

## Back to the Roots: Polynomial System Solving, Linear Algebra, Systems Theory

Philippe Dreesen<sup>\*,\*\*</sup> Kim Batselier<sup>\*,\*\*</sup> Bart De Moor<sup>\*,\*\*</sup>

<sup>\*</sup> *KU Leuven, Department of Electrical Engineering ESAT/SCD, Kasteelpark Arenberg 10-2446, B-3000 Leuven, Belgium.*

<sup>\*\*</sup> *KU Leuven, IBBT Future Health Department.*

*e-mail: {philippe.dreesen, bart.demoor}@esat.kuleuven.be*

---

**Abstract:** Multivariate polynomial system solving and polynomial optimization problems arise as central problems in many systems theory, identification and control settings. Traditionally, methods for solving polynomial equations have been developed in the area of algebraic geometry. Although a large body of literature is available, it is known as one of the most inaccessible fields of mathematics. In this paper we present a method for solving systems of polynomial equations employing numerical linear algebra and systems theory tools only, such as realization theory, SVD/QR, and eigenvalue computations. The task at hand is translated into the realm of linear algebra by separating coefficients and monomials into a coefficient matrix multiplied with a basis of monomials. Applying realization theory to the structure in the monomial basis allows to find all solutions of the system from eigenvalue computations. Solving a polynomial optimization problem is shown to be equivalent to an extremal eigenvalue problem. Relevant applications are found in identification and control, such as the global optimization of structured total least squares problems.

Keywords: Eigenvalues, Eigenvectors, Realization Theory, Systems Theory, Global Optimization, Linear Algebra, Linearization, Matrix Formulation, Multivariate Polynomials

---

### 1. INTRODUCTION

Polynomial system solving and polynomial optimization problems arise as central tasks in system identification and control settings. In this paper, we address the polynomial root-finding problem, that is, given a system of polynomial equations, find all solutions for which the equations hold. Polynomial systems are classically solved by local root-finding methods, such as the Newton method, but they have the disadvantage that they only guarantee to find a solution in the vicinity of the initial starting point. Computer algebra methods, such as Gröbner bases (Buchberger, 1965) are often intractable and numerically infeasible due to the symbolic nature of the calculations involved. In this paper we explore a different approach by looking at the problem from the linear algebra perspective, as in the works by Lazard (1981); Emiris and Mourrain (1999); Stetter (2004), among others. Although the connections between polynomials and linear algebra are largely unknown, it can be shown that there are many interesting links with linear algebra underlying the task at hand. Translating the problem into the world of numerical linear algebra makes available a variety of numerically reliable techniques (Golub and Van Loan, 1996).

Polynomial optimization problems are encountered when one is interested in finding the best solution to a problem, and can be addressed in the same way: the Lagrange multipliers method provides the necessary conditions for optimality as a system of polynomial equations. An instance of a polynomial optimization problem is the least-squares approximation of a given full-rank matrix by a

matrix of lower rank. This problem is traditionally solved by computing the SVD (Van Huffel and Vandewalle, 1991). When additional structure constraints are imposed on the matrices involved, the resulting problem is called the structured total least squares (STLS) problem. The STLS problem is a central task in systems theory, identification and control, underlying many modeling problems, such as errors-in-variables system identification, approximate realization and model reduction (see Markovsky (2008) for an elaborate survey on the problem and its applications). Formally the STLS problem is written as

$$\min_{B,v} \|A - B\|_F^2, \text{ s.t. } Bv = 0, B \text{ structured}, v^T v = 1,$$

where  $A$  is a given (full-rank) data matrix with a given structure which is to be retained in the low-rank approximation  $B$ . Examples of specific matrix structures which are often encountered in system identification and control are Hankel, Toeplitz, Sylvester, known sparsity patterns, etc.

In Section 2 we will review some well-known connections between univariate polynomial root-finding, linear algebra and systems theory. Section 3 discusses the root-finding method for the generic multivariate problem, and Section 4 discusses the degenerate case. Finally, the method will be applied to a STLS problem in Section 5 and we will point out some open problems and future research directions in Section 6.

## 2. UNIVARIATE POLYNOMIALS: LINEAR ALGEBRA AND REALIZATION THEORY

Let us start with a brief review of the well-known connections between univariate polynomial root-finding, linear algebra and systems theory. First, the eigenvalues of an  $n \times n$  matrix  $A$  correspond to the  $n$  roots of its characteristic polynomial, which is defined as  $p(\lambda) = \det(A - \lambda I_n)$ . Second, the converse is true: with every univariate polynomial  $f(x) = x^n + a_{n-1}x^{n-1} + \dots + a_0$ , a matrix can be associated of which the eigenvalues correspond to the roots of  $f(x)$ . The roots of  $f(x)$  can be obtained as the eigenvalues of the companion matrix, as shown in

$$\left[ \begin{array}{c|ccc} 0_{(n-1) \times 1} & I_{n-1} & & \\ \hline -a_0 & -a_1 & \dots & -a_{n-1} \end{array} \right] \begin{bmatrix} x^0 \\ \vdots \\ x^{n-1} \end{bmatrix} = x \begin{bmatrix} x^0 \\ \vdots \\ x^{n-1} \end{bmatrix}.$$

Note that this expression bears a lot of resemblance to realization theory. Third, checking whether two univariate polynomials  $f_1(x) = a_r x^r + a_{r-1} x^{r-1} + \dots + a_0$  and  $f_2(x) = b_s x^s + b_{s-1} x^{s-1} + \dots + b_0$  have a common root is equivalent with investigating the determinant of the Sylvester matrix, a certain structured square matrix built from the coefficients of the polynomials  $f_1$  and  $f_2$ . Multiplying  $f_1(x)$  and  $f_2(x)$  with powers of  $x$  and setting the results to zero yields a square system of linear equations, represented as the matrix equation

$$\begin{matrix} s \text{ rows} \\ r \text{ rows} \end{matrix} \left\{ \begin{array}{c} \left[ \begin{array}{cccc} a_0 & a_1 & \dots & a_r \\ & \ddots & \ddots & \ddots \\ & & a_0 & a_1 & \dots & a_r \\ \hline b_0 & b_1 & \dots & b_s \\ & \ddots & \ddots & \ddots \\ & & b_0 & b_1 & \dots & b_s \end{array} \right] \begin{bmatrix} x^0 \\ x^1 \\ \vdots \\ x^{r+s-1} \end{bmatrix} = 0. \end{array} \right.$$

The matrix in this expression is called the Sylvester matrix of  $f_1$  and  $f_2$  and is of size  $(r + s) \times (r + s)$ , having a zero determinant when  $f_1(x)$  and  $f_2(x)$  have a common root, which can easily be understood from its construction.

Inspired by these well-known links in the univariate case, we explore the generalization to the multivariate case and develop a conceptually simple method for multivariate root-finding which illustrates that many interesting links and results can be obtained from a straightforward linear algebra approach, combined with notions from realization theory.

## 3. SOLVING MULTIVARIATE POLYNOMIALS: LINEAR ALGEBRA VIEWPOINT

Although many interesting results were obtained in the works of Sylvester (1853) and Macaulay (1902), the importance for linear algebra was (re)discovered only recently by Lazard (1981), Stetter (2004), and Emiris and Mourrain (1999), among others. As opposed to much of the existing literature in (computational) algebraic geometry, which is mostly centered on Gröbner basis algorithms (Buchberger, 1965), we strongly think that there is a need for obtaining insights and developing novel algorithms to bridge the fields of algebraic geometry, (numerical) linear algebra and systems theory. Our aim is to develop a framework making such methods accessible to applied mathematicians and systems and control scientists and engineers without

knowledge of algebraic geometry. Although many of the presented ideas are already known in some form, the main purpose of this contribution is to provide an accessible method showing the clear connections between multivariate polynomial system solving, linear algebra and systems theory (e.g., realization theory), and illustrate their relevance for system identification and control applications.

Consider a system of  $n$  polynomial equations  $f_i$  in  $n$  unknowns  $x_1, \dots, x_n$ ,

$$\begin{cases} f_1(x_1, \dots, x_n) = 0, \\ \vdots \\ f_n(x_1, \dots, x_n) = 0, \end{cases} \quad (1)$$

having degrees  $d_1, \dots, d_n$ , and define the maximal degree as  $d_o = \max(d_1, \dots, d_n)$ . We assume that we are dealing with a generic (or non-deficient) system, meaning that (1) has a solution set which is zero-dimensional and the number of solutions of is given by the Bézout number  $m_B$ . An instance for which the genericity assumption holds is when all possible monomials occur and corresponding coefficients are chosen at random.

*Theorem 1.* (Bézout number  $m_B$  (Cox et al., 2005)). A generic system (1) defines a zero-dimensional solution set, consisting of  $m_B$  single roots, where  $m_B$  is equal to the product of the degrees, called the Bézout number

$$m_B = \prod_{i=1}^n d_i. \quad (2)$$

For now we assume that (1) obeys the genericity assumption. In practical cases, there will often be fewer solutions; ways to detect and handle the non-generic case are discussed in Section 4.

The generic version of the root-finding algorithm proceeds in three steps. First a coefficient matrix is constructed from the input equations in much the same way as the Sylvester matrix is constructed in the univariate case. Second a numerical basis for the nullspace of the coefficient is computed. Third the specific structure in the monomial basis is employed to translate the root-finding problem into an eigenvalue problem.

The polynomial system is written as a system of linear equations by separating the coefficients and the monomials. This is achieved by constructing a suitably sized Macaulay-like coefficient matrix, which turns out to contain all the information necessary to find the solutions of the set of equations.

*Proposition 2.* (Construction  $M(d_o)$ ).

Since the maximal degree occurring in the input equations (1) is  $d_o$ , each equation can be represented as a row vector  $m_i^T$  of coefficients multiplied with a vector  $k(d_o)$  containing all possible monomials with degrees from 0 up to  $d_o$ , resulting in  $m_i^T k(d_o) = 0$  for  $i = 1, \dots, n$ . The elements of  $k(d_o)$  are ordered by a user-chosen graded monomial ordering scheme<sup>1</sup> in compliance with the order in which the corresponding coefficients occur in  $m_i^T$ .

Observe that the equations  $f_i$  for which  $d_i < d_o$  can be ‘shifted’ internally: by multiplying  $f_i$  by monomials of

<sup>1</sup> A graded monomial ordering scheme means that the monomials are ordered by increasing degrees.

degrees 1 up to  $d_o - d_i$ , new, so-called ‘shifted’ equations of degree  $\leq d_o$  are obtained, that are adjoined to the input system. This leads to a linear algebra representation of the polynomial system as  $M(d_o)k(d_o) = 0$ .

*Example 3.* Consider a simple system of two polynomial equations in two unknowns

$$\begin{cases} f_1(x_1, x_2) = 2x_1^2 - x_2 = 0, \\ f_2(x_1, x_2) = 3x_1 - 4x_2 + 5 = 0. \end{cases}$$

Since  $f_1$  is of degree 2, and  $f_2$  is of degree 1,  $f_2$  is shifted internally. Following the construction of above, we write the input system as a  $4 \times 6$  coefficient matrix multiplying a monomial basis vector,

$$M(d_o)k(d_o) = 0 \Leftrightarrow \begin{bmatrix} 0 & 0 & -1 & 2 & 0 & 0 \\ 5 & 3 & -4 & 0 & 0 & 0 \\ 0 & 5 & 0 & 3 & -4 & 0 \\ 0 & 0 & 5 & 0 & 3 & -4 \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_1^2 \\ x_1x_2 \\ x_2^2 \end{bmatrix} = 0,$$

where  $M(d_o)$  is a coefficient matrix containing in its rows the coefficients of both the original equations  $f_1$  and  $f_2$  and the shifted versions of  $f_2$ .

Observe that each solution of (1) defines a vector with the specific structure of  $k(d_o)$  lying in the nullspace of the coefficient matrix  $M(d_o)$ , which is obvious from the construction of the coefficient matrix and can easily be verified for Example 3. Conversely, it turns out to be possible to find the solutions from a numerical basis of the nullspace of the coefficient matrix. However, in general, the initial coefficient matrix  $M(d_o)$  built from the input equations (and their internally shifted versions) does not suffice for finding the solutions of the system. In order to fully capture the nature of the problem into a linear system of equations, it turns out to be necessary to extend the system of equations beyond the ‘internal shifts’.

*Proposition 4.* (Construction coefficient matrix  $M(d)$ ).

Starting from the construction of the initial matrix representation of the system of polynomials, as obtained above, the same procedure of multiplying the input equations with monomials, yields new equations, thereby increasing the maximal degree of the working set of equations to  $d > d_o$ . All possible equations that can be constructed in this way are included in the matrix-vector representation of (1). Adjoining the newly generated equations naturally leads to the extension of the coefficient matrix  $M(d_o)$  to  $M(d)$  and the basis of monomials  $k(d_o)$  to  $k(d)$ . Finally the system is represented as a matrix equation  $M(d)k(d) = 0$ .

The evaluation of the – for the time being – unknown solutions of the system in the monomial vectors constitutes a generalized Vandermonde matrix  $K(d)$ .

*Definition 5.* (Canonical Nullspace  $K(d)$ ).

Each solution of the system (1) which is represented as the matrix equation  $M(d)k(d) = 0$  defines a vector in the nullspace of the coefficient matrix  $M(d)$ . These vectors are obtained by evaluating the monomial vector  $k(d)$  at a certain solution, denoted by  $k^{(i)}(d)$ , for  $i = 1, \dots, m_B$ . The generalized Vandermonde matrix obtained by evaluating the  $m_B$  monomial vectors at the solutions, i.e.,  $K(d) = [k^{(1)}(d), \dots, k^{(m_B)}(d)]$ , is called the canonical nullspace  $K(d)$ . (The specific order in which the  $m_B$  vectors are placed in  $K(d)$  is not important and can be chosen freely).

Before moving on to the root-finding, we will review a fundamental theorem which prescribes up to which degree the coefficient matrix  $M(d)$  has to be constructed in order to be able to retrieve the number of roots of the system (1) from the nullspace of  $M(d)$ .

*Theorem 6.* ( $m_B = \text{corank}(M(d^*))$ ) (Macaulay, 1902)).

For a generic system (1), the evaluations of the corank<sup>2</sup> of  $M(d)$  for increasing  $d = d_o, d_o + 1, d_o + 2, \dots$  stabilize at an integer. An upper bound for the required degree at which the corank converges was derived by Macaulay (1902) as

$$d^* = \sum_{i=1}^n d_i - n + 1.$$

The corank of  $M(d^*)$  corresponds to the number of solutions of (1), that is,  $\text{corank}(M(d^*)) = m_B$ .

For the purpose of notational convenience, the suitably-sized matrix  $M(d^*)$ , the corresponding monomial basis vectors  $k(d^*)$  and the canonical nullspace  $K(d_*)$  are henceforth usually denoted  $M$ ,  $k$  and  $K$ , respectively.

To retrieve the roots themselves, we need to reconstruct the monomial structure in  $K$ , which is for the time being unknown. Starting from a computed basis for the nullspace of  $M$ , which can be obtained from a SVD or QR decomposition, we will make use of the multiplication structure present in the monomial vectors  $k$ .

*Proposition 7.* (Shifting  $k$  with monomial or polynomial).

Consider a monomial vector  $k = k(d)$  in which all possible monomials of degrees 0 up to  $d$  occur. Such a vector  $k$  obeys a specific shift structure: multiplication of  $k$  by a monomial  $x^\gamma = x_1^{\gamma_1} x_2^{\gamma_2} \dots x_n^{\gamma_n}$  with degree  $|\gamma| = \sum_{i=1}^n \gamma_i$  maps the entries of  $k$  of degrees 0 up to  $d - |\gamma|$  to entries in  $k$  of degrees  $|\gamma|$  up to  $d$ . The multiplication with  $x^\gamma$  can alternatively be expressed by means of row selection matrices that operate on  $k$  as  $S_1 k x^\gamma = S_2 k$ , where  $S_1$  selects all monomials of degrees 0 up to  $d - |\gamma|$  and  $S_2$  selects the rows onto which the monomials  $S_1 k$  are mapped by multiplication by  $x^\gamma$ .

More generally, one can multiply a monomial vector  $k$  with a polynomial equation  $\sigma$  as

$$S_1 k \sigma = S_2 k,$$

where now  $S_2$  takes linear combinations of rows of  $k$  instead of selecting rows only.

*Example 8.* Consider a monomial vector  $k$  of degree three in two variables, given by

$$k = (1, x_1, x_2, x_1^2, x_1x_2, x_2^2, x_1^3, x_1^2x_2, x_1x_2^2, x_2^3)^T,$$

and shift function  $\sigma = 3x_1^2 + 2x_2^2$ . We have

$$(1, x_1, x_2)^T (3x_1^2 + 2x_2^2) = (3x_1^2 + 2x_2^2, 3x_1^3 + 2x_1x_2^2, 3x_1^2x_2 + 2x_2^3)^T$$

or,

$$S_1 k \sigma = S_2 k,$$

where

$$S_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \text{and}$$

<sup>2</sup> We define the corank to be the dimension of the right nullspace of the matrix  $M$ .

$$S_2 = \begin{pmatrix} 0 & 0 & 0 & 3 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 2 \end{pmatrix}.$$

Proposition 7 can alternatively be applied to the canonical nullspace  $K$  at once, leading to

$$S_1 K D_\sigma = S_2 K,$$

where  $D_\sigma$  is a diagonal matrix containing a user-chosen shift function  $\sigma$ . Due to the diagonal structure of  $D_\sigma$  containing the  $m_B$  evaluations of the shift function  $\sigma$ , each of the  $m_B$  vectors in  $K$  are considered separately.

*Remark 9.* Typical choices for the shift function  $\sigma$  are the solution components  $x_1, \dots, x_n$  themselves, or, in the case of a polynomial optimization problem, the objective function (cf. Remark 16). The specific choice of the shift function  $\sigma$  is arbitrary and will in most cases not affect the remainder of the root-finding method, as long as one avoids to end up with having so-called multiple solutions, that is, a shift function taking the same value for several solutions. An easy way to avoid multiple solutions is by taking a random linear combination of all unknowns  $x_i$ .

*Theorem 10.* (Root-finding as eigenvalue problem). After constructing  $M = M(d^*)$  with  $\text{corank}(M) = m_B$ , the canonical nullspace  $K$  is not available directly. Instead a basis for the nullspace of  $M$  is computed as  $Z$  with  $K = ZV$  where  $V$  is a nonsingular matrix representing a linear transformation. Combining  $S_1 K D_\sigma = S_2 K$  from Proposition 7 with  $K = ZV$  yields the generalized eigenvalue problem

$$S_1 Z (V D_\sigma V^{-1}) = S_2 Z.$$

*Corollary 11.* (Generic root-finding).

Let  $B$  be the  $m_B \times m_B$  matrix containing the first  $m_B$  linearly independent rows of  $Z$ , or  $B = S_1 Z$ , and let  $W = ZB^{-1}$ . The evaluations of the user-chosen shift function  $\sigma$  at each of the  $m_B$  solutions of (1) are found as the eigenvalues of

$$A = S_2 W.$$

Once the eigenvalue decomposition is computed as  $A = V D_\sigma V^{-1}$ , the generalized Vandermonde structure of  $K$  can be reconstructed such that the components  $x_i$  and how they match up with one another is revealed.

*Corollary 12.* (Reconstructing  $K$ ).

By computing  $WV$ , the generalized Vandermonde structure of the canonical nullspace can be reconstructed. Each column of  $WV$  now lies in the nullspace of  $M$  and obeys to the shift structure in  $k$ , however, the columns of  $WV$  have yet to be rescaled such that the first entries equal one, as to reconstruct  $K$ . The entries corresponding to  $x_1, x_2, \dots, x_n$  can now be read off directly from the reconstruction of  $K$ .

*Example 13.* Example 3 is continued. The required degree is determined as  $d^* = 2$  and we construct the coefficient matrix  $M = M(2)$ , of size  $4 \times 6$ , with  $\text{corank}(M) = 2$ . A basis for the nullspace of  $M$  is computed using an SVD, and following the root-finding method of above, the solutions  $(x_1, x_2)$  are found as  $(1.00000, 2.00000)$  and  $(-0.62500, 0.78125)$ .

#### 4. NON-GENERIC POLYNOMIAL SYSTEMS

In the previous section, it was assumed that (1) has  $m_B = \prod_i d_i$  solutions. In practice, a system of polynomial

equations often has fewer solutions than predicted by the Bézout number (2). This deficiency can be due to the existence of multiple roots, zero coefficients in the system (1) or algebraic relations among the coefficients, leading to the existence of ‘spurious’ solutions, which are undesired in practical cases.

##### 4.1 Multiple Solutions

For multiple roots, the nullspace of  $M$  does not only contain the canonical vectors  $k$  evaluated at the solutions, but also linear combinations of differential forms where the monomial basis vectors  $k$  are differentiated with respect to the unknowns  $x_i$ . A full analysis and characterization is described by Möller and Stetter (1995) and Dayton and Zeng (2005). Except for the well-known loss of accuracy in computing multiple eigenvalues, multiple roots pose no problem for our method, and the algorithm from the previous section returns all roots correctly.

*Remark 14.* The part of the generic algorithm which does not work in the case of multiple roots is the reconstruction of the canonical nullspace  $K$ : once the individual solution components are computed, they should be matched by performing an exhaustive search over the  $n$  components of the roots.

##### 4.2 Solutions at Infinity

A classical result in algebraic geometry states that for a zero-dimensional solution set, the Bézout number  $m_B$  corresponds to the number of solutions in projective space (counted with multiplicity). Only in the generic case, the number of (affine) solutions corresponds to the Bézout number. In most practical cases there are also solutions which are called the solutions at infinity. There are two ways of dealing with solutions at infinity, namely 1) by analyzing the problem in projective space, and 2) by performing an affine root-count.

First, it is possible to analyse the task at hand in projective space. Projective geometry incorporates the points at infinity as regular points, after which they can easily be identified and discarded (Cox et al., 2007, 2005).

The second method for dealing with solutions at infinity is to perform a root-counting method to obtain the number of affine solutions after which the roots at infinity can be separated from the affine roots. Existing algebraic methods addressing this problem, such as the BKK bound, or the multihomogeneous Bézout number are described in Cox et al. (2007, 2005).

It turns out to be possible to perform the affine root-count via our matrix method as well! Homogenizing the equations easily shows that the roots at infinity are characterized by the highest degree terms in the equations: indeed, denoting the homogenization variable as  $x_0$ , the roots at infinity have  $x_0 = 0$ , so all lower degree terms drop out of the equations. By monitoring the indices of the linearly independent rows of  $Z(d)$  as the degree  $d$  in  $M(d)$  increases, one can easily determine which indices correspond to the affine roots, as they remain fixed; the indices corresponding with the roots at infinity always occur in the highest-degree blocks. An alternative way of

performing the affine root-counting is described in Batse-lier et al. (2011).

*Proposition 15.* (Affine root-counting).

Let  $B$  be composed of the first  $\text{corank}(M)$  linearly independent rows in  $Z$  as  $B = Z(v, \cdot)$ , where  $v$  contains the row indices of the linearly independent rows of  $Z$ . This can alternatively be written as  $B = S_1 Z$ . By inspecting the indices of the linearly independent rows of  $Z(d)$  as  $d$  increases, one separates the  $m_a$  affine roots from the roots at infinity, where the row indices corresponding to the affine solutions are denoted  $v_a$ , partitioning  $v = [v_a^T, v_\infty^T]^T$ , where  $v_a$  contains the indices corresponding to the  $m_a$  affine roots.

Multiplication of the rows  $Z(v_a, \cdot)$  by a user-chosen shift function  $\sigma$  determines the row selection matrix  $S_2$ . Let  $W = ZB^{-1}$  and construct  $A = S_2 W$ . Now truncate  $A = A(:, 1 : m_a)$  such that only the first  $m_a$  columns are retained to obtain an  $m_a \times m_a$  eigenvalue matrix  $A$ . The evaluations of the user-chosen shift function  $\sigma$  at each of the  $m_a$  affine solutions of (1) are found as the eigenvalues of  $A$ , as in Corollary 11, where now only the affine roots are retained.

The complete algorithm is summarized as follows:

*Algorithm 1.* (Affine root-finding).

**input:** system of  $n$  equations  $f_i$  having degrees  $d_i$  in  $n$  unknowns  $x_i$

**output:**  $m_a$  affine solutions for a user-chosen shift function  $\sigma$

- (1) Construct  $M = M(d^*)$  with dimensions  $p \times q$  with  $\text{corank}(M) = q - \text{rank}(M)$ .
- (2) Compute a numerical basis for the nullspace of  $M$ , denoted  $Z$ .
- (3) Determine the number of affine roots  $m_a$  by monitoring the indices of the linearly independent rows in  $Z$ .
- (4) Select the first  $m_a$  linearly independent rows in  $Z$  by means of the row-selection matrix  $S_1$ , i.e.,  $B = S_1 Z$ . Let  $W = ZB^{-1}$ .
- (5) Express the shift relation from Proposition 7 for a user-chosen shift  $\sigma$  to obtain  $A = S_2 W$ , and truncate  $A = A(:, 1 : m_a)$ .
- (6) Compute the eigenvalue decomposition of  $A = VD_\sigma V^{-1}$ . The  $m_a$  eigenvalues in  $D_\sigma$  correspond to the evaluations of the function  $\sigma$  at the  $m_a$  affine solutions.
- (7) Reconstructing the canonical nullspace as in Corollary 12 from  $W(:, 1 : m_a)V$  reveals the mutual matching between the solution components  $x_i$ .

*Remark 16.* When dealing with a polynomial optimization problem, the (polynomial) objective criterion itself can be used as shift function  $\sigma$ . The corresponding eigenvalue problem then returns as its eigenvalues the evaluations of the objective criterion in the critical points of the Lagrangian. The minimal or maximal real eigenvalue corresponds in this case to the minimizer or maximizer of the objective criterion and power iterations (Golub and Van Loan, 1996) can be employed to solve for the optimal solution directly.

*Remark 17.* When both multiple solutions and roots at infinity occur, Algorithm 1 still works. Moreover, even when

one is dealing with a positive-dimensional solution set at infinity, while having a zero-dimensional affine solution set, monitoring the indices of the linearly independent rows in  $Z(d)$  correctly reveals the number of affine roots.

## 5. APPLICATION: GLOBAL OPTIMIZATION FOR STRUCTURED TOTAL LEAST SQUARES

In this example a  $3 \times 3$  Hankel structured total least squares (STLS) problem is solved as a system of polynomial equations to find the globally optimal low-rank approximation to a given data matrix. All simulations were done in MATLAB. The numerical results were confirmed by a polynomial homotopy continuation method and Gröbner Basis computations.

*Example 18.* Let  $A$  be a given  $3 \times 3$  data matrix of full rank, having Hankel matrix structure. De Moor (1993, 1994) proposes a non-linear generalization of the SVD to solve the STLS problem, which is called the Riemannian SVD, which essentially comprises a system of multivariate polynomial equations. Let  $v = [v_1 \ v_2 \ v_3]^T$  and  $l = [l_1 \ l_2 \ l_3]^T$ . The Riemannian SVD equations are

$$\begin{cases} Av = T_v T_v^T l, \\ A^T l = T_l T_l^T v, \\ v^T v = 1 \end{cases} \quad (3)$$

where  $T_v$  and  $T_l$  capture the required Hankel structure constraint (distinguishing the Riemannian SVD from the SVD) and are defined as follows

$$T_v = \begin{bmatrix} v_1 & v_2 & v_3 \\ & v_1 & v_2 & v_3 \\ & & v_1 & v_2 & v_3 \end{bmatrix},$$

and  $T_l$  is defined similarly. The best low-rank approximation of  $A$  is reconstructed from the pair of  $(v, l)$  vectors which minimize the objective criterion

$$J(v) = v^T A^T (T_v T_v^T)^{-1} A v,$$

as described in De Moor (1993, 1994).

Consider the  $3 \times 3$  full-rank Hankel matrix

$$A = \begin{bmatrix} 7 & -2 & 5 \\ -2 & 5 & 6 \\ 5 & 6 & -1 \end{bmatrix},$$

which is approximated by a Hankel matrix  $B$  of rank 2.

Replacing the normalization constraint  $v^T v = 1$  in (3) by  $v_1 = 1$  reduces the number of variables by one. The first equation in  $Av = T_v T_v^T l$  now does not carry any information anymore since it stems from a derivation with respect to  $v_1 = 1$ , and hence it can be dropped. The resulting system is composed of five polynomial equations in five unknowns, where all equations are of degree three. We apply the method as described in Algorithm 1. For  $d^* = 11$ , the matrix  $M = M(d^*)$  has size  $6435 \times 4368$  and is extremely sparse with only 60489 nonzero elements. A basis for the nullspace of this  $M$  is computed, and the root-counting technique reveals there are 39 affine solutions. The STLS optimization problem and the (real) values of the critical points are represented graphically in Figure 1. The optimal rank-2 Hankel matrix approximation of  $A$  is ultimately retrieved as  $B$ , where

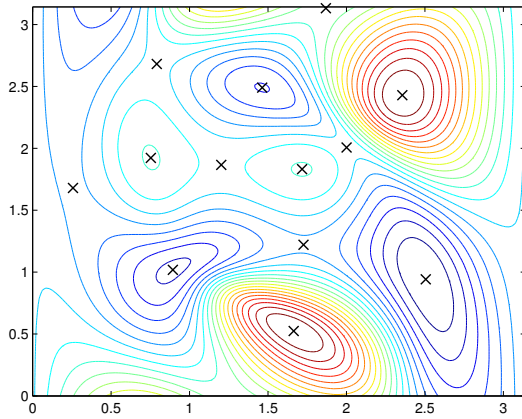


Fig. 1. Level sets of the minimization problem of the  $3 \times 3$  Hankel STLS  $J(v) = v^T A^T (T_v T_v^T)^{-1} A v$  showing several local optima together with the roots retrieved by the presented algorithm. The proposed method is able to identify all 13 critical points ( $\times$ ), and hence guarantees to retrieve the globally optimal solution. The  $x$  and  $y$  axes depict the spherical coordinates of  $v$ , i.e.,  $x = \tan^{-1} v_2/v_1$  and  $y = \cos^{-1} v_3$ .

$$B = \begin{bmatrix} 7.6582 & -0.1908 & 3.2120 \\ -0.1908 & 3.2120 & 1.8342 \\ 3.2120 & 1.8342 & 2.4897 \end{bmatrix},$$

which is indeed of rank 2.

## 6. CONCLUSIONS

Several concepts from algebraic geometry have been translated into their linear algebra counterparts, and a root-finding method based on numerical algebra operations, such as nullspace computations and eigenvalue decompositions was presented. Since the encountered matrix sizes can easily become prohibitively large, ongoing research focuses on devising efficient methods for generating the nullspace of the specifically structured and extremely sparse coefficient matrix. We will look also into how one can solve for the optimizing (real) solution directly in the case of polynomial optimization problems. In this respect, relations with sums-of-squares polynomial optimization (Parrilo, 2000) and the work by Laurent and Rostalski (2012) are of interest.

## ACKNOWLEDGEMENTS

Research supported by Research Council KU Leuven: GOA/11/05 Ambiorics, GOA/10/09 MaNet, CoE EF/05/006 Optimization in Engineering (OPTEC) en PFV/10/002 (OPTEC), IOF-SCORES4CHEM, several PhD / postdoc & fellow grants; Flemish Government: FWO: PhD / postdoc grants, projects: G0226.06 (cooperative systems and optimization), G0321.06 (Tensors), G.0302.07 (SVM/Kernel), G.0320.08 (convex MPC), G.0558.08 (Robust MHE), G.0557.08 (Glycemia2), G.0588.09 (Brain-machine) research communities (WOG: ICCoS, AN-MMM, MLDM); G.0377.09 (Mechatronics MPC); IWT: PhD Grants, Eureka-Flite+, SBO LeCoPro, SBO Climaqs, SBO POM, O&O-Dsquare; Belgian Federal Science Policy Office: IUAP P6/04 (DYSCO, Dynamical systems, control and optimization, 2007-2011); IBBT; EU: ERNSI; FP7-HD-MPC (INFOS-ICT-223854), COST intelliCIS, FP7-EMBOCON (ICT-248940), FP7-SADCO (MC ITN-264735), ERC HIGHWIND (259166); Contract Research: AMINAL; Other: Helmholtz: vICERP; ACCM. PD is a research assistant at the KU Leuven, Belgium and is supported

by the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen). KB is a research assistant at the KU Leuven, Belgium. BDM is a professor at the KU Leuven, Belgium. The scientific responsibility is assumed by its authors.

## REFERENCES

- K. Batselier, P. Dreesen, and B. De Moor. Global optimization and prediction error methods. Technical Report 11-199, KU Leuven ESAT/SCD/SISTA, 2011. Accepted for publication (SYSID2012).
- B. Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*. PhD thesis, University of Innsbruck, 1965.
- D. A. Cox, J. Little, and D. O'Shea. *Using Algebraic Geometry*. Springer-Verlag, New York, second edition, 2005.
- D. A. Cox, J. B. Little, and D. O'Shea. *Ideals, Varieties and Algorithms*. Springer-Verlag, third edition, 2007.
- B. H. Dayton and Z. Zeng. Computing the multiplicity structure in solving polynomial systems. In *Proc. of ISSAC '05*, pages 116–123. ACM Press, 2005.
- B. De Moor. Structured total least squares and  $L_2$  approximation problems. *Lin. Alg. Appl.*, 188/189:163–207, 1993.
- B. De Moor. Total least squares for affinely structured matrices and the noisy realization problem. *IEEE Trans. Signal Process.*, 42(11):3104–3113, 1994.
- I.Z. Emiris and B. Mourrain. Matrices in elimination theory. *J. Symb. Comput.*, 28(1–2):3–44, 1999. ISSN 0747-7171.
- G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, USA, third edition, 1996.
- M. Laurent and Ph. Rostalski. The approach of moments for polynomial equations. In *Handbook on Semidefinite, Conic and Polynomial Optimization*. Springer-Verlag, 2012.
- D. Lazard. Résolution des systèmes d'équations algébriques. *Theor. Comput. Sci.*, 15:77–110, 1981.
- F. S. Macaulay. On some formulae in elimination. *Proc. London Math. Soc.*, 35:3–27, 1902.
- I. Markovskiy. Structured low-rank approximation and its applications. *Automatica*, 44:891–909, 2008.
- H. M. Möller and H. J. Stetter. Multivariate polynomial equations with multiple zeros solved by matrix eigenproblems. *Numer. Math.*, 70:311–329, 1995.
- P. A. Parrilo. *Structured Semidefinite Programs and Semi-algebraic Geometry Methods in Robustness and Optimization*. PhD thesis, California Institute of Technology, May 2000.
- H. J. Stetter. *Numerical Polynomial Algebra*. SIAM, 2004.
- J. J. Sylvester. On a theory of syzygetic relations of two rational integral functions, comprising an application to the theory of Sturm's function and that of the greatest algebraical common measure. *Trans. Roy. Soc. Lond.*, 1853.
- S. Van Huffel and J. Vandewalle. *The Total Least Squares Problem: Computational Aspects and Analysis*, volume 9 of *Frontiers in Applied Mathematics*. SIAM, Philadelphia, 1991.