Automatica 50 (2014) 3030–3037

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

# A matrix nullspace approach for solving equality-constrained multivariable polynomial least-squares problems ☆

Matthew S. Hölzel [a], Dennis S. Bernstein [b]

[a] University of Bremen and the German Aerospace Center (DLR), 28359 Bremen, Germany
[b] University of Michigan, Ann Arbor, MI 48109, USA

ARTICLE INFO

*Article history:*
Received 12 July 2013
Received in revised form
18 July 2014
Accepted 22 July 2014
Available online 28 October 2014

*Keywords:*
Identification algorithms
Least squares
Multivariable polynomial

ABSTRACT

We present an elimination theory-based method for solving equality-constrained multivariable polynomial least-squares problems in system identification. While most algorithms in elimination theory rely upon Groebner bases and symbolic multivariable polynomial division algorithms, we present an algorithm which is based on computing the nullspace of a large sparse matrix and the zeros of a scalar, univariate polynomial.



## 1. Introduction

As system identification stretches the boundaries of optimal estimation toward ever more complicated scenarios, that is, with nonlinearities present and under non-Gaussian noise assumptions, the optimization problems that need to be solved also begin to push the available solvers to their limits. For instance, it is easy to construct scenarios for which the log-likelihood function of a maximum likelihood method or the objective function of a prediction-error method are highly nonconvex (Box, Jenkins, & Reinsel, 2008; Brockwell & Davis, 2006; Ljung, 1999; Pintelon & Schoukens, 2001; Speyer & Chung, 2008; Wang & Garnier, 2012). In our view, this is a growing problem, since properties such as the estimate's variance might only be valid for the global minimizer (or maximizer) of the optimization problem (Box et al., 2008; Ljung, 1999; Pintelon & Schoukens, 2001), although many optimization methods will only guarantee that we find a local minimizer (Nocedal & Wright, 2006). A common shortcut is to solve a regularized or relaxed version of the true optimization problem (Ho & Kalman, 1966), although this approach may inadvertently introduce additional minimizers.

In this paper, we present a global method for solving a class of optimization problems that arise in system identification, specifically, equality-constrained multivariable polynomial least-squares problems. Although this problem has been addressed by the algebraic geometry community via elimination theory, all of the available literature appears to revolve around Groebner bases and symbolic multivariable polynomial division algorithms (Buchberger, 1985; Cox, Little, & O'Shea, 2007). Here we show how to solve the same problem using linear algebra techniques. This line of research is conceptually similar to the idea of solving univariate polynomial problems using linear algebra techniques (Gohberg, Lancaster, & Rodman, 2009; Holzel & Bernstein, 2011, 2012), although we deal with multivariable polynomials, which require a new set of machinery.

The method we introduce is based on computing the nullspace of a large sparse matrix, and computing the zeros of a scalar, univariate polynomial. We introduce a novel nullspace algorithm to accomplish this goal, although any nullspace method (QR, SVD, etc.) could easily be substituted in the main algorithm. In our view, the main contribution of this paper is the formulation of these multivariable optimization problems in a way for which standard tools such as nullspace computation methods and eigenvalue solvers can be directly applied. In this way, advances in sparse nullspace techniques can be easily and directly applied to this large class of optimization problems. The method we present does not rely on an initial guess, and will yield the set of local and global minimizers to equality-constrained multivariable polynomial optimization problems when there exist a finite number of local and

☆ This work was supported in part by a NASA ASP Fellowship. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Alessandro Chiuso under the direction of Editor Torsten Söderström.

*E-mail addresses:* hoelzel@uni-bremen.de (M.S. Hölzel), dsbaero@umich.edu (D.S. Bernstein).

http://dx.doi.org/10.1016/j.automatica.2014.10.039
0005-1098/

global minimizers. We demonstrate the algorithm on a nonlinear ARX model identification problem.

## 2. Problem statement

Here we introduce the class of problems that our method is capable of solving. In the next section we will present some common optimization problems which fit into this framework.

First, we introduce some definitions:

- A *monomial* $e$ in $x_1, \ldots, x_n$ is a product of the form
$$e = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n} \tag{1}$$
where $\alpha_1, \ldots, \alpha_n$ are nonnegative integers.
- The *total degree* of the monomial $e$ is the sum $\alpha_1 + \cdots + \alpha_n$. Specifically, we write
$$\deg(e) = \alpha_1 + \cdots + \alpha_n. \tag{2}$$
- A *polynomial* $f$ in $x_1, \ldots, x_n$ is a finite linear combination of monomials in $x_1, \ldots, x_n$, that is,
$$f = \sum_{i=1}^{k} a_i e_i \tag{3}$$
where $k$ is a finite positive integer, $a_1, \ldots, a_k$ are scalars, and $e_1, \ldots, e_k$ are monomials in $x_1, \ldots, x_n$.
- If $f$ is a polynomial in a single variable, for instance, if $f$ is a polynomial in $x_1$, then $f$ is called a *univariate polynomial.*
- The set of polynomials in $x_1, \ldots, x_n$ with coefficients $a_1, \ldots, a_k \in \mathbb{R}$ is denoted by $\mathbb{R}[x_1, \ldots, x_n]$.
- If $a_1, \ldots, a_k$ are all nonzero and $e_1, \ldots, e_k$ are unique, then the *total degree* of $f$ is $\max(\deg(e_1), \ldots, \deg(e_k))$. Specifically, we write
$$\deg(f) = \max\left(\deg(e_1), \ldots, \deg(e_k)\right). \tag{4}$$

Now, using these definitions we can precisely formulate the problem statement:

**Problem 1.** Given $g_1, \ldots, g_\ell, h_1, \ldots, h_m \in \mathbb{R}[x_1, \ldots, x_n]$, where $g_1, \ldots, g_\ell$ have total degrees less than or equal to $s$, and $h_1, \ldots, h_m$ have total degrees less than or equal to $t$,
$$\begin{aligned} &\underset{x=(x_1,\ldots,x_n)}{\text{minimize}} && g_1^2(x) + \cdots + g_\ell^2(x) \\ &\text{s.t.} && h_1(x) = \cdots = h_m(x) = 0. \end{aligned} \tag{5}$$
□

The first step we will take toward solving this problem is to make it look more like a linear matrix problem. To accomplish this, we introduce some notation:

**Notation.** Let $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ and let $s$ denote a positive integer. Then
$$x^{\otimes s} \triangleq \underset{1}{x} \otimes \underset{2}{x} \otimes \underset{3}{x} \otimes \underset{\cdots}{\cdots} \otimes \underset{s}{x}$$
where $\otimes$ represents the Kronecker product, that is, $x^{\otimes s}$ is the result of repetitively applying the Kronecker product $s - 1$ times to the vector $x$. □

**Example 1.** Let $x = \begin{bmatrix} x_1, & x_2 \end{bmatrix}^T$ and $s = 2$. Then
$$\begin{bmatrix} 1 \\ x \end{bmatrix}^{\otimes 2} = \begin{bmatrix} 1, & x_1, & x_2, & x_1, & x_1^2, & x_1x_2, & x_2, & x_1x_2, & x_2^2 \end{bmatrix}^T$$
where we can see that the vector $\begin{bmatrix} 1, & x^T \end{bmatrix}^{\otimes 2}$ contains every monomial in $x_1, x_2$ of total degree less than or equal to 2. □

The observation of Example 1 is generalized with the following fact:

**Fact 1.** *For every $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ and every positive integer $s$, the vector $\begin{bmatrix} 1, & x^T \end{bmatrix}^{\otimes s}$ contains every monomial in $x_1, \ldots, x_n$ of total degree less than or equal to $s$. Hence for every polynomial $g_i \in \mathbb{R}[x_1, \ldots, x_n]$ with total degree less than or equal to $s$, there exists $G_i \in \mathbb{R}^{1\times(n+1)^s}$ such that*
$$g_i = G_i \begin{bmatrix} 1 \\ x \end{bmatrix}^{\otimes s}. \tag{6}$$
□

Finally, using Fact 1, we can reformulate Problem 1 into an equivalent, more matrix-like form:

**Problem 2.** Given $G \in \mathbb{R}^{\ell\times(n+1)^s}$ and $H \in \mathbb{R}^{m\times(n+1)^t}$,
$$\underset{x=(x_1,\ldots,x_n)}{\text{minimize}} \left\| G \begin{bmatrix} 1 \\ x \end{bmatrix}^{\otimes s} \right\|_2^2, \quad \text{s.t.} \quad H \begin{bmatrix} 1 \\ x \end{bmatrix}^{\otimes t} = 0_{m\times 1}. \tag{7}$$
□

**Remark.** Consider Problem 2, where $G_1, \ldots, G_\ell \in \mathbb{R}^{1\times(n+1)^s}$ denote the rows of $G$, and $H_1, \ldots, H_m \in \mathbb{R}^{1\times(n+1)^t}$ denote the rows of $H$, that is,
$$G \triangleq \begin{bmatrix} G_1 \\ \vdots \\ G_\ell \end{bmatrix} \in \mathbb{R}^{\ell\times(n+1)^s}, \qquad H \triangleq \begin{bmatrix} H_1 \\ \vdots \\ H_m \end{bmatrix} \in \mathbb{R}^{m\times(n+1)^t}. \tag{8}$$
Then Problem 2 is equivalent to Problem 1, where for every $i \in [1, \ell]$ and $j \in [1, m]$, the polynomials $g_i$ and $h_j$ are given by
$$g_i = G_i \begin{bmatrix} 1 \\ x \end{bmatrix}^{\otimes s}, \qquad h_j = H_j \begin{bmatrix} 1 \\ x \end{bmatrix}^{\otimes t}. \tag{9}$$
□

### 2.1. Special cases

To help the reader get a better grasp of the types of problems covered by Problems 1 and 2, we show two common problems which can be cast in this framework.

#### 2.1.1. Equality-constrained linear least-squares

Consider the equality-constrained linear least-squares problem:
$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \left\| \tilde{G}x - \tilde{b} \right\|_2^2 \quad \text{s.t.}\ \tilde{H}x = \tilde{d}. \tag{10}$$
Then letting
$$G \triangleq \begin{bmatrix} -\tilde{b}, & \tilde{G} \end{bmatrix}, \qquad H \triangleq \begin{bmatrix} -\tilde{d}, & \tilde{H} \end{bmatrix} \tag{11}$$
we have that (10) is equivalent to Problem 2, where $s = t = 1$, and $G$ and $H$ are given by (11).

#### 2.1.2. Equality-constrained bilinear least-squares

Consider the equality-constrained bilinear least-squares problem:
$$\begin{aligned} &\underset{v\in\mathbb{R}^p, w\in\mathbb{R}^q}{\text{minimize}} && \left\| G_v v + G_w w + G_{vw}\left(v \otimes w\right) - b \right\|_2^2 \\ &\text{s.t.} && H_v v + H_w w + H_{vw}\left(v \otimes w\right) = d. \end{aligned} \tag{12}$$

Then letting

$$x \triangleq \begin{bmatrix} v^T, & w^T \end{bmatrix}^T \tag{13}$$

$$P \triangleq \begin{bmatrix} I_p, & 0_{p\times q} \end{bmatrix} \otimes \begin{bmatrix} 0_{q\times(p+1)}, & I_q \end{bmatrix} \tag{14}$$

$$G \triangleq \begin{bmatrix} -b, & G_v, & G_w, & G_{vw}P \end{bmatrix} \tag{15}$$

$$H \triangleq \begin{bmatrix} -d, & H_v, & H_w, & H_{vw}P \end{bmatrix} \tag{16}$$

we have that (12) is equivalent to Problem 2, where $s = t = 2$, $n = p + q$, and $G$ and $H$ are given by (15) and (16), respectively.

### 2.2. Problem 2 is fundamentally nonlinear

The purpose of transforming Problem 1 into Problem 2 was to obtain a problem that looked more like a standard linear matrix problem. Unfortunately, we may have done too good of a job. Specifically, examining Problem 2, it may be tempting to think that when $s = t$ or $H = 0$, we can simply replace the vector $\begin{bmatrix} 1 \\ x \end{bmatrix}^{\otimes s}$ with a vector $\theta$, and to instead solve the problem

$$\underset{\theta}{\text{minimize}} \ \|G\theta\|_2^2, \quad \text{s.t.} \quad H\theta = 0_{m\times 1}. \tag{17}$$

However, although finding all of the minimizers of (17) may be computationally easy, in general the solutions $\theta$ of (17) will not be exactly decomposable into the form $\begin{bmatrix} 1 \\ x \end{bmatrix}^{\otimes s}$. The two exceptions are when Problem 2 is linear, and when the minimizer of Problem 2 has a zero cost function, that is,

$$\left\| G \begin{bmatrix} 1 \\ x \end{bmatrix}^{\otimes s} \right\|_2^2 = 0. \tag{18}$$

The fact that solving (17) is generally not an alternative to solving Problem 2 is demonstrated with the following example:

**Example 2.** Let $s = t = 2$ and let $x$ be a scalar, that is, $n = 1$. Then

$$\begin{bmatrix} 1 \\ x \end{bmatrix}^{\otimes s} = \begin{bmatrix} 1, & x, & x, & x^2 \end{bmatrix}^T. \tag{19}$$

Furthermore, let

$$b \triangleq \begin{bmatrix} 2, & 3, & 4 \end{bmatrix} \tag{20}$$

$$G \triangleq \begin{bmatrix} -b^T, & I_3 \end{bmatrix} \tag{21}$$

$$H \triangleq \begin{bmatrix} 9, & -1, & -1, & -1 \end{bmatrix}. \tag{22}$$

Then solving (17) for $\theta$, we find that

$$\theta = \beta \begin{bmatrix} 1, & 2, & 3, & 4 \end{bmatrix}^T \tag{23}$$

where $\beta$ is an arbitrary scalar in $\mathbb{R}$.

Next, note that the constraint equation of Problem 2 reads:

$$9 - 2x - x^2 = 0. \tag{24}$$

Hence $x = -1 \pm \sqrt{10}$, and therefore, from (19), $\theta$ must be of the form

$$\theta = \begin{bmatrix} 1, & -1 \pm \sqrt{10}, & -1 \pm \sqrt{10}, & 11 \mp 2\sqrt{10} \end{bmatrix}^T. \tag{25}$$

However, since there is no $\beta$ for which (23) is equivalent to (25), it follows that Problem 2 has a different minimizer than (17). Specifically, Problem 2 is fundamentally nonlinear, and cannot be replaced by the optimization problem (17). □

## 3. Necessary conditions of optimality

Here we develop the Lagrangian necessary conditions of optimality for Problem 2. Much like in the linear case, we will solve Problem 2 by finding the set of solutions of the necessary conditions of optimality. However, first we introduce some more matrix notation:

**Notation.** Let $p$ and $q$ be positive integers, and let $u = (u_1, \ldots, u_{pq}) \in \mathbb{R}^{pq}$. Then

$$\operatorname{unvec}(u, p, q) \triangleq \begin{bmatrix} u_1 & u_{p+1} & \cdots & u_{(q-1)p+1} \\ \vdots & \vdots & & \vdots \\ u_p & u_{2p} & \cdots & u_{pq} \end{bmatrix} \tag{26}$$

$$\operatorname{vec}(\operatorname{unvec}(u, p, q)) \triangleq \begin{bmatrix} u_1, & \ldots, & u_{pq} \end{bmatrix}^T. \tag{27}$$

□

We will also find the following fact useful (Bernstein, 2009):

**Fact 2.** *Let $\tilde{p}$, $p$, $q$, and $\tilde{q}$ be positive integers. Also, let $u \in \mathbb{R}^{pq}$, $V \in \mathbb{R}^{\tilde{p}\times p}$, and $W \in \mathbb{R}^{q\times\tilde{q}}$. Then*

$$\operatorname{vec}\Big(V \cdot \operatorname{unvec}(u, p, q) \cdot W\Big) = \left(W^T \otimes V\right)\operatorname{vec}(u). \tag{28}$$

□

The necessary conditions of optimality are summarized in the following lemma:

**Lemma 1.** *Consider Problem 2, where $s$ and $t$ are positive integers, $x \in \mathbb{R}^n$, $G \in \mathbb{R}^{\ell\times(n+1)^s}$, and $H \in \mathbb{R}^{m\times(n+1)^t}$. Also, let*

$$\eta \triangleq n + m + 1 \tag{29}$$

$$r \triangleq \max(2s - 1, t) \tag{30}$$

$$\tilde{G} \triangleq \begin{bmatrix} I_{(n+1)} \\ 0_{m\times(n+1)} \end{bmatrix}^{\otimes 2s} \operatorname{vec}(G^T G) \tag{31}$$

$$\tilde{H} \triangleq \left( \begin{bmatrix} I_{(n+1)} \\ 0_{m\times(n+1)} \end{bmatrix}^{\otimes t} \otimes \begin{bmatrix} 0_{(n+1)\times m} \\ I_m \end{bmatrix} \right) \operatorname{vec}(H) \tag{32}$$

$$\tilde{D} \triangleq \begin{bmatrix} \tilde{G} \\ 0_{(\eta^{r+1}-\eta^{2s})\times 1} \end{bmatrix} + \begin{bmatrix} \tilde{H} \\ 0_{(\eta^{r+1}-\eta^{t+1})\times 1} \end{bmatrix}. \tag{33}$$

*If $x$ is a minimizer of Problem 2, then there exists $\lambda \in \mathbb{R}^m$ such that*

$$D \begin{bmatrix} 1 \\ x \\ \lambda \end{bmatrix}^{\otimes r} = 0_{(n+m)\times 1} \tag{34}$$

*where $\lambda = (\lambda_1, \ldots, \lambda_m)$ is a vector of the Lagrange multipliers, and $D \in \mathbb{R}^{(n+m)\times\eta^r}$ is given by*

$$D \triangleq \begin{bmatrix} 0_{(n+m)\times 1}, & I_{n+m} \end{bmatrix} \tilde{\tilde{D}} \tag{35}$$

$$\tilde{\tilde{D}} \triangleq \operatorname{unvec}\left( \sum_{i=1}^{r+1} \left( I_{\eta^{i-1}} \otimes P_{\eta^{r-i+1},\eta} \right) \tilde{D}, \eta, \eta^r \right). \tag{36}$$

**Proof.** First, let $\lambda \triangleq \begin{bmatrix} \lambda_1 & \cdots & \lambda_m \end{bmatrix}^T$ denote the vector of Lagrange multipliers, and let

$$u \triangleq \begin{bmatrix} 1, & x^T, & \lambda^T \end{bmatrix}^T, \qquad y \triangleq \begin{bmatrix} x^T, & \lambda^T \end{bmatrix}^T.$$

Then from Fact 2, the unconstrained portion of the cost (the first term in (7)) is given by

$$J_{\text{unc}} = \begin{bmatrix} 1, & x^T \end{bmatrix}^{\otimes 2s} \operatorname{vec}(G^T G) = \left(u^{\otimes 2s}\right)^T \tilde{G}$$

where $\tilde{G}$ is given by (31). Thus the Lagrange function is given by

$$\begin{aligned}\Lambda &= J_{\text{unc}} + \lambda^T H \begin{bmatrix} 1 \\ x \end{bmatrix}^{\otimes t} \\ &= J_{\text{unc}} + \left(\begin{bmatrix} 1, & x^T \end{bmatrix}^{\otimes t} \otimes \lambda^T\right) \operatorname{vec}(H) \\ &= J_{\text{unc}} + \begin{bmatrix} 1, & y^T \end{bmatrix}^{\otimes (t+1)} \tilde{H} \\ &= \left(u^{\otimes (r+1)}\right)^T \tilde{D}\end{aligned}$$

where $\tilde{H}$ and $\tilde{D}$ are given by (32) and (33), respectively. Next, differentiating $\Lambda$ with respect to $u$, we find that

$$\frac{\partial \Lambda}{\partial u} = \sum_{i=1}^{r+1} \left(u^{\otimes(i-1)} \otimes I_\eta \otimes u^{\otimes(r+1-i)}\right)^T \tilde{D}.$$

Therefore, from the definition of the Kronecker permutation matrix (see Bernstein, 2009, p. 448):

$$P_{\eta^{r-i+1},\eta}\left(I_\eta \otimes u^{\otimes r-i+1}\right) = u^{\otimes r-i+1} \otimes I_\eta$$

we have that

$$\frac{\partial \Lambda}{\partial u} = \left(u^{\otimes r} \otimes I_\eta\right)^T \sum_{i=1}^{r+1}\left(I_{\eta^{i-1}} \otimes P_{\eta^{r-i+1},\eta}\right)\tilde{D}.$$

Finally, viewing the summation term as a vectorization, and "unveccing" the right-hand side, we have that

$$\frac{\partial \Lambda}{\partial u} = \tilde{\tilde{D}} u^{\otimes r}$$

where $\tilde{\tilde{D}}$ is given by (36). Specifically, the Jacobian with respect to $y$ is given by

$$\frac{\partial \Lambda}{\partial y}^T = D u^{\otimes r}$$

where $D$ is given by (35). Therefore, setting the Jacobian equal to zero, we have (34). □

## 4. Elimination theory

Elimination theory deals with removing variables from systems of multivariable polynomial equations, such as the set of necessary conditions (34). This is normally accomplished through the use of Groebner bases with respect to some type of lexicographic ordering (Cox et al., 2007). However, while the theory is quite powerful, to the knowledge of these authors, all of the algorithms available for computing Groebner bases revolve around symbolic iterative multivariable polynomial division algorithms. Here we will attempt to perform the same basic function of elimination theory (removing variables from systems of multivariable polynomial equations) numerically.

**Definition.** Let $d_1, \ldots, d_p \in \mathbb{R}[y_1, \ldots, y_q]$ have total degrees less than or equal to $r$, let $D \in \mathbb{R}^{p\times(q+1)^r}$, and let $z = (z_1, \ldots, z_q) \in \mathbb{C}^q$. Then

(i) $z$ is a *zero of* $d_1, \ldots, d_p$ if

$$d_1(z) = \cdots = d_p(z) = 0 \tag{37}$$

in which case, we say that $z_i$ is a partial $i$-zero of $d_1, \ldots, d_p$.

(ii) $z$ is an $r$-*zero of* $D$ if

$$D \begin{bmatrix} 1 \\ z \end{bmatrix}^{\otimes r} = 0_{p\times 1} \tag{38}$$

in which case, we say that $z_i$ is a partial $(r, i)$-zero of $D$. □

**Theorem 1.** *Let $d_1, \ldots, d_p \in \mathbb{R}[y_1, \ldots, y_q]$ and let $i \in [1, q]$. If there exist a finite number of partial i-zeros of $d_1, \ldots, d_p$, then there exist $a_{i,1} \ldots, a_{i,p} \in \mathbb{R}[y_1, \ldots, y_q]$ and a nonzero univariate polynomial $c_i \in \mathbb{R}[y_i]$ such that*

$$\sum_{j=1}^{p} a_{i,j} d_j = c_i. \tag{39}$$

*Furthermore, if $z_i \in \mathbb{C}$ is a partial i-zero of $d_1, \ldots, d_p$, then $c_i(z_i) = 0$.*

**Proof.** The result (39) is a direct result of the Hilbert's well-known Nullstellensatz (Cox et al., 2007). □

**Corollary 1.** *Let $y \in \mathbb{R}^q$, $D \in \mathbb{R}^{p\times(q+1)^r}$, and $i \in [1, q]$. If there exist a finite number of partial $(r, i)$-zeros of $D$, then there exist a positive integer $b_i$, a nonzero $A \in \mathbb{R}^{(q+1)^{b_i}\times p}$, and a nonzero $C_i \in \mathbb{R}^{1\times 2^{(b_i+r)}}$ such that*

$$\begin{bmatrix} 1, & y^T \end{bmatrix}^{\otimes b_i} AD \begin{bmatrix} 1 \\ y \end{bmatrix}^{\otimes r} = C_i \begin{bmatrix} 1 \\ y_i \end{bmatrix}^{\otimes (b_i+r)}. \tag{40}$$

*Furthermore, if $z_i \in \mathbb{C}$ is a partial $(r, i)$-zero of $D$, then*

$$C_i \begin{bmatrix} 1 \\ z_i \end{bmatrix}^{\otimes (b_i+r)} = 0. \tag{41}$$

**Proof.** Corollary 1 is a direct result of Theorem 1, where the polynomial notation has been replaced with Kronecker notation using Fact 1. □

Theorem 1 and Corollary 1 show that when there exist a finite number of partial zeros, we can always find a nonzero univariate polynomial which is in the range of the original set of multivariable polynomials. This is beneficial since once the equation set is reduced to a univariate polynomial, we can solve for all of the solutions using standard polynomial root solvers, after which we can combine the partial zeros to determine all of the minimizers of our original optimization problem, namely, Problem 2.

### 4.1. Computing the zeros

Here we introduce an algorithm that uses Corollary 1 to compute the partial zeros of our set of necessary conditions (34), after which we show how to compute all of the zeros of (34).

First, note that from (40):

$$\begin{bmatrix} 1, & y^T \end{bmatrix}^{\otimes b_i} AD \begin{bmatrix} 1 \\ y \end{bmatrix}^{\otimes r} = \begin{bmatrix} 1, & y^T \end{bmatrix}^{\otimes (b_i+r)} \left(D^T \otimes I_{(q+1)^{b_i}}\right) \operatorname{vec}\left(A\right) \tag{42}$$

which, for most matrices $A$, is a polynomial in $y_1, \ldots, y_q$, although from Corollary 1 we know that there exists at least one $b_i$ and $A$ for which (42) is a nonzero univariate polynomial in $y_i$.

Next, note that vectors of the form $\begin{bmatrix} 1, & y^T \end{bmatrix}^{\otimes (b_i+r)}$ contain redundant monomials. For instance, in Example 1, the only monomials that are not redundant are 1, $x_1^2$, and $x_2^2$. The matrix $\Delta_{q,b_i+r}$ is defined to be a binary matrix which combines redundant entries of a vector of this form into a vector with all of the unique monomials in $y_1, \ldots, y_q$. Specifically, if a polynomial $d_i$ is given by

$$d_i = \begin{bmatrix} 1, & y^T \end{bmatrix}^{\otimes (b_i+r)} D_i^T \tag{43}$$

then all of the unique terms of $d_i$ are expressed in the vector

$$\vec{d}_i \triangleq \Delta_{q,b_i+r} \operatorname{diag}\left(\begin{bmatrix} 1, & y^T \end{bmatrix}^{\otimes (b_i+r)}\right) D_i^T. \tag{44}$$

For instance, if $q = b_i + r = 2$ and

$$D_i = \begin{bmatrix} 1, & 2, & 3, & 4, & 5, & 6, & 7, & 8, & 9 \end{bmatrix} \tag{45}$$

then

$$\Delta_{2,2} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{46}$$

$$d_i = 1 + 6y_1 + 10y_2 + 5y_1^2 + 14y_1y_2 + 9y_2^2 \tag{47}$$

$$\vec{d}_i = \begin{bmatrix} 1, & 6y_1, & 10y_2, & 5y_1^2, & 14y_1y_2, & 9y_2^2 \end{bmatrix}^T. \tag{48}$$

Finally, note that all of the monomials in $y_i$, such as $1, y_i, y_i^2, \ldots,$ can be extracted from $\begin{bmatrix} 1, & y^T \end{bmatrix}^{\otimes(b_i+r)}$ using the diagonal matrix:

$$\Psi_{q,b_i+r,i} \triangleq \operatorname{diag}\left(\begin{bmatrix} 1, & 0_{1\times(i-1)}, & 1, & 0_{1\times(q-i)} \end{bmatrix}^{\otimes(b_i+r)}\right). \tag{49}$$

Furthermore, the vector

$$e \triangleq \begin{bmatrix} 1, & y^T \end{bmatrix}^{\otimes(b_i+r)} \left(I - \Psi_{q,b_i+r,i}\right) \tag{50}$$

contains all of the monomials in $y_1, \ldots, y_q$, excluding the monomials in $y_i$. Hence a matrix $A$ for which (42) evaluates to a nonzero univariate polynomial in $y_i$, is one for which

$$\Delta_{q,b_i+r}\left(I - \Psi_{q,b_i+r,i}\right)\left(D^T \otimes I_{(q+1)^{b_i}}\right)\operatorname{vec}(A) = 0 \tag{51}$$

and

$$\Delta_{q,b_i+r}\Psi_{q,b_i+r,i}\left(D^T \otimes I_{(q+1)^{b_i}}\right)\operatorname{vec}(A) \neq 0. \tag{52}$$

The only remaining issue is to determine $b_i$. However, this is solved by incrementing $b_i$ until a feasible solution is found.

**Algorithm 1.** Let $D \in \mathbb{R}^{p\times(q+1)^r}$ and $i \in [1, q]$. Also, assume that there exist a finite number of partial $(r, i)$-zeros of $D$. Then the following algorithm yields a set $\mathcal{Z}_i$ which contains the partial $(r, i)$-zeros of $D$, that is, if $z_i \in \mathbb{C}$ is a partial $(r, i)$-zero of $D$, then $z_i \in \mathcal{Z}_i$.

(1) Set $b_i = 0$.
(2) Increment $b_i$ by 1.
(3) Compute a basis $V \in \mathbb{R}^{p(q+1)^{b_i}\times\nu}$ for the nullspace of

$$\Delta_{q,b_i+r}\left(I - \Psi_{q,b_i+r,i}\right)\left(D^T \otimes I_{(q+1)^{b_i}}\right). \tag{53}$$

(4) If $V$ is empty ($\nu = 0$), or

$$\Delta_{q,b_i+r}\Psi_{q,b_i+r,i}\left(D^T \otimes I_{(q+1)^{b_i}}\right)V = 0 \tag{54}$$

return to step 2.
(5) Set

$$C = \left[\Psi_{q,b_i+r,i}\left(D^T \otimes I_{(q+1)^{b_i}}\right)V\right]^T \tag{55}$$

where $C$ is a matrix of coefficients for a set of univariate polynomial equations in $y_i$, that is,

$$C\begin{bmatrix} 1 \\ y \end{bmatrix}^{\otimes(b_i+r)} = \begin{bmatrix} c_1 \\ \vdots \\ c_\nu \end{bmatrix} \tag{56}$$

where $c_1, \ldots, c_\nu \in \mathbb{R}[y_i]$.
(6) Compute the set $Z_1$ of zeros of $c_1$ using a univariate polynomial root solver.
(7) Set $j = 1$ and $\mathcal{Z}_{i,1} = Z_1$.
(8) Increment $j$ by 1.
(9) Compute the set $Z_j$ of zeros of $c_j$ using a univariate polynomial root solver.
(10) Set $\mathcal{Z}_{i,j} = \mathcal{Z}_{i,j-1} \cap Z_j$.
(11) If $j < \nu$ and $\mathcal{Z}_{i,j}$ is not an empty set, return to step 8.
(12) Return $\mathcal{Z}_i \triangleq \mathcal{Z}_{i,j}$, where $\mathcal{Z}_i$ denotes a set which contains the partial $(r, i)$-zeros of $D$. □

**Remark.** From Theorem 1 and Corollary 1, we are guaranteed that there will exist a nonnegative $b_i$ and a nonzero $C$ in Algorithm 1. □

**Remark.** Once a nonzero $C$ has been determined in Algorithm 1, there are several ways of determining the set $\mathcal{Z}_i$. An alternative method is to choose a univariate polynomial $c_i$ in (56), compute the zeros of $c_i$, and choose one of the zeros $z_i$ of $c_i$. Then $z_i \in \mathcal{Z}_i$ if $c_1(z_i) = \cdots = c_\nu(z_i) = 0$. In this way, looping over all of the zeros of $c_i$, we could determine the set $\mathcal{Z}_i$. □

Next, we put together a simple algorithm for determining all of the local and global minimizers of Problem 2. Note that usually we do not explicitly need to know the Lagrange multipliers, and hence we do not need to use Algorithm 1 to determine the partial zeros of the necessary condition equations (34) corresponding to the Lagrange multipliers.

**Algorithm 2.** Consider Problem 2 and the necessary conditions of optimality for its solution (Lemma 1). Furthermore, assume that there exist a finite number of minimizers of Problem 2. Then the following algorithm yields the set $\mathcal{Z}$ of local and global minimizers of Problem 2.

(1) Set $i = 0$.
(2) Increment $i$ by 1.
(3) Apply Algorithm 1 to $D$, yielding the set of partial solutions $\mathcal{Z}_i$.
(4) Let $\xi_i$ denote the number of elements of $\mathcal{Z}_i$.
(5) If $i < n$, return to step 2.
(6) Construct the set $\mathcal{P}$ of all of the $\xi_1\xi_2\cdots\xi_q$ combinations possible by choosing one element from each $\mathcal{Z}_i$.
(7) Set $j = 0$ and $\mathcal{Z} = \{\}$, the empty set.
(8) Increment $j$ by 1.
(9) Choose an element of $y \in \mathcal{P}$ and remove $y$ from $\mathcal{P}$.
(10) If $y$ is a $t$-zero of $H$, add $y$ to the set $\mathcal{Z}$.
(11) If $j < \xi_1\xi_2\cdots\xi_q$, return to step 8.
(12) Return $\mathcal{Z}$, where $\mathcal{Z}$ denotes the set of local and global minimizers of Problem 2. □

## 5. Sparse nullspace calculation

By far, the most computationally expensive step in Algorithm 1 is the computation of the nullspace of (53), which is particularly difficult since (53) has $p(q+1)^{b_i}$ columns. Therefore if the problem requires a large $b_i$ (which is unknowable *a priori*), then the dimensions can become very large very fast. However, (53) also becomes sparse as $b_i$ increases, as evidenced by the product $D^T \otimes I_{(q+1)^{b_i}}$ in (53). Hence the practicality of Algorithm 2, and thus the solvability of Problem 2 using the present non-Groebner-based approach revolves around our ability to compute the nullspace of large space matrices reliably.

Unfortunately, computing the nullspace of a large sparse matrix is not a straightforward matter, since the most numerically reliable methods, the singular value and QR decomposition, are typically infeasible from a memory and computation point of view. This is primarily because the nullspace in both of these algorithms is orthogonal, and hence the sparsity of the original matrix is typically not passed along to the nullspace. Here we propose an alternative method for computing the nullspace of large sparse matrices. First, consider the following fact:

**Fact 3.** *Let $A \in \mathbb{R}^{\ell\times k}$ and $a = (a_1,\ldots,a_k) \in \mathbb{R}^{1\times k}$, where $a$ is nonzero and $j$ denotes the index of the largest element of $a$, that is,*

$$|a_j| \triangleq \max(|a_1|,\ldots,|a_k|). \tag{57}$$

*Also, let*

$$d \triangleq \left[(a_1/a_j),\ \ldots,\ (a_{j-1}/a_j)\right] \in \mathbb{R}^{1\times(j-1)} \tag{58}$$

$$e \triangleq \left[(a_{j+1}/a_j),\ \ldots,\ (a_k/a_j)\right] \in \mathbb{R}^{1\times(k-j)} \tag{59}$$

$$f \triangleq \begin{bmatrix} d, & e\end{bmatrix} \in \mathbb{R}^{1\times(k-1)} \tag{60}$$

$$U \triangleq \begin{bmatrix} I_{j-1} & 0_{(j-1)\times(k-j)} \\ -d & -e \\ 0_{(k-j)\times(j-1)} & I_{k-j}\end{bmatrix} \in \mathbb{R}^{k\times(k-1)} \tag{61}$$

*and let $V \in \mathbb{R}^{(k-1)\times\nu}$ be a basis for the nullspace of $AU$. Then*

(i) *$U$ is a basis for the nullspace of $a$.*
(ii) *$V' \triangleq UV$ is a basis for the nullspace of $A' \triangleq \begin{bmatrix} a \\ A\end{bmatrix}$.*
(iii) *The singular values of $U$ are given by*

$$\sigma_1 = \sqrt{1+ff^T}, \qquad \sigma_2 = \cdots = \sigma_{k-1} = 1. \quad \square$$

**Proof.** First, since $a$ is nonzero, the dimension of the nullspace of $a$ is $k-1$. Furthermore, since $\operatorname{rank}[U] = k-1$ and $aU = 0_{1\times(k-1)}$, it follows that $U$ is a basis for the nullspace of $a$.

Next, suppose that $y \in \mathbb{R}^k$ is in the nullspace of $A'$. Then $ay = 0$ and $Ay = 0$. Furthermore, since $U$ is a basis for the nullspace of $a$, there exists $w \in \mathbb{R}^{k-1}$ such that $y = Uw$. Finally, since $Ay = AUw = 0_{\ell\times 1}$ and $V$ is a basis for the nullspace of $AU$, there exists $z \in \mathbb{R}^\nu$ such that $w = Vz$. Therefore if $y$ is in the nullspace of $A'$, then $y$ is of the form $y = UVz$, that is, $V' = UV$ is a basis for the nullspace of $A'$.

Finally, recall that the singular values of $U$ are the square roots of the eigenvalues of $U^TU$, where

$$U^TU = \begin{bmatrix} I_{j-1} + d^Td & d^Te \\ e^Td & I_{k-j}+e^Te\end{bmatrix} = I_{k-1} + f^Tf.$$

Hence if $f = 0$, then all of the $k-1$ singular values of $U$ are 1. Otherwise,

$$U^TUf^T = \left(1+ff^T\right)f^T$$

and hence one of the singular values is given by $\sqrt{1+ff^T}$, while the remaining $k-2$ singular values are given by 1. $\square$

**Remark.** If $a = 0_{1\times k}$, then the nullspace of $A$ is the same as the nullspace of $\begin{bmatrix} a \\ A\end{bmatrix}$ since the nullspace of $0_{1\times k}$ is the $k\times k$ identity matrix. $\square$

**Algorithm 3.** Let $A_1,\ldots,A_\ell \in \mathbb{R}^{1\times k}$ and

$$A \triangleq \begin{bmatrix} A_1 \\ \vdots \\ A_\ell\end{bmatrix}. \tag{62}$$

Then the following algorithm, based on Fact 3, yields a basis $V$ for the nullspace of $A$.

(1) Set $i = 0$, $V_1 = I_k$, and $\nu_1 = k$.
(2) Increment $i$ by 1.
(3) Set $\begin{bmatrix} b_1 & \cdots & b_{\nu_i}\end{bmatrix} = A_iV_i$, where $b_1,\ldots,b_{\nu_i} \in \mathbb{R}$.
(4) If $b_1 = \cdots = b_{\nu_i} = 0$, return to step 2.
(5) Determine the index $j$ of the largest element of $b_1,\ldots,b_{\nu_i}$, that is,

$$|b_j| \triangleq \max\left(|b_1|,\ldots,|b_{\nu_i}|\right). \tag{63}$$

(6) Set $d$, $e$, and $U$ to be given by

$$d = \left[(b_1/b_j),\ \ldots,\ (b_{j-1}/b_j)\right] \tag{64}$$

$$e = \left[(b_{j+1}/b_j),\ \ldots,\ (b_{\nu_i}/b_j)\right] \tag{65}$$

$$U = \begin{bmatrix} I_{j-1} & 0_{(j-1)\times(\nu_i-j)} \\ -d & -e \\ 0_{(\nu_i-j)\times(j-1)} & I_{\nu_i-j}\end{bmatrix}. \tag{66}$$

(7) Update the nullspace, that is, set $V_i = V_{i-1}U$ and $\nu_i = \nu_{i-1} - 1$.
(8) If $i < \ell$, return to step 2.
(9) Return $V = V_i$, where $V$ denotes a basis for the nullspace of $A$. $\square$

**Remark.** By examining the structure of $U$ in Fact 3 and Algorithm 3, we can see that, at each step of Algorithm 3, the nullspace $V_i \in \mathbb{R}^{k\times\nu_i}$ of $\begin{bmatrix} A_1^T & \cdots & A_i^T\end{bmatrix}^T$ is sparse. In particular, we have that

$$\#\text{ of nonzero entries of } V_i \le \nu_i\,(k-\nu_i+1),$$

where, in general, the bound is reached only if $A$ is dense. Hence the density of $V_i$ is less than or equal to $(k-\nu_i+1)/k$. $\square$

### 5.1. Lexicographic ordering

Further reduction in the computational complexity of Algorithm 1 can be achieved by rephrasing our problem in terms of a type of lexicographic ordering. Specifically, as demonstrated in (43)–(48), there are, in general, redundant terms in vectors of the form $\begin{bmatrix} 1, & x^T\end{bmatrix}^{\otimes s}$. These redundant terms artificially increase the dimensions of the matrix that we need to compute the nullspace of. The following table shows the size of $\begin{bmatrix} 1, & x^T\end{bmatrix}^{\otimes s}$, along with its lexicographic size in parentheses, where $x \in \mathbb{R}^n$:

| $n$ | $s=2$ | $s=3$ | $s=4$ | $s=5$ |
|---|---|---|---|---|
| 2 | 9(6) | 27(10) | 81(15) | 243(21) |
| 5 | 36(21) | 216(56) | 1296(126) | 7776(252) |
| 10 | 121(66) | 1331(286) | 14 641(1001) | 161 051(3003) |

Optimized implementations of the aforementioned algorithms would save all of the internally computed matrices in some type of lexicographic ordering. However, like every algorithm, there will always remain a practical limit on the size of problems that we can solve.

## 6. Nonlinear ARX model identification

Consider the nonlinear ARX system:

$$y_k = ay_{k-1} + bu_k + a^2u_k^2 + aby_{k-1}u_k + acy_{k-1}^2 + v_k \tag{67}$$

where $a$, $b$, and $c$ are unknown, $u \in \mathbb{R}$ denotes a known input, $y \in \mathbb{R}$ denotes a measured output, and $v \in \mathbb{R}$ denotes an unknown i.i.d. zero-mean Gaussian noise sequence with variance $\sigma_v^2$. Furthermore, let $u_0,\ldots,u_N$ and $y_0,\ldots,y_N$ be measured, and let

$$a = 0.1, \qquad b = -1, \qquad c = -0.01, \qquad y_0 = 0, \qquad N = 100$$

$$u_k = \exp(-0.007k^2) + \sin\left(\frac{k}{2}\right) - \cos\left(\frac{k}{3}\right).$$

Then the triple $(a,b,c)$ can be estimated by solving Problem 2, where $n = 3$, $s = 2$, $H = 0$, and

$$x \triangleq \begin{bmatrix} a, & b, & c\end{bmatrix}$$

$$G_i \triangleq \begin{bmatrix} -y_i, & y_{i-1}, & u_i, & 0_{1\times 2}, & u_i^2, & y_{i-1}u_i, & y_{i-1}^2, & 0_{1\times 8}\end{bmatrix} \tag{68}$$

where $G_i$ denotes the $i$th row of $G \in \mathbb{R}^{N\times 16}$. Specifically, using a Python-based implementation of Algorithms 1–3, this takes approximately 0.7 s on the author's computer.

**Remark.** Since Python is not a compiled language, a C or Fortran-based implementation of the nullspace algorithm could see a significant speed improvement. $\square$

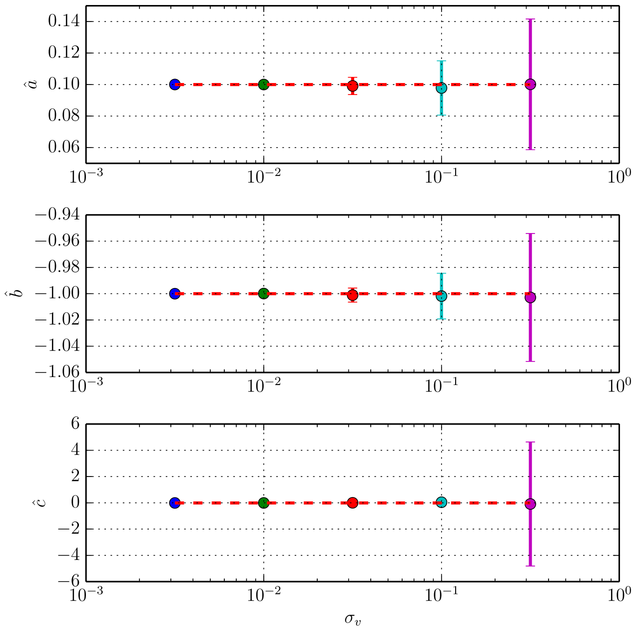


**Fig. 1.** Mean and standard deviation of the global minimizers of Problem 2 for the system (67) with several noise standard deviations $\sigma_v$, and 200 realizations of the noise sequence $v$ for each standard deviation.

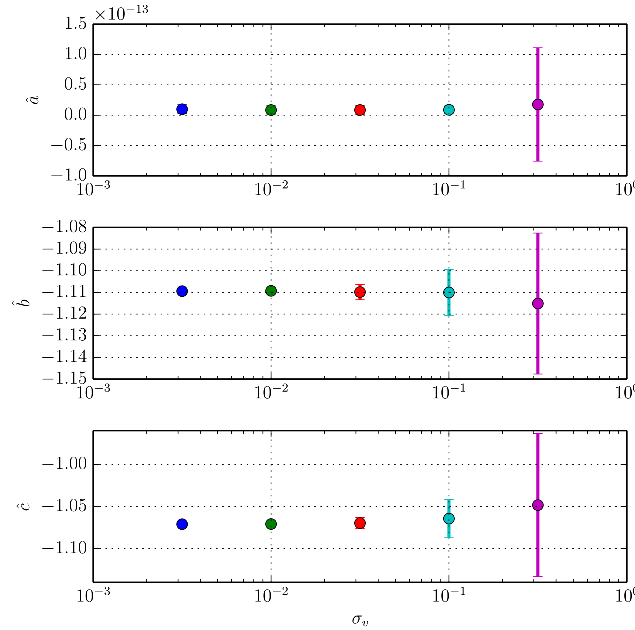


**Fig. 2.** Mean and standard deviation of the local minimizers of Problem 2 for the system (67) with several noise standard deviations $\sigma_v$, and 200 realizations of the noise sequence $v$ for each standard deviation. Note that in some cases, there was only a global minimizer.

### 6.1. Global and local minimizers

Algorithms 1–3 yield all of global and local minimizers of Problem 2. For example, when there is no noise ($\sigma_v = 0$) in the system (67), and the rows of $G$ in Problem 2 are given by (68), then there is only one minimizer of Problem 2. Specifically, the global minimizer is the exact triple $(a, b, c) = (0.1, -1, -0.01)$. Hence Algorithms 1–3 return one estimate: $(\hat{a}, \hat{b}, \hat{c}) = (0.1, -1, -0.01) \pm$ 1e−15.

When there is noise present ($\sigma_v > 0$), there can exist additional local minimizers of Problem 2, in which case, Algorithms 1–3 will return more than one estimate. In this case, we can discern which estimates correspond to the local and global minimizers by evaluating the cost function at the estimates. Naturally, the estimate with the lowest cost function is referred to as the global minimizer, while the others are referred to as local minimizers. Note that due to the presence of noise, the minimizing cost will generally not be zero, and the global minimizer will generally not be equal to the exact solution, that is, $(\hat{a}, \hat{b}, \hat{c})_{global} \neq (a, b, c)$.

To demonstrate this further, we consider several values of the noise standard deviation in (67), and 200 realizations of the noise sequence $v$ for each standard deviation. For each realization of $v$, we compute the minimizers of Problem 2 using Algorithms 1–3. The mean and standard deviation of the estimates which correspond to the global minimizers are shown in Fig. 1, along with dotted lines indicating the exact $(a, b, c)$. From Fig. 1, we can see the variances of the global minimizers increase with increasing noise variance, as expected.

For some realizations of $v$, an additional local minimizer of Problem 2 appeared. The mean and standard deviation of the estimates which correspond to this local minimizer are shown in Fig. 2. From Fig. 2, we can see that the local minimizer seems to reoccur at approximately $(\hat{a}, \hat{b}, \hat{c}) = (0, -1.11, -1.07)$. In Section 6.3, we show that local optimization techniques could get stuck at this local minimizer.

Finally, Fig. 3 shows the mean and standard deviation of the cost function in Problem 2, evaluated at the global and local minimizers. From Fig. 3, we can see that the global minimizer cost increases with increasing noise variance, which demonstrates the accumulation of noise in the residuals. However, we can see that the local minimizer cost has a systematic error since the cost function does not approach zero with decreasing noise variance. Note that the standard deviation of the cost of the local minimizers appears to be zero. However, this is due to the logarithmic axis and the fact that the standard deviation is much smaller than the magnitude.

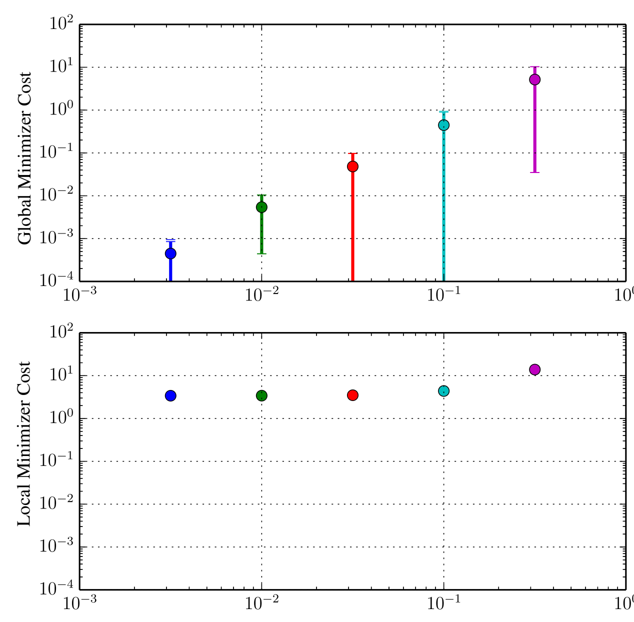


**Fig. 3.** Mean and standard deviation of the cost function in Problem 2 evaluated at the global and local minimizers shown in Figs. 1 and 2.

### 6.2. Groebner basis solution

Computing a Groebner basis for the Lagrange necessary conditions of optimality of Problem 2 will return exactly the same solutions that our method generated since they are both based on

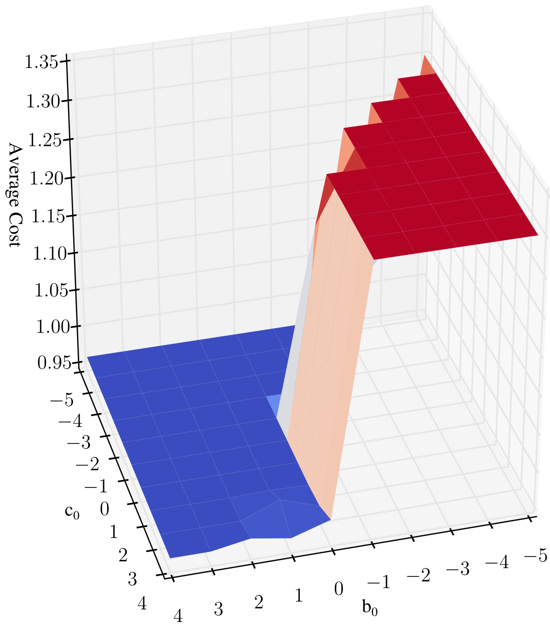


**Fig. 4.** Mean value of Problem 2 cost function evaluated at the minimizers estimated by Levenberg–Marquardt when $\sigma_v = 10^{-1}$ and the initial guess of $\hat{a}$ is $a_0 = 0$. The mean value is shown as a function of the initial guesses $b_0$ and $c_0$.

the same principal. However, all of the Groebner-based implementations that these authors are aware of use symbolic multivariable polynomial division algorithms, which tend to be impractical for typical engineering problems. For instance, when trying to solve the previous problem using the Groebner basis calculator in Python's SymPy, the algorithm did not converge after 10 min of runtime (as opposed to the 0.7 s to run our algorithms, which are written in uncompiled code). Hence we do not consider this to be a feasible alternative.

### 6.3. Local optimization methods

There are a myriad of local optimization methods that we could use to solve Problem 2, however, here we choose the Levenberg–Marquardt implementation in Scipy's Optimize package. Unlike our methods, local optimization techniques require an initial guess of the solution. We use the initial guess of $\hat{a} = 0$, while allowing the initial guesses for $\hat{b}$ and $\hat{c}$ to be in the range $[-5, 4]$. Furthermore, we consider 200 realization of the noise sequence $v$ with standard deviation $\sigma_v = 10^{-1}$.

Minimizing Problem 2 with the Levenberg–Marquardt algorithm (and the rows of $G$ given by (68)), Fig. 4 shows the mean value of the cost function evaluated at the optimized values. From the figure, we can see two distinct plateaus, where the lower plateau corresponds to the average cost of the global minimizer, and the higher plateau corresponds to the cost of the local minimizer. This demonstrates that whereas Algorithms 1–3 always provided both the local and global minimizers, local optimization methods are strongly dependent on the initial guess, and could easily get stuck at a local minimizer.

**Remark.** Algorithms 1–3 take approximately 0.7 s to run. The Levenberg–Marquardt algorithm we used took approximately 0.007 s for a single case. However, since we considered a $10 \times 10$ grid of initial conditions for the Levenberg–Marquardt algorithm, both methods required the same amount of time to complete. □

## 7. Conclusion

We presented an elimination theory-based method for solving equality-constrained multivariable polynomial least-squares problems, that is, for determining all of the local and global minimizers when a finite number of them exist. Furthermore, we showed that this problem amounts to computing the nullspace of a large sparse matrix, and then computing the zeros of a scalar, univariate polynomial.

## 8. Future work

Future work will focus on removing the assumption that there exist a finite number of local minimizers, and providing a more detailed analysis of the numerical properties of our nullspace algorithm.

## References


Bernstein, D. S. (2009). *Matrix mathematics* (2nd ed.). Princeton, NJ: Princeton University Press.

Box, G. E. P., Jenkins, G. M., & Reinsel, G. C. (2008). *Time series analysis: forecasting and control* (4th ed.). Wiley.

Brockwell, P. J., & Davis, R. A. (2006). *Time series: theory and methods* (2nd ed.). New York: Springer-Verlag.

Buchberger, B. (1985). Groebner bases: an algorithmic method in polynomial ideal theory. In N. K. Bose (Ed.), *Multidimensional systems theory: progress, directions and open problems in multidimensional systems*.

Cox, D., Little, J., & O'Shea, D. (2007). *Ideals, varieties, and algorithms* (3rd ed.). New York: Springer-Verlag.

Gohberg, I., Lancaster, P., & Rodman, L. (2009). *Matrix polynomials*. Philadelphia: SIAM.

Ho, B. L., & Kalman, R. E. (1966). Effective construction of linear state-variable models from input/output functions. *Regelungstechnik*, *14*(12), 545–592.

Holzel, M.S., & Bernstein, D.S. (2011). SVD-based computation of zeros of polynomial matrices. In *IEEE conference on decision and control*. Orlando, FL, December (pp. 6962–6966).

Holzel, M. S., & Bernstein, D. S. (2012). From polynomial matrices to Markov parameters and back: theory and numerical algorithms. *Linear Algebra and its Applications*, *437*, 783–808.

Ljung, L. (1999). *System identification: theory for the user* (3rd ed.). Upper Saddle River, NJ: Prentice-Hall.

Nocedal, J., & Wright, S. J. (2006). *Numerical optimization* (2nd ed.). Springer-Verlag.

Pintelon, R., & Schoukens, J. (2001). *System identification: a frequency domain approach* (1st ed.). New York: Wiley–IEEE Press.

Speyer, J. L., & Chung, W. H. (2008). *Stochastic processes, estimation, and control* (1st ed.). Philadelphia: SIAM.

Wang, L., & Garnier, H. (Eds.) (2012). *System identification, environmental modelling, and control system design*. Springer-Verlag.


**Matthew S. Hölzel** leads the *Parallel Computing for Embedded Sensor Systems* Research group at the University of Bremen, in collaboration with the German Aerospace Center (DLR). He received a Ph.D. in Aerospace Engineering from the University of Michigan, Ann Arbor in 2012. His research interests are parallel computing, system identification, and control.

**Dennis S. Bernstein** is a professor in the Aerospace Engineering Department at the University of Michigan, where he received his Ph.D. in 1982. His research interests are in adaptive control and system identification. He is the author of the reference work *Matrix Mathematics* published by Princeton University Press. He was Editor in Chief of the *IEEE Control Systems Magazine* from 2003 to 2011. He has advised more than 35 Ph.D. students.