Prediction Error Method Identification is an Eigenvalue Problem

Kim Batselier * **Philippe Dreesen** * **Bart De Moor** *

* *Department of Electrical Engineering (ESAT)-SCD, KU Leuven / IBBT Future Health Department, 3001 Leuven, Belgium,Email: {kim.batselier,bart.demoor}@esat.kuleuven.be*

Abstract: This article explores the link between prediction error methods, nonlinear polynomial systems and generalized eigenvalue problems. It is shown how the global minimum of the sum of squared prediction errors can be found from solving a nonlinear polynomial system. An algorithm is provided that effectively counts the number of affine solutions of the nonlinear polynomial system and determines these solutions by solving a generalized eigenvalue problem. The proposed method is illustrated by means of an example.

Keywords: Prediction error methods; Global optimization; System Identification; Numerical Methods; Eigenvalue problems; Linear systems; Parameter estimation.

1. INTRODUCTION

Prediction-error methods (PEMs) [Ljung, 1999] have been the dominant method in system identification for the past 30 years. Their success is partly explained by their broad applicability to a wide spectrum of model parameterizations. In addition, models are given with excellent asymptotic properties due to its kinship with maximum likelihood and systems that operate in closed loop can be handled without any special techniques. But these methods also have some drawbacks. For instance, an explicit parameterization of the model is always required and the search for the parameters can be very laborious involving search surfaces that may have many local minima. This parameter search is typically carried out using the damped Gauss-Newton method [Dennis and Schnabel, 1987]. Good initial parameter values are then of crucial importance. These are typically obtained by using subspace methods [Van Overschee and De Moor, 1996]. In this article we will present a numerical method for finding the global optimum which does not require any initial parameter value. This article will show that PEMs are in principle nonlinear polynomial optimization problems. These are in effect equivalent to solving a nonlinear multivariate polynomial system. It has been only in the past 50 years that methods have been developed to solve such multivariate polynomial systems [Buchberger, 1965]. But these methods are purely symbolic and have an inherent difficulty to cope with measured data. It was Stetter [1996, 2004] who made the conceptual link between multivariate polynomial systems and generalized eigenvalue problems. And although Stetter still uses symbolic computations, he demonstrated the natural link between linear algebra and solving nonlinear polynomial systems. This article takes this link a step further by introducing a method [Dreesen et al., 2011] that does not involve any symbolic computations. The main contribution of this article is the introduction of an affine root counting method, a sparse QR-based implementation and its application on prediction error methods. The article is structured as follows: in Section 2 it will be shown how PEMs can be formulated as polynomial optimization problems. Section 3 reviews the basic algorithm to find the roots of a nonlinear polynomial system using only linear algebra. Section 4 introduces a new method to count and find only the affine roots and therefore discard all roots at infinity. Section 5 discusses a practical implementation of the algorithm using a sparse matrix representation and in Section 6 the complete method is illustrated by means of an example. Finally, in Section 7 we present some concluding remarks. Only linear time-invariant (LTI) models are considered in this article.

2. PREDICTION ERROR METHODS ARE POLYNOMIAL OPTIMIZATION PROBLEMS

In this section it is shown that the prediction error scheme for finding the parameters of LTI models is equivalent to solving a nonlinear polynomial optimization problem under certain conditions. Let the input and output of the system be denoted by $u(t)$ and $y(t)$ respectively. The collected past data up to time N is denoted by $Z^N = \{u(1), y(1), \ldots, u(N), y(N)\}$. We will assume that the measured data have been sampled at discrete time instants. The general form of a LTI model is

$$A(q)y(t) = \frac{B(q)}{F(q)}\,u(t) + \frac{C(q)}{D(q)}\,e(t) \tag{1}$$

where $A(q), B(q), C(q), D(q), F(q)$ are polynomials in the shift operator q^{-1} and $e(t)$ is a white noise sequence. The number of coefficients of $A(q), B(q), C(q), D(q), F(q)$ are n_a, n_b, n_c, n_d, n_f respectively. The degree of $B(q)$ includes a time delay of n_k samples. The basic idea behind PEMs involves the description of the LTI model as a one-step ahead predictor $\hat{y}(t|t-1,\theta)$ of the output $y(t)$. The parameter vector θ contains all coefficients of the polynomials in (1). The one-step ahead predictor is given by

$$\hat{y}(t|t-1,\theta) \; = \left[I - \frac{A(q)D(q)}{C(q)}\right] y(t) + \frac{B(q)D(q)}{C(q)F(q)} u(t).$$

Prediction errors $e(t,\theta)$ are then defined as

$$\begin{aligned} e(t,\theta) &= y(t) - \hat{y}(t|t-1,\theta) \\ &= \frac{A(q)D(q)}{C(q)}y(t) - \frac{B(q)D(q)}{C(q)F(q)}u(t) \end{aligned} \qquad (2)$$

The maximal lag n of (2) determines how many times this expression can be written given Z^N. From rewriting (2) as

$$C(q)F(q)e(t,\theta) = A(q)D(q)F(q)y(t) - B(q)D(q)u(t) \quad (3)$$

the maximal lag n is found as

$$n = \max(n_a + n_d + n_f, n_k + n_b + n_d - 1, n_f + n_c).$$

Note that the definition of the prediction errors $e(t,\theta)$ implies that they are equal to $e(t)$. Estimates for the model parameters, given Z^N, are then found as solutions of the following optimization problem

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \sum_{t=n+1}^{N} l(e(t,\theta)) \qquad (4)$$

subject to (2) where l refers to a suitable norm. We will assume the quadratic norm $l(e) = \frac{e^2}{2}$ throughout the rest of this article. By using Lagrange multipliers $\lambda_1, \ldots, \lambda_{N-n}$ these constraints can be added to the cost function

$$\sum_{t=1}^{N} \frac{e(t,\theta)^2}{2N} + \sum_{t=n+1}^{N} \lambda_{t-1} \left[e(t,\theta) - \frac{A(q)D(q)}{C(q)}y(t) + \frac{B(q)D(q)}{C(q)F(q)}u(t) \right]. \qquad (5)$$

The cost function (5) is clearly polynomial which shows that for the chosen norm PEMs correspond to a nonlinear polynomial optimization problem. Taking the partial derivatives of (5) with respect to every unknown, the model parameters and Lagrange multipliers, and equating this to zero results in a multivariate polynomial system of $2N - n + n_a + n_b + n_c + n_d + n_f$ equations and unknowns. PEMs are therefore mathematically equivalent to solving a multivariate polynomial system. It is however possible to simplify the problem. Examining the cost function (5) reveals that the prediction errors $e(t,\theta)$ occur quadratically. Hence, each partial derivative with respect to a certain $e(t,\theta)$ will result in an equation which is linear in that $e(t,\theta)$. This means that each of the N prediction errors can be eliminated from the polynomial system resulting in $N - n + n_a + n_b + n_c + n_d + n_f$ equations and unknowns. The cost for eliminating N equations and indeterminates however is the increase of the degrees of the remaining polynomials. Equation (3) reveals that the degree will increase with $n_f + n_c + 1$ where the $+1$ term is due to the Lagrange multiplier. Now that we have established that in the prediction-error framework one needs to solve a multivariate polynomial system we proceed with an algorithm that solves this problem using numerical linear algebra.

3. PREDICTION ERROR METHODS ARE EIGENVALUE PROBLEMS

Since PEMs are equivalent with solving a multivariate polynomial system it can also be shown that they are in essence an eigenvalue problem. Dreesen et al. [2011] discuss the link between multivariate polynomial system solving and eigenvalue problems by means of the Macaulay matrix. We will give a quick overview of the notation and concepts for the case that there are no roots with multiplicities and roots at infinity. First, a monomial ordering needs to be chosen. This will allow us to order the different monomials when representing the multivariate polynomials as coefficient vectors. Note that any ordering which is defined on $\mathbb{N}_0^n$ automatically implies an ordering on all monomials in n indeterminates. For further details about monomial orderings we refer to [Cox et al., 2007]. For the remainder of this article we will use the following ordering.

Definition 3.1. Graded Xel Order. Let a and $b \in \mathbb{N}_0^n$. We say $a >_{grxel} b$ if

$$|a| = \sum_{i=1}^{n} a_i > |b| = \sum_{i=1}^{n} b_i, \text{ or } |a| = |b| \textit{ and } a >_{xel} b$$

where $a >_{xel} b$ if, in the vector difference $a - b \in \mathbb{Z}^n$, the leftmost nonzero entry is negative.

Example 3.1. In $\mathbb{N}_0^3$ we have that $(2,0,0) >_{grxel} (0,0,1)$ because $|(2,0,0)| > |(0,0,1)|$ and this implies that $x_1^2 >_{grxel} x_3$. Likewise $(0,1,1) >_{grxel} (2,0,0)$ because $(0,1,1) >_{xel} (2,0,0)$ which implies that $x_2x_3 >_{grxel} x_1^2$.

The most important feature of this ordering is that it is graded. This means that we can partition the coefficients of our polynomial in blocks of equal degree. Once a monomial ordering $>$ is chosen we can uniquely identify the largest monomial of a polynomial f according to $>$. This monomial is called the leading monomial of f. In addition, it also becomes possible now to represent multivariate polynomials as vectors. Indeed, one simply stores the coefficients of the polynomial into a vector, graded xel ordered, ascending from left to right.

Example 3.2. The vector representation of $2 + 3x_1 - 4x_2 + x_1x_2 - 7x_2^2$ is

$$\begin{array}{cccccc} 1 & x_1 & x_2 & x_1^2 & x_1x_2 & x_2^2 \\ \big(2 & 3 & -4 & 0 & 1 & -7\big). \end{array}$$

We will denote the vector space of all polynomials in n indeterminates and of degree d by C_d^n. The key concept in solving multivariate polynomial systems is the Macaulay matrix which is the matrix that contains the coefficients of the polynomial system. This matrix is defined in terms of the degree we consider.

Definition 3.2. Given a set of polynomials $f_1, \ldots, f_s \in C_d^n$, each of degree d_i $(i = 1, \ldots, s)$ then the Macaulay matrix of degree d, $M(d)$, is the matrix containing the coefficients of

$$M(d) = \begin{pmatrix} f_1 \\ x_1 f_1 \\ \vdots \\ x_n^{d-d_1} f_1 \\ f_2 \\ x_1 f_2 \\ \vdots \\ x_n^{d-d_s} f_s \end{pmatrix} \qquad (6)$$

where each polynomial f_i is multiplied with all monomials from degree 0 up to $d - d_i$ for all $i = 1, \ldots, s$.

Example 3.3. Suppose the following polynomial system

$$\begin{cases} f_1 : \; xy - 2y = 0 \\ f_2 : \quad\; y - 3 = 0 \end{cases}$$

and fix the degree to 2. The Macaulay matrix is then

$$M(2) \;=\; \begin{array}{c} \\ f_1 \\ f_2 \\ x\,f_2 \\ y\,f_2 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & x & y & x^2 & xy & y^2 \end{array} \\ \left(\begin{array}{cccccc} 0 & 0 & -2 & 0 & 1 & 0 \\ -3 & 0 & 1 & 0 & 0 & 0 \\ 0 & -3 & 0 & 0 & 1 & 0 \\ 0 & 0 & -3 & 0 & 0 & 1 \end{array}\right) \end{array}$$

Note that the Macaulay matrix not only contains the original polynomials $f_1, \ldots, f_s$ but also 'shifted' versions where we define a shift as a multiplication with a monomial. The dependence of this matrix on the degree d is of crucial importance, hence the notation $M(d)$. It can be shown [Macaulay, 1902, Giusti, 1988] that the degree

$$d = \sum_{i=1}^{s} d_i - n + 1 \tag{7}$$

provides an upper bound for the degree for which all the solutions of the polynomial system appear in the kernel of the Macaulay matrix. The number of solutions, which is evidently the dimension of the kernel, is then simply given by the Bezout bound

$$m_B \;=\; \prod_{i=1}^{s} d_i.$$

The canonical kernel K is a generalized Vandermonde matrix. It consists of columns of monomials, graded xel ordered and evaluated in the roots of the polynomial system. This monomial structure allows to use a shift property which is reminiscent of realization theory. This shift property tells us that the multiplication of rows of the kernel with a monomial corresponds with a mapping to other rows of the kernel. This can be written as the following matrix equation

$$S_1 K D \;=\; S_2 K \tag{8}$$

where S_1 and S_2 are row selection matrices and D a diagonal matrix which contains the shift monomial on the diagonal. S_1 will select the first m_B linear independent rows of K. The importance of the graded ordering is here shown since we explicitly want the linear independent rows to be of a degree which is as low as possible. This in order to make sure that the mapping to the other rows, via the row selection matrix S_2, will not lead to rows which are out of bounds. The kernel K is unfortunately unknown but a numerical basis Z for the kernel can be computed. This basis Z is then related to K by $K = ZV$ where V is a linear transformation. Writing the shift property (8) in terms of the numerical basis Z results in the following generalized eigenvalue problem

$$BVD \;=\; AV \tag{9}$$

where $B = S_1 Z$ and $A = S_2 Z$. The eigenvalues D are then the shift monomial evaluated in the different roots of the polynomial system. The canonical kernel K is then easily reconstructed from $K = ZV$. After normalizing K such that its first row contains ones, all solutions can be determined.

4. COUNTING AND FINDING AFFINE ROOTS

In practice, multivariate polynomial systems will also have roots at infinity. These solutions are not desired and need to be removed when formulating the generalized eigenvalue problem (9). It is explained in Dreesen et al. [2011] how solutions at infinity arise from a column-rank deficiency of the coefficient block of highest degree. We will now describe an algorithm that allows to count and find only the desired affine solutions. In order to do this we need to first understand the interpretation of the rank of the Macaulay matrix.

Lemma 1. The rank $r(d)$ of the Macaulay matrix $M(d)$ counts the number of different leading monomials that are present in the row space of the matrix.

Proof 4.1. By performing a left-right column permutation of the Macaulay matrix $M(d)$ and then bringing the matrix into reduced row echelon form one easily sees that there are $r(d)$ different leading monomials which are present in the row space. □

We denote the set of these $r(d)$ leading monomials by $A(d)$. It is now possible to define the following subset of $A(d)$.

Definition 4.1. Given a set of leading monomials $A(d)$, then the reduced monomial system $A^\star(d)$ is defined as the smallest subset of $A(d)$ for which each element of $A(d)$ is divisible by an element of $A^\star(d)$:

$$A^\star(d) \;=\; \{a^\star \in A(d) : \forall\, a \in A(d), \exists\, a^\star \text{such that}\, a^\star | a\}$$

where $a^\star | a$ denotes $a^\star$ divides a.

Before we can proof that the reduced monomial system accounts for only the affine solutions the concept of a pure component needs to be introduced.

Definition 4.2. In C_d^n a monomial x_k^d ($k \in \{1, \ldots, n\}, d \in \mathbb{N}_0^n$) is a pure component and we denote the set of these n monomials by X_n^d.

It is clear from the definition of the reduced monomial system that if pure components are present in $A(d)$ that they will also be present in $A^\star(d)$.

Lemma 2. All monomials in n indeterminates of degree $d \geq d_{\max} = n\ (d_0 - 1) + 1$ can be written as a product of an element of $X_n^{d_0}$ with another monomial.

Proof 4.2. The proof can be completely done in $\mathbb{N}_0^n$ since there is a bijection between the exponents of monomials and $\mathbb{N}_0^n$. We first show that for any degree $d < d_{\max}$, monomials can be found which cannot be written as a product of a pure component and another monomial. For degree $d_{\max} - 1 = n\ (d_0 - 1)$ we can write the following monomial

$$(d_0 - 1, d_0 - 1, \ldots, d_0 - 1) \tag{10}$$

which clearly cannot be written as a product of a pure component and another monomial. It's possible to come up with similar examples for all degrees between d_0 and $d_{\max} - 1$ by just subtracting the necessary amount of a component of (10). For degree $d_{\max} = n\ (d_0 - 1) + 1$ we can write the following monomial

$$(d_0, d_0 - 1, \ldots, d_0 - 1) \tag{11}$$

which is clearly the product of $x_1^{d_0}$ and $x_2^{d_0-1} \ldots\ x_n^{d_0-1}$. Any other monomial of degree $d_{\max}$ can now be formed by rearranging (11) (subtracting from one component and adding to another). If, however, one component is subtracted with a certain amount then the other components should be increased such that the sum of all components remains constant. From this it's easy to see that there will always be at least 1 component $\geq d_0$. □

Now all necessary ingredients are present to proof the condition for the zero-dimensionality of the affine solution set.

Theorem 3. A multivariate polynomial system $f_1, \dots, f_s$ has a zero-dimensional affine solution set if from a certain degree $A(d)$ contains for each indeterminate a pure component. The number of affine solutions are then counted by the right corank of the Macaulay matrix of the corresponding reduced monomial system $A^\star(d)$.

Proof 4.3. Suppose that the condition is satisfied for a certain degree d. Let $d_1, \dots, d_n$ be the degrees of the pure components for each indeterminate $x_1, \dots, x_n$ respectively and let $d_0 = \max(d_1, \dots, d_n)$. Now consider the reduced monomial system $A^\star(d)$. Theorem 2 tells us that from a degree $d_{\max} = n(d_0 - 1) + 1$ onwards all columns of highest degree of the Macaulay matrix of $A^\star(d)$ will be filled and hence its right corank will not increase anymore. This proves the zero-dimensionality of the solution set. In addition, the coefficient block of highest degree being of full column rank also implies that the monomials of $A^\star(d)$ do not account for the solutions at infinity. $\square$

Theorem (3) provides the basis for counting the affine solutions. Algorithm 1 summarizes the different steps in pseudocode. The algorithm iterates over the degree d, starting from the maximal degree in the polynomial system. For every degree the Macaulay matrix $M(d)$ is constructed. From this matrix a basis Z for its kernel, the monomial set $A(d)$ and the reduced monomial system $A^\star(d)$ are determined. Once a reduced monomial system $A^\star(d)$ with all pure components is found, the number of affine solutions are then determined by counting the zero columns of the Macaulay matrix constructed from $A^\star(d)$. Furthermore, the indices of those zero columns provide the indices of the rows of Z that form the B matrix of the generalized eigenvalue problem. These indices hence determine the row selection matrix S_1 of (9). After choosing a monomial shift, the matrix $A = S_2\, Z$ can be constructed and the generalized eigenvalue solved. After the canonical kernel K is reconstructed and normalized such that its first row contains ones, all affine solutions of (4) are effectively found.

```
Algorithm 1. Affine Root-finding
Input: polynomials $f_1, \dots, f_s \in C_d^n$
Output: canonical kernel $K$ of affine roots
  • $d = \max(d_1, \dots, d_s)$
  • $A^\star(d) = \emptyset$
  • WHILE all pure components $\notin A^\star(d)$
      · construct $M(d)$
      · compute basis $Z$
      · compute $A(d)$ from $M(d)$
      · compute $A^\star(d)$ from $A(d)$
      · IF all pure components $\in A^\star(d)$
            construct $A = S_2\, Z$ and $B = S_1\, Z$
            $[V, D] = \mathrm{eig}(A, B)$
            $K = Z\, V$
            normalize $K$
      · ELSE
            $d = d + 1$
      · ENDIF
  • ENDWHILE
```

The following section explains how Algorithm 1 is implemented using two sparse rank-revealing QR factorizations.

5. IMPLEMENTATION OF THE METHOD

The number of rows $p(d)$ of the Macaulay matrix $M(d)$ is given by

$$p(d) \;=\; \sum_{i=1}^{s} \binom{d - \deg(f_i) + n}{n}$$

and the number of columns $q(d)$ by

$$q(d) \;=\; \binom{d + n}{n}.$$

Due to the this combinatorial explosion it quickly becomes difficult to store the full matrix in memory. Fortunately $M(d)$ is extremely sparse and structured. This sparsity can be exploited by using a sparse matrix representation which stores only the nonzero elements in memory. The rank of $M(d)$ and a basis for its kernel need to be computed and a sparse setting is not suited to do a full singular value decomposition. A sparse rank-revealing QR decomposition is available however and the implementation used in this article is by Davis [20xx]. Householder reflections are applied to the sparse $M(d)^T$ matrix to obtain the factorization $M(d)^T = QR$ where R is upper triangular and Q orthogonal. Before the factorization, a column permutation P is applied to $M(d)^T$ to reduce fill-in of R. The rank $r(d)$ of $M(d)$ is estimated from the number of nonzero diagonal elements of R and a basis Z for the kernel is then given by the columns of Q corresponding with zero diagonal elements of R. It is only by computing the QR decomposition of the transpose of $M(d)$ that a basis for the kernel can be computed. The leading monomials $A(d)$ are found from the sparse QR decomposition of $Z^T P_2$ where P_2 is another column permutation. Notice that the monomials of $A(d)$ are the linear independent columns of $M(d)$ and therefore linear dependent rows of Z. Let m denote the number of columns of Z. Since Z is of full column rank, the linear independent rows of Z are given by the first m columns of $Z^T P_2$. These linear independent rows of Z also form the B matrix of the generalized eigenvalue problem. The leading monomials of $A(d)$ are the remaining columns of $Z^T P_2$. The reduced monomial system $A^\star(d)$ is easily obtained from the exponents of the monomials in $A(d)$. Checking whether all pure components are present in $A^\star(d)$ is also done by inspecting its exponents. Once all pure components are found, the matrices A and B are easily obtained from Z. Since the dimension of the eigenvalue problem is typically much less than the dimension of $M(d)$, both matrices can be stored in a full representation. Computing the first QR decomposition is the limiting step in the algorithm. The size of the full Q, which depends on the number of polynomials and unknowns and the degree of each polynomial, determines the applicability of the method. From this it can be seen that the number of data points N will be the restrictive factor in solving these problems. Eliminating the Lagrange multipliers would remove this dependence of the size of the polynomial system on N at the cost of higher degrees. How this elimination can be achieved is not yet clear and the subject of further research.

6. EXAMPLE: MODEL PARAMETER ESTIMATION OF AN OUTPUT ERROR MODEL

This section illustrates the application of the described method on a small example. The model parameters of the following Output-Error model

$$y(t) \; = \; \frac{0.2q^{-1}}{(1-1.6q^{-1}+.89q^{-2})}u(t)+e(t) \qquad (12)$$

are estimated. The system is excited with a white-noise input $u(t)$ of 6 samples. These samples are obtained from a zero-mean Gaussian distribution with a standard deviation of 10. The system output $y(t)$ is then calculated from (12) using zero-mean Gaussian white noise $e(t)$ with a standard deviation of 0.1. After introducing the Lagrange multipliers $\lambda_1, \ldots, \lambda_4$ and eliminating the prediction errors $e(1)\ldots, e(6)$, the following nonlinear polynomial system of 7 polynomials in 7 indeterminates is obtained

$$\left\{\begin{array}{l}
y(3)+f_1y(2)+f_2y(1)-b_1u(2)-6\lambda_1-6\lambda_2f_1-6\lambda_3f_2 \\
\quad -6\lambda_1f_1^2-6f_1\lambda_2f_2-6f_2^2\lambda_1=0 \\
\\
y(4)+f_1y(3)+f_2y(2)-b_1u(3)-6\lambda_2-6\lambda_3f_1-6\lambda_4f_2 \\
\quad -6\lambda_1f_1-6\lambda_2f_1^2-6f_1\lambda_3f_2-6f_2\lambda_1f_1-6\lambda_2f_2^2=0 \\
\\
y(5)+f_1y(4)+f_2y(3)-b_1u(4)-6\lambda_3-6\lambda_4f_1-6\lambda_2f_1 \\
\quad -6\lambda_3f_1^2-6f_1\lambda_4f_2-6\lambda_1f_2-6f_1\lambda_2f_2-6\lambda_3f_2^2=0 \\
\\
y(6)+f_1y(5)+f_2y(4)-b_1u(5)-6\lambda_4-6\lambda_3f_1-6\lambda_4f_1^2 \\
\quad -6\lambda_2f_2-6f_1\lambda_3f_2-6\lambda_4f_2^2=0 \\
\\
\lambda_1y(2)-6\lambda_1^2f_1-6\lambda_1\lambda_2f_2+\lambda_2y(3)-6\lambda_2\lambda_1-6\lambda_2^2f_1 \\
\quad -6\lambda_2\lambda_3f_2+\lambda_3y(4)-6\lambda_3\lambda_2-6\lambda_3^2f_1-6\lambda_3\lambda_4f_2 \\
\quad +\lambda_4y(5)-6\lambda_4\lambda_3-6\lambda_4^2f_1=0 \\
\\
\lambda_1y(1)-6\lambda_1^2f_2+\lambda_2y(2)-6\lambda_2\lambda_1f_1-6\lambda_2^2f_2+\lambda_3y(3) \\
\quad -6\lambda_3\lambda_1-6\lambda_3\lambda_2f_1-6\lambda_3^2f_2+\lambda_4y(4)-6\lambda_4\lambda_2 \\
\quad -6\lambda_4\lambda_3f_1-6\lambda_4^2f_2=0 \\
\\
-\lambda_1u(2)-\lambda_2u(3)-\lambda_3u(4)-\lambda_4u(5)=0.
\end{array}\right. \qquad (13)$$

Note that in this example

$$\begin{aligned}
x_1 &= b_1 \\
x_2 &= f_1 \\
x_3 &= f_2 \\
x_4 &= \lambda_1 \\
x_5 &= \lambda_2 \\
x_6 &= \lambda_3 \\
x_7 &= \lambda_4.
\end{aligned}$$

The reduced monomial system $A^\star(d)$ contains all pure components from the degree $d = 9$. This is well below the upper bound (7) which evaluates to $6(3-1)+1 = 13$. Indeed, (7) is in practice a very pessimistic upper bound. The Macaulay matrix $M(9)$ is a 16731 by 11440 matrix. The pure components are given by $\{b_1^4, f_1^3, f_2^4, \lambda_1^3, \lambda_2^3, \lambda_3^3, \lambda_4\}$. The affine solution set consists of 43 solutions of a total of 801. Only 7 of the 43 are real. The solution that minimizes the cost function

$$V = \frac{1}{12}\sum_{t=1}^{6} e(t)^2$$

is given by

$$\begin{aligned}
b_1 &= 0.2174 \\
f_1 &= -1.5738 \\
f_2 &= 0.8506 \\
\lambda_1 &= 0.0004 \\
\lambda_2 &= -0.0009 \\
\lambda_3 &= -0.0099 \\
\lambda_4 &= 0.0180.
\end{aligned}$$

with $V = 0.00822$. The model corresponding with the global miminum is hence given by

$$y(t) \; = \; \frac{0.2174q^{-1}}{(1-1.5738q^{-1}+0.8506q^{-2})}u(t)+e(t). \qquad (14)$$

The 6 remaining affine solutions correspond with non-stable solutions and therefore the Matlab System Identification toolbox [Ljung, 2008] returns exactly the same result. This confirms that the proposed method solves the optimization problem (4) as described in [Ljung, 1999].

7. CONCLUSION

A new method was proposed that guarantees to find the global optimimum of nonlinear polynomial optimization problems and was applied to the prediction-error framework. The method is able to correctly find all affine solutions and remove all solutions at infinity by employing the notion of the reduced monomial system. Finding these roots however is still a daunting task considering the combinatorial explosion of the dimensions of the Macaulay matrix. The dependence of the size of the polynomial system on the number of data points N is a restricting factor in the applicability of the method. Further elimination of the Lagrange multipliers would be desirable but it is not yet clear how this can be achieved. Exploiting the sparsity of the Macaulay matrix already enables us to solve problems which are already impossible to solve in a full matrix representation. Surely, the next step is to exploit the quasi-toeplitz structure of the Macaulay matrix and to eliminate the Lagrange multipliers.

ACKNOWLEDGEMENTS

Kim Batselier is a research assistant at the Katholieke Universiteit Leuven, Belgium. Philippe Dreesen is supported by the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen). Bart De Moor 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) 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, 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); IBBT EU: ERNSI; FP7-HD-MPC (INFSO-ICT-223854), COST intelliCIS, FP7-EMBOCON (ICT-248940), FP7-SADCO (MC ITN-264735), ERC HIGHWIND (259 166) Contract Research: AMINAL Other: Helmholtz: viCERP ACCM. The scientific responsibility is assumed by its authors.

REFERENCES

B. Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal.* PhD thesis, Mathematical Institute, University of Innsbruck, Austria, 1965.

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

T.A. Davis. Algorithm 9xx, SuiteSparseQR: multifrontal multithreaded rank-revealing sparse QR factorization. *submitted to ACM Transactions on Mathematical Software*, V:1–24, 20xx.

J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations (Classics in Applied Mathematics).* Society for Industrial Mathematics, January 1987. ISBN 0898713641.

P Dreesen, K Batselier, and B De Moor. Back to the roots: Polynomial System Solving,Linear Algebra, Systems Theory. Technical report, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2011.

M Giusti. Combinatorial Dimension Theory of Algebraic-Varieties. *Journal of Symbolic Computation*, 6(2-3):249–265, Oct-Dec 1988. ISSN 0747-7171.

L. Ljung. *System Identification: Theory for the User (2nd Edition).* Prentice Hall, 2 edition, January 1999. ISBN 0136566952.

L. Ljung. *System Identification Toolbox for use with MATLAB, Version 7.2.1.* The Mathworks, Inc, Natick, MA, 7th ed edition, 2008.

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

H.J. Stetter. Matrix eigenproblems are at the heart of polynomial system solving. *SIGSAM Bulletin*, 30(4): 22–5, December 1996. ISSN 0163-5824.

H.J. Stetter. *Numerical Polynomial Algebra.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2004. ISBN 0898715571.

P. Van Overschee and B. De Moor. *Subspace Identification for Linear Systems: Theory, Implementation, Applications.* Kluwer Academic Publishers, 1996.