Computing the Supremum of the Real Roots of a Parametric Univariate Polynomial (extended abstract)

Changbo Chen[†,‡], Marc Moreno Maza[‡], Yuzhen Xie[‡]

[†] Chongqing Key Laboratory of Automated Reasoning and Cognition, CIGIT, CAS, China
University of Western Ontario, London, Ontario, Canada

Abstract. Given a bivariate polynomial $p(W, X)$ we aim at computing the supremum of the real values x such that there exists a real value w satisfying $p(w, x) = 0$. We allow the coefficients of p to depend on real parameters. Our approach relies on the notion of border polynomial and takes advantage of triangular decomposition techniques. We report on the implementation of our algorithm and illustrate its effectiveness with problems from the theory of robust control.

1 Introduction

The work reported in this paper is motivated by problems arising in control theory and requiring to compute quantities which depend on parameters. A well-known source of such problems is Model Predictive Control (MPC) for which computational strategies decompose the work on off-line and on-line phases [18, 19] and lead to parametric programming (or parametric optimization). Another source, which can also be handled by parametric programming, is robust and optimal control [11, 12, 20, 23], in particular for linear dynamical systems with real parameters. This is, in fact, the application targeted by the present work.

Although the primary approach for solving parametric programming problems is based on numerical approximation methods [9, 10], a few methods based on symbolic computation [1,12,14] have also been proposed. Symbolic approaches for solving parametric optimization problems have at least the following advantages w.r.t. their numerical counterparts. Firstly, non-convex feasible regions are not a theoretical concern for the symbolic approaches. Secondly, the size of the feasible parameter regions, even when unbounded, does not create extra difficulty. 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. The paper [9] includes an account on the major difficulties faced by the approximation methods used by the numerical approaches in parametric optimization.

Before stating the problem studied here, we present our targeted application. For a linear dynamical system, we aim at computing the $\mathcal{H}_\infty$ norm of its transfer matrix when this latter depends on real parameters. We briefly review the necessary materials, following the notations of [5]. Let A, B, C, D be real matrices with respective formats $n \times n$, $n \times m$, $p \times n$, $p \times m$. Consider the linear dynamical system

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases} \tag{1}$$

with transfer matrix

$$G(s) = C(sI_n - A)^{-1}B + D. \tag{2}$$

When A is said *stable*, that is, when all its eigenvalues of A have negative real part, one defines the $\mathcal{H}_\infty$ norm of the transfer matrix as

$$||G(s)||_\infty \;=\; \sup_{\Re(s)>0} \sigma_{\max}(G(s)) \;=\; \sup_{\omega\in\mathbb{R}} \sigma_{\max}(G(\imath\omega)). \tag{3}$$

Here we have $\sigma_{\max}(F) = \lambda_{\max}^{1/2}(F^*F)$, where $\sigma_{\max}(\cdot)$ and $\lambda_{\max}(\cdot)$ denote respectively the maximum singular value and maximum eigenvalue of a real square matrix.

The $\mathcal{H}_\infty$ norm of a single input single output (SISO) linear system is the peak gain of the frequency response. For a multiple input multiple output (MIMO) system, the $\mathcal{H}_\infty$ norm is the peak gain across all input/output channels. The $\mathcal{H}_\infty$ norm as a measure is thoroughly embedded in modern control theory. For instance, in robust control it takes the role of a robustness measure [23] and in model order reduction it is used as an error measure [16]. In the late 90's, a few algorithms demonstrating fast convergence of iterative approaches and exploiting the properties of the singular values of a transfer matrix have been developed [4–6]. Recently the methods reported in [13] and [2] compute the $\mathcal{H}_\infty$ norm via localizing the common roots of two or three polynomials. A new algorithm that is efficient for descriptor systems is achieved by computing the eigenvalues of certain structured matrix pencils [3]. However, all these methods are numeric and are devoted to linear systems free of parameters.

We are now ready to state the problem studied in this paper. Let $p \in \mathbb{R}[W, H][X]$ be a univariate polynomial in X whose coefficients are multivariate polynomials in two sets of variables $W = W_1, \ldots, W_m$ and $H = H_1, \ldots, H_n$ and, with coefficients in the field $\mathbb{R}$ of real numbers.

For a value of $h \in \mathbb{R}^n$ of H we evaluate p at $H = h$ and obtain a polynomial $p_h \in \mathbb{R}[W][X]$. We denote by $x_{\sup}(h)$ the supremum of the set

$$\Pi_h \;=\; \{x \in \mathbb{R} \;\mid\; (\exists (w_1, \ldots, w_m) \in \mathbb{R}^m) \;\; p_{w,h}(x) = 0\} \tag{4}$$

where $p_{w,h}$ is the polynomial of $\mathbb{R}[X]$ obtained by evaluating p_h at $W = w_1, \ldots, w_m$. In a more compact form, this writes

$$x_{\sup}(h) \;=\; \sup_{w,\, p(w,h,x)=0} x. \tag{5}$$

Note the use of sup (supremum, or least upper bound) instead of max since the set Π_h may not admit a maximum, for instance, if $p = w_1(x - h_1)$. Whether Π_h

admits a maximum or not, a supremum of that set always exists. This results from the *completeness* of the real numbers, thus the following property: every nonempty subset of the set of real numbers that is bounded from above has a supremum that is also a real number. If the set Π_h is empty, by convention we take $-\infty$ as supremum. If the set Π_h is not empty and unbounded from above, then $+\infty$ is its supremum. Now we denote by $x_{\sup}$ the function from $\mathbb{R}^n$ to $\mathbb{R}$ mapping h to $x_{\sup}(h)$.

We view h as a parameter and we call *Parametric Supremum Real Root Problem* (PSRRP for short) the problem of computing $x_{\sup}(h)$ for every h. In the absence of parameters, we denote by $x_{\sup}$ the supremum $\sup_{w,\, p(x,w)=0}\ x$ and call *Supremum Real Root Problem* (SRRP for short) the problem of computing $x_{\sup}$. In the absence of the variables $W_1, \ldots, W_m$, PSRRP remains well defined as above. However, the algorithmic solutions to PSRRP depend on the value of n.

The case $n = 1$, to which this paper is devoted, allows us to propose an algorithmic solution which is a practically efficient and specific to this case. As mentioned above, solving PSRRP with $n = 1$ is motivated by a major application of robust control theory: computing the $\mathcal{H}_\infty$ norm of the transfer matrix of a linear dynamical system with parametric uncertainty.

Before discussing our algorithmic solution to PSRRP, we walk through a few simple examples so as to highlight the different roles of the variables $W = W_1, \ldots, W_m$ and $H = H_1, \ldots, H_n$. We fix $m = n = 1$. Consider the polynomial $p_1 = h_1 x - w_1$, we have $x_{\sup}(h_1) = +\infty$ for all $h_1 \in \mathbb{R}$. Choose another polynomial $p_2 = h_1^2 x - w_1^2 - 1$. We have $x_{\sup}(h_1) = +\infty$ if $h_1 \neq 0$ and $-\infty$ otherwise. Now consider the polynomial $p_3 = x + h_1 w_1^2 - h_1 - 1$. Then, we have

$$x_{\sup}(h_1) = \begin{cases} +\infty & h_1 < 0 \\ h_1 + 1 & h_1 \geq 0. \end{cases}$$

In the subsequent sections, we shall assume $n = 1$ and simply write W instead of W_1. We give now an overview of our results.

In Section 2, an algorithm for SRRP is easily derived from the theory of the border polynomial [17, 21]. We do not claim that our solution is new. In fact, we believe that it is equivalent to that of Kanno and Smith[1] in [13]. However, the use of the theory of the border polynomial makes the presentation of our solution much simpler.

In Section 3, we turn our attention to the parametric case, that is, PSRRP. Here again we consider the border polynomial of $p(W, H, X)$; let us denote it by $f(H, X)$. An additional difficulty comes from the fact that the roots of f, regarded as a univariate polynomial in X, are now functions of the parameters H. In order to adapt the algorithm of Section 2, one needs delineability, that is, to make the graphs of those functions locally disjoint. This is achieved by means of a *real comprehensive triangular decomposition* of $f(H, X) = 0$, regarded as a parametric system with H as parameters. Via point sampling, this delineability

[1] One should note that the primary concern of those Authors is to compute the $\mathcal{H}_\infty$ norm of a linear dynamical system numerically.

property allows us to reduce our computation to the non-parametric case, that is, SRRP. In some exceptional cases (typically when suprema are attained on the variety defined by f) our algorithm cannot conclude, in which cases a full cylindrical algebraic decomposition of $f(H, X)$ is needed.

Section 4 illustrates our algorithm with a few examples, taken from the literature, applied to an implementation realized with the `RegularChains` library `www.regularchains.org`.

2 Solving the Supremum Real Root Problem via BP/DV

Recall that, in the non-parametric case, the problem is, for a given bivariate polynomial $p \in \mathbb{R}[W, X]$ to compute $x_{\sup}$ defined by

$$x_{\sup} \;=\; \sup_{w,\, p(x,w)=0} x. \tag{6}$$

Let us view p as a parametric polynomial with parameter X. The motivation is the following. Consider the real curve $p = 0$ in the (x, w)-plane and assume that it is not empty. Then, two cases arise:

(1) either for every positive real value q there is a point on that curve with q as an X-coordinate and the curve is *unbounded* in the X-direction; then the answer to our supremum problem is $+\infty$;
(2) or the curve is *bounded* in the X-direction and the supremum S is the X-coordinate of a "special" point.

Regarding p as a parametric polynomial in X, and computing its *border polynomial* (BP) [21] or its *discriminant variety* (DV) [15] (which are equivalent notions in the case of a parametric system consisting of a single polynomial equation, as it follows from the results of [17]) will tell us which case we are in. Moreover, if we are in the second case, we will deduce the value of $x_{\sup}$.

The BP/DV of p, regarded as a parametric semi-algebraic system with parameter X, consists of all real X-values at which the real curve $p = 0$ is in one of the following cases:

(1) vertical (i.e. parallel to the W-axis) asymptote,
(2) singular point of the curve,
(3) critical point or singular value of the projection of the curve onto the X-axis.

Moreover, the set of all those X-values is finite and is given by the real roots of the polynomial

$$f = \mathrm{lcoeff}_W(p) \cdot \mathrm{discrim}_W(p), \tag{7}$$

where lcoeff and discrim denote the leading coefficient and the discriminant, respectively.

We make two observations about the polynomial p:

(1) if p admits a univariate factor $u \in \mathbb{R}[W]$ (thus not depending on X) such that $u = 0$ has real solutions, then we clearly have $\sup_{w,\, p(x,w)=0}\ x = +\infty$.

(2) if p admits a univariate factor $u \in \mathbb{R}[X]$ (thus not depending on W) then u divides $\mathrm{lcoeff}_W(p)$ and thus f.

Based on these preliminary observations, we are ready to state our algorithm `SupRealRoot`. Let $\xi_1 < \cdots < \xi_e$ be the real roots of f. Define $\xi_0 = -\infty$ and $\xi_{e+1} = +\infty$. The algorithm below computes $x_{\sup} = \sup\{x \in \mathbb{R} \mid \exists w \in \mathbb{R}\ p(w,x) = 0\}$.

```
SupRealRoot(p) begin
   for i = e+1 downto 1 by −1 do {
      let q be a rational number s.t. ξ_{i−1} < q < ξ_i
      if p(q, W) = 0 has real roots in W then return ξ_i
      if i ≤ e and p(ξ_i, W) = 0 has real roots in W then return ξ_i
   }
   return ξ_0
end
```

Observe that each interval $]\xi_{i-1}, \xi_i[$ is a connected component of the complement of the BP/DV of $p = 0$ regarded as a parametric semi-algebraic system with parameter X. Thus, the following two properties are equivalent.

1. There exists $q \in\ ]\xi_{i-1}, \xi_i[$ such that $p(q, W) = 0$ admits at least one real solution $W = w$.
2. For every $q \in\ ]\xi_{i-1}, \xi_i[$ there exists at least one point on the real curve $p = 0$ with q as X-coordinate.

The correctness of our algorithm `SupRealRoot` follows immediately from the above equivalence.

Using the `RegularChains` library in MAPLE, we have realized a command `SupRealRoot` implementing the above algorithm. This command takes as input a bivariate polynomial $p \in \mathbb{R}[W, X]$ and returns $x_{\sup} = \sup_{w,\, p(x,w)=0} x$ together with additional information in order to support `ParametricSupRealRoot`, as we shall see in the next section. To this end, our command `SupRealRoot` actually returns a pair where the first item is the supremum $x_{\sup}$ and the second one is defined below:

- ∞, if $x_{\sup} = +\infty$ holds,
- 0, if $x_{\sup} = -\infty$,
- a real root index i of an irreducible factor g of the polynomial f defining DV (i.e. the zero locus of BP) such that $g(\xi_i) = 0$, indicating that the supremum is reached between the $(i-1)$-th and i-th roots of g and is equal to the latter,
- $-i$, if the supremum $x_{\sup}$ is equal to the i-th root of g but cannot be reached within a connected component of the complement of DV, i.e., only in DV itself.

For efficiency reasons, in our implementation we have a special case for the factors g of p depending only on w. We first factorize the polynomial f and then apply real root isolation to each irreducible factor. Of course, isolation intervals are refined until they are pairwise disjoint such that real algebraic numbers (namely $\xi_1 < \cdots < \xi_e$) that these intervals encode can be effectively sorted.

3 Solving the Parametric Supremum Real Root Problem via Real Comprehensive Triangular Decomposition

Recall the problem stated in (5): For each parameter value $h \in \mathbb{R}^n$ compute

$$x_{\sup}(h) \;=\; \sup_{w,\, p(w,h,x)=0} x.$$

Similarly to the non-parametric case, we define

$$f = \mathrm{lcoeff}_W(p) \times \mathrm{discrim}_W(p) \;\in \mathbb{R}[H, X].$$

Due to the role of H as a parameter of the problem, we are interested in the real roots $x_1(h) < \cdots < x_e(h)$ of f regarded as a univariate polynomial in X. The difficulty is that the number of these roots depends on h. Thus we need a case discussion for the real roots of f as a function of h.

This case discussion can be provided by the command `RealComprehensive-Triangularize` [7], applied to f and regarding H as parameters. We obtain a partition, $C_1, \ldots, C_e$, of the parameter space into connected components such that above each cell C_i the real X-values satisfying $f = 0$ are given by continuous functions $x_{i_1}(h), x_{i_2}(h), \ldots$ with disjoint graphs (encoded by the data structure $squarefree\_semi\_algebraic\_system$ [7, 8]).

For each cell C_i which is full-dimensional in the parameter space, we perform the following tasks.

(1) Obtain a sample point v_i of the cell C_i[2]
(2) Call the command `SupRealRoot` (as defined in Section 2 for the non-parametric case) at $h = v_i$. Three cases arise.
  (2.1) If the non-parametric `SupRealRoot` command returns a pair of the form $[\xi, m]$ with $\xi \in \{+\infty, -\infty\}$ (that is, with $m \in \{0, \infty\}$), then the function `ParametricMaxRealRoot` returns $[\xi, C_i]$.
  (2.2) If the non-parametric `SupRealRoot` returns a pair of the form $[\xi, m]$ where $m > 0$ holds, then we compute the polynomial g which has ξ as its j-th real root at $h = v_i$ and `ParametricMaxRealRoot` returns $[[j, g], C_i]$.
  (2.3) In all other cases, which can be regarded as exceptional, our method cannot conclude directly and we are led to apply a CAD-based approach, say computing a CAD of $p(x, w, h) = 0$ for $h < x < w$.

In the above algorithm, cells C_i which are not full-dimensional in the parameter space, as well as cells C_i leading to (2.3) (meaning that $s_{\sup}(v_i)$ is attained on $f = 0$) are situations that are encountered rarely in practice, that is, when parameters are specialized to actual values.

Using the `RegularChains` library in Maple, we have realized a command `ParametricSupRealRoot` implementing the above algorithm, which is illustrated in the next section.

[2] In fact, the `RealComprehensiveTriangularize` command computes a sample point with each of the cells $C_1, \ldots, C_e$.

4 Examples

In this section, we illustrate the use of a command `ParametricHinfinityNorm` that we developed in Maple based on the method `ParametricSupRealRoot` described in section 3. The output of `ParametricHinfinityNorm` has similar specifications as `RealComprehensiveTriangularize`: it returns a partition of the parameter space into CAD cells and, above each cell, a formula for the $\mathcal{H}_\infty$ norm of a linear parametric dynamical system, taking its transfer matrix as input. In each case of the output, the displayed result is a pair consisting of two items. The second one is a semi-algebraic system describing a list of CAD cells C. The first item is a pair of the form $[\ell, g(h, x)]$ such that the $\mathcal{H}_\infty$ norm value is the square root of the ℓ-th root (in x) of the $g(h, x) = 0$, which is guaranteed to be delineable for all $h \in C$.

The first example is taken from Problem 4.8 in [22]. Given a transfer function G_s, the problem is to compute $||G_s||_\infty$ using the Bode plot and state space algorithm, respectively for $c = 1, 0.1, 0.01, 0.001$. In our computation below, we treat c as a real parameter with constraint $0 < c <= 1$. The result consists of four cases. Since there is only one parameter, namely c, the corresponding semi-algebraic set is either a point or an open interval. The value of $||G_s||_\infty$ is 1 for the first three cases, i.e. when $c = 1$, or $\frac{1}{2} < c < 1$, or $c = \frac{1}{2}$. The fourth case shows that when $0 < c < \frac{1}{2}$, the value of $||G_s||_\infty$ is the square root of the second real root of the polynomial $f = (256c^8 - 768c^6 + 768c^4 - 256c^2)x^2 + (256c^6 + 32 - 480c^4 + 192c^2)x - 27$, which can be computed by a real root isolation of a univariate polynomial for a specified value of c. For instance, the value of $||G_s||_\infty$ at $c = 0.1$ is 3.575787201.

```
> Gs := Matrix([1/((s^2+2*c*s+1)*(s+1))]);
```

$$Gs := \left[\frac{1}{(s^2 + 2\,c\,s + 1)\,(s + 1)} \right]$$

```
> Hs := ParametricHinfinityNorm(Gs, 's', [c>0, c<=1]);
```

$Hs := [[[[1, \textit{squarefree\_semi\_algebraic\_system}], [\textit{cad\_cell}]], [[1, \textit{squarefree\_semi\_algebraic\_system}], [\textit{cad\_cell}]], [[1, \textit{squarefree\_semi\_algebraic\_system}], [\textit{cad\_cell}]], [[2, \textit{squarefree\_semi\_algebraic\_system}], [\textit{cad\_cell}]]], \textit{polynomial\_ring}]$

```
> Display(Hs[1][1], Hs[-1]);
```

$$[[1, x - 1 = 0], [c = 1]]$$

```
> Display(Hs[1][2], Hs[-1]);
```

$$\left[[1, x - 1 = 0], \left[\mathrm{And}\left(\frac{1}{2} < c, c < 1\right)\right]\right]$$

```
> Display(Hs[1][3], Hs[-1]);
```

$$\left[[1, x - 1 = 0], \left[c = \frac{1}{2}\right]\right]$$

```
> Display(Hs[1][4], Hs[-1]);
```

$$\left[[2, (256\,c^8 - 768\,c^6 + 768\,c^4 - 256\,c^2)\,x^2 + (256\,c^6 + 32 - 480\,c^4 + 192\,c^2)\,x - 27 = 0], \left[\mathrm{And}\left(0 < c, c < \frac{1}{2}\right)\right]\right]$$

The second example is the classical mass-spring-dampler system $m\ddot{x} + b\dot{x} + k = u$, where m is the mass $[kg]$, b is the viscous damping coefficient $[Ns/m]$, k is the spring constant and u is the force input $[N]$. In the following Maple session we apply our functions to study the $\mathcal{H}_\infty$ of the mass-spring-dampler system with positive real parameters m, b, k. We first compute the transfer function from its state space representation A, B, C and $D = 0$. The output of

`ParametricHinfinityNorm` on this system has three cases. The first case reports the CAD cells which are not full-dimensional in the parameter space and are not processed. The second case tells that when $k < \frac{b^2}{4m}$ or $k > \frac{b^2}{4m}$ and $k < \frac{b^2}{2m}$, the `ParametricHinfinityNorm` of this system is $\frac{1}{k}$. The third case means that when $k > \frac{b^2}{2m}$, the `ParametricHinfinityNorm` is $\frac{2m}{b\sqrt{-b^2+4mk}}$.

```
> A := Matrix([[0,1], [-k/m,-b/m]]): B := Matrix([[0], [1/m]]): C := Matrix([1,0]):
  T := DynamicSystems:-TransferFunction(A,B,C):
  T:-tf;
```

$$\left[\frac{1}{m\,s^2+b\,s+k}\right]$$

```
> Hm := ParametricHinfinityNorm(T:-tf, 's', [m>0, k>0, b>0]);
```

$Hm := [[["Not full-dimension, not processed", [cad\_cell, cad\_cell]], [[1, squarefree\_semi\_algebraic\_system], [cad\_cell, cad\_cell]], [[1, squarefree\_semi\_algebraic\_system], [cad\_cell]]], polynomial\_ring]$

```
> Display(Hm[1][1], Hm[-1]);
```

$$\left[\text{"Not full-dimension, not processed"}, \left[\left\{\begin{array}{l} k = \frac{1}{4}\frac{b^2}{m} \\ 0 < m \\ 0 < b \end{array}\right., \left\{\begin{array}{l} k = \frac{1}{2}\frac{b^2}{m} \\ 0 < m \\ 0 < b \end{array}\right.\right]\right]$$

```
> Display(Hm[1][2], Hm[-1]);
```

$$\left[[1, k^2\,x - 1 = 0], \left[\left\{\begin{array}{c} \mathrm{And}\left(0 < k, k < \frac{1}{4}\frac{b^2}{m}\right) \\ 0 < m \\ 0 < b \end{array}\right., \left\{\begin{array}{c} \mathrm{And}\left(\frac{1}{4}\frac{b^2}{m} < k, k < \frac{1}{2}\frac{b^2}{m}\right) \\ 0 < m \\ 0 < b \end{array}\right.\right]\right]$$

```
> Display(Hm[1][3], Hm[-1]);
```

$$\left[[1, (-b^4 + 4\,m\,k\,b^2)\,x - 4\,m^2 = 0], \left[\left\{\begin{array}{c} \frac{1}{2}\frac{b^2}{m} < k \\ 0 < m \\ 0 < b \end{array}\right.\right]\right]$$

5 Concluding Remarks

Taking advantage of the notion of border polynomial and triangular decomposition techniques, we have presented an algorithm and its implementation for computing the supremum of the real roots of a parametric univariate polynomial. The precise formulation of this problem (with the bivariate polynomial $p(W, X)$ whose coefficients are real polynomials in H) targets the computation of the $\mathcal{H}_\infty$ norm of the transfer matrix of a linear dynamical system with parametric uncertainty.

Our implementation allows us to solve the vast majority of the examples that we have found in the literature. A few examples (like the 2-mass-2-spring-2-dampler system, which, in its full generality, has 6 parameters) cannot be solved by our code without specializing some of the parameters. However, our preliminary implementation offers several opportunities for optimization. For instance, in the context of our application to parametric $\mathcal{H}_\infty$ norm computation, the polynomial $f \in \mathbb{R}[H, X]$ defined in Section 3 is the border polynomial of the characteristic polynomial of the square of a transfer matrix: we have observed that f often had several irreducible factors and exploiting this fact when calling `RealComprehensiveTriangularize` should greatly reduce computation costs.

Acknowledgements

This work was supported in by MITACS of Canada, by the NSFC grant: 11301524 and by the grant cstc2012ggB40004.

References

1. H. Anai. A symbolic-numeric approach to multi-parametric programming for control design. In *ICCAS-SICE*, pages 3525–3530, 2009.
2. M. N. Belur and C. Praagman. An efficient algorithm for computing the H-infinity norm. *Automatic Control, IEEE Transactions on*, 56(7):1656–1660, 2011.
3. P. Benner, V. Sima, and M. Voigt. Robust and efficient algorithms for L-infinity norm computation for descriptor systems. *Robust Control Design*, 7(1):195–200, 2012.
4. 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-infinity norm. *Systems and Control Letters*, 15(1):1–7, 1990.
5. S. Boyd, V. Balakrishnan, and P. Kabamba. A bisection method for computing the H-infinity norm of a transfer matrix and related problems. *Mathematics of Control, Signals and Systems*, 2(3):207–219, 1989.
6. N. A. Bruinsma and M. Steinbuch. A fast algorithm to compute the H-infinity norm of a transfer function matrix. *Systems and Control Letters*, 14(4):287–293, 1990.
7. C. Chen, J. H. Davenport, M. Moreno Maza, B. Xia, and R. Xiao. Computing with semi-algebraic sets represented by triangular decomposition. In *Proceedings of the 36th international symposium on symbolic and algebraic computation*, ISSAC '11, pages 75–82, New York, NY, USA, 2011. ACM. To appear in the Journal of Symbolic Computation.
8. C. Chen, J.H. Davenport, F. Lemaire, M. Moreno Maza, N. Phisanbut, B. Xia, R. Xiao, and Y. Xie. Solving semi-algebraic systems with the RegularChains library in Maple. In *Proc. of MACIS*, pages 38–51, 2011.
9. L. F. Domnguez, D. A. Narciso, and E. N. Pistikopoulos. Recent advances in multiparametric nonlinear programming. *Computers & Chemical Engineering*, 34(5):707–716, 2010.
10. A. V. Fiacco. *Introduction to sensitivity and stability analysis in nonlinear programming*. Academic Press, USA, 1983.
11. I. A. Fotiou, P. A. Parrilo, and M. Morari. Nonlinear parametric optimization using cylindrical algebraic decomposition. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, pages 3735–3740, 2005.
12. I. A. Fotiou, P. Rostalski, P. A. Parrilo, and M. Morari. Parametric optimization and optimal control using algebraic geometry. *International Journal of Control*, 79(11):1340–1358, 2006.
13. M. Kanno and M. C. Smith. Validated numerical computation of the L-infinity norm for linear dynamical systems. *Journal of Symbolic Computation*, 41(6):697–707, 2006.
14. M. Kanno, K. Yokoyama, H. Anai, and S. Hara. Parametric optimization in control using the sum of roots for parametric polynomial spectral factorization. In *ISSAC*, pages 211–218, 2007.

15. D. Lazard and F. Rouillier. Solving parametric polynomial systems. *J. Symb. Comput.*, 42(6):636–667, 2007.
16. A. Megretski. H-infinity model reduction with guaranteed suboptimality bound. In *American Control Conference*, 2006.
17. M. Moreno Maza, B. Xia, and R. Xiao. On solving parametric polynomial systems. In *Proceedings of the Fourth Internationa Conference on Mathematical Aspects of Computer Science and Information Sciences (MACIS 2011), Edited by Stefan Raschau*, pages 205–215, 2011. To appear in the journal Mathematics in Computer Science.
18. E. N. Pistikopoulos, V. Dua, N. A. Bozinis, A. Bemporad, and M. Morari. On-line Optimization via Off-line Parametric Optimization Tools. In *International Symposium on Process Systems Engineering*, pages 183–188, Keystone, USA, July 2000.
19. E. N. Pistikopoulos, M. Georgiadis, and V. Dua. *Multi-parametric programming: theory, algorithms and applications, vol.1.* Wiley-VCH, 2007.
20. A. Varga and P. A. Parrilo. Fast algorithms for solving H-infinity norm minimization problems. In *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, volume 1, pages 261–266, 2001.
21. L. Yang, X. Hou, and B. Xia. Automated discovering and proving for geometric inequalities. In *Automated Deduction in Geometry*, pages 30–46, 1998.
22. K. Zhou and J. Doyle. *Essentials of Robust Control.* Prentice-Hall, 1998.
23. K. Zhou, J. Doyle, and K. Glover. *Robust and Optimal Control.* Prentice-Hall, 1996.