SIAM J. Matrix Anal. Appl.
Vol. 41, No. 3, pp. 1260–1283

A HYBRID PENALTY METHOD FOR A CLASS OF OPTIMIZATION PROBLEMS WITH MULTIPLE RANK CONSTRAINTS*

TIANXIANG LIU†, IVAN MARKOVSKY‡, TING KEI PONG§, AND AKIKO TAKEDA¶

Abstract. In this paper, we consider the problem of minimizing a smooth objective over multiple rank constraints on Hankel structured matrices. These kinds of problems arise in system identification, system theory, and signal processing, where the rank constraints are typically "hard constraints." To solve these problems, we propose a hybrid penalty method that combines a penalty method with a postprocessing scheme. Specifically, we solve the penalty subproblems until the penalty parameter reaches a given threshold, and then switch to a local alternating "pseudoprojection" method to further reduce constraint violation. Pseudoprojection is a generalization of the concept of projection. We show that a pseudoprojection onto a *single* low-rank Hankel structured matrix constraint can be computed efficiently by existing software such as SLRA [I. Markovsky and K. Usevich, *J. Comput. Appl. Math.*, 256 (2014), pp. 278–292], under mild assumptions. We also demonstrate how the penalty subproblems in the hybrid penalty method can be solved by pseudoprojection-based optimization methods, and then present some convergence results for our hybrid penalty method. Finally, the efficiency of our method is illustrated by numerical examples.

Key words. Hankel structure, system identification, hybrid penalty method, pseudoprojection

AMS subject classifications. 15B05, 90C30

DOI. 10.1137/19M1269919

1. Introduction. Many data modeling problems can be posed and solved as *structured low-rank approximation* problems, i.e., problems of approximating matrices by preserving the structure but reducing the rank [13]. The to-be-approximated matrices are constructed from data, and the model's complexity is related to the rank of the approximation—the lower the rank, the simpler the model. However, the simpler the model, the higher the approximation error. One way to deal with this fundamental trade-off between model complexity and model accuracy is to solve a sequence of low-rank approximation problems with increasing bounds on the rank.

*Received by the editors June 24, 2019; accepted for publication (in revised form) by D. Orban June 25, 2020; published electronically September 1, 2020.
https://doi.org/10.1137/19M1269919
Funding: The research of the first author is supported in part by JSPS KAKENHI grant 19H04069. The research of the second author has received funding from the European Research Council (ERC) under the European Union's Seventh Framework Programme (FP7/2007–2013)/ERC grant agreement 258581 "Structured low-rank approximation: Theory, algorithms, and applications" and Fond for Scientific Research Vlaanderen (FWO) projects G028015N "Decoupling multivariate polynomials in nonlinear system identification" and G090117N "Block-oriented nonlinear identification using Volterra series"; and Fonds de la Recherche Scientifique (FNRS) – FWO Vlaanderen under Excellence of Science (EOS) Project 30468160 "Structured low-rank matrix/tensor approximation: numerical optimization-based algorithms and applications." The research of the third author is supported in part by Hong Kong Research Grants Council PolyU153004/18p. The research of the last author is supported in part by JSPS KAKENHI grants 17H01699 and 19H04069.
†RIKEN Center for Advanced Intelligence Project, 1-4-1, Nihonbashi, Chuo-ku, Tokyo 103-0027, Japan (tianxiang.liu@riken.jp).
‡Department ELEC, Vrije Universiteit Brussel (VUB), Pleinlaan 2, 1050 Brussels, Belgium (imarkovs@vub.ac.be).
§Department of Applied Mathematics, Hong Kong Polytechnic University, Hong Kong, People's Republic of China (tk.pong@polyu.edu.hk).
¶Department of Creative Informatics, Graduate School of Information Science and Technology, University of Tokyo, Tokyo, Japan, and RIKEN Center for Advanced Intelligence Project, 1-4-1, Nihonbashi, Chuo-ku, Tokyo 103-0027, Japan (takeda@mist.i.u-tokyo.ac.jp, akiko.takeda@riken.jp).

A classical approach for estimating the bound on the rank involves solving the order estimation problem. For more information about the order estimation problem, see the overview paper [23].

In static linear data modeling problems, i.e., models defined by linear algebraic equations, the data matrices are *unstructured.* All spectral and Frobenius norm optimal unstructured low-rank approximations can be obtained from truncation of the singular value decomposition [4]. This result, known as the Eckart–Young–Mirsky theorem [2], is at the heart of dimensionality reduction methods in machine learning [22]. Unstructured low-rank approximation is equivalent to the principal component analysis in statistics and the total least squares in numerical linear algebra [15].

The object of system theory, control, and signal processing is dynamical models. In linear time-invariant data modeling problems, i.e., for models defined by linear constant-coefficient difference equations, the data matrix is *Hankel structured* [1, 6, 11, 17]. To see this, consider a system defined by the equation

$$p_0 y(t) + p_1 y(t+1) + \cdots + p_s y(t+s) = 0 \quad \text{for} \quad t = 1, \dots, T-s.$$

By definition, the time series $y = [y(1), \dots, y(T)]^\top \in \mathbb{R}^T$ is a trajectory of the system if

$$p\mathcal{H}_{s+1}(y) = 0,$$

where $p := [p_0 \; p_1 \; \cdots \; p_s] \neq 0$ is the parameter vector of the system and

$$\mathcal{H}_{s+1}(y) := \begin{bmatrix} y(1) & y(2) & y(3) & \cdots & y(T-s) \\ y(2) & y(3) & \iddots & & y(T-s+1) \\ y(3) & \iddots & & & \vdots \\ \vdots & & & & \\ y(s+1) & y(s+2) & \cdots & & y(T) \end{bmatrix}$$

is a Hankel matrix,[1] constructed from the time series. Therefore, $\operatorname{rank}(\mathcal{H}_{s+1}(y)) \leq s$. The resulting Hankel structured low-rank approximation problem does not admit an analytic solution in terms of the singular value decomposition. For this reason, numerous local optimization [12] as well as convex relaxation [3] methods are proposed for solving it.

In this paper, we consider a generalization of the Hankel structured low-rank approximation problem to *multiple rank constraints.* An application that motivates this generalization is the common dynamics estimation problem in multichannel signal processing [14, 16, 19]. Modeling each channel separately requires an individual rank constraint of a Hankel matrix in the optimization problem. Imposing the assumption that the channels have common dynamics then leads to an additional (coupling) rank constraint. The problem of common dynamics estimation is closely related to the problem of approximate common factor computation of multiple polynomials in computer algebra [7, 26]. Specifically, we consider the following optimization problem with multiple rank constraints:

[1]A Hankel matrix is usually defined to be a square matrix. However, here we use a more general definition which does not require a Hankel matrix to be square.

$$
\begin{aligned}
&\min_{y_1,\cdots,y_N\in\mathbb{R}^n} && f(y)\\
&\quad\text{s.t.} && \operatorname{rank}(\mathcal{H}_{r_i+1}(y_i)) \le r_i, \quad i=1,\dots,N,\\
& && \operatorname{rank}\left([\mathcal{H}_{r+1}(y_1)\ \mathcal{H}_{r+1}(y_2)\ \cdots\ \mathcal{H}_{r+1}(y_N)]\right) \le r,
\end{aligned} \tag{1.1}
$$

where $y = vec(y_1 \cdots y_N)$ (see section 2 for notation), r_i and r are positive integers satisfying $r_i \le r \le \lfloor \frac{n-1}{2} \rfloor$ $(i = 1, \dots, N)$, and f represents the loss function, which is nonnegative, level-bounded, and smooth with Lipschitz continuous gradient. For example, $f(y) = \frac{1}{2}\|y - \overline{y}\|^2$, where $\overline{y} \in \mathbb{R}^{Nn}$ is the noisy observation signal.

For constrained problems such as (1.1) with smooth objectives, a classical solution method is the gradient projection algorithm, whose iterations require projections onto the feasible set. However, the coupling structure of the last constraint in (1.1) makes projection onto the feasible set a challenging problem: indeed, even the projection onto the set defined by *each* single constraint in (1.1) does not admit a closed-form solution. Thus, variants of proximal gradient algorithms cannot be directly applied to solving (1.1). Fortunately, we can show that one can obtain a so-called pseudoprojection (see Definition 2.2) onto the set defined by *each* single constraint by some existing solvers such as SLRA [18], under mild assumptions.

Motivated by this, we adopt a penalty approach and construct penalty subproblems whose feasible regions are either $\mathbb{R}^n$ or defined by either the first N constraints or the last constraint in (1.1): the pseudoprojections are easy to compute in all these cases. We then propose an algorithm vNPG$_{\text{major}}$ for the penalty subproblems, making explicit use of the difference-of-convex (DC) structure of the penalty functions. The algorithm vNPG$_{\text{major}}$ is a variant of NPG$_{\text{major}}$ in [9, Algorithm 2] and is based on computing pseudoprojections, which can be done efficiently for the feasible region of the penalty subproblems.

While approximate solutions to (1.1) can now be obtained by our penalty method, such solutions are typically *not* feasible for (1.1). This is not ideal for applications such as system identification in which solution feasibility is an important concern [11]. Even though constraint violation can theoretically be reduced via solving a sequence of penalty subproblems with increasing weights in the penalty functions, in practice this strategy results in high computational cost and numerical instability. To resolve this issue, we shift to a postprocessing method after obtaining a moderately accurate solution by our penalty method. Specifically, starting from such a solution obtained from the penalty method, we apply an alternating pseudoprojection method, alternating between the set defined by the first N constraints in (1.1) and that defined by the last constraint there, to reduce constraint violation.

Our main contributions are highlighted as follows:

- We propose a hybrid penalty method (Algorithm 3.2) for solving (1.1): a penalty scheme allowing three different kinds of penalty subproblems, followed by an alternating pseudoprojection method for postprocessing. An algorithm, vNPG$_{\text{major}}$ (Algorithm 3.1), is proposed for the penalty subproblems.
- We prove some convergence results for the hybrid penalty method, including an error bound for the penalty method (Theorem 3.2) and the convergence rate for the alternating pseudoprojection method (Theorem 3.4).
- We demonstrate how a pseudoprojection can be obtained by the solver SLRA [18] in section 4, under mild assumptions.

The rest of this paper is organized as follows. In section 2, we introduce notation and some basic properties of Hankel operators. The hybrid penalty method and the corresponding convergence analysis are presented in section 3. In section 4, we demonstrate how to compute pseudoprojections. Numerical simulation results are presented in section 5. Finally, we give some concluding remarks in section 6.

2. Notation and preliminaries.

Throughout this paper, we let $\mathbb{R}^n$ denote the n-dimensional Euclidean space and $\|\cdot\|$ denote the Euclidean norm induced by vector inner product $\langle\cdot,\cdot\rangle$. For an $x \in \mathbb{R}^n$, we let $x(i)$ denote its ith entry. For vectors $y_1,\dots,y_N \in \mathbb{R}^n$, we let $vec(y_1 \cdots y_N) := [y_1^\top \cdots\ y_N^\top]^\top \in \mathbb{R}^{Nn}$. Given a matrix $A \in \mathbb{R}^{m\times n}$, we let $\|A\|_F$ denote its Frobenius norm, $\|A\|_2$ denote its spectral norm, $A^\top$ denote its transpose, and $A(i,j)$ denote its (i,j)th entry. For A, $B \in \mathbb{R}^{m\times n}$, we denote the matrix inner product by $\langle A, B\rangle := \sum_{i=1}^m \sum_{j=1}^n A(i,j)B(i,j)$. For a linear operator $\mathcal{A}$, we use $\mathcal{A}^*$, $\mathrm{Range}(\mathcal{A})$, and $\ker(\mathcal{A})$ to denote its adjoint, range, and kernel, respectively.

For an extended real-valued function $h : \mathbb{R}^n \to \mathbb{R}\cup\{\infty\}$, we say that h is proper if $\mathrm{dom}\, h := \{x : h(x) < \infty\} \neq \emptyset$ and is closed if it is lower semicontinuous. Following [21, Definition 8.3], for a proper closed function $h : \mathbb{R}^n \to \mathbb{R}\cup\{\infty\}$, the regular subdifferential of h at $y \in \mathrm{dom}\, h$ is defined as

$$\widehat{\partial} h(y) := \left\{ u : \liminf_{\substack{v\to y\\ v\neq y}} \frac{h(v) - h(y) - u^\top(v-y)}{\|v-y\|} \geq 0 \right\},$$

and the (limiting) subdifferential of h at $y \in \mathrm{dom}\, h$ is defined as

$$\partial h(y) := \{u : \exists u^t \to u, y^t \xrightarrow{h} y \text{ with } u^t \in \widehat{\partial} h(y^t) \text{ for each } t\},$$

where $y^t \xrightarrow{h} y$ means both $h(y^t) \to h(y)$ and $y^t \to y$. We say that $\overline{y}$ is a stationary point of h if $0 \in \partial h(\overline{y})$. It is known from [21, Theorem 10.1] that any local minimizer of h is a stationary point.

For a nonempty closed set $\Omega \subseteq \mathbb{R}^n$, we let δ_Ω denote the indicator function of Ω, which is zero in Ω and is infinity otherwise. The regular normal cone and (limiting) normal cone of Ω at $y \in \Omega$ are defined by $\widehat{N}_\Omega(y) := \widehat{\partial}\delta_\Omega(y)$, and $N_\Omega(y) := \partial\delta_\Omega(y)$, respectively. We use $\mathrm{dist}(x,\Omega)$ to denote the distance from an $x \in \mathbb{R}^n$ to Ω and $\mathcal{P}_\Omega(x)$ to denote the projection, i.e., $\mathrm{dist}(x,\Omega) := \inf_{y\in\Omega}\|x-y\|$ and $\mathcal{P}_\Omega(x) := \arg\min_{y\in\Omega}\|x-y\|$. For a nonempty closed set $\Omega \subseteq \mathbb{R}^{m\times n}$, the distance from an $X \in \mathbb{R}^{m\times n}$ to Ω and its projection are defined with respect to the Frobenius norm:

$$\mathrm{dist}(X,\Omega) := \inf_{Y\in\Omega}\|X-Y\|_F \quad\text{and}\quad \mathcal{P}_\Omega(X) := \operatorname*{arg\,min}_{Y\in\Omega}\|X-Y\|_F.$$

We next recall the definition of prox-regular sets; see [21, Exercise 13.31].

Definition 2.1 (prox-regular sets). *A closed set Ω is prox-regular at $\overline{x} \in \Omega$ for $\overline{v} \in N_\Omega(\overline{x})$ if there exist $\epsilon > 0$ and $\sigma \geq 0$ such that whenever $x \in \Omega$ and $v \in N_\Omega(x)$ with $\|x-\overline{x}\| < \epsilon$ and $\|v-\overline{v}\| < \epsilon$, it holds that*

$$\langle v, y-x\rangle \leq \frac{\sigma}{2}\|y-x\|^2 \text{ for all } y \in \Omega \text{ with } \|y-\overline{x}\| < \epsilon.$$

Furthermore, Ω is prox-regular at $\overline{x}$ if it is prox-regular at $\overline{x}$ for all $\overline{v} \in N_\Omega(\overline{x})$.

We now define the notion of pseudoprojection, which will be used in our subsequent discussions.

Definition 2.2 (pseudoprojection). *Let $\Omega \subseteq \mathbb{R}^n$ be a nonempty closed set, $u \in \Omega$, and $x \in \mathbb{R}^n$. The pseudoprojection $\mathcal{P}^s_\Omega(x;u)$ of x onto Ω with respect to u is the collection of all $y \in \Omega$ satisfying*

(a) (Stationarity) $x - y \in N_\Omega(y)$; *and*
(b) (Function value improvement) $\|y - x\| \le \|u - x\|$.

Notice that any element of the pseudoprojection is a stationary point of the corresponding projection problem, i.e., it is a stationary point of the function $w \mapsto \frac{1}{2}\|w - x\|^2 + \delta_\Omega(w)$. Also, each such element improves the function value of the corresponding projection problem relative to a given point $u \in \Omega$. Pseudoprojection onto a nonempty closed set is always nonempty; indeed, in view of [21, Example 6.16] and [21, Proposition 6.5], we have $\mathcal{P}_\Omega(x) \subseteq \mathcal{P}^s_\Omega(x; u)$ for all $x \in \mathbb{R}^n$ and all $u \in \Omega$. We will discuss how to obtain a pseudoprojection onto specific sets[2] in our applications in section 4.

For notational simplicity and for rewriting our problem conveniently with respect to the variable y, we define linear operators $\mathcal{L}_i : \mathbb{R}^{Nn} \to \mathbb{R}^{(r_i+1)\times(n-r_i)}$ $(i = 1, \ldots, N)$ and $\mathcal{L} : \mathbb{R}^{Nn} \to \mathbb{R}^{(r+1)\times N(n-r)}$ as

$$\begin{aligned} \mathcal{L}_i(y) &:= \mathcal{H}_{r_i+1}(y_i), \quad i = 1, \ldots, N, \\ \mathcal{L}(y) &:= [\mathcal{H}_{r+1}(y_1)\ \mathcal{H}_{r+1}(y_2)\ \cdots\ \mathcal{H}_{r+1}(y_N)], \end{aligned} \tag{2.1}$$

where $y = vec(y_1 \cdots y_N) \in \mathbb{R}^{Nn}$, and r_i $(i = 1, \ldots, N)$ and r are defined in (1.1). We now present some properties of the linear operators $\mathcal{H}_l(\cdot)$ and $\mathcal{L}^*$.

LEMMA 2.3. *For any* $Y \in \mathbb{R}^{(r+1)\times(n-r)}$,

$$\mathcal{H}^*_{r+1}(Y) = \left[Y(1,1) \cdots \overbrace{\sum_{i+j=k+1} Y(i,j)}^{\text{the } k\text{th element}} \cdots Y(r+1, n-r) \right]^\top \in \mathbb{R}^n.$$

LEMMA 2.4. *For any* $W_i \in \mathbb{R}^{(r+1)\times(n-r)}$, $i = 1, \ldots, N$, *it holds that*

$$\mathcal{L}^*[W_1\ W_2\ \cdots\ W_N] = vec\left(\mathcal{H}^*_{r+1}(W_1)\ \mathcal{H}^*_{r+1}(W_2)\ \cdots\ \mathcal{H}^*_{r+1}(W_N)\right).$$

Proof. Fix any $W_i \in \mathbb{R}^{(r+1)\times(n-r)}$, $i = 1, \ldots, N$. According to the definition of adjoint, for any $y = vec(y_1\ \cdots\ y_N) \in \mathbb{R}^{Nn}$, we have

$$\begin{aligned} \langle \mathcal{L}^*[W_1\ W_2\ \cdots\ W_N], y\rangle &= \langle [W_1\ W_2\ \cdots\ W_N], \mathcal{L}\left(vec\left(y_1\ \cdots\ y_N\right)\right)\rangle \\ &= \langle [W_1\ W_2\ \cdots\ W_N], [\mathcal{H}_{r+1}(y_1)\ \mathcal{H}_{r+1}(y_2)\ \cdots\ \mathcal{H}_{r+1}(y_N)]\rangle \\ &= \sum_{i=1}^N \langle W_i, \mathcal{H}_{r+1}(y_i)\rangle = \sum_{i=1}^N \left\langle \mathcal{H}^*_{r+1}(W_i), y_i\right\rangle. \end{aligned}$$

Then the conclusion follows from this and the arbitrariness of y. This completes the proof. ∎

3. A hybrid penalty method. Notice that there are multiple rank constraints in (1.1), making it difficult to compute the projection onto the feasible set. To handle these constraints, one intuitive idea is to use a penalty method to "reduce" the number of constraints. However, when feasibility is important (e.g., in applications such as

[2]With an abuse of terminology, we simply call an element of the pseudoprojection a pseudo-projection.

system identification [11]), we have to increase the weights in the penalty function to improve the feasibility of approximate solutions returned by penalty methods, which leads to high computational cost and numerical instability in practice. One way out would be to shift to a local refinement method after obtaining a moderately accurate solution by the penalty method.

Based on these intuitive ideas, our solution method will then consist of two stages: a penalty method, followed by a postprocessing scheme. We will describe the penalty method in section 3.1, the postprocessing scheme in section 3.2, and the hybrid penalty method and its convergence analysis in section 3.3.

3.1. Stage 1: A penalty method. To describe the penalty method, we first rewrite (1.1) as follows, using notation in (2.1):

$$\begin{aligned} \min_{y\in\mathbb{R}^{Nn}} \quad & f(y) \\ \text{s.t.} \quad & \operatorname{rank}(\mathcal{L}_i(y)) \le r_i, \quad i = 1,\ldots,N, \\ & \operatorname{rank}(\mathcal{L}(y)) \le r. \end{aligned}$$

This can be further equivalently rewritten as

$$\min_{y\in\mathbb{R}^{Nn}} F(y) := f(y) + \delta_\Omega(y) + \sum_{i=1}^{k} \delta_{C_i}(\mathcal{A}_i(y)), \tag{3.1}$$

with three ways of setting k, $\mathcal{A}_i$, Ω, and C_i:

- Variant I: $k = 1$, $\mathcal{A}_1 = \mathcal{L}$ and
$$\Omega = \{y : \operatorname{rank}(\mathcal{L}_i(y)) \le r_i, \quad i = 1,\ldots,N\}, \quad C_1 := \{Y : \operatorname{rank}(Y) \le r\}.$$
- Variant II: $k = N$, $\mathcal{A}_i = \mathcal{L}_i$ $(i = 1,\ldots,N)$ and
$$\Omega = \{y : \operatorname{rank}(\mathcal{L}(y)) \le r\}, \quad C_i = \{Y : \operatorname{rank}(Y) \le r_i\}, \quad i = 1,\ldots,N.$$
- Variant III: $k = N+1$, $\mathcal{A}_i = \mathcal{L}_i$ $(i = 1,\ldots,N)$, $\mathcal{A}_{N+1} = \mathcal{L}$ and
$$\Omega = \mathbb{R}^{Nn},\ C_i = \{Y\colon \operatorname{rank}(Y) \le r_i\},\ i = 1,\ldots,N,\ C_{N+1} = \{Y\colon \operatorname{rank}(Y) \le r\}.$$

Here, we present three variants for model (3.1), which correspond to three specific penalty schemes. In practice, which variant should be applied depends on the particular instance of the problem. Notice that for the above three variants, the projection onto C_i has a closed-form solution; see [5, Example 7.4.52]. On the other hand, although the projection onto Ω may not admit a closed-form solution, a stationary point as defined in Definition 2.2 of the associated projection problem can be approximately and efficiently obtained by some existing solvers such as SLRA [18], as we will show in section 4, under mild assumptions.

Now we are ready to describe our penalty method. We first replace the constraints $\mathcal{A}_i(y) \in C_i$ $(i = 1,\ldots,k)$ in (3.1) by a penalty for violating the constraints to obtain the auxiliary function

$$F_\lambda(y) = f(y) + \delta_\Omega(y) + \sum_{i=1}^{k} \frac{1}{2\lambda}\operatorname{dist}^2(\mathcal{A}_i(y), C_i), \tag{3.2}$$

where $\lambda > 0$ is the penalty parameter. Then we approximately minimize the auxiliary function $F_\lambda(y)$ and update y while decreasing λ.

Now we consider the subproblem of the penalty method, i.e., minimizing F_λ in (3.2) with fixed penalty parameter λ. Note that each term of the penalty function in

Algorithm 3.1 vNPG$_{\text{major}}$ for subproblem (minimizing (3.5)) of penalty method

Step 0. Choose $y^0 \in \Omega$, $L_{\max} > L_{\min} > 0$, $\tau > 1$, $c > 0$ and an integer $M \geq 0$. Specify a stopping criterion. Set $l = 0$.
Step 1. Pick any $\xi^l \in \sum_{i=1}^k \frac{1}{\lambda}\mathcal{A}_i^*(\mathcal{P}_{C_i}(\mathcal{A}_i(y^l)))$ and arbitrarily choose $L_l^0 \in [L_{\min}, L_{\max}]$. For $L_{l,i} = L_l^0 \tau^i$, $i = 0, 1, \ldots$, compute

$$u_i^l \in \mathcal{P}_\Omega^s \left(y^l - \frac{1}{L_{l,i}}(\nabla h(y^l) - \xi^l); y^l \right) \tag{3.3}$$

until some u_i^l satisfies

$$F_\lambda(u_i^l) \leq \max_{[l-M]_+ \leq j \leq l} F_\lambda(y^j) - \frac{c}{2}\|u_i^l - y^l\|^2. \tag{3.4}$$

Step 2. Let $\overline{L}_l = L_{l,i}$, $y^{l+1} = u_i^l$, and $l \leftarrow l + 1$. Go to Step 1 unless the stopping criterion is met.

(3.2) can be written as the Moreau envelope of indicator function $\delta_{C_i}(\cdot)$. Using the DC decomposition of the Moreau envelope as in [9, equation 6], we see that

$$\begin{aligned} F_\lambda(y) &= f(y) + \delta_\Omega(y) + \sum_{i=1}^k \inf_{Y_i} \left\{ \delta_{C_i}(Y_i) + \frac{1}{2\lambda}\|Y_i - \mathcal{A}_i(y)\|_F^2 \right\} \\ &= f(y) + \delta_\Omega(y) + \sum_{i=1}^k \left\{ \frac{1}{2\lambda}\|\mathcal{A}_i(y)\|_F^2 - \sup_{Y_i \in C_i} \left\{ \frac{1}{\lambda}\left\langle \mathcal{A}_i^*(Y_i), y \right\rangle - \frac{1}{2\lambda}\|Y_i\|_F^2 \right\} \right\} \\ &= \underbrace{f(y) + \sum_{i=1}^k \frac{1}{2\lambda}\|\mathcal{A}_i(y)\|_F^2}_{h(y)} + \delta_\Omega(y) - \underbrace{\sum_{i=1}^k \sup_{Y_i \in C_i} \left\{ \frac{1}{\lambda}\left\langle \mathcal{A}_i^*(Y_i), y \right\rangle - \frac{1}{2\lambda}\|Y_i\|_F^2 \right\}}_{g(y)}, \end{aligned} \tag{3.5}$$

where h is a smooth function with a Lipschitz continuous gradient whose Lipschitz continuity modulus depends on λ, and g is a convex function with

$$\sum_{i=1}^k \frac{1}{\lambda}\mathcal{A}_i^*\left(\mathcal{P}_{C_i}(\mathcal{A}_i(y))\right) \subseteq \partial g(y);$$

this inclusion follows from [9, equation 7] and will be used for constructing majorants when a variant of proximal gradient method is applied to minimizing F_λ. Recall that the projection onto C_i has a closed-form solution. Thus, for Variant III, in which $\Omega = \mathbb{R}^{Nn}$, F_λ can be minimized via NPG$_{\text{major}}$ in [9, Algorithm 2]. However, for Variants I and II, the projection onto Ω is not easy to compute. Fortunately, one can obtain a special stationary point for the corresponding projection problems via specific solvers: as we shall see in section 4, such a point belongs to the set of pseudoprojection (see Definition 2.2) under mild assumptions. Thus, we propose Algorithm 3.1 below as a variant of NPG$_{\text{major}}$, which we call vNPG$_{\text{major}}$, where we replace the projection in the subproblem by pseudoprojection.

The line-search loop stopping criterion (3.4) in Algorithm 3.1 is discussed in section 3.3, where it is shown that (3.4) is achieved after a finite number of iterations.

3.2. Stage 2: Postprocessing scheme. After we obtain an approximate solution by the penalty method, we shift to a postprocessing method. A natural and simple choice for postprocessing is the alternating projection method. Let

$$\begin{aligned} \Omega_1 &:= \left\{ y \in \mathbb{R}^{Nn} : \operatorname{rank}(\mathcal{L}_i(y)) \le r_i, \quad i = 1, \dots, N \right\}, \\ \Omega_2 &:= \{ y \in \mathbb{R}^{Nn} : \operatorname{rank}(\mathcal{L}(y)) \le r \}. \end{aligned} \tag{3.6}$$

In the classical alternating projection method, one has to find the global minimizers of the following problems in each iteration, for some $\widetilde{y}$:

$$\min_{y = vec(y_1 \cdots y_N) \in \mathbb{R}^{Nn}} \ \frac{1}{2}\|y - \widetilde{y}\|^2 \ \text{ s.t. } \ \operatorname{rank}(\mathcal{H}_{r_i+1}(y_i)) \le r_i, \ i = 1, \dots, N. \tag{3.7}$$

$$\min_{y = vec(y_1 \cdots y_N) \in \mathbb{R}^{Nn}} \ \frac{1}{2}\|y - \widetilde{y}\|^2 \ \text{ s.t. } \ \operatorname{rank}([\mathcal{H}_{r+1}(y_1) \cdots \mathcal{H}_{r+1}(y_N)]) \le r. \tag{3.8}$$

However, these problems are in general difficult to solve globally. Fortunately, as mentioned in section 3.1, we can obtain a point in the set of pseudoprojection efficiently, under mild assumptions. Thus, we adopt the following *alternating pseudoprojection method* for postprocessing: start at some $x^0 \in \Omega_2$ and $z^0 \in \Omega_1$, let

$$z^{t+1} \in \mathcal{P}^s_{\Omega_1}(x^t; z^t) \quad \text{and} \quad x^{t+1} \in \mathcal{P}^s_{\Omega_2}(z^{t+1}; x^t), \quad t = 0, 1, \dots. \tag{3.9}$$

We will discuss how to compute the pseudoprojections in (3.9) in detail in section 4. Notice that while the postprocessing (3.9) improves the feasibility of the solution, it may also increase the function value f at the solution. We will compare the function values before and after postprocessing in our numerical experiments in section 5.

3.3. Hybrid penalty method for (1.1) and convergence analysis. The hybrid penalty method for solving (1.1), which consists of the penalty method discussed in section 3.1 and the postprocessing method discussed in section 3.2, is presented as Algorithm 3.2.

Parameters in Step 0. In Algorithm 3.2, y^{feas} and y^0 can always be chosen as 0. In our numerical experiments in section 5, to take advantage of the given data, we choose $y^0 \in \Omega$ as a pseudoprojection of the noisy signal $\overline{y}$ onto Ω, obtained by calling SLRA in [15] with the default setting. The value of $\overline{\lambda}$ should be properly chosen. If $\overline{\lambda}$ is too large, the penalty method in Algorithm 3.2 will terminate prematurely and thus return a relatively bad approximate solution. On the other hand, if $\overline{\lambda}$ is too small, it will lead to high computational cost and numerical instability in the penalty method part. In section 5, we choose a $\overline{\lambda}$ to strike a balance between solution quality and computational cost of the penalty method.

For the rest of the section, we will analyze the convergence of the hybrid penalty method, including the convergence analysis for the penalty method in section 3.3.2 and the convergence rate for the postprocessing method in section 3.3.3. Before proceeding, we first show that the criteria (3.4) and (3.10) are achieved after a finite number of iterations.

3.3.1. Finite termination of (3.4) and (3.10). The following theorem is about the finite termination of the line-search criterion (3.4) and the termination criterion (3.10), i.e., they can be satisfied after finitely many inner iterations. The proof is similar to that in [9, Proposition 1].

THEOREM 3.1. *The line-search criterion* (3.4) *is achieved after a finite number of iterations. Moreover,* $\{\overline{L}_l\}$ *is bounded. Furthermore, the termination criterion* (3.10) *for Algorithm* 3.1 *is achieved after a finite number of iterations.*

Algorithm 3.2 A hybrid penalty method for (1.1)

Penalty method for (3.1)

Step 0. Pick two sequences of positive numbers with $\epsilon_t \downarrow 0$ and $\lambda_t \downarrow 0$, choose a $\overline{\lambda} \geq 0$, $y^{\rm feas} \in \Omega \cap \bigcap_{i=1}^k \mathcal{A}_i^{-1}(C_i)$ and $y^0 \in \Omega$. Set $t = 0$.

Step 1. If $F_{\lambda_t}(y^t) \leq F_{\lambda_t}(y^{\rm feas})$, set $y^{t,0} = y^t$. Else, set $y^{t,0} = y^{\rm feas}$.

Step 2. Approximately minimize F_{λ_t} by Algorithm 3.1, starting at $y^{t,0}$ and terminating at y^{t,l_t} when the following three conditions hold:

$$\begin{gathered}\|y^{t,l_t+1} - y^{t,l_t}\| \leq \epsilon_t, \qquad F_{\lambda_t}(y^{t,l_t}) \leq F_{\lambda_t}(y^{t,0}),\\ \mathrm{dist}\left(0, \nabla f(y^{t,l_t}) + N_\Omega(y^{t,l_t+1}) + \sum_{i=1}^k \frac{1}{\lambda_t}\mathcal{A}_i^*\big(\mathcal{A}_i(y^{t,l_t}) - \mathcal{P}_{C_i}(\mathcal{A}_i(y^{t,l_t}))\big)\right) \leq \epsilon_t.\end{gathered} \tag{3.10}$$

Step 3. Update $y^{t+1} = y^{t,l_t}$ and $t \leftarrow t+1$. If $\lambda_t < \overline{\lambda}$ and $\overline{\lambda} > 0$, go to Step 4; otherwise go to Step 1.

Postprocessing method involving the sets in (3.6)

Step 4. Let $x^0 \in \mathcal{P}^s_{\Omega_2}(y^{t+1}; 0)$ and $z^0 \in \mathcal{P}^s_{\Omega_1}(y^{t+1}; 0)$, use alternative pseudo-projection as follows until a termination criterion is met:

$$z^{t+1} \in \mathcal{P}^s_{\Omega_1}(x^t; z^t) \quad \text{and} \quad x^{t+1} \in \mathcal{P}^s_{\Omega_2}(z^{t+1}; x^t) \quad t = 0, 1, \ldots. \tag{3.11}$$

Proof. We start by discussing the line-search criterion. First, we observe from (3.3) and Definition 2.2 that

$$\left\| u_i^l - \left(y^l - \frac{1}{L_{l,i}}\left(\nabla h(y^l) - \xi^l\right)\right)\right\|^2 \leq \left\| y^l - \left(y^l - \frac{1}{L_{l,i}}\left(\nabla h(y^l) - \xi^l\right)\right)\right\|^2,$$

which is equivalent to

$$\left\langle \nabla h(y^l) - \xi^l, u_i^l - y^l \right\rangle \leq -\frac{L_{l,i}}{2}\|u_i^l - y^l\|^2. \tag{3.12}$$

Next, recall from the definition of ξ^l and [9, equation 7] that

$$\xi^l \in \sum_{i=1}^k \frac{1}{\lambda}\mathcal{A}_i^*\left(\mathcal{P}_{C_i}\left(\mathcal{A}_i(y^l)\right)\right) \subseteq \partial g(y^l). \tag{3.13}$$

Using (3.12) and (3.13) together with $u_i^l \in \Omega$, the L-smoothness of h and the convexity of g give (here, we let L denote the Lipschitz continuity modulus of ∇h)

$$\begin{aligned} F_\lambda(u_i^l) &= h(u_i^l) - g(u_i^l) \leq h(y^l) + \left\langle \nabla h(y^l), u_i^l - y^l\right\rangle + \frac{L}{2}\|u_i^l - y^l\|^2 - g(u_i^l)\\ &\leq h(y^l) + \left\langle \nabla h(y^l), u_i^l - y^l\right\rangle + \frac{L}{2}\|u_i^l - y^l\|^2 - g(y^l) - \left\langle \xi^l, u_i^l - y^l\right\rangle\\ &= F_\lambda(y^l) + \left\langle \nabla h(y^l) - \xi^l, u_i^l - y^l\right\rangle + \frac{L}{2}\|u_i^l - y^l\|^2 \leq F_\lambda(y^l) + \frac{L - L_{l,i}}{2}\|u_i^l - y^l\|^2. \end{aligned}$$

Thus, we see that (3.4) is satisfied whenever $L_{l,i} \geq L + c$. From the definition of $L_{l,i}$, this latter inequality must hold when i satisfies $\tau^i L_{\min} \geq L + c$. Thus, the inequality must hold when $i \geq \tilde{i} := \max\{\lceil \frac{\log(L+c)-\log(L_{\min})}{\log\tau} \rceil, 1\}$, implying that the line-search criterion (3.4) is achieved after a finite number of iterations. Moreover, we have $\overline{L}_l \leq \tau^{\tilde{i}} L_{\max}$ for all l, which proves the boundedness of $\{\overline{L}_l\}$.

Next, let $\{y^l\}$ be generated by Algorithm 3.1 starting at a $y^{t,0}$ in Step 2 of Algorithm 3.2. We show that the termination criteria (3.10) hold after finitely many iterations in Algorithm 3.1 (with y^l in place of y^{t,l_t} and y^{l+1} in place of y^{t,l_t+1} in (3.10)). First, from (3.4), it is easy to see that the second inequality in (3.10) holds. Moreover, using a similar line of arguments as in [27, Lemma 4], we can show that

$$\lim_{l\to\infty} \|y^{l+1} - y^l\| = 0. \tag{3.14}$$

Thus, the first inequality in (3.10) also holds after a finite number of iterations in Algorithm 3.1. Finally, we note from (3.3) and Definition 2.2 that

$$y^l - \frac{1}{\overline{L}_l}(\nabla h(y^l) - \xi^l) - y^{l+1} \in N_\Omega(y^{l+1}).$$

Using this together with the definition of h in (3.5), we further obtain

$$\overline{L}_l(y^l - y^{l+1}) - \nabla f(y^l) - \sum_{i=1}^k \frac{1}{\lambda}\mathcal{A}_i^*\mathcal{A}_i(y^l) + \xi^l \in N_\Omega(y^{l+1}).$$

Combining this relation with (3.13) gives

$$\operatorname{dist}\left(0, \nabla f(y^l) + N_\Omega(y^{l+1}) + \sum_{i=1}^k \frac{1}{\lambda}\mathcal{A}_i^*\left(\mathcal{A}_i(y^l) - \mathcal{P}_{C_i}(\mathcal{A}_i(y^l))\right)\right) \leq \overline{L}_l\|y^{l+1} - y^l\|.$$

This inequality together with (3.14) and the boundedness of $\{\overline{L}_l\}$ shows that the third inequality in (3.10) holds after a finite number of iterations. This completes the proof. ∎

3.3.2. Convergence analysis for the penalty method in Algorithm 3.2. Notice that when $\overline{\lambda} = 0$, the penalty method in Algorithm 3.2 is exactly the same as [9, Algorithm 1]. Thus, we know from [9, Theorem 2] that the sequence $\{y^t\}$ is bounded and that any accumulation point of $\{y^t\}$, say, y^*, is a stationary point of (3.1) under the following classical constraint qualification:

$$\begin{aligned} &x_0 + \sum_{i=1}^k \mathcal{A}_i^*(x_i) = 0 \;\text{ and }\; x_0 \in N_\Omega(y^*), \quad x_i \in N_{C_i}(\mathcal{A}_i(y^*)) \text{ for } i = 1, \ldots, k \\ &\qquad\qquad \Longrightarrow\; x_i = 0 \;\text{ for } i = 0, \ldots, k. \end{aligned}$$

We next estimate the violation of the constraints for the solution given by the penalty method in Algorithm 3.2 in the following theorem. It implies that the constraint violation can be suppressed by terminating the algorithm at a small λ_t.

THEOREM 3.2. *Let $\{y^t\}$ be the sequence generated by the penalty method in Algorithm 3.2 for solving (3.1). Then we have for $t \geq 1$ and $i = 1, \ldots, k$ that*

$$\operatorname{dist}\left(\mathcal{A}_i(y^t), C_i\right) \leq \sqrt{2\lambda_{t-1} f(y^{\mathrm{feas}})}.$$

Proof. Note from the nonnegativity of f, the definition of y^t, the second inequality in (3.10), and the choice of $y^{t,0}$ and $y^{\rm feas}$ that for $i = 1, \dots, k$,

$$\begin{aligned}\frac{1}{2\lambda_{t-1}}\text{dist}^2(\mathcal{A}_i(y^t), C_i) &\le F_{\lambda_{t-1}}(y^t) = F_{\lambda_{t-1}}(y^{t-1,l_{t-1}})\\ &\le F_{\lambda_{t-1}}(y^{t-1,0}) \le F_{\lambda_{t-1}}(y^{\rm feas}) = f(y^{\rm feas}).\end{aligned}$$

This completes the proof. ∎

3.3.3. Convergence analysis of the postprocessing method in Algorithm 3.2. First, we present the following theorem which will be used later for the convergence analysis of the postprocessing method in Algorithm 3.2.

THEOREM 3.3. *Let Ω_2 be defined as in* (3.6). *Then Ω_2 is prox-regular at any $\overline{y} \in \Omega_2$ that satisfies* $\text{rank}(\mathcal{L}(\overline{y})) = r$.

Proof. First, we can rewrite Ω_2 as

$$\Omega_2 = \{y \in \mathbb{R}^{Nn} : \mathcal{L}(y) \in C\} \ \text{ with } \ C := \{Y \in \mathbb{R}^{(r+1)\times N(n-r)} : \text{rank}(Y) \le r\}.$$

By [20, Corollary 2.3], we see that Ω_2 is prox-regular at $\overline{y} \in \Omega_2$ if the following conditions hold:

(a) there is no $z \neq 0$ in $N_C(\mathcal{L}(\overline{y}))$ with $\mathcal{L}^* z = 0$;

(b) for every $\overline{v} \in N_{\Omega_2}(\overline{y})$, the set C is prox-regular at $\mathcal{L}(\overline{y})$ for every $z \in N_C(\mathcal{L}(\overline{y}))$ with $\mathcal{L}^* z = \overline{v}$.

We will prove that the above two statements hold. First, we prove (a). Using $\text{rank}(\mathcal{L}(\overline{y})) = r$ and noting that by assumption, we have $r \le \frac{n-1}{2}$ and hence $N(n-r) \ge r+1$, we see from [10, Proposition 3.6] that

$$N_C(\mathcal{L}(\overline{y})) = \Big\{W : [\ker(W)]^\perp \cap [\ker(\mathcal{L}(\overline{y}))]^\perp = \{0\} \text{ and } \text{rank}(W) \le 1\Big\}. \tag{3.15}$$

On the other hand, we see from Lemma 2.4 that for any $W = [W_1\ W_2\ \cdots\ W_N]$ with $W_\ell \in \mathbb{R}^{(r+1)\times(n-r)}$ ($\ell = 1, \dots, N$), we have

$$\mathcal{L}^*[W_1\ W_2\ \cdots\ W_N] = vec\left(\mathcal{H}^*_{r+1}(W_1)\ \mathcal{H}^*_{r+1}(W_2)\ \cdots\ \mathcal{H}^*_{r+1}(W_N)\right). \tag{3.16}$$

Suppose that there exists some $\widehat{W} = [\widehat{W}_1\ \cdots\ \widehat{W}_N] \in N_C(\mathcal{L}(\overline{y})) \cap \ker(\mathcal{L}^*)$ with $\widehat{W}_\ell \in \mathbb{R}^{(r+1)\times(n-r)}$ ($\ell = 1, \dots, N$). We then know from (3.15) and (3.16) that

$$\text{rank}(\widehat{W}) \le 1 \ \text{ and } \ \mathcal{H}^*_{r+1}(\widehat{W}_\ell) = 0 \ \text{ for all } \ell = 1, \dots, N. \tag{3.17}$$

Now we fix any ℓ. Note from (3.17) and Lemma 2.3 that

$$\text{rank}(\widehat{W}_\ell) \le 1, \qquad \sum_{i+j=k+1} \widehat{W}_\ell(i,j) = 0, \ \text{ for any } \ k = 1, \dots, n. \tag{3.18}$$

We claim that $\widehat{W}_\ell = 0$. To prove this, we establish the following equivalent statement. For each $k = 1, \dots, n$, all elements in the following set equal 0:

$$S_k := \Big\{\widehat{W}_\ell(i,j) : i + j = k + 1\Big\}.$$

First, it is easy to see from the equality in (3.18) that all elements in S_1 and S_n are zero. Now we prove that every element in S_k is zero by induction for each $k = 1, 2, \dots, n-1$.

Suppose that there exists some $K \geq 1$ so that every element in $\bigcup_{\ell=1}^{K} S_\ell$ is zero. Let $\widehat{W}_\ell(\bar{i}, \bar{j})$ and $\widehat{W}_\ell(\widehat{i}, \widehat{j})$ be any two elements in S_{K+1} with $\bar{i} < \widehat{i}$. We then know from the first inequality in (3.18) that the 2×2 submatrix formed by $\widehat{W}_\ell(\bar{i}, \widehat{j})$, $\widehat{W}_\ell(\bar{i}, \bar{j})$, $\widehat{W}_\ell(\widehat{i}, \widehat{j})$, and $\widehat{W}_\ell(\widehat{i}, \bar{j})$ is singular. Since $\bar{i} + \widehat{j} < \widehat{i} + \widehat{j} = K + 2$, we conclude that $\widehat{W}_\ell(\bar{i}, \widehat{j}) = 0$ by the induction hypothesis. Consequently, there is at least one 0 in $\{\widehat{W}_\ell(\bar{i}, \bar{j}), \widehat{W}_\ell(\widehat{i}, \widehat{j})\}$. By the arbitrariness of these two elements in S_{K+1}, we see that there is at most one nonzero element in S_{K+1}. This together with the equality in (3.18) implies that every element in S_{K+1} equals 0. Thus, we have $\widehat{W}_\ell = 0$ by induction. Since ℓ is arbitrary, we see further that $\widehat{W} = 0$. This proves that $N_C(\mathcal{L}(\bar{y})) \cap \ker(\mathcal{L}^*) = \{0\}$, which is equivalent to statement (a).

Now we prove (b). Using $\operatorname{rank}(\mathcal{L}(\bar{y})) = r$, we know from [10, Proposition 3.8] that C is prox-regular at $\mathcal{L}(\bar{y})$. Then by the definition of prox-regularity, we see that (b) holds. This completes the proof. ∎

Since (3.11) involves the pseudoprojection instead of the actual projection, the postprocessing method in Algorithm 3.2 is different from the classical alternating projection method. Nevertheless, we can still show that the postprocessing method in Algorithm 3.2 has local linear convergence under commonly used assumptions for establishing local linear convergence of the alternating projection method (see, for example, the assumptions used in [8, Theorem 5.16] and [10, Theorem 4.2]). The proof follows the same line of arguments as in [8, Theorem 5.2]. We include the proof in the appendix for the convenience of the readers.

THEOREM 3.4. *Let Ω_1 and Ω_2 be defined as in* (3.6) *and suppose that there exists some $\bar{y} \in \Omega_1 \cap \Omega_2$ such that* $\operatorname{rank}(\mathcal{L}(\bar{y})) = r$ *and* $N_{\Omega_1}(\bar{y}) \cap -N_{\Omega_2}(\bar{y}) = \{0\}$. *Then for any initial points $x^0 \in \Omega_2$ and $z^0 \in \Omega_1$ near $\bar{y}$, any sequence generated by the following iterations converges to a point in $\Omega_1 \cap \Omega_2$ R-linearly:*

$$z^{t+1} \in \mathcal{P}^s_{\Omega_1}(x^t; z^t) \quad \text{and} \quad x^{t+1} \in \mathcal{P}^s_{\Omega_2}(z^{t+1}; x^t), \quad t = 0, 1, \ldots. \tag{3.19}$$

4. Subproblem: Pseudoprojection.

In this section, we consider the pseudoprojection subproblems (3.3) in Algorithm 3.1 and (3.11) in Algorithm 3.2. Recall that their corresponding *projection* problems can be put in the general form

$$\min_{y \in \mathbb{R}^d} \ \frac{1}{2}\|y - \widehat{y}\|^2 \ \text{ s.t. } \ \operatorname{rank}(\mathcal{A}(y)) \leq m; \tag{4.1}$$

here, $\mathcal{A}(y) \in \mathbb{R}^{p \times q}$, and d, m, p, q, and $\mathcal{A}$ are given as in (4.2) or (4.3) below, corresponding to (3.7) and (3.8), respectively:

(4.2) $d = n$, $m = r_i$, $p = r_i + 1$, $q = n - r_i$, $\mathcal{A}(y) = \mathcal{H}_{r_i+1}(y)$.

(4.3) $d = Nn$, $m = r$, $p = r + 1$, $q = N(n - r)$, $\mathcal{A}(y) = [\mathcal{H}_{r+1}(y_1) \cdots \mathcal{H}_{r+1}(y_N)]$.

The pseudoprojection problem corresponding to (4.1) can now be stated as follows: given $\widehat{y} \in \mathbb{R}^d$ and some reference point $y_b \in \mathbb{R}^d$ satisfying $\operatorname{rank}(\mathcal{A}(y_b)) \leq m$, compute

$$y_s \in \mathcal{P}^s_{\{y:\, \operatorname{rank}(\mathcal{A}(y)) \leq m\}}(\widehat{y}; y_b).$$

In what follows, we will describe how such a y_s can be obtained by the solver SLRA in [18]. Recall that SLRA was developed based on the following key observation:

$$\operatorname{rank}(\mathcal{A}(y)) \leq m \iff \exists \text{ full row-rank matrix } R \in \mathbb{R}^{(p-m) \times p} \text{ such that } R\mathcal{A}(y) = 0.$$

In view of this, algorithms were developed in [18] to approximately solve the following equivalent formulation of (4.1):

$$\min_{R\in\mathbb{R}^{(p-m)\times p}} \ \Psi(R) \ \text{ s.t. } \ RR^\top = I, \tag{4.4}$$

where

$$\Psi(R) := \inf_{y\in\mathbb{R}^d}\left\{\frac{1}{2}\|y-\widehat{y}\|^2 : \ R\mathcal{A}(y) = 0\right\}. \tag{4.5}$$

Notice that under the settings in (4.2) or (4.3), we have $p-m=1$ and hence (4.4) is an optimization problem in $\mathbb{R}^{1\times p}$ and the feasible set reduces to $\{R\in\mathbb{R}^{1\times p} : RR^T = 1\}$. We will show below in section 4.1 that Ψ in (4.5) is smooth on $\mathbb{R}^{1\times p}\backslash\{0\}$. The function value and the gradient of Ψ can be obtained as in [25, equation $(y^\top s)$] and [25, equation $(\nabla_{d\times m})$], respectively. Thus, when gradient-based optimization methods such as those described in [18] are applied to solving (4.4), one obtains a stationary point of the following function:

$$\widetilde{\Psi}(R) := \Psi(R) + \delta_\Theta(R), \text{ where } \Theta := \{R\in\mathbb{R}^{1\times p} : \ RR^T = 1\}. \tag{4.6}$$

We will then discuss in section 4.2 how an element of $\mathcal{P}^s_{\{y:\,\mathrm{rank}(\mathcal{A}(y))\le m\}}(\widehat{y}; y_b)$ can be obtained from such a stationary point under mild assumptions.

4.1. Smoothness of Ψ. In this subsection, we will prove that Ψ is smooth on $\mathbb{R}^{1\times p}\backslash\{0\}$. We start with an auxiliary lemma.

LEMMA 4.1. *Consider* (4.1) *with setting* (4.2) *or* (4.3)*. For any $U\in\mathbb{R}^{1\times q}$ and any $R\in\mathbb{R}^{1\times p}\backslash\{0\}$, if $\mathcal{A}^*(R^\top U) = 0$, then $U = 0$.*

Proof. Assume that $U\in\mathbb{R}^{1\times q}$ and $R\in\mathbb{R}^{1\times p}\backslash\{0\}$ satisfy $\mathcal{A}^*(R^\top U) = 0$. We need to show that $U = 0$.

We first consider (4.1) with setting (4.2). In this case, we have $m = r_i$, $p = r_i+1$, $q = n-r_i$, and $\mathcal{A}(y) = \mathcal{H}_{r_i+1}(y)$. Notice that $R^\top\in\mathbb{R}^{p\times 1} = \mathbb{R}^{r_i+1}$ and $U^\top\in\mathbb{R}^{q\times 1} = \mathbb{R}^{n-r_i}$. Write

$$R = [\,R(1),\ldots,R(r_i+1)\,], \qquad U = [\,U(1),\ldots,U(n-r_i)\,],$$

and $W = R^\top U$. Using Lemma 2.3, we obtain

$$\mathcal{A}^*(R^\top U) = \mathcal{H}^*_{r_i+1}(W) = \begin{bmatrix}\cdots \overbrace{\sum_{s+t=k+1} W(s,t)}^{\text{the } k\text{th element}} \cdots\end{bmatrix}^\top = \begin{bmatrix}\cdots \overbrace{\sum_{s+t=k+1} R(s)U(t)}^{\text{the } k\text{th element}} \cdots\end{bmatrix}^\top$$

$$= \underbrace{\begin{bmatrix} R(1) & R(2) & \cdots & R(r_i+1) & & & & \\ & R(1) & \cdots & R(r_i) & R(r_i+1) & & & \\ & & \ddots & \ddots & & \ddots & & \\ & & & & & R(1) & \cdots & R(r_i+1)\end{bmatrix}^\top}_{\widehat{R}} \begin{bmatrix} U(1)\\ U(2)\\ \vdots \\ U(n-r_i)\end{bmatrix}.$$

Since $\mathcal{A}^*(R^\top U) = 0$, to show that $U = 0$, it suffices to show that the $\widehat{R}\in\mathbb{R}^{n\times(n-r_i)}$ above has full column rank. To this end, we first note from $R\in\mathbb{R}^{1\times(r_i+1)}\backslash\{0\}$ that there is at least one nonzero element in R. Let $\bar{i}$ be the first integer in $1,\ldots,r_i+1$ with $R(\bar{i})\neq 0$. Then the $(n-r_i)\times(n-r_i)$ submatrix of $\widehat{R}$ starting from the $\bar{i}$th row is lower triangular with all diagonal entries being $R(\bar{i}) \neq 0$. Consequently, this submatrix is nonsingular and thus $\widehat{R}$ has full column rank. This completes the proof for this case.

Now we consider (4.1) with setting (4.3). In this case, we have $m = r$, $p = r+1$, $q = N(n-r)$, and $\mathcal{A}(y) = \mathcal{L}(y) = [\mathcal{H}_{r+1}(y_1) \cdots \mathcal{H}_{r+1}(y_N)]$ with $y = vec(y_1 \cdots y_N)$. Notice that $R^\top \in \mathbb{R}^{p\times 1} = \mathbb{R}^{r+1}$ and $U^\top \in \mathbb{R}^{q \times 1} = \mathbb{R}^{N(n-r)}$. Write

$$R = [R(1), \ldots, R(r+1)], \qquad U = [U_1, \ldots, U_N],$$

where $U_i^\top \in \mathbb{R}^{n-r}$ $(i = 1, \ldots, N)$. We then see from Lemma 2.4 that

$$\mathcal{A}^*(R^\top U) = \mathcal{L}^*(R^\top U) = vec\left(\mathcal{H}^*_{r+1}(R^\top U_1) \cdots \overbrace{\mathcal{H}^*_{r+1}(R^\top U_k)}^{\text{the } k\text{th block}} \cdots \mathcal{H}^*_{r+1}(R^\top U_N)\right).$$

Similar to the proof in setting (4.2), we can write the kth block of $\mathcal{A}^*(R^\top U)$ as

$$\underbrace{\begin{bmatrix} R(1) & R(2) & \cdots & R(r+1) & & & & \\ & R(1) & \cdots & R(r) & R(r+1) & & & \\ & & \ddots & \ddots & & \ddots & & \\ & & & & & R(1) & \cdots & R(r+1) \end{bmatrix}^\top}_{\overline{R}} \begin{bmatrix} U_k(1) \\ U_k(2) \\ \vdots \\ U_k(n-r) \end{bmatrix}.$$

Consequently, we have

$$\mathcal{A}^*(R^\top U) = \begin{bmatrix} \overline{R} & & \\ & \ddots & \\ & & \overline{R} \end{bmatrix} U^\top. \tag{4.7}$$

Since $\mathcal{A}^*(R^\top U) = 0$, to prove that $U = 0$, we only need to show that the block diagonal matrix on the right-hand side of (4.7) has full column rank. But then it suffices to show that $\overline{R}$ has full column rank, and this latter claim can be established by following a similar line of arguments as in the proof for setting (4.2). This completes the proof. ∎

Theorem 4.2. *Consider* (4.1) *with setting* (4.2) *or* (4.3). *Then the function* Ψ *defined in* (4.5) *is smooth on* $\mathbb{R}^{1\times p}\backslash\{0\}$.

Proof. In view of [24, equation 5] and recalling that $p - m = 1$ (in both cases (4.2) and (4.3)), we only need to show that for any $R \in \mathbb{R}^{1\times p}\backslash\{0\}$, the linear map $G_R : \mathbb{R}^d \longrightarrow \mathbb{R}^q$ defined as $G_R(y) := (R\mathcal{A}(y))^\top$ is surjective, or equivalently, G_R^* is injective. To proceed, fix any $R \in \mathbb{R}^{1\times p}\backslash\{0\}$ and consider any $z \in \mathbb{R}^q$ with $G_R^*(z) = 0$. Then we have for any $y \in \mathbb{R}^d$ that

$$0 = \langle G_R^*(z), y\rangle = \langle z, G_R(y)\rangle = \langle z, (R\mathcal{A}(y))^\top\rangle = \langle \mathcal{A}^*(R^\top z^\top), y\rangle.$$

Thus we have $\mathcal{A}^*(R^\top z^\top) = 0$, which together with Lemma 4.1 implies that $z = 0$. This completes the proof. ∎

Since Ψ is smooth on $\mathbb{R}^{1\times p}\backslash\{0\}$, we can then apply standard gradient-based optimization methods to solving (4.4) and obtain a stationary point of $\widetilde{\Psi}$ in (4.6). We next discuss how one can obtain a pseudoprojection from such a stationary point.

4.2. Stationarity and improvement of function value. We discuss in this subsection how to obtain a pseudoprojection from a suitable stationary point R^* of $\widetilde{\Psi}$ in (4.6), under mild assumptions. We start by showing how one can construct from R^* a point satisfying the stationarity condition in Definition 2.2.

THEOREM 4.3. *Consider* (4.1) *with setting* (4.2) *or* (4.3). *Let* R^* *be a stationary point of* $\widetilde{\Psi}$ *in* (4.6) *and let* y^* *achieve the infimum in* (4.5) *when* $R = R^*$. *Then*

$$0 \in y^* - \widehat{y} + \mathcal{A}^* \left(N_{\{X:\, \mathrm{rank}(X) \le m\}}(\mathcal{A}(y^*)) \right). \tag{4.8}$$

If in addition $\mathrm{rank}(\mathcal{A}(y^*)) = m$, *then we have*

$$0 \in y^* - \widehat{y} + N_{\{y:\, \mathrm{rank}(\mathcal{A}(y)) \le m\}}(y^*). \tag{4.9}$$

Proof. First, we define

$$\Phi(y, R) := \frac{1}{2}\|y - \widehat{y}\|^2 + \delta_{\{(y,R):\, R\mathcal{A}(y)=0\}}(y, R) + \delta_{\{R:\, RR^\top = 1\}}(R). \tag{4.10}$$

Then we see from (4.6) and the definition of y^* that

$$\widetilde{\Psi}(R^*) = \inf_y \Phi(y, R^*) = \Phi(y^*, R^*). \tag{4.11}$$

On the other hand, we also have from the stationarity of R^* that $0 \in \partial\widetilde{\Psi}(R^*) = \partial(\Psi + \delta_{\{R:\, RR^\top=1\}})(R^*)$. Using this, (4.11), and [21, Theorem 10.13], we see further that

$$(0, 0) \in \partial\Phi(y^*, R^*). \tag{4.12}$$

Next, notice from Lemma 4.1 that for any $U \in \mathbb{R}^{1\times q}$, $y \in \mathbb{R}^d$, $\lambda \in \mathbb{R}$, and $R \in \mathbb{R}^{1\times p}\backslash\{0\}$, the following implication holds:

$$\mathcal{A}^*(R^\top U) = 0 \text{ and } U\mathcal{A}(y)^\top + \lambda R = 0 \quad \Longrightarrow \quad U = 0 \text{ and } \lambda = 0.$$

This corresponds to the linear independence constraint qualification for the following optimization problem:

$$\min_{y\in\mathbb{R}^d, R\in\mathbb{R}^{1\times p}} \ \frac{1}{2}\|y - \widehat{y}\|^2 \ \text{ s.t. } \ R\mathcal{A}(y) = 0 \ \text{ and } \ RR^\top = 1.$$

Using this, the definition of Φ in (4.10), (4.12), and [21, Example 10.8], we deduce that there exist $V^* \in \mathbb{R}^{1\times q}$ and a scalar λ^* such that the following Karash–Kuhn–Tucker conditions hold:

$$y^* - \widehat{y} + \mathcal{A}^*({R^*}^\top V^*) = 0, \quad V^*\left(\mathcal{A}(y^*)\right)^\top + \lambda^* R^* = 0, \tag{4.13}$$

$$R^*{R^*}^\top - 1 = 0, \quad R^*\mathcal{A}(y^*) = 0. \tag{4.14}$$

Multiplying both sides of the second equation in (4.13) from the right by ${R^*}^\top$, and using the two equations in (4.14), we obtain $\lambda^* = 0$ and thus

$$V^*\left(\mathcal{A}(y^*)\right)^\top = 0. \tag{4.15}$$

We now show that

$${R^*}^\top V^* \in N_{\{X:\, \mathrm{rank}(X)\le m\}}(\mathcal{A}(y^*)). \tag{4.16}$$

To proceed, recall that $R^* \in \mathbb{R}^{1\times p}$, which implies $\mathrm{rank}({R^*}^\top V^*) \le 1$. According to [10, Proposition 3.6], in order to establish (4.16), it now remains to show that

$$[\ker({R^*}^\top V^*)]^\perp \cap [\ker(\mathcal{A}(y^*))]^\perp = \{0\}. \tag{4.17}$$

To this end, take any $z \in [\ker({R^*}^\top V^*)]^\perp \cap [\ker(\mathcal{A}(y^*))]^\perp$. Then we have in particular that $z \in [\ker({R^*}^\top V^*)]^\perp = \mathrm{Range}({V^*}^\top R^*)$. This together with (4.15) implies that $\mathcal{A}(y^*)z \in \mathcal{A}(y^*)\mathrm{Range}({V^*}^\top R^*) = \{0\}$. Thus, we must have $z \in \ker(\mathcal{A}(y^*)) \cap [\ker(\mathcal{A}(y^*))]^\perp$ and consequently $z = 0$. This proves (4.17) and hence (4.16). The desired relation (4.8) now follows immediately from (4.13) and (4.16).

Suppose in addition that $\mathrm{rank}(\mathcal{A}(y^*)) = m$. Then we have

$$\begin{aligned}&\mathcal{A}^*\left(N_{\{X:\,\mathrm{rank}(X)\le m\}}(\mathcal{A}(y^*))\right) \overset{\text{(a)}}{\subseteq} \mathcal{A}^*\left(\widehat{N}_{\{X:\,\mathrm{rank}(X)\le m\}}(\mathcal{A}(y^*))\right)\\ &\overset{\text{(b)}}{\subseteq} \widehat{N}_{\{y:\,\mathrm{rank}(\mathcal{A}(y))\le m\}}(y^*) \overset{\text{(c)}}{\subseteq} N_{\{y:\,\mathrm{rank}(\mathcal{A}(y))\le m\}}(y^*),\end{aligned}$$

where (a) follows from [10, Proposition 3.6] and the fact that proximal normal vectors are regular normal vectors [21, Example 6.16], (b) follows from [21, Theorem 10.6], and (c) follows from [21, Proposition 6.5]. This together with (4.8) proves (4.9). This completes the proof. ∎

We next show that if the stationary point R^* of $\widetilde{\Psi}$ in (4.6) is obtained via a gradient-based *descent* optimization method with a suitably chosen initial point, then the y^* that attains the infimum in (4.5) will satisfy the condition on function value improvement in Definition 2.2.

THEOREM 4.4. *Consider* (4.1) *with setting* (4.2) *or* (4.3). *Let* $y_b \in \mathbb{R}^d$ *satisfy* $\mathrm{rank}(\mathcal{A}(y_b)) \le m$ *and let* $R^0 \in \mathbb{R}^{1\times p}\backslash\{0\}$ *satisfy* $R^0\mathcal{A}(y_b) = 0$. *Then for any* $\widetilde{R} \in \mathbb{R}^{1\times p}\backslash\{0\}$ *with* $\Psi(\widetilde{R}) \le \Psi(R^0)$, *we have*

$$\|y_{\widetilde{R}} - \widehat{y}\| \le \|y_b - \widehat{y}\|, \tag{4.18}$$

where $y_{\widetilde{R}}$ *attains the infimum in* (4.5) *when* $R = \widetilde{R}$.

Proof. First, we see from $R^0\mathcal{A}(y_b) = 0$ and the definition of Ψ in (4.5) that $\Psi(R^0) \le \frac{1}{2}\|y_b - \widehat{y}\|^2$. This together with the assumption $\Psi(\widetilde{R}) \le \Psi(R^0)$ and the fact that $y_{\widetilde{R}}$ attains the infimum in (4.5) when $R = \widetilde{R}$ shows that

$$\frac{1}{2}\|y_{\widetilde{R}} - \widehat{y}\|^2 = \Psi(\widetilde{R}) \le \Psi(R^0) \le \frac{1}{2}\|y_b - \widehat{y}\|^2.$$

This completes the proof. ∎

Remark 4.5 (obtaining pseudoprojection in cases (4.2) or (4.3)). Let $y_b \in \mathbb{R}^d$ satisfy $\mathrm{rank}(\mathcal{A}(y_b)) \le m$ and let $R^0 \in \mathbb{R}^{1\times p}\backslash\{0\}$ satisfy $R^0\mathcal{A}(y_b) = 0$. Then one can apply some standard gradient-based descent methods such as those implemented in SLRA [18] for solving (4.4) with R^0 as the initialization: these methods typically generate a sequence $\{R^k\}$ so that any accumulation point, say, R^*, is stationary for $\widetilde{\Psi}$ in (4.6) and satisfies $\Psi(R^*) \le \Psi(R^0)$. Suppose y_{R^*} achieves the infimum in (4.5) when $R = R^*$. Then we know from (4.9) in Theorem 4.3 and (4.18) in Theorem 4.4 that if $\mathrm{rank}(\mathcal{A}(y_{R^*})) = m$ holds, then $y_{R^*} \in \mathcal{P}^s_{\mathrm{rank}(\mathcal{A}(y))\le m}(\widehat{y}; y_b)$.

4.3. Conjecture related to Theorem 4.3. In this subsection, we revisit the assumption $\operatorname{rank}(\mathcal{A}(y^*)) = m$ in Theorem 4.3. We would like to understand how likely such a condition is fulfilled by the y^* that achieves the infimum in (4.5), with $R = R^*$ being a stationary point of $\widetilde{\Psi}$ in (4.6). Notice that if R^* is indeed an optimal solution of $\widetilde{\Psi}$, such a y^* is an optimal solution of (4.1). Thus, we will first study whether $\operatorname{rank}(\mathcal{A}(y^*)) = m$ when y^* is an optimal solution of (4.1). Specifically, we make the following conjecture.

Conjecture 4.6. Let s be a positive integer. Suppose that $\widehat{y} \in \mathbb{R}^n$ satisfies the condition $\operatorname{rank}(\mathcal{H}_{s+1}(\widehat{y})) = s+1$ and let y^* solve the following optimization problem:

$$\min_{y\in\mathbb{R}^n} \quad \frac{1}{2}\|y-\widehat{y}\|^2 \quad \text{s.t.} \quad \operatorname{rank}\left(\mathcal{H}_{s+1}(y)\right) \le s. \tag{4.19}$$

Then we have $\operatorname{rank}\left(\mathcal{H}_{s+1}(y^*)\right) = s$.

We do not know whether Conjecture 4.6 holds true for all positive numbers s. However, we are able to prove that it holds true when $s = 1$.

PROPOSITION 4.7. *Conjecture* 4.6 *holds true when* $s = 1$.

Proof. Since $s = 1$, we only need to show that there exists $\overline{y} \in \mathbb{R}^n$ with rank $(\mathcal{H}_2(\overline{y})) = 1$ and $\|\overline{y}-\widehat{y}\|^2 < \|\widehat{y}\|^2$. First of all, since $\operatorname{rank}(\mathcal{H}_2(\widehat{y})) = 2$, we must have $n \ge 3$. We consider two cases:

$$\text{(i) } \widehat{y}(1) \neq 0 \ \text{ or } \ \widehat{y}(n) \neq 0; \qquad \text{(ii) } \widehat{y}(1) = 0 \ \text{ and } \ \widehat{y}(n) = 0.$$

For case (i), we let $\overline{y} = [\widehat{y}(1)\ 0\cdots 0]^\top$ when $\widehat{y}(1) \neq 0$, and $\overline{y} = [0\cdots 0\ \widehat{y}(n)]^\top$ when $\widehat{y}(n) \neq 0$. Then $\operatorname{rank}(\mathcal{H}_2(\overline{y})) = 1$ and

$$\|\overline{y}-\widehat{y}\|^2 = \sum_{i=2}^{n} \widehat{y}^2(i) < \|\widehat{y}\|^2 \quad \text{or} \quad \|\overline{y}-\widehat{y}\|^2 = \sum_{i=1}^{n-1} \widehat{y}^2(i) < \|\widehat{y}\|^2.$$

Now we consider case (ii). Notice that there exists at least one nonzero element in $\{\widehat{y}(2),\cdots,\widehat{y}(n-1)\}$ because $\operatorname{rank}(\mathcal{H}_2(\widehat{y})) = 2$. Hence, there are at most $n-2$ distinct real roots for the polynomial equation $\sum_{i=2}^{n-1} \widehat{y}(i)(z)^{i-1} = 0$. Let $\bar{z} \neq 0$ be a real number different from these roots. Then we have $\sum_{i=0}^{n-1} (\overline{z})^{2i} > 0$. Let

$$\overline{c} = \sum_{i=2}^{n-1} \widehat{y}(i)(\overline{z})^{i-1} \Big/ \sum_{i=0}^{n-1} (\overline{z})^{2i} \quad \text{and} \quad \overline{y} = [\overline{c} \ \ \overline{cz} \ \cdots \ \overline{cz}^{n-1}]^\top.$$

Then $\overline{c} \neq 0$ and $\operatorname{rank}\left(\mathcal{H}_2(\overline{y})\right) = 1$. Consequently,

$$\|\overline{y}-\widehat{y}\|^2 - \|\widehat{y}\|^2 = \|\overline{y}\|^2 - 2\overline{y}^\top\widehat{y} = \overline{c}^2 \sum_{i=0}^{n-1} (\overline{z})^{2i} - 2\overline{c} \sum_{i=2}^{n-1} \widehat{y}(i)(\overline{z})^{i-1} = -\overline{c}^2 \sum_{i=0}^{n-1} (\overline{z})^{2i} < 0.$$

This completes the proof. $\square$

5. Numerical experiments. In this section, we will conduct numerical experiments for our hybrid penalty method, i.e., Algorithm 3.2. All numerical experiments are performed in MATLAB R2019a on a 64-bit PC with 3.8 GHz Intel Core i5 Quad-Core and 8 GB of DDR4 RAM.

We consider the following problem with two rank constraints:

$$\begin{aligned}
&\min_{y_1\in\mathbb{R}^n, y_2\in\mathbb{R}^n} && \frac{1}{2}\|y_1-\overline{y}_1\|_W^2+\frac{1}{2}\|y_2-\overline{y}_2\|_W^2\\
&\quad\text{s.t.} && \operatorname{rank}\left(\mathcal{H}_{n_1+n_c+1}(y_1)\right)\le n_1+n_c,\\
&&& \operatorname{rank}\left(\mathcal{H}_{n_2+n_c+1}(y_2)\right)\le n_2+n_c,\\
&&& \operatorname{rank}\left([\mathcal{H}_{n_1+n_2+n_c+1}(y_1)\ \ \mathcal{H}_{n_1+n_2+n_c+1}(y_2)]\right)\le n_1+n_2+n_c,
\end{aligned} \tag{5.1}$$

where $\|y\|_W := \sqrt{y^\top W y}$, W is the $n\times n$ diagonal matrix so that $W(i,i)$ equals 1 when i is odd and equals 10 when i is even, n_1, n_2, and n_c are given positive integers, and $\overline{y}_1\in\mathbb{R}^n$ and $\overline{y}_2\in\mathbb{R}^n$ are known noisy signals.

Let HB_1, HB_2, and HB_3 represent the three hybrid penalty methods which solve (5.1) by Algorithm 3.2 via the reformulation (3.1) with Variant I, Variant II, and Variant III discussed in section 3.1, respectively. Let AP represent the alternating pseudoprojection algorithm (3.9) applied directly to the sets Ω_1 and Ω_2 defined in (3.6), constructed based on the data from (5.1).

Data generation. We first consider $n=50$ with two 3-tuples $(n_1,n_2,n_c)=(2,2,2)$ and $(n_1,n_2,n_c)=(2,6,4)$, and then consider $n=100$ and $n=200$ with fixed 3-tuple $(n_1,n_2,n_c)=(2,6,4)$. For each 3-tuple, we first randomly generate two signals y_1 and y_2 from two marginally stable linear time-invariant systems of order at most n_1+n_c and n_2+n_c, respectively, which have n_c common poles. Then we let $\overline{y}_1=y_1+\sigma\cdot W^{-1/2}\xi_1$ and $\overline{y}_2=y_2+\sigma\cdot W^{-1/2}\xi_2$, where $\sigma=0.1$ is the noise factor, and ξ_1 and ξ_2 are random vectors with independent and identically distributed standard Gaussian entries.

HB_1, HB_2, and HB_3. In Algorithm 3.1, we set $L_{\max}=10^8$, $L_{\min}=10^{-8}$, $\tau=2$, $c=10^{-4}$, $M=4$, $L_0^0=1$, and for $l\ge 1$,

$$L_l^0=\max\left\{\min\left\{\frac{\left(y^l-y^{l-1}\right)^\top\left(\nabla h(y^l)-\nabla h(y^{l-1})\right)}{\|y^l-y^{l-1}\|^2}, L_{\max}\right\}, L_{\min}\right\}.$$

All pseudoprojection subproblems that arise are approximately solved by calling SLRA [18] with default setting (except that the R^0 is specified as in Remark 4.5). We terminate Algorithm 3.1 when the number of iterations exceeds 10^8 or

$$\frac{\|y^l-y^{l-1}\|}{\max\{\|y^l\|,1\}}<\epsilon_t/\overline{L}_{l-1}\quad\text{or}\quad\frac{\left|F_{\lambda_t}(y^l)-F_{\lambda_t}(y^{l-1})\right|}{\max\{|F_{\lambda_t}(y^l)|,1\}}<10^{-10}.$$

For the penalty method in Algorithm 3.2, we set $y^{\text{feas}}=0$, $\lambda_t=\lambda_{t-1}/5$ with initial $\lambda_0=0.1$, $\overline{\lambda}=10^{-4}$ and $\epsilon_t=\max\{\epsilon_{t-1}/1.5,10^{-6}\}$ with initial $\epsilon_0=10^{-5}$. Let $\overline{y}=vec\,(\overline{y}_1\ \overline{y}_2)$. We set the initial point y^0 for HB_1 and HB_2 as a pseudoprojection of $\overline{y}$ onto Ω_1 and Ω_2, respectively, obtained by calling SLRA in [18] with default setting (the reference point is the origin). For HB_3, we set $y^0=\overline{y}$.

For the postprocessing method in Algorithm 3.2, we also call SLRA in [18] with default settings to approximately compute a pseudoprojection (except that the R^0 is

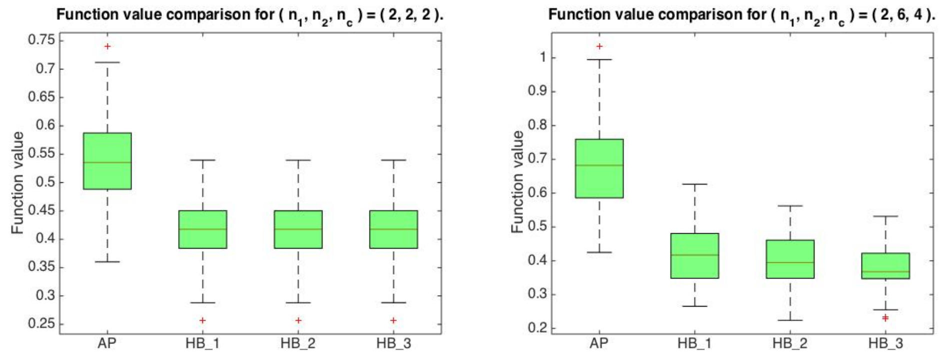


FIG. 1. *Comparing terminating function values among* AP, HB_1, HB_2*, and* HB_3 *for* $n = 50$.

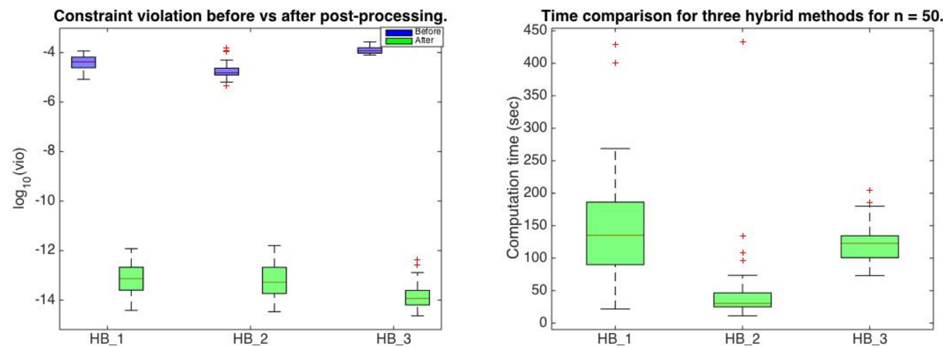


FIG. 2. *Comparing constraint violations and computation times (in seconds) among* HB_1, HB_2*, and* HB_3 *for* $n = 50$ *and* $(n_1, n_2, n_c) = (2, 6, 4)$.

specified as in Remark 4.5), and terminate it when the number of iterations exceeds 10^5 or

$$\frac{\max\{\|x^t - x^{t-1}\|, \|z^t - z^{t-1}\|\}}{\max\{\|x^{t-1}\|, \|z^{t-1}\|, 1\}} < 10^{-10}.$$

We output z^t as the approximate solution.

AP. In this method, we start at $\overline{y} = vec(\overline{y}_1\ \overline{y}_2)$ and call SLRA in [18] with default setting (except that the R^0 is specified as in Remark 4.5) to approximately compute a pseudoprojection onto Ω_1 and Ω_2 defined in (3.6) (the initial reference points are the origin). We also output z^t as the approximate solution.

Numerical results. In Figure 1, we compare the four methods AP, HB_1, HB_2, and HB_3 in terms of terminating function values over 100 random instances for $n = 50$ and $(n_1, n_2, n_c) = (2, 2, 2)$ and over 30 random instances for $n = 50$ and $(n_1, n_2, n_c) = (2, 6, 4)$.[3] One can see that while the three hybrid penalty methods HB_1, HB_2, and HB_3 have comparable performance, they always outperform AP.

In Figure 2, we compare the three hybrid penalty methods HB_1, HB_2, and HB_3 in terms of constraint violation (before and after postprocessing) and computation

[3] For each 3-tuple, we first generate y_1 and y_2 as described above. For these two fixed signals, we generate 100 (and, resp., 30) random noisy signals $\bar{y}_1$ and $\bar{y}_2$ and solve the corresponding instances.

time over 30 random instances for $n = 50$ and $(n_1, n_2, n_c) = (2, 6, 4)$. We measure constraint violation by $\log_{10}(vio)$, with vio given by

$$\max\left\{\frac{\mathrm{dist}(\mathcal{H}_{m_1+1}(y_1^*), \Xi_{m_1})}{\|\mathcal{H}_{m_1+1}(y_1^*)\|_2}, \frac{\mathrm{dist}(\mathcal{H}_{m_2+1}(y_2^*), \Xi_{m_2})}{\|\mathcal{H}_{m_2+1}(y_2^*)\|_2}, \frac{\mathrm{dist}([\mathcal{H}_{m+1}(y_1^*), \mathcal{H}_{m+1}(y_2^*)], \Xi_m)}{\|[H_{m+1}(y_1^*), \mathcal{H}_{m+1}(y_2^*)]\|_2}\right\},$$

where y_1^* and y_2^* are computed solutions, $m_1 = n_1 + n_c$, $m_2 = n_2 + n_c$, $m = n_1 + n_2 + n_c$, and $\Xi_s := \{Y : \mathrm{rank}(Y) \leq s\}$. One can see that the postprocessing scheme significantly reduces constraint violation. On the other hand, HB_2 is faster than HB_1 and HB_3.

In Figures 3 and 4, fixing $(n_1, n_2, n_c) = (2, 6, 4)$, for $n = 100$ and $n = 200$, respectively, we compare the four methods AP, HB_1, HB_2, and HB_3 in terms of terminating function values, and also compare the three hybrid penalty methods HB_1, HB_2, and HB_3 in terms of computation time over 30 random instances.

In Figure 5, fixing $(n_1, n_2, n_c) = (2, 6, 4)$, for $n = 100$ and $n = 200$, respectively, we compare the three hybrid penalty methods HB_1, HB_2, and HB_3 in terms of function values (before and after postprocessing) over 30 random instances. One can see that function values are increased but not significantly after postprocessing.

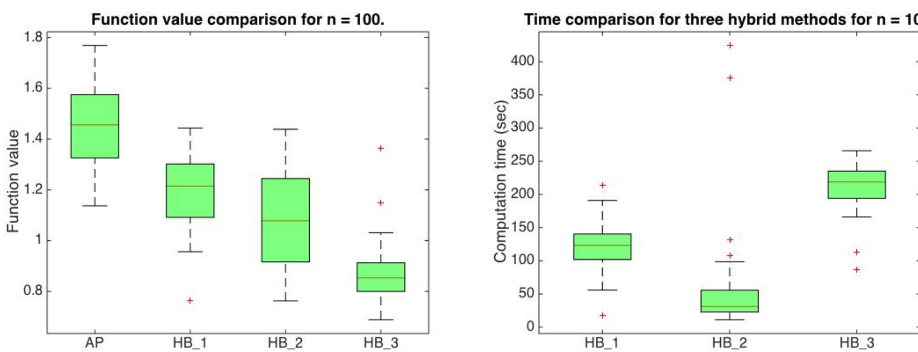


FIG. 3. *Comparing terminating function values and computation times (in seconds) for $n = 100$ and $(n_1, n_2, n_c) = (2, 6, 4)$.*

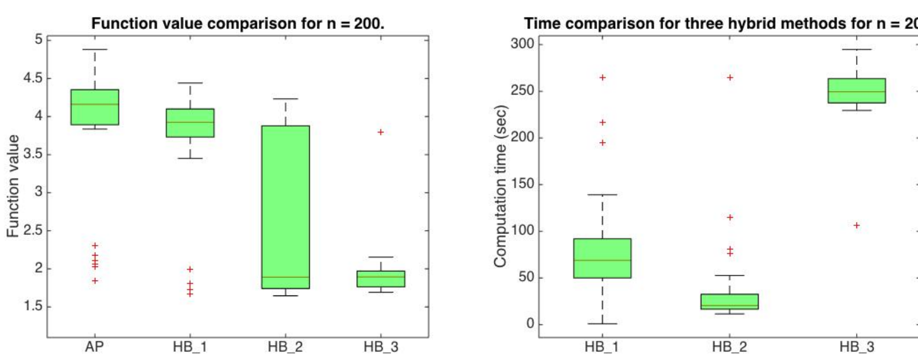
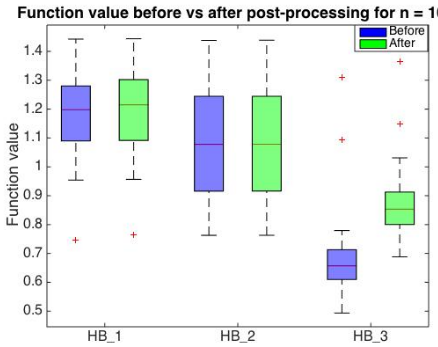


FIG. 4. *Comparing terminating function values and computation times (in seconds) for $n = 200$ and $(n_1, n_2, n_c) = (2, 6, 4)$.*



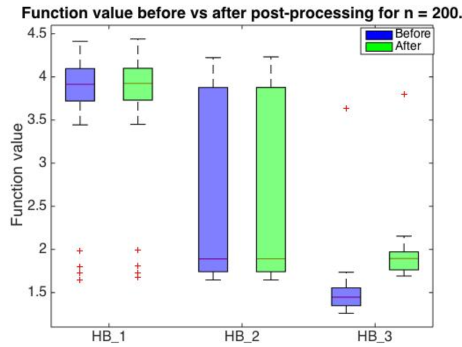


FIG. 5. *Comparing terminating function values before and after postprocessing among* HB_1, HB_2, *and* HB_3 *for* $n = 100$ *and* $n = 200$ *with* $(n_1, n_2, n_c) = (2, 6, 4)$.

6. Concluding remarks. In this paper, we propose a hybrid penalty method for solving (1.1). The hybrid penalty method consists of two parts: a penalty scheme which makes use of a special penalty function as in [9] and a postprocessing method for reducing constraint violation. Both the penalty subproblems and the subproblems in the postprocessing method involve the new concept of pseudoprojections: we discussed in section 4 in detail how pseudoprojections can be computed efficiently by some existing software such as [18], under mild assumptions.

There are several open questions related to pseudoprojection computation. For instance, we still do not know how likely the condition $\operatorname{rank}(\mathcal{A}(y^*)) = m$ holds for the y^* that achieves the infimum in (4.5) (with $R = R^*$ being a stationary point of $\widetilde{\Psi}$ in (4.6)).[4] Even assuming y^* is a solution of (4.19), we can only establish $\operatorname{rank}(\mathcal{H}_{s+1}(y^*)) = s$ when $s = 1$. The case for $s > 1$ is still open.

Appendix A. Proof of Theorem 3.4. Before proving Theorem 3.4, we first state two auxiliary lemmas without proofs. The proof of Lemma A.1 can be found in the first paragraph in the proof of [8, Theorem 5.16], and Lemma A.2 follows from Theorem 3.3 and the same argument as in the proof of [8, Theorem 5.16].

LEMMA A.1. *Let* Ω_1 *and* Ω_2 *be defined as in* (3.6), $\overline{y} \in \Omega_1 \cap \Omega_2$ *and define*

$$\overline{c} := \max \{\langle u, v\rangle : u \in N_{\Omega_1}(\overline{y}) \cap B, \quad v \in -N_{\Omega_2}(\overline{y}) \cap B\}, \tag{A.1}$$

where B *is the closed unit ball. Then* $N_{\Omega_1}(\overline{y}) \cap -N_{\Omega_2}(\overline{y}) = \{0\}$ *if and only if* $\overline{c} < 1$.

LEMMA A.2. *Let* Ω_1 *and* Ω_2 *be defined as in* (3.6). *Suppose that there exists some* $\overline{y} \in \Omega_1 \cap \Omega_2$ *such that* $\operatorname{rank}(\mathcal{L}(\overline{y})) = r$ *and* $N_{\Omega_1}(\overline{y}) \cap -N_{\Omega_2}(\overline{y}) = \{0\}$. *Let* $\overline{c}$ *be defined as in* (A.1). *Then for any* $c \in (\overline{c}, 1)$, *there exist some* $\epsilon > 0$ *and* $\delta \in [0, \frac{1-c}{2})$ *such that*

$$\left.\begin{array}{ll} x \in \Omega_1 \cap B_\epsilon(\overline{y}), & u \in N_{\Omega_1}(x) \cap B \\ z \in \Omega_2 \cap B_\epsilon(\overline{y}), & v \in -N_{\Omega_2}(z) \cap B \end{array}\right\} \Longrightarrow \langle u, v\rangle \le c, \tag{A.2}$$

$$\left.\begin{array}{l} x, z \in \Omega_2 \cap B_\epsilon(\overline{y}) \\ v \in N_{\Omega_2}(z) \cap B \end{array}\right\} \Longrightarrow \langle v, x - z\rangle \le \delta\|x - z\|, \tag{A.3}$$

where $B_\epsilon(\overline{y})$ *is the closed ball with center* $\overline{y}$ *and radius* ϵ, *and* B *is the closed unit ball.*

[4]In the numerical experiments in section 5, the condition $\operatorname{rank}(\mathcal{A}(y^*)) = m$ almost never fails for the solution y^* returned by SLRA: for over 99.9% of our calls to SLRA, the mth singular value of $\mathcal{A}(y^*)$ is significantly larger than its next singular value.

We now prove Theorem 3.4. The proof follows the same line of arguments as in [8, Theorem 5.2].

Proof. Fix any $c \in (\bar{c}, 1)$ with $\bar{c}$ defined as in (A.1), and let δ and ϵ be given as in Lemma A.2. We first claim that

$$\left.\begin{array}{l}\|z^{t+1}-\overline{y}\| \leq \frac{\epsilon}{2} \\ \|z^{t+1}-x^t\| \leq \frac{\epsilon}{2}\end{array}\right\} \Longrightarrow \|x^{t+1}-z^{t+1}\| \leq c_0\|x^t - z^{t+1}\|, \tag{A.4}$$

where $c_0 := c + 2\delta$. To prove this, note from (3.19) and Definition 2.2 that

$$x^t - z^{t+1} \in N_{\Omega_1}(z^{t+1}), \;\; z^{t+1} - x^{t+1} \in N_{\Omega_2}(x^{t+1}), \tag{A.5}$$

$$\|z^{t+1} - x^t\| \leq \|z^t - x^t\|, \;\; \|x^{t+1} - z^{t+1}\| \leq \|x^t - z^{t+1}\|. \tag{A.6}$$

If $\|x^{t+1}-z^{t+1}\| = 0$ or $\|x^t - z^{t+1}\| = 0$, we then see from the second inequality in (A.6) that (A.4) holds trivially. Now we assume that $\|x^{t+1}-z^{t+1}\| \neq 0$ and $\|x^t - z^{t+1}\| \neq 0$. We first notice from (A.6), $\|z^{t+1}-\overline{y}\| \leq \frac{\epsilon}{2}$, and $\|z^{t+1}-x^t\| \leq \frac{\epsilon}{2}$ that

$$\|x^{t+1}-\overline{y}\| \leq \|x^{t+1}-z^{t+1}\| + \|z^{t+1}-\overline{y}\| \leq \|z^{t+1}-x^t\| + \|z^{t+1}-\overline{y}\| \leq \epsilon, \tag{A.7}$$

$$\|x^t - x^{t+1}\| \leq \|x^t - z^{t+1}\| + \|z^{t+1}-x^{t+1}\| \leq 2\|x^t - z^{t+1}\|, \tag{A.8}$$

$$\|x^t - \overline{y}\| \leq \|x^t - z^{t+1}\| + \|z^{t+1}-\overline{y}\| \leq \epsilon. \tag{A.9}$$

Using (A.5), (A.7), and $\|z^{t+1}-\overline{y}\| \leq \frac{\epsilon}{2}$, we obtain further that

$$\tfrac{x^t - z^{t+1}}{\|x^t - z^{t+1}\|} \in N_{\Omega_1}(z^{t+1}) \cap B \text{ with } \; z^{t+1} \in \Omega_1 \cap B_\epsilon(\overline{y}), \tag{A.10}$$

$$\tfrac{x^{t+1} - z^{t+1}}{\|x^{t+1} - z^{t+1}\|} \in -N_{\Omega_2}(x^{t+1}) \cap B \text{ with } \; x^{t+1} \in \Omega_2 \cap B_\epsilon(\overline{y}). \tag{A.11}$$

Here, B represents the closed unit ball and $B_\epsilon(\overline{y})$ represents the closed ball with center $\overline{y}$ and radius ϵ. Furthermore, we see from (A.2), (A.10), and (A.11) that

$$\langle x^t - z^{t+1}, x^{t+1} - z^{t+1}\rangle \leq c\|x^t - z^{t+1}\|\|x^{t+1} - z^{t+1}\|. \tag{A.12}$$

On the other hand, in view of (A.7), (A.9), and (A.11), we can apply (A.3) with $x = x^t$, $z = x^{t+1}$, and $v = \frac{z^{t+1}-x^{t+1}}{\|z^{t+1}-x^{t+1}\|}$ to obtain

$$\begin{aligned}\langle x^t - x^{t+1}, z^{t+1} - x^{t+1}\rangle &\leq \delta\|x^t - x^{t+1}\|\|z^{t+1} - x^{t+1}\| \\ &\leq 2\delta\|x^t - z^{t+1}\|\|z^{t+1} - x^{t+1}\|,\end{aligned} \tag{A.13}$$

where the second inequality follows from (A.8). Adding (A.12) and (A.13), we obtain

$$\|x^{t+1} - z^{t+1}\| \leq (c + 2\delta)\|x^t - z^{t+1}\| = c_0\|x^t - z^{t+1}\|,$$

which proves (A.4).

Note from $c_0 = c + 2\delta$ with $c \in (\bar{c}, 1)$ and $\delta \in [0, \frac{1-c}{2})$ that $c_0 \in (0, 1)$. Choose initial points x^0 and z^0 such that $\gamma := \|x^0 - \overline{y}\| + \|z^0 - x^0\| < \frac{(1-c_0)\epsilon}{4}$. Next, we prove the following inequalities by induction:

$$\|z^{t+1} - x^t\| \leq \gamma {c_0}^t < \tfrac{\epsilon}{2}, \tag{A.14}$$

$$\|z^{t+1} - \overline{y}\| \leq 2\gamma \tfrac{1-{c_0}^{t+1}}{1-c_0} < \tfrac{\epsilon}{2}, \tag{A.15}$$

$$\|x^{t+1} - z^{t+1}\| \leq \gamma {c_0}^{t+1}. \tag{A.16}$$

First, we prove that the above three inequalities hold for $t = 0$. Note from $c_0 \in (0, 1)$, the z-update in (3.19), and the definition of γ that

$$\|z^1 - x^0\| \le \|z^0 - x^0\| \le \gamma < \frac{\epsilon}{2} \quad \text{and} \quad \|z^1 - \overline{y}\| \le \|z^1 - x^0\| + \|x^0 - \overline{y}\| \le 2\gamma < \frac{\epsilon}{2},$$

which proves (A.14) and (A.15) for $t = 0$. Then we see from $\|z^1 - x^0\| < \frac{\epsilon}{2}$, $\|z^1 - \overline{y}\| < \frac{\epsilon}{2}$ and (A.4) that

$$\|x^1 - z^1\| \le c_0\|x^0 - z^1\| \le \gamma c_0,$$

which proves (A.16) for $t = 0$. To prove by induction, we assume that (A.14), (A.15), and (A.16) hold for some $t \ge 0$. We know from the z-update, (A.14), and (A.16) that

$$\|z^{t+2} - x^{t+1}\| \le \|z^{t+1} - x^{t+1}\| \le \gamma c_0^{t+1} < \tfrac{\epsilon}{2}.$$

This together with (A.15) and (A.16) implies

$$\begin{aligned}\|z^{t+2} - \overline{y}\| &\le \|z^{t+2} - x^{t+1}\| + \|x^{t+1} - z^{t+1}\| + \|z^{t+1} - \overline{y}\| \\ &\le \gamma c_0^{t+1} + \gamma c_0^{t+1} + 2\gamma \tfrac{1 - c_0^{t+1}}{1 - c_0} = 2\gamma \tfrac{1 - c_0^{t+2}}{1 - c_0} < \tfrac{2\gamma}{1 - c_0} < \tfrac{\epsilon}{2}.\end{aligned}$$

We then see from $\|z^{t+2} - x^{t+1}\| < \frac{\epsilon}{2}$, $\|z^{t+2} - \overline{y}\| < \frac{\epsilon}{2}$ and (A.4) that

$$\|x^{t+2} - z^{t+2}\| \le c_0\|x^{t+1} - z^{t+2}\| \le \gamma c_0^{t+2}.$$

Thus, we proved (A.14), (A.15), and (A.16) for $t + 1$. This completes the induction.

Now we prove that the sequence $\{z^0, x^0, z^1, x^1 \cdots\}$ is a Cauchy sequence. For any t and $k > s \ge t$, we know from (A.14) and (A.16) that

$$\begin{aligned}\|z^k - z^s\| &\le \textstyle\sum_{j=s}^{k-1} \left(\|z^{j+1} - x^j\| + \|x^j - z^j\|\right) \le 2\gamma \left(c_0^s + c_0^{s+1} + \cdots + c_0^{k-1}\right) \le \tfrac{2\gamma c_0^t}{1 - c_0}, \\ \|x^k - x^s\| &\le \textstyle\sum_{j=s}^{k-1} \left(\|x^{j+1} - z^{j+1}\| + \|z^{j+1} - x^j\|\right) \le \gamma \sum_{j=s}^{k-1} c_0^{j+1} + \gamma \sum_{j=s}^{k-1} c_0^j \le \tfrac{\gamma c_0^t(1 + c_0)}{1 - c_0}.\end{aligned}$$

Furthermore, by using (A.16), we have

$$\begin{aligned}\|z^k - x^s\| &\le \|z^k - z^s\| + \|z^s - x^s\| \le \tfrac{2\gamma c_0^t}{1 - c_0} + \gamma c_0^t, \\ \|x^k - z^s\| &\le \|x^k - x^s\| + \|x^s - z^s\| \le \tfrac{\gamma c_0^t(1 + c_0)}{1 - c_0} + \gamma c_0^t.\end{aligned}$$

These prove that the sequence $\{z^0, x^0, z^1, x^1 \cdots\}$ is a Cauchy sequence. Therefore, it converges to some $y^* \in \Omega_1 \cap \Omega_2$ and we have for any t that

$$\|z^t - y^*\| \le \tfrac{2\gamma c_0^t}{1 - c_0} \quad \text{and} \quad \|x^t - y^*\| \le \tfrac{\gamma c_0^t(1 + c_0)}{1 - c_0}.$$

Thus the sequence $\{z^0, x^0, z^1, x^1 \cdots\}$ converges R-linearly. This completes the proof. ∎

REFERENCES

[1] M. Chu, N. Del Buono, L. Lopez, and T. Politi, *On the low-rank approximation of data on the unit sphere*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 46–60.

[2] C. Eckart and G. Young, *The approximation of one matrix by another of lower rank*, Psychometrika, 1 (1936), pp. 211–218.

[3] M. Fazel, *Matrix Rank Minimization with Applications*, Ph.D. thesis, Electrical Engineering Department, Stanford University, 2002.

[4] G. Golub and C. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, 1996.
[5] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1990.
[6] M. Ishteva, K. Usevich, and I. Markovsky, *Factorization approach to structured low-rank approximation with applications*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 1180–1204.
[7] N. K. Karmarkar and Y. N. Lakshman, *On approximate GCDs of univariate polynomials*, J. Symbolic Comput., 26 (1998), pp. 653–666.
[8] A. S. Lewis, D. R. Luke, and J. Malick, *Local linear convergence for alternating and averaged nonconvex projections*, Found. Comput. Math., 9 (2007), pp. 485–513.
[9] T. Liu, T. K. Pong, and A. Takeda, *A successive difference-of-convex approximation method for a class of nonconvex nonsmooth optimization problems*, Math. Program., 176 (2019), pp. 339–367.
[10] D. R. Luke, *Prox-regularity of rank constraint sets and implications for algorithms*, J. Math. Imaging Vision, 47 (2013), pp. 231–238.
[11] I. Markovsky, *Structured low-rank approximation and its applications*, Automatica J. IFAC, 44 (2008), pp. 891–909.
[12] I. Markovsky, *Recent progress on variable projection methods for structured low-rank approximation*, Signal Processing, 96 (2014), pp. 406–419.
[13] I. Markovsky, *Low-Rank Approximation: Algorithms, Implementation, Applications*, Springer, New York, 2019.
[14] I. Markovsky, A. Fazzi, and N. Guglielmi, *Applications of polynomial common factor computation in signal processing*, in Latent Variable Analysis and Signal Separation, Lecture Notes in Comput. Sci. 10891, Springer, New York, 2018, pp. 99–106.
[15] I. Markovsky and S. Van Huffel, *Overview of total least-squares methods*, Signal Processing, 87 (2007), pp. 2283–2302.
[16] I. Markovsky, T. Liu, and A. Takeda, *Subspace methods for multi-channel sum-of-exponentials common dynamics estimation*, in Proceedings of the IEEE Conference on Decision and Control, 2019, pp. 2672–2675.
[17] I. Markovsky and K. Usevich, *Structured low-rank approximation with missing data*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 814–830.
[18] I. Markovsky and K. Usevich, *Software for weighted structured low-rank approximation*, J. Comput. Appl. Math., 256 (2014), pp. 278–292.
[19] J. -M. Papy, L. De Lathauwer, and S. Van Huffel, *Common pole estimation in multi-channel exponential data modeling*, Signal Processing, 86 (2006), pp. 846–858.
[20] R. A. Poliquin and R. T. Rockafellar, *A calculus of prox-regularity*, J. Convex Anal., 17 (2010), pp. 203–210.
[21] R. T. Rockafellar and R. J-B. Wets, *Variational Analysis*, Springer, New York, 1998.
[22] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*, Cambridge University Press, Cambridge, UK, 2004.
[23] P. Stoica and Y. Selen, *Model-order selection: A review of information criterion rules*, IEEE Signal Proc. Magazine, 21 (2004), pp. 36–47.
[24] K. Usevich and I. Markovsky, *Structured low-rank approximation as a rational function minimization*, IFAC Proc., 45 (2012), pp. 722–727.
[25] K. Usevich and I. Markovsky, *Variable projection for affinely structured low-rank approximation in weighted 2-norms*, J. Comput. Appl. Math., 272 (2017), pp. 430–448.
[26] K. Usevich and I. Markovsky, *Variable projection methods for approximate (greatest) common divisor computations*, Theoret. Comput. Sci., 681 (2017), pp. 176–198.
[27] S. J. Wright, R. D. Nowak, and M. A. Figueiredo, *Sparse reconstruction by separable approximation*, IEEE Trans. Signal Process., 57 (2019), pp. 2479–2493.