Optimization on a Grassmann manifold with application to system identification [⋆]

Konstantin Usevich [a], Ivan Markovsky [a]

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

Abstract

In this paper, we consider the problem of optimization of a cost function on a Grassmann manifold. This problem appears in system identification in the behavioral setting, which is a structured low-rank approximation problem. We develop an optimization approach based on switching coordinate charts. This method reduces the optimization problem on the manifold to an optimization problem in a bounded domain of an Euclidean space. We compare the proposed approach with state-of-the-art methods based on data-driven local coordinates and Riemannian geometry, and show the connections between the methods. Compared to the methods based on the local coordinates, the proposed approach allows to use arbitrary optimization methods for solving the corresponding subproblems in the Euclidean space.

Key words: system identification, data-driven local coordinates, Grassmann manifold, structured low-rank approximation, coordinate charts

1 Introduction

System identification problems are usually posed as minimization of a cost function

$$\widehat{\theta} = \arg\min_{\theta\in\Theta} f(\theta), \quad f(\theta) = f(\theta,u,y), \tag{1}$$

where u and y are the observed inputs and outputs of a system and the vector θ from $\Theta \subset \mathbb{R}^m$ parametrizes the considered model class. The model class is often over-parameterized. For example, consider a SISO system defined by

$$\begin{aligned} a_0u_0(t) + \cdots + a_{\ell_a}u_0(t+\ell_a) &= \\ b_0y_0(t) + \cdots + b_{\ell_b}y_0(t+\ell_b)&, \end{aligned} \tag{2}$$

with $\theta = \begin{bmatrix} a_0 \cdots a_{\ell_a}\, b_0 \cdots b_{\ell_b} \end{bmatrix}^\top \in \mathbb{R}^m \setminus \{0\}$, where $m = \ell_a + \ell_b + 2$. Then, for any $\lambda \neq 0$, θ and $\lambda\theta$ represent the same system. For many estimators (see (Pintelon *et al.* 1999)) for the model (2), the cost function (1) does not depend on scaling of θ, i.e.

$$f(\theta) = f(\lambda\theta). \tag{3}$$

Therefore, the parameter θ is often normalized in order to avoid the over-parameterization. Common choices are: fixing one coefficient in θ to 1 (Ljung 1999), or constraining the norm of the parameter vector, for example $\|\theta\|_2 = 1$. As shown in Pintelon *et al.* (1999), the statistical properties of such estimators do not depend on a particular normalization. The choice of the normalization, however, may influence the computational properties of the optimization methods. Several authors propose to use so-called *data-driven local coordinates* around the current iteration of the optimization procedure (McKelvey *et al.* 2004, Wills and Ninness 2008). The latter approach is used by default in the System Identification Toolbox of MATLAB (for state-space models).

In this paper, we consider optimization problems of the form:

$$\underset{R\in\mathscr{R}_{d,m}}{\text{minimize}}\ f(R), \tag{4}$$

where the argument is a $d \times m$ matrix, $d < m$, and $\mathscr{R}_{d,m} := \{R \in \mathbb{R}^{d\times m} : \operatorname{rank} R = d\}$ is the space of full-row rank matrices (equivalently, d-dimensional bases in $\mathbb{R}^m$). In addition, we require that f is *homogeneous of degree* 0, i.e.

$$f(R) = f(UR) \quad \text{for any nonsingular} \quad U \in \mathbb{R}^{d\times d}, \tag{5}$$

which means that $f(R)$ depends only on the row space of R, and does not depend on the choice of basis of this subspace. Therefore, f is defined on the *Grassmann manifold*

$$\mathrm{Gr}_{\mathbb{R}}(d,m) := \{\mathscr{L} \subset \mathbb{R}^m : \mathscr{L} \text{ is a } d\text{-dim. subspace of } \mathbb{R}^m\}.$$

[⋆] 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.

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

Optimization of a function on a Grassmann manifold appears in many contexts, see (Helmke and Moore 1994, Absil *et al.* 2008). A trivial example is optimization of functions of the type (3) (take $d = 1$ in this case). Our motivation for this paper is that the *structured low-rank approximation problem* (see Section 4) can be reduced to optimization on a Grassmann manifold (Usevich and Markovsky 2012). Structured low-rank approximation is a prototypical problem for many problems in systems and control (see (Markovsky 2008)): MIMO system identification, model reduction, approximate common divisor computations, etc.

As for the case of models (3), analogues of different normalizations exist (parameterizations of the Grassmann manifold). In this paper, we focus on parameterizations by coordinate charts, which can be bounded due to a remarkable result of Knuth (1985). Based on this result, we propose a new algorithm for optimization on the Grassmann manifold with switching coordinate charts (switching permutations). The idea of switching coordinate charts in the course of optimization was suggested in (Usevich and Markovsky 2012), but it was not implemented. Also, the bounds suggested in (Usevich and Markovsky 2012) were much larger than that of (Knuth 1985). Switching coordinate charts using results of (Knuth 1985) was also used in matrix pencil iterations for computing Lagrangian invariant subspaces (Mehrmann and Poloni 2012), but these authors do not consider the problem of optimization on a Grassmann manifold.

We compare the proposed method with the state of the art methods used in system identification, which are based on data-driven local coordinates. As a byproduct, we show that the latter methods are very closely connected to the Riemannian optimization methods on the Grassmann manifold (Absil *et al.* 2008). In particular, we show that existing Gauss-Newton/Levenberg-Marquardt methods in data-driven local coordinates can be employed (with a slight modification) for Riemannian optimization.

The paper is organized as follows. In Section 2 we make an overview of parameterizations of the Grassmann manifold, including bounded parameterizations in coordinate charts. In Section 3, we describe the main contribution of the paper: the proposed method of optimization in coordinate charts and correspondence between data-driven local coordinates and Riemannian optimization. In Section 4, we introduce structured low-rank approximation problem and show that it can be reduced to optimization on a Grassmann manifold. We also describe the MIMO identification framework using structured low-rank approximation in the behavioral setting. In Section 5, we compare the considered methods on examples of system identification.

2 Grassmann manifold and its parameterizations

In this section, we consider several parameterizations of $\mathrm{Gr}_{\mathbb{R}}(d,m)$. We use the term *parameterization* for a subset $\mathscr{M}$ of the set of full-row-rank matrices $\mathscr{R}_{d,m}$, which represents all (or almost all) d-dimensional subspaces. We provide examples of parameterizations (including the case $d = 1$) and also make a link to parameterizations used in system identification.

Parameterizations help to remove redundancy in the problem (4), if we replace $\mathscr{R}_{d,m}$ by its subset $\mathscr{M}$. Also, from the viewpoint of optimization, we prefer bounded (compact) parameterizations, because if the cost function is continuous, then it attains a global minimum on a compact set and the corresponding optimization problem is well-posed.

2.1 Orthonormal bases

For any subspace $\mathscr{L} \in \mathrm{Gr}_{\mathbb{R}}(d,m)$, there exists an orthonormal basis (a matrix R with orthonormal rows), hence $\mathrm{Gr}_{\mathbb{R}}(d,m)$ can be parameterized by

$$\mathscr{M}_{ort} := \{R \in \mathbb{R}^{d\times m} : RR^{\top} = I_d\}. \tag{6}$$

The parametrization (6) is ambiguous, because for an $R \in \mathscr{M}_{ort}$ and any orthogonal matrix $U \in \mathbb{R}^{d\times d}$ it holds that $UR \in \mathscr{M}_{ort}$ (a nonsingular basis transformation). In particular, a minimum of (4) on $\mathscr{M}_{ort}$ is not unique.

In the case $d = 1$, the parameterization $\mathscr{M}_{ort}$ becomes the set of $R \in \mathbb{R}^{1\times m}$ with $\|R\|_2 = 1$, and therefore $\mathscr{M}_{ort}$ is an $(m-1)$-sphere in $\mathbb{R}^m$. This corresponds to the normalization $\|\theta\|_2 = 1$ in the problem (1). In Fig. 1, $\mathscr{M}_{ort}$ is shown along with elements of $\mathrm{Gr}_{\mathbb{R}}(1,m)$, which correspond to lines passing through the origin. As seen from Fig. 1, the same subspace is represented by two antipodal points on $\mathscr{M}_{ort}$, for $d = 1$.

2.2 Coordinate charts

Suppose that $R = [Q\, P] \in \mathbb{R}^{d\times m}$ is of full row rank, such that $P \in \mathbb{R}^{d\times d}$ is nonsingular. The matrix $[X\, I_d]$ with $X = P^{-1}Q$ represents the same subspace. Then the set

$$\mathscr{M}_I := \{[X\, I_d] : X \in \mathbb{R}^{d\times(m-d)}\} \subset \mathbb{R}^{d\times m}, \tag{7}$$

parametrizes a part of the Grassmann manifold. This parameterization is not ambiguous: different points of $\mathscr{M}_I$ represent different subspaces (i.e. the map $R \in \mathscr{M}_I \mapsto \mathrm{rowspan}(R)$ is injective). However, $\mathscr{M}_I$ does not represent some subspaces in $\mathrm{Gr}_{\mathbb{R}}(d,m)$ (which cannot be represented by matrices of the form $[X\, I_d]$). It can be shown that these subspaces lie on a submanifold of $\mathrm{Gr}_{\mathbb{R}}(d,m)$ of smaller dimension.

For $d = 1$, $\mathscr{M}_I$ corresponds to an affine hyperplane in $\mathbb{R}^m$ that is defined by the equation $R_{1,m} = 1$ (see Fig. 1). The subspaces from $\mathrm{Gr}_{\mathbb{R}}(1,m)$ that cannot be represented by elements from $\mathscr{M}_I$ correspond to the vectors $R \in \mathbb{R}^{1\times m}$ with $R_{1,m} = 0$. In Fig. 1, this set corresponds to the dashed circle.

Note 1 *The parameterization $\mathscr{M}_I$ is implicitly used in total least squares and structured total least squares problems* *(see Appendix* (A)*). In system identification of models* (2)*, the set $\mathscr{M}_I$ corresponds to the normalization $\theta_m = 1$.*

In order to represent the whole $\mathrm{Gr}_{\mathbb{R}}(d,m)$ by sets similar to (7), recall that any full row rank matrix $R \in \mathbb{R}^{d\times m}$ has d linearly independent columns. Equivalently, by a permutation of columns we can make the last $d \times d$ block of R nonsingular. For a permutation matrix $\Pi \in \mathbb{R}^{m\times m}$ we define the set

$$\mathscr{M}_{\Pi} := \{[X\ I_d]\Pi : X \in \mathbb{R}^{d\times(m-d)}\} \subset \mathbb{R}^{d\times m}. \tag{8}$$

Then the collection of $\mathscr{M}_{\Pi}$ for all possible permutation matrices represent all subspaces from $\mathrm{Gr}_{\mathbb{R}}(d,m)$. For a fixed Π the set of subspaces corresponding to $\mathscr{M}_{\Pi}$ is called a *standard coordinate chart* of the Grassmann manifold (see (Helmke and Moore 1994, App. C)).

For $d = 1$, the set $\mathscr{M}_{\Pi}$ corresponds to affine hyperplanes in $\mathbb{R}^m$ that are defined by equations $R_{1,j} = 1$, $1 \le j \le m$. In Fig. 1, these hyperplanes pass through all sides of the hypercube that have normals in the positive orthant of $\mathbb{R}^m$.

2.3 Bounded representations in coordinate charts

The sets $\mathscr{M}_{\Pi}$ are not compact, and optimization problems in $\mathscr{M}_{\Pi}$ may suffer from ill-posedness (if there is no global minimum in $\mathscr{M}_{\Pi}$) or ill-conditioning (when the minimum is attained for large X). Therefore, we are interested in finding bounded subsets of $\mathscr{M}_{\Pi}$ that represent the whole $\mathrm{Gr}_{\mathbb{R}}(d,m)$.

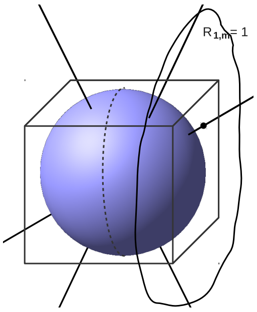

Fig. 1. Grassmann manifold $\mathrm{Gr}_{\mathbb{R}}(1,m)$ and its parameterizations. Black lines — elements of $\mathrm{Gr}_{\mathbb{R}}(1,m)$. The solid sphere — $\mathscr{M}_{ort}$. Hyperplane — $\mathscr{M}_I$. Dashed line — subspaces that cannot be represented by $\mathscr{M}_I$. Black cube — hypercube cut out by $\mathscr{M}_{\Pi}$.

In Fig. 1, one can see that the affine hyperplanes corresponding to $\mathscr{M}_{\Pi}$ (and their central symmetric counterparts) cut out the hypercube $[-1;1]^m$. Therefore, each subspace can be represented by a point on a face of the hypercube. Therefore, for $d = 1$ the manifold can be represented by a union of

$$\mathscr{M}_{\Pi,\square} := \{[X\ I_d]\Pi : X \in [-1;1]^{d\times(m-d)}\} \subset \mathbb{R}^{d\times m}.$$

A remarkable result of Knuth (1985) states that the same holds for $d > 1$.

Theorem 1 (Knuth (1985)) *For $d \ge 1$, any $\mathscr{L} \in \mathrm{Gr}_{\mathbb{R}}(d,m)$ can be represented as an element of $\mathscr{M}_{\Pi,\square}$ for some Π.*

The proof is based on properties of determinants. We note that in (Usevich and Markovsky 2012, Thm. 4), a different approach based on Gaussian elimination was applied, which lead to larger than $\mathscr{M}_{\Pi,\square}$ bounded sets.

3 Optimization on the Grassmann manifold

In this section, we consider methods for local optimization for the problem (4). Therefore, we require that the cost function is smooth on $\mathscr{R}_{d,m}$

3.1 Proposed method with switching permutations

By Theorem 1, the problem (4) is equivalent to

$$\underset{\Pi \text{ — perm.}}{\text{minimize}} \quad \min_{X\in[-1;1]^{d\times(m-d)}} f_{\Pi}(X). \tag{9}$$

where

$$f_{\Pi}(X) := f\big([X\ I_d]\Pi\big). \tag{10}$$

Each subproblem in (9) for a fixed Π is an unconstrained optimization problem on a compact domain, and is well-posed. However, the outer minimization in (9) involves choosing an optimal permutation matrix Π out of $\binom{m}{d}$ permutations.

For local optimization, the exhaustive search over all possible permutations, however, can be avoided by switching between permutations in the course of optimization. The following algorithm finds a locally optimal solution of (9).

Algorithm 1 *Input: f, initial guess $R_0 \in \mathscr{R}_{d,m}$, $\Delta > 1$.*
Output: $\widehat{R} \in \mathscr{R}_{d,m}$ yielding a local minimum of $f(R)$.

```
(1) Set $k = 0$.
(2) For $R_k$ choose $\Pi_k$ and $U_k \in \mathbb{R}^{d\times d}$, such that
        $U_k R_k = [X_k\ I_d]\Pi_k$   and   $X_k \in [-1;1]^{d\times(m-d)}$.
(3) Perform local optimization of $f_{\Pi_k}(X)$ until a convergence criterion is satisfied and unless $|(X)_{i,j}| < \Delta$ for all $i,j$. Set $R_{k+1} = [X_k\ I_d]\Pi_k$, where $X_k$ 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.
```

Algorithm 1 optimizes the cost function on the set $\mathscr{M}_{\Pi}$, and changes the permutation Π if the magnitude of the elements of the current iterate exceeds $\Delta \ge 1$. Choosing $\Delta > 1$ prevents the algorithm from switching near boundaries of $\mathscr{M}_{\Pi,\square}$ (edges of the hypercube for $d > 1$).

In order to complete the description of Algorithm 1, we need a procedure to reduce R_k to the form X_k at step 2. We use the algorithm (Mehrmann and Poloni 2012, Alg. 1) for updating a permutation Π_{k-1} to Π_k.

For $\Delta > 1$, the algorithm (Mehrmann and Poloni 2012, Alg. 1) completes in $\log_\Delta(M)$ steps, where $M = \max_{k,l}(X)_{k,l}$ (see (Knuth 1985) or (Mehrmann and Poloni 2012, Thm. 4.2)). This also justifies choosing the threshold in Algorithm 1 as $\Delta > 1$.

3.2 *Data-driven local coordinates and Riemannian optimization*

In system identification, a standard approach to optimization for overparametrized models is to use data-driven local coordinates (DDLC). Although, DDLC were initially introduced in (McKelvey *et al.* 2004) for state-space models, it was shown in (Wills and Ninness 2008) that an SVD-based modification of a standard Levenberg-Marquardt/Gauss-Newton iteration automatically observes the DDLC parameterization. Therefore, this approach became very popular in system identification (including nonlinear) (Van den Hof *et al.* 2009, Paduart 2010), since the SVD-based algorithm for DDLC does not require knowledge of the properties of the underlying manifold. Also, DDLC is used by default in the System Identification Toolbox of MATLAB (in prediction-error methods for state-space models).

On the other hand, the numerous works on optimization on manifolds (Absil *et al.* 2008) suggest considering the structure of the manifold and using the Riemannian metric to compute the local optimization step. In this section, we derive the DDLC parameterization for the Grassmann manifold and show that it is very closely related to the iterations that use Riemannian geometry (Absil *et al.* 2008). Thus we establish connections between DDLC and Riemannian optimization.

3.2.1 *DDLC representation for Levenberg-Marquardt and Gauss-Newton steps*

Assume that the cost function is of the form

$$f(\theta) = \|g(\theta)\|_2^2, \quad \text{where} \quad g : \mathbb{R}^{n_\theta} \to \mathbb{R}^{n_s}.$$

Let $J_g(\theta) \in \mathbb{R}^{n_s \times n_\theta}$ be the Jacobian of g at the point $\theta \in \mathbb{R}^{n_\theta}$. The standard Gauss-Newton/Levenberg-Marquardt step $\Delta\theta$ is defined as a solution of the normal equations

$$(J_g(\theta)^\top J_g(\theta) + \lambda I_{n_\theta})\Delta\theta = -J_g(\theta)^\top g(\theta), \qquad (11)$$

where $\lambda > 0$ in the Levenberg-Marquardt step and $\lambda = 0$ in the Gauss-Newton step.

For over-parametrized models, the Jacobian is rank-deficient. The rank of the Jacobian n_r corresponds to the number of degrees of freedom of the model (or the dimension of the underlying manifold), see (Wills and Ninness 2008). Therefore, for $\lambda = 0$ (the Gauss-Newton step), there are infinitely many solutions of (11). In the DDLC approach (Wills and Ninness 2008), the possible updates in (11) are restricted to

$$\Delta\theta = P\Delta\theta_r, \quad \Delta\theta_r \in \mathbb{R}^{n_r}, \qquad (12)$$

where $P^\top P = I_{n_r}$, and the column space of $P \in \mathbb{R}^{n_\theta \times n_r}$ contains the row space of J_θ. In this case, instead of function $g(\theta + \Delta\theta)$, the restricted function $h(\Delta\theta_r) := g(\theta + P\Delta\theta_r)$ is considered. Thus a DDLC update is defined as a solution of the reduced normal equations

$$(J_h(\theta)^\top J_h(\theta) + \lambda I)\Delta\theta_r = -J_h(\theta)^\top g(\theta), \qquad (13)$$

which is an equivalent of (11) for the function h. Also, note that the Jacobian of h is expressed as $J_h = J_g P$.

Consider $\Delta\theta = P\Delta\theta_r$, where $\Delta\theta_r$ is obtained from (13). In (Wills and Ninness 2008), it was shown that $\Delta\theta$ obtained in such a way coincides with the solution of the original normal equations (11) for $\lambda > 0$ (Levenberg-Marquardt step) and is the minimum norm solution of (11) for $\lambda = 0$ (Gauss-Newton step). Thus the original Levenberg-Marquardt step automatically observes data-driven local coordinates.

As shown in (Wills and Ninness 2008) both in the case of Levenberg-Marquardt ($\lambda > 0$) and Gauss-Newton ($\lambda = 0$), the DDLC step (12) can be computed as

$$\Delta\theta = -V_1(S_1^2 + \lambda I_{n_r})^{-1} S_1 U_1^\top g(\theta), \qquad (14)$$

where $J_g(\theta) = U_1 S_1 V_1^\top$ is the truncated SVD of $J_g(\theta)$ (i.e., $S_1 \in \mathbb{R}^{n_r \times n_r}$ is a diagonal matrix, and $n_r = \operatorname{rank} J_g(\theta)$). For the Gauss-Newton step, (14) becomes $\Delta\theta = J_g^\dagger(\theta) g(\theta)$. Thus the DDLC Gauss-Newton step can be interpreted as a “robustified” Gauss-Newton step.

3.2.2 *DDLC for the Grassmann manifold*

Consider the case of the Grassmann manifold (the function (4) with property (5)), with $\theta = \operatorname{vec} R$ and $n_r = (m-d)d$ (dimension of the manifold). Then from (5), we have that

$$g\left(\theta + (R^\top \otimes I_d)\operatorname{vec}(U - I_d)\right) = g(\theta),$$

for any nonsingular $U \in \mathbb{R}^{d \times d}$. Therefore, the right nullspace of the J_g contains the columns of the matrix $(R^\top \otimes I_d)$, and the row space of J_g is contained in the column span of

$$P = R_\perp \otimes I_d, \quad R_\perp \in \mathbb{R}^{m \times (m-d)}, \qquad (15)$$

where $R_\perp$ is an orthonormal basis of the right kernel of R, i.e. $R_\perp^\top R_\perp = I_d$ and $RR_\perp = 0$.

By (12), a DDLC update of R should be of the form $XR_\perp^\top$, where $X \in \mathbb{R}^{d \times (m-d)}$. Alternatively, we can rewrite this fact as follows. Let $\Phi = \begin{bmatrix} R_{\perp} & R^{\top} \end{bmatrix}^{\top}$. Then a DDLC step corresponds to an optimization step for the reduced cost function

$$f_{\Phi}(X) := f\left([X\; I_d]\Phi\right), \qquad (16)$$

Therefore, the optimization step in DDLC coordinates is equivalent to making one step of local optimization of f in the local parameterization

$$\mathscr{M}_{\Phi} := \{[X\; I_d]\Phi : X \in \mathbb{R}^{d\times(m-d)}\} \subset \mathbb{R}^{d\times m}. \qquad (17)$$

3.2.3 Riemannian optimization on manifolds

The general-purpose optimization methods on a Riemannian manifold $\mathscr{M}$ from (Absil *et al.* 2008) share a common scheme. From a current iterate x_k, a direction η_k is selected in the tangent space T_{x_k}, based on the derivatives of f at x_k. The new iterate is set to be $x_{k+1} = \mathfrak{R}_{x_k}(\eta_k)$ (compare with $x_{k+1} = x_k + \eta_k$ in optimization on $\mathbb{R}^n$), where $\mathfrak{R}_x : T_x \to \mathscr{M}$ is a *retraction*, which projects a direction to the manifold.

For a function $f = \|g\|_2^2$, $g \in \mathbb{R}^{n_s}$ a Gauss-Newton/Levenberg-Marquardt step (Absil *et al.* 2008, Ch. 8) is a solution η_k of

$$((Dg(x_k))^* \circ Dg(x_k) + \lambda\,\mathrm{id})[\eta_k] = -(Dg(x_k))^*[g(x_k)], \qquad (18)$$

where $Dg(x_k) : T_{x_k} \to R^{n_s}$ is the differential mapping, $(Dg(x_k))^* : R^{n_s} \to T_{x_k}$ is its adjoint, and $\mathrm{id} : T_{x_k} \to T_{x_k}$ is the identity map.

For the case of the Grassmann manifold, points on the manifold and directions in the tangent space have matrix representations (Absil *et al.* 2004), which turn (18) into a linear system of equations. A subspace $\mathscr{L} \in \mathrm{Gr}_{\mathbb{R}}(d,m)$ is represented as $R \in \mathscr{R}_{d,m}$, and the tangent space is represented by $T_R = \{XR_{\perp}^{\top} : X \in \mathbb{R}^{d\times(m-d)}\} \subset \mathbb{R}^{d\times m}$. The first-order derivatives of the functions are related as $Dg(R)[E] = J_g(R)\Pi_{T_R}E$, where Π_{T_R} is the projection on T_R. Using the fact that $\Pi_{T_R} = PP^{\top}$, where P is defined in (15), it is easy to see that (18) is equivalent (13). Thus, we have shown the following.

Note 2 *Finding a search direction in the Riemannian based approach is equivalent to finding a search direction with DDLC (or finding a search direction for the function* (16)*).*

Therefore, the DDLC update (14) is automatically suited for finding the search direction for the Levenberg-Marquardt/Gauss-Newton method in the Riemannian framework (for the Grassmann manifold). The only difference is that in the Riemannian framework (Absil *et al.* 2008) a retraction is used between the optimization steps (see (Absil *et al.* 2004)). Therefore, the Levenberg-Marquardt method with the step (14) can be easily adapted to the Riemannian framework (by introducing an intermediate retraction step). A retraction may be also beneficial for the DDLC algorithms, since without the retraction the magnitude of the rows of R grows and the local geometry changes.

4 Structured low-rank approximation

In this section we briefly review the structured low-rank approximation approach to MIMO system identification, and its reduction to optimization on a Grassmann manifold.

4.1 Structured low-rank approximation

A *matrix structure* $\mathscr{S}(p)$ is an linear matrix-valued function $\mathbb{R}^{n_p} \to \mathbb{R}^{m\times n}$, $m \le n$. Consider the *structured low-rank approximation* (SLRA) problem in the kernel representation.

Problem 2 (SLRA) *Given* $p \in \mathbb{R}^{n_p}$*, structure* $\mathscr{S}$*, weighted 2-norm* $\|\cdot\|_W$ *and natural number* $r < m$

$$\underset{\widehat{p}\in\mathbb{R}^{n_p},R\in\mathscr{R}_{(m-r),m}}{\mathit{minimize}} \quad \|\widehat{p}-p\|_W^2 \quad \textit{subject to} \quad R\mathscr{S}(\widehat{p}) = 0. \qquad (19)$$

In Problem 2, the constraint $R\mathscr{S}(\widehat{p}) = 0$, for $R \in \mathscr{R}_{(m-r),m}$ is equivalent to $\operatorname{rank}\mathscr{S}(\widehat{p}) \le r$. Therefore, Problem 2 is: for a given structured matrix $\mathscr{S}(p)$ find a low-rank structured matrix $\mathscr{S}(\widehat{p})$ which is the closest in the weighted 2-norm $\|\cdot\|_W$. See (Markovsky and Usevich 2014) for the original formulation of SLRA and its reduction to (19).

4.2 Optimization on a Grassmann manifold

Problem 2 can be split into *outer* and *inner minimization*

Problem 3

$$\underset{R\in\mathscr{R}_{(m-r),m}}{\mathit{minimize}} \; f(R), \qquad (20)$$

where

$$f(R) := \min_{\widehat{p}\in\mathbb{R}^{n_p}} \|\widehat{p}-p\|_W^2 \quad \textit{subject to} \quad R\mathscr{S}(\widehat{p}) = 0. \qquad (21)$$

The inner minimization problem (21) has a closed-form solution, and the derivatives of f also can also be evaluated (see (Markovsky and Usevich 2014)).

From (21), $f(R)$ depends only on the row space of R, and satisfies (5). Therefore, the outer minimization (20) is an instance of the problem (4).

4.3 System identification in the behavioral setting

In this subsection, we provide a simplified exposition of system identification of MIMO systems in the behavioral setting. A detailed description of this framework can be found in (Markovsky 2008, Markovsky 2013), and also in (Markovsky *et al.* 2006, Chap. 7,8,11).

Consider a q-variate time series $w : \mathbb{Z} \to \mathbb{R}^q$. We say that w is a *trajectory of a dynamical system with at most* $\mathtt{m}$ *inputs* *and lag at most* ℓ (denoted by $w \in L^q_{\mathtt{m},\ell}$) [1] , if and only if there exist $R_k \in \mathbb{R}^{(q-\mathtt{m})\times q}$, $0 \le k \le \ell$, such that

$$R_0 w(t) + R_1 w(t+1) + \cdots + R_\ell w(t+\ell) = 0, \quad \text{for all } t \quad (22)$$

and $R(z) = \sum_{k=0}^{\ell} R_k z^k$ is a full row rank polynomial matrix.

If $w \in L^q_{\mathtt{m},\ell}$, then there exists an input/output partition of variables $w(t) = \Pi \left[\begin{smallmatrix} u(t) \\ y(t) \end{smallmatrix}\right]$, $u(t) \in \mathbb{R}^{\mathtt{m}}$ and $y(t) \in \mathbb{R}^{q-\mathtt{m}}$, such that $u(t)$ and $y(t)$ are inputs and outputs of a MIMO linear time-invariant system in a state-space form (which explains the name for the set $L^q_{\mathtt{m},\ell}$). State-space parameters (A,B,C,D) of the system (or any other parameters) can be obtained from the coefficients R_k of the difference equation (22).

Example 4 *Consider the case* $q = 2$ *and* $\mathtt{m} = 1$ *(a SISO system), and assume that the input/output partition is known (i.e.* $w(t) = [u(t)\ y(t)]^\top$*). Denote the coefficients of the difference equation (*1×2 *matrices) as* $R_k = [a_k\ b_k]$*. Then the difference equation* (22) *becomes the equation* (2) *with* $\ell_a = \ell_b = \ell$.

We consider the following identification problem statement: for an observed trajectory find the closest trajectory in the model class.

Problem 5 *Given* $w \in \mathbb{R}^{q\times T}$, ℓ, $\mathtt{m}$, *a weight matrix* W

$$\underset{\widehat{w} \in \mathbb{R}^{q\times T}}{\textit{minimize}} \ \|w - \widehat{w}\|_W^2 \quad \textit{subject to} \quad \widehat{w} \in L^q_{\mathtt{m},\ell}.$$

Although Problem 5 is stated as deterministic approximation, it can represent different identification scenarios such as output error or errors-in-variables identification by proper choice of the weight matrix W, see (Markovsky 2008, Markovsky 2013) for more details.

4.4 Transition to structured low-rank approximation

For a time series $w = (w(1), \ldots, w(T))$, we define the *block-Hankel* matrix as

$$\mathscr{H}_L(w) = \begin{bmatrix} w(1) & w(2) & \cdots & w(T-L-1) \\ w(2) & w(3) & \cdots & w(T-L) \\ \vdots & \vdots & \vdots & \vdots \\ w(L) & \cdots & \cdots & w(T) \end{bmatrix} \in \mathbb{R}^{Lq\times(T-L-1)}.$$

Then for w the equation (22) can be rewritten as:

$$\begin{bmatrix} R_0 & \cdots & R_\ell \end{bmatrix} \mathscr{H}_{\ell+1}(w) = 0,$$

and we have that

$$w \in L^q_{\mathtt{m},\ell} \Rightarrow \operatorname{rank} \mathscr{H}_{\ell+1}(w) \le r = q\ell + \mathtt{m}.$$

Therefore, Problem 5 can be reformulated and solved as a structured low-rank approximation problem (Problem 2) for block-Hankel structure. In this case, also the coefficients of the difference equation (22) are recovered (from R in (19)), together with the approximating trajectory.

Splitting Problem 2 into inner and outer minimization has also system-theoretic interpretation. The inner minimization (20) corresponds to finding the closest trajectory to the observed one for given parameters R_k of the system, and thus is an equivalent of Kalman smoothing or calculating the prediction error in prediction-error methods. The outer minimization (20) is equivalent to optimization over all possible systems in the considered model class of systems.

5 Numerical experiments

In this section, we compare several methods for local optimization on the Grassmann manifold. The cost function to be optimized is (21), in examples of system identification for a time series $w \in \mathbb{R}^{q\times T}$, and the model class $L^q_{\mathtt{m},\ell}$.

All the tests are reproducible and can be found in the directory `grassopt` of the package (*SLRA* 2013).

5.1 Description of the methods

We compare five local optimization methods:

- `perm`, $\mathtt{perm}_0$ — optimization with Algorithm 1, $\mathtt{perm}_0$ — denotes optimization without switching of permutations (Π is fixed to I). Methods `perm` and $\mathtt{perm}_0$ use the Levenberg-Marquardt method (Marquardt 1963) for optimization of the function $f_\Pi(X)$. The threshold Δ for the `perm` method is chosen to be $\sqrt{2}$.
- `manopt` — the Riemannian trust-region method from (*Manopt* 2013) with default parameters for the Grassmann manifold and caching.
- `reg` — optimization of the penalized cost function
$$\underset{R \in \mathbb{R}^{d\times m}}{\text{minimize}} \ \widetilde{f}(R) + \gamma \|RR^\top - I_d\|_{\mathrm{F}}^2, \qquad (23)$$
which forces $R \in \mathscr{M}_{ort}$. Note that γ does not need to grow to ∞, see (Markovsky and Usevich 2013). The method `reg` uses the "Newton trust-region" method from MATLAB Optimization Toolbox. The derivatives of the penalty in (23) are calculated analytically (see Appendix B).
- `ddlc` — the Levenberg-Marquardt method in data-driven local coordinates.

All the methods use fast evaluation of $f(R)$ and its derivatives implemented in the `slra` package (*SLRA 2013)*, and

[1] In the standard behavioral terminology, the set of trajectories $L^q_{\mathtt{m},\ell}$ corresponds to the union of all behaviors with the number of inputs at most $\mathtt{m}$ and lag at most ℓ.

are started from the same (default) initial approximation, described in (Markovsky and Usevich 2014). For `ddlc`, `perm`$_0$ and `perm`, the SVD update (14) is used, and the rules of choosing the λ are as in (Pintelon and Schoukens 2012, p. 366). For `reg` and `manopt` the approximation $2J^\top J$ of the Hessian is supplied.

5.2 *Experiments on examples from DAISY*

We test the optimization methods on the benchmark problems from the DAISY database (De Moor *et al.* 1997). The examples include noisy trajectories of nonlinear dynamical systems. Each test example is represented as a row in Table 1, including the names of the test cases, model class hyperparameters and dimensions of the manifold

Table 1
Test cases for experiments with DAISY: names, sizes, model hyperparameters, and dimensions of $\mathrm{Gr}_{\mathbb{R}}(d,m)$

name	T	q	m	ℓ	m	d
erie_n20	57	7	5	1	14	2
destill_n30	90	8	5	1	16	3
heating_system	801	2	1	2	6	1
dryer	1,000	2	1	5	12	1
flutter	1,024	2	1	5	12	1

We run all the methods for all the test cases, with maximum number of iterations set to 200. We provide convergence plots in Fig. 2, where the computational time spent is plotted against the cost function value, for each of the test cases. The time instant for the 0-th iteration was artificially made the same for all the methods (in order to eliminate the time needed for initialization).

The results in Fig. 2 suggest that the method with switching permutations `perm` is competitive with the method `ddlc`. The method `perm` also in some cases performs better than the version with fixed `perm`$_0$, since the computations are less likely to be ill-conditioned. Also, we see that `ddlc` outperforms `manopt`, although they are based on similar ideas. This can be explained first by the fact that `ddlc` is implemented purely in C/C++, and for `manopt` there is an overhead of calling mex-functions from MATLAB. Also, `manopt` works with $2J^\top J$, which squares the condition number of the Jacobian J.

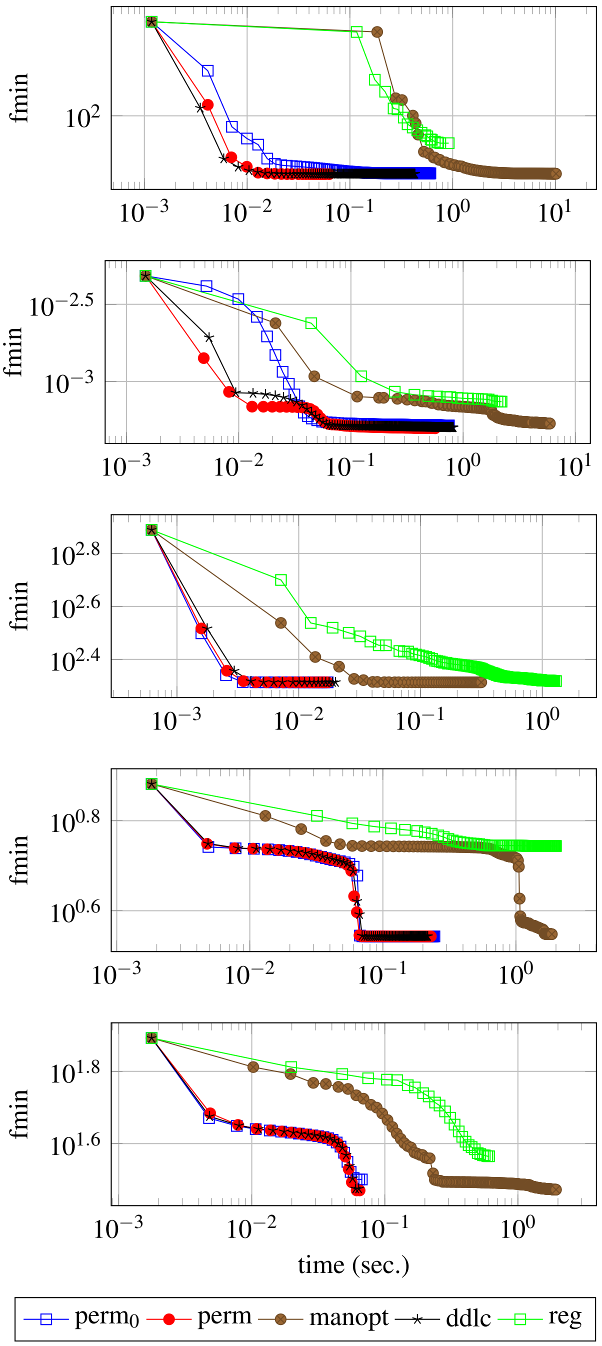

Fig. 2. Convergence of the algorithms (cost function vs. elapsed time). The individual panels correspond to the rows in Table 1.

6 Conclusions

Our experiments show that for the case of the Grassmann manifold, the proposed method with switching coordinate charts can be considered as an alternative to the state-of-the-art approach of data-driven local coordinates. Although there are no theoretical bounds on the number of switches, it is usually low in practice. We also showed that the data-driven local coordinates approach is closely related to optimization in Riemannian geometry, thus both communities (system identification and manifold optimization) may benefit from this connection. Also, in our opinion, an interesting research question is whether similar results (existence of bounded representations and equivalence between data-driven local coordinates and Riemannian optimization) hold for the state-space representation of systems or other over-parameterized models.

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". We also would like to thank Bamdev Mishra and three anonymous reviewers for their valuable suggestions.

A Structured total least squares problem

The *structured total least squares* problem (Markovsky *et al.* 2006, Ch. 4) is formulated as

$$\underset{\widehat{p}\in\mathbb{R}^{n_p},X\in\mathbb{R}^{d\times r}}{\text{minimize}} \ \|\widehat{p}-p\|_2^2 \ \text{ subject to } \ A_{\widehat{p}}X^\top = B_{\widehat{p}}, \qquad \text{(A.1)}$$

where $A_p \in \mathbb{R}^{n\times r}$, $B_p \in \mathbb{R}^{n\times d}$ and $[A_p\ B_p]^\top := \mathscr{S}(p)$ is a structured matrix (see Section 4). Since

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

the problem (A.1) is equivalent to (20) with the additional constraint $R \in \mathscr{M}_I$, i.e $R = [-X\ I_d]$, $X \in \mathbb{R}^{d\times r}$. Therefore, (A.1) is equivalent to

$$\underset{X\in\mathbb{R}^{d\times r}}{\text{minimize}} \ f_I(X), \qquad \text{(A.2)}$$

where f_I is f_Π from (10), defined for $\Pi = I$, and f from is defined in (21). Therefore, the function f_I is the cost function for the STLS problem (A.1). Moreover, the function f_Π for general Π is a cost function for a transformed STLS problem.

Note 3 *Each subproblem in* (9) *is equivalent to the problem* (A.2) *for the structure* $\Pi\mathscr{S}(p)$. *Minimization of* (16) *is equivalent to the problem* (A.2) *for the structure* $\Phi\mathscr{S}(p)$.

B Derivatives of the penalty term

For a function $g : \mathbb{R}^{d\times m} \to \mathbb{R}$ we denote by $\nabla_{d\times m} g$ the matrix gradient of g (i.e., $\operatorname{vec}(\nabla_{d\times m} g) := \nabla(\operatorname{vec} g)$), and by $H(g)$ the Hessian of g with respect to $\operatorname{vec} R$. Then for the function $g(R) = \|RR^\top - I_d\|_F^2$ and arbitrary $E \in \mathbb{R}^{d\times m}$ we have that

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

References

Absil, P.-A., R. Mahony and R. Sepulchre (2004). 'Riemannian geometry of grassmann manifolds with a view on algorithmic computation'. *Acta Applicandae Mathematica* **80**(2), 199–220.

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

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)*.

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

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

Ljung, L. (1999). *System identification*. Wiley Online Library.

Manopt (2013). http://manopt.org/. — a Matlab toolbox for optimization on manifolds.

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

Markovsky, I. (2013). 'A software package for system identification in the behavioral setting'. *Control Engineering Practice* **21**, 1422–1436.

Markovsky, I. and K. Usevich (2013). 'Structured low-rank approximation with missing data'. *SIAM Journal on Matrix Analysis and Applications* **34**(2), 814–830.

Markovsky, I. and K. Usevich (2014). 'Software for weighted structured low-rank approximation'. *Journal of Computational and Applied Mathematics* **256**, 278–292.

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.

McKelvey, T., Anders Helmersson and Thomas Ribarits (2004). 'Data driven local coordinates for multivariable linear systems and their application to system identification'. *Automatica* **40**(9), 1629 – 1635.

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

Paduart, J. (2010). Identification of Nonlinear Systems using Polynomial Nonlinear State Space Models. PhD thesis. Vrije Universiteit Brussel.

Pintelon, R. and J. Schoukens (2012). *System Identification: A Frequency Domain Approach*. 2nd edn. Wiley.

Pintelon, R., J Schoukens, G Vandersteen and Y Rolain (1999). 'Identification of invariants of (over) parameterized models: Finite sample results'. *IEEE Trans. on Aut. Control* **44**(5), 1073–1077.

SLRA (2013). http://github.com/slra/slra/. — package for structured low-rank approximation.

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

Van den Hof, P. M., J. F.M. Van Doren and S. G. Douma (2009). Identification of parameters in large scale physical model structures, for the purpose of model-based operations. In P. M. Hof, C. Scherer and P. S. Heuberger (Eds.). 'Model-Based Control:'. Springer US. pp. 125–143.

Wills, A. and B. Ninness (2008). 'On gradient-based search for multivariable system estimates'. *IEEE Transactions on Automatic Control* **53**(1), 298–306.