

Subspace Identification With Guaranteed Stability Using Constrained Optimization

Seth L. Lacy and Dennis S. Bernstein

Abstract—In system identification, the true system is often known to be stable. However, due to finite sample constraints, modeling errors, plant disturbances and measurement noise, the identified model may be unstable. We present a constrained optimization method to ensure asymptotic stability of the identified model in the context of subspace identification methods. In subspace identification, we first obtain an estimate of the state sequence or extended observability matrix and then solve a least squares optimization problem to estimate the system parameters. To ensure asymptotic stability of the identified model, we write the least-squares optimization problem as a convex linear programming problem with mixed equality, quadratic, and positive-semidefinite constraints suitable for existing convex optimization codes such as SeDuMi. We present examples to illustrate the method and compare to existing approaches.

Index Terms—System identification, stability, subspace identification, convex optimization, linear systems.

I. INTRODUCTION

Although linear system identification is widely viewed as a mature subject [2], [8], [12], [13], [18], recent advances in the development of subspace methods have significantly enhanced the available tools for these problems [3], [5]–[7], [10], [11], [15]–[22]. Subspace methods differ from classical least squares methods in that they provide estimates of a state sequence or an extended observability matrix, which is subsequently used to estimate the system parameters. These methods are multivariable, they allow the use of arbitrary identification signals, and they provide estimates of the plant disturbance and sensor noise statistics. In addition, subspace methods are based on the computationally tractable singular value decomposition and least squares procedures.

In practice, system identification is practical only when the system to be identified is asymptotically stable or, in worst case, linearly unstable as in a system possessing a rigid body mode. Even when the system is known to be asymptotically stable, finite sampling constraints, modeling errors, and plant and measurement noise often give rise to unstable models. Within subspace methods, techniques have been developed to enforce the stability of the identified model. One approach to the problem of identifying stable realizations is to compute a (possibly) unstable realization, and then reflect the unstable eigenvalues into the unit disk leaving the stable eigenvalues unchanged. Our approach does not guarantee that the stable eigenvalues will remain invariant, nor is it clear that this is desirable. The method developed in [9] enforces stability by replacing the last p rows of the extended observability matrix by zeros, where p is the number of outputs. Since this technique distorts the estimated observability matrix, alternative approaches involving data augmentation were developed in [1]. In [15], a regularization term is added to the least squares cost function minimization problem, where the amount of regularization is computed through a

generalized eigenvalue problem. The spectral radius of the minimizing solution has an upper bound given in terms of singular values, which can be chosen by the user. In [4] the maximum singular value of the dynamics matrix is constrained. In [10] a weighted cost function is used and minimized for a particular choice of weighting matrices. All of the aforementioned methods are inherently conservative. Our method is conservative in that the Lyapunov inequalities $P - APA^T > 0_n$ and $P > 0_n$ are over-constrained as $P - APA^T \geq \delta I_n$ and $P \geq \delta I_n$, where $\delta > 0$ can be chosen arbitrarily small.

The objective of this note is to develop an alternative approach to identifying stable models within the context of subspace identification. Our approach is based on constrained optimization, where the least squares optimization procedure used to obtain parameter estimates is replaced by a constrained weighted least squares optimization problem involving a stability constraint. The resulting convex optimization problem is a linear programming problem over symmetric cones. This approach is direct, and thus it does not distort the extended observability matrix with zeros as in [9], modify the estimated state sequence, extended observability matrix, or input sequence with fictitious data as in [1], or augment the cost function to indirectly implement a singular value constraint as in [15]. The cost function is a weighted Frobenius matrix norm, different from the unweighted least squares cost function. This technique can be applied to an estimate of the extended observability matrix and to an estimate of the state sequence.

To solve this convex optimization problem, we adapt the SeDuMi MATLAB code [14] to the constrained least squares problem. This code solves linear programming problems with convex symmetric cone constraints, that is, constraints involving mixed equality, inequality, quadratic, and positive-semidefinite constraints. We reformulate the problem of identifying a stable model given the state sequence in Section II. In Section III, we reformulate the problem of identifying a stable model given the extended observability matrix. In Section IV, we provide several numerical examples and compare the results to previous techniques.

II. STATE SEQUENCE PROBLEM DESCRIPTION

The identification problem is to estimate the coefficient matrices of the system

$$x(k+1) = Ax(k) + Bu(k) \quad (2.1)$$

$$y(k) = Cx(k) + Du(k) \quad (2.2)$$

given measurements of $u(k : k + \ell - 1)$, $y(k : k + \ell - 1)$, and an estimate of the state sequence $x(k : k + \ell)$ obtained from a subspace identification algorithm, where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$, and $D \in \mathbb{R}^{p \times m}$. Our goal is to obtain an estimate of the dynamics matrix A that is asymptotically stable.

The least squares problem is to minimize

$$\begin{aligned} J_x^2(A, B, C, D) &\triangleq \left\| \begin{bmatrix} L_x & 0 \\ 0 & L_y \end{bmatrix} \begin{bmatrix} x(k+1 : k+\ell) \\ y(k : k+\ell-1) \end{bmatrix} \right. \\ &\quad \left. - \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x(k : k+\ell-1) \\ u(k : k+\ell-1) \end{bmatrix} \right\|_F^2 \\ &= J_1^2(A, B) + J_2^2(C, D) \end{aligned} \quad (2.3)$$

where $u(k : k + \ell - 1) \in \mathbb{R}^{m \times \ell}$, $x(k : k + \ell - 1) \in \mathbb{R}^{n \times \ell}$, $x(k+1 : k + \ell) \in \mathbb{R}^{n \times \ell}$, $y(k : k + \ell - 1) \in \mathbb{R}^{p \times \ell}$, $L_x \in \mathbb{R}^{n \times n}$, $L_y \in \mathbb{R}^{p \times p}$, and $R_x \in \mathbb{R}^{\ell \times \ell}$ are weighting matrices and

$$\begin{aligned} J_1(A, B) &\triangleq \left\| L_x (x(k+1 : k+\ell) - [A B] \right. \\ &\quad \left. \times \begin{bmatrix} x(k : k+\ell-1) \\ u(k : k+\ell-1) \end{bmatrix}) \right\|_F \end{aligned} \quad (2.4)$$

Manuscript received July 20, 2002; revised October 19, 2002, February 25, 2003, and March 12, 2003. Recommended by Associate Editor A. Garulli. This work was supported in part by the National Aeronautics and Space Administration under Grant GSRP NGT4-52421, by the Air Force Office of Scientific Research under Grant F49620-01-1-0094, and Laboratory Research Initiative 00VS17COR.

S. L. Lacy is with the Space Vehicles Directorate, Air Force Research Laboratory, Kirtland AFB, NM 87117 USA (e-mail: seth.lacy@kirtland.af.mil).

D. S. Bernstein is with the Aerospace Engineering Department, University of Michigan, Ann Arbor, MI 48109 USA (dsbaero@umich.edu).

Digital Object Identifier 10.1109/TAC.2003.814273

$$J_2(C, D) \triangleq \|L_y(y(k : k + \ell - 1) - [C \ D] \times \begin{bmatrix} x(k : k + \ell - 1) \\ u(k : k + \ell - 1) \end{bmatrix}) R_x\|_{\mathbb{F}}. \quad (2.5)$$

The notation $u(k : k + \ell - 1)$ denotes the matrix $[u(k), \dots, u(k + \ell - 1)]$. The minimizers of (2.4) and (2.5) and, thus, (2.3) are given by

$$[\hat{A} \ \hat{B}] = x(k + 1 : k + \ell) \times R_x \left(\begin{bmatrix} x(k : k + \ell - 1) \\ u(k : k + \ell - 1) \end{bmatrix} R_x \right)^{\mathbb{R}} \quad (2.6)$$

$$[\hat{C} \ \hat{D}] = y(k : k + \ell - 1) \times R_x \left(\begin{bmatrix} x(k : k + \ell - 1) \\ u(k : k + \ell - 1) \end{bmatrix} R_x \right)^{\mathbb{R}} \quad (2.7)$$

where $V^{\mathbb{R}} \triangleq V^{\mathbb{T}}(VV^{\mathbb{T}})^+$, $V^{\mathbb{L}} \triangleq (V^{\mathbb{T}}V)^+V^{\mathbb{T}}$, and M^+ denotes the Moore–Penrose generalized inverse of M .

The solutions \hat{A} and \hat{B} are independent of the solutions \hat{C} and \hat{D} . Thus, we can use the least-squares solution (2.7) to find \hat{C} and \hat{D} and focus on minimizing J_1 subject to a stability constraint in place of (2.6). We would like to solve this minimization problem subject to the stability constraint that A is asymptotically stable. The matrix A is asymptotically stable if $\rho(A) < 1$, where $\rho(A)$ denotes the spectral radius of A . Alternatively, we could implement the constraint that A is Lyapunov stable, i.e., that $\rho(A) \leq 1$ and all of the eigenvalues of A on the unit circle are semisimple. We detail the asymptotically stable constraint approach below.

The asymptotic stability of A is equivalent to the Lyapunov inequalities

$$P - APA^{\mathbb{T}} > 0_n, \quad P > 0_n. \quad (2.8)$$

The constraints (2.8) constrain P and A to open sets, while we need closed constraint sets to guarantee that a solution to the convex optimization problem exists. Hence, we enforce (2.8) with the inequalities

$$P - APA^{\mathbb{T}} \geq \delta I_n, \quad P \geq \delta I_n \quad (2.9)$$

where $\delta > 0$. Using Schur complements, the constraints (2.9) are equivalent to

$$S \triangleq \begin{bmatrix} P - \delta I_n & AP \\ PA^{\mathbb{T}} & P \end{bmatrix} \geq 0_{2n}. \quad (2.10)$$

The constraint (2.10) constrains S to a closed convex set.

The problem restated with the linear matrix inequality constraint (2.10) is: $\min J_1(A, B)$ subject to (2.10). We let

$$R_x = \begin{bmatrix} x(k : k + \ell - 1) \\ u(k : k + \ell - 1) \end{bmatrix}^{\mathbb{R}} \begin{bmatrix} PR_{x_1} & 0_{n \times r_2} \\ 0_{m \times r_1} & R_{x_2} \end{bmatrix} \quad (2.11)$$

where $R_{x_1} \in \mathbb{R}^{n \times r_1}$ and $R_{x_2} \in \mathbb{R}^{m \times r_2}$. This weighting matrix will allow us to use a change of variables leading to a convex optimization problem. The cost function is then

$$\begin{aligned} J_1^2(A, B) &= \left\| L_x \left(x(k + 1 : k + \ell) \begin{bmatrix} x(k : k + \ell - 1) \\ u(k : k + \ell - 1) \end{bmatrix}^{\mathbb{R}} \right. \right. \\ &\quad \left. \left. \times \begin{bmatrix} PR_{x_1} & 0_{n \times r_2} \\ 0_{m \times r_1} & R_{x_2} \end{bmatrix} - [APR_{x_1} \ B R_{x_2}] \right) \right\|_{\mathbb{F}}^2 \\ &= J_{11}^2(A) + J_{12}^2(B) \end{aligned} \quad (2.12)$$

where

$$J_{11}^2(A) \triangleq \|L_x(X_1 - A)PR_{x_1}\|_{\mathbb{F}}^2 \quad (2.13)$$

$$J_{12}^2(B) \triangleq \|L_x(X_2 - B)R_{x_2}\|_{\mathbb{F}}^2 \quad (2.14)$$

$$[X_1 \ X_2] \triangleq x(k + 1 : k + \ell) \begin{bmatrix} x(k : k + \ell - 1) \\ u(k : k + \ell - 1) \end{bmatrix}^{\mathbb{R}} \quad (2.15)$$

$X_1 \in \mathbb{R}^{n \times n}$, and $X_2 \in \mathbb{R}^{n \times m}$. Hence, the value \hat{B} of B minimizing (2.14) is $\hat{B} = X_2$ and we can now focus on minimizing J_{11} subject to the stability constraint (2.10).

Defining the change of variables

$$Q \triangleq AP \quad (2.16)$$

the cost function (2.13) and the constraint (2.10) become

$$J_{11}^2(A) = \|L_x(X_1P - Q)R_{x_1}\|_{\mathbb{F}}^2 \quad (2.17)$$

$$\begin{bmatrix} P - \delta I_n & Q \\ Q^{\mathbb{T}} & P \end{bmatrix} \geq 0_{2n}. \quad (2.18)$$

We compute \hat{A} from \hat{Q} and \hat{P} as $\hat{A} \triangleq \hat{Q}\hat{P}^{-1}$.

Our optimization problem is a quadratic programming problem with positive–semidefinite constraints. We rewrite this as the linear programming problem: $\min_{a, Q, P} a$, subject to $a \geq J_{11}(A)$ and (2.18). We can state this problem in a form suitable for use in a convex optimization routine as

$$\min c_x^{\mathbb{T}} z \quad (2.19)$$

subject to

$$[-L_x \ L_x X_1] Z_3 \begin{bmatrix} 0_{n \times r_1} \\ R_{x_1} \end{bmatrix} = Z_2 \quad (2.20)$$

$$[0_n \ I_n] Z_3 \begin{bmatrix} 0_n \\ I_n \end{bmatrix} - \delta I_n = [I_n \ 0_n] Z_3 \begin{bmatrix} I_n \\ 0_n \end{bmatrix} \quad (2.21)$$

$$Z_1 \geq \|Z_2\|_{\mathbb{F}} \quad (2.22)$$

$$Z_3 \geq 0_{2n} \quad (2.23)$$

where $z_i \triangleq \text{vec } Z_i$, $c_x \triangleq [1 \ 0_{1 \times 4n^2 + r_1s + 1}]^{\mathbb{T}} \in \mathbb{R}^{4n^2 + r_1s + 1}$, $z \triangleq [z_1 \ z_2^{\mathbb{T}} \ z_3^{\mathbb{T}}]^{\mathbb{T}} \in \mathbb{R}^{4n^2 + r_1s + 1}$, and $Z_1 = z_1 \in \mathbb{R}$ represents the value of the cost function $J_{11}(A)$, $Z_2 \in \mathbb{R}^{s \times r_1}$ represents the matrix $L_x(X_1P - Q)R_{x_1}$, $Z_3 \in \mathbb{R}^{2n \times 2n}$ represents the matrix in (2.18). The equality constraint (2.20) ensures that the blocks of Z_3 corresponding to P and Q are used to construct the cost function. The equality constraint (2.21) ensures that the matrix P is used to construct the (1,1) and (2,2) blocks of Z_3 . The quadratic constraint (2.22) ensures that we are minimizing the desired cost function. The linear matrix inequality constraint (2.23) enforces the stability constraint (2.18).

To recast the minimization problem in a form suitable for use with a convex optimization code, we rewrite the equality constraints as functions of z_i and obtain the optimization problem

$$\min c_x^{\mathbb{T}} z \quad (2.24)$$

subject to

$$\mathcal{A}_x z = b_x \quad (2.25)$$

$$z_1 \geq \|z_2\| \quad (2.26)$$

$$Z_3 \geq 0_{2n} \quad (2.27)$$

where $\mathcal{A}_x \in \mathbb{R}^{n^2 + r_1s \times 4n^2 + r_1s + 1}$ is given by

$$\mathcal{A}_x \triangleq \begin{bmatrix} 0_{r_1s \times 1} & -I_{r_1s} & [0_{r_1 \times n} R_{x_1}^{\mathbb{T}}] \otimes [-L_x \ L_x X_1] \\ 0_{n^2 \times 1} & 0_{n^2 \times r_1s} & [0_n \ I_n] \otimes [0_n \ I_n] - [I_n \ 0_n] \otimes [I_n \ 0_n] \end{bmatrix} \quad (2.28)$$

and $b_x \in \mathbb{R}^{n^2 + r_1s}$ is given by

$$b_x \triangleq \delta \begin{bmatrix} 0_{r_1s \times 1} \\ \text{vec } I_n \end{bmatrix} \quad (2.29)$$

where \otimes represents the Kronecker product and we have used the identity $\text{vec}(AXB) = (B^{\mathbb{T}} \otimes A) \text{vec } X$. While \mathcal{A}_x has $(n^2 + r_1s)(4n^2 + r_1s + 1) = 4n^4 + (5r_1s + 1)n^2 + r_1^2s^2 + r_1s$ entries, at most $2(r_1s + 1)n^2 + r_1s$ are nonzero. If we choose $L_x = R_{x_1} = I_n$, then

at most $n^3 + 4n^2$ are nonzero. Similarly, at most n of the $n^2 + r_1s$ entries of b_x are nonzero. This optimization problem involves minimizing a linear function over symmetric cones and can be solved using the SeDuMi MATLAB package [14]. Note that SeDuMi automatically forces positive-semidefinite matrices to be symmetric.

III. EXTENDED OBSERVABILITY MATRIX PROBLEM DESCRIPTION

In this section, we estimate the coefficient matrices of (2.1) and (2.2) given an estimate of the extended observability matrix $\Gamma_{0,q}$ obtained from a subspace identification algorithm, where, for $i \leq j$, $\Gamma_{i,j} \in \mathbb{R}^{p(j-i+1) \times n}$ is given by

$$\Gamma_{i,j} \triangleq \begin{bmatrix} CA^i \\ \vdots \\ CA^j \end{bmatrix}. \quad (3.1)$$

Our goal is to obtain a stable estimate of the dynamics matrix A given $\Gamma_{0,q}$.

The least squares problem is to minimize

$$J_\Gamma(A) \triangleq \|L_\Gamma(\Gamma_{1,q} - \Gamma_{0,q-1}A)R_\Gamma\|_F \quad (3.2)$$

where $L_\Gamma \in \mathbb{R}^{s \times qp}$, $R_\Gamma \in \mathbb{R}^{n \times r}$, $\Gamma_{1,q} \in \mathbb{R}^{qp \times n}$, and $\Gamma_{0,q-1} \in \mathbb{R}^{qp \times n}$. The minimizer of (3.2) is

$$\hat{A} \triangleq (L_\Gamma \Gamma_{0,q-1})^\perp L_\Gamma \Gamma_{1,q}. \quad (3.3)$$

Since \hat{A} may not be stable, we minimize (3.2) subject to the constraint that A be asymptotically stable.

As before, we ensure that A is asymptotically stable by using the linear matrix inequality constraint (2.18). The problem restated with the linear matrix inequality constraint is: $\min J_\Gamma(A)$ subject to (2.18). This is a quadratic programming problem with positive-semidefinite constraints. We let

$$R_\Gamma \triangleq PR_{\Gamma_1} \quad (3.4)$$

where $R_{\Gamma_1} \in \mathbb{R}^{n \times r_1}$. This weighting matrix will allow us to use the change of variables (2.16) leading to a convex optimization problem. Using (3.4), the cost function $J_\Gamma(A)$ becomes $J_\Gamma(A) = \|L_\Gamma(\Gamma_{1,q}P - \Gamma_{0,q-1}Q)R_{\Gamma_1}\|_F$. We can rewrite this minimization problem as the linear programming problem

$$\min c_\Gamma^T z \quad (3.5)$$

subject to

$$[-L_\Gamma \Gamma_{0,q-1} \ L_\Gamma \Gamma_{1,q}] Z_3 \begin{bmatrix} 0_{n \times r_1} \\ R_{\Gamma_1} \end{bmatrix} = Z_2 \quad (3.6)$$

$$[0_n \ I_n] Z_3 \begin{bmatrix} 0_n \\ I_n \end{bmatrix} - \delta I_n = [I_n \ 0_n] Z_3 \begin{bmatrix} I_n \\ 0_n \end{bmatrix} \quad (3.7)$$

$$Z_1 \geq \|Z_2\|_F \quad (3.8)$$

$$Z_3 \geq 0 \quad (3.9)$$

where $z_i \triangleq \text{vec}(Z_i)$, $c_\Gamma \triangleq [1 \ 0_{1 \times 4n^2+r_1s+1}]^T \in \mathbb{R}^{4n^2+r_1s+1}$, $z \triangleq [z_1 \ z_2^T \ z_3^T]^T \in \mathbb{R}^{4n^2+r_1s+1}$, and $Z_1 = z_1 \in \mathbb{R}$ represents the value of the cost function J_Γ , $Z_2 \in \mathbb{R}^{s \times r_1}$ represents the matrix $L_\Gamma(\Gamma_{1,q}P - \Gamma_{0,q-1}Q)R_{\Gamma_1}$, and $Z_3 \in \mathbb{R}^{2n \times 2n}$ represents the matrix in (2.18).

Rewriting the equality constraints as functions of z_i yields the optimization problem

$$\min c_\Gamma^T z \quad (3.10)$$

subject to

$$\mathcal{A}_\Gamma z = b_\Gamma \quad (3.11)$$

$$z_1 \geq \|z_2\| \quad (3.12)$$

$$Z_3 \geq 0_{2n} \quad (3.13)$$

where $\mathcal{A}_\Gamma \in \mathbb{R}^{n^2+r_1s \times 4n^2+r_1s+1}$ is given by

$$\mathcal{A}_\Gamma \triangleq \begin{bmatrix} 0_{r_1s \times 1} & -I_{r_1s} & [0_{r_1 \times n} \ R_{\Gamma_1}^T] \otimes [-L_\Gamma \Gamma_{0,q-1} \ L_\Gamma \Gamma_{1,q}] \\ 0_{n^2 \times 1} & 0_{n^2 \times r_1s} & [0_n \ I_n] \otimes [0_n \ I_n] - [I_n \ 0_n] \otimes [I_n \ 0_n] \end{bmatrix} \quad (3.14)$$

and $b_\Gamma \in \mathbb{R}^{n^2+r_1s}$ is given by

$$b_\Gamma \triangleq \delta \begin{bmatrix} 0_{r_1s \times 1} \\ \text{vec } I_n \end{bmatrix}. \quad (3.15)$$

While \mathcal{A}_Γ has $(n^2 + r_1s)(4n^2 + r_1s + 1) = 4n^4 + (5r_1s + 1)n^2 + r_1^2s^2 + r_1s$ entries, at most $2(r_1s + 1)n^2 + r_1s$ are nonzero. If we choose $L_\Gamma = I_{qp}$ and $R_{\Gamma_1} = I_n$, then at most $2(qp+1)n^2$ are nonzero. Similarly, n of the $n^2 + r_1s$ entries of b_Γ are nonzero. This optimization problem involves minimizing a linear function over symmetric cones and can be solved using the SeDuMi MATLAB package [14].

IV. EXAMPLES

Here, we apply the constrained optimization algorithm to several numerical examples. In each example, we add zero-mean white noise w to the output of the system scaled such that the signal-to-noise ratio $S/N \triangleq \|y(1:\ell) - w(1:\ell)\|_F / \|w(1:\ell)\|_F = 10$. The input sequence is a realization of a zero mean unit variance noise sequence. We find the least squares solution, the constrained optimization solution, the augmented data solution [1], and the zero padding solution [9] for each example. We then evaluate the cost functions $J_{x_{LS}}(A, B) \triangleq J_1(A, B)$ with $L_x = I_n$ and $R_x = I_\ell$, and $J_{\Gamma_{LS}}(A) \triangleq J_\Gamma(A)$ with $L_\Gamma = I_{qp}$ and $R_\Gamma = I_n$ at each solution, and compare them to the least squares solutions according to the error metrics $e_x(\hat{A}, \hat{B}) \triangleq (J_{x_{LS}}(\hat{A}, \hat{B}) - J_{x_{LS}}(\hat{A}_{LS}, \hat{B}_{LS})) / J_{x_{LS}}(\hat{A}_{LS}, \hat{B}_{LS})$, and $e_\Gamma(\hat{A}) \triangleq (J_{\Gamma_{LS}}(\hat{A}) - J_{\Gamma_{LS}}(\hat{A}_{LS})) / J_{\Gamma_{LS}}(\hat{A}_{LS})$, where \hat{A} is the solution obtained from the identification method, and \hat{A}_{LS} is the optimal least squares solution without stability enforcement. Finally, we plot the resulting eigenvalues. In all of the examples, we let $L_x = I_n$, $R_{x_1} = I_n$, $L_\Gamma = I_{qp}$, $R_{\Gamma_1} = I_n$, and $\delta = 0.001$. For the state sequence case, we use the method of [3]. For the extended observability case, we use the n4sid algorithm in the MATLAB system identification toolbox. For each example, we run the simulations for 100 realizations of the input and noise sequences and present the locations of the resulting eigenvalues, as well as the average values of the cost functions.

In the first example, the true system (4.1), shown at the top of page 1263, is unstable with $\rho(A) = 1.02$, $m = 2$, $n = 8$, $p = 4$, $q = 4$, and $\ell = 2^8 = 256$. The results for 10 of the 100 simulations are shown in Fig. 1. The least squares solution matches the unstable poles of the true system well, while the stability-constrained solution produced a stable estimate. The least squares and the stability-constrained solutions do not match the stable poles of the true system. The averaged error and the standard deviation of the error in each case are given in Table I.

In the second example, the true system (4.2), shown at the top of page 1263, is asymptotically stable with $\rho(A) = 0.99$, $m = 2$, $n = 8$, $p = 4$, $q = 4$ and $\ell = 2^8 = 256$. The results for 10 of the 100 simulations are shown in Fig. 2. The averaged error and the standard deviation of the error in each case are given in Table II.

It is interesting to note that the weighting matrix (2.11) does not distort the locations of the stable eigenvalues in the state sequence case as compared to the unweighted least-squares solution (2.6). However, the stable eigenvalues obtained using the weighting matrix (3.4) in the extended observability case differ from the eigenvalues of the unweighted least-squares solution (3.3).

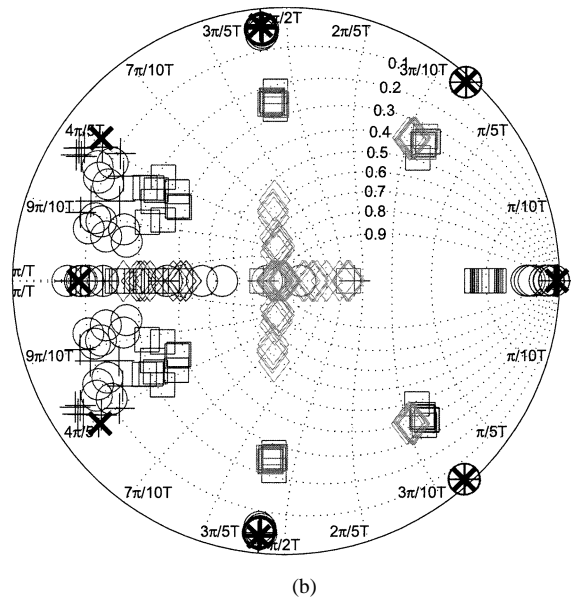
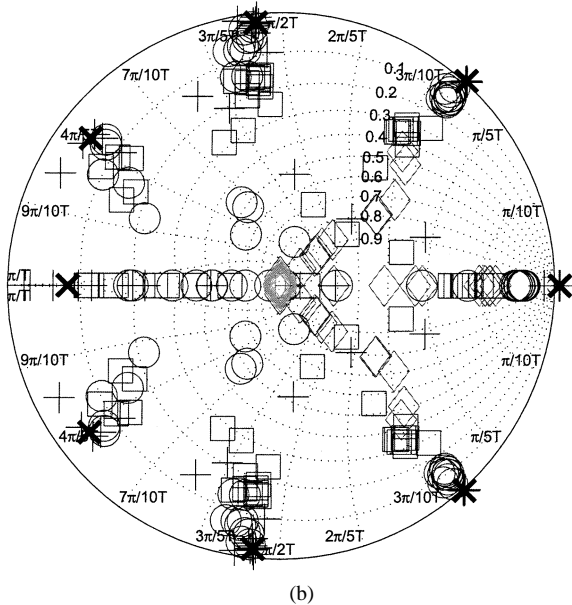
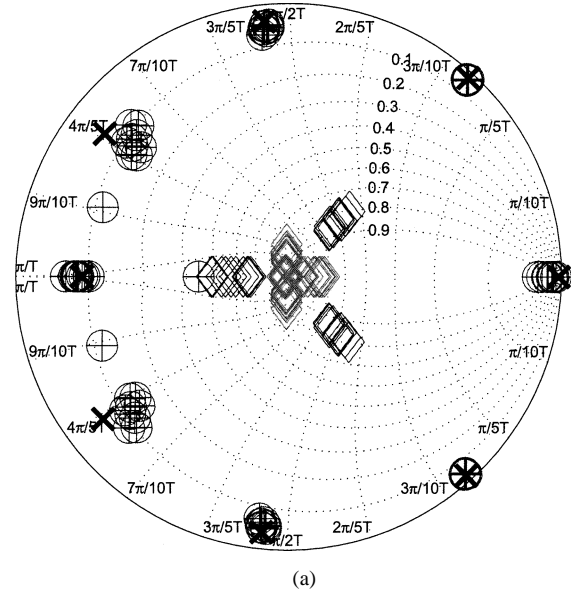
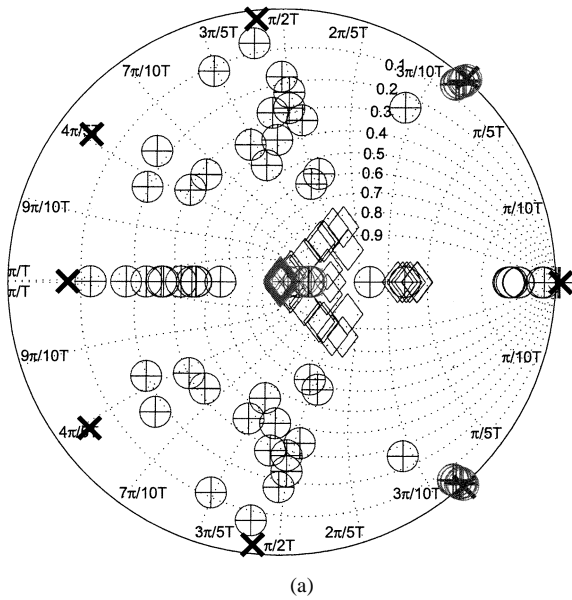


Fig. 1. System poles: \times represent true pole locations, $+$ represent pole estimates using the least squares solution without stability enforcement, \circ represent pole estimates from the constrained optimization algorithm, \diamond represent pole estimates from the data augmentation method [1], and \square represent pole estimates using the zero padding method [9]. (a) State sequence case. (b) Observability matrix case.

Fig. 2. System poles: \times represent true pole locations, $+$ represent pole estimates using the least squares solution without stability enforcement, \circ represent pole estimates from the constrained optimization algorithm, \diamond represent pole estimates from the data augmentation method [1], and \square represent pole estimates using the zero padding method [9]. (a) State sequence case. (b) Observability matrix case.

TABLE I
PERFORMANCE COMPARISON OF IDENTIFIED MODELS FOR THE
UNSTABLE EXAMPLE

Error Metric	Constrained Optimization	Data Augmentation	Zero Padding
e_x	0.1238 ± 0.2084	2.2571 ± 0.7611	—
$\rho(\hat{A})$	0.9575 ± 0.0552	0.4327 ± 0.0650	—
e_r	3.7386 ± 1.0929	17.8565 ± 8.3113	9.7894 ± 3.6829
$\rho(\hat{A})$	0.9331 ± 0.0163	0.6591 ± 0.0967	0.7609 ± 0.0274

TABLE II
PERFORMANCE COMPARISON OF IDENTIFIED MODELS FOR THE
ASYMPTOTICALLY STABLE EXAMPLE

Error Metric	Constrained Optimization	Data Augmentation	Zero Padding
e_x	0.0001 ± 0.0011	3.5751 ± 1.1523	—
$\rho(\hat{A})$	0.9737 ± 0.0053	0.3034 ± 0.0400	—
e_r	0.3812 ± 0.1821	21.6645 ± 6.7593	10.6295 ± 3.4102
$\rho(\hat{A})$	0.9791 ± 0.0037	0.6996 ± 0.0100	0.7309 ± 0.0167

V. CONCLUSION

We applied a convex optimization algorithm to the problem of identifying stable models in the context of subspace identi-

fication. We compared the solutions to previous methods [1], [9]. Our approach ensures that the identified model is asymptotically stable by introducing a weighted least squares cost function and a stability constraint in the form of a linear matrix inequality. Our

$$\left[\begin{array}{c|c} \frac{A}{C} & \frac{B}{D} \end{array} \right] = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0.5100 & 0.5100 & -1.9543 & 1.3296 \\ 1.0200 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.2572 & -0.1496 \\ 0 & 1.0200 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2918 & -0.3654 \\ 0 & 0 & 1.0200 & 0 & 0 & 0 & 0 & 0 & 0.5261 & -0.8707 \\ 0 & 0 & 0 & 1.0200 & 0 & 0 & 0 & 0 & 1.5873 & 1.6333 \\ 0 & 0 & 0 & 0 & 1.0200 & 0 & 0 & 0 & -0.6412 & -1.1886 \\ 0 & 0 & 0 & 0 & 0 & 1.0200 & 0 & 0 & -0.7587 & -1.1782 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.0200 & 0 & -0.4139 & -1.1780 \\ \hline 1.6921 & -1.3086 & -1.2142 & 1.3373 & -0.1417 & 0.7199 & 2.3559 & 1.4775 & -0.2636 & -0.0410 \\ -1.3291 & 0.4212 & 1.0219 & -0.2298 & -0.2106 & 0.1293 & -0.2946 & 0.0729 & -1.7673 & -0.2443 \\ 0.8306 & 0.1074 & -1.5395 & 0.000 & -0.8902 & 0.6599 & 1.2316 & -0.7068 & -2.5699 & -0.7960 \\ 0.9312 & -0.2612 & 0.4026 & -0.5150 & -0.3453 & 1.8806 & 1.0548 & -0.8103 & 0.8632 & 0.1359 \end{bmatrix} \quad (4.1)$$

$$\left[\begin{array}{c|c} \frac{A}{C} & \frac{B}{D} \end{array} \right] = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0.4950 & 0.4950 & -0.8056 & 1.4781 \\ 0.9900 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.5625 & -0.2717 \\ 0 & 0.9900 & 0 & 0 & 0 & 0 & 0 & 0 & 0.7474 & 2.1011 \\ 0 & 0 & 0.9900 & 0 & 0 & 0 & 0 & 0 & 0.2185 & -0.0900 \\ 0 & 0 & 0 & 0.9900 & 0 & 0 & 0 & 0 & 0.7854 & -0.1639 \\ 0 & 0 & 0 & 0 & 0.9900 & 0 & 0 & 0 & -1.7552 & 0.1018 \\ 0 & 0 & 0 & 0 & 0 & 0.9900 & 0 & 0 & 0.9748 & -0.4069 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.9900 & 0 & -1.0410 & -0.8454 \\ \hline -0.5973 & -1.5440 & -0.7101 & -0.7632 & 0.4998 & -1.0601 & -1.8191 & 1.7949 & -1.3871 & 0.8332 \\ 0.1902 & 0.5863 & -0.9760 & 0.2053 & 0.8722 & 0.6516 & 1.8280 & -0.3479 & 0.9179 & 0.2827 \\ -0.1573 & -0.0415 & 1.0886 & 0.3592 & 0.3033 & -0.9744 & -0.7285 & -0.9539 & -1.1044 & 0.0173 \\ -1.0262 & -3.3213 & -1.1302 & -1.6497 & 1.2149 & -0.4110 & -0.5041 & -0.0880 & 0.6041 & 2.2327 \end{bmatrix} \quad (4.2)$$

constrained optimization problem is conservative in that the Lyapunov inequalities $P - APA^T > 0_n$ and $P > 0_n$ are over-constrained as $P - APA^T \geq \delta I_n$ and $P \geq \delta I_n$ with $\delta > 0$. The amount of conservatism, characterized by δ , can be made arbitrarily small. This differs from the methods of [1], [4], [9], and [15], which can be arbitrarily conservative. Our approach is not computationally burdensome, typically requiring 10–20 iterations of the optimization algorithm and a couple seconds for the examples we presented. However, our optimization approach is more complex than some of the alternative methods that require algebraic manipulations and computing the eigenvalues or a singular value decomposition. The benefit of our approach is that the identified model is guaranteed to be stable with little degradation in accuracy compared to the possibly unstable least squares solution, and at low computational cost.

REFERENCES

- [1] N. L. C. Chui and J. M. Maciejowski, "Realization of stable models with subspace methods," *Automatica*, vol. 32, no. 11, pp. 1587–1595, 1996.
- [2] J.-N. Juang, *Applied System Identification*. Upper Saddle River, NJ: Prentice-Hall, 1993.
- [3] S. L. Lacy and D. S. Bernstein, "Subspace identification for nonlinear systems that are linear in unmeasured states," presented at the Conf. Decision Control, Orlando, FL, Dec. 2001.
- [4] —, "Subspace identification with guaranteed stability using constrained optimization," presented at the American Control Conf., Anchorage, AK, May 2002.
- [5] W. E. Larimore, *ADAPT_X Users Manual*. McLean, VA: Adaptics, Inc.
- [6] —, "System identification, reduced-order filtering and modeling via canonical variate analysis," in *Proc. Amer. Control Conf.*, San Francisco, CA, 1983, pp. 445–451.
- [7] —, "Generalized canonical variate analysis of nonlinear systems," in *Proc. 27th IEEE Conf. Decision Control*, Austin, TX, Dec. 1988, pp. 1720–1725.
- [8] L. Ljung, *System Identification: Theory for the User*, 2nd ed, ser. Prentice Hall Information and System Sciences. Upper Saddle River, NJ: Prentice-Hall, Jan. 1999.
- [9] J. M. Maciejowski, "Guaranteed stability with subspace methods," *Syst. Control Lett.*, vol. 26, pp. 153–156, 1995.
- [10] J. Mari, P. Stoica, and T. McKelvey, "Vector ARMA estimation: a reliable subspace approach," *IEEE Trans. Signal Processing*, vol. 48, pp. 2092–2104, July 2000.
- [11] M. Moonen, B. De Moor, L. Vandenbergh, and J. Vandewalle, "On- and off-line identification of linear state-space models," *Int. J. Control*, vol. 49, no. 1, pp. 219–232, 1989.
- [12] R. Pintelon and J. Schoukens, *System Identification: A Frequency Domain Approach*. Piscataway, NJ: IEEE Press, 2001.
- [13] T. Soderstrom and P. Stoica, *System Identification*. Upper Saddle River, NJ: Prentice-Hall, 1989.
- [14] J. F. Sturm, *Using SeDuMi 1.0x, a MATLAB Toolbox for Optimization Over Symmetric Cones*. Maastricht, The Netherlands: Dept. Quantitative Economics, Maastricht Univ., 1999.
- [15] T. Van Gestel, J. A. K. Suykens, P. Van Dooren, and B. De Moor, "Identification of stable models in subspace identification by using regularization," *IEEE Trans. Automat. Contr.*, vol. 46, pp. 1416–1420, Sept. 2001.
- [16] P. Van Overschee and B. De Moor, "N4SID: subspace algorithms for the identification of combined deterministic-stochastic systems," *Automatica*, vol. 30, no. 1, pp. 75–93, Jan. 1994.
- [17] —, "A unifying theorem for three subspace system identification algorithms," *Automatica*, vol. 31, no. 12, pp. 1853–1864, 1995.
- [18] —, *Subspace Identification for Linear Systems: Theory Implementation, Applications*. Norwell, MA: Kluwer, 1996.
- [19] M. Verhaegen, "Subspace model identification part 3. Analysis of the ordinary output-error state-space model identification algorithm," *Int. J. Control*, vol. 58, no. 3, pp. 555–586, 1993.
- [20] M. Verhaegen and P. Dewilde, "Subspace model identification part 1. The output-error state-space model identification class of algorithms," *Int. J. Control*, vol. 56, no. 5, pp. 1187–1210, 1992.
- [21] —, "Subspace model identification part 2. Analysis of the elementary output-error state-space model identification algorithm," *Int. J. Control*, vol. 56, no. 5, pp. 1211–1241, 1992.
- [22] M. Viberg, "Subspace-based methods for the identification of linear time-invariant systems," *Automatica*, vol. 31, no. 12, pp. 1835–1851, 1995.