SIAM J. Matrix Anal. Appl.
Vol. 34, No. 3, pp. 946–977

HANKEL MATRIX RANK MINIMIZATION WITH APPLICATIONS TO SYSTEM IDENTIFICATION AND REALIZATION[*]

MARYAM FAZEL[†], TING KEI PONG[‡], DEFENG SUN[§], AND PAUL TSENG[‡]

(*In honor of Professor Paul Tseng—who went missing while on a kayak trip on the Jinsha river, China, on August* 13, 2009*—for his contributions to the theory and algorithms for large-scale optimization.*)

Abstract. We introduce a flexible optimization framework for nuclear norm minimization of matrices with linear structure, including Hankel, Toeplitz, and moment structures and catalog applications from diverse fields under this framework. We discuss various first-order methods for solving the resulting optimization problem, including alternating direction methods of multipliers, proximal point algorithms, and gradient projection methods. We perform computational experiments to compare these methods on system identification problems and system realization problems. For the system identification problem, the gradient projection method (accelerated by Nesterov's extrapolation techniques) and the proximal point algorithm usually outperform other first-order methods in terms of CPU time on both real and simulated data, for small and large regularization parameters, respectively, while for the system realization problem, the alternating direction method of multipliers, as applied to a certain primal reformulation, usually outperforms other first-order methods in terms of CPU time. We also study the convergence of the proximal alternating direction methods of multipliers used in this paper.

Key words. rank minimization, nuclear norm, Hankel matrix, first-order method, system identification, system realization

AMS subject classifications. 90C06, 90C25, 90C90, 93B30, 93E12

DOI. 10.1137/110853996

1. Introduction.

The matrix rank minimization problem, or minimizing the rank of a matrix subject to convex constraints, has recently attracted much interest. This problem arises in many engineering and statistical modeling applications, where notions of order, dimensionality, or complexity of a model can be expressed by the rank of an appropriate matrix. Thus, choosing the "simplest" model that is consistent with observations or data often translates into finding a matrix with the smallest rank subject to convex constraints. Rank minimization is NP-hard in general, and a popular convex heuristic for it minimizes the nuclear norm of the matrix (the sum of the singular values) instead of its rank [17, 47]. The regularized version of this problem can be written as

$$\min_X \quad \frac{1}{2}\|\mathcal{A}(X)-b\|^2+\mu\|X\|_*, \tag{1.1}$$

where $X\in\mathbb{R}^{m\times n}$ is the optimization variable and $\mathcal{A}:\mathbb{R}^{m\times n}\to\mathbb{R}^p$ is a linear map, $b\in\mathbb{R}^p$, and $\mu>0$ is the trade-off parameter between the nuclear norm and the least-squares fitting error. Problem (1.1) has been widely studied, and recently a

[*]Received by the editors November 4, 2011; accepted for publication (in revised form) by N. Mastronardi April 23, 2013; published electronically July 11, 2013.
http://www.siam.org/journals/simax/34-3/85399.html

[†]Department of Electrical Engineering, University of Washington, Seattle, WA 98195 (mfazel@ee.washington.edu).
[‡]Department of Mathematics, University of Washington, Seattle, WA 98195 (tkpong@uw.edu).
[§]Department of Mathematics and Risk Management Institute, National University of Singapore, Singapore (matsundf@nus.edu.sg).

variety of efficient algorithms have been developed [5, 9, 27, 30, 31, 38, 52]. A special case of this problem is the matrix completion problem [7, 8], which has applications in collaborative filtering and machine learning. In this problem the measurements are simply a subset of the entries of the matrix. The majority of existing work on algorithms for problem (1.1) has concentrated on this special case.

In this paper, we focus on problems where we need to find a matrix X that, in addition to being low-rank, is required to have a certain *linear structure*, for example, (block-)Hankel, (block-)Toeplitz, or moment structure. Hankel (and Toeplitz) structures arise in dynamical systems problems discussed in section 1.1, while moment structure comes up in Lasserre relaxations for minimizing polynomials [28]. We consider problem (1.1), and represent the desired structure by a linear map $X = \mathcal{H}(y)$, where y is our optimization variable. Note that if $\mathcal{H}(y)$ is a moment matrix, we need to add the constraint $\mathcal{H}(y) \succeq 0$.

1.1. Motivating applications.

1.1.1. Applications in linear dynamical systems. Linear time-invariant (LTI) systems have a long and successful history in modeling dynamical phenomena in many fields, from engineering to finance. The goal of fitting an LTI model to observed data gives rise to different classes of optimization problems, depending on whether the model is parametric or black-box, given in time or frequency domain, deterministic or stochastic, as well as on the type of data, e.g., input-output or state measurements (see, e.g., [12, 19, 34]). In all these cases, picking the appropriate model order or complexity and understanding its trade-off with the fitting or validation errors is crucial. In the problems described in this section, the system order or complexity can be expressed as the rank of a Hankel-type matrix. We discuss some of these problems in more detail in sections 4 and 5.

Minimal system realization with time-domain constraints. Consider the problem of designing a discrete-time LTI dynamical system, directly from convex specifications on the system's response in the time domain; see, e.g., [18, 32]. Such a problem arises in designing filters for signal processing and control applications. The objective is to balance the *order* of the linear system with how well the specifications are met. A low-order design is desired since, in practice, it translates into a system that is easier and cheaper to build and analyze. Typical specifications are desired rise-time, settling-time, slew-rate, and overshoot of the filter's response to a step input signal. These specifications can be expressed as upper and lower bounds on the step response over a fixed time horizon, say N time samples. Equivalently, they can be written in terms of the impulse response, which translate into linear inequality constraints on the entries of a Hankel matrix whose rank corresponds to the system order or McMillan degree; see, e.g., [51]. Using the nuclear norm heuristic for rank, we get

$$\begin{array}{ll} \min\limits_{y} & \|\mathcal{H}(y)\|_* \\ \text{s.t.} & l_i \le \sum_{k=1}^{i} y_k \le b_i, \quad i = 1, \ldots, N, \end{array} \tag{1.2}$$

where the optimization variable is $y \in \mathbb{R}^{2N-1}$ with y_i corresponding to the value of the impulse response at time i, l_i and b_i denoting the bounds on the step response given by the specifications, and $\mathcal{H}(y)$ denoting an $N \times N$ Hankel matrix; see [18] for more details. Notice that this problem is not exactly in the form of (1.1); we shall discuss how algorithms proposed in this paper can be extended to tackle this model in Appendix A.

Minimal partial realization. A related problem in linear system theory is the minimal partial realization problem for multi-input, multi-output systems: given a sequence of matrices H_k, $k = 1, \dots, N$, find a minimal state space model, described by a 3-tuple of appropriately sized matrices (A, B, C) such that $H_k = CA^{k-1}B$ [51, Chapter 6]. This problem is related to the above realization problem that handled a single input and a single output. In this problem, the order of the system (minimal size of a state-space representation) is equal to the rank of a *block-Hankel* matrix consisting of the H_k; see [33, section II.A].

Input-output system identification (system ID). Identifying a linear dynamical system given noisy and/or partial observations of its inputs and outputs, also related to time-series analysis, is a fundamental problem studied in a variety of fields [55, 56], including signal processing, control, and robotics; see, e.g., [11, 34]. We will discuss this problem and a Hankel rank formulation for it in detail in section 4.

Stochastic realization. Another fundamental problem in linear system theory is finding a minimal stochastic ARMA (autoregressive moving average) model for a vector random process, given noisy and/or partial estimates of process covariances [12, 35]. The minimal order is the rank of a block-Hankel matrix consisting of the exact covariances. This problem is discussed in detail in section 5.

1.1.2. Other applications.

Shape from moments estimation. Consider a polygonal region P in the complex plane with ordered vertices $z_1, \dots, z_m$. Complex moments of P are defined as

$$\tau_k := k(k-1)\int_P z^{k-2}dxdy, \qquad \tau_0 = \tau_1 = 0,$$

and can be expressed as $\tau_k = \sum_{i=1}^m a_i z_i^k$, where m is the number of vertices and a_i are complex constants. The problem of determining P given its complex moments has been studied in [16, 39, 50] and arises in many applications such as computer tomography, where X-ray is used to estimate moments of mass distribution, and geophysical inversion, where the goal is to estimate the shape of a region from external gravitational measurements. Note that the *number* of vertices is equal to the rank of the Hankel matrix consisting of the moments [16, 24]. In practice, often only noisy or partial measurements of the complex moments are available, and the challenge is to find a polygon with the minimum number of vertices that is consistent with the measurements. Formulating this problem as a rank minimization problem, we propose using the nuclear norm heuristic for rank. This leads to a problem of the form (1.1), where the optimization variable is the vector of complex moments τ.

Moment matrix rank minimization for polynomial optimization. Suppose $p(x)$, $x \in \mathbb{R}^n$, is a polynomial of degree d. Denote the corresponding moment matrix by $M(y)$, where y is the vectors of moments; i.e., y_i corresponds to the ith monomial; see [28, 29]. Moment matrices are important in Lasserre's hierarchy of relaxations for polynomial optimization, where a condition on the rank of the moment matrix in successive relaxations in the hierarchy determines whether the relaxation is exact; see, e.g., [29, section 5].

In the dual problem of representing a polynomial as a sum of squares of other polynomials [45], the rank of the coefficient matrix equals the minimum number of squared polynomials in the representation; thus the nuclear norm (or trace) penalty helps find simpler representations. Note that in these problems we also have an additional positive semidefinite constraint on the desired matrix.

Further applications. Another application arises in video inpainting in computer vision, where features extracted from video frames are interpolated by finding a low-rank completion to a Hankel matrix, and help reconstruct the missing frames or occluded parts of a frame [13].

Finally, our problem formulation also gives a relaxation for the problem known as the structured total least-squares problem [10] studied extensively in the controls community; see [36, 37]. Our algorithms are thus applicable to this set of problems as well. A variety of applications are discussed in [37].

1.2. Our contributions. We introduce a flexible optimization framework for nuclear norm minimization of linearly structured matrices, including Hankel, Toeplitz, and moment matrices. We identify and catalog applications from diverse fields, some of which have not been studied from this perspective before, for example, the shape from moments estimation problem. In view of the wide applicability of the model (1.1), it is important to find efficient algorithms for solving it. Along this direction, recently, Liu and Vandenberghe [32] proposed an interior-point method for solving a reformulation of problem (1.1), where they used the SDP (semidefinite programming) representation for the nuclear norm and exploited the problem structure to efficiently solve the Newton system. They applied their algorithm to the system identification and system realization problems mentioned above. They showed that the cost per iteration of the algorithm grows roughly as $\mathcal{O}(p\,qn^2)$, where $y \in \mathbb{R}^n$ and $X = \mathcal{H}(y) \in \mathbb{R}^{p\times q}$ with $y \mapsto \mathcal{H}(y)$ being injective.

In this paper, we derive various primal and dual reformulations of problem (1.1) (with $X = \mathcal{H}(y)$) and propose several first-order methods for solving the reformulations. In particular, we show that the alternating direction method of multipliers and the proximal point algorithm can be suitably applied to solving reformulations of (1.1). These methods have been widely used in the literature recently for solving (1.1), when no linear structure is imposed on X; see, e.g., [31, 58]. We discuss implementation detail of these methods in sections 3.1 and 3.2. For these methods, typically, each iteration involves a singular value decomposition whose cost grows as $\mathcal{O}(p^2q)$, where $X = \mathcal{H}(y) \in \mathbb{R}^{p\times q}$ and $p < q$; see, e.g., [25, p. 254]. Next, in section 3.3, assuming that $\mathcal{A}^*\mathcal{A}$ is invertible, we show that the (accelerated) gradient projection algorithms can be efficiently applied to solve a dual reformulation of (1.1). In this approach, each iteration also involves a singular value decomposition, and there is an explicit upper bound on the number of iterations required to attain a certain accuracy. This solution approach has been considered recently in [40] for solving the system identification problem.

To demonstrate the computational efficiency of our algorithms, we apply them to solving the input-output system identification problem and the stochastic system realization problem mentioned in the previous subsection. For the system identification problem, we consider both simulated data and real data from the DaISy database [11]. Our computational results show that the accelerated gradient projection algorithm and the proximal point algorithm usually outperform other first-order methods for this application in terms of CPU time for small and large regularization parameters, respectively. We also observe that these methods significantly outperform the interior point implementation proposed in [32]. For the system realization problem, we consider only simulated data, and our computational results show that the alternating direction method of multipliers, as applied to a certain primal reformulation of (1.1), usually outperforms other first-order methods in terms of CPU time.

In addition, in Appendix B, we establish a general convergence result that covers all versions of proximal alternating direction methods of multipliers used in this paper. Specifically, our theorem allows the use of *positive semidefinite* proximal terms under some mild assumptions, and any step length chosen from $(0, \frac{\sqrt{5}+1}{2})$. It covers many existing convergence results and allows more flexible applications of the proximal alternating direction methods of multipliers.

The rest of this paper is organized as follows. In section 1.3, we introduce notation used in this paper. We then derive primal and dual reformulations of (1.1) in section 2, and discuss several first-order methods for solving the reformulations in section 3. In sections 4 and 5, we present computational results of our first-order methods for solving the system identification and system realization problems, respectively. Concluding remarks are given in section 6. Finally, we discuss an alternative formulation for modeling the structured matrix rank minimization problem in Appendix A, and establish a convergence result that covers all versions of proximal alternating direction methods of multipliers used in this paper in Appendix B. A brief description of the classical subspace method for system identification is given in Appendix C, as a supplement to section 4.1.2.

1.3. Notation. In this paper, $\mathbb{R}^n$ denotes the n-dimensional Euclidean space. For a vector $x \in \mathbb{R}^n$, $\|x\|$ denotes the Euclidean norm of x. The set of all $m \times n$ matrices with real entries is denoted by $\mathbb{R}^{m\times n}$. For any $A \in \mathbb{R}^{m\times n}$, $\|A\|$ denotes the spectral norm of A, $\|A\|_F$ denotes the Fröbenius norm of A, $\|A\|_*$ denotes the nuclear norm of A (which is the sum of all singular values of A), and $\mathrm{vec}(A)$ denotes the column vector formed by stacking columns of A one by one. For two matrices A and B in $\mathbb{R}^{m\times n}$, $A \circ B$ denotes the Hadamard (entrywise) product of A and B. If a symmetric matrix A is positive semidefinite, we write $A \succeq 0$. Linear maps will be denoted by scripted letters. For a linear map $\mathcal{A} : \mathbb{R}^{m\times n} \to \mathbb{R}^p$, $\mathcal{A}^*$ denotes the adjoint of $\mathcal{A}$, $\|\mathcal{A}\|$ denotes the spectral norm of $\mathcal{A}$, while $\sigma_{\max}(\mathcal{A})$ and $\sigma_{\min}(\mathcal{A})$ denote the maximum and minimum singular values of $\mathcal{A}$, respectively. Finally, we denote the identity matrix and identity map by I and $\mathcal{I}$, respectively, whose dimensions should be clear from the context.

2. Basic problem formulations.

Consider the following general Hankel matrix nuclear norm minimization problem:

$$v := \min_y f(y) := \frac{1}{2}\|\mathcal{A}(y) - b\|^2 + \mu\|\mathcal{H}(y)\|_*, \tag{2.1}$$

where $\mathcal{A} : \mathbb{R}^{m\times n(j+k-1)} \to \mathbb{R}^p$ is a linear map, $b \in \mathbb{R}^p$, $y = \begin{pmatrix} y_0 & \cdots & y_{j+k-2} \end{pmatrix}$ is an $m \times n(j+k-1)$ matrix with each y_i being an $m \times n$ matrix for $i = 1, \ldots, j+k-2$, and $\mathcal{H}(y) := H_{m,n,j,k}(y)\Upsilon$ with

$$H_{m,n,j,k}(y) := \begin{pmatrix} y_0 & y_1 & \cdots & y_{k-1} \\ y_1 & y_2 & \cdots & y_k \\ \vdots & \vdots & & \vdots \\ y_{j-1} & y_j & \cdots & y_{j+k-2} \end{pmatrix} \in \mathbb{R}^{mj\times nk}$$

and $\Upsilon \in \mathbb{R}^{nk\times q}$. We assume without loss of generality that $\sigma_{\max}(\Upsilon) \leq 1$. In this section, we derive primal and dual reformulations of (2.1).

First, using the substitutions $Y = -\mathcal{H}(y)$ and $z = b - \mathcal{A}(y)$, problem (2.1) can be reformulated as

$$\begin{array}{ll} \min\limits_{Y,z,y} & \frac{1}{2}\|z\|^2 + \mu\|Y\|_* \\ \text{s.t.} & Y + \mathcal{H}(y) = 0, \\ & z + \mathcal{A}(y) = b. \end{array} \tag{2.2}$$

From the reformulation (2.2), we can easily write down the Lagrange dual to (2.2) (and hence, equivalently, to (2.1)) as follows:

$$\begin{aligned} v &= \min_{Y,z,y}\max_{\gamma,\Lambda}\left\{\frac{1}{2}\|z\|^2 + \mu\|Y\|_* - \langle \Lambda, Y + \mathcal{H}(y)\rangle - \langle \gamma, z + \mathcal{A}(y) - b\rangle\right\} \\ &= \max_{\gamma,\Lambda}\min_{Y,z,y}\left\{\frac{1}{2}\|z\|^2 + \mu\|Y\|_* - \langle \Lambda, Y + \mathcal{H}(y)\rangle - \langle \gamma, z + \mathcal{A}(y) - b\rangle\right\} \\ &= \max_{\gamma,\Lambda}\min_{Y,z,y}\left\{\frac{1}{2}\|z\|^2 - \langle \gamma, z\rangle + \mu\|Y\|_* - \langle \Lambda, Y\rangle - \langle \mathcal{H}^*(\Lambda) + \mathcal{A}^*(\gamma), y\rangle + \langle b, \gamma\rangle\right\} \\ &= \max_{\gamma,\Lambda}\left\{-\frac{1}{2}\|\gamma\|^2 + \langle \gamma, b\rangle :\ \mathcal{H}^*(\Lambda) + \mathcal{A}^*(\gamma) = 0,\ \Lambda^T\Lambda \preceq \mu^2 I\right\}, \end{aligned}$$

where the second equality holds because of strong duality [48, Corollary 28.2.2]. The dual problem can thus be rewritten as the minimization problem

$$\begin{array}{ll} \min\limits_{\gamma,\Lambda} & d(\gamma) := \frac{1}{2}\|\gamma\|^2 - b^T\gamma \\ \text{s.t.} & \mathcal{H}^*(\Lambda) + \mathcal{A}^*(\gamma) = 0, \\ & \Lambda^T\Lambda \preceq \mu^2 I. \end{array} \tag{2.3}$$

Alternatively, noting the fact that the nuclear norm is just the dual norm of the spectral norm, one can derive a reduced dual problem as follows:

$$\begin{aligned} v &= \min_y \frac{1}{2}\|\mathcal{A}(y) - b\|^2 + \mu\|\mathcal{H}(y)\|_* = \min_y \max_{\Lambda^T\Lambda\preceq\mu^2 I} \frac{1}{2}\|\mathcal{A}(y) - b\|^2 - \langle \Lambda, \mathcal{H}(y)\rangle \\ &= -\min_{\Lambda^T\Lambda\preceq\mu^2 I} d_2(\Lambda) := \sup_y\left\{\langle \Lambda, \mathcal{H}(y)\rangle - \frac{1}{2}\|\mathcal{A}(y) - b\|^2\right\}, \end{aligned} \tag{2.4}$$

where the third equality holds because of the compactness of the spectral norm ball [48, Corollary 37.3.2]. In the special case when $\mathcal{A}^*\mathcal{A}$ is invertible, the function d_2 has a closed form representation. Indeed, in this case, writing $\mathcal{R}(\Lambda) := (\mathcal{A}^*\mathcal{A})^{-1}\mathcal{H}^*(\Lambda)$ and $\bar{b} := (\mathcal{A}^*\mathcal{A})^{-1}\mathcal{A}^*b$, we obtain that

$$d_2(\Lambda) = \frac{1}{2}(\langle \mathcal{H}^*(\Lambda), \mathcal{R}(\Lambda)\rangle + 2\langle \mathcal{H}^*(\Lambda), \bar{b}\rangle + \langle \mathcal{A}^*b, \bar{b}\rangle - \|b\|^2).$$

Hence, when $\mathcal{A}^*\mathcal{A}$ is invertible, the reduced dual problem (2.4) is equivalent to

$$\begin{array}{ll} \min\limits_{\Lambda} & \frac{1}{2}\langle \mathcal{H}^*(\Lambda), \mathcal{R}(\Lambda)\rangle + \langle \mathcal{H}^*(\Lambda), \bar{b}\rangle + \frac{1}{2}\langle \mathcal{A}^*b, \bar{b}\rangle - \frac{1}{2}\|b\|^2 \\ \text{s.t.} & \Lambda^T\Lambda \preceq \mu^2 I. \end{array} \tag{2.5}$$

Before ending this section, we derive an upper bound on the spectral norm of $\mathcal{H}^*$.

Notice that $\mathcal{H}^*(\Lambda) = H^*_{m,n,j,k}(\Lambda\Upsilon^T)$, where for any $W \in \mathbb{R}^{mj\times nk}$

$$\begin{aligned} H^*_{m,n,j,k}(W) &= H^*_{m,n,j,k}\begin{pmatrix} w_{00} & w_{01} & \cdots & w_{0,k-1} \\ w_{10} & w_{11} & \cdots & w_{1,k-1} \\ \vdots & \vdots & & \vdots \\ w_{j-1,0} & w_{j-1,1} & \cdots & w_{j-1,k-1} \end{pmatrix} \\ &= \begin{pmatrix} w_{00} & w_{01}+w_{10} & w_{02}+w_{11}+w_{20} & \cdots & w_{j-1,k-1} \end{pmatrix} \\ &\in \mathbb{R}^{m\times n(j+k-2)}. \end{aligned} \tag{2.6}$$

It follows from (2.6) that

$$\begin{aligned} &\|H^*_{m,n,j,k}(W)\|_F^2 \\ &\quad = \|w_{00}\|_F^2 + \|w_{01}+w_{10}\|_F^2 + \|w_{02}+w_{11}+w_{20}\|_F^2 + \cdots + \|w_{j-1,k-1}\|_F^2 \\ &\quad \le \|w_{00}\|_F^2 + 2(\|w_{01}\|_F^2 + \|w_{10}\|_F^2) + \cdots + \|w_{j-1,k-1}\|_F^2 \le \mathbf{r}\|W\|_F^2, \end{aligned}$$

where $\mathbf{r} := \min\{j,k\}$. Combining this estimate with $\sigma_{\max}(\Upsilon) \le 1$, we obtain that

$$\|\mathcal{H}^*(\Lambda)\|_F^2 \le \mathbf{r}\|\Lambda\Upsilon^T\|_F^2 \le \mathbf{r}\|\Lambda\|_F^2,$$

and thus the spectral norm of $\mathcal{H}^*$ is less than or equal to $\sqrt{\mathbf{r}}$.

3. Algorithms.

In this section, we discuss several first-order methods for solving (2.1) and (2.3).

3.1. Alternating direction method of multipliers.

In this section, we discuss how the alternating direction method of multipliers (ADMM) can be applied to solving (2.1) and (2.3). To apply the ADMM for solving (2.1), we first introduce the augmented Lagrangian function

$$L_\beta(Y,y,\Lambda) = \frac{1}{2}\|\mathcal{A}(y)-b\|^2 + \mu\|Y\|_* - \langle \Lambda, Y+\mathcal{H}(y)\rangle + \frac{\beta}{2}\|Y+\mathcal{H}(y)\|_F^2$$

for each $\beta > 0$. In the classical ADMM (see, e.g., [2, section 3.4.4]), in each iteration, we minimize L_β with respect to Y and then with respect to y, followed by an update of the multiplier Λ. While minimizing L_β with respect to Y admits an easy closed form solution, minimizing L_β with respect to y does not usually have a simple closed form solution due to the complicated quadratic terms. One way to resolve this is to add a proximal term with (semi)norm induced by a suitable positive (semi)definite matrix to "cancel" out the complicated parts. In this approach, we update

$$y^{k+1} = \arg\min_y \left\{ L_\beta(Y^{k+1},y,\Lambda^k) + \frac{\beta}{2}\|y-y^k\|_{\mathcal{Q}_0}^2 \right\},$$

where $\|\cdot\|_{\mathcal{Q}_0}$ is the (semi)norm induced by the (semi-)inner product $x^T\mathcal{Q}_0 x$,

$$\mathcal{Q}_0 := \left(\mathbf{r} + \frac{(\sigma_{\max}(\mathcal{A}))^2}{\beta}\right)\mathcal{I} - \left(\mathcal{H}^*\mathcal{H} + \frac{1}{\beta}\mathcal{A}^*\mathcal{A}\right) \succeq 0. \tag{3.1}$$

The convergence analysis of this approach has been considered in [26], for example, in the context of variational inequalities. The main motivation for introducing the proximal terms in [26] is to weaken the imposed convergence conditions rather than for the sake of cancellation, as was more recently explained in [59]. All existing convergence results require $\mathcal{Q}_0$ to be positive definite and hence are not general enough to cover our proposed method. In Appendix B, we present a new convergence analysis of this approach. We now present our algorithm as follows.

```
Primal ADMM.
Step 0. Input $(y^0, \Lambda^0)$, $\beta > 0$, $\sigma = \frac{\beta}{\beta\mathbf{r}+(\sigma_{\max}(\mathcal{A}))^2}$, and $\tau \in (0, \frac{\sqrt{5}+1}{2})$.
Step 1. Compute the SVD:
        $-\mathcal{H}(y^k) + \frac{\Lambda^k}{\beta} = U\Sigma V^T$,
    where $U$ and $V$ have orthogonal columns, $\Sigma$ is diagonal. Set
        $Y^{k+1} = U \max\left\{\Sigma - \frac{\mu}{\beta}I, 0\right\} V^T$,
        $y^{k+1} = y^k - \sigma\left(\mathcal{H}^*\left(-\frac{1}{\beta}\Lambda^k + \mathcal{H}(y^k) + Y^{k+1}\right) + \frac{1}{\beta}\mathcal{A}^*(\mathcal{A}(y^k) - b)\right)$,
        $\Lambda^{k+1} = \Lambda^k - \tau\beta(Y^{k+1} + \mathcal{H}(y^{k+1}))$.
Step 2. If a termination criterion is not met, go to Step 1.
```

The ADMM can also be applied to solving the dual problem (2.3). In this approach, we use the following augmented Lagrangian function:

$$l_\beta(\gamma, \Lambda, y) = \frac{1}{2}\|\gamma\|^2 - b^T\gamma + \langle y, \mathcal{H}^*(\Lambda) + \mathcal{A}^*(\gamma)\rangle + \frac{\beta}{2}\|\mathcal{H}^*(\Lambda) + \mathcal{A}^*(\gamma)\|_F^2$$

for some $\beta > 0$.

The algorithm is described as follows.

```
Dual ADMM.
Step 0. Input $(y^0, \Lambda^0)$, $\beta > 0$, $\sigma_1 = \frac{1}{(\sigma_{\max}(\mathcal{A}))^2}$, $\sigma_2 = \frac{1}{\mathbf{r}}$, and $\tau \in (0, \frac{\sqrt{5}+1}{2})$.
Step 1. Set
        $\gamma^{k+1} = \frac{\sigma_1}{\sigma_1 + \beta}\left(b + \beta\frac{\gamma^k}{\sigma_1} - \mathcal{A}(y^k) - \beta\mathcal{A}(\mathcal{H}^*(\Lambda^k) + \mathcal{A}^*(\gamma^k))\right)$.
    Compute the SVD:
        $\Lambda^k - \sigma_2\left(\frac{1}{\beta}\mathcal{H}(y^k) + \mathcal{H}(\mathcal{H}^*(\Lambda^k) + \mathcal{A}^*(\gamma^{k+1}))\right) = U\Sigma V^T$,
    where $U$ and $V$ have orthogonal columns, $\Sigma$ is diagonal. Set
        $\Lambda^{k+1} = U\min\{\Sigma, \mu I\}V^T$,
        $y^{k+1} = y^k + \tau\beta(\mathcal{H}^*(\Lambda^{k+1}) + \mathcal{A}^*(\gamma^{k+1}))$.
Step 2. If a termination criterion is not met, go to Step 1.
```

Indeed, since the minimizer of l_β with respect to γ is usually not easy to find and the minimizer with respect to Λ does not always admit a simple closed form solution, as before, we add suitable proximal terms to cancel out complicated terms. More precisely, in the above algorithm, we update

$$\gamma^{k+1} := \arg\min_{\gamma} \left\{ l_\beta(\gamma, \Lambda^k, y^k) + \frac{\beta}{2}\|\gamma - \gamma^k\|^2_{\mathcal{Q}_1} \right\},$$
$$\Lambda^{k+1} := \arg\min_{\Lambda^T\Lambda \preceq \mu^2 I} \left\{ l_\beta(\gamma^{k+1}, \Lambda, y^k) + \frac{\beta}{2}\|\Lambda - \Lambda^k\|^2_{\mathcal{Q}_2} \right\},$$

where

$$\mathcal{Q}_1 := (\sigma_{\max}(\mathcal{A}))^2\mathcal{I} - \mathcal{A}\mathcal{A}^* \succeq 0, \quad \mathcal{Q}_2 := \mathbf{r}\mathcal{I} - \mathcal{H}\mathcal{H}^* \succeq 0. \tag{3.2}$$

From Theorem B.1 in the appendix, for the sequence $\{(Y^k, y^k, \Lambda^k)\}$ generated from Primal ADMM, $\{y^k\}$ converges to a solution of the problem (2.1), while $\{\Lambda^k\}$ converges to a solution of the problem (2.4). Similarly, for the sequence $\{(y^k, \gamma^k, \Lambda^k)\}$ generated from Dual ADMM, $\{y^k\}$ converges to a solution of (2.1), and $\{(\gamma^k, \Lambda^k)\}$ converges to a solution of (2.3).

3.2. Dual proximal point algorithm. In this section, we discuss how the proximal point algorithm (PPA) can be applied to solving the dual problem. We shall make use of the reduced dual problem (2.4). Fix $\lambda > 0$. For any Λ, define the Moreau–Yosida regularization of d_2 at Λ associated with λ by

$$G_\lambda(\Lambda) = \min_{\Gamma^T\Gamma \preceq \mu^2 I} d_2(\Gamma) + \frac{1}{2\lambda}\|\Gamma - \Lambda\|_F^2.$$

Using this definition and the definition of d_2 in (2.4), we obtain that

$$\begin{aligned} G_\lambda(\Lambda) &= \min_{\Gamma^T\Gamma \preceq \mu^2 I} \sup_y \left\{ \langle \Gamma, \mathcal{H}(y)\rangle - \frac{1}{2}\|\mathcal{A}(y) - b\|^2 \right\} + \frac{1}{2\lambda}\|\Gamma - \Lambda\|_F^2 \\ &= \sup_y \left\{ \min_{\Gamma^T\Gamma \preceq \mu^2 I} \left\{ \langle \Gamma, \mathcal{H}(y)\rangle + \frac{1}{2\lambda}\|\Gamma - \Lambda\|_F^2 \right\} - \frac{1}{2}\|\mathcal{A}(y) - b\|^2 \right\}, \end{aligned}$$

where the second equality holds due to the compactness of the spectral norm ball [48, Corollary 37.3.2]. Furthermore, it is not hard to show that

$$\begin{aligned} &\min_{\Gamma^T\Gamma \preceq \mu^2 I} \left\{ \langle \mathcal{H}(y), \Gamma\rangle + \frac{1}{2\lambda}\|\Lambda - \Gamma\|_F^2 \right\} \\ &\qquad = \langle \mathcal{H}(y), \Lambda\rangle - \frac{\lambda}{2}\|\mathcal{H}(y)\|_F^2 + \frac{1}{2\lambda}\|\mathcal{P}_\mu(\Lambda - \lambda\mathcal{H}(y))\|_F^2, \end{aligned}$$

where $\mathcal{P}_\mu(W)$ is the unique optimal solution to the following convex optimization problem:

$$\min_Z \|Z\|_* + \frac{1}{2\mu}\|Z - W\|_F^2.$$

Thus, $G_\lambda(\Lambda) = \sup_y \Theta_\lambda(y; \Lambda)$, where

$$\Theta_\lambda(y;\Lambda) := \langle \mathcal{H}(y), \Lambda\rangle - \frac{\lambda}{2}\|\mathcal{H}(y)\|_F^2 + \frac{1}{2\lambda}\|\mathcal{P}_\mu(\Lambda - \lambda\mathcal{H}(y))\|_F^2 - \frac{1}{2}\|\mathcal{A}(y) - b\|^2.$$

Recall from the Moreau–Yosida regularization theory that $\mathcal{P}_\mu(\cdot)$ is globally Lipschitz continuous with modulus 1 and that $\|\mathcal{P}_\mu(\cdot)\|_F^2$ is continuously differentiable with

$\nabla(\|\mathcal{P}_\mu(Y)\|_F^2) = 2\mathcal{P}_\mu(Y)$. Hence, $\Theta_\lambda(\cdot;\Lambda)$ is a continuously differentiable concave function in y with

$$\nabla_y\Theta_\lambda(y;\Lambda) = \mathcal{H}^*(\Lambda - \lambda\mathcal{H}(y)) - \mathcal{H}^*\mathcal{P}_\mu(\Lambda - \lambda\mathcal{H}(y)) - \mathcal{A}^*(\mathcal{A}(y) - b).$$

In addition, we have that

$$\begin{aligned}&\|\nabla_y\Theta_\lambda(y';\Lambda) - \nabla_y\Theta_\lambda(y;\Lambda)\|_F \\ &\le (\lambda\|\mathcal{H}^*\mathcal{H}\| + \|\mathcal{A}^*\mathcal{A}\|)\,\|y' - y\|_F + \|\mathcal{H}^*\mathcal{P}_\mu(\Lambda - \lambda\mathcal{H}(y')) - \mathcal{H}^*\mathcal{P}_\mu(\Lambda - \lambda\mathcal{H}(y))\|_F \\ &\le \left(\lambda\mathbf{r} + (\sigma_{\max}(\mathcal{A}))^2\right)\|y' - y\|_F + \lambda\mathbf{r}\|y' - y\|_F,\end{aligned}$$

which implies that $\nabla_y\Theta_\lambda(\cdot;\Lambda)$ is Lipschitz continuous with Lipschitz modulus

$$2\lambda\mathbf{r} + (\sigma_{\max}(\mathcal{A}))^2. \tag{3.3}$$

We are now ready to describe the PPA for solving the dual problem (2.4).

```
Dual PPA.
Step 0. Input $(y^0, \Lambda^0)$ and $\lambda_0 > 0$.
Step 1. (Find an approximate maximizer $y^{k+1} \approx \arg\max \Theta_{\lambda_k}(y;\Lambda^k)$.)
        Input $u^0 := y^k$. Let $s_0, tol_k > 0$, $1 > t, \sigma > 0$.
        While $\|\nabla_y\Theta_{\lambda_k}(u^l;\Lambda^k)\| > tol_k$ do
          (a) Let $\bar{s}_l$ be the largest element of $\{s_\nu, ts_\nu, t^2 s_\nu, \ldots\}$ satisfying
                  $\Theta_{\lambda_k}(u^l[s];\Lambda^k) \ge \Theta_{\lambda_k}(u^l;\Lambda^k) + \sigma s\|\nabla_y\Theta_{\lambda_k}(u^l;\Lambda^k)\|_F^2$,
              where $u^l[s] = u^l + s\nabla_y\Theta_{\lambda_k}(u^l;\Lambda^k)$.
          (b) Set $u^{l+1} \leftarrow u^l[\bar{s}_l]$ and update $s_\nu$.
        End (while)
        Set $y^{k+1} \leftarrow u^{l+1}$.
Step 2. Compute the SVD:
                  $\Lambda^k - \lambda_k\mathcal{H}(y^{k+1}) = U\Sigma V^T$.
        Set
                  $\Lambda^{k+1} = U\min(\Sigma, \mu I)V^T$.
Step 3. If a termination criterion is not met, update $\lambda_k$. Go to Step 1.
```

3.3. Dual gradient projection methods. In this section we assume that $\mathcal{A}^*\mathcal{A}$ is invertible. Recall that in this case the dual of (2.1) is given by (2.5). Moreover, we have

$$\begin{aligned}&\|\nabla d_2(\Lambda_1) - \nabla d_2(\Lambda_2)\|_F^2 \\ &\quad = \|\mathcal{H}((\mathcal{A}^*\mathcal{A})^{-1}\mathcal{H}^*(\Lambda_1 - \Lambda_2))\|_F^2 \le \mathbf{r}\|(\mathcal{A}^*\mathcal{A})^{-1}\mathcal{H}^*(\Lambda_1 - \Lambda_2)\|_F^2 \\ &\quad \le \frac{\mathbf{r}}{(\sigma_{\min}(\mathcal{A}^*\mathcal{A}))^2}\|\mathcal{H}^*(\Lambda_1 - \Lambda_2)\|_F^2 \le \left(\frac{\mathbf{r}}{\sigma_{\min}(\mathcal{A}^*\mathcal{A})}\right)^2\|\Lambda_1 - \Lambda_2\|_F^2.\end{aligned}$$

This shows that the gradient of d_2 is Lipschitz continuous with Lipschitz constant

$$L_{\mathrm{D}} := \frac{\mathbf{r}}{\sigma_{\min}(\mathcal{A}^*\mathcal{A})}.$$

Since the projection onto the feasible set of (2.5) is simple, we can apply the gradient projection (GP) methods to solving (2.5). One simple version is described as follows.

```
Dual GP.
Step 0. Input $\Lambda^0$ such that ${\Lambda^0}^T\Lambda^0 \preceq \mu^2 I$ and $L > \frac{L_D}{2}$.
Step 1. Compute the SVD:
            $$\Lambda^k - \frac{1}{L}\nabla d_2(\Lambda^k) = U\Sigma V^T,$$
        where $U$ and $V$ have orthogonal columns, $\Sigma$ is diagonal. Set
            $$\Lambda^{k+1} = U\min\{\Sigma, \mu I\}V^T.$$
Step 2. If a termination criterion is not met, go to Step 1.
```

The iterate generated by the Dual GP satisfies

$$d_2(\Lambda^k) - v = \mathcal{O}\left(\frac{L}{k}\right);$$

see, e.g., [53, Theorem 1]. Hence, for faster convergence, a smaller L is favored. Also, note that if y^* and Λ^* are solutions to (2.1) and (2.5), respectively, then we can see from (2.4) that

$$y^* = \mathcal{R}(\Lambda^*) + \bar{b}$$

where $\mathcal{R}(\Lambda) := (\mathcal{A}^*\mathcal{A})^{-1}\mathcal{H}^*(\Lambda)$ and $\bar{b} := (\mathcal{A}^*\mathcal{A})^{-1}\mathcal{A}^* b$. Hence, the sequence

$$y^k := \mathcal{R}(\Lambda^k) + \bar{b} \tag{3.4}$$

can be used to check for termination; see section 4.1.

The Dual GP can be accelerated using Nesterov's extrapolation techniques (see, e.g., [41, 42, 43, 44, 53]). This method has also been used in [46, 52] for nuclear norm-related problems. The method, which we call the dual accelerated gradient projection (Dual AGP), is described below.

```
Dual AGP.
Step 0. Input $\Lambda^0$ such that ${\Lambda^0}^T\Lambda^0 \preceq \mu^2 I$ and $L \geq L_D$. Initialize $\Lambda^{-1} = \Lambda^0$
        and $\theta_{-1} = \theta_0 = 1$. Go to Step 1.
Step 1. Set
            $$\Psi^k = \Lambda^k + \left(\frac{\theta_k}{\theta_{k-1}} - \theta_k\right)(\Lambda^k - \Lambda^{k-1}).$$
        Compute the SVD:
            $$\Psi^k - \frac{1}{L}\nabla d_2(\Psi^k) = U\Sigma V^T,$$
        where $U$ and $V$ have orthogonal columns, $\Sigma$ is diagonal. Update
            $$\Lambda^{k+1} = U\min\{\Sigma, \mu I\}V^T, \quad \theta_{k+1} = \frac{\sqrt{{\theta_k}^4 + 4{\theta_k}^2} - {\theta_k}^2}{2}.$$
Step 2. If a termination criterion is not met, go to Step 1.
```

The sequence generated from the Dual AGP satisfies

$$d_2(\Lambda^k) - v = \mathcal{O}\left(\frac{L}{k^2}\right);$$

see, e.g., [53, Theorem 1]. Hence, a smaller L is favored for faster convergence. Furthermore, a suitable primal variable at each iteration can be generated similarly as in (3.4).

4. System identification.

In this section, we consider the problem of identifying a linear dynamical system from observations of its inputs and outputs. We focus on "output error" system identification, meaning that the output may be noisy. Given a sequence of inputs $u_t \in \mathbb{R}^p$ and measured (noisy) outputs $\tilde{y}_t \in \mathbb{R}^m$, $t = 0, \ldots, N$, the goal is to find a discrete-time, linear time-invariant state space model,

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t, \\ y_t &= Cx_t + Du_t, \end{aligned} \tag{4.1}$$

that satisfies $y_t \approx \tilde{y}_t$ and is low-order (i.e., corresponds to a low-dimensional state vector $x_t \in \mathbb{R}^r$). To determine the model, we need to find the A, B, C, D matrices, the initial state x_0, and the model order r. As described in [32, eq. (23)] (see also Appendix C, [33], and more classically [54, p. 36]), under reasonable assumptions, the minimal model order is equal to the rank of the matrix $H_{m,1,r+1,N+1-r}(y)U^\perp$, where $U^\perp \in \mathbb{R}^{(N+1-r)\times q}$ is a matrix whose columns form an orthogonal basis of the null space of $H_{p,1,r+1,N+1-r}(u)$ and u is the input to the system (see [32] for further details).

Upon relaxing the rank function to the nuclear norm, the trade-off between the fitting error and the nuclear norm (which is a proxy for the model order) is given by the following optimization problem:

$$\min_y \frac{1}{2}\|y - \tilde{y}\|_F^2 + \mu\|H_{m,1,r+1,N+1-r}(y)U^\perp\|_*, \tag{4.2}$$

where $\tilde{y} \in \mathbb{R}^{m\times(N+1)}$, $N \geq 1$, $r \geq 0$, and $\mu > 0$. This problem corresponds to (2.1) with $\Upsilon = U^\perp$, $\mathcal{A}(y) = \mathrm{vec}(y)$, and $b = \mathrm{vec}(\tilde{y})$. Thus

$$\sigma_{\max}(\mathcal{A}) = \sigma_{\max}(\Upsilon) = 1, \quad \mathcal{H}^*(\Lambda) = H^*_{m,1,r+1,N+1-r}(\Lambda(U^\perp)^T).$$

Note that the solution set of (4.2) is nonempty since the function $y \mapsto \|y - \tilde{y}\|_F^2$ is coercive.

4.1. Computational results.

In this section, we compare different algorithms for solving (4.2) on random and real data. Specifically, we consider Primal and Dual ADMM, Dual PPA, Dual GP, and Dual AGP. We note that since $\mathcal{A} = \mathcal{I}$, one could also set $\mathcal{Q}_0 = \mathbf{r}\mathcal{I} - \mathcal{H}^*\mathcal{H}$ for Primal ADMM; however, it is not hard to show that this variant gives the same iterate as Primal ADMM, which uses (3.1).

We initialize all algorithms except Dual PPA at the origin, where the latter algorithm is initialized at an approximate solution obtained from Dual AGP.[1] We terminate the algorithms by checking the relative duality gap with $tol = 10^{-4}$, i.e.,

$$\frac{\min_{t\in\mathcal{K}^k}\{f(y^t), f(\hat{y}^t)\} + \max_{t\in\mathcal{K}^k} d_2(\mathcal{P}(\Lambda^t))}{\max\{1, |\max_{t\in\mathcal{K}^k} d_2(\mathcal{P}(\Lambda^t))|\}} < 10^{-4}, \tag{4.3}$$

[1] We terminate Dual AGP by checking the relative duality gap with $tol = 5\times10^{-3}$, checked every 10 iterations. We also terminate the algorithm early if the change in Fröbenius norm of successive iterates is small ($< 10^{-8}$ for each variable) or the maximum number of iteration hits 2000.

where $\{(y^k, \Lambda^k)\}$ are defined in sections 3.1, 3.2, and 3.3; $\mathcal{P}(\Lambda^k)$ is the projection of Λ^k onto the spectral norm ball with radius μ; and $\hat{y}^k := \tilde{y} + \mathcal{H}^*(\mathcal{P}(\Lambda^k)))$.[2] We set $\mathcal{K}^k = \{1 \le \nu \le k : \ \nu \text{ divides } 10\}$ for all algorithms except for Dual PPA and check (4.3) every 10 iterations. For Dual PPA, we set $\mathcal{K}^k = \{1 \le \nu \le k\}$ and check (4.3) every iteration since the relative duality gap is needed for updating λ_k and tol_k.[3] We also terminate the algorithms early if the change in Fröbenius norm of successive iterates is small ($< 10^{-8}$ for each variable) or the maximum number of iteration hits 2000.

We set $\beta = \frac{\mu \mathbf{r}}{2\sigma_{\max}(\tilde{y})}$ for Primal ADMM, $\beta = \frac{\sigma_{\max}(\tilde{y})}{16\mu \mathbf{r}}$ for Dual ADMM, and take $\tau = 1.61$ for the ADMMs. We set $L = \frac{L_D}{1.95}$ for Dual GP and $L = L_D$ for Dual AGP. All codes are written in MATLAB and run on a Sunfire X4440 with 4 Opteron 8384 quad-core CPUs and 32G of RAM, equipped with CentOS 5.8 and MATLAB 7.12.

4.1.1. Random data.

We randomly generate a matrix $u = \begin{pmatrix} u_0 & \cdots & u_N \end{pmatrix} \in \mathbb{R}^{p\times(N+1)}$ with standard Gaussian entries (i.e., 0 mean and variance 1), and let $\bar{r}$ be the true order of the system. We then generate matrices $A \in \mathbb{R}^{\bar{r}\times\bar{r}}$, $B \in \mathbb{R}^{\bar{r}\times p}$, $C \in \mathbb{R}^{m\times\bar{r}}$, and $D \in \mathbb{R}^{m\times p}$ with i.i.d. standard Gaussian entries and normalize them to have spectral norm 1. We also generate a vector $x_0 \in \mathbb{R}^{\bar{r}}$, again with standard Gaussian entries. The output $\bar{y} = \begin{pmatrix} \bar{y}_0 & \cdots & \bar{y}_N \end{pmatrix} \in \mathbb{R}^{m\times(N+1)}$ is then generated using a state-space model: for each $t = 0, \ldots, N$,

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t, \\ \bar{y}_t &= Cx_t + Du_t. \end{aligned}$$

To model the measurement noise, we add noise to the output $\bar{y}$ to get $\tilde{y} = \bar{y} + \sigma\epsilon$, where ϵ has Gaussian entries with mean 0 and variance 1, and $\sigma > 0$. Finally, $U^\perp$ is a matrix whose columns form an orthonormal basis of the nullspace of $H_{p,1,2\bar{r}+2,N-2\bar{r}}(u)$. Note that in theory we require the r used in determining the size of the Hankel matrix to be larger than the true order of the system. However, in practice, we often don't know the true system order and only have a guess or estimate for it. Therefore, when we set the size of the Hankel matrix in our problem, as a rule of thumb, we use roughly twice the estimated order; i.e., $r = 2\bar{r} + 1$.

In the tests below, we consider $p = 5$, $m = 5, 10$, $\bar{r} = 10, 20$, and $N + 1 = 2000, 4000$. We pick $\sigma = 5 \times 10^{-2}$, which corresponds roughly to 5% noise. The statistics of the test problems used are reported in Table 4.1.[4] We run our algorithms for $\mu = 10^{-2}, 10^{-1}, 1, 10$. Our computational results are reported in Table 4.2, where **iter** stands for the number of iterations, **cpu** is the CPU time taken, and **obj** is the primal objective value at termination corresponding to each algorithm. The words "Primal" and "Dual" are abbreviated as "P." and "D.", respectively. For Dual PPA, the CPU time and number of iterations for Dual AGP used for initialization are in parenthesis, and the CPU time not in parenthesis refers to the total runtime. The word "max" denotes the maximum number of iterations. The fastest algorithm(s) in each instance is highlighted in bold. We see that the gradient projection algorithms usually work best when μ is small, with Dual AGP usually more robust (i.e., solving

[2]This is modeled after (3.4). Thus, we have $\hat{y}^k = y^k$ for Dual GP and Dual AGP.

[3]For Dual PPA, we set $t = 0.3$, $\sigma = 10^{-4}$, $s_0 = \frac{1.95}{L}$, with L given in (3.3). For each $\nu \ge 1$, we set $s_\nu = 1.11 s_{\nu-1}$ if the maximum stepsize $s_{\nu-1}$ was used in previous line-search, and fix it otherwise. The parameter λ_k is initialized at $\lambda_0 = 1$ and is doubled based on changes in gap_k, the relative duality gap in the kth outer iteration. The tol_k decreases from 0.04 based on the change of gap_k and is bounded below by 10^{-3}.

[4]Note that each of our algorithm requires an SVD of a matrix of size $m(r+1) \times q$ in each iteration.

TABLE 4.1
Parameters for the randomly generated test problems.

P	$N+1$	$\bar{r}$	m	q
1	2000	10	5	1869
2	2000	10	10	1869
3	2000	20	5	1749
4	2000	20	10	1749
5	4000	10	5	3869
6	4000	10	10	3869
7	4000	20	5	3749
8	4000	20	10	3749

TABLE 4.2
Computational results for problems from Table 4.1. The fastest algorithm is highlighted in bold.

		D. GP	D. AGP	D. PPA	P. ADMM	D. ADMM
P	μ	iter/cpu/obj	iter/cpu/obj	iter/cpu/obj	iter/cpu/obj	iter/cpu/obj
1	0.01	**6/0.7/3.76e+0**	9/1.1/3.76e+0	1(9)/1.4(1.0)/3.76e+0	30/3.8/3.76e+0	10/1.2/3.76e+0
	0.10	**10/1.1/2.87e+1**	**10/1.2/2.87e+1**	1(10)/1.6(1.2)/2.87e+1	30/3.8/2.87e+1	20/2.3/2.87e+1
	1.00	150/16.0/1.70e+2	70/7.8/1.70e+2	10(20)/6.1(2.3)/1.70e+2	40/5.1/1.70e+2	**40/4.5/1.70e+2**
	10.00	1310/139.2/1.01e+3	240/26.5/1.01e+3	33(50)/22.5(5.6)/1.01e+3	190/24.0/1.01e+3	**180/19.6/1.01e+3**
2	0.01	**7/1.8/5.19e+0**	10/2.8/5.19e+0	1(10)/4.3(2.7)/5.19e+0	20/5.9/5.19e+0	10/2.7/5.19e+0
	0.10	20/5.1/3.29e+1	20/5.2/3.29e+1	**1(10)/4.1(2.7)/3.29e+1**	30/8.8/3.29e+1	30/7.8/3.29e+1
	1.00	210/51.5/1.05e+2	90/22.9/1.05e+2	18(20)/19.7(5.2)/1.05e+2	50/14.7/1.05e+2	**50/12.9/1.05e+2**
	10.00	900/222.0/4.26e+2	420/106.9/4.26e+2	**17(70)/37.0(17.9)/4.26e+2**	240/70.3/4.26e+2	230/58.8/4.26e+2
3	0.01	**9/2.1/4.57e+0**	10/2.4/4.57e+0	1(10)/3.0(2.4)/4.57e+0	30/7.9/4.57e+0	10/2.4/4.57e+0
	0.10	380/84.4/2.06e+1	**70/15.9/2.06e+1**	22(20)/19.9(4.7)/2.06e+1	240/63.3/2.06e+1	510/116.0/2.06e+1
	1.00	730/163.0/8.60e+1	180/40.6/8.60e+1	17(40)/27.8(8.9)/8.60e+1	130/33.4/8.60e+1	**100/22.3/8.60e+1**
	10.00	max/429.4/3.70e+2	420/94.6/3.70e+2	**18(140)/65.6(31.5)/3.70e+2**	620/162.8/3.70e+2	500/111.6/3.70e+2
4	0.01	**9/4.7/8.69e+0**	10/5.6/8.69e+0	1(10)/7.2(5.7)/8.69e+0	20/12.2/8.69e+0	10/5.8/8.69e+0
	0.10	460/243.2/3.65e+1	**80/43.5/3.65e+1**	22(20)/50.0(11.0)/3.65e+1	290/180.6/3.65e+1	620/337.4/3.65e+1
	1.00	400/210.0/1.33e+2	170/91.5/1.33e+2	13(30)/61.2(16.4)/1.33e+2	130/ 81.4/1.33e+2	**80/43.6/1.33e+2**
	10.00	max/1043.7/6.19e+2	470/251.6/6.19e+2	**20(100)/162.1(53.2)/6.19e+2**	510/315.9/6.19e+2	410/221.5/6.19e+2
5	0.01	**6/1.7/4.67e+0**	8/2.4/4.67e+0	1(8)/4.0(2.3)/4.67e+0	30/9.0/4.67e+0	10/2.9/4.67e+0
	0.10	**10/2.8/3.73e+1**	**10/3.0/3.73e+1**	1(10)/3.8(2.9)/3.73e+1	20/6.0/3.73e+1	**10/3.0/3.73e+1**
	1.00	150/36.6/1.75e+2	60/15.5/1.75e+2	10(20)/13.7(5.5)/1.75e+2	40/11.9/1.75e+2	**40/10.7/1.75e+2**
	10.00	1070/261.3/9.39e+2	270/68.7/9.39e+2	31(50)/79.6(13.0)/9.39e+2	170/49.5/9.39e+2	**150/38.8/9.39e+2**
6	0.01	**6/3.5/7.57e+0**	9/5.4/7.57e+0	1(9)/8.7(5.2)/7.57e+0	20/12.9/7.57e+0	10/6.0/7.57e+0
	0.10	**10/5.6/5.56e+1**	10/6.1/5.56e+1	1(10)/7.7(6.1)/5.56e+1	20/12.8/5.56e+1	10/6.0/5.56e+1
	1.00	200/104.8/1.63e+2	80/44.7/1.63e+2	10(20)/33.6(11.7)/1.63e+2	50/31.8/1.63e+2	**50/28.1/1.63e+2**
	10.00	470/245.6/7.75e+2	250/137.4/7.75e+2	**13(50)/60.0(28.2)/7.75e+2**	190/120.4/7.75e+2	180/99.7/7.75e+2
7	0.01	**7/3.9/6.91e+0**	10/5.8/6.91e+0	1(10)/8.6(5.8)/6.91e+0	30/17.8/6.91e+0	10/5.7/6.91e+0
	0.10	170/85.2/3.66e+1	**50/26.5/3.66e+1**	10(10)/39.3(5.7)/3.66e+1	160/94.3/3.66e+1	330/171.0/3.66e+1
	1.00	360/177.2/1.32e+2	130/67.6/1.32e+2	12(30)/48.4(16.0)/1.32e+2	90/53.3/1.32e+2	**70/37.2/1.32e+2**
	10.00	980/481.8/6.09e+2	410/212.6/6.09e+2	**12(90)/99.4(47.3)/6.09e+2**	470/280.3/6.09e+2	360/189.0/6.09e+2
8	0.01	**7/9.2/1.29e+1**	10/14.1/1.29e+1	1(10)/17.7(14.1)/1.29e+1	20/30.0/1.29e+1	10/14.1/1.29e+1
	0.10	230/293.8/6.34e+1	**60/78.4/6.34e+1**	13(10)/127.6(14.0)/6.34e+1	210/310.5/6.34e+1	450/597.3/6.34e+1
	1.00	320/405.1/1.77e+2	140/182.2/1.77e+2	14(30)/141.7(40.6)/1.77e+2	100/150.0/1.77e+2	**70/91.8/1.77e+2**
	10.00	850/1067.0/8.53e+2	430/569.8/8.53e+2	**15(80)/224.7(105.1)/8.53e+2**	420/625.0/8.53e+2	340/444.8/8.53e+2

all instances within 2000 iterations) than Dual GP. Furthermore, Dual PPA and Dual ADMM usually work best when μ is large.

4.1.2. Real data from DaISy database. In this section, we consider eight benchmark problems from DaISy (Database for the Identification of Systems) [11]. A brief description of the data is given in Table 4.3. Each data set is given in form of a y (output) and a u (input).

In our first test, we look at the computational efficiency of the first-order methods in solving large scale instances. We take the first 25% of the inputs and outputs for testing purposes. As a rule of thumb, we set $r = 41$ and compute $U^{\perp}$ from the input accordingly. We apply the five algorithms from Table 4.2 to solving these problems for different values of μ, using the same parameters as in the previous section. The results are reported in Table 4.4, where **iter** and **cpu** are as in Table 4.2, while **nr** denotes the number of singular values of $\mathcal{H}(y^*)$ that are larger than $0.005\,\sigma_{\max}(\mathcal{H}(y^*))$, and **err** denotes the fitting error measured by $\|y^* - \tilde{y}\|_F$; here, y^* is the approximate optimal solution of (4.2) obtained by the algorithms. We see that Dual AGP and Dual PPA work best. Furthermore, we see that **nr** decreases with μ, while **err** increases as μ

Table 4.3
Description of test problems, taken from DaISy [11].

P	**Description**	$N+1$	m	q
1	[96-007] CD player arm	513	2	388
2	[98-002] Continuous stirring tank reactor	1876	2	1793
3	[96-006] Hair dryer	251	1	168
4	[97-002] Steam heat exchanger	1001	1	918
5	[96-011] Heat flow density	421	1	296
6	[97-003] Industrial winding process	626	2	375
7	[96-002] Glass furnace	312	6	145
8	[96-016] Industrial dryer	217	3	50

Table 4.4
Computational results for problems from Table 4.3. *The fastest algorithm is highlighted in bold.*

		D. GP	D. AGP	D. PPA	P. ADMM	D. ADMM
P	μ	iter/cpu/nr/err	iter/cpu/nr/err	iter/cpu/nr/err	iter/cpu/nr/err	iter/cpu/nr/err
1	0.01	60/1.1/35/3.2e-1	**30/0.6/35/3.2e-1**	**9(10)/0.6(0.2)/35/3.1e-1**	100/2.2/35/3.2e-1	210/4.1/35/3.2e-1
	0.10	1470/27.6/23/1.6e+0	**180/3.4/23/1.6e+0**	73(30)/4.1(0.6)/23/1.6e+0	290/6.3/23/1.6e+0	610/11.7/23/1.6e+0
	1.00	max/36.9/10/2.8e+0	1200/22.6/5/2.8e+0	45(180)/11.8(3.4)/5/2.8e+0	**360/7.7/5/2.8e+0**	460/8.8/5/2.8e+0
	10.00	max/37.0/26/3.0e+0	max/37.5/16/3.2e+0	42(max)/69.7(37.3)/3/3.2e+0	max/42.6/3/3.2e+0	**1760/33.3/4/3.2e+0**
2	0.01	**10/0.8/6/2.7e-1**	**10/0.9/6/2.7e-1**	1(10)/1.1(0.9)/6/2.7e-1	1920/174.7/6/2.6e-1	max/157.0/6/2.9e-1
	0.10	30/2.3/6/2.0e+0	**10/0.9/6/2.0e+0**	1(10)/1.2(0.9)/6/2.0e+0	660/60.7/6/2.0e+0	1290/102.8/6/2.0e+0
	1.00	50/3.8/3/1.4e+1	**20/1.6/3/1.4e+1**	1(10)/3.0(0.9)/3/1.4e+1	480/43.6/3/1.4e+1	1010/78.7/3/1.4e+1
	10.00	130/9.7/1/6.5e+1	**50/4.0/1/6.5e+1**	6(10)/16.7(0.9)/1/6.5e+1	340/30.8/1/6.5e+1	710/54.3/1/6.5e+1
3	0.01	**20/0.1/19/3.0e-1**	**20/0.1/19/3.0e-1**	1(10)/0.1(0.1)/19/3.0e-1	120/0.7/19/3.0e-1	240/1.2/19/3.0e-1
	0.10	240/1.2/6/9.9e-1	**80/0.4/6/9.9e-1**	32(10)/0.5(0.1)/6/9.8e-1	250/1.4/6/9.9e-1	530/2.7/6/9.9e-1
	1.00	550/2.7/3/2.6e+0	160/0.8/3/2.6e+0	17(40)/0.6(0.2)/3/2.6e+0	**90/0.5/3/2.6e+0**	120/0.6/3/2.6e+0
	10.00	390/1.9/1/1.2e+1	390/2.0/1/1.2e+1	25(130)/2.0(0.6)/1/1.2e+1	470/2.7/1/1.2e+1	**330/1.7/1/1.2e+1**
4	0.01	**7/0.1/2/4.0e-1**	10/0.2/2/4.0e-1	1(10)/0.3(0.2)/2/4.0e-1	50/1.0/2/4.0e-1	110/2.0/2/4.0e-1
	0.10	30/0.5/2/3.3e+0	**10/0.2/2/3.4e+0**	1(10)/0.3(0.2)/2/3.4e+0	170/3.5/2/3.3e+0	340/6.1/2/3.3e+0
	1.00	100/1.7/2/1.1e+1	**20/0.4/2/1.1e+1**	2(10)/1.0(0.2)/2/1.1e+1	120/2.5/2/1.1e+1	240/4.3/2/1.1e+1
	10.00	160/2.7/1/6.0e+1	**80/1.5/1/6.0e+1**	7(10)/4.8(0.2)/1/6.0e+1	**70/1.5/1/6.0e+1**	140/2.5/1/6.0e+1
5	0.01	**9/0.1/42/3.8e-1**	**10/0.1/42/3.8e-1**	1(10)/0.1(0.1)/42/3.8e-1	30/0.3/42/3.8e-1	**10/0.1/42/3.8e-1**
	0.10	410/3.0/10/2.0e+0	**80/0.6/10/2.0e+0**	**22(20)/0.6(0.2)/11/2.0e+0**	170/1.4/10/2.0e+0	340/2.5/10/2.0e+0
	1.00	850/6.1/3/2.8e+0	360/2.6/3/2.8e+0	39(60)/3.6(0.5)/3/2.8e+0	130/1.1/3/2.8e+0	**100/0.8/3/2.8e+0**
	10.00	700/5.0/1/7.1e+0	700/5.1/1/7.1e+0	**29(210)/2.6(1.5)/1/7.1e+0**	780/6.5/1/7.1e+0	640/4.8/1/7.1e+0
6	0.01	**10/0.2/79/4.9e-1**	**10/0.2/79/4.9e-1**	1(10)/0.3(0.2)/79/4.9e-1	30/0.7/79/4.9e-1	20/0.4/79/5.0e-1
	0.10	190/3.6/49/3.3e+0	**60/1.2/49/3.3e+0**	17(10)/1.5(0.2)/49/3.3e+0	70/1.5/49/3.3e+0	150/3.0/49/3.3e+0
	1.00	max/37.5/5/5.8e+0	320/6.2/5/5.8e+0	40(70)/4.1(1.4)/5/5.8e+0	210/4.6/5/5.8e+0	**160/3.1/5/5.8e+0**
	10.00	990/18.6/2/1.4e+1	830/15.8/2/1.4e+1	**26(210)/7.8(4.0)/2/1.4e+1**	1440/31.5/2/1.4e+1	830/16.2/2/1.4e+1
7	0.01	**10/0.2/100/5.9e-1**	**10/0.2/100/5.9e-1**	1(10)/0.3(0.2)/100/5.9e-1	30/0.9/100/5.9e-1	20/0.5/100/5.9e-1
	0.10	340/7.9/37/3.4e+0	**70/1.7/37/3.4e+0**	17(20)/1.8(0.5)/37/3.3e+0	150/4.0/37/3.4e+0	300/7.2/37/3.4e+0
	1.00	1640/36.7/8/8.5e+0	250/5.8/8/8.5e+0	32(40)/5.7(0.9)/8/8.5e+0	170/4.4/8/8.6e+0	**150/3.5/8/8.5e+0**
	10.00	max/44.7/3/2.8e+1	530/12.1/2/2.8e+1	**27(130)/6.1(3.0)/2/2.8e+1**	850/22.4/2/2.8e+1	550/12.8/2/2.8e+1
8	0.01	**10/0.1/38/2.7e-1**	**10/0.1/38/2.7e-1**	1(10)/0.1(0.1)/38/2.7e-1	70/0.4/38/2.7e-1	150/0.8/38/2.7e-1
	0.10	250/1.3/23/1.8e+0	70/0.4/23/1.8e+0	**15(10)/0.3(0.1)/23/1.8e+0**	310/1.9/23/1.8e+0	640/3.5/23/1.8e+0
	1.00	max/10.5/12/6.9e+0	380/2.0/11/6.9e+0	**66(50)/1.5(0.3)/12/6.9e+0**	480/2.9/11/6.9e+0	1010/5.9/12/6.9e+0
	10.00	max/10.4/ 2/1.6e+1	1460/7.7/2/1.6e+1	46(260)/13.9(1.4)/2/1.6e+1	**800/4.8/2/1.6e+1**	1130/6.1/2/1.6e+1

increases.

To further illustrate the performance of the first-order methods on real data, as a second test, we consider the first problem in Table 4.3 and attempt to identify the system order using solutions obtained from Dual AGP. We follow the procedure used in [32, section 5.3], which we summarize in the following five steps:

Step 1. Take two sets of data points. The first set, called the identification set, contains $N_I + 1$ data points from time 0 to N_I, while the second set, called the validation set, contains $N_V + 1$ data points from time 0 to N_V, and $N_V > N_I$. Form $\tilde{y}_t$, $t = 0, \ldots, N_V$, from the data points.

Step 2. Consider a series of μ from 10^{-4} to 10 and fix an r.

Step 3. For each μ:
- Solve the corresponding problem (4.2) by Dual AGP to obtain y_μ. For faster implementation we initialize the Dual AGP at $\Lambda^0 = 0$ for $\mu = 10^{-4}$, and then for each subsequent μ, we initialize at $\Lambda^0 = \Lambda^{\mu_-}$, the optimal solution of (2.5) for the previous μ_-. (Warmstart)
- Estimate the state-space matrices A, B, C, D, and x_0 in (4.1) as done

Table 4.5
Identification errors and validation errors for CD player arm data.

μ	rank$(\mathcal{H}(y_\mu))$	err_{id}	err_{v}	$\|\mathcal{H}(y_\mu)\|_*$
1.233	6	0.141	0.230	0.185
1.385	6	0.143	0.247	0.127
1.556	3	0.178	0.187	0.104
1.748	3	0.178	0.187	0.097
1.963	3	0.177	0.186	0.090

in [32, section 5.2]. This is the standard subspace identification method; see, for example, [34, section 10.6]. For the sake of completeness, we also describe the details in Appendix C.

- Compute identification errors (err_{id}) and validation errors (err_{v}) as in [32, eq. (5.7)], using respectively the $N_I + 1$ and $N_V + 1$ data points:

$$err_{\mathrm{id}} = \sqrt{\frac{\sum_{t=0}^{N_I} \|\tilde{y}_t - \widehat{y}_t\|^2}{\sum_{t=0}^{N_I} \|\tilde{y}_t - \bar{y}_I\|^2}}, \qquad err_{\mathrm{v}} = \sqrt{\frac{\sum_{t=0}^{N_V} \|\tilde{y}_t - \widehat{y}_t\|^2}{\sum_{t=0}^{N_V} \|\tilde{y}_t - \bar{y}_V\|^2}},$$

where

$$\bar{y}_I = \frac{1}{N_I + 1} \sum_{t=0}^{N_I} \tilde{y}_t, \qquad \bar{y}_V = \frac{1}{N_V + 1} \sum_{t=0}^{N_V} \tilde{y}_t,$$

and $\widehat{y}_t$ is the output generated from (4.1) using the estimated A, B, C, D, and x_0.

Step 4. Plot identification and validation errors against the nuclear norm of $\mathcal{H}(y_\mu)$. Identify the μ^* where the "best" trade-off between the errors occurs.

Step 5. The estimated system order is set to be the numerical rank of $\mathcal{H}(y_{\mu^*})$, where y_{μ^*} is the solution of (4.2) with $\mu = \mu^*$. The system parameters are then given by the corresponding A, B, C, D, and x_0.

For comparison with the work [32], we use the same number of data points for identification and validation as in [32, Table 5.1], i.e., 200 points for identification and 600 points for validation for the CD player arm problem. Furthermore, we mimic their choice and set

$$r = \left\lfloor \frac{N_I + 2}{p + m + 1} \right\rfloor. \tag{4.4}$$

Table 4.5 shows some identification errors (err_{id}) and validation errors (err_{v}), as well as the corresponding rank[5] and nuclear norm of $\mathcal{H}(y_\mu)$, while Figure 4.1 plots identification and validation errors against the nuclear norm of $\mathcal{H}(y_\mu)$. From these, we conclude that $\mu^* = 1.556$ and the estimated system order in this case is 3, which agrees with the result obtained in [32, Table 5.2]. We also show in Figure 4.2 a plot of the top 20 singular values of $\mathcal{H}(y_{\mu^*})$ (normalized by dividing by $\sigma_{\max}(\mathcal{H}(y_{\mu^*}))$) and that of $\mathcal{H}(\tilde{y})$ (normalized by dividing by $\sigma_{\max}(\mathcal{H}(\tilde{y}))$). We see that it is relatively easier to identify a numerical rank of 3 for $\mathcal{H}(y_{\mu^*})$ than for $\mathcal{H}(\tilde{y})$.

Another commonly used method for identifying system order is the subspace algorithm, which is implemented in the MATLAB function `n4sid`. A detailed comparison

[5]We determine the rank by looking at the plot of the top 20 singular values of $\mathcal{H}(y_\mu)$.

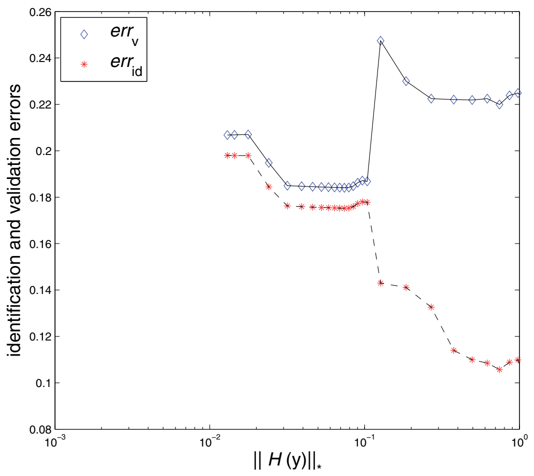

FIG. 4.1. *Plot of err_{id} and err_{v} against $\|\mathcal{H}(y)\|_*$.*

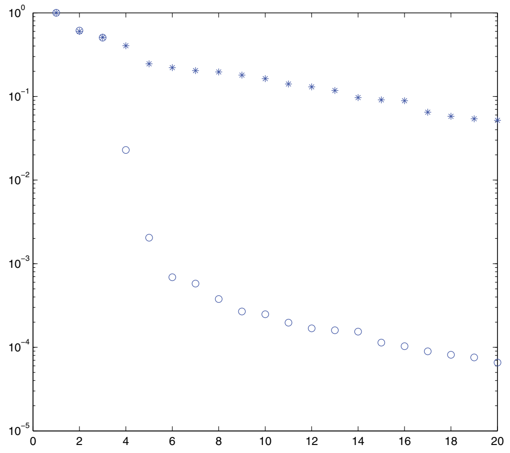

FIG. 4.2. *Plot of top* 20 *singular values of $\mathcal{H}(\tilde{y})$ and $\mathcal{H}(y_{\mu^*})$, normalized by the respective spectral norm of the matrices, where $*$ denotes the weighted singular values for $\mathcal{H}(\tilde{y})$ and $\circ$ denotes the weighted singular values for $\mathcal{H}(y_{\mu^*})$.*

between our approach described above and the MATLAB function `n4sid` has been presented in [32, Table 5.2]. Since our first-order method usually obtains a solution to (4.2) with lower accuracy than the interior point method used in [32], and the MATLAB function `n4sid` has been updated since the publication of [32], it is worth repeating the comparison for the problems in Table 4.3. In the test below, we use the same number of data points for identification and validation as in [32, Table 5.1] and use (4.4). We report the system orders (ord), identification errors (err_{id}), and validation errors (err_{v}) obtained via our approach outlined above (nn), the `n4sid` with default order (n4sid$_d$) called using the option `'best'`, and the `n4sid` using the

TABLE 4.6
Comparing our approach and n4sid on problems from Table 4.3.

	nn	`n4sid`$_d$	`n4sid`$_n$
P	ord/$err_{\rm id}$/$err_{\rm v}$/μ^*	ord/$err_{\rm id}$/$err_{\rm v}$	ord/$err_{\rm id}$/$err_{\rm v}$
1	3/0.18/0.19/1.556	2/0.21/0.23	3/0.16/0.18
2	3/0.19/0.20/1.556	3/0.21/0.58	3/2.20/5.77
3	4/0.069/0.12/0.215	2/0.13/0.18	4/0.080/0.13
4	5/0.11/0.23/0.343	2/3.36/5.38	5/5.47/7.36
5	8/0.14/0.13/1.385	1/0.18/0.17	8/0.14/0.13
6	3/0.17/0.17/0.546	2/0.52/0.58	3/0.18/0.17
7	4/0.22/0.40/1.556	2/0.41/0.53	4/0.60/0.64
8	6/0.28/0.29/2.205	1/0.47/0.29	6/0.31/0.27

order obtained from our approach (n4sid$_n$) in Table 4.6. We also report the μ^* used in our approach. We observe that the orders and errors obtained from our approach are usually slightly different from those in [32, Table 5.2]. This can be attributed to the solution accuracy in solving (4.2) and the threshold for determining the rank of $\mathcal{H}(y_\mu)$.[6] On the other hand, we still observe that our approach usually yields lower identification and validation errors than do the two versions of `n4sid`.[7]

Comparison with interior point solver from [32]. Finally, we apply the interior point method in [32] to solving the first problem in Table 4.3 (i.e., with $N = 512$). We use the code (written in Python and calling `cvxopt` 1.1.3, which is installed from source using Python 2.6.5) available on the authors' Web page, compiled with Python 2.6.5. While the interior point method provides a solution of higher accuracy (the relative duality gap is usually around 10^{-6}, rather than the 10^{-4} as required in our first-order methods), it is much slower. We observe that the number of iterations and CPU time for the interior point method are not sensitive to the change of μ and are usually around 9 iterations and 1000 seconds, respectively. In particular, when $\mu \le 1$, this method is significantly slower than Dual AGP, which takes between 1 and 40 seconds for this range of μ. The relative performance can also be explained via the cost per iteration. For the interior point method from [32], since $\mathcal{H}(y) \in \mathbb{R}^{m(r+1)\times q}$ and $y \in \mathbb{R}^{m\times(N+1)}$, the cost per iteration is $\mathcal{O}(m^3rN^2q)$, with $m(N+1) \ge q \ge m(r+1)$, and the method requires around 10 iterations.[8] On the other hand, for the first-order methods, although the method takes around 1000 iterations, each iteration only

[6]We determine the rank by looking at the plot of the top 20 singular values of $\mathcal{H}(y_\mu)$, except for problems 4 and 7, where the cut-off is not clear and we truncate at $10^{-4}\cdot\sigma_{\max}(\mathcal{H}(y_\mu))$ and $10^{-2}\cdot\sigma_{\max}(\mathcal{H}(y_\mu))$, respectively.

[7]We note two observations:

(a) The errors for problems 2 and 4 obtained from `n4sid` are exceptionally large. In view of this, we also tried to vary the input order for `n4sid` from 1 to 12 for these two problems and observed the corresponding identification and validation errors. For problem 2, the smallest $(err_{\rm id}, err_{\rm v}) = (0.19, 0.39)$, which occurs when the order is 10, while for problem 4, the smallest $(err_{\rm id}, err_{\rm v}) = (0.14, 0.22)$, which occurs when the order is 8.

(b) The commands `n4sid`$_n$ and `n4sid`$_d$ give different errors for problem 2 even though the order is the same. It seems that a different set of A, B, C, D is obtained if one directly inputs to `n4sid` the order obtained from calling `n4sid` with the option `'best'` (this was observed for all eight problems). To be precise, here is how we call `n4sid`$_d$: `n4sid`(data,'best','nk',zeros(1,m),'InitialState','Estimate','CovarianceModel','None','Focus', 'Stability', 'DisturbanceModel','None').

[8]The cost per iteration—more precisely, the cost for computing the Newton direction—was derived under the assumption that $y \mapsto \mathcal{H}(y)$ is injective for the simpler problem, $\min_y \|\mathcal{H}(y) - B\|_*$ for a given matrix B. The analysis can be adapted for (2.1) if $y \mapsto \mathcal{H}(y)$ is injective, and gives the same iteration complexity.

involves an SVD of cost $\mathcal{O}(m^2r^2q)$ [25, Page 254] and several matrix multiplications for forming the products $H_{m,1,r+1,N+1-r}(y)U^\perp$ and $\Lambda(U^\perp)^T$, each of cost $\mathcal{O}(mNrq)$. Thus, the first order methods appear more suitable for larger scale identification problems.

5. Stochastic system realization. Another fundamental problem in linear system theory is finding a minimal stochastic ARMA (autoregressive moving average) model for a vector random process, given noisy and/or partial estimates of process covariances. Covariance estimates are often obtained from sample averages of a sequence of noisy observations of the process, hence including both measurement noise and error due to the finite sample size. Here we focus on a form of this problem, described in [33, section II.B]; see also [35]. Consider a state-space model of an ARMA process $y_t \in \mathbb{R}^n$,

$$\begin{aligned} x_{t+1} &= Ax_t + Be_t, \\ y_t &= Cx_t + e_t, \end{aligned}$$

where $x_t \in \mathbb{R}^r$ and e_t is white noise with covariance matrix Q. The process covariances $h_i = \mathbf{E}(y_t {y_{t+i}}^T) \in \mathbb{R}^{n\times n}$ satisfy

$$h_0 = CPC^T + Q, \quad h_t = CA^{t-1}D,\ t \geq 1,$$

where $D = APC^T + BQ$ and $P = \mathbf{E}(x_t x_t^T)$ satisfies the Lyapunov equation $P = APA^T + BQB^T$. In a stochastic realization problem, we are given noisy estimates of h_i, denoted by $\tilde{h}_i$, $i = 1, \ldots, T-1$, and the goal is to find the minimal model order r as well as the model parameters A, B, C, Q. It is known that the minimal order is equal to the rank of the block-Hankel matrix $H_{n,n,j,k}$ consisting of the exact process covariances [35]. (As in the system ID problem, we need the Hankel matrix to be large enough; that is, j and k should be larger than the rank.)

A general form of this problem, allowing for both noise and missing covariance information, can be stated as follows:

$$\min_y \frac{1}{2}\|w \circ (y - \tilde{h})\|_F^2 + \mu\|H_{m,n,j,k}(y)\|_*, \tag{5.1}$$

where $\circ$ denotes the Hadamard (or entrywise) product and $w = \begin{pmatrix} w_0 & \cdots & w_{j+k-2}\end{pmatrix}$ is an $m \times n(j+k-1)$ matrix, with each w_i being an $m\times n$ zero matrix or a matrix of all ones (denoting the case where some covariance blocks are unknown or missing), or a matrix with $0, 1$ entries (denoting the case where some covariance entries are missing). The problem corresponds to (2.1) with $\Upsilon = I$, $\mathcal{A}(y) = \mathrm{vec}(w\circ y)$, and $b = \mathrm{vec}(w \circ \tilde{h})$. Thus,

$$\sigma_{\max}(\mathcal{A}) = \sigma_{\max}(\Upsilon) = 1, \quad \mathcal{H}^*(\Lambda) = H^*_{m,n,j,k}(\Lambda).$$

Notice that the solution set of (5.1) is nonempty since the function $y \mapsto \|\mathcal{H}(y)\|_*$ is coercive.

5.1. Computational results. In this section, we compare different algorithms for solving (5.1). Specifically, we consider Primal ADMM, Dual ADMM, and Dual PPA. Since the action of $(\beta\mathcal{H}^*\mathcal{H} + \mathcal{A}^*\mathcal{A})^{-1}$ on vectors can be easily computed for any $\beta > 0$, we also consider a variant of Primal ADMM (referred to as Primal ADMM2) with $\mathcal{Q}_0 = 0$ instead of (3.1). Furthermore, notice that the quadratic term in (5.1) is *not* strictly convex in y; the dual problem will then have additional linear constraints that make gradient projection expensive computationally. Thus, we do not consider gradient projection algorithms for solving (5.1).

We initialize all algorithms except Dual PPA at the origin, where the latter algorithm is initialized at an approximate solution obtained from Primal ADMM2,[9] and terminate the algorithms by checking

(5.2)
$$\max\left\{\frac{\min_{t\in\mathcal{K}^k} f(y^t)+d(-w\circ\mathcal{H}^*(\mathcal{P}(\Lambda^k)))}{\max\{1,|d(-w\circ\mathcal{H}^*(\mathcal{P}(\Lambda^k)))|\}},\frac{\|\mathcal{H}^*(\mathcal{P}(\Lambda^k))-w\circ\mathcal{H}^*(\mathcal{P}(\Lambda^k))\|_F}{\max\{1,\|\mathcal{H}^*(\mathcal{P}(\Lambda^k))\|_F\}}\right\} < 10^{-4},$$

with $\{(y^k,\Lambda^k)\}$ defined as in sections 3.1 and 3.2; $\mathcal{P}$ is the projection onto the spectral norm ball with radius μ. We set $\mathcal{K}^k=\{1\le\nu\le k:\ \nu \text{ divides } 10\}$ for all algorithms except the Dual PPA and check (5.2) every 10 iterations. For Dual PPA we set $\mathcal{K}^k=\{1\le\nu\le k\}$ and check (5.2) every iteration. For the ADMMs we still set $\tau=1.61$, but we set $\beta=\frac{\mu\mathbf{r}}{2\sigma_{\max}(h)}$ for Primal ADMM and Primal ADMM2 and $\beta=\frac{\sigma_{\max}(h)}{8\mu\mathbf{r}}$ for Dual ADMM. For Dual PPA, we use the same parameters as in the previous section.

Following [33, section II(B)], we generate matrices $A\in\mathbb{R}^{r\times r}$, $B\in\mathbb{R}^{r\times n}$, and $C\in\mathbb{R}^{n\times r}$ with i.i.d. standard Gaussian entries (i.e., with mean 0 and variance 1). These matrices are then normalized to have spectral norm 1. We also randomly generate an initial state $x_0\sim N(0,I)$ and noise vectors $e_t\sim N(0,I)$ for $t=0,\dots,T-1$, again with i.i.d. standard Gaussian entries. We then generate an output $\bar{y}_t$, $t=0,\dots,T-1$, according to the state-space model:

$$\begin{aligned} x_{t+1}&=Ax_t+Be_t,\\ \bar{y}_t&=Cx_t+e_t. \end{aligned}$$

To model measurement noise, we further add noise to $\bar{y}$ and get $\tilde{y}=\bar{y}+\sigma\epsilon$, where ϵ has i.i.d. Gaussian entries with mean 0 and variance 1. We then set, for each $i=0,\dots,k-1$,

$$\tilde{h}_i=\frac{1}{T}\sum_{t=0}^{T-1-i}\tilde{y}_{t+i}\tilde{y}_t^T,$$

and $\tilde{h}_i$ is zero for $i\ge k$. Finally, set $w=\begin{pmatrix} w_0 & \cdots & w_{j+k-2}\end{pmatrix}$ such that $w_0=\cdots=w_{k-1}$ equals the matrix of all ones, and zero otherwise. In the tests below, we consider $T=1000$, $n=20,30$, and $k=100,200$. We use $r=10$ and hence set $j=21$. We further pick $\sigma=5\times10^{-2}$. The statistics of the test problems used are reported in the caption for Table 5.1; recall that $q=nk$ in this case. We run our algorithms for a range of values of μ, namely $\mu=10^{-2},10^{-1},1,10$, in our simulations below to study the performance of the algorithms for different values of μ. The computational results are reported in Table 5.1. We see that Primal ADMM2 works best in all instances.[10]

[9] We terminate Primal ADMM2 by checking relative duality gap and relative dual feasibility with $tol=5\times10^{-3}$, checked every 10 iterations. We also terminate the algorithm early if the change in Fröbenius norm of successive iterates is small ($<10^{-8}$ for each variable) or the maximum number of iteration hits 2000.

[10] We note that since the action of $(\mathcal{I}+\beta\mathcal{A}\mathcal{A}^*)^{-1}$ on vectors is easy to compute, one could have chosen $\mathcal{Q}_1=0$ rather than $\mathcal{I}-\mathcal{A}\mathcal{A}^*$ in Dual ADMM. However, since $\gamma^0=0$, it is not hard to check that the variant with $\mathcal{Q}_1=0$ would generate exactly the same sequence as Dual ADMM, which uses (3.2).

TABLE 5.1
Computational results for the stochastic system realization problem. For Problems 1 and 2, $k = 100$, with $n = 20$ and 30, respectively; for problems 3 and 4, $k = 200$, with $n = 20$ and 30, respectively.

		P. ADMM	P. ADMM2	D. ADMM	D. PPA
P	μ	iter/cpu/obj	iter/cpu/obj	iter/cpu/obj	iter/cpu/obj
1	0.01	360/207.3/5.38e+0	**30/17.3/5.38e+0**	70/35.4/5.38e+0	55(20)/96.4(11.4)/5.38e+0
	0.10	**70/39.4/2.73e+1**	**70/39.9/2.73e+1**	140/69.3/2.73e+1	51(20)/71.8(11.2)/2.73e+1
	1.00	200/113.5/4.18e+1	**20/11.3/4.18e+1**	110/54.5/4.18e+1	14(10)/28.0(5.5)/4.18e+1
	10.00	180/103.3/4.38e+1	**20/11.2/4.38e+1**	180/90.4/4.38e+1	45(10)/86.3(5.5)/4.38e+1
2	0.01	490/788.5/9.12e+0	**30/47.4/9.12e+0**	100/142.4/9.12e+0	112(20)/451.5(31.8)/9.12e+0
	0.10	80/126.8/4.99e+1	**40/64.6/4.99e+1**	80/114.3/4.99e+1	43(10)/726.2(15.2)/4.99e+1
	1.00	180/287.2/7.04e+1	**20/31.2/7.04e+1**	100/141.7/7.04e+1	16(10)/165.1(15.3)/7.04e+1
	10.00	180/287.3/7.16e+1	**20/31.3/7.16e+1**	180/255.1/7.16e+1	39(10)/222.1(15.3)/7.16e+1
3	0.01	410/499.7/7.16e+0	**30/35.7/7.16e+0**	90/95.3/7.16e+0	61(20)/215.6(23.7)/7.16e+0
	0.10	70/84.2/4.02e+1	**40/47.6/4.02e+1**	50/53.2/4.02e+1	42(10)/273.4(11.7)/4.02e+1
	1.00	160/194.9/5.29e+1	**20/23.8/5.29e+1**	80/85.5/5.29e+1	7(10)/35.6(11.6)/5.29e+1
	10.00	180/216.6/5.33e+1	**20/23.7/5.33e+1**	180/191.6/5.33e+1	35(10)/195.6(11.5)/5.33e+1
4	0.01	550/1932.1/1.35e+1	**30/106.9/1.35e+1**	90/281.4/1.35e+1	51(20)/2307.9(69.0)/1.35e+1
	0.10	110/384.1/8.74e+1	**40/138.8/8.74e+1**	50/157.1/8.74e+1	26(10)/953.2(32.8)/8.74e+1
	1.00	130/457.2/1.23e+2	**20/71.6/1.23e+2**	80/250.5/1.23e+2	8(10)/92.8(32.8)/1.23e+2
	10.00	180/629.7/1.25e+2	**20/67.5/1.25e+2**	180/550.1/1.25e+2	39(10)/470.9(33.1)/1.25e+2

Comparison with other methods for system realization.. We also adapted Cadzow's method [4], an approach based on alternating projections, that has been used in engineering applications for similar problems. We start by specifying an estimate $\tilde{r}$ of the order and $\tilde{\epsilon} > 0$ of the noise level. In each iteration, we first project onto the set of matrices with a Hankel structure to obtain H^k, then project onto the ball centered at $\tilde{h}$ with radius $\tilde{\epsilon}$ to obtain Y^k, and finally project onto the (nonconvex) set of matrices with rank less than $\tilde{r}$ to obtain R^k. The algorithm is terminated when

$$\frac{\max\{\|H^k - Y^k\|_F, \|R^k - Y^k\|_F\}}{\max\{1, \|H^k\|_F\}} < 10^{-3}.$$

In the examples we tested, the performance and convergence of this algorithm is very sensitive to the values of the two parameters, the noise level and the order estimate. Indeed, it is known to be suboptimal as discussed in [10, section V.A]. On the other hand, the convex optimization approach presented here depends only on one parameter (μ), and the runtime does not change significantly as μ varies.

Finally, we mention that another common approach to solving the system realization problem is via subspace methods [35]. For a detailed comparison between the nuclear norm approach and the subspace method applied to this problem, we refer readers to [33, section 2B], where problem (5.1) was solved via an interior-point solver.

6. Concluding remarks.

We studied the optimization problem of minimizing the nuclear norm of matrices with linear structure, including Hankel, Toeplitz, and moment structures. We showed that solving this problem leads to a systematic approach to capturing the trade-off between a model's order and complexity and fitting (and validation) errors, a trade-off that arises in many modeling applications. We then focused on first-order methods for solving the resulting optimization problem. In our computational experiments, for the system identification problem, the gradient projection method (accelerated by Nesterov's extrapolation techniques) and the dual proximal point algorithm usually outperform other first-order methods in terms of CPU time for large and small regularization parameters, respectively. For the system realization problem, the alternating direction method of multipliers, as applied to a certain primal reformulation, usually outperforms other first-order methods in terms of CPU time. In our tests, we also observe that these methods outperform the interior-point implementation proposed in [32] for system identification problems.

We remark that most of the algorithms proposed in this work can easily be adapted to solving (1.1) with a *general* linear structure $X = \mathcal{L}(y)$, as long as $\mathcal{L}(y)$ can be computed efficiently and the norm of $\mathcal{L}$ can be estimated. We chose to focus our discussion on the Hankel structure to relate to our motivating applications discussed in the introduction, which mainly concern Hankel structure.

An interesting direction for future work on this problem is whether there are conditions under which the nuclear norm heuristic can be theoretically guaranteed to find the minimum-rank solution. In order to make analogies with the existing low-rank matrix recovery framework, we can consider the following problem: how many generic, random linear measurements of a rank-r, $n \times n$ Hankel matrix suffice for correct recovery of the Hankel matrix? If we ignore the Hankel structure, existing results on recovery from random Gaussian measurements require $\mathcal{O}(nr)$ measurements [6]; however, it is expected that the number would be much lower due to the Hankel structure.

Another future research direction involves applying our algorithms to the broader range of applications identified in the introduction, particularly moment-based problems. Some of these problems have further structure that the algorithms can exploit.

Appendix A. An alternative formulation. In this appendix, we describe an alternative formulation for modeling the structured matrix rank minimization. Instead of minimizing a least-squares fitting error, we constrain the difference $\mathcal{A}(y) - b$ to be in a set. The problem is then formulated as

$$\begin{array}{ll} \min\limits_{y} & \|\mathcal{H}(y)\|_* \\ \text{s.t.} & \mathcal{A}(y) - b \in \Omega, \end{array} \tag{A.1}$$

where Ω is the closed convex set modeling uncertainty. For instance, problem (1.2) can be modeled as (A.1) with $\Omega = [l_1, b_1] \times [l_2, b_2] \times \cdots \times [l_n, b_n]$.

As in section 2, we can rewrite (A.1) as

$$\begin{array}{rl} v_1 := \min\limits_{Y,z,y} & \|Y\|_* \\ \text{s.t.} & Y + \mathcal{H}(y) = 0, \\ & z + \mathcal{A}(y) = b, \\ & z \in \Omega. \end{array} \tag{A.2}$$

It is then not hard to show that the dual to (A.2) is equivalent to solving

$$\begin{array}{rl} -v_1 = \min\limits_{\gamma,\Lambda} & s_\Omega(\gamma) - b^T\gamma \\ \text{s.t.} & \mathcal{H}^*(\Lambda) + \mathcal{A}^*(\gamma) = 0, \\ & \Lambda^T\Lambda \preceq I, \end{array} \tag{A.3}$$

where $s_\Omega(\gamma) := \sup_{y\in\Omega} y^T\gamma$ is the support function of the set Ω.

Unlike (2.3), the objective function of (A.3) is in general not differentiable; hence gradient projection methods are not easily applicable. However, the ADMMs and the dual PPA can be suitably applied to solving (A.2) and (A.3). The efficiency in solving the subproblems depends on the specific form of Ω.

Appendix B. Convergence of the proximal ADMM. In this appendix, we provide a convergence proof of a proximal alternating direction method of multipliers, which covers all versions of proximal ADMMs used in this paper. Consider the convex optimization problem with the following separable structure:

$$\begin{array}{ll} \min\limits_{x,y} & f(x)+g(y) \\ \text{s.t.} & Ax+By=c, \end{array} \tag{B.1}$$

where $f:\mathcal{X}\to(-\infty,+\infty]$ and $g:\mathcal{Y}\to(-\infty,+\infty]$ are closed proper convex functions, $A:\mathcal{X}\to\mathcal{Z}$ and $B:\mathcal{Y}\to\mathcal{Z}$ are linear operators, and $\mathcal{X},\mathcal{Y}$, and $\mathcal{Z}$ are real finite dimensional Euclidean spaces with inner product $\langle\cdot,\cdot\rangle$ and its induced norm $\|\cdot\|$. The proximal ADMM for solving (B.1) takes the following form:

Proximal ADMM
Step 0. Input $(x^0,y^0,z^0)\in\operatorname{dom} f\times\operatorname{dom} g\times\mathcal{Z}$.
Step 1. Set

$$\left\{\begin{array}{l} x^{k+1}=\arg\min\limits_{x\in\mathcal{X}} f(x)-\langle z^k,Ax\rangle \\ \qquad\quad +\dfrac{\lambda}{2}\|Ax+By^k-c\|^2+\dfrac{1}{2}\|x-x^k\|_S^2, \\ y^{k+1}=\arg\min\limits_{y\in\mathcal{Y}} g(y)-\langle z^k,By\rangle \\ \qquad\quad +\dfrac{\lambda}{2}\|Ax^{k+1}+By-c\|^2+\dfrac{1}{2}\|y-y^k\|_T^2, \\ z^{k+1}=z^k-\tau\,\lambda(Ax^{k+1}+By^{k+1}-c), \end{array}\right. \tag{B.2}$$

where $\lambda>0$ is the penalty parameter, $\tau\in(0,(1+\sqrt{5})/2)$ is the step length (one can take $\tau\in(0,2)$ when the x-part or the y-part disappears), and S and T are two self-adjoint positive semidefinite, not necessarily positive definite, operators on $\mathcal{X}$ and $\mathcal{Y}$, respectively.
Step 2. If a termination criterion is not met, go to Step 1.

When $S=0$ and $T=0$, the proximal ADMM (B.2) reduces to the classical ADMM introduced by Glowinski and Marroco [23] and Gabay and Mericier [22]. It was shown by Eckstein and Bertsekas [15] that the ADMM with $\tau=1$, as a special case of the Douglas–Rachford splitting [21], is actually an application of the proximal point algorithm on the dual problem by means of a specially constructed splitting operator. Based on the same argument, by further applying a change of variables to the operators, Eckstein [14] presented the first proximal ADMM as in (B.2) with $S=\alpha\mathcal{I}$ and $T=\beta\mathcal{I}$ for positive constants $\alpha>0$ and $\beta>0$. Later, He et al. [26] further extended the idea of Eckstein [14] to monotone variational inequalities to allow λ, S, and T to be replaced by different parameters λ_k, S_k, and T_k in each iteration. The convergence results provided in [14] and [26] for the proximal ADMM both need S and T to be positive definite, which makes the convergence analysis easily accessible but may limit the applications of the method. Very recently, Xu and Wu [57] provided a nice extension on convergence analysis from the classical ADMM to the proximal ADMM (B.2), allowing the step length τ to stay in the larger range $(0,(1+\sqrt{5})/2)$, which is desirable in many applications in the following scenarios: (i) $T=0$ and $S+\lambda A^*A=\alpha\mathcal{I}$; (ii) $S=0$ and $T+\lambda B^*B=\beta\mathcal{I}$; and (iii) $S+\lambda A^*A=\alpha\mathcal{I}$ and $T+\lambda B^*B=\beta\mathcal{I}$, where $\alpha>\lambda\sigma_{\max}(A^*A)$ and $\beta>\lambda\sigma_{\max}(B^*B)$ are two positive constants. On the one hand, to require S or T to be positive definite instead of being positive semidefinite will not make much difference in numerical computations. On the other hand, the restriction on the positive definiteness will exclude some interesting situations such as the classical ADMM. Here we will provide a unified convergence theorem by requiring S and T to be positive semidefinite only. This will allow us to apply the proximal ADMM to more interesting problems (for such a case, see the second remark after the proof of our theorem) as well as provide a guide on choices of S and T (both S and T should be as small as possible). Actually, by exploiting the potential composite structures of f and g, we may not even need $S + \lambda A^*A$ or $T + \lambda B^*B$ to be positive definite, a commonly used assumption for existing ADMM and its variants. Our proof is quite routine in some sense, as the variational tools employed are quite standard, and is closely based on the essential ideas developed by Fortin and Glowinski [20] for the convergence of the ADMM with $B = \mathcal{I}$. For clarity and ease of reference, we include a proof here.

For technical reasons, we consider the following constraint qualification:

CQ: There exists $(x_0, y_0) \in \mathrm{ri}(\mathrm{dom}\, f \times \mathrm{dom}\, g) \cap P$, where P is the constraint set in (B.1).

Under the **CQ**, it follows from [48, Corollaries 28.2.2 and 28.3.1] that $(\bar{x}, \bar{y}) \in \mathcal{X} \times \mathcal{Y}$ is an optimal solution to problem (B.1) if and only if there exists a Lagrange multiplier $\bar{z} \in \mathcal{Z}$ such that

$$A^*\bar{z} \in \partial f(\bar{x}), \quad B^*\bar{z} \in \partial g(\bar{y}), \quad A\bar{x} + B\bar{y} - c = 0, \tag{B.3}$$

where ∂f and ∂g are the subdifferential mappings of f and g, respectively. Moreover, any $\bar{z} \in \mathcal{Z}$ satisfying (B.3) is an optimal solution to the dual problem of (B.1). On the other hand, since the subdifferential mappings of the closed proper convex functions are maximal monotone [49, Theorem 12.17], there exist two self-adjoint and positive semidefinite operators Σ_f and Σ_g such that for all $x, \hat{x} \in \mathrm{dom}(f)$, $u \in \partial f(x)$, and $\hat{u} \in \partial f(\hat{x})$,

$$f(x) \geq f(\hat{x}) + \langle \hat{u}, x - \hat{x} \rangle + \frac{1}{2}\|x - \hat{x}\|^2_{\Sigma_f} \text{ and } \langle u - \hat{u}, x - \hat{x} \rangle \geq \|x - \hat{x}\|^2_{\Sigma_f}, \tag{B.4}$$

and for all $y, \hat{y} \in \mathrm{dom}(g)$, $v \in \partial g(y)$, and $\hat{v} \in \partial g(\hat{y})$,

$$g(y) \geq g(\hat{y}) + \langle \hat{v}, y - \hat{y} \rangle + \frac{1}{2}\|y - \hat{y}\|^2_{\Sigma_g} \text{ and } \langle v - \hat{v}, y - \hat{y} \rangle \geq \|y - \hat{y}\|^2_{\Sigma_g}. \tag{B.5}$$

For notational convenience, define for $(x, y) \in \mathcal{X} \times \mathcal{Y}$ that $h(x, y) := Ax + By$ and for $(x, y, z) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$ that

$$\theta(x, y, z) := (\tau\lambda)^{-1}\|z - \bar{z}\|^2 + \|x - \bar{x}\|^2_S + \|y - \bar{y}\|^2_T + \lambda\|B(y - \bar{y})\|^2. \tag{B.6}$$

Theorem B.1. *Assume that the solution set of* (B.1) *is nonempty and that the* **CQ** *holds. Assume also that both* $\Sigma_f + S + \lambda A^*A$ *and* $\Sigma_g + T + \lambda B^*B$ *are positive definite. Let* $\{(x^k, y^k, z^k)\}$ *be generated from the proximal ADMM. For* $k = 1, 2, \ldots,$ *denote*

$$\begin{cases} \delta_{k+1} := \min(\tau, 1 + \tau - \tau^2)\lambda||B(y^{k+1} - y^k)\|^2 + \|y^{k+1} - y^k\|^2_T, \\ t_{k+1} := \delta_{k+1} + \|x^{k+1} - x^k\|^2_S + 2\|x^{k+1} - \bar{x}\|^2_{\Sigma_f} + 2\|y^{k+1} - \bar{y}\|^2_{\Sigma_g}, \\ \psi_{k+1} := \theta(x^{k+1}, y^{k+1}, z^{k+1}) + ||y^{k+1} - y^k\|^2_T. \end{cases} \tag{B.7}$$

Then, the following results hold:

(a) *if $\tau \in (0,1]$, we have for $k \geq 1$ that*

$$\psi_{k+1} + (1-\tau)\lambda\|h(x^{k+1}, y^{k+1}) - c\|^2 - [\psi_k + (1-\tau)\lambda\|h(x^k, y^k) - c\|^2] \tag{B.8}$$
$$\leq -[t_{k+1} + \lambda\|h(x^{k+1}, y^{k+1}) - c\|^2];$$

(b) *if $\tau > 1$, we have for $k \geq 1$ that*

$$\psi_{k+1} + (1-\tau^{-1})\lambda\|h(x^{k+1}, y^{k+1}) - c\|^2 - [\psi_k + (1-\tau^{-1})\lambda\|h(x^k, y^k) - c\|^2] \tag{B.9}$$
$$\leq -[t_{k+1} + \tau^{-1}(1+\tau-\tau^2)\lambda\|h(x^{k+1}, y^{k+1}) - c\|^2];$$

(c) *if $\tau \in (0, (1+\sqrt{5})/2)$, then the sequence $\{(x^k, y^k)\}$ converges to an optimal solution to* (B.1) *and $\{z^k\}$ converges to an optimal solution to the dual problem of* (B.1)*;*

(d) *and if A is vacuous and $f \equiv 0$, then for $\tau \in (0,2)$ we have that $x^{k+1} = x^0 = \bar{x}$, and for $k \geq 0$ it holds that*

$$(\tau\lambda)^{-1}\|z^{k+1} - \bar{z}\|^2 + \|y^{k+1} - \bar{y}\|_T^2 \leq (\tau\lambda)^{-1}\|z^k - \bar{z}\|^2 + \|y^k - \bar{y}\|_T^2 \tag{B.10}$$
$$-[(2-\tau)\lambda\|B(y^{k+1}) - c\|^2 + \|y^{k+1} - y^k\|_T^2 + 2\|y^{k+1} - \bar{y}\|_{\Sigma_g}^2],$$

and the sequence $\{(x^k, y^k)\}$ converges to an optimal solution to (B.1)*, and $\{z^k\}$ converges to an optimal solution to the dual problem of* (B.1)*.*

Proof. Obviously, the sequence $\{(x^k, y^k, z^k)\}$ is well defined under the assumptions given in this theorem. Notice that the iteration scheme (B.2) of the proximal ADMM can be rewritten as: for $k = 0, 1, 2, \ldots,$

$$\begin{cases} 0 \in \partial f(x^{k+1}) - A^*[z^k - \lambda(h(x^{k+1}, y^k) - c)] + S(x^{k+1} - x^k), \\ 0 \in \partial g(y^{k+1}) - B^*[z^k - \lambda(h(x^{k+1}, y^{k+1}) - c)] + T(y^{k+1} - y^k), \\ 0 = h(x^{k+1}, y^{k+1}) - c + (\tau\lambda)^{-1}(z^{k+1} - z^k), \end{cases}$$

which implies

$$\begin{cases} A^*[z^{k+1} - (1-\tau)\lambda(h(x^{k+1}, y^{k+1}) - c) - \lambda B(y^k - y^{k+1})] \\ \qquad - S(x^{k+1} - x^k) \in \partial f(x^{k+1}), \\ B^*[z^{k+1} - (1-\tau)\lambda(h(x^{k+1}, y^{k+1}) - c)] \\ \qquad - T(y^{k+1} - y^k) \in \partial g(y^{k+1}). \end{cases} \tag{B.11}$$

Define $x_e^k := x^k - \bar{x}$ for notational simplicity, and define y_e^k and z_e^k similarly. Let

$$\begin{aligned} u^{k+1} &:= z^{k+1} - (1-\tau)\lambda(h(x^{k+1}, y^{k+1}) - c) - \lambda B(y^k - y^{k+1}), \\ v^{k+1} &:= z^{k+1} - (1-\tau)\lambda(h(x^{k+1}, y^{k+1}) - c). \end{aligned}$$

Combining (B.4) and (B.5) with (B.3) and (B.11), we have

$$\langle A^*u^{k+1} - S(x^{k+1} - x^k) - A^*\bar{z}, x^{k+1} - \bar{x}\rangle \geq \|x^{k+1} - \bar{x}\|_{\Sigma_f}^2,$$
$$\langle B^*v^{k+1} - T(y^{k+1} - y^k) - B^*\bar{z}, y^{k+1} - \bar{y}\rangle \geq \|y^{k+1} - \bar{y}\|_{\Sigma_g}^2.$$

Adding up these two inequalities and using the definitions of u^{k+1} and v^{k+1} and the relation

$$h(x_e^{k+1}, y_e^{k+1}) = h(x^{k+1}, y^{k+1}) - c = (\tau\lambda)^{-1}(z^k - z^{k+1}) = (\tau\lambda)^{-1}(z_e^k - z_e^{k+1}),$$

we obtain that

(B.12)
$$\begin{aligned}&(\tau\lambda)^{-1}\langle z_e^{k+1}, z_e^k - z_e^{k+1}\rangle - (1-\tau)\lambda\|h(x^{k+1},y^{k+1})-c\|^2\\ &\quad + \lambda\langle B(y^{k+1}-y^k), h(x^{k+1},y^{k+1})-c\rangle - \lambda\langle B(y^{k+1}-y^k), By_e^{k+1}\rangle\\ &\quad - \langle S(x^{k+1}-x^k), x_e^{k+1}\rangle - \langle T(y^{k+1}-y^k), y_e^{k+1}\rangle \geq \|x_e^{k+1}\|_{\Sigma_f}^2 + \|y_e^{k+1}\|_{\Sigma_g}^2.\end{aligned}$$

Next, we shall estimate the term $\lambda\langle B(y^{k+1}-y^k), h(x^{k+1},y^{k+1})-c\rangle$ in (B.12). By noticing from (B.11) that

$$B^*v^{k+1} - T(y^{k+1}-y^k) \in \partial g(y^{k+1}) \quad \text{and} \quad B^*v^k - T(y^k - y^{k-1}) \in \partial g(y^k),$$

we obtain from the maximal monotonicity of $\partial g(\cdot)$ that

$$\begin{aligned}&\langle y^{k+1}-y^k, B^*v^{k+1} - T(y^{k+1}-y^k) - [B^*v^k - T(y^k-y^{k-1})]\rangle \geq 0\\ \Rightarrow\ &\langle B(y^{k+1}-y^k), v^{k+1}-v^k\rangle \geq \|y^{k+1}-y^k\|_T^2 - \langle y^{k+1}-y^k, T(y^k-y^{k-1})\rangle\,.\end{aligned}$$

Thus, by letting $\mu_{k+1} := (1-\tau)\lambda\langle B(y^{k+1}-y^k), h(x^k,y^k)-c\rangle$, we have

$$\begin{aligned}&\lambda\langle B(y^{k+1}-y^k), h(x^{k+1},y^{k+1})-c\rangle\\ &= (1-\tau)\lambda\langle B(y^{k+1}-y^k), h(x^{k+1},y^{k+1})-c\rangle + \langle B(y^{k+1}-y^k), z^k - z^{k+1}\rangle\\ &= \mu_{k+1} + \langle B(y^{k+1}-y^k), v^k - v^{k+1}\rangle\\ &\leq \mu_{k+1} - \|y^{k+1}-y^k\|_T^2 + \langle y^{k+1}-y^k, T(y^k-y^{k-1})\rangle\\ &\leq \mu_{k+1} - \frac{1}{2}\|y^{k+1}-y^k\|_T^2 + \frac{1}{2}\|y^k-y^{k-1}\|_T^2,\end{aligned}$$

which, together with (B.12), implies

$$\begin{aligned}&(\tau\lambda)^{-1}\langle z_e^{k+1}, z_e^k - z_e^{k+1}\rangle - (1-\tau)\lambda\|h(x^{k+1},y^{k+1})-c\|^2 + \mu_{k+1}\\ &\quad - \frac{1}{2}\|y^{k+1}-y^k\|_T^2 + \frac{1}{2}\|y^k-y^{k-1}\|_T^2 - \lambda\langle B(y^{k+1}-y^k), By_e^{k+1}\rangle\\ &\quad - \langle S(x^{k+1}-x^k), x_e^{k+1}\rangle - \langle T(y^{k+1}-y^k), y_e^{k+1}\rangle \geq \|x_e^{k+1}\|_{\Sigma_f}^2 + \|y_e^{k+1}\|_{\Sigma_g}^2.\end{aligned}$$

Using $z^{k+1}-z^k = -\tau\lambda[h(x^{k+1},y^{k+1})-c]$ and elementary relations $\langle u,v\rangle = \frac{1}{2}(\|u\|^2 + \|v\|^2 - \|u-v\|^2) = \frac{1}{2}(\|u+v\|^2 - \|u\|^2 - \|v\|^2)$, we further obtain that

(B.13)
$$\begin{aligned}&(\tau\lambda)^{-1}(\|z_e^k\|^2 - \|z_e^{k+1}\|^2) - (2-\tau)\lambda\|h(x^{k+1},y^{k+1})-c\|^2 + 2\mu_{k+1}\\ &- \|y^{k+1}-y^k\|_T^2 + \|y^k-y^{k-1}\|_T^2 - \lambda\|B(y^{k+1}-y^k)\|^2 - \lambda\|By_e^{k+1}\|^2 + \lambda\|By_e^k\|^2\\ &- \|x^{k+1}-x^k\|_S^2 - \|x_e^{k+1}\|_S^2 + \|x_e^k\|_S^2 - \|y^{k+1}-y^k\|_T^2 - \|y_e^{k+1}\|_T^2 + \|y_e^k\|_T^2\\ \geq\ & 2\|x_e^{k+1}\|_{\Sigma_f}^2 + 2\|y_e^{k+1}\|_{\Sigma_g}^2.\end{aligned}$$

Now we are ready to consider the two cases for the steplengths $\tau \in (0,1]$ and $\tau > 1$, respectively.

Case I. $\tau \in (0,1]$. In this case, by using the fact that

$$2\langle B(y^{k+1}-y^k), h(x^k,y^k)-c\rangle \leq \|B(y^{k+1}-y^k)\|^2 + \|h(x^k,y^k)-c\|^2$$

and the definition $2\mu_{k+1} = 2(1-\tau)\lambda\langle B(y^{k+1}-y^k), h(x^k,y^k)-c\rangle$, after some simple manipulations we can obtain from (B.13) that (B.8) holds.

Case II. $\tau > 1$. Similarly, by using the inequality

$$-2\langle B(y^{k+1}-y^k), h(x^k,y^k)-c\rangle \le \tau\|B(y^{k+1}-y^k)\|^2 + \tau^{-1}\|h(x^k,y^k)-c\|^2,$$

we know from (B.13) that (B.9) holds.

Assume that $\tau \in (0,(1+\sqrt{5})/2)$. We can now consider the convergence of the sequence $\{(x^k,y^k,z^k)\}$. From (B.7)–(B.9), we see immediately that the sequence $\theta(x^{k+1},y^{k+1},z^{k+1}) + \|y^{k+1}-y^k\|_T^2$ is bounded and

$$\text{(B.14)}\qquad \lim_{k\to\infty} t_{k+1} = 0 \text{ and } \lim_{k\to\infty}\|z^{k+1}-z^k\| = \lim_{k\to\infty}(\tau\lambda)^{-1}\|h(x^{k+1},y^{k+1})-c\| = 0.$$

By using the definitions of θ in (B.6) and t_{k+1} in (B.7), we see that the three sequences $\{\|z^{k+1}\|\}$, $\{\|y_e^{k+1}\|^2_{(\Sigma_g+T+\lambda B^*B)}\}$, and $\{\|x_e^{k+1}\|^2_{(\Sigma_f+S)}\}$ are all bounded. Since $\Sigma_g + T + \lambda B^*B$ is assumed to be positive definite, the sequence $\{\|y^{k+1}\|\}$ is also bounded. Furthermore, by using

$$\text{(B.15)}\quad \|Ax_e^{k+1}\| \le \|Ax_e^{k+1}+By_e^{k+1}\| + \|By_e^{k+1}\| = \|h(x^{k+1},y^{k+1})-c\| + \|By_e^{k+1}\|,$$

we also know that the sequence $\{\|Ax_e^{k+1}\|\}$ is bounded, and so is the sequence $\{\|x_e^{k+1}\|^2_{(\Sigma_f+S+\lambda A^*A)}\}$. This shows that the sequence $\{\|x^{k+1}\|\}$ is also bounded, as the operator $\Sigma_f + S + \lambda A^*A$ is assumed to be positive definite. Thus, the sequence $\{(x^k,y^k,z^k)\}$ is bounded.

Since the sequence $\{(x^k,y^k,z^k)\}$ is bounded, there exists a subsequence $\{(x^{k_i},y^{k_i}, z^{k_i})\}$ that converges to a cluster point, say $(x^\infty,y^\infty,z^\infty)$. We next show that (x^∞,y^∞) is an optimal solution to problem (B.1) and z^∞ is a corresponding Lagrange multiplier.

To this end, we first note from (B.14) that

$$\text{(B.16)}\qquad \begin{aligned} \lim_{k\to\infty}\|B(y^{k+1}-y^k)\| = 0, \qquad &\lim_{k\to\infty}\|(x^{k+1}-x^k)\|_S = 0,\\ \text{and}\quad &\lim_{k\to\infty}\|(y^{k+1}-y^k)\|_T = 0.\end{aligned}$$

By using the inequality

$$\|Ax^{k+1}+By^k-c\| \le \|Ax^{k+1}+By^{k+1}-c\| + \|B(y^{k+1}-y^k)\|,$$

we deduce from (B.16) and (B.14) that

$$\text{(B.17)}\qquad \lim_{k\to\infty}\|z^{k+1}-z^k\| = 0 \quad\text{and}\quad \lim_{k\to\infty}\|Ax^{k+1}+By^k-c\| = 0.$$

Taking limits on both sides of (B.11) along the subsequence $\{(x^{k_i},y^{k_i},z^{k_i})\}$, using (B.16), (B.17), and invoking the closedness of the graphs of ∂f and ∂g [3, p. 80], we obtain that

$$A^*z^\infty \in \partial f(x^\infty), \quad B^*z^\infty \in \partial g(y^\infty), \quad Ax^\infty + By^\infty - c = 0;$$

i.e., $(x^\infty,y^\infty,z^\infty)$ satisfies (B.3). From the discussion prior to the theorem, we conclude that (x^∞,y^∞) is an optimal solution to problem (B.1) and that z^∞ is a corresponding Lagrange multiplier.

To complete the proof of part (c), we show that $(x^\infty,y^\infty,z^\infty)$ is actually the unique limit of $\{(x^k,y^k,z^k)\}$. Recall that $(x^\infty,y^\infty,z^\infty)$ satisfies (B.3). Hence, we could replace $(\bar{x}, \bar{y}, \bar{z})$ with $(x^\infty, y^\infty, z^\infty)$ in the above arguments, starting from (B.4) and (B.5). Consequently, for $\tau \in (0, 1]$ the subsequence $\{\psi_{k_i} + (1-\tau)\lambda\|h(x^{k_i}, y^{k_i}) - c\|^2\}$ converges to 0 as $k_i \to \infty$, and for $\tau \in (1, (1+\sqrt{5})/2)$ the subsequence $\{\psi_{k_i} + (1-\tau^{-1})\lambda\|h(x^{k_i}, y^{k_i}) - c\|^2\}$ converges to 0 as $k_i \to \infty$. Since the two subsequences are from two nonincreasing sequences and since the sequence $\{\|h(x^k, y^k) - c\|\}$ converges to zero,

$$\lim_{k\to\infty} \theta(x^k, y^k, z^k) + ||y^k - y^{k-1}||_T^2 = \lim_{k\to\infty} \psi_k = 0. \tag{B.18}$$

From this, we see immediately that $\lim_{k\to\infty} z^k = z^\infty$. Moreover, we obtain from (B.18) and (B.14) that

$$\lim_{k\to\infty} (\|y_e^k\|_{\Sigma_g}^2 + \|y_e^k\|_T^2 + \lambda\|By_e^k\|^2) + (\|x_e^k\|_{\Sigma_f}^2 + \|x_e^k\|_S^2) = 0.$$

Hence, we have $\lim_{k\to\infty} y^k = y^\infty$ as the operator $\Sigma_g + T + \lambda B^* B$ is positive definite. This, together with (B.15), further implies $\lim_{k\to\infty} \|Ax_e^k\| = 0$. Thus, we have

$$\lim_{k\to\infty} \|x_e^k\|_{\Sigma_f}^2 + \|x_e^k\|_S^2 + \lambda\|Ax_e^k\|^2 = 0,$$

which, from the fact that the operator $\Sigma_f + S + \lambda A^* A$ is positive definite, ensures that $\lim_{k\to\infty} x^k = x^\infty$. Therefore, we have shown that the whole sequence $\{(x^k, y^k, z^k)\}$ converges to $(x^\infty, y^\infty, z^\infty)$ if $\tau \in (0, (1+\sqrt{5})/2)$.

Finally, by observing that the third and the fourth terms on the left-hand side of (B.12) cancel each other out if A is vacuous, we can easily start from (B.12) to get (B.10). The convergence of $\{(x^k, y^k, z^k)\}$ can be obtained in a similar but simpler manner as in the proof for part (c). $\square$

Remark B.1. Theorem B.1 provides general conditions for the convergence of the proximal ADMM. It includes some recent results in the literature as special cases. For example, the convergence of the proximal ADMM has been considered by Attouch and Soueycatt [1] for the case $S = \frac{1}{\lambda}\mathcal{I}$ and $T = \frac{1}{\lambda}\mathcal{I}$. Moreover, Zhang, Burger, and Osher [59] considered the case when $B = \mathcal{I}$ and S is chosen to be positive definite and established a slightly weaker convergence result, that is, that all the cluster points of $\{(x^k, y^k)\}$ and $\{z^k\}$ are optimal solutions to the primal and dual problems, respectively. Furthermore, Yang and Zhang [58] applied the proximal ADMM to solve l_1-norm minimization problems in compressive sensing, allowing the step length to be chosen under some conditions other than only 1. It is notable that when the step length is chosen to be 1, the iteration scheme in [58] simply reduces to (B.2) with $A = \mathcal{I}$ and a self-adjoint and positive definite operator T. It is also not hard to see that Theorem B.1 covers all three convergence results in the aforementioned work of Xu and Wu [57].

Remark B.2. Suppose that f takes the form of $f_0 + p$ with f_0 being a convex quadratic function whose Hessian is Σ and p being a "simple" closed proper convex function whose proximal mapping, under the weighted inner product $\langle\cdot,\cdot\rangle_D := \langle\cdot, D\,\cdot\rangle$ for a given self-adjoint and positive definite operator D, admits an efficient algorithm. Then the condition in Theorem B.1 actually allows the flexibility to introduce an S to "cancel" out also the second-order term in f_0, in addition to the second-order term in the quadratic penalty, such that

$$S + (\Sigma + \lambda A^* A) = \alpha D,$$

where α is the smallest positive number such that S is positive semidefinite. That is, S makes up the difference between αD and $\Sigma + \lambda A^* A$. In the extreme case of

$\Sigma + \lambda A^*A = \beta D$ for some positive number β, one may and should just take $S = 0$. These comments on f are also applicable to the function g. Furthermore, if either the x-part or the y-part disappears, one can allow the step length τ to take values in the larger interval $(0, 2)$.

Appendix C. Subspace identification algorithm for system identification. In this appendix, we discuss how the rank of the matrix $\mathcal{H}(y)$ is related to the system order in the system identification problem in section 4. We will follow closely the discussion in [32, section 5]. Based on this discussion, we describe a version of the classical subspace method as used in Step 3 of our approach for system identification, discussed in section 4.1.2.

Notice that if there is no output measurement noise, then (4.1) is satisfied with $y = \tilde{y}$. Assume further that r is larger than the true order of the system. Then we obtain from (4.1) that

$$H_{m,1,r+1,N+1-r}(y) = GX + LH_{p,1,r+1,N+1-r}(u), \tag{C.1}$$

where $X = \begin{pmatrix} x_0 & x_1 & \cdots & x(N-r) \end{pmatrix}$ and

$$G = \begin{pmatrix} C \\ CA \\ \vdots \\ CA^r \end{pmatrix}, \qquad L = \begin{pmatrix} D & 0 & 0 & \cdots & 0 \\ CB & D & 0 & \cdots & 0 \\ CAB & CB & D & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{r-1}B & CA^{r-2}B & \cdots & \cdots & D \end{pmatrix}. \tag{C.2}$$

Thus, if $U^\perp$ denotes the orthogonal matrix whose columns span the nullspace of $H_{p,1,r+1,N+1-r}(u)$, by multiplying both sides of (C.1) by $U^\perp$, we get

$$\mathcal{H}(y) = H_{m,1,r+1,N+1-r}(y)U^\perp = GXU^\perp. \tag{C.3}$$

Hence, the rank of $\mathcal{H}(y)$ equals the rank of G (which is equal to the true system order) if there is no rank cancellation in the product $GXU^\perp$. This latter property holds generically for a randomly chosen input sequence u.

The subspace method for system identification with noisy output, which we use in Step 3 of our algorithm, can be derived based on the above relations. We describe the method below.

- Set the threshold for the (numerical) rank of the matrix $\mathcal{H}(y_\mu)$ at some suitable integer n. This will be the approximate system order. Motivated by (C.3), we next find $\hat{G} \in \mathbb{R}^{m(r+1)\times n}$ whose columns span the range of the rank-n approximation of $\mathcal{H}(y_\mu)$, where

$$\hat{G} = \begin{pmatrix} \hat{G}_0 \\ \hat{G}_1 \\ \vdots \\ \hat{G}_r \end{pmatrix},$$

  with each $\hat{G}_i \in \mathbb{R}^{m\times n}$ for all $i = 0, \ldots, r$.
- Motivated by (C.2), we then set C to be the first block of $\hat{G}$; i.e., we set $C = \hat{G}_0$.

- Again motivated by the definition of G in (C.2), we solve the following system in the least-squares sense to obtain A:

$$\begin{pmatrix} \hat{G}_1 \\ \vdots \\ \hat{G}_r \end{pmatrix} = \begin{pmatrix} \hat{G}_0 \\ \vdots \\ \hat{G}_{r-1} \end{pmatrix} A.$$

- With the above A and C, we solve for B, D, and x_0 in the least-squares sense using the following system of equations, which is essentially (C.1) but with y replaced by the noisy output $\tilde{y}$:

$$CA^t x_0 + \sum_{k=0}^{t-1} CA^{t-k-1} Bu_k + Du_t = \tilde{y}_t, \qquad t = 0, \ldots, N.$$

Acknowledgments. The first and second authors would like to thank Zhang Liu and Lieven Vandenberghe for many useful discussions. The third author would like to thank his Ph.D. student Mr. Weimin Miao at the National University of Singapore for discussions on the convergence of alternating direction methods of multipliers. We would also like to thank the anonymous referees for their many suggestions that help improve the manuscript. This work was initiated by the first, third, and fourth authors in May, 2009, when the third author was spending his sabbatical leave with Professor Paul Tseng at the University of Washington, Seattle. With Professor Tseng missing from August, 2009, it seemed that this work would never be finished. Fortunately, the second author, Professor Tseng's last Ph.D. student, kindly agreed to join our force to get this work done.

REFERENCES

[1] H. Attouch and M. Soueycatt, *Augmented Lagrangian and proximal alternating direction methods of multipliers in Hilbert spaces. Applications to games, PDE's and control*, Pac. J. Optim., 5 (2009), pp. 17–37.

[2] D. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Prentice–Hall, Englewood Cliffs, NJ, 1989.

[3] J. M. Borwein and A. S. Lewis, *Convex Analysis and Nonlinear Optimization*, 2nd ed., Springer, New York, 2006.

[4] J. A. Cadzow, *Signal enhancement—A composite property mapping algorithm*, IEEE Trans. Acoust. Speech Signal Process., 36 (1988), pp. 49–62.

[5] J.-F. Cai, E. J. Candès, and Z. Shen, *A singular value thresholding algorithm for matrix completion*, SIAM J. Optim., 20 (2010), pp. 1956–1982.

[6] E. J. Candés and Y. Plan, *Tight oracle bounds for low-rank matrix recovery from a minimal number of random measurements*, IEEE Trans. Inform. Theory, 57 (2011), pp. 2342–2359.

[7] E. J. Candés and Y. Plan, *Matrix completion with noise*, Proc. IEEE, 98 (2010), pp. 925–936.

[8] E. J. Candés and B. Recht, *Exact matrix completion via convex optimization*, Found. Comput. Math., 9 (2009), pp. 717–772.

[9] S. Ma, D. Goldfarb, and L. Chen, *Fixed point and Bregman iterative methods for matrix rank minimization*, Math. Program., 128 (2011), pp. 321–353.

[10] B. De Moor, *Total least squares for affinely structured matrices and the noisy realization problem*, IEEE Trans. Signal Process., 42 (1994), pp. 3104–3113.

[11] B. De Moor, P. De Gersem, B. De Schutter, and W. Favoreel, *DaISy: A database for the identification of systems*, Journal A, 38 (1997), pp. 4–5.

[12] B. De Schutter, *Minimal state-space realization in linear system theory: An overview*, J. Comput. Appl. Math., 121 (2000), pp. 331–354.

[13] T. Ding, M. Sznaier, and O. I. Camps, *A rank minimization approach to video inpainting*, in Proceedings of the IEEE Conference on Computer Vision, Rio de Janeiro, Brazil, 2007, pp. 1–8.

[14] J. Eckstein, *Some saddle-function splitting methods for convex programming*, Optim. Methods Softw., 4 (1994), pp. 75–83.

[15] J. Eckstein and D. P. Bertsekas, *On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators*, Math. Prog., 55 (1992), pp. 293–318.

[16] M. Elad and P. Milanfar, *Shape from moments—An estimation theory perspective*, IEEE Trans. Signal Process., 52 (2004), pp. 1814–1829.

[17] M. Fazel, *Matrix Rank Minimization with Applications*, Ph.D. Thesis, Department of Electrical Engineering, Stanford University, Stanford, CA, 2002.

[18] M. Fazel, H. Hindi, and S. Boyd, Log-det *heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices*, in Proceedings of the American Control Conference, Denver, CO, 2003, pp. 2156–2162.

[19] M. Fazel, H. Hindi, and S. Boyd, *A rank minimization heuristic with application to minimum order system approximation*, in Proceedings of the American Control Conference, Arlington, VA, 2001, pp. 4734–4739.

[20] M. Fortin and R. Glowinski, *On decomposition-coordination methods using an augmented Lagrangian*, in Augmented Lagrangian Methods: Applications to the Solution of Boundary Problems, M. Fortin and R. Glowinski, eds., North–Holland, Amsterdam, 1983, pp. 97–146.

[21] D. Gabay, *Applications of the method of multipliers to variational inequalities*, in Augmented Lagrangian Methods: Applications to the Solution of Boundary Problems, M. Fortin and R. Glowinski, eds., North–Holland, Amsterdam, 1983, pp. 299–331.

[22] D. Gabay and B. Mercier, *A dual algorithm for the solution of nonlinear variational problems via finite element approximations*, Comput. Math. Appl. 2 (1976), pp. 17–40.

[23] R. Glowinski and A. Marroco, *Sur l'approximation, par elements finis d'ordre un, et la resolution, par penalisation-dualit'e, d'une classe de problemes de Dirichlet non lineares*, Revue Francaise d'Automatique, Informatique et Recherche Op'erationelle, 9 (R-2) (1975), pp. 41–76.

[24] G. H. Golub, P. Milanfar, and J. Varah, *A stable numerical method for inverting shape from moments*, SIAM J. Sci. Comput., 21 (1999), pp. 1222–1243.

[25] G. H. Golub and C. F. Van Loan, *Matrix Computation*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.

[26] B. He, L. Liao, D. Han, and H. Yang, *A new inexact alternating directions method for monotone variational inequalities*, Math. Program., 92 (2002), pp. 103–118.

[27] R. H. Keshavan, A. Montanari, and S. Oh, *Matrix completion from a few entries*, IEEE Trans. Inform. Theory, 56 (2010), pp. 2980–2998.

[28] J. B. Lasserre, *Global optimization with polynomials and the problem of moments*, SIAM J. Optim., 11 (2001), pp. 796–817.

[29] M. Laurent, *Sums of squares, moment matrices and optimization over polynomials*, in Emerging Applications of Algebraic Geometry, IMA Vol. Math. Appl. 149, M. Putinar and S. Sullivant, eds., Springer, New York, 2009, pp. 157–270.

[30] K. Lee and Y. Bresler, *ADMiRA: Atomic decomposition for minimum rank approximation*, IEEE Trans. Inform. Theory, 56 (2010), pp. 4402–4416.

[31] Y. Liu, D. F. Sun, and K.-C. Toh, *An implementable proximal point algorithmic framework for nuclear norm minimization*, Math. Program., 133 (2012), pp. 399–436.

[32] Z. Liu and L. Vandenberghe, *Interior-point method for nuclear norm approximation with application to system identification*, SIAM. J. Matrix Anal., 31 (2009), pp. 1235–1256.

[33] Z. Liu and L. Vandenberghe, *Semidefinite programming methods for system realization and identification*, in Proceedings of the 48th IEEE Conference on Decision and Control, 2009, pp. 4676–4681.

[34] L. Ljung, *System Identification: Theory for the User*, Prentice–Hall, Englewood Cliffs, NJ, 1999.

[35] J. Mari, P. Stoica, and T. McKelvey, *Vector ARMA estimation: A reliable subspace approach*, IEEE Trans. Signal Process., 48 (2000), pp. 2092–2104.

[36] I. Markovsky, J. C. Willems, S. Van Huffel, and B. De Morr, *Exact and Approximate Modeling of Linear Systems: A Behavioral Approach*, SIAM Monogr. Math. Model. Comput. 11, SIAM, Philadelphia, 2006.

[37] I. Markovsky, J. C. Willems, S. Van Huffel, B. De Morr, and R. Pintelon, *Application of structured total least squares for system identification and model reduction*, IEEE Trans. Automat. Control., 50 (2005), pp. 1490–1500.

[38] R. Meka, P. Jain, and I. S. Dhillon, *Guaranteed rank minimization via singular value projection*, in Proceedings of the Neural Information Processing Systems Conference, 2010, pp. 937–945.

[39] P. Milanfar, G. Verghese, W. Karl, and A. Willsky, *Reconstructing polygons from moments with connections to array processing*, IEEE Trans. Signal Process., 43 (1995), pp. 432–443.

[40] K. Mohan and M. Fazel, *Reweighted nuclear norm minimization with application to system identification*, in Proceedings of the American Control Conference, 2010, pp. 2953–2959.

[41] Y. Nesterov, *A method for solving a convex programming problem with convergence rate* $O(1/k^2)$, Soviet Math. Dokl., 27 (1983), pp. 372–376.

[42] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2003.

[43] Yu. Nesterov, *Excessive gap technique in nonsmooth convex minimization*, SIAM J. Optim., 16 (2005), pp. 235–249.

[44] Y. Nesterov, *Smooth minimization of non-smooth functions*, Math. Program., 103 (2005), pp. 127–152.

[45] P. A. Parrilo, *Semidefinite programming relaxations for semialgebraic problems*, Math. Program., 96 (2003), pp. 293–320.

[46] T. K. Pong, P. Tseng, S. Ji, and J. Ye, *Trace norm regularization: Reformulations, algorithms, and multi-task learning*, SIAM J. Optim., 20 (2010), pp. 3465–3489.

[47] B. Recht, M. Fazel, and P. A. Parrilo, *Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization*, SIAM Rev., 52 (2010), pp. 471–501.

[48] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[49] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*, Springer, New York, 1998.

[50] M. Schuermans, P. Lemmerling, L. De Lathauwer, and S. Van Huffel, *The use of total least squares data fitting in the shape from moments problem*, Signal Process., 86 (2006), pp. 1109–1115.

[51] E. D. Sontag, *Mathematical Control Theory: Deterministic Finite Dimensional Systems*, 2nd ed., Springer, New York, 1998.

[52] K.-C. Toh and S. Yun, *An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems*, Pac. J. Optim., 6 (2010), pp. 615–640.

[53] P. Tseng, *Approximation accuracy, gradient methods, and error bound for structured convex optimization*, Math. Program., 125 (2010), pp. 263–295.

[54] P. Van Overschee and B. De Moor, *Subspace Identification for Linear Systems: Theory, Implementation, Applications*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.

[55] J. C. Willems, *From time series to linear system—Part* I. *Finite dimensional linear time invariant systems*, Automatica, 22 (1986), pp. 561–580.

[56] J. C. Willems, *From time series to linear system—Part* II. *Exact modelling*, Automatica, 22 (1986), pp. 675–694.

[57] M. H. Xu and T. Wu, *A class of linearized proximal alternating direction methods*, J. Optim. Theory Appl., 151 (2011), pp. 321–337.

[58] J. Yang and Y. Zhang, *Alternating direction algorithms for* ℓ_1*-problems in compressive sensing*, SIAM J. Sci. Comput., 33 (2011), pp. 250–278.

[59] X. Zhang, M. Burger, and S. Osher, *A unified primal-dual algorithm framework based on Bregman iteration*, J. Sci. Comput., 46 (2011), pp. 20–46.