Back to the Roots – From Polynomial System Solving to Linear Algebra

Philippe Dreesen * **Bart De Moor** *

* *Katholieke Universiteit Leuven,*
Department of Electrical Engineering ESAT-SCD
Kasteelpark Arenberg 10, B-3000 Leuven, Belgium
(e-mail: {philippe.dreesen,bart.demoor}@esat.kuleuven.be)

Abstract: The tasks of solving systems of multivariate polynomial equations and polynomial optimization problems arise as central problems in many systems theory, identification and control settings. Traditionally, methods for solving systems of polynomial equations have been developed in the area of algebraic geometry. A large body of literature is available in this domain, however, most of the computational techniques developed in this research area require exact algebraic computations and suffer from numerical issues. We present a method for solving systems of polynomial equations employing numerical linear algebra tools only. This contribution mainly serves a didactic purpose, and aims at revealing that the task at hand can be tackled by the use of linear algebra operations, such as SVDs and eigenvalue computations, without requiring the knowledge of advanced algebraic geometry results. The problem is translated into the realm of linear algebra by separating coefficients and monomials into a coefficient matrix multiplied with a basis of monomials. The number of affine solutions is found from rank tests on the coefficient matrix. By exploiting the structure present in the monomial bases, all solutions of the system can be found from eigenvalue computations. Relevant applications are found in identification and control, such as solving structured total least squares problems and optimal control.

Keywords: Eigenvalues, Eigenvectors, Global Optimization, Linear Algebra, Linearization, Matrix Formulation, Multivariable Polynomials, Polynomial Methods, 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. The case of polynomial optimization problems can be translated into the same framework by applying the Lagrange multipliers method, which provides the necessary conditions for optimality as a system of polynomial equations.

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. Applying algebraic methods (for instance Gröbner bases as initiated by Buchberger (1965) and currently dominating the field of computer algebra) is often intractable and numerically infeasible due to the symbolic nature of the calculations involved. In this work, we are exploring a different approach: we look at the problem from the linear algebra perspective. Although the link between polynomials and linear algebra is largely unknown, it can be shown that there are many interesting links with linear algebra underlying the task at hand. This approach translates the problem into the world of numerical linear algebra, in which a variety of numerically reliable techniques is available.

1.1 Univariate Case: Well-known Facts

Starting from the univariate case, we briefly review the connections between polynomials and linear algebra, which will serve as the main ingredients for the multivariate case. Connections between polynomial root-finding and linear algebra are well-known for the univariate case. The most basic fact in this respect is that the eigenvalues of an $n \times n$ matrix A correspond to the n roots of its characteristic polynomial $p(\lambda) = \det(A - \lambda I)$. Conversely, with every univariate monic polynomial $f(x) = x^n + a_{n-1}x^{n-1} + \ldots + a_0$, a companion matrix can be associated. The eigenvalues of this $n \times n$ matrix correspond to the roots of $f(x) = 0$: consider

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

The third fact is due to Sylvester. He discovered that checking whether two polynomials $f_1(x)$ and $f_2(x)$ have a common root can be achieved by investigating the determinant of a certain structured square matrix built from the coefficients of the equations. Consider the polynomials $f_1(x)$ and $f_2(x)$,

$$\begin{cases} f_1(x) = a_r x^r + a_{r-1}x^{r-1} + \ldots + a_0, \\ f_2(x) = b_s x^s + b_{s-1}x^{s-1} + \ldots + b_0. \end{cases} \quad (2)$$

By multiplying the input equations f_i with powers of x, the following square system is obtained:

$$\left[\begin{array}{cccccc} a_0 & a_1 & \ldots & a_r & & \\ & a_0 & a_1 & \ldots & a_r & \\ & & \ddots & \ddots & & \ddots \\ & & & a_0 & a_1 & \ldots & a_r \\ \hline b_0 & b_1 & \ldots & b_s & & \\ & b_0 & b_1 & \ldots & b_s & \\ & & \ddots & \ddots & & \ddots \\ & & & b_0 & b_1 & \ldots & b_s \end{array}\right] \left[\begin{array}{c} x^0 \\ x^1 \\ \vdots \\ x^{r+s-1} \end{array}\right] = 0, \quad (3)$$

where the f_1-part of the coefficient matrix has s rows, and the f_2-part has r rows, resulting in a matrix of size $(r+s)\times(r+s)$. The Sylvester matrix has a zero determinant if the equations f_1 and f_2 have a common root: Indeed, the evaluation of the vector containing the powers of x at the common root composes a nonzero vector in the nullspace.

1.2 Multivariate Case

Macaulay was one of the first algebraists who generalized Sylvester's construction to the multivariate case. Many interesting results were obtained in the works of these early algebraists (Sylvester, 1853; Macaulay, 1902). Nevertheless, such methods and results are written in the mathematical language of their time, and their relevance for (numerical) linear algebra was rediscovered only recently by Stetter and coworkers (Auzinger and Stetter, 1988), and later by Emiris and Mourrain (1999) and Stetter (2004).

Our aim is to further explore these approaches and present a method illustrating that many interesting links and results can be obtained from a straightforward linear algebra approach. There is an enormous potential for obtaining insights and developing novel algorithms to bridge the fields of algebraic geometry and (numerical) linear algebra. We intend to present results coming from algebraic geometry in a way such that they become accessible to applied mathematicians and engineers.

2. METHOD

2.1 Basic Method: No Degeneracies

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

$$\begin{cases} f_1 = 0, \\ f_2 = 0, \\ \quad \vdots \\ f_n = 0. \end{cases} \quad (4)$$

having total degrees $d_1, \ldots, d_n$, and define the max total degree $d_\circ = \max(d_1, \ldots, d_n)$. Assume that (4) defines a zero-dimensional solution set, consisting of s single roots, where s is equal to the product of the total degrees, i.e. the Bézout bound,

$$s = \prod_i d_i. \quad (5)$$

By separating the coefficients and the monomials, a system of linear equations can be constructed. This is achieved by constructing a suitably sized Macaulay-like matrix [1], which turns out to contain all the information necessary to find the solutions of the set of equations. Since the max total degree occurring in the equations is $d_\circ$, each equation can be represented as a vector of coefficients multiplied with a vector containing all possible monomials with total degrees from 0 up to $d_\circ$, or

$$m_i^T v_{d_\circ} = 0, \quad i = 1, \ldots, n. \quad (6)$$

The elements of $v_{d_\circ}$ are ordered by some user-chosen monomial ordering scheme. The specific ordering used in $v_{d_\circ}$ is arbitrary as long as all monomials are considered and the corresponding coefficients in m_i are placed accordingly. Note that the equations for which $d_i < d_\circ$ holds, can be 'shifted' internally: by multiplying f_i by all monomials of degrees 1 up to $d_\circ - d_i$, new equations of total degree $\leq d_\circ$ are found, which are adjoined to the input system. After maximally shifting the input equations internally, the set of equations we obtain by this procedure can be represented in matrix-vector notation as $M_{d_\circ} v_{d_\circ} = 0$. Each row of $M_{d_\circ} v_{d_\circ} = 0$ corresponds to an input equation (and shifted versions if necessary). $M_{d_\circ}$ is a matrix containing the coefficients of the input equations (and their shifted versions) and $v_{d_\circ}$ a column vector containing all monomials of degrees less and equal to $d_\circ$.

A first connection to algebraic geometry can be made here; the row space of the coefficient matrix $M_{d_\circ}$ spans the ideal $I = \langle f_1, \ldots, f_n \rangle$ up to the degree $d_\circ$, denoted $I_{\leq d_\circ}$. A link with varieties, another central notion in algebraic geometry is natural as well: a variety is defined as the solution set of a system of equations, and is denoted $V(I)$. From the linear algebra viewpoint, the solutions of the linear system $M_{d_\circ} v_{d_\circ} = 0$ live in the nullspace of $M_{d_\circ}$. Roughly speaking, the dual notions of ideal and variety correspond to the coefficient matrix and its nullspace. In the case I is a radical ideal, the variety can be linked directly to the nullspace of the coefficient matrix; if the ideal is not radical, there are multiplicities and spurious solutions which have to be accounted for (in the next section we will study the degenerate case). Indeed, by evaluating the monomial basis vector $v_{d_\circ}$ at the solutions of the system yields vectors lying in the nullspace of $M_{d_\circ}$. We call the matrix which contains the evaluations of the monomial basis vector $v_{d_\circ}$ at the s solutions the canonical nullspace $V_{d_\circ}$.

Example 1. Consider a simple system of two polynomial equations in two unknowns x_1 and x_2:

$$\begin{cases} f_1(x_1, x_2) = 2x_1^2 - x_2 = 0, \\ f_2(x_1, x_2) = 3x_1 - 4x_2 + 5 = 0, \end{cases} \quad (7)$$

having true solutions $(1, 2)$ and $(-5/8, 25/32)$. Since the equation f_2 is of degree 1, and f_1 is of degree 2, f_2 can be shifted internally. Following the construction of above, we find the initial 4×6 coefficient matrix M_2 and a monomial basis vector v_2 (having a certain user-chosen monomial ordering) as follows:

[1] The matrix we are using is Macaulay's dialytic matrix, of which the subdeterminants were used by Macaulay.

$$M_{d_\circ} v_{d_\circ} = 0 \quad \Leftrightarrow \quad \begin{bmatrix} 0 & 0 & -1 & 2 & 0 & 0 \\ 5 & 3 & -4 & 0 & 0 & 0 \\ 0 & 5 & 0 & 3 & -4 & 0 \\ 0 & 0 & 5 & 0 & 3 & -4 \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_1^2 \\ x_1 x_2 \\ x_2^2 \end{bmatrix} = 0. \quad (8)$$

In general, the initial coefficient matrix $M_{d_\circ}$ 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, it turns out to be necessary to extend the system of equations beyond the internal shifts. By the same procedure of multiplying the input equations with monomials, new equations are added to the set, thereby increasing the maximal total degree of the working set of equations. Again, all possible equations that can be generated in this way are to be included. Inclusion of the newly generated equations in the set of input equations naturally leads to the extension of $I_{d_\circ}$ to I_δ, and hence of the coefficient matrix $M_{d_\circ}$ to M_δ, where $\delta > d_\circ$ is the total degree of all equations after performing the shifts. Note that in this procedure, also the basis of monomials $v_{d_\circ}$ has to be extended accordingly to v_δ: finally the system can be represented as

$$M_\delta v_\delta = 0. \quad (9)$$

This immediately raises the question to which degree δ the matrix M has to be extended in order to be able to reconstruct the solutions correctly. A result from algebraic geometry can be used to obtain a bound on the required degree δ. The coranks[2] of the matrices M_δ for increasing δ correspond to the evaluations of the Hilbert function of I, $HF_I(\delta)$. Since the linear algebra approach links the algebraic variety $V(I)$ to the nullspace of the matrix M_δ, it should come as no surprise that the number of solutions of the input equations corresponds to the corank of M_δ: For a sufficiently large δ, the Hilbert function becomes polynomial, of which the degree defines dimension of the variety $V(I)$. The degree at which the Hilbert function becomes polynomial is called the index of regularity, denoted d_{reg}. An upper bound on the required degree d_{reg} can be obtained through a classical algebraic geometry result, and is given by $d^\star = n(d_\circ - 1) + 1$ (Giusti, 1988). This guarantees that the matrix $M_{d^\star}$ is suitably sized to find all solutions from its nullspace. Since we are assuming that the solution set of the system of polynomials is zero-dimensional, the Hilbert polynomial will be a constant. The corank of M_δ stabilizes at an integer, corresponding to the number of solutions of the input equations. Henceforth, the sufficiently sized matrix $M_{d^\star}$, the corresponding monomial basis vectors $v_{d^\star}$ and the canonical nullspace $V_{d_\star}$ are denoted M, v and V, respectively, for notational convenience.

The final step of the method consists of computing the solutions by solving eigenvalue problems. The monomial basis vector v obeys a very specific shift structure: multiplication of v with a monomial maps low-degree entries of v onto high-degree entries. For example

$$(1, x_1, x_2, x_1^2, x_1x_2, x_2^2)^T x_1 = \quad (10)$$
$$(x_1, x_1^2, x_1x_2, x_1^3, x_1^2x_2, x_1x_2^2)^T. \quad (11)$$

This property can be captured using row selection matrices. If we denote the vector of all monomials in two variables up to degree three by v, we can express the previous shift-property as

$$S_1 v x_1 = S_2 v, \quad (12)$$

where S_1 selects from v the rows 1, 2, 3, 4, 5, 6, which are mapped by the multiplication to rows 2, 4, 5, 7, 8, 9, selected by S_2. In general, this trick can be applied to the canonical nullspace V, leading to

$$S_1 V D = S_2 V, \quad (13)$$

where S_1 and S_2 are row selection matrices selecting rows from V and D is a diagonal matrix containing a user-chosen shift-monomial evaluated at the s solutions. Since V is not available directly, a basis for the nullspace of M is computed as Z. V can be written as $V = ZT$ with T a nonsingular matrix representing the linear transformation. Together with (13), this yields the generalized eigenvalue problem

$$S_1 Z (T D T^{-1}) = S_2 Z, \quad (14)$$

from which the solutions can be found as the eigenvalues of $(S_2 Z, S_1 Z)$.

Note that it is possible to reconstruct the canonical nullspace V from the eigenvalue problem: recall that $V = ZT$, and that the linear transformation T is obtained as the matrix containing the eigenvectors of (14). After reconstructing V, the components of x_i and how they match up can be read off directly from its columns.

Example 2. We continue with the equations from Example 1. The required degree is determined as $d^\star = 3$ and we find the corresponding coefficient matrix M_3, of size 9×10. A basis for the nullspace of M_3 is computed using an SVD, and from this, the generalized eigenvalue problem (14) returns as solutions $(1, 2)$ and $(-0.62500, 0.78125)$.

2.2 Degenerate Cases

In the previous section, it was assumed that (4) has $\prod_i d_i$ solutions. In practical cases, a system of polynomial equations very often has fewer solutions than expected from the Bézout bound (5). This is due to the existence of multiple roots, zero coefficients in the input system or algebraic relations among the coefficients. This can lead to the existence of 'spurious' solutions, which have a mathematical meaning, but are undesired in most practical cases.

Multiple roots pose no real problem for our method: the algorithm from the previous section returns all roots correctly. (We do note, however, that there is a certain loss in numerical accuracy here due to multiple roots.) The only part of the basic algorithm which does not work is the reconstruction of the canonical kernel V. It turns out that in this case, the nullspace of M does not only contain the canonical vectors v evaluated at the solutions, but also (linear combinations of) differential forms where v is differentiated with respect to the unknowns x_i. For a full analysis and characterization, we refer to Möller and Stetter (1995); Dayton and Zeng (2005).

[2] We define the corank to be the dimension of the nullspace of the matrix M_δ.

A classical result in algebraic geometry is that the Bézout bound corresponds to the number of solutions in projective complex space, counted with multiplicity. The 'true' solutions we are interested in, are the so-called affine solutions; the remaining solutions (the spurious ones) are called the solutions at infinity. There are two possible methods for dealing with solutions at infinity. We know from algebraic geometry that it is possible to analyse the task at hand in projective space. Projective geometry incorporates the points at infinity as regular points, which is achieved by homogenizing the input equations. Homogenization introduces a so-called homogenization variable, say x_0, by which terms of a degree lower than the total degree of the whole equation is multiplied (if necessary, a term is multiplied by powers of x_0), such that all terms in an equation are of equal total degree. Determining whether a retrieved solution in projective space corresponds to a root at infinity or an affine root now reduces to checking whether $x_0 = 0$ or $x_0 \neq 0$, respectively. Note also that homogenization introduces an equivalence relation among the points in projective space: any point r is equivalent to any other point q by a scaling factor α, $r = \alpha p$. In order to remove this scale invariance (it can be perceived as a positive dimensional variety, which cannot be tackled by our algorithm), a random linear form is usually added to the homogenized system to obtain an intersection with a linear hyperplane and to 'select' any one of the equivalent points in projective space. In summary, homogenization provides a technique to tackle the problem in a general way: all solutions in projective space are obtained, and selecting the affine solutions boils down to numerically checking whether the homogenization variable is zero or not. However, the dimensionality of the problem also increases: homogenization introduces a new variable and a new equation to the input system.

The second method for dealing with solutions at infinity is to perform a root-counting method to obtain the number of affine solutions. Current algebraic methods which tackle this problem, such as the BKK bound, or the multihomogeneous Bézout bound are described in Cox et al. (2007, 2005). It turns out to be possible to perform the affine root-counting via our matrix method as well! If we homogenize the equations, it is easy to see that the roots at infinity are solely determined by the highest degree terms in the equations. Indeed, for the roots at infinity, the homogenization variable $x_0 = 0$, such that lower degree terms drop out. We now present a method to determine the number of affine roots from inspecting the coranks of the highest degree blocks of M. Consider a suitably large matrix M, for a degree $d = n(d_\circ - 1) + 1$. Denote by $N_{[\delta:d]}$ the matrix built from the columns of M corresponding to the monomials of degrees δ up to d. The sequence $c(\delta) = \text{corank}(N_{[\delta:d]})$ for $\delta = d : -1 : 0$ is a monotone increasing sequence, having a constant level on certain subsequent values for δ, say from $\delta_1^\star$ to $\delta_2^\star$. It turns out that the number of affine roots can be computed as $s_a = \text{corank}(M) - c(\delta^\star)$. (The reason for the fact that one also has to look at lower degree blocks is similar as in the case there are affine multiplicities; in the case we are dealing with roots at infinity, there is an influence on lower-degree blocks.) Even if there is a variety at infinity of higher dimension, this method can be applied and it reveals the correct number of affine roots. In some cases, it is necessary to extend M to a degree $> d$ in order to detect the constant level; however, after the root-counting, M_d can always be used to retrieve the roots themselves. We are currently investigating how this technique can be related to (and proved by) insights from the early work by Auzinger and Stetter (1988).

Once the number of affine solutions is determined, the algorithm stated in the previous section can be used after a minor modification. The corank of M corresponds to the number of solutions in projective space, but we can now separate the affine roots from the roots at infinity. After computing a basis for the nullspace of M as Z, a column compressed version $\bar{Z}$ is computed, where the first s_a columns comprise a matrix of rank s_a. This ensures that the vectors in $\bar{Z}$ span the space in which the affine solutions lie.

The complete algorithm is summarized as follows:

Algorithm 1. (Root-finding Algorithm)

input: a system of n equations f_i having total degrees d_i in n unknowns x_i

output: the s_a affine solutions for a user-chosen shift function σ

(1) Build M_d for $d = n(d_\circ - 1) + 1$
(2) Affine root-count returns s_a
(3) Compute Z as a numerical basis for nullspace of M_d (if $s_a < \prod_i d_i$, perform column compression to obtain $\bar{Z}$
(4) Select from Z (or $\bar{Z}$) s_a linearly independent rows to obtain $S_1 Z$)
(5) Find the s_a rows from Z (or $\bar{Z}$) (or linear combinations thereof) corresponding to the multiplication of B with the user-chosen shift function σ, yielding $S_2 Z$
(6) Compute the s_a solutions for the user-chosen shift function σ as the eigenvalues of $(S_2 Z, S_1 Z)$

2.3 Polynomial Optimization Problems

In engineering and applied mathematics one is often interested in finding the best solution to a posed problem. This corresponds to solving an optimization problem: minimize a certain objective criterion, if necessary subject to constraints on the unknowns. When we are dealing with polynomial optimization, in which both the objective function and the constraints are polynomial equations, the presented method can be employed as well: the Lagrange multipliers method provides the necessary conditions for optimality as a system of polynomial equations. This system can be solved using the presented method in the previous paragraphs, yielding all solutions and hence including the global solution of the optimization problem.

Observe that, instead of choosing a monomial as a shift function to construct the eigenvalue problem, any polynomial function can be chosen (the selection matrices have to be constructed accordingly such that they compose the shifted function). In the case of polynomial optimization, it is natural to choose the objective criterion itself as the shift function, such that the corresponding eigenvalue problem $(S_2 Z, S_1 Z)$ returns as its eigenvalues the evaluations of the objective criterion in the critical points of the Lagrangian. The minimal or maximal real eigenvalue of this problem corresponds to the minimizer or maximizer of the objective criterion and hence power iterations (Golub and Van Loan,

1996) can be employed to solve for the optimal solution directly. Note that the complex eigenvalues have to be dismissed – future work will point out how to solve for the optimizing real eigenvalue directly.

3. EXAMPLES AND RESULTS

We discuss two numerical examples. The first example solves a 3×3 Hankel structured total least squares problem as a system of polynomial equations and finds the globally optimal low-rank approximation to a given data matrix. The second example phrases an optimal control problem with a nonlinear system as a polynomial optimization problem. All simulations were done in MATLAB. The numerical results were confirmed by a polynomial homotopy continuation method and Gröbner Basis computations.

3.1 Structured Total Least Squares

The total least squares problem tackles the task of least squares approximation of a matrix by a matrix of lower rank, in which all measured data are subject to noise. This problem is traditionally solved by computing the SVD (see Van Huffel and Vandewalle (1991); Golub and Van Loan (1996) for a complete treatment). When additional constraints are imposed on the matrices involved, e.g., linear matrix structure such as Hankel or Toeplitz structure, or element-wise weighting, the so-called Riemannian SVD was proposed in De Moor (1993, 1994, 1997) to solve the so-called structured total least squares problem. The structured low-rank approximation problem is a central task in systems theory, identification and control, underlying many modeling problems, such as EIV system identification, approximate realization and model reduction (see Markovsky (2008) for an elaborate survey on the problem and its applications).

Example 3. Let A be a given 3×3 data matrix of full rank, having a linear matrix structure (e.g., Hankel). The structured low-rank approximation problem is written as the so-called Riemannian SVD: Let $v = [\, v_1 \;\; v_2 \;\; v_3 \,]^T$ and $l = [\, l_1 \;\; l_2 \;\; l_3 \,]^T$. The Riemannian SVD equations are

$$\begin{cases} Av &= T_v T_v^T l, \\ A^T l &= T_l T_l^T v, \\ v^T v &= 1 \end{cases} \tag{15}$$

where T_v and T_l capture the required Hankel structure constraint, and are defined as follows (see De Moor (1993, 1994) for details)

$$T_v = \begin{bmatrix} v_1 & v_2 & v_3 & & \\ & v_1 & v_2 & v_3 & \\ & & v_1 & v_2 & v_3 \end{bmatrix}, \tag{16}$$

T_l is defined similarly. Note that replacing the normalization constraint $v^T v = 1$ in (15) by $v_1 = 1$ reduces the number of variables by one. Moreover, one equation is redundant now: the first equation in $Av = T_v T_v^T l$ does not carry any information since it comes from a derivation with respect to $v_1 = 1$. Finally, the best low-rank approximation of A is reconstructed from the pair of (v, l) vectors which minimize the objective $J(v) = v^T A^T D_v^{-1} A v$ as described in De Moor (1993, 1994).

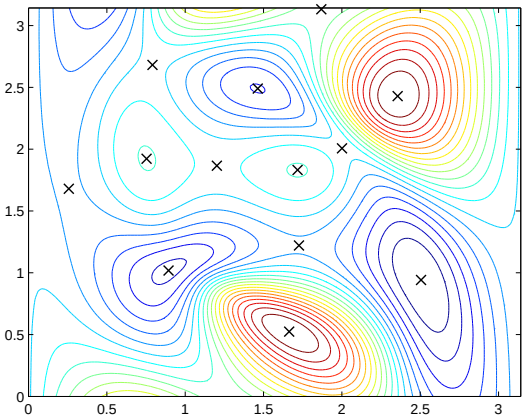

Fig. 1. The plot of the equivalent minimization problem of the 3×3 Hankel STLS $v^T A^T D_v^{-1} A v$ shows several local optima together with the roots retrieved by the presented algorithm. The proposed method is able to identify all critical points ($\times$) of the optimization problem, and hence guarantees to retrieve the globally optimal solution.

Consider the 3×3 full-rank Hankel matrix

$$A = \begin{bmatrix} 7 & -2 & 5 \\ -2 & 5 & 6 \\ 5 & 6 & -1 \end{bmatrix}, \tag{17}$$

which will be approximated by a Hankel matrix B of rank 2. After replacing the normalization constraint by $v_1 = 1$, (15) comprises a system of five polynomial equations (all of total degree three) in five unknowns. We apply the method as described in Section 2. An upper bound for the index of regularity is found as $d^\star = 11$. For this degree, the matrix M has size 6435×4368 and is very sparse with 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 solutions are computed, the complex roots are dismissed, and the 13 real roots are retrieved successfully. Since in this case, there are only two free variables in v, the equivalent minimization problem $J(v) = v^T A^T D_v^{-1} A v$ can be represented graphically. The level sets of $J(v)$ are shown together with the computed values of its critical points in Figure 1. The optimal rank-2 Hankel matrix approximation of A is finally retrieved as

$$B = \begin{bmatrix} 7.6582 & -0.1908 & 3.2120 \\ -0.1908 & 3.2120 & 1.8342 \\ 3.2120 & 1.8342 & 2.4897 \end{bmatrix}. \tag{18}$$

3.2 Nonlinear Optimal Control

The following objective J is a prototypical example of the cost criterion employed in an optimal control setting in discrete time:

$$\min_{x_k, u_k} J = q \sum_{k=0}^{N} x_k^2 + r \sum_{k=0}^{N-1} u_k^2, \tag{19}$$

where N is the time horizon of the problem, q and r are user-chosen weights, and the optimization problem is subject to the dynamical system equations

$$x_{k+1} = f(x_k, u_k), \quad k = 0, \ldots, N-1, \tag{20}$$

where $f(\cdot,\cdot)$ is a polynomial function and x_k and u_k represent the state variables (=output) at time k and input at time k, respectively, and are scalars. The initial state is given by $x_0 = c$.

Example 4. Consider a dynamical nonlinear SISO system which is governed by the nonlinear state equation

$$x_{k+1} = x_k^2 + 5x_k + u_k x_k, \tag{21}$$

with initial state $x_0 = 1$, a time horizon $N = 3$ and weights $q = 1$ and $r = 2$. Since the state variables x_k can be easily eliminated by recursion on the state equations, the problem can be written as an unconstrained optimization problem in the variables u_k, $k = 0, 1, 2$. Setting the derivatives of J with respect to u_k, $k = 0, 1, 2$ to zero yields a system of three polynomial equations in three unknowns of degrees 7, 7 and 6, respectively. Applying our algorithm results in an M matrix of size 1470×1540 having 66255 nonzero entries, from which the optimal input sequence is found as $u = (-2.8076, -0.2435, -0.1553)$.

4. DISCUSSION AND FUTURE WORK

In this paper, we have described a method for solving systems of polynomial equations and polynomial optimization problems. Several concepts from algebraic geometry have been translated into their linear algebra equivalents, and a root-finding method based on numerical algebra operations, such as SVD and eigenvalue decompositions was presented. The method operates implicitly in projective complex space, in which the mathematical description of the solution of the task at hand is convenient due to closedness and inclusion of the concept of infinity. In practical problems, however, one is often only interested in the real solutions in affine space. Further research will focus on how these requirements can be taken into account, and moreover, in the case of polynomial optimization problems, how one can solve for the optimizing solution directly.

There is a cost in linearizing the problem and having the guarantee to find all solutions (or the globally optimal). Even for small-scale examples, the matrix sizes can become prohibitively large. We are currently investigating ways in which the specific structure and sparsity in the coefficient matrix can be exploited in order to devise more efficient algorithms. Also the relation with sums-of-squares polynomial optimization (Parrilo, 2000) will be studied. The objective is to reveal the most direct connection between the given problem and an eigenvalue problem yielding the (optimal) solution.

ACKNOWLEDGEMENTS

PD is supported by the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen). BDM is a full professor at the Katholieke Universiteit Leuven, Belgium. Research supported by Research Council KUL: GOA/11/05 Ambiorics, GOA/10/09 MaNet, CoE EF/05/006 Optimization in Engineering(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, ANMMM, 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); EU: ERNSI; FP7-HD-MPC (INFSO-ICT-223854), COST intelliCIS, FP7-EMBOCON (ICT-248940); Contract Research: AMINAL; Other: Helmholtz: viCERP, ACCM, Bauknecht, Hoerbiger.

REFERENCES

Auzinger, W. and Stetter, H.J. (1988). An elimination algorithm for the computation of all zeros of a system of multivariate polynomial equations. Proc. Int. Conf. Num. Math., 11–30. Birkhäuser.

Buchberger, B. (1965). *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*. Ph.D. thesis, University of Innsbruck.

Cox, D.A., Little, J., and O'Shea, D. (2005). *Using Algebraic Geometry*. Springer-Verlag, New York, second edition.

Cox, D.A., Little, J.B., and O'Shea, D. (2007). *Ideals, Varieties and Algorithms*. Springer-Verlag, third edition.

Dayton, B. and Zeng, Z. (2005). Computing the multiplicity structure in solving polynomial systems. In *Proc. of ISSAC '05*, 116–123. ACM Press.

De Moor, B. (1993). Structured total least squares and L_2 approximation problems. *Lin. Alg. Appl.*, 188/189, 163–207.

De Moor, B. (1994). Total least squares for affinely structured matrices and the noisy realization problem. *IEEE Trans. Signal Process.*, 42(11), 3104–3113.

De Moor, B. (1997). Linear system identification, structured total least squares and the Riemannian SVD. In S. Van Huffel (ed.), *Recent advances in Total Least Squares Techniques and Errors-In-Variables modeling*, 225–238. SIAM, Philadelphia.

Emiris, I. and Mourrain, B. (1999). Matrices in elimination theory. *J. Symb. Comput.*, 28(1–2), 3–44.

Giusti, M. (1988). Combinatorial dimension theory of algebraic varieties. *J. Symb. Comput.*, 6(2–3), 249–265.

Golub, G.H. and Van Loan, C.F. (1996). *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, USA, third edition.

Macaulay, F.S. (1902). On some formulae in elimination. *Proc. London Math. Soc.*, 35, 3–27.

Markovsky, I. (2008). Structured low-rank approximation and its applications. *Automatica*, 44, 891–909.

Möller, H.M. and Stetter, H.J. (1995). Multivariate polynomial equations with multiple zeros solved by matrix eigenproblems. *Numer. Math.*, 70, 311–329.

Parrilo, P.A. (2000). *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*. Ph.D. thesis, California Institute of Technology.

Stetter, H.J. (2004). *Numerical Polynomial Algebra*. SIAM.

Sylvester, J.J. (1853). On a theory of syzygetic relations of two rational integral functions, comprising an application to the theory of sturms function and that of the greatest algebraical common measure. *Trans. Roy. Soc. Lond.*

Van Huffel, S. and Vandewalle, J. (1991). *The Total Least Squares Problem: Computational Aspects and Analysis*, volume 9 of *Frontiers in Applied Mathematics*. SIAM, Philadelphia.