Vrije Universiteit Brussel

![VUB Vrije Universiteit Brussel logo]

A gradient system approach for Hankel structured low-rank approximation

Fazzi, Antonio; Guglielmi, Nicola; Markovsky, Ivan

A gradient system approach for Hankel structured low-rank approximation

Antonio Fazzi[a,b,*], Nicola Guglielmi[a], Ivan Markovsky[b]

[a]*Gran Sasso Science Institute (GSSI), Viale F. Crispi 7, 67100 L'Aquila, Italy*
[b]*Vrije Universiteit Brussel (VUB), Department ELEC, Pleinlaan 2, 1050 Brussels, Belgium*

Abstract

Rank deficient Hankel matrices are at the core of several applications. However, in practice, the coefficients of these matrices are noisy due to e.g. measurements errors and computational errors, so generically the involved matrices are full rank. This motivates the problem of Hankel structured low-rank approximation. Structured low-rank approximation problems, in general, do not have a global and efficient solution technique. In this paper we propose a local optimization approach based on a two-levels iteration. Experimental results show that the proposed algorithm usually achieves good accuracy and shows a higher robustness with respect to the initial approximation, compared to alternative approaches.

Keywords: Hankel matrix, low-rank approximation, gradient system, structured matrix perturbation
2000 MSC: 15B99, 41A29, 65Y20, 68W25

1. Introduction

1.1. Preliminaries

A Hankel matrix $H \in \mathbb{R}^{m \times n}$ is a structured matrix where the entry on the i-th row and the j-th column depends only on the sum $i + j$. Hankel matrices can be associated in a natural way to vectors or time series. For a given $m \in \mathbb{N}$, let $H_m \in \mathbb{R}^{m \times n}$ ($m \leq n$) be the Hankel matrix built from the real numbers $p_1, p_2, \ldots, p_T$ (with $T = n + m - 1$) as follows

$$H_m(p) = \begin{pmatrix} p_1 & p_2 & \cdots & p_n \\ p_2 & p_3 & \cdots & p_{n+1} \\ \vdots & \vdots & & \vdots \\ p_m & p_{m+1} & \cdots & p_T \end{pmatrix}$$

---

*Corresponding author
Email addresses:* `antonio.fazzi@gssi.it` (Antonio Fazzi), `nicola.guglielmi@gssi.it` (Nicola Guglielmi), `imarkovs@vub.ac.be` (Ivan Markovsky)

We denote by $\mathcal{H}^{m \times n}$ the subspace of $m \times n$ Hankel matrices. Given a vector

$$p = (p_1, \ldots, p_T)^\top,$$

we define the Hankel matrix

$$H = H_m(p) \in \mathcal{H}^{m \times (T-m+1)}.$$

Conversely, given a Hankel matrix

$$H \in \mathcal{H}^{m \times (T-m+1)},$$

we define the vector

$$\mathbf{vect}(H) = p = (p_1, \ldots, p_T)^\top.$$

Hankel structured matrices arise in applications in different areas of sciences, such as control theory, approximation and interpolation problems, polynomials computations (see for example [1, 2]). Most applications involve the computation of the distance between a given Hankel matrix and a singular one (preserving the same structure). In the following we focus our attention on system identification of linear time-invariant models and polygons from moments reconstruction. The structure preserving property is of interest both theoretically (because of the applications involving singular Hankel matrices) [3, 4, 5] and in practice (due to the presence of noise in real life problems) [6].

In this paper, we analyze the problem of computing the structured distance to singularity in the case of (scalar) Hankel matrices. Because of the association previously discussed between a Hankel matrix and the vector of its coefficients, we consider the following formulation of the problem

$$\min_{\substack{\tilde{p} \in \mathbb{R}^T \\ H_m(\tilde{p}) \text{ rank deficient}}} \|p - \tilde{p}\|_w \tag{1}$$

where the norm

$$\|p\|_w = \left(\sum_{i=1}^T w_i |p_i|^2 \right)^{1/2}.$$

For example, the weights can be chosen such that $\|p\|_w = \|H_m(p)\|_F$.

1.2. Main contribution of the paper

The approximation of a structured matrix by a matrix of lower rank which preserves the same structure is a classical problem in numerical linear algebra [5, 6]. The constraint on the structure of the computed solution makes the problem harder in comparison with unstructured low-rank approximation. Currently, there is no analytical solution neither standard solution strategies. Looking at numerical schemes, global optimization approaches are usually computationally expensive, hence the most common methods for solving (1) are local optimization [7] or convex relaxation methods [8].

The algorithm we propose here is based on the ideas presented in [9], where the approximate common factor problem is restated as a structured low-rank approximation problem for Sylvester matrices (block Toeplitz matrices). On the other hand we take inspiration from the ideas presented in [10], where the authors study the behavior of the eigenvalues of Toeplitz matrices under finite (structure preserving) perturbations. The results in [9, 10] motivate the work presented in this paper because of the similarities between Toeplitz and Hankel matrices.

We propose a double iteration method for Hankel structured low-rank approximation. According to the formulation in (1), we start from a data vector p (whose associated Hankel matrix is full rank) and we aim at modifying the matrix $H_m(p)$ in order to make it singular, by adding a perturbation of the form $\epsilon\delta$ (where δ is a norm 1 vector, while ϵ is a scalar measuring the norm of the perturbation). The two values of ϵ and δ are updated on two different independent levels:

- at the inner level we fix the norm of the perturbation ϵ and we look for the vector δ which minimizes the smallest singular value of $H_m(p + \epsilon\delta)$. This is done by an ODE for δ which is given by the gradient system for the smallest singular value of the matrix;

- at the outer level we need to move the value of ϵ (increase the norm of the perturbation) till the smallest singular value is zero.

Outline. The goal of the paper is to provide a numerical algorithm for the solution of (1) and to test it on the applications of system identification and polygons from moments reconstruction in order to observe its performances. The application to identification of linear time-invariant dynamical models is presented in Section 2.1 and the polygons from moments reconstruction is presented in Section 2.2. In Section 3 we present the algorithm, describing in details the two levels iteration, and showing the monotonicity of the smallest singular value at the inner level. We perform then some numerical tests in Section 4 doing a comparison (whenever it is possible) with the SLRA package [11]. A summary and possible directions for future work are listed in Section 5.

2. Applications

We describe in this section some applications involving low-rank approximation of Hankel matrices.

2.1. System Identification

Hankel matrices play a central role in system theory and identification. An important result states that the rank deficiency of a Hankel matrix built from some data $p = (p(1), p(2), \ldots, p(T)) \in \mathbb{R}^T$ is equivalent with the fact that such data p is an impulse response of a linear time-invariant dynamical system of order equal to the rank of the matrix. In particular it is known that a linear

3

time-invariant dynamical system of order m can be defined through a difference equation [12, Theorem 7.2] of the form

$$R_0 p(t) + R_1 p(t+1) + \cdots + R_m p(t+m) = 0 \quad \text{for all } t. \tag{2}$$

where $R = (R_0, R_1, \ldots, R_m) \neq 0$ is a vector of real numbers. In the problem of system identification we are given a finite trajectory of the system $p = (p(1), p(2), \ldots, p(T)) \in \mathbb{R}^T$ and the order m and we aim to find the generating model (defined by the vector of parameters R). If we write equation (2) in matrix form we have

$$(R_0, R_1, \ldots, R_m) H_{m+1}(p) = 0, \tag{3}$$

which means the Hankel matrix $H_{m+1}(p)$ is rank deficient.

We deduce that a necessary and sufficient condition for the time series p to be generated by a linear time-invariant dynamical model is $H_{m+1}(p)$ to be rank deficient. However in practical applications p can be corrupted by noise, so we deal with a full rank Hankel matrix, and we are interested in computing the closest rank deficient Hankel matrix. The vector R is then computed as the kernel of the approximating matrix.

Model (3) is used in several contexts, ranging from biomedical signal processing, vibrational analysis of dynamical structure, industrial process system identification, stochastic identification and telecommunications [13, 14, 15, 16, 17].

2.2. Polygons from moments reconstruction

The *polygons from moments problem* consists in reconstructing a (binary) simply connected and nondegenerate polygon from some complex quantities called *moments*. The mathematical framework of the problem and its derivation can be found, e.g., in [18, 19]; we briefly summarize here how to restate the problem as a Hankel low-rank approximation problem.

A polygon is reconstructed from its n vertices $z_1, \ldots, z_n$, and each z_i is a complex number. For a given integer number $N > 2n$, the so called *complex moments* are defined as [18]

$$\tau_k = \sum_{j=1}^{n} a_j z_j^k \quad for \ k = 0, \ldots N \tag{4}$$

where

$$a_j = \frac{2A_j}{(z_j - z_{j-1})(z_j - z_{j+1})}, \quad A_j = \frac{i}{4} \det \begin{pmatrix} z_{j-1} & \bar{z}_{j-1} & 1 \\ z_j & \bar{z}_j & 1 \\ z_{j+1} & \bar{z}_{j+1} & 1 \end{pmatrix}$$

and we assume that the set of vertices is cyclic (so $z_0 = z_n, z_{n+1} = z_1$ and so on). Equation (4) can be written in matrix form as

$$\begin{pmatrix} \tau_0 \\ \tau_1 \\ \vdots \\ \tau_N \end{pmatrix} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ z_1 & z_2 & \cdots & z_n \\ \vdots & \vdots & \ddots & \vdots \\ z_1^N & z_2^N & \cdots & z_n^N \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} \iff \tau = Za \tag{5}$$

Using Prony's method [20] it is possible to show that the set of vertices can be computed from the vector of complex moments τ in (5). Define the polynomial

$$P(z) = \prod_{j=1}^{n}(z - z_j) = z^n + \sum_{j=1}^{n} p_j z^{n-j}$$

and the vector $p = (p_n, p_{n-1}, \ldots, p_1)$. In this way, the problem is equivalent to the computation of the vector p. Premultiplying equation (5) by the following Toeplitz matrix

$$K_{N+1} = \begin{pmatrix} p_n & p_{n-1} & \cdots & p_1 & 1 & & \\ & \ddots & \ddots & & \ddots & \ddots & \\ & & p_n & p_{n-1} & \cdots & p_1 & 1 \end{pmatrix} :$$

we get

$$K_{N+1}\tau = K_{N+1}Za = 0,$$

where the last equality comes from the definition of $P(z)$. The identity $K_{N+1}\tau = 0$ can be then written as

$$\begin{pmatrix} \tau_0 & \tau_1 & \cdots & \tau_n \\ \tau_1 & \tau_2 & \cdots & \tau_{n+1} \\ \vdots & \vdots & & \vdots \\ \tau_{N-n} & \tau_{N-n+1} & \cdots & \tau_N \end{pmatrix} \begin{pmatrix} p \\ 1 \end{pmatrix} = 0. \tag{6}$$

Equation (6) shows that the Hankel matrix of the moments is rank deficient. In realistic applications the measurements of the complex moments is affected by noise, so that we expect the Hankel matrix $H_{N-n}(\tau)$ to be full rank. By computing the closest rank deficient Hankel matrix and its kernel, we can approximately reconstruct the set of vertices $z_1, \ldots, z_n$.

Remark 1. In the application to polygons from moments reconstruction, the data are complex-valued.

3. The algorithm

In this section, we propose an algorithm for the numerical solution of scalar Hankel structured low-rank approximation problem (1).

The optimization problem we aim at solving is written as follows.

Problem 1. *Given a vector* $p \in \mathbb{R}^T$, *the Hankel matrix* $H_m(p) \in \mathbb{R}^{m \times n}, m \leq n$ *and the norm* $\|\cdot\|_w$, *compute*

$$\min_{\tilde{p} \in \mathbb{R}^T} \|p - \tilde{p}\|_w \quad subject\ to \quad rank(H_m(\tilde{p})) < m. \tag{7}$$

The objective function we minimize is the Frobenius norm of the difference of the initial Hankel matrix $H_m(p)$ and the approximating rank deficient Hankel matrix $H_m(\tilde{p})$. We illustrate later how to generalize the proposed approach to different weighted norms.

3.1. A double iteration algorithm

Let $\sigma_{\min}(A)$ be the smallest singular value of the matrix A. It is well known that the rank of a matrix can be computed looking at its singular values. For a matrix H, the value $\sigma_{\min}(H)$ measures its (unstructured) distance to singularity. Our aim here is to iteratively decrease the value of $\sigma_{\min}(H)$ but preserving the Hankel structure. This is done by modifying (in a structured way) the starting Hankel matrix H in a way that makes the functional $\sigma_{\min}(H)$ decreasing till it reaches a fixed (small) tolerance. In such a way we are able to achieve (to the given tolerance) the rank constraint and preserve the structure at the same time.

In particular, starting from a (full rank) Hankel matrix $H_m(p)$, the perturbed (Hankel) matrix has the form $H_m(p + \epsilon\delta)$, with δ a norm 1 vector and $\epsilon \in \mathbb{R}$ a number which measure the norm of the perturbation on the starting vector p; the values of the two parameters δ and ϵ are updated independently on two different levels:

- for a fixed value of ϵ we compute an optimal perturbation δ (by looking at the stationary points of a suitable gradient system of ordinary differential equations);

- once we have computed the perturbation δ, which we maintain fixed, we update the value of ϵ in order to get closer to an admissible solution (a vector associated with a rank deficient Hankel matrix).

3.2. How to decrease the smallest singular value

In this section we consider ϵ as fixed and we want to compute a perturbation vector δ (of norm 1) in such a way that $\sigma_{\min}(H_m(p + \epsilon\delta))$ is minimum among all the possible norm-1 perturbations. In other words we are looking for the norm 1 vector δ which makes $\sigma_{\min}(H_m(p + \epsilon\delta))$ decreasing along the steepest descent direction. We make use of the following standard result about perturbation of eigenvalues for positive semidefinite matrices [21], which is adapted to the case of singular values

Lemma 1. *Let $D(t)$ be a differentiable matrix-valued function for t in a neighborhood of $t_0 = 0$. Let $D(t) = U(t)\Sigma(t)V(t)^\top$ be a smooth (with respect to t) singular value decomposition of the matrix $D(t)$ and $\sigma(t)$ be a certain singular value of $D(t)$ converging to a simple singular value σ_0 of $D(0)$. If u_0, v_0 are the associated left and right singular vectors, respectively, the function $\sigma(t)$ is differentiable near $t = 0$ with*

$$\dot{\sigma} = u_0^\top \dot{D} v_0.$$

Lemma 1 is applied to the singular values of the matrix $H_m(p + \epsilon\delta)$. Remembering that both p and ϵ are constant, the derivative of $\sigma_{\min}(H_m(\tilde{p}))$ is given by

$$\dot{\sigma} = u^\top H_m(\epsilon\dot{\delta})v, \tag{8}$$

where u, v are the left and right singular vectors associated with $\sigma_{\min}(H_m(\tilde{p}))$.

For a pair of real matrices A, B we let

$$\langle A, B \rangle = \text{trace}(A^\top B) = \sum_{i,j=1}^{n} A_{ij} B_{ij}$$

the Frobenius inner product, inducing the norm $\|A\| = \langle A, A \rangle^{1/2}$. If instead A and B are vectors, $\langle A, B \rangle$ denotes the standard inner product.

The optimal descent direction for the singular value of interest is obtained by minimizing the following function

$$u^\top H_m(\dot{\delta})v = \langle uv^\top, H_m(\dot{\delta}) \rangle = \langle P_{\mathcal{H}}(uv^\top), H_m(\dot{\delta}) \rangle \tag{9}$$

where $P_{\mathcal{H}}(B)$ denotes the orthogonal projection of the matrix B onto the subspace of Hankel matrices $\mathcal{H} = \mathcal{H}^{m \times n}$.

Since we may work directly on the vectors associated to the Hankel matrices, the objective function in (9) can be written as

$$\langle \mathbf{vect}(P_{\mathcal{H}}(uv^\top)), \dot{\delta} \rangle \tag{10}$$

The explicit formula for the operator $P_{\mathcal{H}}(\cdot)$ is given in the following Lemma 2

Lemma 2. *Let $\mathcal{H}^{m \times n}$ be the linear manifold of real $m \times n$ Hankel matrices, and let $B \in \mathbb{R}^{m \times n}$ an arbitrary matrix. The orthogonal projection (with respect to the Frobenius inner product) of B onto $\mathcal{H}$ is given by*

$$P_{\mathcal{H}}(B) = H_m(q)$$

where

$$q_i = \begin{cases} \dfrac{1}{i} \displaystyle\sum_{j=1}^{i} B_{j,i-j+1} & for \quad i = 1, \ldots, m \\ \dfrac{1}{m} \displaystyle\sum_{j=1}^{m} B_{j,i-j+1} & for \quad i = m+1, \ldots, n-1 \end{cases}$$

$$q_{m+n-i} = \dfrac{1}{i} \sum_{j=1}^{i} B_{j,m+n-i-j+1} \quad for \quad i = 1, \ldots, m.$$

Proof. The solution is given by the orthogonal projection with respect to the Frobenius inner product which is simply obtained by taking the averages along the anti-diagonals of the matrix B. $\qquad\square$

A further important result follows.

Lemma 3. *Let $H(p)$ be a Hankel matrix. If $\sigma > 0$ is a simple singular value of $H(p)$ and u and v are the corresponding left and right singular vectors, then*

$$P_{\mathcal{H}}(uv^\top) \neq 0.$$

Proof. Assume, by contradiction, $P_{\mathcal{H}}(uv^\top) = 0$. We have

$$0 = \langle P_{\mathcal{H}}(uv^\top), H(p) \rangle = \langle uv^\top, H(p) \rangle = u^\top H(p) v = \sigma.$$

The proof is completed since $\sigma > 0$ by assumption, so by following the chain we get the contradiction $0 > 0$. $\qquad\square$

Consider the singular value $\sigma_{\min}(H_m(\tilde{p}))$, and let u, v be the corresponding left and right singular vectors, respectively. The steepest descent direction for the considered singular value is given by the solution of the following optimization problem:

$$\dot{\delta}^* = \underset{\substack{\dot{\delta} \in \mathbb{R}^T \\ \|\dot{\delta}\|_2 = 1}}{\arg\min}\, u^\top H_m(\dot{\delta}) v \qquad \text{subject to } \langle \delta, \dot{\delta} \rangle = 0 \tag{11}$$

where the constraint on the norm guarantees the uniqueness of the solution (which represents a direction) while the last constraint guarantees the norm conservation of δ, which we have assumed. We give the solution of the problem (11) in the following Lemma

Lemma 4. *Let u, v be the left and right singular vectors of $H_m(\tilde{p}) = H_m(p + \epsilon\delta)$ with $\delta \in \mathbb{R}^T$ of unit norm. The solution of the optimization problem (11) is given by*

$$\mu\dot{\delta}^* = \boldsymbol{vect}(-P_{\mathcal{H}}(uv^\top)) + \langle \delta, \boldsymbol{vect}(P_{\mathcal{H}}(uv^\top)) \rangle \delta \tag{12}$$

where μ is the norm of the vector in the right hand side.

Proof. In the Frobenius metric, the minimizing direction for function (9) is reached for $H_m(\dot{\delta}) = -P_{\mathcal{H}}(uv^\top)$, consequently $\boldsymbol{vect}(-P_{\mathcal{H}}(uv^\top))$ is the solution of the unconstrained version of (11).

For an arbitrary vector y (of suitable dimension), the projection onto the space orthogonal to δ is given by

$$P(y) = y - \langle \delta, y \rangle \delta.$$

The claim follows by choosing $y = \boldsymbol{vect}(-P_{\mathcal{H}}(uv^\top))$ and normalizing in order to satisfy the constraint on the norm. $\qquad\square$

Lemma 4 is a key result for the proposed method. Since (12) gives the unit norm steepest descent direction δ for the smallest singular value, we may omit the scaling factor μ by normalizing $\dot{\delta}$ (to have norm 1) and consider the gradient system for σ

$$\dot{\delta} = -\boldsymbol{vect}(P_{\mathcal{H}}(uv^\top)) + \langle \delta, \boldsymbol{vect}(P_{\mathcal{H}}(uv^\top)) \rangle \delta$$
$$\delta(0) = \delta_0 \quad \text{with} \quad \|\delta_0\| = 1 \tag{13}$$

that (locally) minimizes the smallest singular value σ of $H_m(p + \epsilon\delta)$ on the set $\{\delta : \|\delta\| = 1\}$.

Considering the initial value $\delta(0)$ of unit norm then we can see that

$$\frac{d}{dt}\|\delta(t)\|^2 = 2 \langle \dot{\delta}(t), \delta(t) \rangle = 0$$

where we have replaced $\dot{\delta}$ by the right-hand side of the ODE (13), which implies norm conservation (this relation holds true since we impose $\delta(t)$ to have norm 1).

By construction (13) is the gradient system for the objective functional σ (the smallest singular value of the perturbed Hankel matrix) under the constraint that the perturbation has fixed norm ε (i.e. δ has unit norm).

Thus it follows directly that the objective functional is monotonically decreasing along the solution of (13).

Theorem 1. *Let $\delta(t) \in \mathbb{R}^T$ be a solution of (13). If $\sigma(t)$ is the smallest singular value of the Hankel matrix $H_m(\tilde{p}) = H_m(p + \epsilon\delta)$, then*

$$\dot{\sigma}(t) \le 0.$$

Proof. To prove the result we recall that $\dot{\sigma} = u^\top H_m(\dot{\delta})v$ (omitting the constant term) and we compute its point of minimum by using the expression in (10). We replace now the expression of $\dot{\delta}$ using equation (13), and observe that

$$\langle \mathbf{vect}(P_{\mathcal{H}}(uv^\top)), \mathbf{vect}(P_{\mathcal{H}}(uv^\top)) \rangle = \|\mathbf{vect}(P_{\mathcal{H}}(uv^\top))\|^2.$$
$$\langle \mathbf{vect}(P_{\mathcal{H}}(uv^\top)), \delta \rangle \langle \mathbf{vect}(P_{\mathcal{H}}(uv^\top)), \delta \rangle = \langle \mathbf{vect}(P_{\mathcal{H}}(uv^\top)), \delta \rangle^2.$$

By adding the two terms with the correct signs, we get

$$\langle \mathbf{vect}(P_{\mathcal{H}}(uv^\top)), \dot{\delta} \rangle = -\|\mathbf{vect}(P_{\mathcal{H}}(uv^\top))\|^2 + \langle \mathbf{vect}(P_{\mathcal{H}}(uv^\top)), \delta \rangle^2 \le 0$$

using Cauchy-Schwarz inequality and recalling that δ has norm 1. $\qquad \square$

Remark 2. Tha assumption on δ to have norm 1 plays a key role. However we remind that the whole perturbation is given by $\epsilon\delta$ whose norm is ϵ (fixed at the inner level). The scaling factor for the direction vector δ is arbitrary, so we choosed it to be norm 1 without loss of generality.

The previous results allow to state that the points of (local) minimum for the objective functional ($\delta \to \sigma_{\min}(H_m(p+\epsilon\delta))$) are the stationary points of the gradient system (13) (δ such that $\dot{\delta} = 0$). Moreover Lemma 3 guarantees that such points of minimum correspond only to the zeros of the derivative and vice versa. Consequently, the goal of the inner iteration is to integrate equation (13) until stationary points.

Equation (13) is a differential equation for the vector δ. In order to compute its stationary points we choose to adopt an Explicit Euler scheme, because the function evaluation (a singular value decomposition) at each step is quite expensive. Because of the significant computational cost we preferred to avoid both higher order explicit schemes and implicit method (the first require more

than one svd factorization at each step, the latter require an extra solution of a system). The numerical scheme for the integration of the ODE (13) is summarized in Algorithm 1.

Algorithm 1: Numerical solution of the ODE (13)

Data: δ, σ, u, v, h (step Euler method) and γ (step size reduction), ϵ.
Result: δ^*, σ^*
begin

1   Compute $\dot{\delta} = \mathbf{vect}(-P_{\mathcal{H}}(uv^\top)) + \langle \delta, \mathbf{vect}(P_{\mathcal{H}}(uv^\top)) \rangle \delta$

2   Euler step $\rightarrow \delta^1 = \delta + h\dot{\delta}$

3   Normalize δ^1 dividing it by its norm

4   Compute the singular value $\sigma^1 = \sigma_{\min}(H_m(p + \epsilon \delta^1))$

5   Compute the singular vectors u^1 and v^1 associated with σ^1

6   **if** $\sigma^1 > \sigma$ **then**
        reject the result and reduce the step h by a factor γ
        repeat from line 2
    else
        accept the result; set $\sigma^* = \sigma^1$, $\delta^* = \delta^1$

7   **if** $\sigma^* - \sigma < tol$ **or** $\sigma^* \leq tol$ **then**
           $\llcorner$ **return**

The singular triplet u, v, σ at each step is computed through the Matlab function *svds* in order to store only the needed vectors and not all the factorization of the Hankel matrix. However the numerical results got from *svds* and *svd* are not exactly the same in floating point arithmetic, so the use of *svds* can lead to a less accurate solution because of algorithmic errors but the code is expected to run faster.

3.3. How to compute ϵ?

The goal is that of modifying ϵ so that the branch of smallest singular values $\sigma(\epsilon)$ of $H_m(p + \epsilon \delta(\epsilon))$ (where $\delta(\epsilon)$ stands for the minimizer at a given ϵ) smoothly reaches the minimum value zero.

Free and constrained dynamics

Once we get the optimal perturbation $\delta(\epsilon)$ (for a given value of ϵ) we need to iteratively update the value of ϵ till we reach the sought solution (a vector $\tilde{p} = p + \epsilon^* \delta(\epsilon^*)$ associated with a rank deficient Hankel matrix $H(\tilde{p})$). A possible way to proceed would be to increase the value of ϵ by a constant increment at each step; however, if we proceed in this way on some randomly chosen test problem, we may observe something unexpected, that is an apparent loss of monotonicity of the objective functional $\epsilon \rightarrow \sigma_{\min}(H_m(p + \epsilon \delta))$.

The observed behavior may be explained in terms of the choice of the initial datum for the ODE (13). Consider the value $\epsilon_1 \in \mathbb{R}$, and integrate the equation (13) till converging to an optimal perturbation δ_1 (of norm 1) which corresponds to a (locally) minimal singular value $\sigma(\epsilon_1) = \sigma_{\min}(H_m(p + \epsilon_1 \delta_1(\epsilon_1))$.

Passing to the next iteration we update $\epsilon_2 = \epsilon_1 + \Delta\epsilon$ where $\Delta\epsilon$ is a constant term, and a natural choice for the initial datum for the ODE (13) is $p + \epsilon_2 \delta_1(\epsilon_1)$. In general we may have $\sigma_{\min}(H_m(p + \epsilon_2 \delta_1(\epsilon_1))) > \sigma(\epsilon_1)$, and this could persist also once we get the optimal perturbation $\delta_2(\epsilon_2)$ corresponding to ϵ_2 (i.e., it can happen that $\sigma(\epsilon_2) > \sigma(\epsilon_1)$, which means we are not following the smallest singular value in a smooth way).

To deal with this issue we need to introduce an intermediate different ODE, which makes the branch $\sigma(\epsilon)$ continuous. Since this is obtained by omitting the norm constraint on δ at the inner level we call it *free dynamics*.

Free dynamics for σ. The basic idea of this computational strategy is to start each iteration exactly from the endpoint of the previous one: we iteratively alternate two different dynamics in order to preserve the monotonicity of the function $\sigma_{\min}(H_m(p + \epsilon\delta))$ with respect to ϵ (the monotonicity with respect to δ is a property of the gradient system).

Previously we considered (13), which is an ODE whose solution is a vector $\delta(t)$ of unit norm, so that the norm of the perturbation on the starting data vector p is given by the value of ϵ. What happens if we remove the constraints on the computed solution? The unconstrained optimization problem is still a gradient system for the smallest singular value σ, although not preserving $\|\delta(t)\| = 1$, and the modified ODE is given by

$$\dot{\delta} = - \mathbf{vect}(P_{\mathcal{H}}(uv^\top)) \tag{14}$$

Assume we have solved (13) for $\epsilon = \epsilon_1$ and in the outer iteration a value $\epsilon = \epsilon_2 > \epsilon_1$ is proposed. The initial datum for δ, that we choose, is given by $\delta(\epsilon_1)$, which has unit norm and has been computed by solving (13). At some time $\bar{t}$ the solution of (14)—which is expected to increase in norm in order to decrease the smallest singular value—has norm

$$\delta(\bar{t}) = \frac{\epsilon_2}{\epsilon_1}.$$

At this point we consider again (13) with $\epsilon = \epsilon_2$ and initial datum $\delta(0) = \delta(\bar{t})/\|\delta(\bar{t})\|$. In this way we have a global continuity with respect to t and ϵ of δ and consequently of σ, and avoid jumps on different branches which may determine a loss of the monotonicity property that we expect. If we assume that ε^* is the minimum value such that σ nihilates, i.e., $\sigma(\varepsilon^*) = 0$, we have that $\sigma(\epsilon) > 0$ as far as $\varepsilon < \varepsilon^*$. Then, in order to decrease σ towards 0, the ODE (14) (which is not norm-preserving) has necessarily to increase the norm of the perturbation by its gradient system structure. The monotonicity of this integration comes from the choice of the integration scheme (look at Algorithm 1 without the normalization step 3) since we only accept the computed solutions corresponding to decreasing values for the function to be minimized σ. We remark that this fact holds true till the norm of the perturbation is less than ϵ^* in order to avoid jumps on different branches of the same function.

Remark 3. The coupling of (13) and (14) described in this section imply monotonicity of $\sigma(\varepsilon\delta(\varepsilon))$ with respect to ε when when $\varepsilon < \varepsilon^*$. This is because of the gradient system structure of both systems of ODEs and the choice of the integration schemes.

The numerical scheme for the integration of (14) is very similar to the one presented in Algorithm 1, (with the obvious changes). The difference with respect to the previous case is that, by removing the constraints on the computed solution during the integration of the equation, the norm of the perturbation δ increases, so that it is natural to link the stopping criterion with the norm of such a perturbation.

In this way we have a globally decreasing trajectory for $\sigma_{\min}(H_m(\tilde{p}))$ (with respect to the outer iteration), as we show in Algorithm 2 (where the uppercase f denotes that the corresponding quantities are associated with the free dynamic).

Algorithm 2: Computation of ϵ^*

Data: p, *tol*, Δ

Result: ϵ_o approximation of ϵ^*

begin

1   Set $\epsilon = 10^{-2}$ % starting value

2   Integrate the equation (13)

     store $\delta, \sigma = \sigma_{\min}(H_m(p + \epsilon\delta))$

3   **while** $\sigma > tol$ **do**

4       $\epsilon_1 = \epsilon + \Delta$

5   integrate the equation (14) with initial value $p + \epsilon/\epsilon_1\delta$

        and stop when $\epsilon/\epsilon_1\|\delta^f\|_2 \geq 1$

        store $\delta^f, \sigma^f = \sigma_{\min}(H_m(p + \epsilon/\epsilon_1\delta^f))$

        set $\epsilon_1 = \epsilon\|\delta^f\|_2$

6   integrate equation (13) with initial value $p + \epsilon_1\delta^f$

        store $\delta, \sigma = \sigma_{\min}(H_m(p + \epsilon_1\delta))$

        set $\epsilon = \epsilon_1$

7    $\epsilon_o = \epsilon$

If we apply Algorithm 2 to some randomly chosen test problem, we get the result of Figure 1, where we can observe both the monotonicity of the objective functional and the succession between free and constrained dynamics.

Remark 4. In the previous sections we considered real numbers just to simplify the notation. All the arguments still hold true also in the complex case (as needed in the solution of the problem described in Section 2.2).

3.4. Different weights and missing coefficients

A key result of the proposed approach is Lemma 1, hence the algorithm is based on the solution of matrix nearness problems in the Frobenius metric. This motivates the choice of the Frobenius norm of the Hankel matrix in (7). The algorithm is then restated on the vector p by choosing $p = \mathbf{vect}(H_m(p))$.
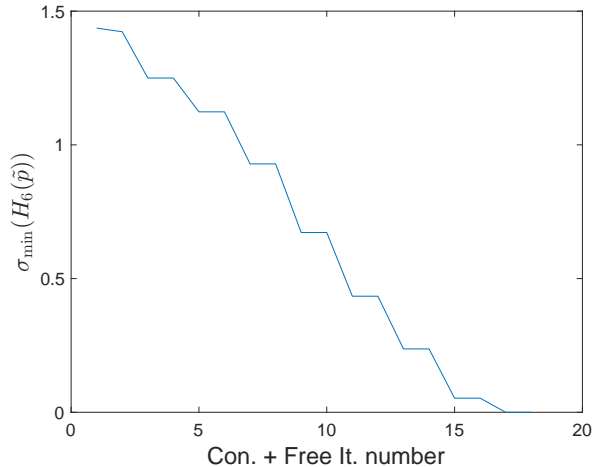
Figure 1: Plot of the functional $\sigma_{\min}(H_3(\hat{p}))$ as function of the iteration number.

However it is possible to add some different weights on the coefficients of p by preserving all the properties previously shown. This is done by appropriately changing the projection operator $P_{\mathcal{H}}$. In particular, once we average along the anti diagonals to compute the coefficients of the projected Hankel matrix, we multiply each entry of the output vector by the corresponding weight. Using Matlab notation for the entry-wise product and denoting as δ_w the (normalized) vector whose entries are weighted, equation (13) would be replaced by

$$\dot{\delta}_w = -w. * \mathbf{vect}(P_{\mathcal{H}}(uv^\top)) + \langle \delta_w, w. * \mathbf{vect}(P_{\mathcal{H}}(uv^\top)) \rangle \delta_w.$$

Similarly we can also fix some entries of the vector p: It is sufficient to set as 0 the corresponding weights so that the algorithm leaves the coefficients untouched during the iterations, and perturb only a subset of the entries of p.

In the case some coefficients are missing, it is enough to choose some initial values for them (e.g. by averaging their neighbors) and run the algorithm.

4. Numerical results

In this section we run some numerical examples. We consider both problems described in Section 2. In the problem of system identification we make a comparison with the function *slra* from the SLRA Toolbox [11] in order to test the performances of the proposed algorithm (we cannot do the same with the problem of polygons reconstruction since the available function *slra* does not work with complex numbers).

4.1. Identification of linear time-invariant models

The problem is the one described in Section 2.1. The goal of the experiments is to recover a time series from its noisy entries. First, we construct a time series,

13

which is a response of a linear time-invariant model. The simulation setup is as follows:

1. a random linear time-invariant system of order n is selected (`drss` function in MATLAB);

2. the "true data" p_0 is the response of the system selected on step 1 to a random initial condition;

3. the "noisy data" p is

$$p = p_0 + \tau r \|p_0\|_2 / \|r\|_2, \tag{15}$$

where r is a zero mean white Gaussian random vector with unit variance and $\tau > 0$ is parameter (called the noise level).

In the following, we compare the results computed by the proposed approach with the ones computed by the function *slra* from [11]. However the solution computed by the proposed method is not exactly rank deficient, so that a further comparison is done with a refinement obtained by *combining* the two functions: we initialize the function *slra* with the solution found by the ODE based method. In this way, we obtain a rank deficient Hankel matrix.

Test on the accuracy of the computed solution. In this first example we test the accuracy on the computed solution. On a single run we expect that different local optimization methods compute different solutions, hence we look at the average behavior on several runs of the two algorithms.

Figure 2 shows the results from a model of order $n = 5$. We observe how
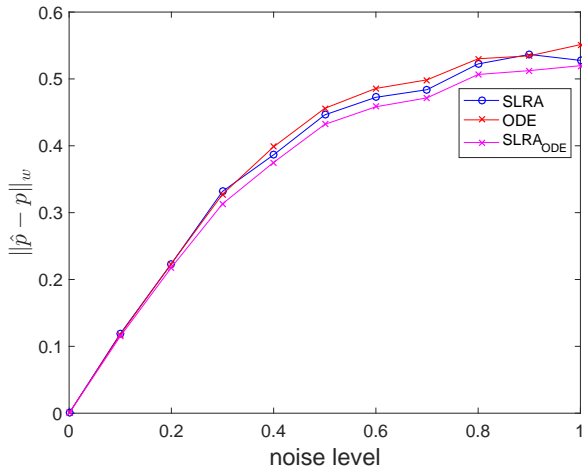
Figure 2: Identification of the output of a linear time-invariant model of order 5 as function of the noise level (average over fifty perturbations).

14

the distances associated with all the computed solutions are very close; however the function *slra* initialized with the solution of the proposed approach is able to achieve smaller values of distance. The computational time of the function *slra* is smaller than the one of the proposed ODE based method.

The next experiment shows an interesting property of the proposed local optimization approach.

Dependence on the initial estimate. We analyzed in the previous experiment which is the distance computed by the different algorithms; however the solutions found by local optimization approaches usually strongly depend on the initial estimate. An interesting advantage of the proposed algorithm appears to be the robustness with respect to the initial approximation, i.e., the computed solutions are (almost) independent from the initial estimate. In practice, this means that choosing two different initial directions, the proposed ODE-based algorithm finds (almost) the same solutions.

The standard initialization for the two algorithms (the ones used by default) are the following:

- ODE: the starting optimal perturbation δ is chosen as the steepest descent direction for the smallest singular value of the Hankel matrix $H_m(p)$, i.e., $\mathbf{vect}(-P_{\mathcal{H}}(uv^\top))$ (divided by its norm), where u, v are the left and right singular vector associated with $\sigma_{\min}(H_m(p))$;

- SLRA: the initial approximation is the unstructured low-rank approximation of the Hankel matrix $H_m(p)$.

In the next experiment (Figure 3) we perturb (using a random perturbation coming from a normal distribution with zero mean and standard deviation 0.5) the standard initial estimates of the two algorithms in order to observe how the final computed solutions change. (Again, we show the average behavior over several runs.) It happens that all the plots associated with the ODE based algorithm are very close (they are all almost overlapping) so we cannot distinguish them in the figure. On the other hand, about the function *slra*, the different plots associated with different initial estimate spread off more, especially when the level of noise increases (hence the optimization problem becomes more difficult). We can also observe that by changing the initial estimate the solution computed by *slra* is less accurate (comparing with the results in Figure 2). But the initialization of *slra* with the solution computed by the ODE method still leads to the best results; in this case we can not distinguish the plots hence the independence from the initial condition is more clear (this is because the outputs of the ODE method are all very close).

Remark 5. Because of normalization issues we are not able to establish a perfect link between the initial estimates for the two different algorithms. However a perturbation on the optimal initial estimate (the one used by the algorithms by default) it should be enough to show the robustness of the proposed algorithm with respect to the initial condition.
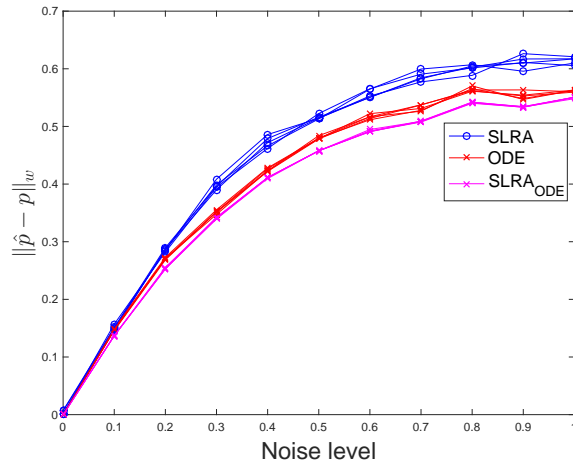
Figure 3: Identification of the output of system of order 5: error computed as function of the noise level (average over fifty perturbations). Each plot corresponds to a different initial estimate.

4.2. Reconstructing polygons from moments

We show here some numerical examples for the problem described in Section 2.2 or in details in [18]. We consider a triangle whose vertices are given by the following three points in the complex plane:

$$
\begin{aligned}
z_1 &= -0.4655 + 0.2201i, \\
z_2 &= 0.0082 + 0.4599i \\
z_3 &= -0.3283 - 0.1809i.
\end{aligned}
\tag{16}
$$

Following the procedure described in [18] we aim at reconstructing such a triangle from a set of (perturbed) complex moments by solving a Hankel low-rank approximation problem. Since we deal with complex valued data we only use the proposed method to study the problem without any comparison.

We start from a set $T = (\tau_0, \ldots, \tau_{N-1})$ of N complex moments and we add a random Gaussian perturbation to both their real and imaginary part. Then we solve the Hankel low-rank approximation problem for the matrix $H_{N-n}(T)$ and we recover the set of vertices from its kernel, as described in Section 2.2 [18]. The error between the exact and the approximating solution is measured by looking at the two sets of vertices:

$$
e = \left\| \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} - \begin{pmatrix} \hat{z}_1 \\ \hat{z}_2 \\ \hat{z}_3 \end{pmatrix} \right\|_2
$$

where the vertices are ordered by decreasing real part. All the following results are the average over fifty runs (we generate fifty random perturbations and we
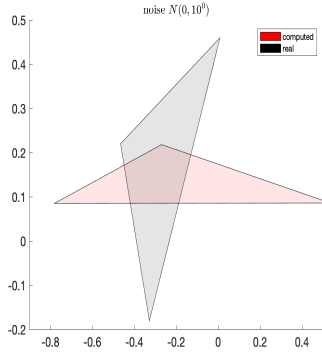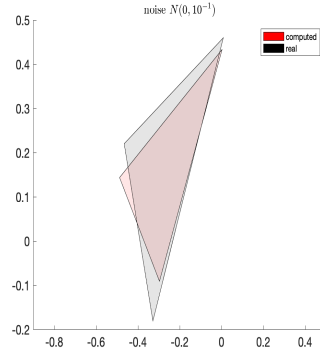
16

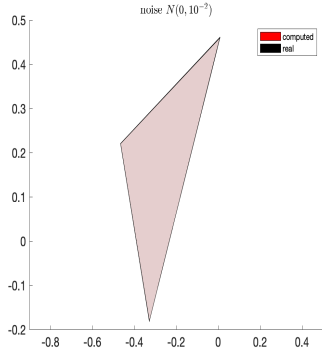Figure 4: Noise level 10^0

Figure 5: Noise level 10^{-1}

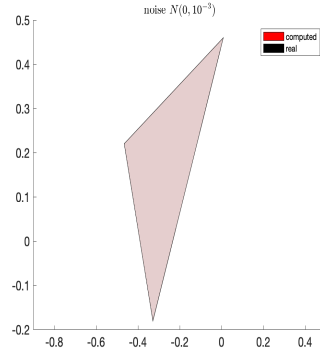Figure 6: Noise level 10^{-2},

Figure 7: Noise level 10^{-3}

Figure 8: Triangle from moments reconstruction: numerical results for different noise levels

consider the average solution so that the results of the experiments are not misleading),

First we show some graphical results, in order to see what actually happens. In Figure 8 we observe the numerical results for different levels of noise (the scalar τ in (15)) and 9 complex moments.

In the following analysis we want to analyze how the error behaves as function of the level of noise (for a fixed number of moments) and as function of the number of moments (for a fixed level of noise).

In Figure 9 we observe how the error increases linearly (using a logarithmic scale on both axis) with the level of noise (we considered 9 complex moments).

The next analysis can help to understand how the error changes with the
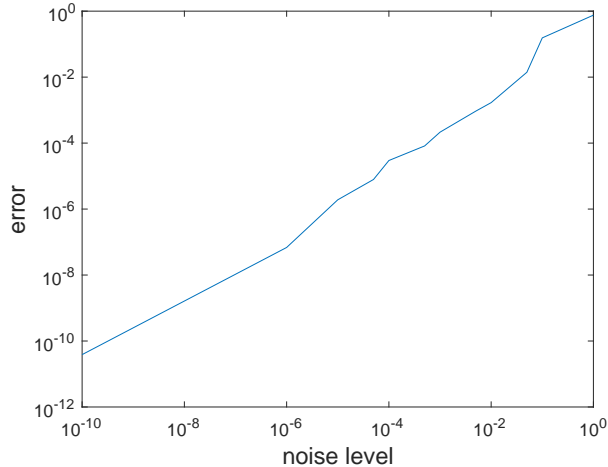
Figure 9: Error as function of the noise level

number of considered complex moments. We fix the noise level to 10^{-3}. First of all we observe how the modulus of the (exact) complex moments behaves for an increasing number N (Figure 10). The first two moments are always zero by definition. In [18] it is proved that a minimum of $2n + 1$ moments (n is the number of vertices) is necessary for the reconstruction. From Figure 10 we can see that the modulus of the moments is decreasing, so we can expect that the smaller is the magnitude the less is the influence of a moment in the reconstruction. We can plot now how the error changes for an increasing number of complex moments (Figure 11), where we observe how the error is decreasing for an increasing number of complex moments. Consequently, according to the needed accuracy on the computed solution we can choose to use a certain number of complex moments in order to optimize the computational cost.

Remark 6. This last analysis on the dependence of the error from the number of complex moments is strongly linked to the choice of the set of vertices. This is because the complex moments (and their magnitudes) are functions of the vertices.

5. Conclusion and future work

We proposed a new algorithm for solving Hankel structured low-rank approximation problems. It is based on a double iteration method which makes the smallest singular value of the data matrix decreasing along the direction described by a gradient system. We saw how the algorithm performs similarly as the function *slra* for what concerns the accuracy on the computed solution and how to use it as initial approximation in order to achieve an improvement. Moreover the proposed approach turns out to be more robust with respect to the
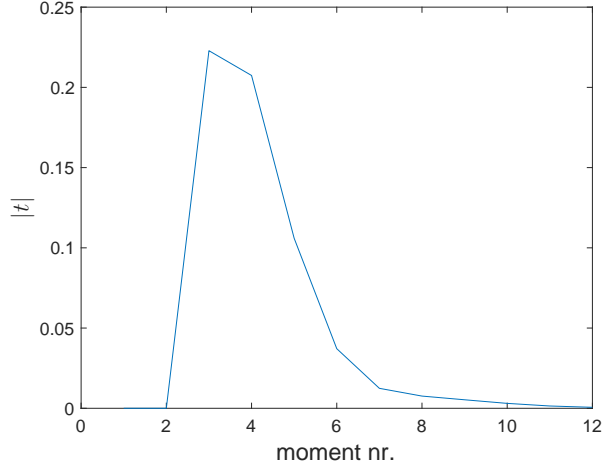
18

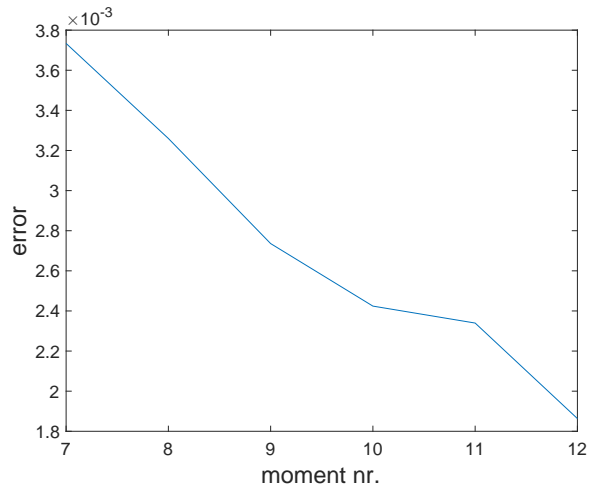Figure 10: Modulus of the first (exact) 16 complex moments for the vertices in (16)

Figure 11: Error for an increasing number of complex moments. The noise level is fixed.

initial approximation given in input. The algorithm can be extended to block Hankel matrices, and to mosaic Hankel matrices (block matrices whose blocks are Hankel matrices) arising in applications in the field of system theory and identification. A more challenging task is to extend the proposed approach to the problem of Hankel low-rank approximation with multiple rank constraints appearing, e.g., in the common dynamics estimation problem in multi-channel signal processing [22]. Moreover, a similar strategy can be adopted to compute rank reductions greater than one. The optimization of the proposed algorithm

in terms of computational cost and time can be object of future work.

Acknowledgements

References

[1] D. Bini, V. Y. Pan, Polynomials and Matrix Computations, Birkäuser Boston, Boston, MA, 1994.

[2] V. Peller, Hankel Operators and their Applications, Springer, New York, 2003.

[3] H. Park, L. Zhang, J. B. Rosen, Low Rank Approximation of a Hankel Matrix by Structured Total Least Norm, BIT 39 (1999) 757–779.

[4] A. K. Shaw, S. Pokala, R. Kumaresan, Toeplitz and Hankel matrix approximation using structured approach, in: Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, Seattle, WA, 1998, pp. 2349–2352.

[5] M. T. Chu, R. E. Funderlic, L. J. Plemmons, Structured low rank approximation, Linear Algebra Its Appl. 366 (2003) 157–172.

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

[7] I. Markovsky, Recent progress on variable projection methods for structured low-rank approximation, Signal Processing 96 (2014) 406–419.

[8] M. Fazel, Matrix rank minimizqtion with applications, Phd thesis, Elec. Eng. Dept., Stanford University, 2002.

[9] A. Fazzi, N. Guglielmi, I. Markovsky, An ODE based method for computing the Approximate Greatest Common Divisor of polynomials, Numer. Algorithms 81 (2019) 719–740.

[10] P. Buttà, N. Guglielmi, S. Noschese, Computing the structured pseudospectrum of a Toeplitz matrix and its extreme points, SIAM J. Matrix Anal. Appl. 33 (2012) 1300–1319.

[11] K. Usevich, I. Markovsky, Variable projection for affinely structured low-rank approximation in weighted 2-norms, J. Comput. Appl. Math. 272 (2014) 430–448.

[12] I. Markovsky, J. C. Willems, S. Van Huffel, B. De Moor, Exact and Approximate Modeling of Linear Systems: A Behavioral Approach, SIAM, 2006.

[13] S. Van Huffel, P. Lemmerling, Total Least Squares and Errors-In-Variables modeling, Springer-Science + Business-Media, Dordrecht, 2002.

[14] A. Bultheel, B. De Moor, Rational approximations in linear systems and control, J. Comput. Appl. Math. 121 (2000) 355–378.

[15] B. De Moor, Structured total least squares and l_2 approximation problems, Linear Algebra Its Appl. 188–189 (1993) 163–205.

[16] B. De Moor, Total least squares for affinely structured matrices and the noisy realization problem, IEEE Transactions on Signal Processing 42 (1994) 3104–3113.

[17] B. De Moor, Structured Total Least Squares for Hankel Matrices, in: A. Paulraj, V. Roychowdhury, C. D. Schaper (Eds.), Communications, Computation, Control, and Signal Processing, Springer, Boston, MA, 1997, pp. 243–258.

[18] P. Milanfar, G. C. Verghese, W. C. Karl, A. S. Willsky, Reconstructing polygons from moments with connections to array processing, IEEE Transactions on Signal Processing 43 (1995) 432–443.

[19] P. Milanfar, M. Putinar, J. Varah, B. Gustafsson, G. Golub, Shape reconstruction from moments: Theory, algorithms, and applications, in: Proc. SPIE 4116, Advanced Signal Processing Algorithms, Architectures, and Implementations X, 2000, pp. 406–416.

[20] A. Hildebrand, Introduction to Numerical analysis, McGraw-Hill, New York, 1956.

[21] T. Kato, Perturbation theory for linear operators, Springer-Verlag, 1995.

[22] I. Markovsky, T. Liu, A. Takeda, Subspace methods for multi-channel sum-of-exponentials common dynamics estimation, in: Proc. of the IEEE Conf. on Decision and Control, Nice, France, 2019, pp. 2672–2675.