Convex Bounds for Equation Error in Stable Nonlinear Identification

Jack Umenberger and Ian R. Manchester

***Abstract*—Equation error, a.k.a. one-step-ahead prediction error, is a common quality-of-fit metric in dynamical system identification and learning. In this paper, we use Lagrangian relaxation to construct a convex upper bound on equation error that can be optimized over a convex set of nonlinear models that are guaranteed to be contracting, a strong form of nonlinear stability. We provide theoretical results on the tightness of the relaxation, and show that the method compares favourably to established methods on a variety of case studies.**

***Index Terms*—Identification, stability of nonlinear systems.**

I. Introduction

SYSTEM identification, a.k.a. learning dynamical systems from data, is a task of central importance in many areas of science and engineering. Major challenges include capturing long-term dynamic interactions, model stability, and non-convexity of optimization criteria [1], [2].

This paper concerns identification of *stable* discrete-time nonlinear state-space models of the form

$$x_t = a_\theta(x_{t-1}, w_t), \quad y_t = g_\theta(x_t, w_t). \tag{1}$$

Here, x_t is the internal state, and w_t, y_t denote the input and output, respectively. a_θ and g_θ are finitely parametrized by θ. State-space models offer a flexible description of dynamic behavior; special cases include autoregressive models, Wiener/Hammerstein models, and recurrent neural networks.

Models that lack stability can achieve good fit to training data, and yet behave unpredictably for previously unseen inputs. However, verifying model stability, let alone enforcing stability a priori, is a formidable challenge [1]. Similar problems have been encountered in the machine learning community (e.g. [2]), and in model reduction, where the data come from "snapshots" of a more complex model (e.g. [3]).

To quantify the quality-of-fit of the identified model, we will use the *equation error*:

$$J_{ee}(\theta) := \sum_{t=2}^{N} |\tilde{x}_t - a_\theta(\tilde{x}_{t-1}, \tilde{w}_t)|^2 + \sum_{t=1}^{N} |\tilde{y}_t - g_\theta(\tilde{x}_t, \tilde{w}_t)|^2,$$

a.k.a. *one-step prediction error*, where $\{\tilde{w}\}_{t=1}^N$ and $\{\tilde{y}_t\}_{t=1}^N$ denote measured inputs and outputs, respectively, from some dynamical system, and $\{\tilde{x}_t\}_{t=1}^N$ denotes estimates of the internal states. Here w_t is a "generalized input" in the sense that it can be given by any function of the previous measurable "physical" inputs, up until time t, e.g., $\tilde{w}_t = f_w(\tilde{u}_t, \ldots, \tilde{u}_{t-d})$. Equation error J_{ee} is a metric frequently used in identification problems, arising in scenarios such as:

Scenario one: minimization of equation error is a subproblem in many common approaches to system identification, e.g., subspace identification [4] where $\tilde{x}_t$ are estimated using matrix factorization, or expectation maximization (EM) for maximum likelihood identification, where $\tilde{x}_t$ denote samples from the joint smoothing distribution, c.f., [5].

Scenario two: equation error may be used as a surrogate for simulation error, i.e. the long-term prediction accuracy:

$$J_{se}(\theta) := \sum\nolimits_{t=1}^{N} |\tilde{y}_t - g_\theta(x_t, \tilde{w}_t)|^2,$$

a.k.a. output error, where $x_t = a_\theta(x_{t-1}, \tilde{w}_t)$, $x_1 = \tilde{x}_1$. Direct optimization of J_{se} is often highly challenging. By contrast J_{ee} is a convex function of θ when a_θ and g_θ are linearly parametrized. If desired, the minimizer of J_{ee} can then be used to initialize local optimization (e.g. via a gradient descent or Newton's method) of J_{se}.

The problem that we address in this paper can be summarized as follows: given training data $\{\tilde{u}_t, \tilde{x}_t, \tilde{y}_t\}_{t=1}^N$, find a model of the form (1) that (i) minimizes equation error J_{ee}, and (ii) is stable, in the sense defined in §II. For non-trivial model structures, characterising the subset of stable models in a computationally tractable way is the main difficulty.

The problem of model stability in identification of linear time invariant (LTI) systems has attracted considerable attention. In subspace identification a number of strategies to guarantee stability have been proposed, including: augmenting the extended observability matrix [6], regularization [7], eigenvalue/pole location constraints [8], [9], and iterative constraint generation approaches [10]. In all of these methods, there is a trade-off between (i) minimizing the desired cost-function, J_{ee}, at the expense of conservatism in the model class, and (ii) optimizing over all stable models, at the expense of distorting the cost-function. There are fewer results for guaranteeing stability of nonlinear models. In [11] results were given for reduction of linear models in feedback with certain classes of nonlinear dynamic models. The present paper builds upon the work of [12], [13], which proposed a convex parametrization of state-space models with guaranteed incremental stability and contraction properties, as well as a family of convex upper bounds on simulation error via *Lagrangian relaxation* [14].

In this paper, we investigate the use of Lagrangian relaxation (LR) to construct convex upper bounds on equation error that can be optimized over convex parametrizations of stable nonlinear models. We refer to this construction as LR of equation

J. Umenberger is with the Dept. of Information Technology, Uppsala University, Uppsala, 751 05, Sweden. I.R. Manchester is with the Australian Centre for Field Robotics (ACFR), Department of Aerospace, Mechanical and Mechatronic Engineering, University of Sydney, NSW 2006, Australia. E-mails: jack.umenberger@it.uu.se, i.manchester@acfr.usyd.edu.au.

error (LREE). Unlike [13], this does not provide a true bound on simulation error, but it offers significant advantages in terms of computation, and performs better empirically in terms of simulation error on some case studies.

The notation we use is as follows: the cone of real, symmetric nonnegative (positive) definite matrices is denoted $\mathbb{S}^n_+$ ($\mathbb{S}^n_{++}$). $A \succeq B$ ($A \succ B$) denotes $A - B \in \mathbb{S}^n_+$ ($A - B \in \mathbb{S}^n_{++}$). The $n \times n$ identity matrix is denoted I_n. The transpose of a matrix a is denoted a', and $|a|^2_Q$ is shorthand for $a'Qa$. For a polynomial p, $p \in$ sos denotes membership in the cone of sum-of-squares polynomials [15].

II. Parametrizations of stable models

In this paper, we will impose the following strong notion of stability on the identified models:

Definition 1 (Global incremental ℓ^2 stability)**.** *The model* (1) *is said to be* stable *if the sequence* $\{\bar{x}_t - \hat{x}_t\}_{t=1}^{\infty}$ *is square summable for every two solutions* $(\bar{w}, \bar{x})$ *and* $(\hat{w}, \hat{x})$ *of* (1)*, subject to the same input* $\bar{w} = \hat{w}$*.*

The main obstacle to ensuring stability of an identified model is the nonconvexity of the simultaneous search for model parameters *and* a certificate of stability (e.g., Lyapunov function). To circumvent this nonconvexity, we will work with the following class of *implicit* state-space models:

$$e_\theta(x_{t+1}, u_{t+1}) = f_\theta(x_t, u_t), \quad y_t = g_\theta(x_t, u_t), \tag{2}$$

where $e_\theta : \mathbb{R}^{n_x \times n_u} \mapsto \mathbb{R}^{n_x}$, $f_\theta : \mathbb{R}^{n_x \times n_u} \mapsto \mathbb{R}^{n_x}$ and $g_\theta : \mathbb{R}^{n_x \times n_u} \mapsto \mathbb{R}^{n_y}$ are multivariate polynomials or trigonometric polynomials, *linearly parametrized* by unknown model parameters $\theta \in \mathbb{R}^{n_\theta}$. Explicit dependence on θ is henceforth dropped for brevity. The model (2) is a generalization of the implicit models introduced in [12], which considered the special case of $e(x, u) = e(x)$. We shall enforce that $e(\cdot, u)$ is a bijection for all u, i.e., for all $u \in \mathbb{R}^{n_u}$ and $b \in \mathbb{R}^{n_x}$ there exists a solution $s \in \mathbb{R}^{n_x}$ to $e(s, u) = b$. This implies that (2) is equivalent to (1) with $a(x_{t-1}, w_t) = e^{-1}(f(x_{t-1}, u_{t-1}), u_t)$ and $w_t = (u_t, u_{t-1})$.

The convex model sets in [13] are based on contraction analysis [16] and differential L^2 gain [17], which utilize an augmented dynamical system comprising (2) and the differential dynamics $E(x_{t+1}, u_{t+1})\delta_{t+1} = F(x_t, u_t)\delta_t$, where $E(x, u) = \frac{\partial e(x,u)}{\partial x}$ and $F(x, u) = \frac{\partial f(x,u)}{\partial x}$. Consider the differential dissipation inequality $V_{t+1}(x_{t+1}, \delta_{t+1}) - V_t(x_t, \delta_t) \leq -\mu|\delta_t|^2$, where $V : \mathbb{Z} \times \mathbb{R}^{n_x \times n_x} \mapsto \mathbb{R}$ is a positive definite differential storage function. It can be shown that if the dissipation inequality holds for all solutions of (2) and the differential dynamics, then the model (2) is stable, c.f., [13, Thm 5]. [13, Thm 5] concerns a time-invariant V, however, the proof is identical for time-varying V_t. With the metric $V_t(x_t, \delta_t) = |E(x_t, u_t)\delta_t|^2_{P^{-1}}$, for $P \in \mathbb{S}^{n_x}_{++}$, the family of linear matrix inequalities (LMIs)

$$M(\theta, P, x, u) := \begin{bmatrix} E(x,u) + E(x,u)' - P - \mu I & F(x,u)' \\ F(x,u) & P \end{bmatrix} \succeq 0, \tag{3}$$

implies the dissipation inequality and, therefore, model stability, c.f. [13, §V]. Furthermore, $E(x, u) + E(x, u)' \succeq P + \mu I$, $\forall x, u$ ensures that the model is well-posed (i.e. $e(\cdot, \cdot)$ is a bijection), c.f. [13, Thm 1] To ensure that (3) holds for all x, u, one can constrain the polynomial $v'M(\theta, P, x, u)v$ to be sum-of-squares (SOS) [15], where $v \in \mathbb{R}^{2n_x}$ are indeterminate variables. This leads to the convex set of stable models:

$$\Theta := \{\theta : \exists P \in \mathbb{S}^{n_x}_{++} \text{ s.t. } v'M(\theta, P, x, u)v \in \text{sos}\}. \tag{4}$$

Note that the model $e(x, u) = x$, $f(x, u) = 0$ and $P = I$ leads to $M = I$, which always satisfies the conditions in (4) and hence the optimization problems we consider in this paper are never infeasible.

III. Lagrangian relaxation of equation error

Recall the problem statement from §I, which can be expressed as the optimization problem $\min_{\theta\in\Theta} J_{ee}(\theta)$, where Θ is the convex parametrization of stable models (4). For ease of exposition, consider the simplified problem

$$\min_\theta \ |\tilde{x}_t - a(\tilde{x}_{t-1}, \tilde{w}_t)|^2, \ \text{ s.t. } \theta \in \Theta, \tag{5}$$

i.e., minimization of equation error for a single data point, where $a(x_t, w_t) = e^{-1}(f(x_{t-1}, u_{t-1}), u_t)$. The problem

$$\min_{\theta, x_t} \ |\tilde{x}_t - x_t|^2, \ \text{ s.t. } e(x_t, \tilde{u}_t) = f(\tilde{x}_{t-1}, \tilde{u}_{t-1}), \ \theta \in \Theta \tag{6}$$

is equivalent to (5), as it has the same objective and feasible set. For clarity, we will refer to $\epsilon_t := e(\tilde{x}_t, \tilde{u}_t) - f(\tilde{x}_{t-1}, \tilde{u}_{t-1})$ as *implicit* equation error (at time t), to distinguish it from J_{ee}, which will henceforth refer to as *explicit* equation error. Problem (6) is not jointly convex in θ and x_t. In fact, by making use of implicit models, it may appear as though we have simply shifted the problem of nonconvexity from the model stability constraint to the cost function. In this section, we develop a convex approximation to this nonconvex problem based on Lagrangian relaxation (LR). To apply LR to (6) we construct the function

$$\hat{J}_t(\theta) := \sup_{x_t}\{|\tilde{x}_t - x_t|^2 - 2\lambda_t(x_t)'\left(e(x_t, \tilde{u}_t) - f(\tilde{x}_{t-1}, \tilde{u}_{t-1})\right)\} \tag{7}$$

where $\lambda_t(x_t)$ may be interpreted as a Lagrangian multiplier. For arbitrary $\lambda_t(x_t)$, the function $\hat{J}_t(\theta)$ has two key properties [13]: (i) it is convex in θ, as it is the supremum of an infinite family of convex functions, (ii) it is an upper bound for objective in (6). Problem (6) may then be approximated by the convex program $\min_{\theta\in\Theta} \hat{J}_t(\theta)$. A convex upper bound for J_{ee} can be constructed by summing the bounds for (6), i.e.,

$$\hat{J}(\theta) := \sum\nolimits_{t=2}^{N} \hat{J}_t(\theta) \geq J_{ee}. \tag{8}$$

We refer to $\hat{J}(\theta)$ as LR of equation error (LREE). The simultaneous search for θ and $\lambda_t(x_t)$ is not jointly convex; therefore, $\lambda_t(x_t)$ must be specified in advance. For the rest of this paper, we will make use of the multiplier $\lambda_t(x_t) = x_t - \tilde{x}_t$. This choice of multiplier simplifies computation of the bound, c.f. (12), and ensures perfect model recovery when $\tilde{x}$ constitutes noiseless states from a true model in Θ. It is possible to iteratively optimize $\lambda_t(x_t)$ given some 'current best guess' of the model, c.f. [18, §4.4] where this is applied for simulation error minimization.

Though convex, minimization of $\hat{J}(\theta)$ is non-trivial, as evaluating (7) requires the supremum of a multivariate polynomial.

A. Approximation via Sum-of-Squares Programming

In this section we propose three approximations of the problem $\min_{\theta\in\Theta} \hat{J}(\theta)$ based on sum-of-squares (SOS) programming [15], each of which is in a form that can implemented in a straightforward way using SOS parsers such as [22].

Firstly, we introduce the slack variable s_t and a SOS constraint to ensure $s_t \geq \hat{J}_t$:

$$s_t - |\tilde{x}_t - x_t|^2 + 2\lambda_t(x_t)'\left(e(x_t,\tilde{u}_t) - f(\tilde{x}_{t-1},\tilde{u}_{t-1})\right) \in \text{sos}, \tag{9}$$

for all $t = 2,\dots,N$. Combining this with the stability constraint (4) gives our first and most precise SOS upper bound for $\min_{\theta\in\Theta} J_{ee}(\theta)$:

SOS Program 1:

$$\min_{\theta,\{s_t\}_{t=2}^N} \sum\nolimits_{t=2}^N s_t, \ \text{s.t. } \theta\in\Theta, \ (9). \tag{10}$$

A simpler approximation can be obtained by replacing each SOS constraint (9) with an LMI constraint by linearizing the nonlinear term $e(x_t,\tilde{u}_t)$ at $\tilde{x}_t$, i.e. using the constraint

$$e(\tilde{x}_t,\tilde{u}_t) + E(\tilde{x}_t,\tilde{u}_t)(x_t - \tilde{x}_t) \approx f(\tilde{x}_{t-1},\tilde{u}_{t-1}), \tag{11}$$

for which the supremum over x_t can be explicitly calculated:

$$\begin{aligned} s_t &\geq \sup_{x_t} |\tilde{x}_t - x_t|^2 - 2\lambda_t(x_t)'\left(E(\tilde{x}_t,\tilde{u}_t)(x_t-\tilde{x}_t) + \epsilon_t\right), \\ &= \epsilon_t'(E(\tilde{x}_t,\tilde{u}_t) + E(\tilde{x}_t,\tilde{u}_t)' - I)^{-1}\epsilon_t =: \hat{J}_t^\ell(\theta), \end{aligned} \tag{12}$$

for all $t = 2,\dots,N$. Note that each constraint (12) can be represented as an LMI in s_t, θ via Schur complement. This leads to the second approximation:

SOS Program 2:

$$\min_{\theta,\{s_t\}_{t=2}^N} \sum\nolimits_{t=2}^N s_t, \ \text{s.t. } \theta\in\Theta, \ (12). \tag{13}$$

We refer to (13) as minimization of ‘linearized LREE’ and denote $\hat{J}^\ell(\theta) := \sum_{t=1}^N \hat{J}_t^\ell(\theta)$. The linearization (11) means that (13) does not minimize a true upper bound on $\hat{J}(\theta)$, rather an approximation, but in the authors experience both (10) and (13) have been observed to give very similar bounds, and consequently, very similar models. A disadvantage of (10) and (13) is that the number of constraints grows linearly with the number of data points, N, used for training.

The third approximation provides a true upper bound on $\hat{J}(\theta)$ and results in a SOS program that does not grow in complexity as a function of N, though it can be more conservative than the first approximation. The construction is based on an empirical moments representation, similar to that developed in [19] for J_{se}. The idea is as follows: instead of introducing a slack variable s_t for each data point, as in (9), we replace s_t with a function r such that

$$\begin{aligned} & r([\hat{x}_t,\hat{x}_{t-1},\hat{u}_t,\hat{u}_{t-1}]) - |\hat{x}_t - x_t|^2 \\ & + 2\lambda_t(x_t)'\left(e(x_t,\hat{u}_t) - f(\hat{x}_{t-1},\hat{u}_{t-1})\right) \geq 0, \end{aligned} \tag{14}$$

for all possible problem data $\hat{x}_t,\hat{x}_{t-1},\hat{u}_t,\hat{u}_{t-1}$ and all x_t. The function r is chosen to be a polynomial of degree d, such that nonnegativity in (14) can be enforced via SOS. When (14) holds, $r([\tilde{x}_t,\tilde{x}_{t-1},\tilde{u}_t,\tilde{u}_{t-1}]) \geq \hat{J}_t$ and so $\sum_{t=2}^N r([\tilde{x}_t,\tilde{x}_{t-1},\tilde{u}_t,\tilde{u}_{t-1}]) \geq \sum_{t=2}^N \hat{J}_t = \hat{J}$. We can then minimize an upper bound for $\min_{\theta\in\Theta}\hat{J}(\theta)$ by the following:

SOS Program 3:

$$\min_{\theta_r,\theta} \sum\nolimits_{t=2}^N r([\tilde{x}_t,\tilde{x}_{t-1},\tilde{u}_t,\tilde{u}_{t-1}]), \ \text{s.t. } \theta\in\Theta, (14), \tag{15}$$

where θ_r parametrizes r. Problem (15) has just two SOS constraints, $\theta\in\Theta$ and (14), regardless of N. For any given data points $[\tilde{x}_t,\tilde{x}_{t-1},\tilde{u}_t,\tilde{u}_{t-1}]$, r in (14) is an approximation of s_t in (9). The quality of the approximation is improved by increasing degree d of r, at the expense of increased computational cost. Before proceeding, we note that SOS programming is powerful, but can be computationally expensive: e.g., for an $n = n_x + n_u$-variate polynomial of degree $2d$, the dimension of the Gram matrix is $\binom{n+d}{d}$, and complexity is quartic (worst-case) in this dimension. As such, though constant in N, complexity grows quickly with *model size*.

To conclude this section we note that in the special case of LTI models, i.e. $Ex_{t+1} = Fx_t + Ku_t$ and $y_t = Cx_t + Du_t$, from which $A = E^{-1}F$ and $B = E^{-1}K$ can be recovered, several of the approximations become exact. First, there is no conservatism due to the stability constraint, i.e., Θ is a convex parametrization of all stable linear systems. Second, the LR $\hat{J}_t$ in (7) and the linearized LR $\hat{J}_t^\ell$ in (12) are equivalent, i.e., LR can be applied directly. Similarly, an exact empirical moments representation of $\hat{J}$ exists with quadratic r. One can enforce (14) exactly as an LMI, and (15) can be solved as an SDP. No approximations based on SOS or linearization are necessary.

IV. Quality of Bound on Equation Error

In §III we introduced a convex bound on *explicit* equation error, J_{ee}. In this section, we provide results on the quality of this bound. First, consider the case in which problem data $\{\tilde{u}_t,\tilde{x}_t\}_{t=1}^N$ represents noiseless inputs and states from some true model $\theta^*\in\Theta$, such that $J_{ee}(\theta^*) = 0$. From (12) it is apparent that $\hat{J}_t^\ell(\theta^*) = 0 \ \forall t$ with noiseless data, i.e., the bound $\hat{J}^\ell$ is tight to J_{ee} at the true model parameters, and SOS program 2 gives perfect model recovery. Next, we turn our attention to identification with noisy problem data. Consider minimization of explicit equation error J_{ee}, under the following assumptions:

1. The model belongs to (1) with $a(x,w) = a_\phi$ linearly parametrized by ϕ, to distinguish from the implicit model (2) which is linearly parametrized by θ,
2. The global minimizer $\phi^* := \arg\min_\phi J_{ee}(\phi)$ gives stable a_{ϕ^*} that is contracting w.r.t. the identity metric, i.e. $A_{\phi^*}'A_{\phi^*} \preceq -I \ \forall x,w$, where $A_{\phi^*} = \nabla_x a_{\phi^*}$.

Under these assumptions, SOS program 2 returns the same solution as unconstrained least squares, $\min_\phi J_{ee}(\phi)$.

Proposition 1. *Consider models of the form* (2) *with* $e(x,w) = Ex$ *for* $E\in\mathbb{R}^{n_x\times n_x}$ *and* $f_\theta(x,w)$*, parametrized by* θ*, such that* $f_\theta(x,w) = a_{\phi^*}(x,w)$ *for some value of* θ*. Then, under assumptions 1 and 2,* $(E^*,\theta^*) = \arg\min_{E,\theta}\hat{J}^\ell(E,\theta)$ s.t. $(E,\theta)\in\Theta$*, i.e.,* SOS program 2*, c.f.* (13)*, are such that* $(E^*)^{-1}f_{\theta^*}(x,w) = a_{\phi^*}(x,w)$*.*

Proof. For $e(x,w) = Ex$, the linearized constraint (11) is exact, and so linearized LREE $\hat{J}^\ell$ in (13) is equivalent to

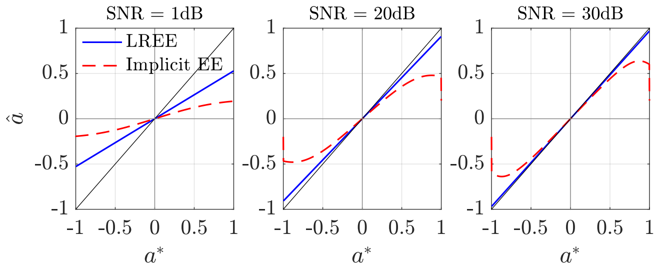


Fig. 1: Bias in stable identification for scalar linear systems, c.f. . $\hat{a}$ denotes the pole given by minimization of (19) (implicit EE) or (13) (LREE). a_* denotes the true system pole in (17). Signal-to-noise ratio (SNR) = z/σ.

'direct' LREE. Substituting $E = I$ and $f = a_{\phi^*}$ into (7) gives $\hat{J}^\ell = J_{ee}$, i.e., our convex bound is tight at ϕ^*. □

Note that Assumption 2 will not hold for unbounded functions a_ϕ, such as polynomials (except in the linear case). The assumption may, however, hold for bounded rational functions and trigonometric polynomials.

Next, we elucidate the advantages of LREE in §III compared to the simpler convex program

$$\min_{\theta\in\Theta} \sum\nolimits_{t=2}^{N} |e(\tilde{x}_t, \tilde{u}_t) - f(\tilde{x}_{t-1}, \tilde{u}_{t-1})|^2, \tag{16}$$

i.e., minimization of *implicit* equation error. We study the scalar linear case, and prove that LREE avoids a bias introduced by (16). Empirical evidence in §V demonstrates that this improvement in performance carries over to the nonlinear case. Consider the scalar linear output error model,

$$x_{t+1} = a_* x_t + b_* u_t, \quad u_t \sim \mathcal{N}(0,1), \tag{17a}$$
$$\tilde{x}_t = x_t + v_t, \quad v_t \sim \mathcal{N}(0,\sigma). \tag{17b}$$

Minimization of J_{ee} for (17) is an example of Scenario 2, c.f. §I, in which J_{ee} is used as a proxy for J_{se}. In the limit as $N \to \infty$ explicit equation error is given by

$$\sum\nolimits_{t=2}^{N} |\tilde{x}_{t+1} - a\tilde{x}_t - b\tilde{u}_t|^2 = \sigma(1+a^2) + z(a-a_*)^2 + (b-b_*)^2, \tag{18}$$

where $z = b_*/(1-a_*^2)$. The global minimizer is given by $a_{\text{ls}} = za_*/(\sigma+z)$, $b_{\text{ls}} = b_*$. Approximating (minimization of) (18) with (16) leads to:

Proposition 2. *For the implicit dynamics $ex_{t+1} = fx_t + ku_t$, in the limit $N \to \infty$ the cost in (16) is equivalent to*

$$\left(\sigma(1+a^2) + z(a-a_*)^2 + (b-b_*)^2\right)/(1-a^2)^2, \tag{19}$$

where $a = f/e$ and $b = k/e$.

Alternatively, LREE as in (13) returns the same solution as least squares, i.e. $(a_{\text{ls}}, b_{\text{ls}})$, as a consequence of Proposition 1. The global minimizer of (19) lacks a closed form expression that offers any insight, so we plot the minimizer, as a function of a_*, in Figure 1. It is apparent that LREE is significantly more accurate, especially when $|a_*| \approx 1$.

V. Case studies

In this section we examine the proposed method via a number of case studies. Matlab code that generates these results has been made available online [20].

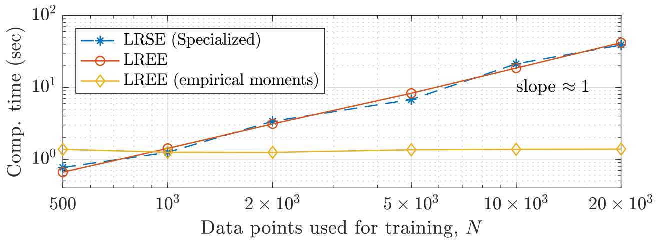
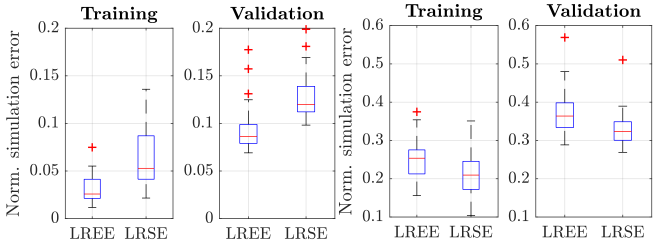


Fig. 2: Computation time as a function of training data length for a nonlinear identification problem, c.f., §V-A. LRSE is solved with the specialized algorithm in [21], LREE denotes the implementation in (13), i.e. SOS program 2 and LREE (empirical moments) denotes the empirical moments implementation of linearized LREE., i.e. SOS program 3. The polynomial r has degree 7, c.f., (15).

A. Comparison to relaxation of simulation error

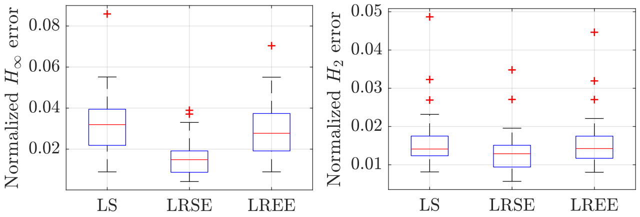
In this section, we explore differences between LREE and LR of simulation error (LRSE), c.f. [13, §V], [21], which provides a convex upper bound on simulation error, J_{se}. More precisely, we actually compare against LR of an approximation of J_{se} called *linearized* simulation error, c.f. [13, §V]. As discussed in §I, one of the motivations for minimization of J_{ee} is to provide a cheap, convex surrogate for J_{se}, i.e. Scenario two. LRSE also provides a convex approximation (upper bound) to J_{se}, although the construction requires computing the supremum of a function w.r.t. $x_{1:N} \in \mathbb{R}^{Nn_x}$. A naive implementation of LRSE leads to an SDP with computational complexity that scales as a cubic function of the number of data points, N; however, by carefully exploiting structure, complexity can be reduced to $O(N)$, c.f. [21]. In contrast, LREE provides a convex approximation (upper bound) to equation error. As shown in §III, its construction requires the supremum over a single state, x_t, to be computed N times, c.f., (7). Consequently, a direct implementation of (10) or (13) leads to convex programs that scale linearly with N, *without* the use of custom solvers. Furthermore, unlike LRSE, LREE admits an empirical moments representation (15), for which complexity remains constant with N. An empirical comparison of computation time is provided in Figure 2. All optimization problems were formulated with Yalmip [22], and solved using Mosek 8.1.0.37.

Next, we compare the quality of models identified with LREE and LRSE. LREE approximates J_{ee}, which is a function of all state-estimates $\tilde{x}_{1:N}$. LRSE approximates J_{se}, which depends only on $\tilde{x}_1$. As such, it is reasonable to expect that LREE will be more sensitive than LRSE to the quality of $\tilde{x}_{1:N}$. In what follows, we investigate this sensitivity on numerical examples. First, we consider identification of a simulated mass-spring-damper, for which the spring has a nonlinear force-displacement relationship. The system is excited by an applied force (measured input), and the displacement of the mass s is the measured output. We use two methods to obtain $\tilde{x}_{1:N}$. Figure 3(a) shows the results when $\tilde{x}_t \approx [s_t, \dot{s}_t]'$, i.e., position and velocity ($\dot{s}$ is approximated by applying a high-pass filter to s). That is, we exploit *a priori* knowledge of mechanical systems to construct high-quality state estimates. In this case, LREE performs better than LRSE. Figure 3(b) shows the results when $\tilde{x}_t \approx [s_t, s_{t-1}]'$, i.e., a truncated



(a) States from high-pass (HP) filter. (b) States from lagged outputs.

Fig. 3: Comparison of LREE and LRSE on a nonlinear mass-spring-damper. LREE performs better when the state estimates are of high quality, as in (a). The converse is true in (b), where states are arguably of lower quality.



(a) Normalized H_∞ error. (b) Normalized H_2 error.

Fig. 4: Comparison of: i) explicit equation error minimization (LS), LREE, and LRSE for a closed-loop subspace identification problem.

history of outputs. In this case, where $\tilde{x}$ are of arguably lower quality than before, the situation is reversed: LRSE performs better than LREE. One cannot draw general conclusions from a single experiment, but these results suggest that when you have high-quality state estimates $\tilde{x}_{1:N}$, equation error provides a good measure of model fit, and so LREE performs well. The converse is true when the state estimates are poor.

Next, we consider a problem that is ubiquitous in applications: identification of an LTI system while operating under closed-loop control. It is well known that methods based on equation error minimization, e.g. subspace identification, produce biased estimates in this setting [23]. Our experimental set-up is as follows: we randomly generate 30 stable LTI systems ($n_x = 7$) using Matlab's `drss`. Each system is simulated under closed-loop control, provided by a state-feedback linear quadratic regulator. White noise (SNR 27dB) is added to the output. State estimates $\tilde{x}_{1:N}$ are then produced by a subspace algorithm [4]. We emphasize that our goal is not to benchmark against state-of-the-art subspace algorithms, but rather, compare LREE and LRSE in the challenging closed-loop setting. The results are presented in Figure 4. It is apparent that LRSE has superior performance over both least squares and LREE.

In summary, LREE provides a computationally cheaper alternative to LRSE, and may even give better models when the state estimates $\tilde{x}$ are of 'high quality'. Conversely, when equation error is inappropriate due to poor-quality states or closed-loop identification LRSE may still be preferred.

B. Liquid-saturated steam heat exchanger

In this study, we identify a system in which water is heated by pressurized steam in a copper tube. The input $\tilde{u}_t$ is the

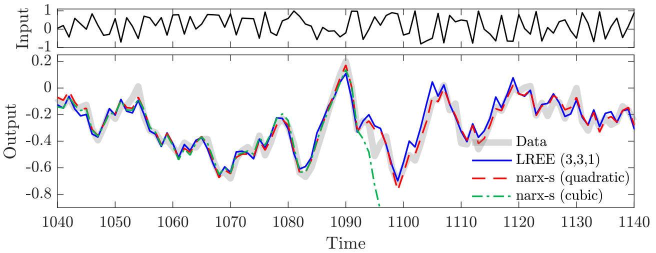


Fig. 5: Performance of identified models on validation set for the heat exchanger, c.f. §V-B. Despite simulating reliably for the first 1000 points, cubic *narx-s* diverges after an additional 95 timesteps of simulation.

liquid flow rate, and the output $\tilde{y}_t$ is the outlet liquid temperature. For further details, and problem data, c.f. Dataset 97-002 in the DaISy database [24]. We use $N = 1000$ datapoints for training, and the next 2000 datapoints in the sequence for validation. For states, we take $\tilde{x}_t = [\tilde{y}_t, \tilde{y}_{t-1}, \tilde{y}_{t-2}]' \in \mathbb{R}^3$.

We fit the following models: (i) *LREE*, as in SOS program 2 (13), (ii) *LRSE*, as in [21], (iii) *iEE-s*, using implicit EE (16), and (iv) *iEE*, using implicit EE *without* the stability constraint (but with a well-posedness constraint). All models use $e(x_t) = f(x_{t-1}, u_t, u_{t-1})$. We also fit (using the Matlab System Identification Toolbox) nonlinear ARX models $y_t = f_{\text{narx}}(\tilde{y}_{t-1}, \tilde{y}_{t-2}, \tilde{y}_{t-3}, \tilde{u}_t, \tilde{u}_{t-1})$ where f_{narx} is polynomial.

Results are presented in Table I, from which we make the following observations. First, for methods that enforce stability (*LREE*, *LRSE*, and *iEE-s*), performance on validation data improved with model complexity. The stability constraint appears to act as a regularizer. On the other hand, for NARX models there is clearly a trade-off between model complexity and validation performance. In particular, the most complex NARX model (cubic *narx-s*) achieves the best fit during training, but diverges during validation. An example trajectory is shown in Figure 5.

In Table II we further explore this phenomenon. The NARX models that performed best on validation data (quadratic *narx-p* and *narx-s*) are observed to diverge when subjected to synthetic inputs of slightly larger magnitude than the training data. In contrast, the proposed method provides models that are globally stable for all possible inputs.

So while NARX can provide very accurate models, indeed quadratic *narx-s* achieves the lowest validation error of all models in our test, the task of regressor selection can be challenging. In particular, it is difficult to guarantee sensible behaviour on unseen inputs, and therefore there are benefits to model sets that guarantee stability *a priori.*

Within the methods guaranteeing stability, the approach of [13], [21] (LRSE) offers slightly superior performance than LREE, at the expense of greater computation time.

C. Stable subspace identification for temporal textures

In this section we briefly illustrate the application to linear system identification, combining LREE with subspace identification. We compare least squares/N4SID (LS), LREE, the method of [8] (LB), and the method of [10] (CG). The data come from a temporal texture modeling problem, similar to that studied in [10], c.f. [25] for data. The goal is to learn

TABLE I: Computation time (sec) and simulation error (×100) on training (fit) and validation (val) for the heat exchanger, c.f. §V-B. For the state-space models of the form (2), *model* denotes the degrees of the polynomials (e, f, g). For the NARX models *narx-p* and *narx-s* (fit by targeting `prediction` and `simulation`, respectively), *model* denotes the monomials used in the nonlinear function f_{narx}. Computation times for *iEE-s* and *iEE* were < 1 sec.

	LREE			*LRSE*			*iEE-s*		*iEE*			*narx-p*			*narx-s*		
model	time	fit	val	time	fit	val	fit	val	fit	val	model	time	fit	val	time	fit	val
(1,1,1)	1.89	3.98	8.28	2.25	3.80	7.60	4.97	10.0	4.00	7.79	linear	0.065	2.82	11.0	5.47	2.37	10.9
(3,1,1)	3.98	2.70	8.59	18.1	2.31	6.32	4.18	9.88	5.96	17.6	quadratic	0.320	1.60	4.24	33.1	0.787	2.42
(3,3,1)	4.02	2.38	4.92	59.3	1.90	4.79	3.48	7.56	6.87	18.1	cubic	0.903	1.96	5.28	166	0.709	∞

TABLE II: Frequency of NARX model divergence (%, for 50 trials) for synthetic inputs (uniformly distributed, as in the validation set).

Max input, $\lvert u \rvert$	1.2	1.4	1.5	1.6	1.8	2.0	2.2
narx-p (quadratic)	0	0	0	0	0	32	100
narx-s (quadratic)	4	54	98	100	100	100	100

TABLE III: Performance of various methods on the temporal texture problem: dataset `steam.y.sub2` from [25]. Reconstruction error is defined as $(J_{ee} - J_{ee}(\theta_{\text{ls}}))/J_{ee}(\theta_{\text{ls}})$, where θ_{ls} is the least squares (LS) solution. Validation error is normalized simulation error on validation data, initialized from the final state used in training and simulated without inputs. $N = 80$ and $n_x = 40$ as in [10].

Method	LS	**LREE**	LB	CG
Reconstruction error	0	0.658	1.62	15.5
Validation error	1.077	0.963	1.038	1.217
Spectral radius	1.009	0.9920	0.9964	0.9990

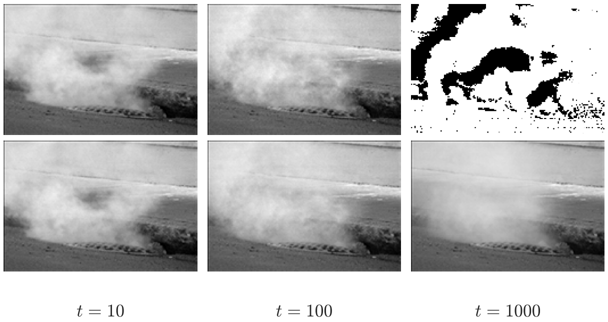
a model that generates realistic-looking sequences of steam rising from a manhole cover. Training data is a sequence of $N = 80$ images (115×170 pixels). A reduced representation, attained by a singular value decomposition, is used as a state basis ($n_x = 40$). Unconstrained minimization of J_{ee} (least squares) leads to an unstable model, which produces unrealistic sequences during long-term simulations (excited by white noise), c.f. Figure 6. Of the methods that guarantee stability, LREE achieves the best performance, c.f. Table III.

References

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

[2] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International Conference on Machine Learning*, 2013, pp. 1310–1318.

[3] D. J. Lucia, P. S. Beran, and W. A. Silva, "Reduced-order modeling: new approaches for computational physics," *Progress in Aerospace Sciences*, vol. 40, no. 1-2, pp. 51–117, 2004.



$t = 10$ $t = 100$ $t = 1000$

Fig. 6: Synthesized steam sequences from least squares (LS) (top) and LREE (bottom). The LS model is unstable, which leads to extreme saturation.

[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] T. B. Schön, A. Wills, and B. Ninness, "System identification of nonlinear state-space models," *Automatica*, vol. 47, no. 1, 2011.

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

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

[8] 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.

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

[10] S. Siddiqi, B. Boots, and G. J. Gordon, "A constraint generation approach to learning stable linear dynamical systems," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2007.

[11] B. Besselink, N. van de Wouw, and H. Nijmeijer, "Model reduction for nonlinear systems with incremental gain or passivity properties," *Automatica*, vol. 49, no. 4, pp. 861–872, 2013.

[12] M. M. Tobenkin, I. R. Manchester, J. Wang, A. Megretski, and R. Tedrake, "Convex optimization in identification of stable non-linear state space models," in *Proceedings of the 49th IEEE Conference on Decision and Control, CDC, Atlanta, USA*, 2010, pp. 7232–7237.

[13] M. M. Tobenkin, I. R. Manchester, and A. Megretski, "Convex parameterizations and fidelity bounds for nonlinear identification and reduced-order modelling," *IEEE Transactions on Automatic Control*, vol. 62, no. 7, pp. 3679–3686, 2017.

[14] C. Lemaréchal, "Lagrangian relaxation," in *Computational Combinatorial Optimization*. Springer, 2001, pp. 112–156.

[15] P. A. Parrilo, "Semidefinite programming relaxations for semialgebraic problems," *Mathematical Programming*, vol. 96, no. 2, 2003.

[16] W. Lohmiller and J.-J. E. Slotine, "On contraction analysis for non-linear systems," *Automatica*, vol. 34, no. 6, pp. 683–696, 1998.

[17] I. R. Manchester and J. J. E. Slotine, "Robust control contraction metrics: A convex approach to nonlinear state-feedback H^∞ control," *IEEE Control Systems Letters*, vol. 2, no. 3, pp. 333–338, 2018.

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

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

[20] J. Umenberger and I. R. Manchester, "Matlab files to reproduce selected examples." 2018. [Online]. Available: https://github.com/imanchester/NonlinearID

[21] ——, "Specialized Interior Point Algorithm for Stable Nonlinear System Identification," *accepted for IEEE Trans. Automatic Control. preprint: arXiv:1803.01066*, 2018.

[22] 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

[23] A. Chiuso and G. Picci, "Consistency analysis of some closed-loop subspace identification methods," *Automatica*, vol. 41, no. 3, 2005.

[24] B.L.R De Moor. DaISy: Database for the Identification of Systems, Dept. of Electrical Eng., ESAT/STADIUS, KU Leuven, Belgium. [Online]. Available: http://homes.esat.kuleuven.be/~smc/daisy/

[25] M. Szummer and R. W. Picard, "Temporal texture modeling," in *IEEE Intl. Conf. Image Processing*, vol. 3, Sep. 1996, pp. 823–826.