Computer algebra methods for testing the structural stability of multidimensional systems

Yacine Bouzidi
INRIA Saclay, DISCO project,
L2S, Supelec, Gif-sur-Yvette, France
Email: yacine.bouzidi@inria.fr

Alban Quadrat
INRIA Saclay, DISCO project,
L2S, Supelec, Gif-sur-Yvette, France
Email: alban.quadrat@inria.fr

Fabrice Rouillier
INRIA Paris-Rocquencourt, Ouragan,
IMJ, Paris, France
Email: fabrice.rouillier@inria.fr

***Abstract*—In this paper, we present new computer algebra based methods for testing the structural stability of n-D discrete linear systems (with $n \geqslant 2$). More precisely, starting from the usual stability conditions which resumes to deciding if an hypersurface has points in the unit polydisc, we show that the problem is equivalent to deciding if an algebraic set has real points and use state-of-the-art algorithms for this purpose. Our strategy has been implemented in `Maple` and its relevance demonstrated through numerous experimentations.**

I. Introduction

An important question in the study of multidimensional systems concerns their stability which is a necessary condition for the latter to work properly. In this paper, we are interested in testing the structural stability of multidimensional discrete linear systems.

Given a discrete linear system described within the frequency domain by the transfer function:

$$G(z_1,\ldots,z_n)=\frac{N(z_1,\ldots,z_n)}{D(z_1,\ldots,z_n)}, \qquad (1)$$

where N and D are polynomials in the complex variables $z_1,\ldots,z_n$ with real coefficients. This system is said to be *structurally stable* if the denominator of its transfer function is devoid from zeros in the complex unit polydisc $\mathbb{D}^n := \prod_{k=1}^{n}\{z_k \in \mathbb{C} \mid |z_k| \leqslant 1\}$, or in other words:

$$D(z_1,\ldots,z_n) \neq 0 \text{ for } |z_1| \leqslant 1,\ldots,|z_n| \leqslant 1. \qquad (2)$$

Actually, the simplicity of (2) contrasts significantly with the difficulty to develop efficient implementations for testing it. A first step toward this objective is the formulation of new conditions that are equivalent to the above condition but easier to handle. The following theorems, due to Strintzis [28] and DeCarlo et al. [12], are two good representatives of these reformulations.

Theorem 1 (Strintzis [28]). *Condition (2) is equivalent to the following set of conditions:*

$$\begin{cases} D(0,\ldots,0,z_n) \neq 0, & |z_n| \leqslant 1,\\ D(0,\ldots,0,z_{n-1},z_n) \neq 0, & |z_{n-1}| \leqslant 1,\ |z_n| = 1,\\ \qquad\vdots & \qquad\vdots\\ D(0,z_2,\ldots,z_n) \neq 0, & |z_2| \leqslant 1,\ |z_i| = 1, i > 2,\\ D(z_1,z_2,\ldots,z_n) \neq 0, & |z_1| \leqslant 1,\ |z_i| = 1, i > 1. \end{cases}$$

Theorem 2 (DeCarlo et al. [12]). *Condition (2) is equivalent to the following set of conditions:*

$$\begin{cases} D(z_1,1,\ldots,1) \neq 0, & |z_1| \leqslant 1,\\ D(1,z_2,1,\ldots,1) \neq 0, & |z_2| \leqslant 1,\\ \qquad\vdots & \quad\vdots\\ D(1,\ldots,1,z_n) \neq 0, & |z_n| \leqslant 1,\\ D(z_1,\ldots,z_n) \neq 0, & |z_1| = \ldots = |z_n| = 1. \end{cases}$$

Recent algebraic methods for testing the stability of n-dimensional discrete linear systems are, for the majority, based on the previous sets of conditions. On the one hand, the specific case of 2-dimensional systems has attracted considerable attention and numerous efficient tests have been proposed (see, for instance, [6], [17], [8], [29], [15] and the references therein). Common to all these tests is that they proceed recursively on the variables, reducing the computations with a 2-D polynomial to computations with a set of 1-D polynomials using algebraic tools like *resultant* and *sub-resultant* polynomials [4]. Such a recursive approach, which shows its relevance for 2-dimensional systems, becomes rather involved when it comes to n-dimensional systems with $n > 2$ mainly due to the exponential increase of the degree of the intermediate polynomials. This fact prevents the above 2-dimensional tests from being efficiently generalized to n-dimensional systems.

On the other hand, few implementations exist for n-dimensional systems with $n > 2$. Among the recent work on this problem, one can mention the work of Serban and Najim [27] where, using an extension of the 1-D Schur-Cohn criterion, the authors propose a new stability condition as an alternative of the set of conditions of Theorems 1 and 2. As a result, the stability is expressed as a positivity condition of $n-1$ polynomials on the torus $\mathbb{T}^{n-1} := \prod_{k=1}^{n-1}\{z_k \in \mathbb{C} \mid |z_k| = 1\}$. Unfortunately, such a condition becomes considerably hard to test as soon as the involved systems are not of low degree in few variables. To achieve practical efficiency, Dumitrescu [13], [14] proposes a sum-of-square approach to test the last Decarlo's condition (Theorem 2). The described method is however conservative, i.e. it provides only a sufficient stability condition. In all generality, it should be stressed that the existing stability tests for n-dimensional systems are either nonconservative but inefficient, or efficient (polynomial time) but conservative.

In this paper, our objective is to develop an n-D stability test which is nonconservative and increases the range of systems that can be reached in practice compared to the existing counterparts. In our approach, we start from the stability conditions given by DeCarlo et al. [12] and design a stability test based on classical real algebraic geometry algorithms. From the computation point of view, one can obviously remark that the n first conditions of Theorem 2 are easily checkable using classical univariate stability tests (see, for instance, [20], [18], [5], [7]). We thus focus in this work on testing efficiently the last condition of Theorem 2, i.e.:

$$D(z_1, \ldots, z_n) \neq 0 \quad \text{for} \quad |z_1| = \ldots = |z_n| = 1. \tag{3}$$

Our approach is roughly to show that the existence of complex zeros of $D(z_1, \ldots, z_n)$ on the torus $\mathbb{T}^n$ is equivalent to the existence of real zeros of some polynomial systems in the real space $\mathbb{R}^n$, these systems being computed from $D(z_1, \ldots, z_n)$ using certain transformations. The existence of real zeros of such systems is then checked using classical methods for testing the emptiness of real algebraic sets.

Our paper is organized as follows: In Section II, we overview in broad lines computer algebra methods for computing the real zeros of *semi-algebraic sets* (namely, sets of polynomial equations and inequalities) and recall the basic ideas behind these methods. In Section III, starting from (3), we show how one can compute – via some transformations – new conditions that can be tested efficiently using above methods. Finally, in Section IV, we illustrate our stability test on a set of non-trivial examples and show its practical efficiency through some experimental results.

II. Real zeros of semi-algebraic sets

A system is said to be *structurally unstable* if the subset of $\mathbb{C}^n$ defined by

$$E := \{(z_1, \ldots, z_n) \in \mathbb{C}^n \mid D(z_1, \ldots, z_n) = 0,\ |z_1| \leqslant 1, \ldots, |z_n| \leqslant 1\}$$

is not empty. E is a semi-algebraic subset of $\mathbb{R}^{2n}$. Indeed, if we note $z_k := x_k + i\, y_k$, where x_k (resp., y_k) is the real part (resp., the imaginary part) of z_k and i is the imaginary unit, then the polynomial $D(z_1, \ldots, z_n)$ can be rewritten as $D(z_1, \ldots, z_n) = \mathcal{R}(x_1, \ldots, x_n, y_1, \ldots, y_n) + i\, \mathcal{C}(x_1, \ldots, x_n, y_1, \ldots, y_n)$, where $\mathcal{R}, \mathcal{C} \in \mathbb{R}[x_1, \ldots, x_n, y_1, \ldots, y_n]$, and the inequalities $|z_k| \leqslant 1$ as $x_k^2 + y_k^2 \leqslant 1$, which shows that:

$$\begin{aligned} E \quad \approx \quad & \{(x_1, \ldots, x_n, y_1, \ldots, y_n) \in \mathbb{R}^{2n} \mid \\ & \mathcal{R}(x_1, \ldots, x_n, y_1, \ldots, y_n) = 0, \\ & \mathcal{C}(x_1, \ldots, x_n, y_1, \ldots, y_n) = 0, \\ & x_k^2 + y_k^2 \leqslant 1,\ k = 1, \ldots, n\}. \end{aligned}$$

Therefore, testing (2) resumes to testing that the above semi-algebraic set does not have any real components. This test can be performed using classical computer algebra methods for computing the real zeros of semi-algebraic systems which we are now going to briefly overview.

A. General symbolic simplifications

To study the real zeros of semi-algebraic systems, there are two classes of available algorithms. Those based on *Cylindrical Algebraic Decompositions* [4] and those based on the study of the *critical points* of well chosen functions (see, e.g., [4]).

In Cylindrical Algebraic Decomposition based methods, the elementary block is the study of univariate polynomials (with multivariate polynomials as coefficients). In short, the original algorithm due to Collins [10] takes as input a set of polynomials $\{P_1, \ldots, P_s\} \subset \mathbb{R}[x_1, \ldots, x_n]$ and computes a partition of $\mathbb{R}^k$ in *cells* where each P_i has a constant sign (for instance, the cylindrical algebraic decomposition of a set of univariate polynomials in $\mathbb{R}[x]$ is an union of points and open intervals that form a partition of $\mathbb{R}$). The cylindrical algebraic decomposition's algorithm is recursive with respect to the variables: at the step l, one has a set of polynomials $\{P_{l,1}, \ldots, P_{l,s_l}\} \subset \mathbb{R}[x_1, \ldots, x_{k-l-1}][x_{k-l}]$ and the goal is to compute a set of polynomials $\{P_{l+1,1}, \ldots, P_{l+1,s_{l+1}}\} \subset \mathbb{R}[x_1, \ldots, x_{k-l-1}]$ such that the sign conditions realized by $\{P_{l+1,1}, \ldots, P_{l+1,s_{l+1}}\}$ define a partition of $\mathbb{R}^{k-l-1}$ which naturally lifts on a partition of $\mathbb{R}^{k-l}$ compatible with $\{P_{l,1}, \ldots, P_{l,s_l}\}$ (where each cell realizes a unique sign condition with respect to these polynomials). Passing from the set $\{P_{l,1}, \ldots, P_{l,s_l}\}$ to the set $\{P_{l+1,1}, \ldots, P_{l+1,s_{l+1}}\}$ is referred to as the *projection* step. It consists essentially in computing resultants (and sub-resultants) for all possible pairs of polynomials as well as the discriminants of each polynomial. The *lifting* step consists essentially in isolating the real roots of univariate polynomials with real algebraic numbers as coefficients, which can be viewed as solving some triangular *zero-dimensional system* (system with a finite number of complex solutions).

Such methods compute much more than required in our case, the partition itself being useless, and their computation require a number of arithmetic operations which is doubly exponential in the number of variables of the polynomial ring, due, at least, to the iterative computation of the resultants (and sub-resultants). Moreover, it is worth mentioning that, to some extend, most of the existing algorithms for testing the stability of multidimensional systems can be viewed as particular variants of cylindrical algebraic decomposition's methods.

On the other hand, critical points based methods basically compute at least one point in each real connected component of a given semi-algebraic set, which is sufficient in our case. Roughly speaking, the basic principle of these methods consists in computing zero-dimensional systems whose real zeros meet each connected component of the original set.

For instance, in the case of an hypersurface $V_{\mathbb{R}} := \{(\alpha_1, \ldots, \alpha_k) \in \mathbb{R}^k \mid P(\alpha_1, \ldots, \alpha_k) = 0, P \in \mathbb{Q}[x_1, \ldots, x_k]\}$, the ancestral method by Seidenberg [26] consists in taking one point $A \in \mathbb{R}^k \setminus V_{\mathbb{R}}$ and to study the critical points of the distance function to A in restriction to $V_{\mathbb{R}}$ which are all included in the algebraic set $\mathcal{C}_A(V_{\mathbb{R}}) := \{\alpha \in V_{\mathbb{R}} \mid \operatorname{grad}_\alpha(P) // \mathrm{A}\alpha\}$, where $\operatorname{grad}_x(P)$ denotes the gradient of P. When no circle of center A belongs to $V_{\mathbb{R}}$ and $V_{\mathbb{R}}$ is smooth (the singular locus is empty), this set of points is finite and meets each semi-algebraic connected component of $V_{\mathbb{R}}$.

Different functions can be used in place of the distance function. For example, when the hypersurface is smooth and compact, the projection with respect to some variable will also have a finite number of critical points and this set of points will meet each real connected component of the latter (see [2], [25]). There are several ways to circonvent the hypothesis (compactness, smoothness): deform the variety to get a compact and smooth one [23], [24], introduce more general notions of critical points [3] or study separately the subsets that might cause troubles (singular locus) [1].

For systems of equations and, more generally systems of inequations, several strategies are proposed by different authors (see [4], [2]). Some are based on the use of sums of squares (to reduce the problem of studying an algebraic set to the problem of studying an hypersurface) [4], infinitesimal deformations (add some variables to avoid singularities or deal with inequations) [4], adapted definitions of critical points (generalized critical values to circumvent the compactness hypothesis), but the basic ideas stay the same.

As already said, critical points methods compute less information than the Cylindrical Algebraic Decomposition, but they are sufficient in our case since we just have to decide if a semi-algebraic set is empty or not. A key advantage of these methods is that they transform the problem into solving an algebraic zero-dimensional system, this transformation being performed within a number of arithmetic operations that is single exponential in the number of variables.

B. Symbolic resolution of univariate polynomials and zero-dimensional polynomial systems

Whatever the preprocessing described in the above section used, the resolution of univariate polynomials and more generally zero-dimensional polynomial systems is the final step. In our case, it is essentially the matter of deciding whether or not a polynomial system admits real solutions.

For polynomial systems with a finite number of solutions, we make use of an additional processing that will turn the problem into a univariate one by computing a univariate parameterization of all the solutions.

Given a zero-dimensional system generating an ideal $I \subset \mathbb{Q}[x_1, \ldots, x_n]$, the quotient algebra $\frac{\mathbb{Q}[x_1,\ldots,x_n]}{I}$ is then a $\mathbb{Q}$-vector space of finite dimension equal to the number of complex zeros of I counted with multiplicities [11]. Assuming that a basis of this finite-dimensional $\mathbb{Q}$-vector space as well as the matrices of multiplication by each variable x_k are known [1], which is the case when I is described by any *Gröbner basis* [11] or alternatively if a *border basis* of I is known [21], a *Rational Univariate Representation* of the zeros of I ($V_{\mathbb{C}}(I)$) can be computed by performing simple linear algebra operations. For instance, the algorithm described in [22] provides univariate polynomials $f, f_1, f_{x_1}, \ldots, f_{x_n} \in \mathbb{Q}[T]$ and a linear form $t := a_1 x_1 + \ldots + a_n x_n$ such that

$$\begin{array}{cccc} \phi_t : & V_{\mathbb{C}}(I) & \approx & V_{\mathbb{C}}(f) \\ & \alpha & \longmapsto & t(\alpha) \\ & \left(\frac{f_{x_1}(\beta)}{f_1(\beta)}, \ldots, \frac{f_{x_n}(\beta)}{f_1(\beta)}\right) & \longleftarrow & \beta \end{array}$$

where $\approx$ stands for a one-to-one correspondence between algebraic varieties and:

$$\begin{cases} V_{\mathbb{C}}(I) := \{\alpha \in \mathbb{C}^n \mid \forall\, P \in I : \; P(\alpha) = 0\}, \\ V_{\mathbb{C}}(f) := \{\beta \in \mathbb{C} \mid f(\beta) = 0\}. \end{cases}$$

This representation defines a bijection between the zeros of I and those of f that preserves the multiplicities and the real zeros.

Note that there exist other algorithms for computing a univariate parametrization of the solutions that do not requires the pre-calculation of Gröbner bases (see for instance [16]). Moreover, for the specific case of systems with only two variables, univariate parameterizations can be efficiently obtained using fast algorithms for computing resultant and subresultant (see [9]).

[1]The matrix of the multiplication by x_k is the matrix of the endomorphism of $\frac{\mathbb{Q}[x_1,\ldots,x_n]}{I}$ which associates to any $h \in \frac{\mathbb{Q}[x_1,\ldots,x_n]}{I}$ the reduction of the polynomial $x_k\, h$ modulo I.

Once such a representation is obtained, computing the real solutions of the system defined by I, or more specifically, deciding whether or not a system has real solutions, resumes to computing the real roots of the univariate polynomial $f \in \mathbb{Q}[T]$, or deciding whether or not this polynomial has real roots. This can be done using classical exact algorithms such as Sturm's sequences or methods based on Descarte's rule of signs [4].

III. TRANSFORMATIONS

As shown in the above section, one can reduce the test of (2) to that of the emptiness of a semi-algebraic set in $\mathbb{R}^{2n}$, which can be achieved using the methods described in the above section. Such an approach presents however the important drawback of doubling the number of variables. Indeed, all the methods for testing the emptiness of a real semi-algebraic set being at least single exponential in the number of variables, doubling this quantity will naturally induces an exponential increase of the computational cost, this makes the test of (2) using these methods very quickly inefficient in practice.

Our approach, which avoids doubling the number of variables, is to consider equivalent conditions of DeCarlo et al. (Theorem 2) rather than (2). As mentioned in the introduction, the n first conditions of Theorem 2 are easy to test since it resumes to apply univariate stability tests. For testing the last condition, i.e. $D(z_1, \ldots, z_n) \neq 0$ for $|z_1| = \ldots = |z_n| = 1$, our idea is to apply transformations that map the torus $\mathbb{T}^n$ to the real space $\mathbb{R}^n$. More precisely, for each complex variable z_k, we can perform a change of variable $z_k := \phi(x_k)$ such that $z_k \in \mathbb{T}$ if and only if $x_k \in \mathbb{R}$. Accordingly, unlike the transformation based on the real coordinates expression, these transformations keep the number of variables unchanged.

A first natural transformation that we can think about is the classical unit circle parametrization. Indeed, it is well-known that the unit circle $\mathbb{T}$ deprived from the point -1 admits as a parametrization $\left(\frac{1-x^2}{1+x^2}, \frac{2\,x}{1+x^2}\right)$ where $x \in \mathbb{R}$. Hence, starting from a polynomial $D(z_1, \ldots, z_n)$, one can replace each variable z_k by the expression $\frac{1-x_k^2}{1+x_k^2} + i\,\frac{2\,x_k}{1+x_k^2}$. This yields a rational fraction in $\mathbb{C}(x_1, \ldots, x_n)$ whose numerator writes as $\mathcal{R}(x_1, \ldots, x_n) + i\,\mathcal{C}(x_1, \ldots, x_n)$, where $\mathcal{R}, \mathcal{C}$ are polynomials in $\mathbb{R}[x_1, \ldots, x_n]$. Accordingly the following result holds.

Proposition 3. *Let $D(z_1, \ldots, z_n) \in \mathbb{R}[z_1, \ldots, z_n]$. One can compute two polynomials $\mathcal{R}(x_1, \ldots, x_n)$ and $\mathcal{C}(x_1, \ldots, x_n)$ such that:*

$$\mathcal{V}_{\mathbb{C}}(D(z_1, \ldots, z_n)) \cap [\mathbb{T} \setminus \{-1\}]^n = \emptyset$$
$$\Longleftrightarrow \mathcal{V}_{\mathbb{R}}(\mathcal{R}(x_1, \ldots, x_n), \mathcal{C}(x_1, \ldots, x_n)) = \emptyset.$$

Remark 4. *If we denote by d_k the degree of $D(z_1, \ldots, z_n)$ with respect to the variable z_k, one can easily remark that the previous transformation yields two polynomials $\mathcal{R}(x_1, \ldots, x_n)$ and $\mathcal{C}(x_1, \ldots, x_n)$ of total degrees bounded by $2 \times \sum_{k=1}^{n} d_k$.*

In order to reduce the growth of degree induced by the above change of variables, one can opt for another classical transformation, the so-called *Möbius transformation*, which we recall the general form in the following definition.

Definition 5. *A Möbius transformation is a rational function $\phi : \overline{\mathbb{C}} \to \overline{\mathbb{C}}$ [2] of the form $\phi(x) = \frac{a\,x+b}{c\,x+d}$, where $a, b, c, d \in \mathbb{C}$ are fixed and $ad - bc \neq 0$. We write formally $\phi\left(-\frac{d}{c}\right) = \infty$ and $\phi(\infty) = \frac{a}{c}$.*

Denoting by $\mathcal{H}$ the class of circles of arbitrary radius in $\overline{\mathbb{C}}$ (this class includes lines which can be considered as circles of infinite radius), the set of Möbius transformations have the property of mapping $\mathcal{H}$ to itself, i.e. each circle in $\overline{\mathbb{C}}$ is mapped to another circle in $\overline{\mathbb{C}}$. In particular, one can easily notice that the transformation $\phi(z) = \frac{z-i}{z+i}$, corresponding to the Möbius transformation where $a = 1$, $b = -i$, $c = 1$ and $d = i$, maps the real line $\overline{\mathbb{R}} := \mathbb{R} \cup \{\infty\}$ to the torus $\mathbb{T}$. Hence, given a polynomial $D(z_1, \ldots, z_n)$, one can replace each variable z_k by $\frac{x_k - i}{x_k + i}$ which yields a rational fraction in $\mathbb{C}(x_1, \ldots, x_n)$ whose numerator writes as $\mathcal{R}(x_1, \ldots, x_n) + i\,\mathcal{C}(x_1, \ldots, x_n)$. Similarly as above, one obtain the following result.

Proposition 6. *Let $D(z_1, \ldots, z_n) \in \mathbb{R}[z_1, \ldots, z_n]$. One can compute two polynomials $\mathcal{R}(x_1, \ldots, x_n)$ and $\mathcal{C}(x_1, \ldots, x_n)$ such that:*

$$\mathcal{V}_{\mathbb{C}}(D(z_1, \ldots, z_n)) \cap [\mathbb{T} \setminus \{1\}]^n = \emptyset$$
$$\Longleftrightarrow \mathcal{V}_{\mathbb{R}}(\mathcal{R}(x_1, \ldots, x_n), \mathcal{C}(x_1, \ldots, x_n)) = \emptyset.$$

Unlike the transformation based on the parametrization of the unit circle, the previous transformation yields polynomials $\mathcal{R}(x_1, \ldots, x_n)$ and $\mathcal{C}(x_1, \ldots, x_n)$ of total degree bounded only by $\sum_{k=1}^{n} d_k$.

Hence, starting from a polynomial $D(z_1, \ldots, z_n) \in \mathbb{R}[z_1, \ldots, z_n]$, one can test that $D(z_1, \ldots, z_n)$ does not have complex zeros on $[\mathcal{T} \setminus \{1\}]^n$ by first computing the polynomials $\mathcal{R}(x_1, \ldots, x_n)$ and $\mathcal{C}(x_1, \ldots, x_n)$ and then checking that the polynomial system $\{\mathcal{R}(x_1, \ldots, x_n) = \mathcal{C}(x_1, \ldots, x_n) = 0\}$ does not have any solution in $\mathbb{R}^n$.

However, in order to check that the polynomial $D(z_1, \ldots, z_n)$ satisfies (3), the above test is not sufficient since it excludes the points on the torus that have at least one coordinate equal to one. One also needs to check that the polynomial $D(z_1, \ldots, z_n)$ does not vanish at any of these points. To do so, we proceed in the following way. Starting from $D(z_1, \ldots, z_n)$, in a first stage, we compute

[2] $\overline{\mathbb{C}}$ denotes the extended complex plane, so that $\overline{\mathbb{C}} := \mathbb{C} \cup \{\infty\}$.

the polynomials $D_k(z_1, \ldots, z_{k-1}, z_{k+1}, \ldots, z_n) := D(z_1, \ldots, z_{k-1}, 1, z_{k+1}, \ldots, z_n)$ for $k = 1, \ldots, n$. To each D_k, we apply the above Möbius transformation followed by the test of Proposition 6. Similarly as above, this test allows us to check whether or not each D_k does not have complex zeros on $[\mathbb{T} \setminus \{1\}]^{n-1}$. But we still need to check that D_k does not vanish on the excluded points (points that have at least one coordinate in $\{z_1, \ldots, z_{k-1}, z_{k+1}, \ldots, z_n\}$ equal to one), this can be done (in a second stage) in the same way as above by considering the polynomials D_{k_l} computed from each D_k after substituting the variable z_l by one. Proceeding recursively until obtaining polynomials in one variable of the form $D(1, \ldots, 1, z_k, 1, \ldots, 1)$ allows us to check that $D(z_1, \ldots, z_n)$ does not vanish at any point of the torus.

Note that at each stage m of the algorithm, the set of polynomials we need to consider are exactly the polynomials obtained from $D(z_1, \ldots, z_n)$ after substituting m variables by 1. From this observation, we obtain the following algorithm.

Algorithm 1 Intersection with the torus

procedure `IntersectionEmpty`$(D(z_1, \ldots, z_n))$ $\triangleright$ return true if $D(z_1, \ldots, z_n)$ satisfies (3)
  for $k = 0$ to $n - 1$ **do**
    Compute S_k, the set of polynomials obtained from $D(z_1, \ldots, z_n)$ after substituting k variables by 1
    for each D_k in S_k **do**
      $\{\mathcal{R}, \mathcal{C}\} =$ `Möbius_transform`(D_k)
      if $\mathcal{V}_{\mathbb{R}}(\{\mathcal{R}, \mathcal{C}\}) \neq \emptyset$ **then**
        return False
      end if
    end for
  end for
  return True
end procedure

We finish this section by summarizing our n-D stability test. Given a polynomial $D(z_1, \ldots, z_n)$, the latter proceeds in two steps.

Step 1: For each $k = 1, \ldots, n$, determine whether $D(1, \ldots, z_k, \ldots, 1)$ is stable. If not, then return `False` and exit (according to Theorem 2, the system is not stable).

Step 2: If the test of Step 1 does not return `False`, then return *IntersectionEmpty*$(D(z_1, \ldots, z_n))$.

Remark 7. *One can notice that, in Algorithm 1, the polynomials considered at the stage $n - 1$ correspond to $D(1, \ldots, z_k, \ldots, 1)$ with $k = 1, \ldots, n$. Since these polynomials are checked, in step 1 of the stability test, to be devoid from zero in $\mathbb{D}$, one may skip testing them in Algorithm 1 and stop the main loop at the stage $n-2$.*

IV. Examples and experiment

We have implemented our stability test in `Maple`. This procedure named `IsStable` takes as input a polynomial defining the denominator of a transfer function and returns true if this polynomial satisfies the conditions of Theorem 2 and false otherwise. The n first conditions of this theorem (step 1) are tested using the classical 1-D Bistritz test [5] which we have implemented in `Maple` while the last condition (step 2) is tested using Algorithm 1. For testing the emptiness of a real algebraic set, we use the `Maple` routine `HasRealRoots` of the package `RootFinding` which takes as input an algebraic system and return true if and only if the latter has real solutions [3].

In the following, we first illustrate the two steps of our procedure on a 2-D and 3-D example and then, report in Table I the running times of the latter tested on a set of polynomials [4].

Example 1. We consider the 2-D polynomial $D(z_1, z_2) = (12 + 10\,z_1 + 2\,z_1^2) + (6 + 5\,z_1 + z_1^2)\,z_2$, which appeared in several papers dealing with stability tests ([29], [19]) and is known to be devoid from zero in the unit bi-disc $\mathbb{D}^2$.

The first step of our procedure consists in checking that the polynomials $D(z_1, 1) = 3\,z_1^2 + 15\,z_1 + 18$ and $D(1, z_2) = 12\,z_2 + 24$ are stable which can be done simply by looking at their roots ($-3, -2$ and -2).

In the second step, we run Algorithm 1 on $D(z_1, z_2)$. As we have already checked that $D(z_1, 1)$ and $D(1, z_2)$ are stable, the only polynomial we have to consider is $D(z_1, z_2)$ itself. The Möbius transformation of this polynomial yields the system

$$\{36\,x_1^2\,x_2 - 10\,x_1 - 6\,x_2 = 0,\ 12\,x_1^2 + 30\,x_1\,x_2 - 2 = 0\}$$

which does not admit any real solution.

Example 2. We consider the 3-D polynomial $D(z_1, z_2, z_3) = (z_1^2 + z_2^2 + 4)\,(z_1 + z_2 + z_3 + 5)$, which is known to be devoid from zero in $\mathbb{D}^3$ [19].

Our procedure first checks that the stability of:

$$D(z_1, 1, 1) = (z_1^2 + 5)\,(z_1 + 7),$$
$$D(1, z_2, 1) = (z_2^2 + 5)\,(z_2 + 7),$$
$$D(1, 1, z_3) = 6\,z_3 + 42.$$

Then, Algorithm 1 is run on $D(z_1, z_2, z_3)$ which yields the following polynomials

$$D(z_1, z_2, z_3) = (z_1^2 + z_2^2 + 4)\,(z_1 + z_2 + z_3 + 5),$$
$$D(z_1, 1, z_3) = (z_1^2 + 5)\,(z_1 + z_3 + 6),$$
$$D(1, z_2, z_3) = (z_2^2 + 5)\,(z_2 + z_3 + 6),$$
$$D(z_1, z_2, 1) = (z_1^2 + z_2^2 + 4)\,(z_1 + z_2 + 6),$$

[3] This routine is based on the computation of the set of critical points of a function restricted to the algebraic variety of the system.

[4] The experiments have been conducted on 1.90 GHz 3-Core Intel i3-3227U with 3MB of L3 cache under Linux platform.

as well as a set of systems obtained after applying the Möbius transformation to each of them. For this example, the main computation is devoted to deciding if the following system has real zeros or not:

$$\left\{\begin{array}{l} 48\,x_1^3\,x_2^3\,x_3 - 72\,x_1^3\,x_2^2 - 96\,x_1^3\,x_2\,x_3 - 72\,x_1^2\,x_2^3 \\ -184\,x_1^2\,x_2^2\,x_3 - 96\,x_1\,x_2^3\,x_3 + 24x_1^3 + 120\,x_1^2\,x_2 \\ +72\,x_1^2\,x_3 + 120\,x_1\,x_2^2 + 176\,x_1\,x_2\,x_3 + 24\,x_2^3 \\ +72\,x_2^2\,x_3 - 40\,x_1 - 40\,x_2 - 24\,x_3 = 0, \\ 36\,x_1^3\,x_2^3 + 100\,x_1^3\,x_2^2\,x_3 + 100\,x_1^2\,x_2^3\,x_3 - 68\,x_1^3\,x_2 \\ -36\,x_1^3\,x_3 - 124\,x_1^2\,x_2^2 - 180\,x_1^2\,x_2\,x_3 - 68\,x_1\,x_2^3 \\ -180\,x_1\,x_2^2\,x_3 - 36\,x_2^3\,x_3 + 44\,x_1^2 + 116\,x_1\,x_2 \\ +68\,x_1\,x_3 + 44\,x_2^2 + 68\,x_2\,x_3 - 12 = 0. \end{array}\right.$$

Given as inputs of `HasRealRoots`, this system and the ones corresponding to the other polynomials turn out to be devoid from real zeros.

The following table shows the average running times in seconds for `IsStable` on random polynomials in 2 and 3 variables with rational coefficients. The algorithm was also able to solve problems in 4 variables with degree 7 (dense polynomials with 75% of non-zero terms) and 12 (sparse polynomials with 25% of non-zero terms) in less than 10 mins.

nb var \ degree		3	5	8	10	12
2	sparse	0.1	0.12	0.3	0.5	4
	dense	0.1	0.2	0.9	3.0	12
3	sparse	0.15	0.3	0.8	3	15
	dense	1	2	12	53	342

TABLE I: CPU times in seconds of `IsStable` run on random polynomials in 2 and 3 variables with rational coefficients.

REFERENCES

[1] P. Aubry, F. Rouillier, and M. Safey El Din. Real solving for positive dimensional systems. *Journal of Symbolic Computation*, 34(6):543 – 560, 2002.

[2] B. Bank, M. Giusti, J. Heintz, and G. Mbakop. Polar varieties and efficient real elimination. *Mathematische Zeitschrift*, 238(1):115–144, 2001.

[3] B. Bank, M. Giusti, J. Heintz, and L. M. Pardo. Generalized polar varieties: Geometry and algorithms. *Journal of complexity*, 21(4):377–412, 2005.

[4] S. Basu, R. Pollack, and M.-F. Roy. *Algorithms in Real Algebraic Geometry*, volume 10 of *Algorithms and Computation in Mathematics*. Springer-Verlag, 2nd edition, 2006.

[5] Y. Bistritz. Zero location with respect to the unit circle of discrete-time linear system polynomials. *Proceedings of the IEEE*, 72(9):1131–1142, 1984.

[6] Y. Bistritz. An immittance-type stability test for 2-d digital filters. In *Signals, Systems and Computers, 1994. 1994 Conference Record of the Twenty-Eighth Asilomar Conference on*, volume 2, pages 918–922 vol.2, Oct 1994.

[7] Y. Bistritz. Zero location of polynomials with respect to the unit-circle unhampered by nonessential singularities. *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, 49(3):305–314, 2002.

[8] Y. Bistritz. Testing stability of 2-D discrete systems by a set of real 1-D stability tests. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 51(7):1312–1320, July 2004.

[9] Y. Bouzidi. *Solving bivariate algebraic systems and topology of plane curves*. PhD thesis, Université de Lorraine, 2014.

[10] G. Collins. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. *Springer Lecture Notes in Computer Science 33*, 33:515–532, 1975.

[11] D. Cox, J. Little, and D. O'Shea. *Ideals, varieties, and algorithms an introduction to computational algebraic geometry and commutative algebra*. Undergraduate texts in mathematics. Springer-Verlag New York-Berlin-Paris, 1992.

[12] R. A. Decarlo, J. Murray, and R. Saeks. Multivariable Nyquist theory. *International Journal of Control*, 25(5):657–675, 1977.

[13] B. Dumitrescu. Stability test of multidimensional discrete-time systems via sum-of-squares decomposition. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 53(4):928–936, April 2006.

[14] B. Dumitrescu. LMI stability tests for the Fornasini-Marchesini model. *Signal Processing, IEEE Transactions on*, 56(8):4091–4095, 2008.

[15] P. Fu, J. Chen, and S.-I. Niculescu. Author's reply to a counterexample to generalized eigenvalue-based stability tests for 2-d linear systems: Necessary and sufficient conditions. *Automatica*, 46(1):236–237, 2010.

[16] M. Giusti, G. Lecerf, and B. Salvy. A Gröbner free alternative for solving polynomial systems. *J. of Complexity*, 17(1):154–211, 2001.

[17] X. Hu and E. Jury. On two-dimensional filter stability test. *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, 41(7):457–462, Jul 1994.

[18] E. I. Jury. *Theory and Application of the z-Transform Method*, volume 3. Wiley New York, 1964.

[19] L. Li, L. Xu, and Z. Lin. Stability and stabilisation of linear multidimensional discrete systems in the frequency domain. *International Journal of Control*, 86(11):1969–1989, 2013.

[20] M. Marden. The geometry of the zeros of a polynomial in the complex plane. In *Amer. Math. Soc*, 1949.

[21] B. Mourrain. A new criterion for normal form algorithms. In *Applied algebra, algebraic algorithms and error-correcting codes*, pages 430–442. Springer, 1999.

[22] F. Rouillier. Solving zero-dimensional systems through the rational univariate representation. *J. of Applicable Algebra in Engineering, Communication and Computing*, 9(5):433–461, 1999.

[23] F. Rouillier, M.-F. Roy, and M. Safey El Din. Finding at least one point in each connected component of a real algebraic set defined by a single equation. *Journal of Complexity*, 16(4):716–750, 2000.

[24] M. Safey El Din. Finding sampling points on real hypersurfaces is easier in singular situations. *MEGA (Effective Methods in Algebraic Geometry) Electronic proceedings*, 2005.

[25] M. Safey El Din and E. Schost. Polar varieties and computation of one point in each connected component of a smooth real algebraic set. In *Proceedings of ISSAC 2003*, pages 224–231. ACM, 2003.

[26] A. Seidenberg. A new decision method for elementary algebra. *Annals of Mathematics*, pages 365–374, 1954.

[27] I. Serban and M. Najim. Multidimensional systems: BIBO stability test based on functional Schur coefficients. *Signal Processing, IEEE Transactions on*, 55(11):5277–5285, 2007.

[28] M. Strintzis. Tests of stability of multidimensional filters. *IEEE Transactions on Circuits and Systems*, 24:432–437, 1977.

[29] L. Xu, M. Yamada, Z. Lin, O. Saito, and Y. Anazawa. Further improvements on Bose's 2-D stability test. *International Journal of Control Automation and Systems*, 2:319–332, 2004.