Specialized Algorithm for Identification of Stable Linear Systems using Lagrangian Relaxation

Jack Umenberger and Ian R. Manchester

Abstract— **Recently Lagrangian relaxation has been used to generate convex approximations of the challenging simulation error minimization problem arising in system identification. In this paper, we present a specialized algorithm to optimize the convex bounds generated by Lagrangian relaxation, applicable to linear state-space models. The algorithm demonstrates superior scalability over general-purpose semidefinite programming solvers. In addition, we show empirically that Lagrangian relaxation is more resilient to a biasing effect commonly observed in other identification methods that guarantee model stability.**

I. INTRODUCTION

Linear time invariant (LTI) state-space models provide a useful approximation of dynamical system behavior in a multitude of applications. In some instances, models may be derived from first-principles; however, when such physical models are either unknown or too complex to be used for efficient simulation or control design, some form of data-driven modeling, i.e. system identification, is appropriate [1].

Two major problems in system identification are model instability and non-convexity of the standard quality-of-fit cost functions [2]. Subspace identification methods have been very successful for identifying linear state-space models from input-output data [3], [4]. However, the standard subspace algorithms do not ensure model stability and the least-squares model fit does not directly relate to long-term open-loop simulation error (a.k.a output error).

Stability of identified state space models generated by subspace identification was studied in [5], where it was noted that stability can be enforced by inserting blocks of zeros in the shifted state matrix. In [6] stability was imposed via regularization, while LMI parametrizations of stable models were given in [7]. In [8], a similar method allowed pole locations to be constrained to polytopic convex sets.

In a series of recent papers a new approach was developed for obtaining convex parameterizations of stable models and convex upper-bounds for simulation error [9], [10], and extended to problems with noisy data in [11], [12]. The precursor to these methods was the work of [13] (see also [14, Section 4]) which proposed a Lagrangian relaxation of simulation error minimization for models with nonlinear output feedback. Lagrangian relaxation (a.k.a the S-procedure), is a technique used extensively in robust control to convert difficult constrained optimization problems to tractable unconstrained approximations [15], [16].

In this work we apply the method of Lagrangian relaxation to the identification of LTI state-space models. The main contribution of this paper appears in Section IV where we develop a specialized algorithm for Lagrangian relaxation of simulation error minimization that is orders of magnitude faster than generic semidefinite programing solvers, allowing minimization of simulation error over longer time horizons, and for models of higher dimension. Having proposed an algorithm that makes Lagrangian relaxation computationally tractable in practice, Section V presents evidence that this approach is more resilient to a biasing effect observed in other methods that guarantee stability, in which noisy data leads to identified models that are 'too stable'.

II. PRELIMINARIES

A. Notation

The cone of real, symmetric nonnegative (positive) definite matrices is denoted by $\mathbb{S}_+^n$ ($\mathbb{S}_{++}^n$). The $n \times n$ identity matrix is denoted I_n. Let vec : $\mathbb{R}^{m \times n} \mapsto \mathbb{R}^{mn}$ denote the function that stacks the columns of a matrix to produce a column vector. The Kronecker product is denoted $\otimes$. The transpose of a matrix a is denoted a', and $|a|_Q^2$ is shorthand for $a'Qa$.

B. Linear state-space models

This paper concerns the identification of discrete-time LTI models of the form

$$x_{t+1} = Ax_t + Bu_t \tag{1a}$$

$$y_t = Cx_t + Du_t \tag{1b}$$

where $u \in \mathbb{R}^{n_u}$, $y \in \mathbb{R}^{n_y}$ and $x \in \mathbb{R}^{n_x}$ denote the input, output and state, respectively. In particular, we are interested in identifying stable models, for which A is Schur (i.e. spectral radius less than unity). We denote all unknown model parameters by the variable $\zeta = \{A, B, C, D\}$.

C. Problem data

We assume data of the form $\mathcal{Z}_T = \{\tilde{u}_t, \tilde{y}_t, \tilde{x}_t\}_{t=1}^T$ where $\tilde{u}$, $\tilde{y}$ denote measurements of the observed input and output, and $\tilde{x}$ denotes estimates of the, typically unobserved, state variable. For linear systems, subspace methods are a popular and effective approach to generating state estimates $\tilde{x}$ from measured input/output data [4]. Furthermore, these methods also yield effective estimates of the state dimension, n_x.

This work was supported in part by the Australian Research Council.

The authors are with the Australian Centre for Field Robotics (ACFR), Department of Aerospace, Mechanical and Mechatronic Engineering, University of Sydney, NSW 2006, Australia {j.umenberger, i.manchester}@acfr.usyd.edu.au

D. Simulation error

In system identification, a common measure of model quality, given a dataset $\mathcal{Z}_T$, is the *simulation error*, defined $\mathcal{E} \triangleq \sum_{t=1}^{T} |\tilde{y}_t - y_t|^2$ where y_t represents the *simulated output*, given by $y_t = CA^{t-1}\tilde{x}_1 + \Sigma_{\tau=1}^{t-1} CA^{t-1-\tau}B\tilde{u}_\tau + D\tilde{u}_t$. This dependence on the simulated output renders $\mathcal{E}$ a highly nonlinear function of the model parameters.

E. Problem statement

In this paper, our goal is to search over all stable linear systems to find that which minimizes (globally) the simulation error. Nonconvexity of the feasible set and nonlinearity of the objective, make this is a challenging optimization problem that we do not attempt to solve directly. Rather, we proceed by applying the method of Lagrangian relaxation to obtain a convex upper bound on simulation error.

III. LAGRANGIAN RELAXATION

The application of Lagrangian relaxation to identification of dynamical systems was proposed in [13]. In this section we recap the key ideas that underpin this approach and present some new results specific to the LTI case.

A. Relaxation of simulation error minimization

Global minimization of simulation error, i.e. the problem

$$\min_{\zeta,x} \sum_{t=1}^{T} |\tilde{y}_t - Cx_t - D\tilde{u}_t|^2$$
$$\text{s.t. } x_{t+1} = Ax_t + B\tilde{u}_t, \; x_1 = \tilde{x}_1$$

is made formidable by the highly nonconvex dependence of feasible state sequences, x, on model parameters; specifically $x_t = A^{t-1}\tilde{x}_1 + \Sigma_{\tau=1}^{t-1} A^{t-1-\tau}B\tilde{u}_\tau$. The Lagrangian relaxation of this problem takes the form

$$\hat{J}_\lambda(\zeta) \triangleq \sup_x \sum_{t=1}^{T} \{|\tilde{y}_t - Cx_t - D\tilde{u}_t|^2 \\ -\lambda_{t+1}(x_{t+1})'(x_{t+1} - Ax_t - B\tilde{u}_t)\} \quad (2)$$

for a sequence of Lagrange multipliers, λ_t. Notice that the multiplier $\lambda_t(x_t)$ is a function of the state variable; however, for brevity, we drop x_t from the notation. For arbitrary λ, the function $\hat{J}_\lambda(\zeta)$ has two key properties:

i. It is convex in ζ, as it is the supremum of an infinite family of convex functions; see Section 3.2.3 of [17].

ii. It is an upper bound for the simulation error. To see this, observe that if the supremizing x is such that $x_{t+1} = Ax_t + B\tilde{u}_t \; \forall t$, then $\hat{J}_\lambda(\zeta) = \mathcal{E}$, which implies that the supremum over all x can be no smaller.

B. Implicit models and stability guarantees

It is known that formulating the Lagrangian relaxation over equivalent constraints can give non-equivalent bounds. In light of this, we search over the following class of implicit models

$$Ex_{t+1} = Fx_t + Ku_t \quad (3a)$$
$$y_t = Cx_t + Du_t \quad (3b)$$

where E is invertible, such that $A = E^{-1}F$ and $B = E^{-1}K$. Let $\theta = \{E, F, K, C, D, P\}$ denote the parameters of the implicit model for which we shall search, along with a matrix $P \in \mathbb{S}_{++}^{n_x}$, the role of which is explained in the sequel. The bound then becomes

$$\hat{J}_\lambda(\theta) \triangleq \sup_x \sum_{t=1}^{T} \{|\tilde{y}_t - Cx_t - D\tilde{u}_t|^2 \\ -\lambda_{t+1}(x_{t+1})'(Ex_{t+1} - Fx_t - K\tilde{u}_t)\}. \quad (4)$$

One may interpret this formulation as that of (2) with the multipliers $E'\lambda_t$, thereby allowing a (partial) search for multipliers and model parameters simultaneously, while preserving convexity.

Furthermore, this implicit representation permits the definition of a convex parametrization of all stable LTI models:

Lemma 1. *Let Θ denote the set of all models θ of the form (3) and $P \in \mathbb{S}_{++}^{n_x}$ that satisfy the LMI*

$$M(\theta) = \begin{bmatrix} E + E' - P & F' & C' \\ F & P & 0 \\ C & 0 & I \end{bmatrix} > 0 \quad (5)$$

i.e. $\Theta \triangleq \{\theta : M(\theta) > 0\}$.

Then a model ζ of the form (1) is stable iff there exists E such that $\theta = \{E, EA, EB, C, D, P\} \in \Theta$.

Proof. Refer to [11, Section 3.2], in particular: Lemma 4 and Corollary 5. □

C. Choice of multiplier and finiteness of bound

The quality of the bound given by (4) is highly dependent on the choice of Lagrange multiplier, λ. Unfortunately, the simultaneous search over θ and λ to minimize (4) is not jointly convex, and thus the multiplier must be specified in advance. Given an estimated state sequence $\{\tilde{x}_t\}_{t=1}^{T}$, we consider multipliers of the form $\lambda_t = x_t - \tilde{x}_t$. This choice of multiplier guarantees that the supremum in (4) is finite:

Lemma 2. *Given any arbitrary sequence of vectors $\{\tilde{x}_t\}_{t=1}^{T}$, the supremum in (4) is finite, for Lagrange multipliers $\lambda_t = x_t - \tilde{x}_t$ and $\theta \in \Theta$.*

Furthermore, this choice of multiplier guarantees 'tightness' of the bound $\hat{J}_\lambda$ under the following (idealized) circumstances:

Lemma 3. *Let $\{\tilde{x}_t, \tilde{y}_t\}_{t=1}^{T}$ denote simulated,* noiseless *states and outputs of a stable model ζ^* in response to an arbitrary input $\{\tilde{u}_t\}_{t=1}^{T}$. Then for any $\theta^* \in \Theta$ that represents an equivalent implicit parametrization of ζ^*, we have $\hat{J}_\lambda(\theta^*) = 0$ with $\lambda_t = x_t - \tilde{x}_t$; i.e. the convex bound is tight.*

Proofs of Lemmas 2 and 3 are straightforward and omitted due to space restrictions.

D. Alternate problem formulation

To simplify the material in subsequent sections we reformulate the Lagrangian relaxation in terms of $\Delta_t = x_t - \tilde{x}_t$ and the so-called equation errors:

$$\epsilon_t = F\tilde{x}_t + K\tilde{u}_t - E\tilde{x}_{t+1}, \quad \eta_t = C\tilde{x}_t + D\tilde{u}_t - \tilde{y}_t.$$

Minimization of simulation error may now be formulated as

$$\min_{\theta,\Delta} \quad |\mathcal{G}(\theta)\Delta + \eta(\theta)|^2 \quad \text{s.t} \quad \mathcal{F}(\theta)\Delta - \epsilon(\theta) = 0 \qquad (6)$$

where $\Delta = \text{vec}([\Delta_1, \ldots, \Delta_T])$, $\eta(\theta) = \text{vec}([\eta_1, \ldots, \eta_T])$, $\epsilon(\theta) = \text{vec}([0, \epsilon_1, \ldots, \epsilon_{T-1}])$, $\mathcal{G}(\theta) = I_T \otimes C$ and

$$\mathcal{F}(\theta) = \begin{bmatrix} E & 0 & 0 & \cdots \\ -F & E & 0 & \ddots \\ 0 & -F & E & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix}.$$

Defining the Lagrangian by

$$J_\lambda(\theta, \Delta) = |\mathcal{G}(\theta)\Delta + \eta(\theta)|^2 - 2\Delta'(\mathcal{F}(\theta)\Delta - \epsilon(\theta)) \qquad (7)$$

the function $\hat{J}_\lambda(\theta)$ given in (4) is exactly equivalent to

$$\hat{J}_\lambda(\theta) = \max_{\Delta} \; J_\lambda(\theta, \Delta). \qquad (8)$$

Notice that the sup in (4) can be replaced with a max in (8) when $\theta \in \Theta$ as $J_\lambda(\theta, \Delta)$ is a concave quadratic function in Δ, by Lemma 2. This fact also leads to a closed form expression for the maximizing Δ^*,

$$\arg\max_{\Delta} \; J_\lambda(\theta, \Delta) = -(\mathcal{G}'\mathcal{G} - \mathcal{F}' - \mathcal{F})^{-1}(\mathcal{G}'\eta + \epsilon). \qquad (9)$$

E. Optimization as a semidefinite program

The convex optimization problem $\min_{\theta\in\Theta} \; \hat{J}_\lambda(\theta)$ can be formulated as a SDP, for which there exist good general purpose solvers. However, such a formulation introduces many additional variables, which leads to poor scalability; see, e.g., the results of Section IV-F. This motivates the search for a more computationally tractable alternative to SDP, which is the subject the following section.

IV. SPECIALIZED ALGORITHM

In this section we present the main contribution of this paper: an efficient, scalable algorithm for optimization of the Lagrangian relaxation, i.e. the problem $\min_{\theta\in\Theta} \; \hat{J}_\lambda(\theta)$. A complete listing is provided in Algorithm 1.

A. Path-following interior point method

Recall from Section III that the constraint $\theta \in \Theta$, equivalent to the LMI $M(\theta) > 0$, ensures both stability of the identified model and a finite, analytical solution to $\arg\max_\Delta \; J_\lambda(\theta, \Delta)$, which is needed to evaluate $\hat{J}_\lambda(\theta)$. To solve this constrained optimization problem, we employ a path-following interior point method (see, e.g., [18]). The basic idea is to introduce a barrier function that tends towards infinity at the boundary of the feasible set. For the constraint $M(\theta) > 0$, the usual choice [18, §6.4] is

$$\phi(\theta) = \begin{cases} -\log\det M(\theta) & M(\theta) > 0 \\ \infty & M(\theta) \not> 0 \end{cases}.$$

The barrier function, weighted by a scalar τ, is then added to the objective $\hat{J}_\lambda(\theta)$ and we solve a sequence of *unconstrained* optimization problems

$$\min_{\theta} \; f_\tau(\theta) = \hat{J}_\lambda(\theta) + \tau\phi(\theta)$$

for decreasing τ. In the specialized algorithm we propose, each of these unconstrained optimization problems is solved using a *quasi-Newton* method, which exploits the following two insights:

i. The structure of $\hat{J}_\lambda(\theta)$ permits an efficient computation of the gradient.

ii. The structure of $f_\tau(\theta)$ permits an approximation of the Hessian that combines BFGS estimation of $\nabla^2 \hat{J}_\lambda$ with analytical calculation of $\nabla^2\phi$.

B. Initialization

Interior point methods are iterative algorithms, requiring some initial parameter value θ_0, often chosen to be the analytic center. For the algorithm we propose, a valid initialization may be obtained by generating an arbitrary stable model (e.g., using Matlab's `drss` function) and setting $E = P$ as the solution of the Lyapunov equation $A'PA - P + C'C < 0$ (solved, e.g., by Matlab's `dlyap` function), leading to $\theta_0 = \{P, PA, PB, C, D, P\}$. During experimentation, the performance of the algorithm was insensitive to initialization of τ; we found $\tau_0 = 10^4$ to be an effective choice.

C. Gradient computation

The gradient of $f_\tau(\theta)$ w.r.t θ is given by

$$\nabla f_\tau(\theta) = \nabla \hat{J}_\lambda(\theta) + \tau\nabla\phi(\theta). \qquad (10)$$

The gradient of $\phi(\theta)$ is straightforward to compute. As $M(\theta)$ is linear in θ, there exists an affine mapping such that $\text{vec}(M(\theta)) = \mathcal{A}\theta + b$, where $\mathcal{A}$ and b are constant. Recall that for $g(Z) = \log\det Z$, where $Z \in \mathbb{S}_{++}$, we have $\nabla g = Z^{-1}$, and so by the chain rule

$$\nabla\phi = \left[\frac{\partial\phi}{\partial\theta_1}, \ldots, \frac{\partial\phi}{\partial\theta_{n_\theta}}\right]' = \mathcal{A}'\text{vec}(M(\theta)^{-1}). \qquad (11)$$

Now we seek an expression for the gradient of $\hat{J}_\lambda(\theta)$ at a particular parameter $\theta_0 \in \Theta$. From (8) we have $\hat{J}_\lambda(\theta) = J_\lambda(\theta_0, \Delta^*)$ where Δ^* is the solution to the linear system, given by (9). The gradient is then given by

$$\frac{\partial \hat{J}_\lambda}{\partial\theta} = \frac{\partial J_\lambda}{\partial\theta} + \frac{\partial J_\lambda}{\partial\Delta}\frac{\partial\Delta^*}{\partial\theta}.$$

As Δ^* is the maximizer of the smooth function $J_\lambda(\theta_0, \Delta)$, we have $\frac{\partial J_\lambda}{\partial\Delta} = 0$ at $\Delta = \Delta^*$, and so

$$\frac{\partial\hat{J}}{\partial\theta} = \left.\frac{\partial J_\lambda}{\partial\theta}\right|_{\theta=\theta_0, \Delta=\Delta^*}. \qquad (12)$$

The key point is that neither $\frac{\partial J_\lambda}{\partial\Delta}$ nor $\frac{\partial\Delta^*}{\partial\theta}$ need be computed to calculate the gradient of $\hat{J}_\lambda(\theta)$, which now reduces to

$$\frac{\partial\hat{J}}{\partial\theta_i}(\theta) = 2(\mathcal{G}\Delta^* + \eta)'(\mathcal{G}_i\Delta^* + \eta_i) - 2(\Delta^*)'(\mathcal{F}_i\Delta^* - \epsilon_i) \qquad (13)$$

where $\mathcal{G}_i, \eta_i, \mathcal{F}_i, \epsilon_i$ denote $\frac{\partial\mathcal{G}}{\partial\theta_i}, \frac{\partial\eta}{\partial\theta_i}, \frac{\partial\mathcal{F}}{\partial\theta_i}, \frac{\partial\epsilon}{\partial\theta_i}$, respectively.

D. Hessian approximation

The Hessian of $f_\tau(\theta)$ w.r.t θ is given by

$$\nabla^2 f_\tau(\theta) = \nabla^2 \hat{J}_\lambda(\theta) + \tau\nabla^2\phi(\theta).$$

The Hessian of $\phi(\theta)$ is also straightforward to compute, but somewhat cumbersome to express. Let $\mathcal{B} : \mathbb{S}^n \mapsto \mathbb{S}^{n^2}$ denote the function that maps a symmetric matrix $Z \in \mathbb{S}^n$ to the $n \times n$ block matrix, in which the $(i,j)^{\text{th}}$ block is given by $Z(:,j)Z(:,i)'$, where $Z(:,i)$ denotes the i^{th} column of Z. Then, by an application of the chain rule, the Hessian of the barrier function is given by

$$\nabla^2\phi = \begin{bmatrix} \frac{\partial^2\phi}{\partial\theta_1^2} & \frac{\partial^2\phi}{\partial\theta_1\partial\theta_2} & \cdots \\ \vdots & & \ddots \end{bmatrix} = \mathcal{A}'\mathcal{B}(M(\theta)^{-1})\mathcal{A}. \quad (14)$$

The Hessian of $\hat{J}_\lambda(\theta)$ involves, among other terms, $\frac{\partial\Delta^*}{\partial\theta}$. Rather than compute this quantity directly, at each barrier iteration we use an approximation that satisfies the secant condition, as in BFGS (see, e.g., [19, Section 6.1])); refer to Algorithm 1 (L18) for details.

E. Convergence

For each τ, 'quasi-Newton' iterations (L7-22) terminate when at least one of the following convergence criteria is satisfied: i) change in $f_\tau(\theta)$ is less than a prescribed tolerance, δ_f; ii) the maximum absolute value of an element of $\nabla f_\tau(\theta)$ is less than δ_g; iii) the step size αd_k is less than δ_f. The 'barrier iterations' (and thus, the algorithm) terminate when the change in $\hat{J}_\lambda(\theta)$ is less than a prescribed tolerance, δ_J. Recommended values for these parameters are summarized in Table I.

TABLE I: Parameter values for Algorithm 1.

Parameter	Description	Value
τ_0	Initial barrier weight	10^4
β	Barrier weight division factor	50
δ_f	Quasi-Newton objective tolerance	10^{-10}
δ_g	Quasi-Newton gradient tolerance	10^{-10}
δ_J	Objective convergence tolerance	10^{-11}
`maxit`	Max no. of quasi-Newton iterations	10^4

F. Computation time compared to semidefinite programming

We conclude this section by comparing computation time for the specialized Algorithm 1 with generic SDP solvers. Each SDP was formulated with Yalmip [20] and solved twice, once with SeDuMi [21] and again with Mosek v7.0.0.119. Table II shows computation times for identification SISO models of increasing order, n_x. The problem data was produced by models of order n_x randomly generated by Matlab's `drss` function, and $T = 400$ data points were used for identification. Tolerances were selected such that each solver converged to the same solution.

Similarly, Table III records computation time for each algorithm as the length of the training data set is increased. Each trial consisted of fitting a 4th order model to a system randomly generated by `drss`, of the same size. Notice that for $T \geq 2000$, both Mosek and SeDuMi failed to return a solution.

TABLE II: Computation time (in seconds, to 3 s.f.) for varying model order n_x and $T = 400$, averaged over 5 trials.

Model size, n_x	2	4	6	8
Specialized algorithm	2.48	9.41	27.6	70.1
Mosek 7.0.0.119	162	882	2550	7340
SeDuMi	319	2300	8200	22800

TABLE III: Computation time (in seconds, to 3 s.f.) for varying data length T and $n_x = 4$, averaged over 5 trials.

Data points, T	200	500	1000	2000
Specialized algorithm	4.76	11.6	22.9	73.7
Mosek 7.0.0.119	100	1750	16400	-
SeDuMi	211	6240	67300	-

It is clear that the proposed specialized algorithm demonstrates vastly superior scalability compared to generic SDP solvers. This improved performance makes Lagrangian relaxation computationally tractable in practice, particularly for problems of higher model order and many data points.

V. CASE STUDIES

In this section we investigate the quality of models identified with Lagrangian relaxation as compared to subspace identification with a model stability constraint [7].

A. Identification of randomly generated systems

To demonstrate the performance of Algorithm 1 on a wide variety of models, we conducted the following numerical experiment: Matlab's `drss` function was used to randomly generate forty 8th order SISO models. Each model was simulated over $T = 400$ time steps to generate input/output data $\{\tilde{u}_t, \tilde{y}_t\}_{t=1}^T$. A subspace algorithm was used obtain an approximate state sequence $\{\tilde{x}_t\}_{t=1}^T$ in a balanced basis; refer to [22] for details. 8th order models were then fit using two methods: i) Lagrangian relaxation, and ii) equation error subject to a model stability constraint, as in [7], henceforth referred to as 'stable subspace ID'. This process was repeated eight times for each model, over four different SNR.

The results of this experiment are shown in Figure 1, which records the validation error of each identified model, defined as the simulation error for a new validation input signal with no output noise, normalized by $\sum_{t=1}^T |\tilde{y}_t|^2$. It is clear that models identified with Algorithm 1 outperform those from stable subspace ID in the majority (86%) of trials.

B. Identification of a flexible beam

To gain further insight into the apparent superior performance of the Lagrangian relaxation, we consider identification of a flexible beam, which serves as a useful model of cantilever structures arising in many engineering applications. In particular, we fit 8th order models to a 12th order (6-link) beam model (see Figure 2 for Bode plot), which introduces a degree of *undermodeling* that is prevalent in practical applications. Once more, we compare models from Lagrangian relaxation to stable subspace ID.

Algorithm 1 MIN-LAGRANGIAN($\mathcal{Z}_T$)

```
Initialize $\theta_j = \theta_0$ as described in Section IV-B
Initialize $\tau_0$ to some arbitrarily large value
Initialize approximate $\nabla^2 \hat{J}_\lambda(\theta_0)$: $B = I_{n_\theta}$
while $|\hat{J}_\lambda(\theta_j) - \hat{J}_\lambda(\theta_{j-1})| > \delta_J$ do
    $\theta_k \leftarrow \theta_j$
    Set $f_\tau(\theta) = \hat{J}(\theta) + \tau_j \phi(\theta)$
    for $k = 1 : \texttt{maxit}$ do
        Compute $\nabla \hat{J}_\lambda(\theta_k)$ using (13)
        Compute $\nabla \phi(\theta_k)$ using (11)
        Form $\nabla f_\tau(\theta_k)$ using (10)
        Compute $\nabla^2 \phi(\theta_k)$ using (14)
        Approximate $\nabla^2 f_\tau(\theta_k)$ using $\nabla^2 \hat{J}_\lambda(\theta_k) \approx B$
            $$H_k = B + \tau_j \nabla^2 \phi(\theta_k)$$
        Solve for search direction $d_k = -H_k^{-1} \nabla f_\tau(\theta_k)$
        Compute the step length $\alpha$ by a backtracking line
        search to satisfy the Wolfe conditions
        Update the parameter estimate: $\theta_{k+1} = \theta_k + \alpha d_k$
        Set $s_k = \theta_{k+1} - \theta_k$
        Set $y_k = \nabla \hat{J}(\theta_{k+1}) - \nabla \hat{J}(\theta_k)$
        Update the approximation of $\nabla^2 \hat{J}(\theta_{k+1})$
            $$B \leftarrow B - \frac{B s_k s_k' B}{s_k' B s_k} + \frac{y_k y_k'}{y_k' s_k}$$
        if $|f_\tau(\theta_{k+1}) - f_\tau(\theta_k)| < \delta_f$ or
           $\|\nabla f_\tau(\theta_{k+1})\|_\infty < \delta_g$ or
           $\|\alpha d_k\|_\infty < \delta_f$ then
            $\theta_j \leftarrow \theta_k$ and break
        end if
    end for
    Set $\tau_{j+1} = \tau_j / \beta$ for some constant $\beta$
end while
return $\theta_j$
```

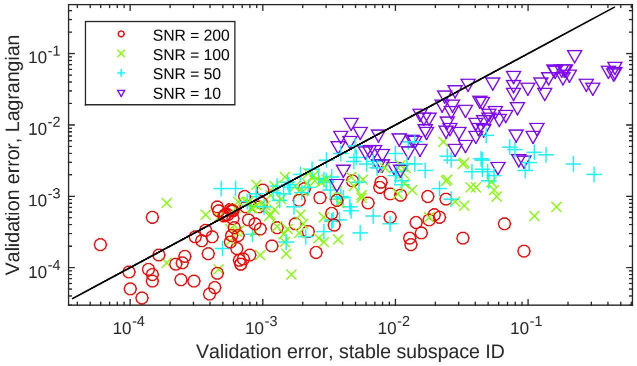


Fig. 1: Performance of Algorithm 1 compared with stable subspace ID for the identification of forty 8th order SISO models, randomly generated by Matlab's `drss` function.

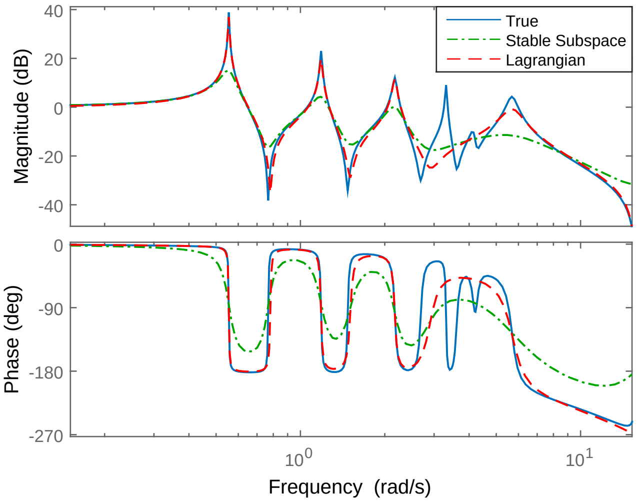


Fig. 2: Bode plots for the true 12th order flexible beam model (blue), and 8th order models identified by Lagrangian relaxation (red) and stable subspace ID (green). The true model was excited by white-noise, low-pass filtered with 20 rad/s 3dB cut-off frequency; the output SNR was 50 (17dB).

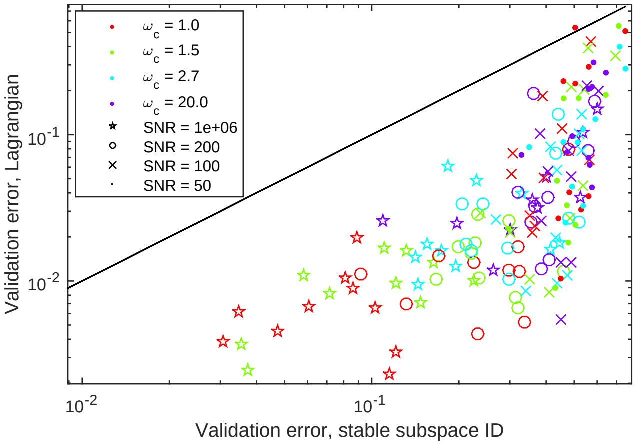
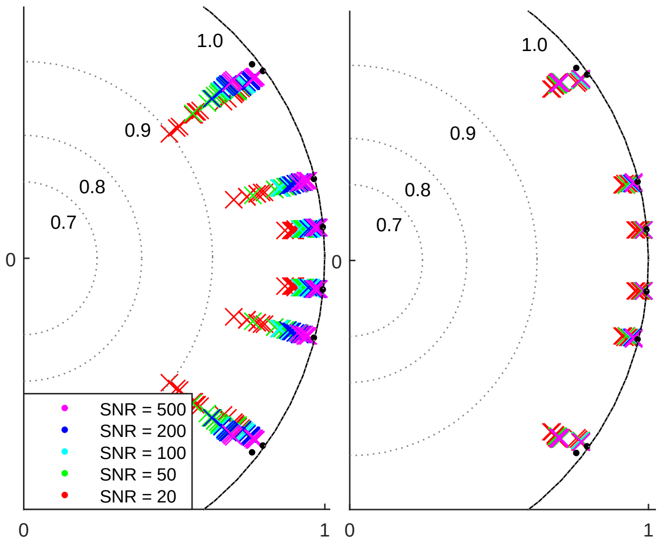


Fig. 3: Validation errors for 8th models identified with Algorithm 1 (vertical) and stable subspace ID (horizontal). Problem data generated by 12th order beam in Figure 2, for various frequency of excitation input and SNR.

The numerical experiment was conducted as follows: the true model was simulated for $T = 400$ time steps, excited by low-pass filtered white noise (of normalized signal power for each 3dB cut-off frequency, ω_c). The simulated output was corrupted by additive Gaussian noise (of various SNR, defined as the ratio of output signal power to output noise variance). As above, approximate states in a balanced basis were obtained by the subspace method of [22].

The results of this experiment are shown in Figure 3, where it is clear that models identified by Lagrangian relaxation outperform those from stable subspace ID. Further evidence of this is given in Figure 2, which shows (for a typical case) that models from Lagrangian relaxation better capture the dominant resonant peaks of the true model.



(a) Stable subspace ID. (b) Lagrangian relaxation.

Fig. 4: Pole locations of 8th order models fit to an 8th order flexible beam. The small dots denote the poles of the true model, '×' the poles of identified models.

C. Resilience to biasing effect

In this section we suggest that the improved performance of Lagrangian relaxation over other methods ensuring stability, e.g. [7], is the result of greater resilience to a biasing effect in which noisy data leads to models that are 'too stable'. This phenomenon, which is visible in Figure 2, is more clearly demonstrated in Figure 4, where the poles of models identified by stable subspace ID are shown to have been shifted considerably towards the center of the unit circle, compared to those of the models from Lagrangian relaxation.

VI. Conclusion

This work has developed a specialized algorithm for optimization of a Lagrangian relaxation of the simulation error minimization problem for LTI state-space models. This algorithm is more efficient than generic SDP solvers, demonstrating vastly superior scalability with both model dimension and number of data points used for identification. In addition, Lagrangian relaxation was shown empirically to be more resilient to a biasing effect observed in other methods that ensure stability, e.g. [7]; this resilience resulted in models that better captured resonant behavior.

Future research shall extend the specialized algorithm to the identification of nonlinear systems, which may be accomplished by considering Lagrangian relaxation of the so-called 'linearized simulation error' (see [9, §1.E]). In this setting, a key challenge is the expansion of the model class to include nonlinear systems, one approach for which introduces sum-of-squares constraints [9].

Finally, for identification of systems that include stochastic disturbances in the state transition (1a), the recent work of [23] proposed a reformulation of the expectation maximization algorithm, which culminates in a large simulation error minimization problem. A modified version of Algorithm 1 would enable application of [23] to systems of higher dimension, and is the subject of current research.

References

[1] L. Ljung, *System Identification: Theory for the User (2nd Edition)*, 2nd ed. Prentice Hall, Jan. 1999.

[2] ——, "Perspectives on system identification," *Annual Reviews in Control*, vol. 34, no. 1, pp. 1–12, Apr. 2010.

[3] W. E. Larimore, "System identification, reduced-order filtering and modeling via canonical variate analysis," in *American Control Conference, 1983*. IEEE, 1983, pp. 445–451.

[4] 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, 1994.

[5] J. M. Maciejowski, "Guaranteed stability with subspace methods," *Systems & Control Letters*, vol. 26, no. 2, pp. 153–156, 1995.

[6] T. Van Gestel, J. A. Suykens, P. Van Dooren, and B. De Moor, "Identification of stable models in subspace identification by using regularization," *Automatic Control, IEEE Transactions on*, vol. 46, no. 9, pp. 1416–1420, 2001.

[7] S. L. Lacy and D. S. Bernstein, "Subspace identification with guaranteed stability using constrained optimization," *Automatic Control, IEEE Transactions on*, vol. 48, no. 7, pp. 1259–1263, 2003.

[8] D. N. Miller and R. A. De Callafon, "Subspace identification with eigenvalue constraints," *Automatica*, vol. 49, no. 8, pp. 2468–2473, 2013.

[9] M. M. Tobenkin, I. R. Manchester, J. Wang, A. Megretski, and R. Tedrake, "Convex optimization in identification of stable non-linear state space models," *49th IEEE Conference on Decision and Control (CDC)*, no. 4, pp. 7232–7237, Dec. 2010.

[10] B. N. B. Bond, Z. Mahmood, Y. Li, R. Sredojevic, A. Megretski, V. Stojanovi, Y. Avniel, and L. Daniel, "Compact modeling of nonlinear analog circuits using system identification via semidefinite programming and incremental stability certification," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 29, no. 8, pp. 1149–1162, 2010.

[11] I. Manchester, M. M. Tobenkin, and A. Megretski, "Stable nonlinear system identification: Convexity, model class, and consistency," in *Proceedings of the 16th IFAC Symposium on System Identification (SYSID)*, Brussels, Belgium, 2012.

[12] M. M. Tobenkin, I. R. Manchester, and A. Megretski, "Stable nonlinear identification from noisy repeated experiments via convex optimization," in *American Control Conference (ACC), 2013*. IEEE, 2013, pp. 3936–3941.

[13] A. Megretski, "Convex optimization in robust identification of nonlinear feedback," *2008 47th IEEE Conference on Decision and Control*, pp. 1370–1374, 2008.

[14] M. M. Tobenkin, "Robustness Analysis for Identification and Control of Nonlinear Systems," Ph.D. dissertation, Massachusetts Institute of Technology, 2014.

[15] V. A. Yakubovich, "S-procedure in nonlinear control theory," *Vestnik Leningrad University*, vol. 1, pp. 62–77, 1971.

[16] I. Pólik and T. Terlaky, "A survey of the s-lemma," *SIAM review*, vol. 49, no. 3, pp. 371–418, 2007.

[17] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge: Cambridge University Press, 2004.

[18] Y. Nesterov, A. Nemirovskii, and Y. Ye, *Interior-point polynomial algorithms in convex programming*. SIAM, 1994, vol. 13.

[19] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.

[20] J. Löfberg, "Yalmip : A toolbox for modeling and optimization in MATLAB," in *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004. [Online]. Available: http://users.isy.liu.se/johanl/yalmip

[21] J. F. Sturm, "Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones," *Optimization methods and software*, vol. 11, no. 1-4, pp. 625–653, 1999.

[22] M. Moonen and J. Ramos, "A subspace algorithm for balanced state space system identification," *Automatic Control, IEEE Transactions on*, vol. 38, no. 11, pp. 1727–1729, 1993.

[23] J. Umenberger, J. Wagberg, I. Manchester, and T. Schön, "On identification via EM with latent disturbances and Lagrangian relaxation," in *Proceedings of the 17th IFAC Symposium on System Identification (SYSID)*, Beijing, China, 2015.