2 - 2, \\ h_3 = \big(\gamma - 1\big)\ \big(\gamma + 1\big)\ \omega_d^4, \end{array} \right.$$

$\deg(n_3) = 12$ and $\deg_\omega(n_3) = 6$.

In this case, the output of Algorithm 6 applied to $\left\{ n_3(\omega,\ \gamma), \frac{\partial n_3}{\partial \omega}(\omega,\ \gamma) \right\}$ along with $\{\alpha > 0,\ \omega_d > 0,\ \xi_n > 0\}$ using `Maple` functions is

$$\left\{ \begin{array}{l} [C_1,\, 8],\ [C_2,\, 6],\ [C_3,\, 8]\ [C_4,\, 6],\ [C_5,\, 4],\ [C_6,\, 2]\ [C_7,\, 8],\ [C_8,\, 6],\ [C_9,\, 4], \\ [C_{10},\, 2],\ [C_{11},\, 8],\ [C_{12},\, 6]\ [C_{13},\, 4],\ [C_{14},\, 2]\ [C_{15},\, 8],\ [C_{16},\, 4],\ [C_{17},\, 4] \\ [C_{18},\, 2],\ [C_{19},\, 8],\ [C_{20},\, 6],\ [C_{21},\, 4]\ [C_{22},\, 2],\ [C_{23},\, 8],\ [C_{24},\, 6]\ [C_{25},\, 4] \end{array} \right\},$$

such that $\{C_1, \ldots, C_{25}\}$ is the corresponding cell decomposition of the parameter space.

We study finally an example where the transfer matrix needs to be depending on two parameters only and we consider the substitution

$$\{L = 0,\ \omega_n = 1,\ R = 1,\ \alpha = 1, \xi_n = 1\}$$

in G. In this case, we obtain the transfer matrix

$$G_2 = \begin{pmatrix} \dfrac{s\ \left(s^2 + 2\ s + 1\right)\ \omega_d^2}{s^2\ +\ 2\ \omega_d\ \xi_d\ s + \omega_d^2} \\ 1 \end{pmatrix}$$

depending on three parameters $\{\omega_d,\ \xi_d\} \subset \mathbb{R}_{>0}$. The corresponding polynomial system $\left\{n_2(\omega,\ \gamma), \frac{\partial n_2}{\partial \omega}(\omega,\ \gamma)\right\} \subset \mathbb{Q}[\omega_d, \xi_d][\omega, \gamma]$ is such that

$$n_2(\omega,\ \gamma) = \omega_d^4\ \omega^6 - f_3\ \omega^4 - g_3\ \omega_d^2\ \omega^2 - h_3,$$

where

$$\begin{cases} f_3 = \gamma^2 - 2\ \omega_d^4 - 1, \\ g_3 = \left(4\ \xi_d^2 - 2\right)\ \gamma^2 - \omega_d^2 - \left(4\ \xi_d^2 - 2\right), \\ h_3 = \left(\gamma - 1\right)\ \left(\gamma + 1\right)\ \omega_d^4, \end{cases}$$

$\deg(n_2) = 10$ and $\deg_\omega(n_2) = 6$.

In this case, the output of Algorithm 6 applied to $\left\{n_2(\omega,\ \gamma), \frac{\partial n_2}{\partial \omega}(\omega,\ \gamma)\right\}$ along with $\left\{\omega_d > 0,\ \xi_d > 0\right\}$ using `Maple` functions is

$$\left\{ \begin{array}{c} [C_1,\ 8],\ [C_2,\ 8],\ [C_3,\ 8]\ [C_4,\ 8],\ [C_5,\ 4],\ [C_6,\ 8]\ [C_7,\ 4],\ [C_8,\ 4], \\ [C_9,\ 8],\ [C_{10},\ 4],\ [C_{11},\ 2],\ [C_{12},\ 4]\ [C_{13},\ 4],\ [C_{14},\ 8]\ [C_{15},\ 6],\ [C_{16},\ 2], \\ [C_{17},\ 4],\ [C_{18},\ 4],\ [C_{19},\ 8],\ [C_{20},\ 4],\ [C_{21},\ 4]\ [C_{22},\ 2],\ [C_{23},\ 6],\ [C_{24},\ 6] \end{array} \right\},$$

such that $\{C_1, \ldots, C_{24}\}$ is the corresponding cell decomposition of the parameter space.

Chapter 5

Conclusion

In this dissertation, we were interested in computing the L^∞-norm of finite-dimensional linear time-invariant systems represented by their transfer matrix. After defining this norm and studying its properties, we have modeled this problem to a problem of computing the maximal y-projection of the real solutions $(x,\ y)$ of a bivariate polynomial system $\Sigma = \{P, \frac{\partial P}{\partial x}\}$, with $P \in \mathbb{R}[x, y]$. Two cases then arose. The first one was when the system does not depend on parameters, i.e., when $P \in \mathbb{Z}[x, y]$.

In this case, for computing the maximal y-projection of the system real solutions, we have proposed three different symbolic-numeric algorithms. The first one is called the *RUR method*, named after the famous Rational Univariate Representation algorithm for solving bivariate polynomial systems, and it mainly consists in first computing a separating linear form $t = y + s\,x$ to shear the original system into a generic one, using the shearing map $(x,\ y) \mapsto (x,\ t - s\,x)$, so that no two solutions are horizontally aligned. Then, by computing a rational univariate representation, it represents the variables x, y as rational functions in one variable t. Using a one-to-one correspondence between the real roots of a univariate polynomial $f \in \mathbb{Q}[t]$ and the real solutions of the polynomial system, the problem is reduced to isolating the univariate polynomial f. After some interval evaluation, this approach leads to obtaining isolating boxes of the real solutions of the system.

We applied this algorithm to our polynomial system and then managed to choose the isolating interval of the maximal y-projection of the system real solutions, after refining the y-intervals up to the separation bound of a particular polynomial. More precisely, this polynomial is univariate in y such that it embodies the y-projection of the system solutions, called the resultant polynomial of P and $\frac{\partial P}{\partial x}$ with respect to the variable x, written as $\mathrm{Res}\big(P, \frac{\partial P}{\partial x}, x\big) \in \mathbb{Q}[y]$. We have computed the worst-case bit complexity of this algorithm in terms of the degrees of P with respect to both variables x and y, to obtain

$\tilde{\mathcal{O}}_B\big(d_y\, d_x^3\, (d_y^2 + d_x\, d_y + d_x\, \tau)\big)$ bit operations in the worst-case, and then implemented it using `Maple` tools.

This RUR method is systematic, meaning that whenever it is called, it returns the whole system solutions in which we have to pick the isolating interval of the maximal y-projection. In the second proposed method, the *Roots Separation method*, we focused only on the maximal y-projection of the system solutions. This was done by first projecting the system solutions to the y-axis, simply by computing the real roots, represented by their isolating intervals, of the univariate resultant polynomial $\mathrm{Res}\big(P, \frac{\partial P}{\partial x}, x\big) \in \mathbb{Q}[y]$. The next step was to verify between the y-projections which one is the best candidate to be searched value. For this purpose, we used a special separating linear map that puts the system in a generic position. What made this linear map special was that it enables us to locate the searched value simply by localizing the maximal y-projection of the real solutions of the sheared system. This was also computed by isolating the resultant polynomial of the sheared polynomials with respect to the variable x. However, this method came with a high cost since the slope of the chosen shearing linear map was of a very large size and led to a large coefficient bitsize of the sheared polynomials. This high coefficient bitsize was carried along in the computation of the resultant polynomial of the sheared polynomials with respect to x and in isolating its real roots. This isolation was the bottleneck of this algorithm and was of worst-case bit complexity $\tilde{\mathcal{O}}_B(d_x^5\, d_y^4\, \tau)$.

We finally proposed a third method named *Sturm-Habicht method* that also focused only on the y-projections of the system solutions in order to locate the maximal one that corresponds to a real solution. Differently from the previous methods, this method did not use any shear and used instead the real root counting principle. This was done to verify the existence of a real root for the gcd polynomial of the system polynomials over the maximal y-projection $\bar{y} \in \mathbb{R}$. Then, to reduce the cost of counting the number of real roots of the gcd polynomial, we took advantage of the fact that the curve $P = 0$ is bounded on the y-direction by the searched value. Thus, the problem of real root counting was then applied to the polynomial P over the algebraic value $\bar{y}$. For this purpose, we used the Sturm-Habiht sequence corresponding to the polynomial P seen as a univariate polynomial in x, which is a signed subresultant sequence of $P \in \mathbb{Z}[y][x]$ and its derivative. The reason for using this sequence was mainly the fact that it is stable under specialization. This property allowed us to count the number of real roots of the univariate polynomial $P(x, \bar{y})$ by evaluating the sequence of principle subresultants over $\bar{y}$ and studying the signs sequence. This evaluation was the bottleneck of this algorithm and was of worst-case bit complexity $\tilde{\mathcal{O}}_B\big(d_x^4\, d_y^2\, (d_y + \tau)\big)$.

Additionally, we saw that when the real curve $P = 0$ does not show any isolated

singular point, the evaluation can be done over a rational value between the largest algebraic y-projections. This has slightly reduced the worst-case bit complexity to $\tilde{\mathcal{O}}_B(d_x^4\, d_y^2\, \tau)$.

We then implemented the three methods using `Maple` tools to compare their practical complexity. We have concluded that since the Stum-Habicht method is adaptive, it has the best average complexity, as seen in the experiments.

Finally, we generalised the proposed approach to the case where $P \in \mathbb{Z}[\alpha][x, y]$ depends on a parameters set $\alpha = [\alpha_1, \ldots, \alpha_d] \in \mathbb{R}^d$. In this case, we have seen that structure of the system solutions depends on the parameters. Moreover, the y-projection of the system solutions were represented by continuous functions in the parameters, where the position of their curves also varies with respect to the parameters values.

Thus, in our proposed algorithm, we have excluded from the parameters space the "bad" parameter values where any two curves of the functions representing the y-projection of the solutions "collapse". Then, using a cylindrical algebraic decomposition, we have decomposed the space of "good" parameter values into connected open sets, named cells, where the y-curves are well positioned. This allowed us to choose a rational sample point in each cell, then apply an algorithm from the non-parametric approach to obtain the index $\texttt{index}_\texttt{i}$ of the searched value in the ordered set of the real roots of $\mathrm{Res}\big(P, \frac{\partial P}{\partial x}, x\big) \in \mathbb{Q}[y]$. Hence, above each cell, we could assure that the searched value is represented by the $\texttt{index}_\texttt{i}{}^{\text{th}}$ y-curve.

Bibliography

[1] M. E. Alonso et al. "Zeros, multiplicities, and idempotents for zero-dimensional systems". In: *Algorithms in algebraic geometry and applications (Santander, 1994)*.

[2] H. Anai. "A symbolic-numeric approach to multi-parametric programming for control design". In: *2009 ICCAS-SICE*. IEEE. 2009, pp. 3525–3530.

[3] D. S. Arnon, G. E. Collins, and S. McCallum. "Cylindrical algebraic decomposition I: The basic algorithm". In: *SIAM Journal on Computing* 13.4 (1984), pp. 865–877.

[4] P. Aubry, D. Lazard, and M. M. Maza. "On the theories of triangular sets". In: *Journal of Symbolic Computation* 28.1-2 (1999), pp. 105–124.

[5] A. Avan den Boom et al. "SLICOT, A Subroutine Library in Control and Systems Theory". In: *IFAC Symposium on Computer Aided Design in Control Systems*. Vol. 24. 1991, pp. 71–76.

[6] S. Basu, R. Pollack, and M. F. Roy. *Algorithms in Real Algebraic Geometry*. Springer Verlag, 2006.

[7] S. Basu, R. Pollack, and M. F. Roy. *Existential theory of the reals, volume 10 of algorithms and computation in mathematics*. 2006.

[8] M. N. Belur and C. Praagman. "An Efficient Algorithm for Computing the $\mathcal{H}_\infty$-norm". In: *IEEE Transactions on automatic control* 56.7 (2011), pp. 1656–1660.

[9] P. Benner, R. Byers, and E. Barth. "Algorithm 800: Fortran 77 subroutines for computing the eigenvalues of Hamiltonian matrices I: The square-reduced method". In: *ACM Transactions on Mathematical Software* 26.1 (2000), pp. 49–77.

[10] P. Benner and T. Mitchell. "Faster and More Accurate Computation of the H_∞ Norm via Optimization". In: *SIAM Journal on Scientific Computing* 40.5 (2018), pp. 3609–3635.

[11] P. Benner, V. Sima, and M. Voigt. "L_∞-Norm Computation for Continuous-Time Descriptor Systems Using Structured Matrix Pencils". In: *IEEE Transactions on Automatic Control* 57.1 (2012), pp. 233–238.

[12] P. Benner, V. Sima, and M. Voigt. "Robust and Efficient Algorithms for $\mathcal{L}_\infty$-norm Computation for Descriptor Systems". In: *IFAC Proceedings Volumes* 45.13 (2012), pp. 195–200.

[13] E. Berberich, P. Emeliyanenko, and M. Sagraloff. "An elimination method for solving bivariate polynomial systems: Eliminating the usual drawbacks". In: *2011 Proceedings of the Thirteenth Workshop on Algorithm Engineering and Experiments (ALENEX)*. SIAM. 2011, pp. 35–47.

[14] M. Bizzarri and M. Lávička. "A symbolic-numerical approach to approximate parameterizations of space curves using graphs of critical points". In: *Journal of Computational and Applied Mathematics* 242 (2013), pp. 107–124.

[15] Y. Bouzidi et al. "Improved algorithm for computing separating linear forms for bivariate systems". In: *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*. 2014, pp. 75–82.

[16] Y. Bouzidi et al. "Separating linear forms and rational univariate representations of bivariate systems". In: *Journal of Symbolic Computation* 68 (2015), pp. 84–119.

[17] Y. Bouzidi et al. "Solving bivariate systems using rational univariate representations". In: *Journal of Complexity* 37 (2016), pp. 34–75.

[18] S. Boyd and V. Balakrishnan. "A regularity result for the singular values of a transfer matrix and a quadratically convergent algorithm for computing its L_∞-norm". In: *Systems and Control Letters* 15 (1990), pp. 1–7.

[19] S. Boyd, V. Balakrishnan, and P. Kabamba. "A bisection method for computing the H_∞ norm of a transfer matrix and related problems". In: *Math. Control, Signals, and Systems* 2 (1989), pp. 207–220.

[20] R. W. Brocket. *Finite Dimensional Linear Systems*. 74. SIAM, 2015.

[21] C. W. Brown. "Improved projection for cylindrical algebraic decomposition". In: *Journal of Symbolic Computation* 32.5 (2001), pp. 447–465.

[22] N. A. Bruinsma and M. Steinbucha. "A fast algorithm to compute compute the H_∞-norm of a transfer function matrix". In: *Systems and Control Letters* 14 (1990), pp. 287–293.

[23] B. F. Caviness. *Jeremy. R. Johnson, editors. Quantifier Elimination and Cylindrical Algebraic Decomposition.* 1998.

[24] Changbo Chen, Marc Moreno Mazza, and Yuzhen Xie. "Computing the Supremum of the Real Roots of a Parametric Univariate Polynomial". In: (2013).

[25] J. S. Cheng, X. S. Gao, and J. Li. "Root isolation for bivariate polynomial systems with local generic position method". In: *Proceedings of the 2009 international symposium on Symbolic and algebraic computation.* ACM. 2009, pp. 103–110.

[26] J. S. Cheng, X. S. Gao, and C. K. Yap. "Complete numerical isolation of real roots in zero-dimensional triangular systems". In: *Journal of Symbolic Computation* 44.7 (2009), pp. 768–785.

[27] G. E. Collins. "Quantifier elimination for real closed fields by cylindrical algebraic decompostion". In: *Automata theory and formal languages.* Springer, 1975, pp. 134–183.

[28] G. E. Collins and A. G. Akritas. "Polynomial real root isolation using Descarte's rule of signs". In: *Proceedings of the third ACM symposium on Symbolic and algebraic computation.* 1976, pp. 272–275.

[29] G. E. Collins and H. Hong. "Partial cylindrical algebraic decomposition for quantifier elimination". In: *Journal of Symbolic Computation* 12.3 (1991), pp. 299–328.

[30] G. E. Collins, J. R. Johnson, and W. Krandick. "Interval arithmetic in cylindrical algebraic decomposition". In: *Journal of Symbolic Computation* 34.2 (2002), pp. 145–157.

[31] D. A. Cox, J. Little, and D. O'Shea. *Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra.* Springer Science & Business Media, 2013.

[32] D. A. Cox, J. Little, and D. O'Shea. "Using Algebraic Geometry". In: *Graduate Texts in Mathematics* 185 (1998).

[33] R. Curtain and H. Zwart. *Introduction to Infinite-Dimensional Linear Systems Theory.* Springer-Verlag, 1995.

[34] R. Descartes. *Géométrie.(1636) In: A source book in Mathematics. Massachussetts.* 1969.

[35] D. I. Diochnos, I. Z. Emiris, and E. Tsigaridas. "On the asymptotic and practical complexity of solving bivariate systems over the reals". In: *Journal of Symbolic Computation* 44.7 (2009), pp. 818–835.

[36] A. Dolzmann, A. Seidl, and T. Sturm. "Efficient projection orders for CAD". In: *Proceedings of the 2004 international symposium on Symbolic and algebraic computation*. 2004, pp. 111–118.

[37] A. Dolzmann, T. Sturm, and V. Weispfenning. "Real quantifier elimination in practice". In: *Algorithmic algebra and number theory*. Springer, 1999, pp. 221–247.

[38] L. F. Dominguez, D. A. Narciso, and E. N. Pistikopoulos. "Recent advances in multiparametric nonlinear programming". In: *Computers & Chemical Engineering* 34.5 (2010), pp. 707–716.

[39] J. C. Doyle, B. A. Francis, and Tannebaum A. R. *Feedback Control Theory*. Dover Publications, 1992.

[40] D. Eisenbud. *Commutative algebra: with a view toward algebraic geometry*. Vol. 150. Springer Science & Business Media, 2013.

[41] M. El Kahoui. "An elementary approach to subresultants theory". In: *Journal of Symbolic Computation* 35.3 (2003), pp. 281–292.

[42] P. Emeliyanenko and M. Sagraloff. "On the complexity of solving a bivariate polynomial system". In: *Proceedings of the 37th International Symposium on Symbolic and Algebraic Computation*. 2012, pp. 154–161.

[43] I. Z. Emiris and E. Tsigaridas. "Real solving of bivariate polynomial systems". In: *International Workshop on Computer Algebra in Scientific Computing*. Springer. 2005, pp. 150–161.

[44] I. A. Fotiou, P. A. Parrilo, and M. Morari. "Nonlinear parametric optimization using cylindrical algebraic decomposition". In: *Proceedings of the 44th IEEE Conference on Decision and Control*. IEEE. 2005, pp. 3735–3740.

[45] I. A. Fotiou et al. "Parametric optimization and optimal control using algebraic geometry methods". In: *International Journal of Control* 79.11 (2006), pp. 1340–1358.

[46] P.M. Gahinet and P. Apkarian. "Numerical computation of the L_∞ norm revisited". In: *31st IEEE Conference on Decision and Control*. 1992.

[47] J. Garloff and A. P. Smith. "Investigation of a subdivision based algorithm for solving systems of polynomial equations". In: (2000).

[48] Y. Genin, P. Dooren, and V. Vermaut. "Convergence of the calculation of H_∞ norms and related questions". In: *Mathematical Theory of Networks and Systems: proceedings of the MTNS-98 symposium*. 1998, pp. 429–432.

[49] J. Gerhard, D. J. Jeffrey, and G. Moroz. "A package for solving parametric polynomial systems". In: *ACM Communications in Computer Algebra* 43.3/4 (2010), pp. 61–72.

[50] M. Giusti, G. Lecerf, and B. Salvy. "A Gröbner free alternative for polynomial system solving". In: *Journal of complexity* 17.1 (2001), pp. 154–211.

[51] K. Glover and D. McFarlane. "Robust stabilization of normalized coprime factor plant descriptor". In: *IEEE Transactions on Automatic Control* 34 (8 1989), pp. 821–230.

[52] L. Gonzalez-Vega and M. El Kahoui. "An improved upper complexity bound for the topology computation of a real algebraic plane curve". In: *Journal of Complexity* 12.4 (1996), pp. 527–544.

[53] L. Gonzalez-Vega et al. "Sturm—Habicht Sequences, Determinants and Real Roots of Univariate Polynomials". In: *Quantifier Elimination and Cylindrical Algebraic Decomposition*. Springer, 1998, pp. 300–316.

[54] G. M. Greuel and G. Pfister. *A Singular introduction to commutative algebra*. Springer Science & Business Media, 2012.

[55] W. Habicht. "Eine verallgemeinerung des sturmschen wurzelzählverfahrens". In: *Commentarii Mathematici Helvetici* 21.1 (1948), pp. 99–116.

[56] J. W. Helton. "Orbit Structure of the Mobius Transformation Semigroup Acting on H_∞ (Broadband Matching)". In: *Topics in Functional Analysis: Advances in Mathematics Supplementary Studies* 3 (1978), pp. 129–157.

[57] D. Henrion, M. Sebek, and M. Hromcik. "On computing the H_∞-norm of a polynomial matrix fraction". In: *European Control Conference (ECC)*. 2001.

[58] C. Hermite. "Extrait d'une lettre de Mr. Ch. Hermite de Paris à Mr. Borchardt de Berlin sur le nombre des racines d'une équation algébrique comprises entre des limites données." In: *Journal für die reine und angewandte Mathematik* 52 (1856), pp. 39–51.

[59] H. Hong. "An improvement of the projection operator in cylindrical algebraic decomposition". In: *Proceedings of the international symposium on Symbolic and algebraic computation*. 1990, pp. 261–264.

[60] M. Kanno and M. C. Smith. "Validated numerical computation of the L_∞-norm for linear dynamical systems". In: *Journal of Symbolic Computation* 41.6 (2006), pp. 697–707.

[61] M. Kanno et al. "Parametric optimization in control using the sum of roots for parametric polynomial spectral factorization". In: *Proceedings of the 2007 international symposium on Symbolic and algebraic computation*. 2007, pp. 211–218.

[62] D. Lazard. "Solving zero-dimensional algebraic systems". In: *Journal of symbolic computation* 13.2 (1992), pp. 117–131.

[63] D. Lazard and F. Rouillier. "Solving parametric polynomial systems". In: *Journal of Symbolic Computation* 42.6 (2007), pp. 636–667.

[64] S. Lazard, M. Pouget, and F. Rouillier. "Bivariate triangular decompositions in the presence of asymptotes". In: *Journal of Symbolic Computation* 82 (2017), pp. 123–133.

[65] H. P. Le and M. Safey El Din. "Solving parametric systems of polynomial equations over the reals through Hermite matrices". In: *arXiv preprint arXiv:2011.14136* (2020).

[66] X. Li, M. M. Maza, and W. Pan. "Computations modulo regular chains". In: *Proceedings of the 2009 international symposium on Symbolic and algebraic computation*. 2009, pp. 239–246.

[67] T. Lickteig and M. F. Roy. "Sylvester–Habicht sequences and fast Cauchy index computation". In: *Journal of Symbolic Computation* 31.3 (2001), pp. 315–341.

[68] K. M. Lynch and F. C. Park. *Modern robotics*. Cambridge University Press, 2017.

[69] K. Mahler. "An inequality for the discriminant of a polynomial." In: *Michigan Mathematical Journal* 11.3 (1964), pp. 257–262.

[70] M. Marden. *Geometry of polynomials*. 3. American Mathematical Soc., 1949.

[71] V. J. Mathews and G. Sicuranza. *Polynomial signal processing*. John Wiley & Sons, Inc., 2000.

[72] S. McCallum. "An improved projection operation for cylindrical algebraic decomposition of three-dimensional space". In: *Journal of Symbolic Computation* 5.1-2 (1988), pp. 141–161.

[73] K. Mehlhorn, M. Sagraloff, and P. Wang. "From approximate factorization to root isolation with application to cylindrical algebraic decomposition". In: *Journal of Symbolic Computation* 66 (2015), pp. 34–69.

[74] B. Mourrain and J. P. Pavone. "Subdivision methods for solving polynomial equations". In: *Journal of Symbolic Computation* 44.3 (2009), pp. 292–306.

[75] S. J. Orfanidis. *Introduction to signal processing.* Prentice-Hall, Inc., 1995.

[76] A. Quadrat. "The homological perturbation lemma and its applications to robust stabilization". In: *8th IFAC Symposium on Robust Control Design.* Vol. 48. 2015, pp. 07–12.

[77] G. Rance. "Commande H-infini paramétrique et application aux viseurs gyrostabilisés". PhD thesis. Université Paris-Saclay, 2018.

[78] G. Rance et al. "A symbolic-numeric method for the parametric H_∞ loop-shaping design problem". In: *8.* Vol. 22nd International Symposium on Mathematical Theory of Networks and Systems. 2016.

[79] G. Rance et al. "Explicit H_∞ controllers for 1st to 3rd order single-input single-output systems with parameters". In: *8.* Vol. IFAC 2017 Workshop Congress. 2017.

[80] G. Rance et al. "Explicit H_∞ controllers for 4th order single-input single-output systems with parameters and their applications to the two mass-spring system with damping". In: *8.* Vol. IFAC 2017 Workshop Congress. 2017.

[81] J. Renegar. "On the worst-case arithmetic complexity of approximating zeros of systems of polynomials". In: *SIAM Journal on Computing* 18.2 (1989), pp. 350–370.

[82] W. C. Rheinboldt. *Methods for solving systems of nonlinear equations.* SIAM, 1998.

[83] R. Rioboo. "Real algebraic closure of an ordered field: implementation in axiom". In: *Papers from the international symposium on Symbolic and algebraic computation.* 1992, pp. 206–215.

[84] G. Robel. "On the computation the infinity norm". In: *IEEE Transactions on Automatic Control* 34 (1989), pp. 383–391.

[85] A. Rosales et al. "Formation control and trajectory tracking of mobile robotic systems–a Linear Algebra approach". In: *Robotica* 29.3 (2011), pp. 335–349.

[86] F. Rouillier. "Algorithmes pour l'étude des solutions réelles des systèmes polynomiaux". Habilitation à diriger des recherches. Université Pierre et Marie Curie - Paris 6, Mar. 2007.

[87] F. Rouillier. "Solving zero-dimensional systems through the rational univariate representation". In: *Applicable Algebra in Engineering, Communication and Computing* 9.5 (1999), pp. 433–461.

[88] F. Rouillier and P. Zimmermann. "Efficient isolation of polynomial's real roots". In: *Journal of Computational and Applied Mathematics* 162.1 (2004), pp. 33–50.

[89] V. Rovenski and V. Y. Rovenskii. *Geometry of Curves and Surfaces with MAPLE*. Springer Science & Business Media, 2000.

[90] S. M. Rump. "Solving algebraic problems with high accuracy". In: *A new approach to scientific computation*. Elsevier, 1983, pp. 51–120.

[91] M. Sagraloff and C. K. Yap. "A simple but exact and efficient algorithm for complex root isolation. 36th ISSAC". In: (2011).

[92] R. Seidel and N. Wolpert. "On the exact computation of the topology of real algebraic curves". In: *Proceedings of the twenty-first annual symposium on Computational geometry*. 2005, pp. 107–115.

[93] E. C. Sherbrooke and N. M. Patrikalakis. "Computation of the solutions of nonlinear polynomial systems". In: *Computer Aided Geometric Design* 10.5 (1993), pp. 379–405.

[94] J. Shipman. "Improving the fundamental theorem of algebra". In: *The Mathematical Intelligencer* 29.4 (2007), pp. 9–14.

[95] V. Sima. "Efficient algorithm for L_∞-norm calculations". In: *IFAC Proceedings Volumes*. Vol. 39. 2006, pp. 519–524.

[96] A. Strzebonski and E. Tsigaridas. "Univariate real root isolation in an extension field". In: *Proceedings of the 36th international symposium on Symbolic and algebraic computation*. 2011, pp. 321–328.

[97] C. Sturm. "Mémoire sur la résolution des équations numériques". In: *Collected Works of Charles François Sturm*. Springer, 2009, pp. 345–390.

[98] J. J. Sylvester. "XVIII. On a theory of the syzygetic relations of two rational integral functions, comprising an application to the theory of Sturm's functions, and that of the greatest algebraical common measure". In: *Philosophical transactions of the Royal Society of London* 143 (1853), pp. 407–548.

[99] A. Tannenbaum. "Feedback stabilization of linear dynamical plants with uncertainty in the gain factor". In: *International Journal of Control* 32.1 (1980), pp. 1–16.

[100] A. Varga and P. Parrilo. "Fast algorithms for solving H_∞-norm minimization problems". In: *Conference on Decision and Control*. 2001, pp. 261–266.

[101] M. Vidyasagar. *Nonlinear systems analysis*. SIAM, 2002.

[102] G. Vinnicombe. *Uncertainty and Feedback, H_∞ loop-shaping and the ν-gap metric*. Imperial College Press, 2001.

[103] J. Von Zur Gathen and J. Gerhard. *Modern computer algebra*. Cambridge university press, 2013.

[104] X. Wang. "A simple proof of Descartes's rule of signs". In: *The American Mathematical Monthly* 111.6 (2004), p. 525.

[105] C. K. Yap. *Fundamental problems of algorithmic algebra*. Vol. 49. Oxford University Press Oxford, 2000.

[106] G. Zames. "Feedback and optimal sensitivity: Model reference transformations, multiplicative seminorms, and approximate inverses". In: *IEEE Transactions on Automatic Control* 26.2 (Apr. 1981), pp. 301–320. ISSN: 0018-9286.

[107] K. Zhou, J. C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice Hall, 1996.