Nuclear norm system identification with missing inputs and outputs

Zhang Liu[a,*], Anders Hansson[b,1], Lieven Vandenberghe[c]

[a] *Northrop Grumman Corporation, 16710 Via Del Campo Court, San Diego, CA 92127, USA*
[b] *Division of Automatic Control, Linköping University, SE–581 83 Linköping, Sweden*
[c] *Electrical Engineering Department, UCLA, 66–147L Engineering IV, Los Angeles, CA 90095, USA*

Abstract

We present a system identification method for problems with partially missing inputs and outputs. The method is based on a subspace formulation and uses the nuclear norm heuristic for structured low-rank matrix approximation, with the missing input and output values as the optimization variables. We also present a fast implementation of the alternating direction method of multipliers (ADMM) to solve regularized or non-regularized nuclear norm optimization problems with Hankel structure. This makes it possible to solve quite large system identification problems. Experimental results show that the nuclear norm optimization approach to subspace identification is comparable to the standard subspace methods when no inputs and outputs are missing, and that the performance degrades gracefully as the percentage of missing inputs and outputs increases.

Keywords: nuclear norm, system identification, subspace method, Hankel structure, low-rank matrix approximation

1. Introduction

Nuclear norm optimization methods for structured low-rank matrix approximation have been discussed in several recent papers on system identification. The idea was first proposed by Fazel, Hindi, and Boyd [1, 2], who pointed out the benefits of minimizing the nuclear norm (sum of singular values) of a matrix-valued function as a convex heuristic for minimizing its rank. Replacing the rank of a matrix by its nuclear norm can be justified as a convex relaxation (the nuclear norm $\|X\|_* = \sum_i \sigma_i(X)$ is the largest convex lower bound of $\mathbf{rank}(X)$ on the ball $\{X \mid \|X\|_2 = \sigma_1(X) \leq 1\}$); see [1, theorem 1]. It is further motivated by the empirical observation that minimum nuclear norm solutions often have low rank. Moreover in certain applications (for example, low-rank matrix completion) the quality of the heuristic can be demonstrated analytically [3, 4, 5, 6].

[*] Corresponding author

Email addresses: zhang.liu@gmail.com (Zhang Liu), anders.g.hansson@liu.se (Anders Hansson), lieven.vandenberghe@ucla.edu (Lieven Vandenberghe)

[1] This work was carried out while the author was a Visiting Professor at UCLA.

As a practical technique for making low-rank matrix approximations, the nuclear norm approach offers the advantage that it preserves linear structure in the matrix approximation, unlike the singular value decomposition (SVD) commonly used in identification methods. It is especially useful for low-rank approximation problems with additional convex constraints or convex regularization terms in the cost function.

In this paper, we first evaluate a nuclear norm variant of modern subspace identification algorithms [7, 8]. Earlier experiments with nuclear norm formulations of a basic subspace method have indicated that the technique can be quite effective for system identification [9, 10, 11, 12]. The basic formulation studied in these papers did not include the instrumental variables and matrix weights that are commonly used in state-of-the-art subspace methods. These features were added to the method in [13]. We present experiments with randomly generated data sets and data sets from the DaISy benchmark collection [14]. The results show a modest improvement over SVD based subspace methods.

As several authors have pointed out, nuclear norm approximation is a promising technique for estimation with missing data. Applications of low-rank Hankel matrix completion via nuclear norm optimization are discussed in [15, 16, 17]. As a second contribution in this paper, we therefore describe and evaluate a subspace identification algorithm for identification problems with partially missing inputs and outputs. The method is based on minimizing the nuclear norm of the stacked input and output Hankel matrices. The experiments show that the performance degrades slowly as the percentage of missing inputs and outputs increases. In several instances a very high number of missing data (up to 50%) can be tolerated.

We use the alternating direction method of multipliers (ADMM) to solve regularized and non-regularized nuclear norm optimization problems [12]. The third contribution of the paper is to describe two techniques that improve the efficiency of the ADMM for regularized nuclear norm minimization in identification. We show that Hankel structure can be exploited to speed up a key step in the algorithm. Another improvement substantially reduces the amount of work when an entire regularization path is computed.

Outline and notation. The paper is organized as follows. We review the most common subspace identification algorithms in section 2 and then formulate nuclear norm variants of these methods in section 3. The ADMM implementation is discussed in section 4. Section 5 contains the identification experiments.

We will frequently encounter block Hankel matrices constructed from sequences of vectors. The notation $H_{i,j,k}$ will be used to denote the $j \times k$ block Hankel matrix

$$H_{i,j,k} = \begin{bmatrix} h(i) & h(i+1) & h(i+2) & \dots & h(i+k-1) \\ h(i+1) & h(i+2) & h(i+3) & \dots & h(i+k) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h(i+j-1) & h(i+j) & h(i+j+1) & \dots & h(i+j+k-2) \end{bmatrix} \tag{1}$$

where $h(t)$ is a sequence of vectors.

2. Subspace system identification

In this section we review the basic ideas of subspace identification methods for linear time-invariant systems [7, 8]. These methods estimate linear state-space models with process and measurement noise. Without loss of generality we can adopt the Kalman normal form

$$\begin{array}{rcl} x(k+1) & = & Ax(k)+Bu(k)+Ke(k) \\ y(k) & = & Cx(k)+Du(k)+e(k), \end{array} \tag{2}$$

with $x(k)\in\mathbf{R}^{n_x}$, $u(k)\in\mathbf{R}^{n_m}$, $e(k)\in\mathbf{R}^{n_p}$, and $y(k)\in\mathbf{R}^{n_p}$ [7, page 99]. It is assumed that $e(k)$ is ergodic, zero-mean, white noise.

The starting point of the derivation is the matrix equation

$$Y_{0,r,N}=O_rX_{0,1,N}+S_rU_{0,r,N}+E \tag{3}$$

which follows from the state-space equations (2). The matrices $Y_{0,r,N}$ and $U_{0,r,N}$ are block Hankel matrices constructed from the sequences $y(k)$, $u(k)$ for $k=0,\ldots,r+N-2$, using the notation (1). The matrix $X_{0,1,N}$ has as its columns the states $x(k)$, $k=0,\ldots,N-1$. The matrices O_r and S_r are defined as

$$O_r=\begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{r-1}\end{bmatrix}, \qquad S_r=\begin{bmatrix} D & 0 & 0 & \ldots & 0 \\ CB & D & 0 & \cdots & 0 \\ CAB & CB & D & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{r-2}B & CA^{r-3}B & CA^{r-4}B & \ldots & D\end{bmatrix},$$

and E contains the contribution of the noise sequence $e(k)$ to the output. The subspace methods discussed in this paper have in common that they first estimate the range space of the extended observability matrix O_r and then determine a system realization from the estimate of range(O_r). (These methods therefore require that r is taken greater than n_x; the column dimension N then follows from r and the length of the input and output sequences. The values of r used in our experiments will be mentioned in section 5.) Subspace algorithms of this type are described in detail in [7, §10.6] and [8, chapter 9]. Other approaches, which first estimate the matrix of states $X_{0,1,N}$ (see, for example, [18]) will not be discussed here.

2.1. Extended observability matrix

Basic algorithm. In the simplest variant the matrix $Y_{0,r,N}$ is multiplied on the right with a projection matrix that projects on the nullspace of $U_{0,r,N}$ [19]. This gives the equation

$$Y_{0,r,N}\Pi_{0,r,N}=O_rX_{0,1,N}\Pi_{0,r,N}+E\Pi_{0,r,N}, \tag{4}$$

where $\Pi_{0,r,N}$ is the orthogonal projection matrix on the nullspace of $U_{0,r,N}$. Suppose the matrix $X_{0,1,N}\Pi_{0,r,N}$ has full row rank, *i.e.*, $\mathbf{rank}\,X_{0,1,N}=n_x$ and no rank cancellation occurs in the product $X_{0,1,N}\Pi_{0,r,N}$. Equivalently,

$$\mathbf{rank}\begin{bmatrix} X_{0,1,N} \\ U_{0,r,N}\end{bmatrix}=n_x+\mathbf{rank}\,U_{0,r,N}. \tag{5}$$

This condition holds generically when the inputs are chosen at random. Under this assumption the first term on the right-hand side of (4) has rank n_x and its range equals the range of O_r. In the absence of noise ($E = 0$), one therefore has

$$n_x = \mathbf{rank}\left(Y_{0,r,N}\Pi_{0,r,N}\right), \qquad \mathrm{range}(O_r) = \mathrm{range}\left(Y_{0,r,N}\Pi_{0,r,N}\right).$$

In the presence of noise ($E \neq 0$), these identities hold only approximately and one can estimate n_x and range(O_r) from a low-rank approximation of $Y_{0,r,N}\Pi_{0,r,N}$, obtained by truncating an SVD.

An efficient implementation of this scheme is the MOESP (MIMO Output-Error State-Space) algorithm [20]. In this method one first computes an LQ factorization

$$\begin{bmatrix} U_{0,r,N} \\ Y_{0,r,N} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} \tag{6}$$

of the stacked input and output Hankel matrices. The diagonal blocks L_{11} and L_{22} are triangular matrices of order rn_m and rn_p, respectively. The matrices Q_1 and Q_2 have N columns and satisfy $Q_1Q_1^T = I$, $Q_2Q_2^T = I$, $Q_1Q_2^T = 0$. We have $\Pi_{0,r,N} = I - Q_1^TQ_1$ and

$$Y_{0,r,N}\Pi_{0,r,N} = (L_{21}Q_1 + L_{22}Q_2)(I - Q_1^TQ_1) = L_{22}Q_2.$$

Hence

$$\mathrm{range}(Y_{0,r,N}\Pi_{0,r,N}) = \mathrm{range}(L_{22})$$

and the range space of O_r can be estimated from an SVD of L_{22}.

Instrumental variables. The basic projection method described in the previous paragraph is not consistent: the range of $Y_{0,r,N}\Pi_{0,r,N}$ does not necessarily converge to the range of O_r as N goes to infinity. This deficiency can be resolved by the use of *instrumental variables* [21, 22]. We define an instrumental variable matrix

$$\Phi = \begin{bmatrix} U_{-s,s,N} \\ Y_{-s,s,N} \end{bmatrix} \tag{7}$$

by combining Hankel matrices of 'past' inputs and outputs. (More generally, one can use different row dimensions for the two Hankel matrices in Φ, but we will take them equal for simplicity. In the experiments of section 5 we will use $s = r$.) Multiplying (4) on the right with Φ^T gives

$$Y_{0,r,N}\Pi_{0,r,N}\Phi^T = O_rX_{0,1,N}\Pi_{0,r,N}\Phi^T + E\Pi_{0,r,N}\Phi^T.$$

It can be shown that $\lim_{N\to\infty}(1/N)E\Pi_{0,r,N}\Phi^T = 0$ and that, under weak assumptions, the limit

$$\lim_{N\to\infty}\frac{1}{N}X_{0,1,N}\Pi_{0,r,N}\Phi^T$$

has full rank n_x (see [8, §9.6] for a detailed discussion). As a consequence, the range of $Y_{0,r,N}\Pi_{0,r,N}\Phi^T$ gives a consistent estimate of the range of O_r. In practice, for finite N, a truncated SVD of $Y_{0,r,N}\Pi_{0,r,N}\Phi^T$ is used to estimate range(O_r).

As for the basic projection method, the instrumental variable scheme can be implemented using an LQ factorization

$$\begin{bmatrix} U_{0,r,N} \\ \Phi \\ Y_{0,r,N} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix} \tag{8}$$

of the stacked input and output Hankel matrices [8, section 9.6.1]. The diagonal blocks in L are triangular of order rn_m, $s(n_m + n_p)$, and rn_p. The matrices Q_i have column dimension N and satisfy the orthogonality properties $Q_i Q_i^T = I$ and $Q_i Q_j^T = 0$ for $i \neq j$. It is readily shown that

$$\begin{aligned} Y_{0,r,N}\Pi_{0,r,N}\Phi^T &= (L_{31}Q_1 + L_{32}Q_2 + L_{33}Q_3)(I - Q_1^T Q_1)(L_{21}Q_1 + L_{22}Q_2)^T \\ &= (L_{31}Q_1 + L_{32}Q_2 + L_{33}Q_3)Q_2^T L_{22}^T \\ &= L_{32}L_{22}^T. \end{aligned}$$

The dominant left singular vectors of $Y_{0,r,N}\Pi_{0,r,N}\Phi^T$ can be therefore computed from an SVD of $L_{32}L_{22}^T$.

Weight matrices. The accuracy of subspace methods can be further improved by multiplying the matrix $Y_{0,r,N}\Pi_{0,r,N}\Phi^T$ on both sides with nonsingular weight matrices before computing an SVD. The most general scheme therefore involves a matrix of the form

$$G = W_1 Y_{0,r,N}\Pi_{0,r,N}\Phi^T W_2. \tag{9}$$

or, equivalently, $G = W_1 L_{32} L_{22}^T W_2$ with L_{22} and L_{32} defined in (8). After truncating the SVD

$$G = \begin{bmatrix} P & P_e \end{bmatrix} \begin{bmatrix} \Sigma & 0 \\ 0 & \Sigma_e \end{bmatrix} \begin{bmatrix} Q & Q_e \end{bmatrix}^T, \tag{10}$$

by discarding the smallest singular values Σ_e one obtains an estimate of the range of O_r:

$$\mathrm{range}(O_r) \approx \mathrm{range}(W_1^{-1} P). \tag{11}$$

Four major variants (PO-MOESP, N4SID, IVM, CVA) of this method have been proposed, which differ by the choice of weight matrices W_1 and W_2. Here we only mention the expressions for the PO-MOESP and IVM methods and refer the reader to [22], [7, page 351], [8, §9.6.4] for details on the other two methods.

- *PO-MOESP* [23]. The PO-MOESP (Past Outputs MOESP) algorithm uses the weights

$$W_1 = I, \qquad W_2 = (\Phi\Pi_{0,r,N}\Phi^T)^{-1/2}.$$

- *IVM* [24]. The IVM (Instrumental Variable Method) uses the weights

$$W_1 = \left(Y_{0,r,N}\Pi_{0,r,N}Y_{0,r,N}^T\right)^{-1/2}, \qquad W_2 = \left(\Phi\Phi^T\right)^{-1/2}.$$

Note that the weights can be expressed in several equivalent forms. In particular, one can assume without loss of generality that W_1 and W_2 are symmetric positive definite. We also note that the size of the matrix G in (9) is $rn_p \times s(n_m + n_p)$. This is typically much smaller than the dimension $rn_p \times N$ of the matrix $Y_{0,r,N}\Pi_{0,r,N}$ used in the basic method.

2.2. System realization

Once an estimate of range(O_r) has been determined as described in the previous section, it is straightforward to calculate a system realization and an estimate of the initial state (as well as the entire state sequence). Several methods have been proposed for this that differ in the order in which the estimates of the system matrices and state sequence are computed; see [7, Section 10.6] and [8, Section 9.6.2]. Here we outline the main steps of the realization method described in [7].

Let $V \in \mathbf{R}^{rn_p \times n_x}$ be a matrix whose columns form a basis of our estimate of range(O_r). Partition V in r block rows $V_0, \ldots, V_{r-1}$ of size $n_p \times n_x$. Then one can take as estimates of C and A the matrices

$$\hat{C} = V_0, \qquad \hat{A} = \operatorname{argmin} \sum_{i=1}^{r-1} \|V_i - V_{i-1}\hat{A}\|_F^2, \tag{12}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. From $\hat{C}$ and $\hat{A}$, estimates of B, D, and $x(0)$ can be computed by solving a least-squares problem:

$$(\hat{B}, \hat{D}, \hat{x}_0) = \operatorname{argmin} \sum_{k=0}^{N+r-2} \|\hat{C}\hat{A}^k\hat{x}_0 + \sum_{i=0}^{k-1} \hat{C}\hat{A}^{k-i}\hat{B}u(i) + \hat{D}u(k) - y(k)\|_2^2. \tag{13}$$

If a model of the noise in (2) is required, it can be obtained by first estimating the state sequence $X_{0,1,N}$ and from this an estimate of the process and measurement noise covariances. The Kalman gain K can then be determined by solving a discrete-time Riccati equation (see [8, page 333]).

3. Identification by nuclear norm optimization

The key step in the subspace methods described above is an SVD of the matrix G defined in (9), used to estimate the range of the extended observability matrix. The use of instrumental variables guarantees that the estimate is consistent, *i.e.*, range(O_r) is estimated correctly in the limit as N goes to infinity. However for finite data there is no guarantee of optimality. In particular, a matrix $V \in \mathbf{R}^{rn_p \times n_x}$ whose columns span the subspace range$(W_1^{-1}P)$ defined in (11), does not necessarily possess the shift structure $V_i = V_{i-1}A$ between the block rows V_i of an extended observability matrix.

The reliance on the SVD for the low-rank approximation also makes it difficult to extend the subspace methods to problems with missing input or output measurement data (for which parts of the matrices $Y_{0,r,N}$ and $U_{0,r,N}$ are unknown), to incorporate prior knowledge (for example, bounds on the outputs), or to add regularization terms on the model outputs and inputs. Minimizing the nuclear norm provides an interesting alternative, as a heuristic for low-rank approximation problems that cannot be handled via an SVD, in particular, approximation problems with structured low-rank matrices and problems that include additional constraints or objectives. In this section, we discuss several variations of the subspace methods of section 2 based on this heuristic. We focus on applications to identification with missing data. Various other applications of the nuclear norm heuristic in system identification are discussed in [25, 15, 16, 17, 11].

Complete inputs and outputs. We first consider an identification problem with complete data. Following the approach in [9] we distinguish between the model outputs $y(k)$, which will be the optimization variables in the formulation, and the measured data $u_{\mathrm{meas}}(k)$, $y_{\mathrm{meas}}(k)$. The model outputs $y(k)$ are computed by solving a regularized nuclear norm problem

$$\text{minimize} \quad \|G(\mathbf{y})\|_* + \lambda \sum_{k \in T} \|y(k) - y_{\mathrm{meas}}(k)\|_2^2. \tag{14}$$

The optimization variable is the sequence $\mathbf{y} = (y(0), \ldots, y(N+r-2))$. The first term in the objective is the nuclear norm of the matrix

$$G(\mathbf{y}) = W_1 Y_{0,r,N} \Pi_{0,r,N} \Phi^T W_2$$

defined as in (4), where we use the measured inputs and outputs to construct W_1, W_2, $\Pi_{0,r,N}$, and Φ, and define $Y_{0,r,N}$ as the Hankel matrix constructed from the model outputs $y(k)$. Therefore $G(\mathbf{y})$ is a linear function of $\mathbf{y}$. The second term in the objective is a quadratic penalty on the deviation between the computed model outputs and the measurement data. The index set T is defined as $T = \{0, 1, \ldots, N+r-2\}$ and λ is a positive weight. In the formulation (14) we try to find values of the outputs that are close to the measured values $y_{\mathrm{meas}}(k)$ and make the matrix $G(\mathbf{y})$ low-rank (without guaranteeing that we minimize the rank of $G(\mathbf{y})$). After computing the sequence $\mathbf{y}$, one can use $G(\mathbf{y})$ as G in (9) to obtain an estimate of the range of the extended observability matrix and proceed with a system realization as described in section 2.2.

In the experiments of section 5 we will determine λ by solving the problem for a range of values of λ and choosing the model with the best fit on a validation data sequence.

Missing outputs. The formulation (14) is easily extended to problems where part of the measured output sequence $y_{\mathrm{meas}}(k)$ is missing. In this case we define T as the set of indices for which $y_{\mathrm{meas}}(k)$ is available. A second difference is that we exclude the outputs from the instrumental variable and use $\Phi = U_{-s,s,N}$ instead of (7). (This choice of instrumental variable is used in the Past Input (PI) variant of MOESP for identification of output-error models; see [8, §9.5], [7, page 351].) The missing outputs also limit the choices of the weight matrices. For example, the weight W_1 of IVM requires complete outputs. With these modifications, we can solve the same regularized nuclear norm problem (14) with variables $\mathbf{y} = (y(0), \ldots, y(N+r-2))$ to estimate corrected values of the measured outputs and simultaneously estimate the missing outputs. As a useful variation, one can optimize over the missing output values only by solving

$$\begin{array}{ll} \text{minimize} & \|G(\mathbf{y})\|_* \\ \text{subject to} & y(k) = y_{\mathrm{meas}}(k), \quad k \in T. \end{array} \tag{15}$$

The purpose here is to simply complete the output sequence. This is often referred to as *imputation*.

Missing inputs and outputs. The formulations (14) and (15) are not easily extended to problems with missing inputs because the matrix G depends nonlinearly on the inputs. When there are missing or corrupted values in both the input and output sequences, we therefore solve a regularized nuclear norm optimization problem of the form

$$\text{minimize} \quad \|F(\mathbf{u},\mathbf{y})\|_* + \lambda \sum_{k\in T_\text{o}} \|y(k) - y_\text{meas}(k)\|_2^2 + \gamma \sum_{k\in T_\text{i}} \|u(k) - u_\text{meas}(k)\|_2^2 \tag{16}$$

where T_o and T_i are the time instances at which output measurements or input measurements are available. The optimization variables are the sequences $\mathbf{u} = (u(-s), \ldots, u(r+N-2))$ and $\mathbf{y} = (y(-s), \ldots, y(r+N-2))$, and F is the stacked input-output Hankel matrix on the left-hand side of (8), *i.e.*, after reordering the rows, the matrix

$$F(\mathbf{u},\mathbf{y}) = \begin{bmatrix} U_{-s,s+r,N} \\ Y_{-s,s+r,N} \end{bmatrix}. \tag{17}$$

Two variations are

$$\begin{array}{ll} \text{minimize} & \|F(\mathbf{u},\mathbf{y})\|_* + \lambda \sum_{k\in T_\text{o}} \|y(k) - y_\text{meas}(k)\|_2^2 \\ \text{subject to} & u(k) = u_\text{meas}(k), \quad k \in T_\text{i} \end{array} \tag{18}$$

which is useful if we only wish to complete an incomplete input sequence without modifying the available input values, and

$$\begin{array}{ll} \text{minimize} & \|F(\mathbf{u},\mathbf{y})\|_* \\ \text{subject to} & u(k) = u_\text{meas}(k), \quad k \in T_\text{i} \\ & y(k) = y_\text{meas}(k), \quad k \in T_\text{o} \end{array} \tag{19}$$

which amounts to completing the input and output sequences, without modifying the available values. After solving the optimization, the range of the extended observability matrix can be estimated from the stacked input-output matrix constructed from the optimized $\mathbf{u}$ and $\mathbf{y}$, for example, via the LQ factorization (8). Notice that the left hand side of (8) can be obtained by reordering the rows of (17).

Minimizing the nuclear norm of the stacked Hankel matrix (17) is closely related to the second algorithm in [19], in which an SVD of the stacked matrix is used to estimate the range space of O_r. The algorithm is motivated by the fact that when the persistent excitation and full input rank assumptions (5) hold and the data are exact ($E = 0$ in (3)), then

$$\mathbf{rank} \begin{bmatrix} U_{0,r,N} \\ Y_{0,r,N} \end{bmatrix} = n_x + \mathbf{rank}\, U_{0,r,N}.$$

The input Hankel matrix is typically full rank, so the rank of the stacked Hankel matrix equals the true model order plus a constant.

Table 1: ADMM algorithm

1. Initialize x, X, Z, ρ. For example, set $x = 0$, $X = A_0$, $Z = 0$, $\rho = 1$.
2. Update $x := \mathrm{argmin}_{\hat{x}} L_\rho(\hat{x}, X, Z)$. See (21).
3. Update $X := \mathrm{argmin}_{\hat{X}} L_\rho(x, \hat{X}, Z)$. See (23).
4. Update $Z := Z + \rho(\mathcal{A}(x) + A_0 - X)$.
5. Terminate if $\|r_\mathrm{p}\|_F \leq \epsilon_\mathrm{p}$ and $\|r_\mathrm{d}\|_2 \leq \epsilon_\mathrm{d}$ (see (24)–(27)). Otherwise, go to step 2.

4. ADMM algorithm

The alternating direction method of multipliers (ADMM) is a popular method for large-scale and distributed convex optimization [26]. Its effectiveness for nuclear norm optimization, including nuclear norm problems arising in system identification, has been demonstrated in [12], along with several other first-order methods. In this section we give an outline of the ADMM implementation that was used for the experiments in section 5. We also describe two improvements that exploit specific structure in the system identification applications.

To simplify notation we state the algorithm for a generic nuclear norm optimization problem with a quadratic regularization term:

$$\text{minimize} \quad \|\mathcal{A}(x) + A_0\|_* + \frac{1}{2}(x-a)^T H(x-a). \tag{20}$$

The variable is a vector $x \in \mathbf{R}^n$. The first term in the objective is the nuclear norm of a $p \times q$ matrix $\mathcal{A}(x) + A_0$ where $\mathcal{A} : \mathbf{R}^n \to \mathbf{R}^{p\times q}$ is a linear mapping. The parameters in the second, quadratic, term in the objective of (20) are a vector $a \in \mathbf{R}^n$ and a positive semidefinite matrix $H \in \mathbf{S}^n$ (the set of symmetric matrices of order n).

To derive the ADMM iteration we first write (20) as

$$\begin{array}{ll} \text{minimize} & \|X\|_* + (1/2)(x-a)^T H(x-a) \\ \text{subject to} & \mathcal{A}(x) + A_0 = X \end{array}$$

with two variables $x \in \mathbf{R}^n$ and $X \in \mathbf{R}^{p\times q}$. The *augmented Lagrangian* for this problem is

$$L_\rho(x, X, Z) = \|X\|_* + \frac{1}{2}(x-a)^T H(x-a) + \mathbf{Tr}(Z^T(\mathcal{A}(x) + A_0 - X)) + \frac{\rho}{2}\|\mathcal{A}(x) + A_0 - X\|_F^2,$$

where ρ is a positive penalty parameter. Each iteration of the ADMM consists of a minimization of L_ρ over x, a minimization of L_ρ over X, and a simple update of the dual variable Z. This is summarized in table 1.

The update in step 2 requires the solution of a linear equation, since $L_\rho(\hat{x}, X, Z)$ is quadratic in $\hat{x}$. Setting the gradient of $L_\rho(\hat{x}, X, Z)$ with respect to $\hat{x}$ equal to zero gives the equation

$$(H + \rho M)\hat{x} = \mathcal{A}_\mathrm{adj}(\rho X + \rho A_0 - Z) + Ha \tag{21}$$

where $\mathcal{A}_{\mathrm{adj}}$ is the adjoint of the mapping $\mathcal{A}$ and M is the positive semidefinite matrix defined by the identity

$$Mz = \mathcal{A}_{\mathrm{adj}}(\mathcal{A}(z)) \quad \forall z. \tag{22}$$

Step 2 of the algorithm is discussed in more detail in the following two sections.

The minimizer X in step 3 can be expressed as

$$\begin{aligned} X &= \operatorname*{argmin}_{\hat{X}} \left(\|\hat{X}\|_* + \frac{\rho}{2}\|\hat{X} - \mathcal{A}(x) - A_0 - (1/\rho)Z\|_F^2 \right) \\ &= \sum_{i=1}^{\min\{p,q\}} \max\{0, \sigma_i - \frac{1}{\rho}\}\ u_i v_i^T \end{aligned} \tag{23}$$

where u_i, v_i, σ_i are given by a singular value decomposition

$$\mathcal{A}(x) + A_0 + \frac{1}{\rho}Z = \sum_{i=1}^{\min\{p,q\}} \sigma_i u_i v_i^T$$

(see [27, theorem 2]). This operation is called 'singular value soft-thresholding'.

The residuals and tolerances in the stopping criterion in step 5 are defined as follows [26]:

$$r_{\mathrm{p}} = \mathcal{A}(x) + A_0 - X \tag{24}$$

$$r_{\mathrm{d}} = \rho\mathcal{A}_{\mathrm{adj}}(X_{\mathrm{prev}} - X) \tag{25}$$

$$\epsilon_{\mathrm{p}} = \sqrt{pq}\,\epsilon_{\mathrm{abs}} + \epsilon_{\mathrm{rel}} \max\{\|\mathcal{A}(x)\|_F, \|X\|_F, \|\mathcal{A}_0\|_F\} \tag{26}$$

$$\epsilon_{\mathrm{d}} = \sqrt{n}\epsilon_{\mathrm{abs}} + \epsilon_{\mathrm{rel}}\, \|\mathcal{A}_{\mathrm{adj}}(Z)\|_2, \tag{27}$$

Typical values for the relative and absolute tolerances are $\epsilon_{\mathrm{rel}} = 10^{-3}$ and $\epsilon_{\mathrm{abs}} = 10^{-6}$. The matrix X_{prev} in (25) is the value of X in the previous iteration.

Instead of using a fixed penalty parameter ρ, one can vary ρ to improve the speed of convergence. An example of such a scheme is to adapt ρ at the end of each ADMM iteration as follows [28]

$$\rho := \begin{cases} \tau\rho & \text{if } \|r_{\mathrm{p}}\|_F > \mu\|r_{\mathrm{d}}\|_2 \\ \rho/\tau & \text{if } \|r_{\mathrm{d}}\|_2 > \mu\|r_{\mathrm{p}}\|_F \\ \rho & \text{otherwise.} \end{cases}$$

This scheme depends on parameters $\mu > 1$, $\tau > 1$ (typical values are $\mu = 10$ and $\tau = 2$). Note that varying ρ has an important consequence on the algorithm in table 1. If ρ is fixed, the coefficient matrix $H + \rho M$ in the equation (21) that is solved in step 2 of each iteration is constant throughout the algorithm. Therefore only one costly factorization of $H + \rho M$ is required. If we change ρ after step 5, a new factorization of $H + \rho M$ is needed before returning to step 2. We explain in §4.2 how the extra cost of repeated factorizations can be avoided.

4.1. Hankel structure

An important improvement in the algorithm efficiency can be achieved by exploiting the Hankel structure in the subspace system identification applications. The mapping $\mathcal{A}$ in these applications can be expressed as

$$\mathcal{A}(x) = L\mathcal{H}(x)R$$

where $x = (h_1, h_2, \ldots, h_{r+N-1})$ with $h_i \in \mathbf{R}^u$, and $\mathcal{H}(x)$ is a block Hankel matrix

$$\mathcal{H}(x) = \left[\begin{array}{cccc} h_1 & h_2 & \cdots & h_N \\ h_2 & h_3 & \cdots & h_{N+1} \\ \vdots & \vdots & \ddots & \vdots \\ h_r & h_{r+1} & \cdots & h_{r+N-1} \end{array}\right]$$

with r block rows and N columns. The matrices L and R are general dense matrices. For example, the matrix $G(\mathbf{y})$ in the nuclear norm identification problem (14) can be written in this form with $L = W_1$, $R = \Pi_{0,r,N}\Phi^T W_2$, and $x = (y(0), \ldots, y(r+N-2))$.

The adjoint of the mapping $\mathcal{A}$ is $\mathcal{A}_{\mathrm{adj}}(Y) = \mathcal{H}_{\mathrm{adj}}(L^T Y R^T)$. The adjoint $\mathcal{H}_{\mathrm{adj}}$ of the Hankel mapping $\mathcal{H}$ maps an $ru \times N$ matrix to a sequence of $r+N-1$ vectors of size u by summing the block entries in the matrix along the anti-diagonals: if X is an $r \times N$ block matrix with blocks $x_{ij} \in \mathbf{R}^u$, then $\mathcal{H}_{\mathrm{adj}}(X) = (y_1, y_2, \ldots, y_{r+N-1})$ with $y_k = \sum_{i+j=k+1} x_{ij}$.

We now show how to exploit Hankel structure when constructing the matrix M in (22). We first consider the scalar case ($h_i \in \mathbf{R}$). The Hankel mapping can be expressed in the following form

$$\mathcal{H}(x) = \frac{1}{K} E^H \,\mathbf{diag}(Fx) G \tag{28}$$

with

$$E = \tilde{T}_{1:r}, \qquad F = \tilde{T}, \qquad G = T_{1:N},$$

where the columns of T are the first $r+N-1$ columns of the discrete Fourier transform (DFT) matrix of order $K \geq 2r + 2N - 3$ and $\tilde{T}$ is the matrix T with its columns in reverse order. The notation $T_{1:N}$ means the first N columns of T. Similarly, $\tilde{T}_{1:r}$ denotes the first r columns of $\tilde{T}$. The representation (28) of Hankel matrices is a permutation of the representation of Toeplitz matrices used in [29] and is closely related to techniques for exploiting Toeplitz structure in linear equations [30, pp 201-202]. The parametrization (28) can be extended to a block Hankel matrix ($h_i \in \mathbf{R}^u$) by defining

$$E = \tilde{T}_{1:r} \otimes I_u, \qquad F = \tilde{T} \otimes I_u, \qquad G = T_{1:N} \otimes \mathbf{1}_u,$$

where $\otimes$ denotes the Kronecker product, I_u is the identity matrix of size u, and $\mathbf{1}_u$ is a u-vector of ones [31].

Using (28), the adjoint of $\mathcal{H}_{\mathrm{adj}}$ can be written as

$$\mathcal{H}_{\mathrm{adj}}(X) = \frac{1}{K} F^H \,\mathbf{diag}(EXG^H).$$

Table 2: Computation time in seconds for constructing the Gram matrix M

N	u	Standard	DFT	FFT
250	1	2.2	0.40	0.31
500	1	7.7	1.9	0.87
1000	1	29	11	2.4
2000	1	116	75	9.4
4000	1	448	533	37
100	3	36	0.66	0.73
250	3	164	4.8	1.9
500	3	545	28	5.8
1000	3	-	185	19
2000	3	-	1401	110
100	5	240	2.4	1.4
250	5	894	19	5.4
500	5	-	112	18
1000	5	-	774	64

Therefore,

$$\begin{aligned} \mathcal{A}_{\mathrm{adj}}(\mathcal{A}(z)) &= \mathcal{H}_{\mathrm{adj}}(L^T L \mathcal{H}(z) R R^T) \\ &= \frac{1}{K^2} F^H \,\mathbf{diag}(E L^T L E^H \,\mathbf{diag}(Fz) G R R^T G^H) \\ &= \frac{1}{K^2} F^H \left((E L^T L E^H) \circ \overline{G R R^T G^H} \right) Fz, \end{aligned}$$

where $\circ$ denotes the Hadamard product. This shows that

$$M = \frac{1}{K^2} F^H \left((E L^T L E^H) \circ \overline{G R R^T G^H} \right) F. \tag{29}$$

The construction of M can be further expedited by using the fast Fourier transform algorithm for the matrix products with E, F, and G.

To give an idea of the value of this technique, we show in table 2 the time needed to construct M using three different methods. The matrices L and R in the example are randomly generated dense matrices of size $ru \times ru$ and $N \times 2ru$. The Hankel matrix $\mathcal{H}(x)$ has size $ru \times N$. In the experiment we fix $r = 30$ and vary N and u. The CPU times are expressed in seconds for 2.3 GHz quad-core laptop with 8 GB of memory using MATLAB 7.10 (R2010a). All times are averaged over five randomly generated examples. Blank entries in the table indicate instances that were not completed due to excessive execution time or an out-of-memory error.

Three methods for constructing M are compared. The method in the first column (labeled 'Standard') is based on first expressing $\mathcal{A}$ as $\mathcal{A}(x) = x_1 A_1 + \cdots + x_n A_n$, and then computing

$M_{ij} = \mathbf{Tr}(A_i A_j)$ without exploiting structure in the coefficients. The methods in two other columns are based on the algorithm described in this section and evaluation of M via (29). The method labeled as 'DFT' generates the matrices E, G, and F explicitly and computes the matrix-matrix products. The method in column 'FFT' uses MATLAB's fast Fourier transform routine `fft`.

4.2. Simultaneous diagonalization

The factorization of the matrix $H + \rho M$ needed for equation (21) is an expensive step in the ADMM algorithm of table 1. If ρ is fixed, only one factorization at the beginning of the algorithm is needed. However, as mentioned, the convergence can be improved by occasionally adapting the parameter ρ. Moreover in applications one is often interested in the trade-off between the two terms in the cost function of (20). Tracing the trade-off curve requires solving the problem multiple times with different scalar multiples of H. In this section we describe a preprocessing that allows one to solve multiple equations of the form

$$(\gamma H + \rho M)x = b, \tag{30}$$

with different coefficients γ, ρ, and to compute the entire regularization trade-off curve with a single factorization at the start of the algorithm. We assume that the matrix $H + M$ is positive definite. The preprocessing is based on a simultaneous diagonalization of H and M [30, §8.7.2]. We first compute the Cholesky factorization

$$H + M = \tilde{L}\tilde{L}^T$$

and a symmetric eigenvalue decomposition

$$\tilde{L}^{-1}H\tilde{L}^{-T} = QDQ^T,$$

where Q is an orthogonal matrix and D is a diagonal matrix. It can be verified that

$$Q^T\tilde{L}^{-1}H\tilde{L}^{-T}Q = D, \qquad Q^T\tilde{L}^{-1}M\tilde{L}^{-T}Q = I - D$$

and therefore the solution x can be obtained by

$$x = \tilde{L}^{-T}Q\left((\gamma - \rho)D + \rho I\right)^{-1}Q^T\tilde{L}^{-1}b.$$

After the initial factorization, the cost of solving the equation is only quadratic in the number of variables.

5. Identification experiments

In this section we evaluate the nuclear norm heuristic in combination with the subspace identification algorithms. We experiment with two scenarios from section 3. The first scenario is the complete-data case, where the measured inputs and outputs are noisy but completely available. In the second scenario a percentage of inputs and outputs is removed.

All the simulations are run on a 2.3 GHz quad-core laptop with 8 GB of memory, using MATLAB 7.10 (R2010a). The code `n4sid` in the MATLAB System Identification toolbox is used to compute baseline solutions that are compared with the results of the nuclear norm methods. The `n4sid` code implements both the MOESP and CVA subspace methods. If the user does not specify a weighting to use, an automatic choice is made. We use the following settings for `n4sid`. The model order is determined automatically in the code by setting `order = 'best'`. With this choice the model order equals the number of singular values of G in (9) above the average value of the smallest and largest singular values on a logarithmic scale. We also specify two additional settings: `nk = zeros(1,m)` and `focus = stability`. The first setting requires the code to estimate the state-space matrix D, and the second setting forces stability of the identified model.

The criterion used to compare the quality of different models is the validation fit measure. The fit measure is computed by the code `compare` in MATLAB's System Identification toolbox. It is defined in percentage as

$$\text{fit} = 100\left(1 - \frac{\|y_{\text{pred}} - y\|}{\|y - \text{mean}(y)\|}\right)$$

for a single output sequence, where y is the validation data output sequence and y_{pred} is the predicted output from the model. For systems with multiple outputs, we report the average of the fit, averaged over the outputs. We always use different data for identification and validation.

The nuclear norm optimization problems are solved using the ADMM algorithm described in section 4. The maximum number of iterations is set to 200. The absolute and relative solution accuracy tolerance are set to respectively $\epsilon_{\text{abs}} = 10^{-6}$ and $\epsilon_{\text{rel}} = 10^{-3}$. The parameters for updating the penalty ρ are set to $\mu = 10$ and $\tau = 2$. The ADMM algorithm typically takes less than 50 iterations (both for regularized and non-regularized nuclear norm optimization). The only time the maximum number of iterations is reached is when a very small regularization parameter λ (*e.g.*, 10^{-3}) is used in the regularized nuclear norm optimization. This is not a concern since solutions with very small λ do not provide good system identification performance and were only included to get the entire regularization trade-off curve.

5.1. Complete inputs and outputs

In the first set of experiments we solve the regularized nuclear norm approximation problem (14), to compute a modified output sequence. The IVM weight matrices W_1 and W_2 are used. In all experiments we set the dimensions r and s to 15. A detailed comparison of the different weightings in nuclear norm based subspace system identification is presented in [13].

We use $G(\mathbf{y})$, with $\mathbf{y}$ the computed output sequence, as G in (9) to compute an estimate of the range of the extended observability matrix and then obtain a system realization via the realization algorithm described in section 2.2. The model order is determined in the same way as in MATLAB's `n4sid` routine, *i.e.*, the number of singular values above the

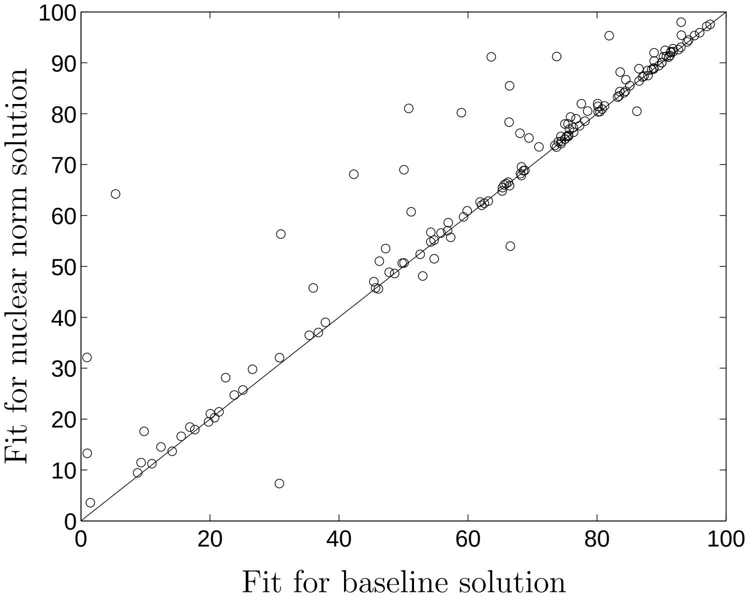

Figure 1: Scatter plot showing the fit for the nuclear norm identification method versus the baseline method for 156 randomly generated data sets.

average value of the smallest and largest singular value on a logarithmic scale. Another possible approach for the system realization is to present the optimized output sequence to `n4sid`. We refer the reader to [13] for numerical results with this method, which provides comparable results to the ones we report here.

We determine λ in (14) by computing solutions for 20 logarithmically-spaced values of λ in the interval 10^{-3} to 10^{3}, and selecting the value that gives the best fit on validation data.

Randomly generated models. We first present results with randomly generated systems. The identification and validation data were generated using state-space models (A, B, C, D) from the MATLAB function `drss`. The Kalman gain K was generated using `randn`. Only single-input-single-output systems were considered. The state dimension n_x varies from 4 to 20 in unit steps. The data length for identification was 300 and the length of the validation data was 1500. The noise $e(k)$ was generated with `randn`, *i.e.*, a white standardized normal distribution with zero mean and unit covariance. The input sequence was generated in the same way as $e(k)$, but scaled with a factor σ that varied from 2 to 10 in unit steps. In this way we obtained examples with a wide range of signal-to-noise ratios. For each combination of the 17 values of n_x and 9 values of σ we generate one instance , *i.e.*, the total number of examples was $17 \times 9 = 156$. It takes about 6 seconds to compute the solution for one example. This time includes the time for 20 runs of the optimization, 20 runs of state-space model determination, and 20 calls to the function `compare`.

Figure 1 shows a scatter plot of fits for the nuclear norm based solution (using IVM weights) versus the baseline solution. We note that the two approaches mostly give about the same fit, but in more than 10% of the cases the nuclear norm method results in a

Table 3: Ten benchmark problems from the Daisy collection [14]. N_{I} is the number of data points used for identification. N_{V} is the number of points used for validation.

	Data set	Description	Inputs	Outputs	N_{I}	N_{V}
1	96-007	CD player arm	2	2	500	1500
2	98-002	Continuous stirring tank reactor	1	2	500	1500
3	96-006	Hair dryer	1	1	300	700
4	97-002	Steam heat exchanger	1	1	1000	3000
5	99-001	SISO heating system	1	1	300	500
6	96-009	Flexible robot arm	1	1	300	700
7	96-011	Heat flow density	2	1	500	1000
8	97-003	Industrial winding process	5	2	500	1500
9	96-002	Glass furnace	3	6	250	750
10	96-016	Industrial dryer	3	3	300	500

significantly better fit.

Examples from the DaISy collection. The second set of results are ten benchmark examples from the DaISy collection [14]. Table 3 provides a brief description of the data sets. Since there is only one input-output sequence for each system, we break up the data sequences in two sections. The first N_{I} data points are used in the model identification, and the next N_{V} data points are used for validation.

Table 4 summarizes the performance measure (validation fit). We note that the nuclear norm solutions are significantly better than the baseline `n4sid` solution in examples 1, 4, and 10. For the other data sets the two solutions are comparable. The times reported in the last column are the total time (in seconds) for computing the nuclear norm solution. This includes the cost of solving (14) for 20 different values of the regularization parameter λ.

5.2. Missing inputs and outputs

In this set of experiments we evaluate the nuclear norm approach for problems with missing inputs and outputs. We reuse the ten benchmark examples from the DaISy database, but remove a percentage of randomly chosen inputs and outputs from the identification sequence.

We solve the regularized nuclear norm optimization problem (18). In each experiment we use $r = s = 30$ if the system is single-output and $r = s = 15$ otherwise. From the optimal input and output sequences $\mathbf{u}$ and $\mathbf{y}$ we reorder the rows of (17) to obtain the left hand side of (8), from which we obtain an estimate of range(O_r) via (11) with weight matrices $W_1 = W_2 = I$. We then compute a system realization by the algorithm described in section 2.2. Twenty optimization problems are solved with values of the regularization parameter λ logarithmically spaced in the interval 10^{-3} to 10^{3}. The model with the best validation fit is selected.

Table 4: The validation fit of subspace system identification via weighted nuclear norm optimization for ten benchmark problems from the DaISy collection. The baseline solution is the model computed by `n4sid`. The times in the last column are the total times for computing 20 solutions on the regularization path.

Data set	Baseline	Nuclear norm	Time (sec.)
1	68.4	73.3	40
2	79.5	80.1	44
3	84.1	86.3	9
4	70.8	76.6	20
5	82.5	84.7	5
6	96.6	96.0	12
7	83.6	86.7	8
8	58.9	57.7	25
9	55.3	59.4	168
10	40.8	48.2	50

Table 5 summarizes the results. (The dashed entries indicate a negative fit.) Except for data set 4, the regularized nuclear norm optimization approach worked surprisingly well even with a high percentage of missing data. For most data sets the fit measure degrades very slowly with increasing percentages of missing data.

Note that a complete validation sequence was used to determine the regularization parameter λ. We also experimented with a non-regularized identification method for missing data based on solving problem (19), and found that the results were only slightly worse than the results in table 5.

6. Conclusions

In this paper we presented a subspace system identification method using the nuclear norm optimization. We investigated the benefit of instrumental variables in the nuclear norm approach to improve the handling of data with colored noise. Experimental results showed that the nuclear norm subspace identification performed only slightly better than the standard SVD-based subspace methods when no inputs and outputs are missing. The main benefit of the nuclear norm approach is its ability to handle data sets with a high percentage of missing inputs and outputs, as well as other identification problems with additional convex constraints on the inputs and outputs. We also presented techniques to improve the efficiency of the alternating direction method of multipliers for regularized and non-regularized nuclear norm optimization with Hankel structure.

Acknowledgment. This material is based upon work supported by the National Science Foundation under Grant No. 1128817.

Table 5: Validation fit for models obtained by a regularized nuclear norm optimization method, applied to ten DaISy problems with different percentages of missing inputs and outputs.

Data set	0%	10%	20%	30%	40%	50%
1	72.0	72.4	72.2	71.8	72.5	71.0
2	84.7	86.2	85.3	85.4	85.2	83.7
3	84.4	88.6	84.7	84.2	80.6	81.0
4	29.7	45.1	35.6	–	–	–
5	84.3	83.9	84.0	83.9	82.9	83.4
6	95.9	95.5	95.5	95.9	89.9	79.4
7	86.5	86.5	86.7	86.3	86.3	85.3
8	67.5	67.4	67.0	66.5	67.5	64.5
9	49.5	34.1	31.5	40.3	44.7	43.2
10	44.1	44.5	41.5	30.7	41.8	29.2

References

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

[2] M. Fazel, Matrix rank minimization with applications, Ph.D. thesis, Stanford University (2002).

[3] E. J. Candès, B. Recht, Exact matrix completion via convex optimization, Foundations of Computational Mathematics 9 (6) (2009) 717–772.

[4] B. Recht, M. Fazel, P. A. Parrilo, Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization, SIAM Review 52 (3) (2010) 471–501.

[5] E. J. Candès, Y. Plan, Matrix completion with noise, Proceedings of the IEEE 98 (6) (2010) 925–936.

[6] E. J. Candès, T. Tao, The power of convex relaxation: near-optimal matrix completion, IEEE Transactions on Information Theory 56 (5) (2010) 2053–2080.

[7] L. Ljung, System Identification, 2nd Edition, Prentice Hall, Upper Saddle River, New Jersey, USA, 1999.

[8] M. Verhaegen, V. Verdult, Filtering and System Identification, Cambridge University Press, New York, 2007.

[9] Z. Liu, L. Vandenberghe, Interior-point method for nuclear norm approximation with application to system identification, SIAM Journal on Matrix Analysis and Applications 31 (3) (2009) 1235–1256.

[10] Z. Liu, L. Vandenberghe, Semidefinite programming methods for system realization and identification, in: Proceedings of the Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference, 2009, pp. 4676–4681.

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

[12] M. Fazel, T. K. Pong, D. Sun, P. Tseng, Hankel matrix rank minimization with applications to system identification and realization. Submitted and revised, 2013.

[13] A. Hansson, Z. Liu, L. Vandenberghe, Subspace system identification via weighted nuclear norm optimization, in: Proceedings of the IEEE 51st Annual Conference on Decision and Control, 2012, pp. 3439–3444.

[14] B. De Moor, P. De Gersem, B. De Schutter, W. Favoreel, DAISY: A database for the identification of systems, Journal A 38 (3) (1997) 4–5.

[15] T. Ding, M. Sznaier, O. Camps, A rank minimization approach to fast dynamic event detection and track matching in video sequences, in: Proceedings of the 46th IEEE Conference on Decision and Control, 2007, pp. 4122–4127.

[16] C. Grossmann, C. N. Jones, M. Morari, System identification via nuclear norm regularization for simulated bed processes from incomplete data sets, in: Proceedings of the 48th IEEE Conference on Decision and Control, 2009, pp. 4692–4697.

[17] I. Markovsky, Data modeling using the nuclear norm heuristic, Technical Report 21936, ECS, University of Southampton, Southampton (2011).

[18] M. Moonen, B. De Moor, L. Vandenberghe, J. Vandewalle, On- and off-line identification of linear state-space models, International Journal of Control 49 (1989) 219–232.

[19] B. De Moor, M. Moonen, L. Vandenberghe, J. Vandewalle, A geometrical approach for the identification of state space models with the singular value decomposition, in: Proceedings of the 1988 IEEE International Conference on Acoustics, Speech, and Signal Processing, 1988, pp. 2244–2247.

[20] M. Verhaegen, P. Dewilde, Subspace model identification. Part 1: The output-error state-space model identification class of algorithms, International Journal of Control 56 (5) (1992) 1187–1210.

[21] M. Verhaegen, Subspace model identification. Part 3: Analysis of the ordinary output-error state space model identification algorithm, International Journal of Control 58 (3) (1993) 555–586.

[22] M. Viberg, B. Wahlberg, B. Ottersten, Analysis of state space system identification methods based on instrumental variables and subspace fitting, Automatica 33 (9) (1997) 1603–1616.

[23] M. Verhaegen, Identification of the deterministic part of MIMO state space models given in innovations form from input-output data, Automatica 30 (1) (1994) 61–74.

[24] M. Viberg, Subspace-based methods for the identification of linear time-invariant systems, Automatica 31 (12) (1995) 1835–1851.

[25] P. M. O. Gebraad, J. W. van Wingerden, G. J. van der Veen, M. Verhaegen, LPV subspace identification using a novel nuclear norm regularization method, in: Proceedings of the American Control Conference, 2011, pp. 165–170.

[26] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Foundations and Trends in Machine Learning 3 (1) (2011) 1–122, Michael Jordan, Editor in Chief.

[27] J.-F. Cai, E. J. Candès, Z. Shen, A singular value thresholding algorithm for matrix completion, Siam Journal of Optimization 20 (4) (2010) 1956–1982.

[28] B. S. He, H. Yang, S. L. Wang, Alternating direction method with self adaptive penalty parameters for monotone variational inequalities, Journal of Optimization Theory and Applications 106 (2) (2000) 337–356.

[29] T. Roh, L. Vandenberghe, Discrete transforms, semidefinite programming, and sum-of-squares representations of nonnegative polynomials, SIAM Journal on Optimization 16 (4) (2006) 939–964.

[30] G. H. Golub, C. F. V. Loan, Matrix Computations, 3rd Edition, John Hopkins University Press, 1996.

[31] T. J. Roh, Low-rank structure in semidefinite programming and sum-of-squares optimization in signal processing, Ph.D. thesis, University of California, Los Angeles (2007).