Global and local optimization methods for structured low-rank approximation ⋆

Konstantin Usevich [a], Ivan Markovsky [a]

[a] *Vrije Universiteit Brussel, Department ELEC, Pleinlaan 2, B-1050, Brussels, Belgium*

Abstract

Many data modeling problems can be posed and solved as a structured low-rank approximation problem. In this paper the variable projection method is used to reformulate the structured low-rank approximation problem as minimization of a multivariate rational cost function on a Grassmann manifold. We compare polynomial algebra methods for global optimization of the rational cost function using two different parametrizations of the manifold. We also propose an algorithm for local optimization based on switching between coordinate charts of the manifold, and compare the proposed method with other local optimization methods.

Key words: system identification, structured low-rank approximation, global optimization, local optimization, variable projection, Grassmann manifold, coordinate charts, structured total least squares, computer algebra, symbolic-numeric computations, rational function minimization, system of polynomial equations

1 Introduction

Structured low-rank approximation is a prototypical problem for many applied problems in systems, control and signal processing. Problems of model reduction, system identification (output-error and error-in-variables), approximate realization, approximate deconvolution, distance to controllability, etc., can be posed and solved as a structured low-rank approximation for a suitably chosen structure, rank and approximation criterion (see (Markovsky 2008) for an overview).

Problem 1 (Structured low-rank approximation) *Given $p \in \mathbb{R}^{n_p}$, structure $\mathscr{S}$, norm $\|\cdot\|$ and natural number $r < m$*

$$\underset{\widehat{p} \in \mathbb{R}^{n_p}}{\text{minimize}} \ \|\widehat{p} - p\| \ \text{ subject to } \ \operatorname{rank} \mathscr{S}(\widehat{p}) \leq r. \qquad (1)$$

The structure $\mathscr{S}$ is a map from a structure parameter space $\mathbb{R}^{n_p}$ to the space of matrices $\mathbb{R}^{m \times n}$. This paper consists of two parts which address global and local optimization methods for solving Problem 1.

Problem 1 is non-convex and there is a need for global optimization methods driven by many applications. Problem 1 can be solved globally by computer algebra methods (Dreesen *et al.* 2012), since the set of all structured matrices of low-rank is an algebraic variety. However, the methods of (Dreesen *et al.* 2012) have high computational complexity and only small examples are feasible (e.g. $n_p = 5$). In this paper we show that (1) can be reduced to minimization of a rational function by using the *variable projection* method. The reformulation as rational minimization involves less indeterminates than (Dreesen *et al.* 2012) and can be used for larger problems (e.g. $n_p = 20$). In this paper we consider only global optimization methods based on symbolic-numeric computations.

There are many available methods for local optimization of (1). The most efficient methods are based on the variable projection principle. They reformulate the problem as minimization of a cost function $f(R)$, where the R is the left kernel of a low-rank structured matrix $\mathscr{S}(\widehat{p})$. Historically, local optimization methods used the *structured total least squares* formulation of Problem 1, where the left kernels are of the form $[X \ -I_d]$. The reduced problem can be ill-posed or ill-conditioned, therefore in recent years attention switched to the well posed structured low-rank approximation. The variable projection principle applied to structured low-rank approximation leads to optimization of a function $f(R)$ on a Grassmann manifold.

⋆ The material in this paper was partially presented at the 16th IFAC Symposium on System Identification (SYSID 2012), July 11–13, Brussels, Belgium. Corresponding author K. Usevich. Tel. +32(0)26292979 Fax +32(0)26292850.

Email addresses: Konstantin.Usevich@vub.ac.be (Konstantin Usevich), Ivan.Markovsky@vub.ac.be (Ivan Markovsky).

In this paper we review different approaches for local optimization on the Grassmann manifold, in particular, constrained optimization methods and optimization methods using Riemannian geometry of the manifold. We also present an optimization method that uses representations of subspaces as $[X\ -I_d]\Pi$, where Π is a permutation matrix (coordinate charts of the Grassmann manifold (Helmke and Moore 1994)). The method switches between permutation matrices in the course of the optimization. This approach is new in the context of structured low-rank approximation, but proved to be efficient in computing Lagrangian invariant subspaces (Mehrmann and Poloni 2012).

The paper is organized as follows. In Section 2 we summarize the variable projection method for affinely structured low-rank approximation. We show that the variable projection method leads to optimization on a Grassmann manifold and provide an interpretation of the structured total least squares method as a restriction of the structured low-rank approximation problem. In Section 3 we review the methods of local optimization on a Grassmann manifold, and introduce a method which is based on the idea of switching coordinate charts in the course of the optimization. In Section 4 we show that the cost function is rational and compare different available approaches for minimization of a multivariate rational function. The methods are compared in terms of theoretical computational complexity and practical efficiency. In Section 5 we show results of numerical experiments for the presented methods.

2 Structured low-rank approximation

2.1 Affine structures and variable projection

An *affine matrix structure* $\mathscr{S}(p)$ is an affine map $\mathbb{R}^{n_p} \to \mathbb{R}^{m\times n}$ defined as

$$\mathscr{S}(p) = S_0 + \sum_{i=1}^{n_p} p_i S_i. \tag{2}$$

In this paper, we assume that $m < n$. The family $\{\mathscr{S}(p) : p \in \mathbb{R}^{n_p}\}$ is called the *$\mathscr{S}$-structured matrices*. We consider the case of unweighted 2-norm in Problem 1. As shown in (Usevich and Markovsky 2012*b*), problem (1) with weighted 2-norm and/or fixed values constraints can be reduced to another problem (1) with 2-norm by adjusting the structure.

The rank constraint $\mathscr{S}(\widehat{p}) \le r$ is equivalent to existence of a full row rank $R \in \mathbb{R}^{d\times m}$, where $d := m - r$, such that

$$R\mathscr{S}(\widehat{p}) = 0. \tag{3}$$

Hence Problem 1 can be reformulated as the following double minimisation problem

$$\underset{R:\,\mathrm{rank}\,R=d}{\text{minimize}}\ f(R), \quad \text{where} \tag{4}$$

$$f(R) := \min_{\widehat{p}\in\mathbb{R}^{n_p}} \|\widehat{p} - p\|_2^2 \ \text{ subject to } (3). \tag{5}$$

The inner minimization problem is a linear *least-norm problem* (Boyd and Vandenberghe 2004, Ch. 6) and has a closed-form solution, which is presented in Section 2.2. Therefore, we can eliminate $\widehat{p}$ from the problem and solve (4). See (Markovsky *et al.* 2006, Ch. 4) or (Markovsky 2012, Ch. 3) for more details on the equivalence between problems (1) and (4).

The cost function (5) is homogeneous in the following sense:

$$f(R) = f(UR) \quad \text{for any nonsingular matrix} \quad U \in \mathbb{R}^{d\times d}.$$

Therefore, $f(R)$ depends only on the row space of R. This implies that f is defined on a *Grassmann manifold* $\mathrm{Gr}_{\mathbb{R}}(d,m)$ (Helmke and Moore 1994, App. C) (the manifold of all d-dimensional subspaces of $\mathbb{R}^m$).

2.2 Inner minimization problem

Equation (3) can be rewritten equivalently as

$$G(R)(\widehat{p} - p) = s(R),$$

where $G(R) \in \mathbb{R}^{nd\times n_p}$ and $s(R) \in \mathbb{R}^{nd\times 1}$ are given by

$$s(R) := \mathrm{vec}\,(R\mathscr{S}(p)),$$
$$G(R) := \begin{bmatrix}\mathrm{vec}\,(RS_1) & \cdots & \mathrm{vec}\,(RS_{n_p})\end{bmatrix},$$

and are linear functions in the elements of R. If the assumption $n_p \ge nd$ holds and $G(R)$ is full row rank, the inner minimization problem (5) is a *least-norm problem* (Boyd and Vandenberghe 2004, Ch. 6). Its solution is given by

$$f(R) = s(R)^\top \Gamma^{-1}(R) s(R), \tag{6}$$

where $\Gamma(R) := G(R)G(R)^\top \in \mathbb{R}^{nd\times nd}$.

Many applications from (Markovsky 2008) can be posed as Problem 1 with the *mosaic Hankel* structure (Markovsky and Usevich 2012*a*). In this case the cost function and its derivatives can be evaluated efficiently (Usevich and Markovsky 2012*b*).

2.3 Structured total least squares

Problem (4) with the additional constraint

$$R = \begin{bmatrix}X & -I_d\end{bmatrix}, \quad X \in \mathbb{R}^{d\times r}.$$

is a *structured total least squares problem* (see Appendix A)

$$\underset{X\in\mathbb{R}^{d\times r}}{\text{minimize}}\ f\left(\begin{bmatrix} X & -I_d \end{bmatrix}\right). \tag{7}$$

The restricted problem (7) is an unconstrained optimization problem in the space $\mathbb{R}^{d\times r}$. This allows us to use a wide class of methods for unconstrained optimization. Another advantage of (7) over (4) is that a subspace is represented by at most one element of the form $[X\ -I_d]$.

The matrices of the form $[X\ -I_d]$ do not represent all subspaces from $\mathrm{Gr}_{\mathbb{R}}(d,m)$. Hence, problem (7) can be ill-posed (if the minimum R^* of $f(R)$ cannot be represented as $[X^*\ -I_d]$) or ill-conditioned (if magnitudes of elements of X^* are large). In the next section we provide ways to parametrize the whole manifold $\mathrm{Gr}_{\mathbb{R}}(d,m)$.

3 Optimization on Grassmann manifold

In this section, we present ways to parametrize the whole manifold $\mathrm{Gr}_{\mathbb{R}}(d,m)$ and solve optimization problems on $\mathrm{Gr}_{\mathbb{R}}(d,m)$.

3.1 Orthogonal bases

For any subspace $\mathscr{R}\in\mathrm{Gr}_{\mathbb{R}}(d,m)$, there exists an orthonormal basis (a matrix R with orthonormal rows, such that $\mathrm{rowspan}\,R=\mathscr{R}$, hence it is sufficient to consider the set

$$\mathscr{M}_{ort}=\{R\in\mathbb{R}^{d\times m}: RR^\top=I_d\}, \tag{8}$$

and reduce (4) to

$$\underset{R\in\mathscr{M}_{ort}}{\text{minimize}}\ f(R). \tag{9}$$

Note 1 *The parametrization* (8) *is ambiguous, because for an $R\in\mathscr{M}_{ort}$ and any orthogonal matrix $U\in\mathbb{R}^{d\times d}$ it holds that $UR\in\mathscr{M}_{ort}$. In particular, an optimal R for problem* (9) *is not unique.*

$\mathscr{M}_{ort}$ is a compact set (*compact Stiefel manifold* (Helmke and Moore 1994, App. C)). This implies that if $f(R)$ is continuous, then it attains a global minimum on $\mathscr{M}_{ort}$, which makes the problem (9) well-posed. The problem (9) is a constrained minimization problem, and various methods can be applied, for example a method in Section 3.3.

3.2 Regularization method

In (Markovsky and Usevich 2013) it was shown that the hard constraint in (9) can be replaced by a soft constraint. Unlike many other penalty methods, the penalty parameter can be rather small.

Theorem 2 ((Markovsky and Usevich 2013)) *Let $\widetilde{f}(R)$ be a homogeneous extension of* (5)*, i.e. $\widetilde{f}(R)=f(R)$ for full row rank R, and $\widetilde{f}(UR)=\widetilde{f}(R)$ for any nonsingular U. Let γ be a strict upper bound on the solution of* (4)*. Then the solutions of*

$$\underset{R\in\mathbb{R}^{d\times m}}{\textit{minimize}}\ f(R)+\gamma\|RR^\top-I_d\|_{\mathrm{F}}^2, \tag{10}$$

coincide with the solutions of (4)*.*

Note 2 *For linear structures (i.e. $S_0=0$) the choice $\gamma=\|p\|_2^2$ is sufficient for the conditions of Theorem 2.*

In order to use Theore 2 for practical optimization we need to compute derivatives of the regularization term. For a function $g:\mathbb{R}^{d\times m}\to\mathbb{R}$ denote its matrix gradient $\nabla_{d\times m}g$ the function that $\mathrm{vec}(\nabla_{d\times m}g)=\nabla(\mathrm{vec}\,g)$. The following lemma holds by straightforward differentiation (see Appendix B).

Lemma 3 *For the function $g(R)=\|RR^\top-I_d\|_2^F$*

$$\begin{aligned}&\nabla_{d\times m}g=4(RR^\top-I_d)R,\\ &H(g)\,vec\,E=4\,vec\left(ER^\top R+RE^\top R+RR^\top E-E\right),\end{aligned}$$

where $H(g)$ is the Hessian of g with respect to $\mathrm{vec}\,R$.

3.3 Optimization in Riemannian geometry

In recent years optimization methods in Riemannian geometry became increasingly popular for solving optimization problems on Riemannian manifolds (Absil *et al.* 2008). Let $\mathscr{M}$ be a Riemannian manifold of dimension N. For each point $x\in\mathscr{M}$ a tangent space $T_x\mathscr{M}$ is associated. The tangent space is an N-dimensional vector space equipped with an inner product $\langle\cdot,\cdot\rangle_x$, that smoothly varies depending on x. The derivatives of a function on the manifold are defined in the tangent space.

The methods of (Absil *et al.* 2008) require a *retraction* $R_x: T_x\mathscr{M}\to\mathscr{M}$ to be defined. The methods in (Absil *et al.* 2008) are based on the following scheme. From a current iteration x_k, a direction ξ_k is selected in the tangent space T_x, based on the derivatives of f at x_k. The new iterate is set to be $x_{k+1}=R_{x_k}(\xi_k)$ (compare with setting $x_{k+1}=x_k+\xi_k$ in optimization on $\mathbb{R}^n$). The direction selection is chosen using an analogue of Newton iteration or trust-region iteration. The latter choice demonstrates better practical performance, as with ordinary optimization on $\mathbb{R}^n$.

The first case where the approach of (Absil *et al.* 2008) can be used is a method of constrained optimization for (9), using the geometry of the Stiefel manifold (8). The Stiefel manifold is a submanifold of $\mathbb{R}^{d\times m}$. The tangent space is the tangent space to the surface described by $RR^\top=I_d$. The retraction is defined through the QR factorisation.

The second case is the direct optimization on the Grassmann manifold. In this case it cannot be constructed as a submanifold of the Euclidean space. Grassmann manifold is constructed as a quotient manifold $\mathbb{R}_*^{d\times m}/\sim$, where

$$R_1 \sim R_2 \iff \operatorname{rowspan} R_1 = \operatorname{rowspan} R_2.$$

The points on the manifold can be represented as $R \in \mathbb{R}_*^{d\times m}$. Usually R is normalized as $RR^\top = I_d$ for convenience. The tangent space is associated with $T_R = \{XR_\perp : X \in \mathbb{R}^{d\times r}\}$. Then the gradient is equal to $\operatorname{grad} f(R) = \nabla f(R)$, and the gradient in the tangent space is

$$\begin{aligned}&\operatorname{tr}\left((XR_\perp)^\top \operatorname{grad} f(R)\right) = \operatorname{tr}\left(X^\top \operatorname{grad} f(R) R_\perp^\top\right)\\ &= \operatorname{tr}\left(X^\top \nabla_X f\left(\begin{bmatrix} X & -I_d\end{bmatrix}\Phi\right)\right),\end{aligned}$$

where

$$\Phi = \begin{bmatrix} R_\perp \\ -R\end{bmatrix}.$$

The same calculations can be performed for the Hessian of f. We have that derivatives in the tangent space to the manifold are equal to the derivatives of $f([X\ I_d]\Phi)$.

3.4 Optimization in coordinate charts

For any $d \times m$ matrix R of full row rank, there is a set of d linearly independent columns. Equivalently, there exists a permutation matrix Π such that

$$R = \begin{bmatrix} Q & -P\end{bmatrix}\Pi,$$

where $Q \in \mathbb{R}^{d\times r}$ and P is a $d \times d$ nonsingular matrix. Hence,

$$\operatorname{rowspan} R = \operatorname{rowspan}\left(\begin{bmatrix} P^{-1}Q & -I_d\end{bmatrix}\Pi\right), \qquad (11)$$

and any subspace $\mathscr{R} \in \operatorname{Gr}_{\mathbb{R}}(d,m)$ can be represented by a matrix of the form $[X\ -I_d]\Pi$. Therefore, the following proposition holds true.

Proposition 4 *Problem* (4) *is equivalent to*

$$\underset{\Pi}{\text{minimize}} \min_{X\in\mathbb{R}^{d\times r}} f_\Pi(X), \qquad (12)$$

where

$$f_\Pi(X) := f\left(\begin{bmatrix} X & -I_d\end{bmatrix}\Pi\right).$$

Note 3 *Each subproblem in* (12)

$$\underset{X\in\mathbb{R}^{d\times r}}{\text{minimize}}\ f_\Pi(X), \qquad (13)$$

is equivalent to problem (7) *for the structure* $\Pi\mathscr{S}(p)$. *Therefore, problem* (4) *can be considered as a union of structured total least squares problems corresponding to the structures* $\mathscr{S}(p)$ *with permuted rows.*

Note 4 *For a fixed* Π, *the map*

$$X \mapsto \operatorname{rowspan}\left(\begin{bmatrix} X & -I_d\end{bmatrix}\Pi\right)$$

is bijective (its inverse is also called a standard coordinate chart (Helmke and Moore 1994, App. C) of the manifold $\operatorname{Gr}_{\mathbb{R}}(d,m)$*).*

3.5 Bounded optimization in coordinate charts

However, problem (13) still suffers from being ill-posed or ill-conditioned. The next theorem helps to reduce the search space of the problems (13) to compact subsets of $\mathbb{R}^{d\times r}$.

Theorem 5 (Knuth (1985)) *For any* $R \in \mathbb{R}^{d\times m}$ *of rank* d *there exists a permutation matrix* Π *such that all the elements of the matrix* $P^{-1}Q$ *in* (11) *are contained in* $[-1;1]$.

Note 5 *In (Usevich and Markovsky 2012*a*) a weaker than Theorem 5 result was proved. The result of (Usevich and Markovsky 2012*a*) was based on Gaussian elimination, and Theorem 5 is based on properties of determinants of submatrices (Knuth 1985).*

Corollary 6 *Problem 1 is equivalent to*

$$\underset{\Pi}{\text{minimize}} \min_{X\in[-1;1]^{d\times r}} f_\Pi(X). \qquad (14)$$

Each subproblem in (14) for a fixed Π is an unconstrained optimization problem on a compact domain, and is well-posed. However, the outer minimization in (14) involves choosing an optimal permutation matrix Π out of $\binom{m}{d}$ permutations.

For local optimization with initial approximation R_0 the exhaustive search over all possible permutations, however, can be avoided by switching between permutations in the course of optimization. The following algorithm, based on Corollary 6, finds a locally optimal solution of (6).

Algorithm 1 *Input: initial approximation p, r,* $\mathscr{S}$, R_0, Δ. *Output:* $\widehat{R}$ *yielding a local minimum of* $f(R)$

(1) Set $k = 0$.
(2) For R_k *choose a permutation* Π_k *and* $U_k \in \mathbb{R}^{d\times d}$, *such that*

$$U_k R_k = \begin{bmatrix} \widehat{X}_k & -I_d\end{bmatrix}\Pi_0 \quad \text{and} \quad \widehat{X}_k \in [-1;1]^{d\times r}.$$

(3) Perform local optimization of f_{Π_k} *until convergence and unless* $|\widehat{x}_{i,j}| < \Delta$ *for all* i, j. *Set* $R_{k+1} = [\widehat{X}_k\ -I_d]\Pi_k$, *where* $\widehat{X}_k$ *is the last iterate of the local optimization subproblem.*

(4) If the local optimization subproblem converged, output $\widehat{R} = R_{k+1}$ *and* ***stop****.*
(5) Else increase k and go to step 2.

Note 6 *A very similar algorithm was used in (Mehrmann and Poloni 2012) for computing Lagrangian invariant subspaces. Here we emphasize that Algorithm 1 can be used for optimization of an arbitrary function on a Grassmann manifold, and any standard local optimization method can be used for each sub-problem between switches.*

The threshold Δ is chosen to be ≥ 1. In order to complete the description of the proposed algorithm, we need a procedure to reduce R_k to the form X_k. We will use an algorithm for updating a permutation Π_{k-1} to Π_k such that the absolute value of corresponding elements is bounded by 1.

Algorithm 2 *Input: X and* Π*.*
Output: X', Π' *such that*

$$\operatorname{rowspan}\begin{bmatrix} X & -I \end{bmatrix}\Pi = \operatorname{rowspan}\begin{bmatrix} X' & -I \end{bmatrix}\Pi'$$

and $|(X')_{i,j}| < 1$.

(1) Set $X' \leftarrow X$, $\Pi' \leftarrow \Pi$.
(2) Find k,l *such that* $|(X')|_{k,l}$ *is maximal.*
(3) If $|(X')|_{k,l} \leq 1$ ***stop****.*
(4) Else define $\Pi_{(l,k+r)}$ *as a matrix permuting l-th and* $(k+r)$*-th columns.*
(5) Partition $[P\,Q] = [X\ -I_d]\Pi_{(l,k+r)}$.
(6) Update $X' \leftarrow -Q^{-1}P$, $\Pi' \leftarrow \Pi_{(l,k+r)}\Pi'$ *and* ***goto*** *2.*

4 Problem 1 as a rational minimization problem: global optimization methods

Since $G(R)$ is linear in R, the matrix $\Gamma(R)$ is a *quadratic polynomial matrix* (having entries which are quadratic functions in elements of R). If $\det\Gamma(R)$ is not a zero polynomial, then we can define the rational inverse of $\Gamma(R)$

$$\Gamma^{-1}(R) = \frac{\operatorname{adj}\big(\Gamma(R)\big)}{\det\big(\Gamma(R)\big)}, \tag{15}$$

where $\operatorname{adj}\Gamma(R)$ is the adjugate matrix (Strang 1988, Ch. 4).

If for a fixed R the matrix $\Gamma(R)$ is nonsingular, then the pseudoinverse coincides with the polynomial inverse and the cost function (5) is equal to

$$f(R) = \frac{s(R)^\top \operatorname{adj}\big(\Gamma(R)\big)s(R)}{\det\big(\Gamma(R)\big)}. \tag{16}$$

Remark 7 *If* $\det(\Gamma(R)) > 0$ *for all matrices R under consideration, then Problem 1 problem is equivalent to minimization of a rational function.*

In this section we consider methods of optimization of a rational function that exploit the algebraic structure of the problem and allow to find the global minima.

4.1 Overview of optimization methods

4.1.1 Semidefinite programming relaxations

Problem (9) or each subproblem of (14) can be reduced to minimization of a rational function $Q(z)/H(z)$, $z \in \mathbb{R}^M$ subject to polynomial constraints $C_k(z) \geq 0$, $1 \leq k \leq s$. This problem is equivalent to minimization over the set of Borel measures

$$\begin{aligned}\underset{\mu}{\text{minimize}} \quad & \int_{\mathbb{R}^M} Q(z)d\mu, \text{ subject to} \\ & \int_{\mathbb{R}^M} H(z)d\mu = 1, \int_{\mathbb{R}^M} C_k(z)d\mu \geq 0,\end{aligned}$$

where the minimum corresponds to an atomic measure μ. Problem is a *generalized moment problem* (Hanzon and Jibetean 2003) and can be solved by semidefinite programming (SDP) relaxations of fixed order. It is known that the relaxations converge to a global minimum (Jibetean and de Klerk 2006), however, the speed of convergence is not guaranteed. Optimization of a rational function via SDP relaxations is implemented in the package GloptiPoly (Hanzon and Jibetean 2003).

4.1.2 Systems of polynomial equations

The second, more general, approach consists of finding stationary points of the cost function as solutions of polynomial systems of equations.

Consider first the STLS problem (13) for a fixed Π. Since $f_\Pi(X)$ is a rational function with no singularities, the local minima of the function are attained at the stationary points, i.e. solutions of the system

$$\frac{\partial}{\partial X_{ij}} f_\Pi(X) = 0, \quad \text{for } i = 1,\dots,d,\ j = 1,\dots,r. \tag{17}$$

Denote

$$f_\Pi(X) = \frac{Q_\Pi(X)}{H_\Pi(X)},$$

where $H(X) \neq 0$. Then the system (17) is equivalent to

$$\begin{aligned} q_{ij}(X) := \frac{\partial Q_\Pi}{\partial X_{ij}}(X)H_\Pi(X) - Q_\Pi(X)\frac{\partial H_\Pi}{\partial X_{ij}}(X) = 0, \\ \text{for } i = 1,\dots,d,\ j = 1,\dots,r, \end{aligned} \tag{18}$$

which is a system of polynomial equations. The number of equations coincides with the number of variables dr.

Problem (9) can be also reduced to the problem of finding stationary points by introducing Lagrange multipliers for the constraint $RR^\top = I_d$. The number of equations in this case is again equal to the number of indeterminates.

4.2 Solving systems of polynomial equations

In what follows, we consider a polynomial system with the number of equations equal to the number of variables

$$\begin{aligned} q_1(z_1,\dots,z_M) &= 0, \\ &\vdots \\ q_M(z_1,\dots,z_M) &= 0, \end{aligned} \tag{19}$$

also called a *complete intersection system.*

A system of multivariate polynomial equations may have infinite number of solutions. Most multivariate solvers deal with the case of finite number of solutions and we consider only this case. The number of complex solutions N (and therefore possible local minima of (4)) is bounded by the Bezout bound (Stetter 2004, Ch. 8):

$$N \leq \prod_{i=1}^{M} \deg(q_i). \tag{20}$$

4.2.1 Resultant-based methods

These methods are based on multivariate resultants. They reduce the solution to univariate polynomial equations, however they are applicable only for small degrees or sparse polynomials (see for example (Sturmfels 2002)). The polynomials in (18) are usually dense and have large degrees, so we do not consider these methods.

4.2.2 Stetter-Moller matrix methods

The methods are based on the following fact: the coordinates of the roots of the system of equations coincide with the eigenvalues of the *multiplication matrices* A_{z_k}, corresponding to the joint eigenspaces of the matrices. See (Stetter 2004, Ch. 2) for more details.

The multiplication matrices are sparse $N \times N$ matrices, where N is the number of complex solutions (in the case of simple roots). To obtain the multiplication matrices, one needs to calculate a Gröbner basis. Having a Gröbner basis, the matrices can be calculated using, for example, *ApCoCoA (Applied Computations in Commutative Algebra).*

4.2.3 Triangular systems and rational univariate representations

One can find an equivalent to (19) triangular system

$$\begin{aligned} g_1(z_1) &= 0 \\ g_2(z_1,z_2) &= 0 \\ &\vdots \\ g_M(z_1,\dots,z_M) &= 0 \end{aligned}$$

and sequentially eliminate variables, solving at each step systems of polynomial equations in one variable. However, this procedure is numerically inaccurate and is being replaced by the rational univariate representation approach of (Rouillier 1999).

Both approaches require computation of a Gröbner basis. Efficient computations with Gröbner bases, in their turn, involve linear algebra computations with $N \times N$ matrices. State of the art algorithms for the above approaches are implemented as packages for *Maple*^{TM}.

4.2.4 Subdivision methods

Note that we are interested to find only the real roots of a polynomial system in a bounded box $[a_1,b_1] \times \cdots \times [a_M,b_M]$. For this particular case, one can exploit properties of the Bernstein polynomials to efficiently locate the real zeros (Mourrain and Pavone 2009).

The complexity of the subdivision algorithm depends on the number of coefficients in the Bernstein representation, which can be bounded by $(\deg(q_i))^M$ for a single polynomial q_i. The Bernstein subdivision algorithms are implemented in the package *SYNAPS (SYmbolic Numeric ApplicationS).*

4.2.5 Homotopy continuation

This method is based on considering a homotopy from a simpler system to the system which is to be solved, and tracking the solutions. In general, this is a heuristic method and there is no guarantee of obtaining all the roots.

All the complex solutions are tracked, therefore the complexity depends on N. This method is implemented in the package PHCpack (Verschelde 1999).

4.2.6 Perturbed systems

For large polynomial degrees, the computation of the Gröbner basis becomes infeasible. In (Hanzon and Jibetean 2003) it is proposed to solve a perturbed system

already in a form of a Gröbner basis

$$\begin{aligned} \lambda z_1^{D+1} + q_1(z_1,\ldots,z_M) &= 0 \\ &\vdots \\ \lambda z_M^{D+1} + q_M(z_1,\ldots,z_M) &= 0 \end{aligned} \tag{21}$$

where $D = \max \deg q_i$. This method is motivated by the problem of minimization of a polynomial $H(Z)$ of degree $2d$, where the perturbation of the system of equations is equivalent to perturbing the polynomial itself

$$H_\lambda(Z) := \lambda(z_1^{2d+2} + \ldots + z_M^{2d+2}) + H(Z). \tag{22}$$

The global minimum of $H(X)$ can be approximated by the global minima of $H_\lambda(X)$ when $\lambda \to 0$ (see (Hanzon and Jibetean 2003)).

We were not able to find a perturbation of a rational function that leads to a system of the form (21). However, we may perturb the system (18), expecting that the roots of the perturbed system (21) do not diverge too far.

4.2.7 Complexity of the methods

As is was mentioned above, the complexity of solving the polynomial system usually depends on the number of complex solutions. First we calculate the total degree of the involved polynomials. The polynomial matrix $\Gamma(R)$ consists of quadratic polynomials, and therefore

$$\deg \det \Gamma(R) \leq 2nd.$$

The degree of the numerator is also bounded by $2nd$, since

$$\deg \operatorname{adj} \Gamma(R) \leq 2(nd-1)$$

and $\deg s(R) = 1$.

Hence, the degrees of the polynomial equations (19) are bounded by $4nd-1$. The total number of solutions is therefore bounded by $(4nd-1)^{dr}$ for the STLS problem (13).

5 Numerical experiments

5.1 Local optimization on Grassmann manifold

In this section we compare the existing methods for local optimization of Problem 1 with the method given in Algorithm 1.

5.1.1 Experimental setup

We compare five local optimization methods:

- `stls` — optimization over X for Problem 7;
- `perm` — optimization with Algorithm 1;
- `genrtr` — optimization on Grassmann manifold in Riemannian geometry (see Section 3.3);
- `fmincon` — optimization with constraints (9);
- `reg` — optimization of regularized cost function (10).

All methods use fast evaluation of cost function and its derivatives implemented in the `slra` package (Markovsky and Usevich 2012*b*). Methods `stls` and `perm` (for each optimization subproblem) use the Levenberg-Marquardt method (Marquardt 1963) of nonlinear least-squares minimization for the function (6). The threshold Δ for the `perm` method is chosen to be $\sqrt{2}$.

Method `fmincon` uses the `fmincon` function from MATLAB Optimization Toolbox (default method without supplied derivatives). Method `reg` uses "Newton trust-region" method of `fminunc` function from MATLAB Optimization Toolbox (the gradient is supplied, the Hessian is computed by `fminunc` using finite differences). Method `genrtr` uses *GenRTR (Generic Riemannian Trust-Region package)* for optimization on Riemannian manifolds, with default parameters for the Grassmann manifold (projection, retraction, optimization settings) taken from (Boumal and Absil 2011). Package GenRTR requires a function of multiplication of the Hessian by a vector h. This product is replaced by its finite difference approximation

$$H(f)h \approx \frac{\nabla f(R+\varepsilon h) - \nabla f(R)}{\varepsilon},$$

where $\varepsilon = 10^{-9}$.

We compare the methods on several benchmark problems in system identification (De Moor *et al.* 1997). The data for each experiment are q-variate time series $w \in \mathbb{R}^{q \times T}$, which is an observed trajectory of a dynamical system

$$w(t) = (u(t), y(t)), \quad y(t) \in \mathbb{R}^{\mathtt{p}}, \quad u(t) \in \mathbb{R}^{q-\mathtt{p}}.$$

For each problem we fix a model class $\mathscr{L}^q_{\mathtt{m},\ell}$ of linear time-invariant system with at most $\mathtt{m} := q - \mathtt{p}$ inputs and lag at most ℓ. (For more details see (Markovsky *et al.* 2006) or (Markovsky 2008).) Then the errors-in-variables identification problem with the model class $\mathscr{L}^q_{\mathtt{m},\ell}$ can be posed as (1) with structure

$$\mathscr{S} = \begin{bmatrix} \mathscr{H}_{\ell+1,T-\ell}(u) \\ \mathscr{H}_{\ell+1,T-\ell}(y) \end{bmatrix}$$

and rank reduction by the number of outputs, i.e.

$$r = q(\ell+1) - \mathtt{p}.$$

For a time series $w \in \mathbb{R}^{q\times T}$

$$\mathscr{H}_{L,K}(p) = \begin{bmatrix} w(1) & w(2) & \cdots & w(K) \\ w(2) & w(3) & \cdots & w(K+1) \\ \vdots & \vdots & \vdots & \vdots \\ w(L) & \cdots & \cdots & w(T) \end{bmatrix} \in \mathbb{R}^{Lq\times K}$$

is a block-Hankel matrix constructed from the time series $w(\cdot)$. For more details see (Markovsky 2008) or (Markovsky and Usevich 2012*a*).

The methods are compared in terms of the % "fit", defined as

$$100\% \cdot \left(1 - \frac{\|w - \widehat{w}^*\|_{\mathrm{F}}}{\|w - \overline{w}\|_{\mathrm{F}}}\right),$$

where $\widehat{w}^*$ is the computed approximation of the trajectory and $\overline{w}$ is the vector of mean values of the variables $w_1, \ldots, w_q$.

5.1.2 Experiments on simulated data

In this section we consider a simulation example with a one-dimensional autonomous system ($q = \mathrm{p} = 1$). We take a trajectory of length $T = 400$ of a marginally stable system of order $\ell = 6$, given by equation

$$w_0(t) = 2\cos\left(\frac{2\pi t}{3}\right) + \cos\left(\frac{2\pi t}{7}\right) + \cos\left(\frac{2\pi t}{10}\right).$$

For different noise levels σ we draw one test realization of

$$w = w_0 + \varepsilon,$$

where ε is a realization of the Gaussian white noise with variance σ^2.

In our experiment for $\sigma = 0.02, \ldots, 0.2$ all methods converged to the same solution (subject to convergence tolerances). The tolerance for the gradient was set to 10^{-5}. All methods are started from the same initial approximation. In Table 1 the number of iterations is presented (maximum number of iterations is set to 200 for all methods).

Results in Table 1 show that with small noise all the methods except `genrtr` converge with a small number of iterations. Algorithm 1 makes just one switch of permutations in all examples.

5.1.3 Experiments on real-life datasets

In this section we test the optimization methods on the benchmark problems from the DAISY database (De Moor *et al.* 1997). A short summary of the considered examples can be found in Table 2. The examples include noisy trajectories of nonlinear dynamical systems. For more details on the examples see (De Moor *et al.* 1997).

Table 1
Number of iterations of the methods

σ	`stls`	`perm`	`genrtr`	`fmincon`	`reg`
0.02	2	2	200	8	7
0.04	3	2	200	9	6
0.06	3	3	200	9	7
0.08	3	3	200	10	4
0.1	3	3	200	9	8
0.12	3	3	200	10	9
0.14	3	3	200	10	10
0.16	3	3	200	11	9
0.18	3	3	200	11	9
0.2	4	4	200	12	13

Table 2
Description of the test cases

name	T	q	m	ℓ
erie_n20	57	7	5	1
destill_n30	90	8	5	1
heating_system	801	2	1	2
dryer	1,000	2	1	5
flutter	1,024	2	1	5

Table 3 shows the fit for each method and each test example. In Table 4 the number of iterations is presented (the limit on the number of iterations is set to 200 for all methods).

Table 3
Fits of the methods

`stls`	`perm`	`genrtr`	`fmincon`	`reg`
99.96	99.96	99.96	99.96	99.96
99.93	99.93	99.92	99.92	99.91
98.66	98.66	98.66	97.8	98.58
96.45	96.45	95.69	95.71	95.69
90.26	90.27	86.46	85.69	82.71

Table 4
Number of iterations of the methods

`stls`	`perm`	`genrtr`	`fmincon`	`reg`	switches
200	34	200	61	38	1
200	34	200	62	12	2
20	21	200	42	200	1
91	89	200	53	200	2
43	200	200	50	2	2

The results in Table 3 and Table 4 show that the developed algorithm `perm` improves the fit while usually making less iterations, compared to the methods `stls` and `fmincon`. The regularized method of (Markovsky and Usevich 2013) also shows good performance. Our experiments also showed

that using an approximation of the Hessian $2J^\top J$ (and another approximation proposed in (Guillaume and Pintelon 1996)) gives worse results both for `genrtr` and `reg`, despite the fact that this approximation is implicitly used in the Levenberg-Marquardt method.

5.2 Global optimization with polynomial methods

In this case we consider problem (1) for $p \in \mathbb{R}^T$ ($n_p = T$), scalar Hankel structure $\mathscr{H}_{m,n}(p)$, and rank $r = m-1$ (rank reduction by 1). This case is equivalent to identification of an autonomous linear-time-invariant system of order $\leq r$ (Markovsky 2008).

By (Usevich and Markovsky 2012*b*), the matrix $G(R)$ has the form

$$G(R) = \begin{bmatrix} r_1 & r_2 & \cdots & r_m & 0 & 0 \\ 0 & \ddots & \ddots & & \ddots & 0 \\ 0 & 0 & r_1 & r_2 & \cdots & r_m \end{bmatrix},$$

and is full row rank for all $R = \begin{bmatrix} r_1 & \cdots & r_m \end{bmatrix} \in \mathbb{R}^{1\times m} \neq 0$. Hence $f(R)$ and $f_\Pi(X)$ are rational functions without singularities on $\mathrm{Gr}_\mathbb{R}(d,m)$.

Our running example is with $m = 3$ rows and varying T. We denote $X = \begin{bmatrix} x_1 & x_2 \end{bmatrix}$ and consider three permutation maps

$$\begin{bmatrix} x_1 & x_2 & -1 \end{bmatrix} \Pi_1 = \begin{bmatrix} x_1 & x_2 & -1 \end{bmatrix},$$
$$\begin{bmatrix} x_1 & x_2 & -1 \end{bmatrix} \Pi_2 = \begin{bmatrix} -1 & x_1 & x_2 \end{bmatrix},$$
$$\begin{bmatrix} x_1 & x_2 & -1 \end{bmatrix} \Pi_3 = \begin{bmatrix} x_1 & -1 & x_2 \end{bmatrix},$$

which exhaust all possible row spaces of full row rank 1×3 matrices. By Corollary 6 it is sufficient to find the minima of the functions f_{Π_k} in the box $[-1,1]\times[-1,1]$.

5.2.1 Case of multiple local minima

Consider a time series

$$\begin{aligned} p_k &= s_k + \varepsilon_k, \quad \varepsilon_k \text{ i.i.d. } N(0,\sigma) \\ s_k &= \left(\frac{3}{4}\right)^{k-1} \sin\left(\frac{2\pi(k-1)}{6}\right) + \sigma\varepsilon_k, \end{aligned} \tag{23}$$

with $T = 12$. In this case $n = 10$ and the polynomials have degree 40, which leads to 1600 possible solutions of the polynomial system. This makes the computation of the Gröbner bases prohibitive. Therefore, we compare the approaches based on homotopy continuation, Bernstein subdivision, SDP relaxations and perturbed system.

For the choice $\sigma = 0.35$, the 2-norm of the noise ε_k is more than two times higher than the variance of the signal s_k. Multiple local minima of the cost function are likely to exist in this case. We consider a particular realization, and also introduce rounding.

$$p^{(1)} = [-0.14, 1, 0.21, -0.42, 0.255, -0.62, \\ 0.315, -0.1, -0.2, -0.21, 0.835, 0.005]^\top.$$

Experiments show that the global minimum is attained for f_{Π_1} in $[-1,1]\times[-1,1]$. Table 5 demonstrates the performance of the methods for minimization of f_{Π_1}.

Table 5
Performance of the methods in Example 1.

Method	Time (s)	Solution	$f(R)$
SDP, ort	55.6	failed	1.45585
SDP, u, 10	1.9	failed	1.45293
SDP, u, 20	19.3	failed	1.45672
SDP, b, 10	4.2	(-0.83663, -0.96014)	1.45290
PHCPack	373.9	(-0.83661, -0.96015)	1.45290
Perturbed	$> 4\cdot 10^3$	(-0.20134, 0.13092)	1.45536
SBDV	< 1	(-0.83661, -0.96015)	1.45290
STLS	< 0.01	(-0.83661, -0.96015)	1.45290

- SDP$_n$ denotes the GloptiPoly method with the order of relaxation n. The tolerance of the underlying SDP solver (SeDuMi) is set to 10^{-10}. "ort" corresponds to problem (9), "u" — to subproblem (13) for fixed Π,"b" — to subproblem of (14) with box constraints (domain bounded by polynomial inequalities $x^2 \leq 1$ and $y^2 \leq 1$). If the method fails to extract the solution, an upper bound on the minimum value is displayed.
- "Perturbed" stands for the solution of (21) by Rational Univariate Representation implementation in Maple. λ is taken to be 4000. (For comparison, the maximal absolute value of the polynomials' coefficients in the unperturbed system is $9.9828\cdot 10^6$ and is attained at the monomial of degree 30.) Eigenvalue computations in this case are severely ill-conditioned.
- For PHCpack and SBDV (Bernstein subdivision solver) the global minimum is computed by selecting the smallest value of the cost function in the box $[-1,1]\times[-1,1]$.
- STLS denotes the solution of problem (7) by `slra` package. The initial approximation is taken from unstructured low-rank approximation.

GloptiPoly with default order of relaxation (SDP, 10) and with increased order (SDP, 20) fails to extract the solution. Bounding the domain helps in this case, whereas solving problem (9) fails as well. SYNAPS and PHCPack find the same solution, however the perturbed system approach fails to find the true minimum because it is close to the boundary of the box.

In Fig. 1 one can see that PHCPack and SYNAPS identify correctly all stationary points. The roots of the perturbed system remain unchanged in the vicinity of 0, but are more

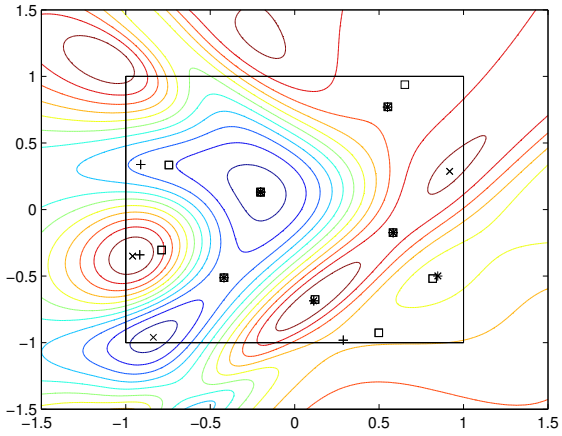

Fig. 1. Cost function for $p^{(1)}$, stationary points found by SYNAPS and PHCPack for the original system (x), and stationary points found by SYNAPS for the perturbed system with $\lambda = 4000$ (+) and $\lambda = 10^6$ (box)

likely to move or disappear if they are close to the boundary of the box.

5.2.2 *Case of single local minimum, $T = 19$*

Consider the time series (23)

$$p^{(2)} = [-0.051,\ 0.570,\ 0.478,\ -0.075,\ -0.348,\ -0.166,\ 0.040,\ 0.068,\ 0.052,\ 0.049,\ -0.071,\ 0.171,\ 0.074,\ -0.115,\ -0.001,\ -0.021,\ -0.012,\ -0.014,\ 0.063]^\top$$

with $n_p = 19$. We take $n = 17$ and polynomials have degree at most 68, which leads to 4624 possible solutions. In this case, we add noise with $\sigma = 0.1$, so that the cost function is most likely to have exactly one local minimum. The results are shown in Table 6.

Table 6
Performance of the methods in Example 2.

Method	Sec.	Solution	$f(R)$
SDP, u, 17	8.4	failed	failed
SDP, u, 24	46.5	failed	failed
SDP, b, 17	26.1	failed	failed
PHCPack	2883.9	(0.74802, 0.87727)	0.74087
SBDV	<3	(-0.55547, 0.63950)	0.07822
STLS	<0.01	(-0.55548, 0.63951)	0.07822

GloptiPoly fails to find the global minimum, reporting "numerical problems". PHCPack also fails to find the global minimum, possibly due to a large number of complex solutions. The Bernstein subdivision solver happens to find the solution with a good accuracy, as well as all stationary points in the box (see Fig. 2).

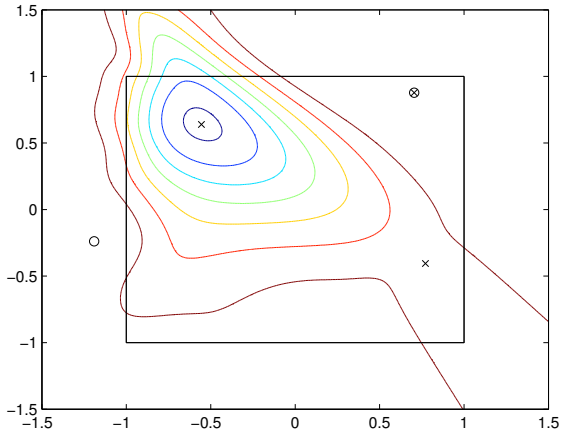

Fig. 2. Cost function for $p^{(2)}$ and stationary points found by SYNAPS (x) and PHCPack (o)

6 Conclusions

We have considered methods for global and local optimization of the structured low-rank approximation problem, based on the variable projection method. We have shown that the cost function is rational and can be minimized by polynomial algebra methods. Unfortunately, due to the high computational complexity of the problem, this approach can be used only for structured low-rank approximation problem of small dimension, say less than 20 structure parameters.

On the other hand, there are efficient methods for evaluation of the cost function and its derivatives, and therefore the cost function can be efficiently optimized locally. Our experiments show that the new optimization method that uses switching of coordinate charts is competitive with the state-of-the-art methods for local optimization on Grassmann manifold. In particular, an advantage of the proposed method is that, between switches, the problem solved is an unconstrained optimization problem (structured total least squares) and existing local optimization method for the structured total least squares subproblem can be reused. Our experiments show that the number of switches is typically low, however, there are no theoretical bounds for it.

Acknowledgements

The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC Grant agreement no. 258581 "Structured low-rank approximation: Theory, algorithms, and applications".

References

Absil, P.-A., R. Mahony and R. Sepulchre (2008). *Optimization Algorithms on Matrix Manifolds*. Princeton University Press. Princeton, NJ.

ApCoCoA (Applied Computations in Commutative Algebra) (n.d.). `http://www.apcocoa.org/`.

Boumal, N. and P.-A. Absil (2011). Rtrmc: A riemannian trust-region method for low-rank matrix completion. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira and K. Weinberger (Eds.). 'Advances in Neural Information Processing Systems 24'. pp. 406–414.

Boyd, S. and L. Vandenberghe (2004). *Convex Optimization*. Cambridge University Press.

De Moor, B., P. De Gersem, B. De Schutter and W. Favoreel (1997). 'Daisy : A database for identification of systems'. *Journal A (Benelux publication of the Belgian Federation of Automatic Control)*.

Dreesen, P., K. Batselier and B. De Moor (2012). Back to the roots: Polynomial system solving, linear algebra, systems theory. In 'Proceedings of 16th IFAC Symposium on System Identification'. pp. 1203–1208.

GenRTR (Generic Riemannian Trust-Region package) (n.d.). `http://www.math.fsu.edu/~cbaker/genrtr/`.

Guillaume, P. and R. Pintelon (1996). 'A Gauss–Newton-like optimization algorithm for "weighted" nonlinear least-squares problems'. *IEEE Trans. Signal Proc.* **44**(9), 2222–2228.

Hanzon, B. and D. Jibetean (2003). 'Global minimization of a multivariate polynomial using matrix methods'. *Journal of Global Optimization* **27**, 1–23.

Helmke, U. and J. B. Moore (1994). *Optimization and Dynamical Systems*. Springer.

Jibetean, D. and E. de Klerk (2006). 'Global optimization of rational functions: a semidefinite programming approach'. *Mathematical Programming* **106**, 93–109.

Knuth, D. E. (1985). 'Semi-optimal bases for linear dependencies'. *Linear and Multilinear Algebra* **17**, 1–4.

Maple™ (n.d.). `http://www.maplesoft.com/products/maple/`.

Markovsky, I. (2008). 'Structured low-rank approximation and its applications'. *Automatica* **44**(4), 891–909.

Markovsky, I. (2012). *Low Rank Approximation: Algorithms, Implementation, Applications*. Communications and Control Engineering. Springer.

Markovsky, I. and K. Usevich (2012*a*). Software for weighted structured low-rank approximation. (submitted).

Markovsky, I. and K. Usevich (2012*b*). Software for weighted structured low-rank approximation. Technical Report 339974. ECS, Univ. of Southampton. `http://eprints.soton.ac.uk/339974/`.

Markovsky, I. and K. Usevich (2013). 'Structured low-rank approximation with missing values'. *SIAM J. Matrix Anal. Appl.* (in review).

Markovsky, I., J. C. Willems, B. De Moor and S. Van Huffel (2006). *Exact and Approximate Modeling of Linear Systems: A Behavioral Approach*. number 11 In 'Monographs on Mathematical Modeling and Computation'. SIAM.

Marquardt, D. (1963). 'An algorithm for least-squares estimation of nonlinear parameters'. *SIAM J. Appl. Math.* **11**, 431–441.

Mehrmann, V. and F. Poloni (2012). 'Doubling algorithms with permuted lagrangian graph bases'. *SIAM J. Matrix Anal. Appl.* **33**(3), 780–805.

Mourrain, B. and J. P. Pavone (2009). 'Subdivision methods for solving polynomial equations'. *Journal of Symbolic Computation* **44**(3), 292–306.

Rouillier, F. (1999). 'Solving zero-dimensional systems through the rational univariate representation'. *Applicable Algebra in Engineering, Communication and Computing* **9**, 433–461.

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

Strang, G. (1988). *Linear Algebra and its Applications*. 3rd edn. Thomson Learning.

Sturmfels, B. (2002). *Solving Systems of Polynomial Equations*. number 97 In 'CBMS Regional Conferences Series'. American Mathematical Society.

SYNAPS (SYmbolic Numeric ApplicationS) (n.d.). `http://www-sop.inria.fr/galaad/software/synaps/`.

Usevich, K. and I. Markovsky (2012*a*). Structured low-rank approximation as a rational function minimization. In 'Proceedings of 16th IFAC Symposium on System Identification'. pp. 722–727.

Usevich, K. and I. Markovsky (2012*b*). Variable projection methods for affinely structured low-rank approximation in weighted 2-norm. (submitted).

Verschelde, J. (1999). 'Algorithm 795: PHCpack: a general-purpose solver for polynomial systems by homotopy continuation'. *ACM Transactions on Mathematical Software* **25**(2), 251–276.

A Structured total least squares problem

The constraint (3) is equivalent to

$$\begin{bmatrix} X & -I_d \end{bmatrix} \mathscr{S}(p) = 0 \quad \Longleftrightarrow \quad A_p X^\top = B_p,$$

where $A_p \in \mathbb{R}^{n\times r}$ and $B_p \in \mathbb{R}^{n\times(d)}$ are defined as

$$\begin{bmatrix} A_p & B_p \end{bmatrix}^\top := \mathscr{S}(p).$$

Then (4) is equivalent to

$$\underset{\widehat{p}\in\mathbb{R}^{np},X\in\mathbb{R}^{d\times r}}{\text{minimize}} \|\widehat{p}-p\|_2 \ \text{ subject to } \ A_{\widehat{p}}X = B_{\widehat{p}}.$$

Note 7 *Problem* (13) *is is a structured total least-squares problem for the structure* $\Pi\mathscr{S}(p)$.

Indeed,

$$\begin{bmatrix} X & -I_d \end{bmatrix} \Pi\mathscr{S}(p) = 0 \quad \Longleftrightarrow \quad AX^\top = B,$$

where the matrices $A \in \mathbb{R}^{n\times r}$ and $B \in \mathbb{R}^{n\times d}$ are defined by

$$\begin{bmatrix} A & B \end{bmatrix}^\top := \Pi\mathscr{S}(p).$$

B Proof of Lemma 3

First, note that if the differential of f is represented as

$$df(R,H) = \operatorname{tr}(AH^\top),$$

then $\nabla_{d\times m}(f) = A$. Then the differential is expresses as

$$\begin{aligned} dg(R,H) = \operatorname{tr}(RR^\top RH^\top + HR^\top RR^\top + RH^\top RR^\top + RR^\top HR^\top) \\ -2\operatorname{tr}(RH^\top + HR^\top) = 4\operatorname{tr}((RR^\top R - R)H^\top), \end{aligned}$$

end the first part of the lemma is proved.

The second part follows from expressing the second-order differential as

$$dg(R,H,E) = 4\,\mathrm{tr}((ER^\top R + RE^\top R + RR^\top E - E)H^\top),$$

which completes the proof. □