Proceedings of the 17th International Symposium on Mathematical
Theory of Networks and Systems, Kyoto, Japan, July 24-28, 2006

An algorithm for approximate common divisor computation

Ivan Markovsky and Sabine Van Huffel

K.U.Leuven, ESAT-SCD, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium,
{ivan.markovsky,sabine.vanhuffel}@esat.kuleuven.ac.be

***Abstract*— In *SIAM J. Matrix Anal. Appl.*, 26(4):1083–1099, 2005, we presented an algorithm for solving Toeplitz structured total least squares problems. The computation of an approximate common divisor of two polynomials is a Sylvester structured total least squares problem. In this paper we adapt the algorithm developed for Toeplitz matrices for the purpose of computing an approximate common divisor of two scalar polynomials. Per iteration the proposed algorithm has linear computational complexity in the degree of the given polynomials.**

Keywords: Approximate common divisor, distance to uncontrollability, structured low rank approximation, total least squares.

I. Introduction

Let $\mathbb{P}_n$ be the set of all scalar polynomials of degree less than or equal to n, i.e.,

$$\mathbb{P}_n := \{\, p \in \mathbb{R}[\xi] \mid \text{degree}(p) \leq n \,\}.$$

The set $\mathbb{P}_n$ is isomorphic to $\mathbb{R}^{n+1}$. Associated with

$$p(\xi) := p_0 + p_1\xi + \cdots + p_n\xi^n \in \mathbb{P}_n$$

is a coefficient vector

$$p := \text{col}(p_0, p_1, \ldots, p_n) \in \mathbb{R}^{n+1}$$

and vice versa. We say that the polynomials $p(\xi) \in \mathbb{P}_n$ and $\hat{p}(\xi) \in \mathbb{P}_n$ are “close” to each other if the distance measure

$$\text{dist}\big(p(\xi), \hat{p}(\xi)\big) := \|p - \hat{p}\|_2^2$$

is “small”, i.e., if the sum of squared coefficients of the difference $p(\xi) - \hat{p}(\xi)$ is small.

Note 1. $\text{dist}\big(p(\xi), \hat{p}(\xi)\big)$ might not be an appropriate distance measure in applications where the polynomial roots rather than coefficients are of primary interest. Polynomial roots might be sensitive (especially for high order polynomials) to perturbations in the coefficients, so that closeness of coefficients does not necessarily imply closeness of roots. Using the quadratic distance measure in terms of the polynomial coefficients, however, simplifies the solution of the approximate common divisor problem defined next.

Problem 1 (Approximate common divisor)**.** Given $a(\xi), b(\xi) \in \mathbb{P}_n$, and $d \in \mathbb{N}$, find polynomials $\hat{a}(\xi), \hat{b}(\xi) \in \mathbb{P}_n$ that have a common divisor $c(\xi)$ of degree d and minimize the approximation error $\text{dist}\big(a(\xi), \hat{a}(\xi)\big) + \text{dist}\big(b(\xi), \hat{b}(\xi)\big)$. The polynomial $c(\xi)$ is an optimal (in the specified sense) approximate common divisor of $a(\xi)$ and $b(\xi)$.

Note 2. The object of interest in solving Problem 1 is the approximate common divisor $c(\xi)$. The approximating polynomials $\hat{a}(\xi)$ and $\hat{b}(\xi)$ are auxiliary variables introduce for the purpose of defining $c(\xi)$.

Note 3. In the generic case when $a(\xi)$ and $b(\xi)$ have no common divisor of degree greater than d, $c(\xi)$ can be called approximate *greatest* common divisor of $a(\xi)$ and $b(\xi)$. We prefer to skip the word greatest in order to account for the case when $c(\xi)$ is a factor of the (exact) greatest common divisor.

Problem 1 has the following system theoretic interpretation. Let σ be the forward shift operator $(\sigma u)(t) := u(t+1)$ in the discrete-time case and the derivative operator $\sigma u := du/dt$ in the continues-time case. Consider the single-input single-output linear time-invariant (LTI) system $\mathscr{B}$ described by the difference or differential equation $a(\sigma)u = b(\sigma)y$. It is well known that the system $\mathscr{B}$ is controllable if and only is $a(\xi)$ and $b(\xi)$ have no common factor. Therefore, Problem 1 has the system theoretic meaning of finding the nearest uncontrollable system $\hat{\mathscr{B}}$ (described by $\hat{a}(\sigma)u = \hat{b}(\sigma)y$) to a given LTI system. The bigger the approximation error $f(c)$ is, the more robust the controllability property of $\mathscr{B}$ is. In particular, with $f(c) = 0$, $\mathscr{B}$ is uncontrollable. Since the order of $\hat{\mathscr{B}}$ is at most $n-d$, Problem 1 has relevance for model reduction.

II. Equivalent optimization problem

By definition, the polynomial $c(\xi) \in \mathbb{R}[\xi]$ is a common divisor of $\hat{a}(\xi)$ and $\hat{b}(\xi)$ if there are $v(\xi), u(\xi) \in \mathbb{R}[\xi]$, such that

$$\hat{a}(\xi) = u(\xi)c(\xi), \qquad \hat{b}(\xi) = v(\xi)c(\xi). \tag{1}$$

With the additional auxiliary variables $v(\xi)$ and $u(\xi)$, Problem 1 becomes the following optimization problem:

$$\min_{\substack{\hat{a}(\xi),\hat{b}(\xi)\in\mathbb{P}_n \\ u(\xi),v(\xi),c(\xi)\in\mathbb{R}[\xi]}} \operatorname{dist}\big(a(\xi),\hat{a}(\xi)\big) + \operatorname{dist}\big(b(\xi),\hat{b}(\xi)\big) \quad \text{subject to} \quad \begin{array}{l} \hat{a}(\xi) = u(\xi)c(\xi) \\ \hat{b}(\xi) = v(\xi)c(\xi) \\ \operatorname{degree}\big(c(\xi)\big) = d \end{array} \tag{2}$$

If $d > n$, Problem 1 has no solution and if $d = n$, it has a trivial solution. Therefore we can assume without loss of generality that $d < n$.

Theorem 1. *The optimization problem (2) is equivalent to*

$$\min_{c_0,\dots,c_{d-1}\in\mathbb{R}} \operatorname{trace}\Big(\begin{bmatrix} a & b \end{bmatrix}^\top \Big(I - T(c)\big(T^\top(c)T(c)\big)^{-1}T^\top(c)\Big) \begin{bmatrix} a & b \end{bmatrix}\Big), \tag{3}$$

where $T(c) \in \mathbb{R}^{(n+1)\times(n-d+1)}$ is a lower triangular banded Toeplitz matrix with first column equal to

$$\operatorname{col}(c_0,\dots,c_{d-1},1,0,\dots,0).$$

Proof: The polynomial equations (1) are equivalent to the following systems of algebraic equations

$$\begin{bmatrix} \hat{a}_0 \\ \hat{a}_1 \\ \vdots \\ \hat{a}_n \end{bmatrix} = \underbrace{\begin{bmatrix} u_0 & & & \\ u_1 & u_0 & & \\ \vdots & u_1 & \ddots & \\ u_{n-d} & \vdots & \ddots & u_0 \\ & u_{n-d} & & u_1 \\ & & \ddots & \vdots \\ & & & u_{n-d} \end{bmatrix}}_{T(u)} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_d \end{bmatrix}, \qquad \begin{bmatrix} \hat{b}_0 \\ \hat{b}_1 \\ \vdots \\ \hat{b}_n \end{bmatrix} = \underbrace{\begin{bmatrix} v_0 & & & \\ v_1 & v_0 & & \\ \vdots & v_1 & \ddots & \\ v_{n-d} & \vdots & \ddots & v_0 \\ & v_{n-d} & & v_1 \\ & & \ddots & \vdots \\ & & & v_{n-d} \end{bmatrix}}_{T(v)} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_d \end{bmatrix}. \tag{4}$$

(All missing elements are zeros.) Rewriting and combining the above equations, we have that $c(\xi) \in \mathbb{R}[\xi]$ is a common factor (with $\operatorname{degree}(c(\xi)) \leq d$) of $\hat{a}(\xi)$ and $\hat{b}(\xi)$ if and only if the system of equations

$$\begin{bmatrix} \hat{a}_0 & \hat{b}_0 \\ \hat{a}_1 & \hat{b}_1 \\ \vdots & \vdots \\ \hat{a}_n & \hat{b}_n \end{bmatrix} = \underbrace{\begin{bmatrix} c_0 & & & \\ c_1 & c_0 & & \\ \vdots & c_1 & \ddots & \\ c_{d-1} & \vdots & \ddots & c_0 \\ c_d & c_{d-1} & & c_1 \\ & c_d & \ddots & \vdots \\ & & \ddots & c_{d-1} \\ & & & c_d \end{bmatrix}}_{T(c)} \begin{bmatrix} u_0 & v_0 \\ u_1 & v_1 \\ \vdots & \vdots \\ u_{n-d} & v_{n-d} \end{bmatrix}$$

has a solution.

The condition $\operatorname{degree}(c(\xi)) = d$ implies that the highest power coefficient c_d of $c(\xi)$ is different from 0. Since $c(\xi)$ is determined up to a scaling factor, we can impose the normalization $c_d = 1$. Conversely, imposing the constraint $c_d = 1$ in the optimization problem to be solved ensures that $\operatorname{degree}(c(\xi)) = d$. Therefore problem (2) is equivalent to

$$\min_{\substack{\hat{a},\hat{b}\in\mathbb{R}^{n+1} \\ u,v\in\mathbb{R}^{n-d+1} \\ c_0,\dots,c_{d-1}\in\mathbb{R}}} \operatorname{trace}\Big(\big(\begin{bmatrix} a & b \end{bmatrix} - \begin{bmatrix} \hat{a} & \hat{b} \end{bmatrix}\big)^\top \big(\begin{bmatrix} a & b \end{bmatrix} - \begin{bmatrix} \hat{a} & \hat{b} \end{bmatrix}\big)\Big) \quad \text{subject to} \quad \begin{bmatrix} \hat{a} & \hat{b} \end{bmatrix} = T(c)\begin{bmatrix} u & v \end{bmatrix}.$$

(The variable c_d appearing in $T(c)$ has been substituted with 1.) Substituting $\begin{bmatrix} \hat{a} & \hat{b} \end{bmatrix}$ in the cost function and minimizing with respect to $\begin{bmatrix} u & v \end{bmatrix}$ by solving a least squares problem gives the equivalent problem (3).

Theorem 1 is the best we could do in attempting to solve Problem 1 analytically. Compared with the original optimization problem (2), in (3) we have eliminated the constraint and the auxiliary decision variables $\hat{a}$, $\hat{b}$, u, and v. This already achieves a significant simplification from a numerical optimization point of view. The equivalent problem (3) is a nonlinear least squares problem and can be solved by standard local optimization methods. Define

$$f(c) := \operatorname{trace}\Big(\begin{bmatrix} a & b \end{bmatrix}^\top \Big(I - T(c)\big(T^\top(c)T(c)\big)^{-1}T^\top(c)\Big)\begin{bmatrix} a & b \end{bmatrix}\Big),$$

to be the cost function of (3). The resulting algorithm for approximate common divisor computation is outlined in Algorithm 1.

Algorithm 1 Optimal approximate common divisor computation.

Input: Vectors $a, b \in \mathbb{R}^{n+1}$ and an integer d.

Compute an initial approximation $c_{\text{ini}} \in \mathbb{R}^{d+1}$.

Execute a standard optimization algorithm, e.g., the BFGS quasi-Newton method, for the minimization (3) with initial approximation c_{ini}.

if $\hat{a}$ and $\hat{b}$ have to be displayed **then**

  Solve the least squares problem $\begin{bmatrix} a & b \end{bmatrix} = T(c)\begin{bmatrix} u & v \end{bmatrix}$ for u and v.

  Define $\hat{a} = u \star c$ and $\hat{b} = v \star c$, where $\star$ denotes discrete convolution.

end if

Output: The approximation $c \in \mathbb{R}^{d+1}$ found by the optimization algorithm upon convergence, the value of the cost function $f(c)$ at the optimal solution, and if computed $\hat{a}$ and $\hat{b}$.

Since

$$f(c) = \operatorname{dist}\big(a(\xi), \hat{a}(\xi)\big) + \operatorname{dist}\big(b(\xi), \hat{b}(\xi)\big)$$

the value of the cost function $f(c)$ shows the approximation errors in treating $c(\xi)$ as an approximate common divisor of $a(\xi)$ and $b(\xi)$. Optionally Algorithm 1 returns a "certificate" $\hat{a}$ and $\hat{b}$ for the claim that $c(\xi)$ is an approximate common divisor of $a(\xi)$ and $b(\xi)$ with approximation accuracy $f(c)$.

In order to complete Algorithm 1 we need to choose an initial approximation c_{ini}. This is discussed in Section III. Also the fact that the analytic expression for $f(c)$ involves the highly structured matrix $T(c)$ suggests that it (and its derivatives) can be evaluated efficiently. This is briefly discussed next.

Efficient cost function evaluation

The most expensive operation in the cost function evaluation is solving the least squares problem

$$\begin{bmatrix} a & b \end{bmatrix} = T(c)\begin{bmatrix} u & v \end{bmatrix}.$$

Since $T(c)$ is a lower triangular, banded, Toeplitz matrix, this operation can be done efficiently. One approach is to compute efficiently the QR factorization of $T(c)$, e.g., via the generalized Schur algorithm [KS95]. Another approach is to solve the normal system of equations

$$T^\top(c)\begin{bmatrix} a & b \end{bmatrix} = T^\top(c)T(c)\begin{bmatrix} u & v \end{bmatrix},$$

exploiting the fact that $T^\top(c)T(c)$ is banded and Toeplitz structured. The first approach is implemented in the function `MB02ID` from the SLICOT library [VSV$^+$04] and the second approach is used in [MVK04], [MVP05] in solving related structured total least squares problem.

Once the least squares problem is solved, the product $T(c)\begin{bmatrix} u & v \end{bmatrix}$ has to be computed. Note that this product computes the convolutions $\begin{bmatrix} c \star u & c \star v \end{bmatrix}$. It is well known that convolution can be performed efficiently by FFT. Exploiting the structure of $T(c)$ in solving the least squares problems and doing the convolving operations efficiently, we obtain cost function evaluation in $O(n)$ operations. In [MVK04], [MVP05] it is shown that the first derivative $f'(c)$ can be evaluated also in $O(n)$ operations, so assuming that $d \ll n$, the overall cost per iteration for Algorithm 1 is $O(n)$.

III. Structured low rank approximation and suboptimal solution by SVD

Suboptimal initial approximation can be computed by the singular value decomposition (SVD) of the Sylvester matrix

$$S(a,b) := \begin{bmatrix} a_0 & & & & b_0 & & & \\ a_1 & a_0 & & & b_1 & b_0 & & \\ \vdots & a_1 & \ddots & & \vdots & b_1 & \ddots & \\ a_n & \vdots & \ddots & a_0 & b_n & \vdots & \ddots & b_0 \\ & a_n & & a_1 & & b_n & & b_1 \\ & & \ddots & \vdots & & & \ddots & \vdots \\ & & & a_n & & & & b_n \end{bmatrix} \in \mathbb{R}^{(2n-d+1)\times(2n-2d+2)}.$$

In order to motivate the SVD method, first we show that problem (2) is a structured low rank approximation problem. Then ignoring the Sylvester structure constraint, a suboptimal solution is obtained from an unstructured low rank approximation, which computation is carried out by the SVD.

From (1) we see that $c(\xi)$ is a common divisor of $\hat{a}(\xi)$ and $\hat{b}(\xi)$ if and only if there are $u(\xi) \in \mathbb{R}[\xi]$ and $v(\xi) \in \mathbb{R}[\xi]$, such that

$$\hat{a}(\xi)v(\xi) = \hat{b}(\xi)u(\xi).$$

With $\text{degree}(c(\xi)) = d$, this polynomial equation is equivalent to the system of algebraic equations

$$S(\hat{a},\hat{b}) \begin{bmatrix} v \\ -u \end{bmatrix} = 0.$$

The matrix $S(\hat{a},\hat{b})$ is called a Sylvester matrix for the polynomials $\hat{a}(\xi)$ and $\hat{b}(\xi)$. The degree constraint for $c(\xi)$ is equivalent to $\text{degree}(u(\xi)) = n-d$, or equivalently $u_{n-d+1} \neq 0$. Since $u(\xi)$ is defined up to a scaling factor, we can impose the normalization $u_{n-d+1} = 1$. This shows that problem (2) is equivalent to

$$\min_{\substack{\hat{a},\hat{b}\in\mathbb{R}^{n+1} \\ u,v\in\mathbb{R}^{n-d+1} \\ u_{n-d+1}=1}} \left\| \begin{bmatrix} a & b \end{bmatrix} - \begin{bmatrix} \hat{a} & \hat{b} \end{bmatrix} \right\|_{\text{F}}^2 \quad \text{subject to} \quad S(\hat{a},\hat{b}) \begin{bmatrix} v \\ -u \end{bmatrix} = 0, \tag{5}$$

where $\|\cdot\|_{\text{F}}$ denotes the Frobenius norm.

The approximate common factor $c(\xi)$ is not explicitly computed in (5). Once the optimal $u(\xi)$ and $v(\xi)$ are known, however, $c(\xi)$ can be found from (1). (By construction these equations have unique solution). Alternatively, without using the auxiliary variables $\hat{a}$ and $\hat{b}$, $c(\xi)$ can be computed from the least squares problem

$$a(\xi) = u(\xi)c(\xi), \qquad b(\xi) = v(\xi)c(\xi),$$

or in linear algebra notation

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} T(u) \\ T(v) \end{bmatrix} c, \tag{6}$$

where $T(u)$ and $T(v)$ are defined in (4).

Problem (5) is a structured low rank approximation problem: it aims to find a Sylvester rank deficient matrix $S(\hat{a},\hat{b})$ as close as possible to a given matrix $S(a,b)$ with the same structure. If $a(\xi)$ and $b(\xi)$ have no common divisor of degree d, $S(a,b)$ is full rank so that an approximation is needed.

It is well known that the (unstructured) low rank approximation problem

$$\min_{M,w} \|S(a,b) - M\|_{\text{F}}^2 \quad \text{subject to} \quad Mw = 0, \quad \|w\| = 1 \tag{7}$$

has an analytic solution in terms of the SVD of $S(a,b)$ [EY36]. The vector $w \in \mathbb{R}^{2(n-d+1)}$ corresponding to the optimal solution of (7) is equal to the right singular vector of $S(a,b)$ corresponding to the smallest singular value. The vector $\text{col}(v,-u)$ composed of the coefficients of the approximate divisors $v(\xi)$ and $-u(\xi)$ is up to a scaling factor (that enforces the normalization constraint $u_{n-d+1} = 1$) equal to w. (The scaling is irrelevant for the computation of c_{ini} and can be skipped.) This gives Algorithm 2 as a method for computing a suboptimal initial approximation.

IV. Numerical examples

We verify the results obtained by Algorithm 1 on examples from [ZY04] and [KL98]. Up to the number of digits shown our results match the ones reported in the literature. In the implementation of Algorithm 1, we use the function `fminunc` from the Optimization Toolbox of MATLAB with cost function evaluations only. The function `fminunc` performs unconstrained nonlinear local optimization using a BFGS quasi-Newton method.

Algorithm 2 Suboptimal approximate common divisor computation.

Input: Vectors $a, b \in \mathbb{R}^{n+1}$ and an integer d.

1: Compute the right singular vector w of the Sylvester matrix $S(a,b)$, corresponding to the smallest singular value.
2: Let $\text{col}(v, -u) := w$, where $u, v \in \mathbb{R}^{n-d+1}$.
3: Solve the least squares problem (6).

Output: The solution c of the least squares problem.

Example 4.1 from [ZY04]

The given polynomials are

$$a(\xi) = (4 + 2\xi + \xi^2)(5 + 2\xi) + 0.05 + 0.03\xi + 0.04\xi^2$$
$$b(\xi) = (4 + 2\xi + \xi^2)(5 + \ \xi) + 0.04 + 0.02\xi + 0.01\xi^2$$

and an approximate common divisor $c(\xi)$ of degree $d = 2$ is sought. Algorithm 1 converges in 4 iteration steps with the following answer

$$c(\xi) = 3.9830 + 1.9998\xi + 1.0000\xi^2.$$

To this approximate common divisor correspond approximating polynomials

$$\hat{a}(\xi) = 20.0500 + 18.0332\xi + 9.0337\xi^2 + 2.0001\xi^3$$
$$\hat{b}(\xi) = 20.0392 + 14.0178\xi + 7.0176\xi^2 + 0.9933\xi^3$$

and the approximation error is

$$f(c) = \text{dist}\big(a(\xi), \hat{a}(\xi)\big) + \text{dist}\big(b(\xi), \hat{b}(\xi)\big) = 1.5831 \times 10^{-4}.$$

Example 4.2, case 1, from [ZY04] (originally given in [KL98])

The given polynomials are

$$a(\xi) = (1 - \xi)(5 - \xi) = 5 - 6\xi + \xi^2$$
$$b(\xi) = (1.1 - \xi)(5.2 - \xi) = 5.72 - 6.3\xi + \xi^2$$

and an approximate common divisor $c(\xi)$ of degree $d = 1$ (a common root) is sought. Algorithm 1 converges in 6 iteration steps with the following answer

$$c(\xi) = -5.0989 + 1.0000\xi.$$

The corresponding approximating polynomials are

$$\hat{a}(\xi) = 4.9994 - 6.0029\xi + 0.9850\xi^2$$
$$\hat{b}(\xi) = 5.7206 - 6.2971\xi + 1.0150\xi^2$$

and the approximation error is $f(c) = 4.6630 \times 10^{-4}$.

V. Discussion and conclusions

The proposed solution method is closely related to a method for solving structured total least squares (STLS) problems presented in [MVK04], [MVP05]. As shown in Section III, Problem 1 is a Sylvester structured low rank approximation problem. The previously published STLS algorithm, however, does not apply (directly) to problems with Sylvester structure. The difficulty is that the Sylvester matrix is a special Toeplitz matrix in which the upper–right and lower–left corners are filled with zeros (that should not be modified in the approximation), while the method of [MVK04], [MVP05] applies to full Toeplitz matrices.

An alternative method for solving STLS problems, called structured total least norm (STLN), has been modified for Sylvester structured matrices and applied to computation of approximate common divisor in [ZY04]. The STLN approach is rather different from the approach present here because it solves directly the original problem (2) and is not based on the elimination idea leading to the equivalent problem (3). In addition, the method of [ZY04] requires rank reduction by d while (for scalar polynomials) our method always needs rank reduction by 1. Finally we address the efficiency of the computations issue, which is not discussed in [ZY04].

A topic for future work is to extend the proposed algorithm to matrix valued polynomials. In this case one needs to treat block-Sylvester matrix and rank reduction by more than one. The corresponding (full) block-Toeplitz case has been solved in [MVP05].

Acknowledgments

Dr. Ivan Markovsky is a postdoctoral researcher and Dr. Sabine Van Huffel is a full professor at the Katholieke Universiteit Leuven, Belgium. We thank Paola Boito for pointing to us the approximate greatest common divisor problem and for useful comments on the manuscript.

Our research is supported by Research Council KUL: GOA-AMBioRICS, GOA-Mefisto 666, several PhD/postdoc & fellow grants; Flemish Government: FWO: PhD/postdoc grants, projects, G.0078.01 (structured matrices), G.0407.02 (support vector machines), G.0269.02 (magnetic resonance spectroscopic imaging), G.0270.02 (nonlinear Lp approximation), G.0360.05 (EEG signal processing), research communities (ICCoS, ANMMM); IWT: PhD Grants; Belgian Federal Science Policy Office IUAP P5/22 ('Dynamical Systems and Control: Computation, Identification and Modelling'); EU: PDT-COIL, BIOPATTERN, ETUMOUR; HPC-EUROPA (RII3-CT-2003-506079), with the support of the European Community — Research Infrastructure Action under the FP6 "Structuring the European Research Area" Program.

References

[EY36] G. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1:211–218, 1936.

[KL98] N. Karmarkar and Y. Lakshman. On approximate GCDs of univariate polynomials. In S. Watt and H. Stetter, editors, *Journal of Symbolic Computation*, volume 26, pages 653–666, 1998. Special issue on Symbolic Numeric Algebra for Polynomials.

[KS95] T. Kailath and A. Sayed. Displacement structure: theory and applications. *SIAM Review*, 37(3):297–386, 1995.

[MVK04] I. Markovsky, S. Van Huffel, and A. Kukush. On the computation of the structured total least squares estimator. *Numer. Linear. Algebra Appl.*, 11:591–608, 2004.

[MVP05] I. Markovsky, S. Van Huffel, and R. Pintelon. Block-Toeplitz/Hankel structured total least squares. *SIAM J. Matrix Anal. Appl.*, 26(4):1083–1099, 2005.

[VSV+04] S. Van Huffel, V. Sima, A. Varga, S. Hammarling, and F. Delebecque. High-performance numerical software for control. *IEEE Control Systems Magazine*, 24:60–76, 2004.

[ZY04] L. Zhi and Z. Yang. Computing approximate GCD of univariate polynomials by structure total least norm. In *MM Research Preprints*, number 24, pages 375–387. Academia Sinica, 2004.