Guaranteed Stable Projection-Based Model Reduction for Indefinite and Unstable Linear Systems

Bradley N Bond
Research Laboratory in Electronics,
Massachusetts Institute of Technology,
Email: bnbond@mit.edu

Luca Daniel
Research Laboratory in Electronics,
Massachusetts Institute of Technology,
Email: luca@mit.edu

Abstract— **In this work we present a stability-preserving projection framework for model reduction of linear systems. Specifically, given one projection matrix (e.g. a right-projection matrix), we derive a set of linear constraints for the other projection matrix (e.g. the left-projection matrix) resulting in a projection framework that is guaranteed to generate a stable reduced model. Several efficient techniques for solving the proposed system of constraints are presented, including an optimization problem formulation for finding the optimal stabilizing projection, and a formulation with computational complexity independent of the size of the original system. The resulting algorithms can create accurate stable and passive models of arbitrary indefinite systems at a significantly cheaper cost than existing methods such as balanced truncation. Nevertheless, our algorithms integrate fully and effortlessly with most of the available standard model order reduction approaches for very large systems generated in VLSI applications (such as moment-matching methods, POD, or Poor Man's TBR), which can guarantee stability and passivity only in very specialized cases. Our algorithms have been tested on a large variety of typical VLSI applications, including field-solver-extracted models of RF inductors for analog applications, power distribution grids for large VLSI digital integrated circuits, and MEMS devices for sensing and actuation applications. The results have been successfully compared to those from existing and much more expensive stabilizing reduction techniques.**

I. Introduction

System-level modeling of interconnected circuit blocks and components producing large VLSI applications (e.g. RF receiver and transmitter chains, CPUs, DSP units, or sensing and actuation chips) is highly dependent upon each individual block model being accurate and stable. A common approach for efficiently handling linear components, such as RF inductors and interconnects, is to replace them with low-order models. If these individual models are not stable, then the entire system may become inaccurate; for instance, time-domain simulations of the full system may blow up. The task of reliably creating stable reduced models is especially difficult considering that many parasitic extractors and field-solvers today extract models with unstructured matrices (e.g. indefinite), or even extract models that are numerically unstable, despite being models of physically stable and passive systems.

The importance of preserving stability and passivity through model reduction was recognized long ago. In projection-based methods, it is common practice to sacrifice accuracy of the reduced model in order to guarantee stability. For example, Krylov-based methods [1], [2], [3], which guarantee accuracy by matching transfer function moments, are typically used only to match moments in one of the two projection matrices. The other projection matrix is then chosen to be the same in order to preserve stability and passivity through a Galerkin projection (i.e. a congruence transform) at the expense of not being able to match additional moments [4], [3]. It has been shown that a Galerkin projection is capable of preserving stability and passivity in stable systems described by sign-definite matrices, such as the RLC network models extracted by some field solvers [5], [6].

However, many stable systems are not described by semi-definite matrices. Such matrices frequently arise when modeling nonlinear systems or when considering linearizations of nonlinear systems, such as small-signal analog circuit models created by linearizing around the DC operating point. Stability-preserving model reduction schemes for nonlinear systems have recently been proposed that require computing stability-preserving projection matrices for a collection of indefinite linear models to obtain a nonlinear projection [7]. There exist a small number of methods, such as balanced truncation, that preserve stability [8] and passivity [9] for asymmetric and indefinite systems, but these techniques are computationally extremely expensive.

Additionally, models of stable passive systems extracted by field solvers may often be unstable due to error arising from discretization and from ignoring higher-order physical effects. These errors become a serious problem when attempting to connect unstable system blocks into a larger stable system, or when modeling distributed systems with frequency-dependent matrices, which requires interpolation between matrices [10]. To our knowledge, no projection method is able to reliably create accurate stable reduced models from originally unstable systems.

In this work we attempt to address the issues of efficient stable reduction of indefinite systems and stable reduction of mildly unstable systems. Specifically, we present a new approach for computing projection matrices that are guaranteed to preserve stability (and passivity). The projection matrices are typically chosen such that the right projection matrix guarantees accuracy (e.g. matching moments), and the left-projection matrix guarantees stability. This is achieved by deriving a set of linear constraints based on the idea of Lyapunov functions, resulting in a linear matrix inequality (LMI) whose solution is a stability-preserving projection matrix. Particular attention was devoted to formulating the LMI problem such that it is independent of the size of the original system, and only depends on the order of the reduced model.

In addition, we further define an optimization cost function to measure the distance between the subspace spanned by all stabilizing projection matrices and some optimal subspace chosen for accuracy (e.g. a subspace that enforces additional moment matching). The solution to the resulting optimization problem is a left-projection matrix that not only guarantees stability but also enforces a larger degree of accuracy.

As a final remark, we note that our proposed method could also be used to compute stabilizing projections for unstable systems.

The remainder of the paper is organized as follows. Section II presents an overview of linear system stability and projection-based model reduction. In Section III we derive a set of linear constraints

978-1-4244-2820-5/08/$25.00 ©2008 IEEE

for the left projection matrix that guarantees a stabilized reduced model. In the same section we further formulate an optimization problem whose solution is a stabilizing projection matrix that is optimal in terms of accuracy. Section IV considers the problem of efficiently computing various stabilizing projections, either analytically or through a low-order LMI. Finally, Section V tests the proposed stabilizing methods on several examples and compares their speed and accuracy to traditional methods.

II. Background

A. System Stability and Matrix Equations

A linear descriptor system

$$E\dot{x} = Ax + Bu, \qquad y = C^T x \tag{1}$$

is said to be stable if all solutions to the autonomous system ($u = 0$) do not stray arbitrarily far from the origin for some set of bounded initial conditions. More specifically, the system is stable if, given $\epsilon > 0$, there exists $\delta > 0$ such that

$$||x(t) - x_0|| < \epsilon, \quad \forall\ t > 0,\ ||x_0|| < \delta.$$

If, in addition to staying bounded, all trajectories eventually converge to the origin

$$\lim_{t\to\infty} ||x(t)|| = 0, \qquad \forall\ x_0,$$

then the system is said to be asymptotically stable. For linear systems we have the familiar constraint that the system is asymptotically stable if the pair (E, A) has all generalized eigenvalues with negative real part. In this case we say that (E, A) is a stable pair.

The preceding stability properties can be verified through the existence of Lyapunov functions [11], [12]. System (1) is said to be stable if there exists a function $L(x)$ which is positive-definite, i.e. $L(x) > 0$ for $x \neq 0$ and $L(0) = 0$, and also satisfies the condition that $\frac{\partial L(x)}{\partial t} \leq 0$. For asymptotic stability, the derivative constraint is slightly stronger: $\frac{\partial L(x)}{\partial t} < 0$ for $x \neq 0$ and $\frac{\partial L(0)}{\partial t} = 0$.

A common choice of Lyapunov functions for linear systems are quadratic functions. If we select the quadratic function $L(x) = x^T E^T P E x$ where P is a symmetric positive definite (SPD) matrix, then the condition on $\frac{dL(x)}{dt}$ becomes

$$E^T P A + A^T P E \preceq 0. \tag{2}$$

If we define $Q = -(E^T P A\ +\ A^T P E)$, then L is a Lyapunov function and the system is stable if Q is a symmetric positive semi-definite (SPSD) matrix, i.e. $Q \succeq 0$. If the system is asymptotically stable, then L is a Lyapunov function if Q is an SPD matrix.

More generally, consider the matrix equation

$$E^T P A + A^T P E = -Q \tag{3}$$

where $E, A, P, Q \in \mathbb{R}^{N\times N}$. If System (1) is asymptotically stable, i.e. the matrix pair (E, A) has all eigenvalues with negative real part, then for any SPD matrix Q there exists a unique SPD matrix P which solves (3). Conversely, if there exist SPD matrices Q, P satisfying (3), then the matrix pair (E, A) has all eigenvalues with negative real part and the system is asymptotically stable [13]. If the matrix E is singular, then there may not exist an SPD solution P for some SPD Q, and if there are solutions they may not be unique.

There are several subtle issues regarding the various types of stability, e.g. conditions under which the system is asymptotically stable for SPSD Q, but for the sake of simplicity we will not bring up such issues in this work and present only the more general conditions.

A stronger notion of stability is passivity. Passivity is a property which guarantees the system does not produce energy, and can be proven, along with other notions of dissipativity, through the use of storage functions. A storage function is essentially a generalized version of the Lyapunov function which incorporates inputs. The derivation of such matrix equation conditions for dissipative systems is similar to that of the Lyapunov equation above, so here we directly introduce the well-known positive-real lemma for passivity. The linear system with $E = I$ is passive if there exists an SPD solution P to the following systems of equations [14], [15]

$$PA + A^T P \preceq 0, \qquad PB = C.$$

For the case of descriptor systems where $E \neq I$, this has been generalized to [16]

$$E^T P A + A^T P E \preceq 0 \tag{4}$$

$$E^T P B = C. \tag{5}$$

B. Projection-Based Model Reduction

Projection-based approaches to model reduction rely on the assumption that the important dynamics of a system are approximately confined to a low dimension subspace. In such cases it is possible to find a right-projection matrix $V \in \mathbb{R}^{N\times q}$ such that $x \approx Vz$ is a good approximation to solutions of System (1). Here $z \in \mathbb{R}^q$ is referred to as the reduced state vector, and we assume that $q << N$. After this approximation, the system can be viewed as an overdetermined set of equations.

$$EV\dot{z} = AVz + Bu$$

The number of equations are typically reduced by forcing the residual of the system to be orthogonal to a space defined by the column span of a matrix U

$$U^T\left(EV\dot{z} - AVz - Bu\right) = 0.$$

We refer to $U \in \mathbb{R}^{N\times q}$ as the left-projection matrix. The final result is an order q linear system

$$\hat{E}\dot{z} = \hat{A}z + \hat{B} \qquad y = \hat{C}^T z \tag{6}$$

where $\hat{E} = U^T E V$, $\hat{A} = U^T A V$, $\hat{B} = U^T B$, and $\hat{C} = V^T C$ are the reduced system matrices.

One common approach to constructing the projection matrices U and V is to pick their columns such that their column span enforces a match between the first moments of the transfer function of the reduced system and those of the original system. Such approaches are projection versions of Pade-type approximations to the transfer function [17], [18]. If we define the k_{th} right Krylov vector v_k at frequency s_r and the k_{th} left Krylov vector u_k at frequency s_l as

$$v_k = \left((s_r E - A)^{-1} E\right)^k (s_r E - A)^{-1} B, \tag{7}$$

$$u_k = C^T (s_l E - A)^{-1} \left(E (s_l E - A)^{-1}\right)^k, \tag{8}$$

then projection matrices V and U constructed such that

$$\text{range}(V) \supset v_k \quad k = 0 \ldots m$$

$$\text{range}(U) \supset u_k \quad k = 0 \ldots m,$$

where range$(V) = \{x \mid x = Vz\}$, will guarantee that the transfer function of the reduced model will match the transfer function of the original model in exactly m derivatives at frequency points s_r and s_l. Other methods such as frequency-domain POD [19] and Poor Man's TBR [20] guarantee zeroth-moment transfer function matching at multiple frequency points.

C. Stable Model Reduction

Although the projection matrices U and V can be separately chosen to enforce accuracy in the resulting reduced model, it is also important to consider stability and passivity. It is common practice to sacrifice additional accuracy in order to preserve stability.

For stable systems described by sign-definite matrices, a Galerkin projection (i.e. congruence transform) will preserve stability and passivity. That is, the reduced model

$$V^T E V \dot{z} = V^T A V z + V^T B u \qquad\qquad y = C^T V z \tag{9}$$

is stable and passive. Such systems include models with SPD descriptor matrix E and system matrix such that $-(A + A^T)$ is SPD, which often arise when modeling RLC networks [3], [6]. Stability and passivity are preserved in this case because the congruence transform preserves definiteness of the matrices. That is, the reduced descriptor matrix $\hat{E} = V^T E V$ is SPD and reduced system matrix $\hat{A} = V^T A V$ is such that $-\hat{A} + \hat{A}^T$ is SPD.

The Galerkin projection scheme is convenient for systems described by symmetric or definite matrices, but it does not guarantee stability or passivity for systems described by arbitrary stable matrix pairs. Such nonsymmetric indefinite systems occur frequently in VLSI applications as the result of a discretization of PDEs, or when considering linearizations of a nonlinear system. Stability for such indefinite systems can be preserved through other projections such as balanced truncation, but due to the order N^3 complexity, such procedures are not practical for large systems.

III. Stabilizing Projection

A. Stability Conditions

Given a linear descriptor System (1) and a pair of projection matrices U and V, we wish to find conditions under which the reduced order model

$$U^T E V \dot{z} = U^T A V z + U^T B u, \qquad\qquad y = C^T V z \tag{10}$$

is stable. In this work we are primarily interested in the selection of the left-projection matrix U. That is, given a right-projection matrix V (generated by any well known technique in VLSI model order reduction such as moment matching, POD or Poor-Man's TBR), we will propose a routine for automatically generating left-projection matrices U which preserve stability.

From Section II-A, we know the reduced system will be stable if there exists an SPD matrix $\hat{P}$ such that

$$\hat{E}^T \hat{P} \hat{A} + \hat{A}^T \hat{P} \hat{E} = -Q_2 \prec 0, \tag{11}$$

for some SPD matrix Q_2. Recall that this equation is the result of the condition $\dot{\hat{L}}(z) < 0$, corresponding to the Lyapunov function $\hat{L}(z) = z^T \hat{E}^T \hat{P} \hat{E} z$. That is, if there exists an SPD solution $\hat{P}$ to condition (11), then $\hat{L}(z)$ is a Lyapunov function for System (10), and thus System (10) is asymptotically stable. Condition (11), however, is not desirable from a computational point of view, because it is quadratic in U

$$V^T E^T U \hat{P} U^T A V + V^T A^T U \hat{P} U^T E V = -Q_2 \prec 0, \tag{12}$$

making it an extremely difficult system of equations to solve.

Without a loss of generality, we can transform condition (12) into a pair of linear constraints by defining $\tilde{U} = U \hat{P} U^T E V$. The resulting stability condition becomes

$$\tilde{U}^T A V + V^T A^T \tilde{U} = -Q_2 \prec 0,$$

which corresponds to the Lyapunov function $\hat{L}(z) = z^T \tilde{U}^T E V z$. In order for $\hat{L}$ to be a Lyapunov function, we also require $\hat{L}(z) > 0$. Thus, we must enforce the additional constraint that $\tilde{U}^T E V = Q_1$ is an SPD matrix. Note that we have now replaced the single quadratic constraint in U with a pair of linear constraints in $\tilde{U}$

$$\tilde{U}^T A V + V^T A^T \tilde{U} = -Q_2 \prec 0, \quad \tilde{U}^T E V = Q_1 \succ 0. \tag{13}$$

Prop. 3.1 (Equality of Stability Constraints): The stability conditions (12) *and* (13) *are equivalent, provided the reduced matrix* $U^T E V$ *has full rank.*

Proof: Assume $U, \hat{P}, Q_2$ solve (12). We may then choose $\tilde{U} = U \hat{P} U^T E V$ as a solution to (13) for Q_2, and we find that $Q_1 = V^T E^T U \hat{P} U^T E V$, which is SPD because it is a congruence transform of an SPD matrix $\hat{P}$ with a full rank matrix operator $U^T E V$. On the other hand, assume $\tilde{U}, Q_1, Q_2$ solves (13). We may then choose $U = \tilde{U}$ as a solution to (12) for Q_2 and for $\hat{P} = (\tilde{U}^T E V)^{-1} = Q_1^{-1}$, which is an SPD matrix because Q_1 is an SPD matrix. Thus, the two sets of constraints are equivalent. ■

As a result of this equivalence between stability conditions, we shall drop the tilde notation from here on and consider only the set of linear constraints (13), now denoted as

$$U^T E V = Q_1 \succ 0 \tag{14}$$

$$U^T A V + V^T A^T U = -Q_2 \prec 0. \tag{15}$$

Any U which satisfies these constraints will create a stable reduced model. In addition, note that there is no requirement that E be invertible, so we may handle the case of singular E matrix, provided $U^T E V$ is non-singular.

It is important to note here that if for numerical reasons we need to further enforce orthogonality between U and EV, then given a U which solves (14) and (15), we may always redefine $U = U(V^T E^T U)^{-1}$ to ensure that $U^T E V = I$. This redefinition does not affect the stability or accuracy of the reduced model because it does not change the column span of U. Such a redefinition can be thought of as multiplying both sides of the reduced model by $\hat{E}^{-1}$, which we know is invertible because we have forced it to be an SPD matrix (Q_1). Additionally, we point out that instead of fixing the right-projection matrix U and finding a stabilizing left-projection V, it is also possible to do the opposite and fix the left-projection U while finding a stabilizing right-projection V.

In this section and for the remainder of the paper we consider the asymptotic stability constraint that Q_2 is an SPD matrix. It is also of course possible to consider the looser constraint that Q_2 is an SPSD matrix without changing any of the results presented in this paper.

B. Feasible Stabilizing Solutions

In the previous section we derived a set of linear constraints for a stabilizing projection matrix U. In this section we examine the conditions guaranteeing that solutions to the system of constraints exist.

First, note that it is possible to transform the matrix equations into a single linear system. We may rewrite equation (15) as a single matrix linear in U and concatenate the result with equation (14) to obtain

$$M^T U = B^T, \qquad M = [V_e, \ V_a], \qquad B = [Q_1, \ -Q_2]. \tag{16}$$

where $V_e = EV$ and V_a is a rearrangement (15) such that $U^T V_a = U^T A V + V^T A^T U$.

Since $U \in \mathbb{R}^{N \times q}$ and $M \in \mathbb{R}^{N \times 2q}$, this is an underdetermined system of equations with q^2 equations and Nq unknowns. If the columns of the constraint matrix M are linearly independent, then there exist an infinite number of stabilizing solutions for any SPD matrices Q_1, Q_2. Note that there is no dependence on the eigenvalues of (E, A). In fact, there exist stabilizing solutions ***even if the original large-order system is unstable***. Thus, not only can we preserve stability, but we also have the option of enforcing stability when needed.

We would like to point out here that creating stable reduced models of unstable original systems may in general result in inaccurate reduced systems. Ideally the large-order extracted systems would be stable and the projection framework would simply preserve stability, but unfortunately this is not the case in the vast majority (if not complete totality) of field solvers and parasitic extractors used in current VLSI design flows to extract the original large order systems. We have found that in most cases the numerically unstable modes in such extracted linear systems correspond to very small eigenvalues and are typically not excited by inputs of interest. Explicitly truncating, or stabilizing, these unstable modes in the large-order system would therefore likely yield accurate stable models. However, this operation is computationally infeasible since it would require the preliminary eigendecomposition of the original large order system. Nevertheless, unstable modes must be eliminated, because a poorly-chosen projection could amplify the unstable modes and cause major instabilities in the reduced model. The approach presented in this paper may be used to eliminate such parasitic-extraction-generated artificial numerical instabilities, avoiding the expensive eigendecomposition of the original large-order system.

A second important observation is that not only are there an infinite number of stabilizing solutions U to the underdetermined System (16), but there are also an infinite number of subspaces spanned by the set of stabilizing solutions U. That is, if we define $T = \text{range}(U)$ to be the space spanned by the columns of U, then there exist an infinite number of stabilizing spaces T. This is an important distinction to make, as there exist many different representations for a single subspace which will produce reduced models all identical up to an invertible change of coordinates.

If the constraint matrix M in (16) has linearly dependent columns and the original system is stable, then there exist stabilizing solutions only for certain SPD matrices Q_1, Q_2.

Prop. 3.2 (Existence of Stabilizing Projection): If (E, A) is an asymptotically stable pair, then for any orthogonal right-projection matrix V and any SPD matrix Q_2, there exists at least one SPD matrix Q_1 such that the linear constraints (14) *and* (15) *are consistent (i.e. a solution exists).*

Proof: By assumption, (E, A) is an asymptotically stable pair, and therefore given any SPD matrix $\tilde{Q}$ there exists an SPD matrix P such that

$$E^T PA + A^T PE = -\tilde{Q}. \tag{17}$$

Furthermore, given any right-projection matrix V, it is also true that

$$V^T E^T PAV + V^T A^T PEV = -V^T \tilde{Q} V, \tag{18}$$

and that $V^T \tilde{Q} V$ is also SPD because congruence transforms preserve definiteness. Thus, we select $U = PEV$ and $Q_2 = V^T \tilde{Q} V$ to satisfy constraint (15), and find that $U^T EV = V^T E^T PEV \succ 0$ satisfies constraint (14). Lastly, we need to show that there exists such a $\tilde{Q}$ for any Q_2. Since Q_2 is SPD, we may factor it such that $q^T q = Q_2$, where $q \in \mathbb{R}^{q \times q}$. Now define $\tilde{q}^T = [X, \ \mathcal{N}(V)]$ where $\mathcal{N}(V)$ is the null space of V^T, and $X \in \mathbb{R}^{N \times q}$ is the solution to $X^T V = q$ (which exists because it is an underdetermined system of equations and V has linearly independent columns by assumption). Finally, define $\tilde{Q} = \tilde{q}^T \tilde{q}$, resulting in $Q_2 = V^T \tilde{Q} V$. ■

If the constraint matrix M has linearly dependent columns and the original system (E, A) is **not** stable, then there may or may not exist a stabilizing projection. It is possible to construct systems where there does not exist a stabilizing projection, e.g. take $E = A = I$ as identity matrices. However, if the number of unstable modes is smaller than the order of the reduced model, then there always exists a stabilizing projection pair U, V that is equivalent to truncating the unstable modes.

The unstable systems we are trying to address here are the ones generated by parasitic extractors discretizing PDEs, hence they typically contain a very small number of numerically generated positive eigenvalues. For those systems we are often able to find stabilizing projections for unstable models. We will consider this problem in more detail in section IV-B.

C. Optimal Stabilizing Projection

Given the large number of stabilizing left projection matrices U, it might be beneficial to spend more computation in an attempt to increase the reduced model accuracy by searching for a stable *and* accurate projection. Consider Figure 1, which compares six different reduced models, all chosen to match exactly two moments at each of the frequencies $0.1, 1$ and 10 GHz, resulting in a sixth-order model. All models match exactly the same moments at the expansion points, but some models are more accurate than others far away from the expansion points as a result of the different left-projection matrices.

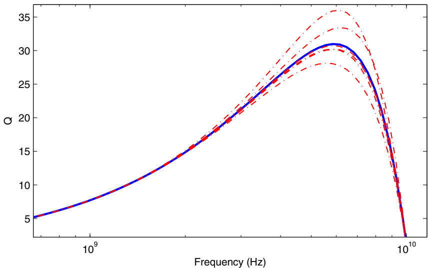

Fig. 1. Six different sixth-order reduced models (dashed red lines) created with the same right-projection matrix V matching moments at $0.1, 1, 10$ GHz and four different left-projection matrices U. The solid blue line indicates the response of the original transmission line model. The curves represent quality factors of an RF inductor model.

Given a matrix U_0 that was constructed to ensure accuracy, we wish to find a stabilizing projection U that is "close" to U_0 in some sense. Formally speaking, given U_0, the optimal solution U solves the optimization problem

$$\begin{aligned} \min_{U,Q_1,Q_2} & \; \|U - U_0\|_s \\ & U^T EV = Q_1 \succ 0 \\ & U^T AV + V^T A^T U = -Q_2 \prec 0 \end{aligned} \tag{19}$$

In this setup our measure of optimality $\|\cdot\|_s$ should quantify the distance between the subspaces spanned by U and U_0. There are many possible norms one can use here to measure this distance. One possible choice is to use the familiar and more computationally affordable L_2 norm. The problem with this choice is that it does not necessarily measure the distance between the subspaces, as $\|U - U_0\|$ could be large while $\|U - \tilde{U}_0\|$ could be small when U and $\tilde{U}$ have the same column span. It is important to note that although optimization problem (19) is likely to produce an accurate stabilizing projection, the optimization procedure is extremely expensive because the number of variables is order Nq, and the cost of LMI solutions is quadratic in the number of decision variables. An efficient suboptimal problem will be introduced in Section IV-C as an alternative approach.

D. Extension to Passivity

Until now we have only considered preserving (or creating) stability. In general, passivity is a more desirable system property, since besides implying stability it also guarantees the system will not produce energy.

If, as was done for stability, we consider the storage function $\hat{L}(x,u) = z^T \hat{E} z$, then the passivity condition for the reduced model becomes

$$\begin{aligned} \hat{A} + \hat{A}^T &= -Q_2 \prec 0 \\ \hat{B} &= \hat{C} \end{aligned}$$

provided $\hat{E} \succ 0$. Since $\hat{B} \neq \hat{C}$ in general even if $B = C$, because $\hat{B} = U^T B$ while $\hat{C} = V^T C$, we must enforce this as an additional constraint

$$U^T E V = Q_1 \succ 0 \tag{20}$$

$$U^T A V + V^T A^T U = -Q_2 \prec 0 \tag{21}$$

$$U^T B = V^T C. \tag{22}$$

Thus, passivity is preserved by adding one additional linear constraint. In the remainder of the paper, in order to keep the explanation simple, we will refer only to stability and consider only the first two linear constraints; however, ***it is always possible and a trivial extension to add the third linear constraint to ensure passivity when needed.***

IV. Efficient Computation of Stabilizing Projections

A. Analytic Solutions for Independent Constraints

Now that we know stabilizing solutions exist, we wish to efficiently compute such solutions U, where U satisifes

$$U^T V_e = Q_1 \succ 0, \qquad U^T V_a + V_a^T U = -Q_2 \prec 0, \tag{23}$$

and we have defined $V_e = EV$ and $V_a = AV$. It is possible to solve this system directly using Kronecker identities to transform the matrix equation into one large linear system, but this would be very expensive with a computational cost on the order of $(Nq)^3$, where N and q are the sizes of the large-order and reduced-order systems respectively. Our goal is to eliminate dependence of the complexity on N.

As stated previously, we can transform the matrix equation into a linear system. Consider factoring the matrix V_a into two matrices Q_a and R_a such that $V_a = Q_a R_a$, where $Q_a \in \mathbb{R}^{N\times q}$ and $R_a \in \mathbb{R}^{q\times q}$. One approach is to use a modified Graham Schmidt orthogonalization for Q_a. This is cheap computationally because only the first q orthogonal columns are needed. For the sake of consistency we may do the same for the other constraint, factoring V_e into Q_e and R_e. We now define a new pair of variables $Y_a = U^T Q_a$ and $Y_e = U^T Q_e$. The result is the linear system

$$[Q_e \; Q_a]^T U = [Y_e \; Y_a]^T \tag{24}$$

where $Y_e, Y_a \in \mathbb{R}^{q\times q}$ are the solutions to the order q matrix equations

$$R_e^T Y_e = Q_1, \qquad R_a^T Y_a + Y_a^T R_a = -Q_2,$$

which can be solved in at worst $O(q^3)$ operations. Note that since V_e and V_a are linearly independent, then Q_e and Q_a are linearly independent, $[Q_e \; Q_a]^T$ has $2q$ linearly independent columns, and thus there always exists a solution to linear system (24). This procedure is summarized in Algorithm 1, and the resulting underdetermined system can be solved efficiently using any standard technique.

Algorithm 1 Linear Constraint Transformation

```
Given matrices $Q_1, Q_2 \succ 0$, and $E, A, V$
Define $V_e = EV$, $V_a = AV$, and then factor into matrices
    $Q_e, Q_a \in \mathbb{R}^{N\times q}$, $R_e, R_a \in \mathbb{R}^{q\times q}$ such that
        $V_e = Q_e R_e \qquad V_a = Q_a R_a$
Solve order $q$ matrix equations separately for $Y_e$ and $Y_a$
        $Y_e^T R_e = Q_1 \qquad Y_a R_a + R_a^T Y_a = -Q_2$
Solve underdetermined linear system for $U$
        $[Q_e \; Q_a]^T U = [Y_e \; Y_a]^T$
```

B. Efficient Solutions for Dependent Constraints

The more interesting case is when the constraint matrix is either redundant or extremely ill-conditioned. In this case there only exist stabilizing solutions U for certain right-hand-sides Q_1, Q_2. This is a more difficult problem because we also must enforce the positivity constraints on Q_1, Q_2.

Since, from Proposition 3.2, we know there exist stabilizing solutions for any SPD matrices Q_1, Q_2, we may formulate the linear system as a feasibility problem

$$\begin{aligned} \min_{U,Q_1,Q_2} \quad & 0 \\ & U^T V_e = Q_1 \succ 0 \\ & U^T V_a + V_a^T U = -Q_2 \prec 0. \end{aligned} \tag{25}$$

Here $\min 0$ simply searches for a triplet (U, Q_1, Q_2) which satisfies all of the constraints. If desired, we may also fix Q_2 and only vary the constraint Q_1, as this does not affect the existence of the solution (shown in Proposition 3.2). Feasibility problem (25) can be solved efficiently using an LMI solver. Note that for the feasibility problem we can leave the second constraint in matrix equation form.

As stated, the feasibility problem is still computationally N-dependent because U has size $N \times q$. Since the system is underdetermined, we may fix $N - 2q$ variables and solve instead a size $2q$ square system. Consider partitioning U, V_a, and V_e such that U_q, V_{aq}, and V_{eq} are the first $2q$ rows of each respectively. Now the size $N \times q$ system

$$\begin{bmatrix} V_{eq}^T & V_{e2}^T \end{bmatrix} \begin{bmatrix} U_q \\ U_2 \end{bmatrix} = Q_1$$

is equivalent to the size $2q \times 2q$ system

$$V_{eq}^T U_q = Q_1 - (V_{e2}^T U_2),$$

where U_2 is a predetermined size $(N - 2q) \times q$ matrix. We may partition the remaining constraint in a similar manner, resulting in a pair of square size $2q$ constraints and a new feasibility problem

$$\begin{aligned} \min_{U,Q_1\succ 0,Q_2\succ 0} \quad & 0 \\ & U_q^T V_{eq} = Q_1 - \Delta Q_1 \\ & U_q^T V_{aq} + V_{aq}^T U_q = -Q_2 - \Delta Q_2 \end{aligned} \tag{26}$$

where we have defined $\Delta Q_1 = V_{e2}^T U_2$ and $\Delta Q_2 = (V_{a2}^T U_2 + U_2^T V_{a2})$. This new problem has $O(q^2)$ constraints and unknowns, resulting in a cost of $O(q^4)$. A feasible solution to the original problem can then be chosen as $U = [U_q^T \; U_2^T]^T$. Thus we may now compute the stabilizing solution by solving an order-q LMI that has ***no dependence on N***, the order of the original system.

There remains one important detail not to be overlooked. We are assuming that the new order $2q$ system has a solution, but this is not necessarily true for all V_{aq}, V_{eq}. For example, it is possible to choose $2q$ rows from V_a and V_e such that V_{aq} and V_{eq} are all zeroes, resulting in no solution. However, by simply selecting the $2q$ rows such that there are no zero rows in either truncated matrix, we have not found this to be a problem in practice. It is also possible to increase the number of rows in U_q to some number p such that $2q < p < N$ if a stabilizing solution is not found when considering only $2q$ rows.

C. Efficient Suboptimal Solutions

We finally consider the problem of efficient computation of optimal stabilizing projections. That is, we present a method for efficiently solving the optimization problem posed in Section III-C. One possible approach is to combine the ideas in Sections III-C and IV-B. Specifically, we will fix some of the elements of U and optimize over a smaller block of unknowns.

Consider defining $U = U_0 + \Delta U$, where U_0 is a left-projection matrix chosen for accuracy and ΔU is the matrix to be solved for. We define U_δ as p rows of ΔU, where $2q \leq p \leq N$, and set the remaining $N - p$ rows to zero. Similarly, we define V_{ep} and V_{ap} as the same p rows of V_e and V_a respectively. Finally, defining the new terms ΔQ_1 and ΔQ_2 as

$$\Delta Q_1 = V_e^T U_0 \qquad \Delta Q_2 = V_a^T U_0 + U_0^T V_a \tag{27}$$

yields the following modified constraints

$$V_{ep}^T U_\delta = Q_1 - \Delta Q_1 \tag{28}$$

$$V_{ap}^T U_\delta + U_\delta^T V_{ap} = -Q_2 - \Delta Q_2. \tag{29}$$

We may now attempt to solve the optimization problem $U_\delta = \operatorname{argmin} ||U_\delta||$ over constraints (28), (29) and $Q_1, Q_2 \succ 0$. Note that if we consider the possibility that the original projection U_0 does preserve stability, then we must change the constraints on Q_1, Q_2 from SPD to SPSD, to allow for the possibility that $U_\delta = 0$. A sample procedure is presented in Algorithm 2.

What we are essentially doing here is searching for a minimal perturbation to the matrix U_0, which was constructed for accuracy, such that the resulting matrix $U = U_0 + \Delta U$ is accurate and guarantees stability. The cost of this algorithm is $p^2 q^2$ due to the LMI solve. Thus if $p = 2q$, the total cost will be order q^4. The larger p is, the more accurate the solution will be. If we allow p to be order $\sqrt{N}$, the total cost will be order Nq^2. Typically $q^2 << N$. We have found this approach to be particularly effective when considering unstable extracted models of RF components that should be stabilizable via a Galerkin projection ($U_0 = V$).

V. Examples

A. Implementation

The proposed methods were tested on several linear system examples. All of the following results were generated using Algorithm 2, the LMIs were solved using SeDuMi [21] with YALMIP [22], and we have chosen $U_0 = V$, where V was constructed to match input moments.

Algorithm 2 Efficient Suboptimal Stabilizing Projection

```
1: Given $E, A, V, U_0$, define
       $V_e = EV, \qquad V_a = AV,$
       $\Delta Q_1 = V_e^T U_0, \qquad \Delta Q_2 = V_a^T U_0 + U_0^T V_a$
2: Select $p$ nonzero rows from $V_e$ and $V_a$ and denote them $V_{ep}$ and $V_{ap}$ respectively
3: Solve the optimization problem for $U_\delta$
       $\min_{U_\delta, Q_1 \succ 0, Q_2 \succ 0} \; ||U_\delta||$
           $U_\delta^T V_{ep} = Q_1 - \Delta Q_1$
           $U_\delta^T V_{ap} + V_{ap}^T U_\delta = -Q_2 - \Delta Q_2.$
4: Define $U = U_0 + \Delta U$, where $\Delta U$ is zeroes except for the $p$ rows corresponding to the rows of $U_\delta$
```

When dealing with large q, it is possible to reduce the total number of decision variables in the LMI from $6q^2$ to $2q^2$ by fixing Q_2 and solving for half of the unknowns analytically. That is possible because although the entire constraint system is linearly dependent, neither the matrix V_e nor the matrix V_a will have linearly dependent columns, provided the matrices E and A are nonsingular.

As a last note, we have found that due to the extreme ill-conditioning of the constraint system, one must normalize the constraints such that V_e and V_a are of the same order. This is equivalent to simply scaling the right hand side matrices Q_1, Q_2. This operation is allowed because the matrices would still be SPD.

B. Inductor Model

The first examples considered are two-turn RF inductor models. The inductor models were created using a public domain electro-magneto-quasi-static (EMQS) mixed-potential integral equation solver [23], [24]. All of the selected extracted large order models are numerically *unstable*, despite modeling passive physical systems.

As a result of instability, the traditional Galerkin projection $U = V$ is not guaranteed to create a stable model. Results from several different inductor models and stable reduced models generated using Algorithm 2 are plotted in Figure 2. All reduced models have order $q = 8$, and were each constructed in under 2 seconds. The original large order systems all have order $N = 647$ and are all unstable.

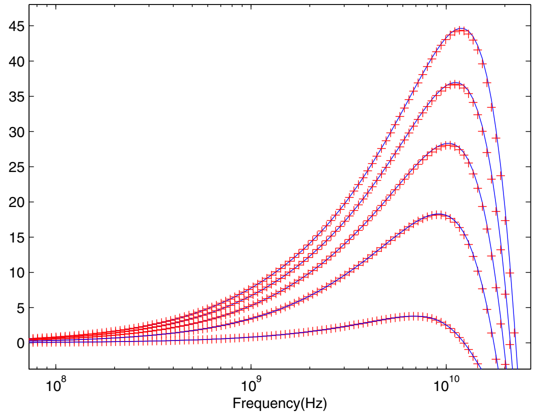

Fig. 2. Quality factor of two-turn inductor models corresponding to various wire width and wire separation parameter values. The original models (blue solid lines) are unstable and all have order $N = 647$, while the reduced models (red plusses) are stable and all have order $q = 8$.

For these inductor models we cannot compare the computational time required to compute the stabilizing projection to other similar methods, because to our knowledge no other available method is guaranteed to create a stabilizing left-projection matrix U for unstable systems when given a right-projection matrix V. However, it is possible to first stabilize the large-order model, e.g. truncate the unstable modes in the large model via eigendecomposition, and then compute a stabilizing left-projection matrix corresponding to the given right-projection matrix V. Doing so by solving a Lyapunov equation for the large model was 10 times slower than our proposed method for this example, and this does not include the time required to stabilize the large-order model via eigendecomposition. When factoring in the cost of the eigendecomposition, our proposed method is 15 times faster.

C. Power Grid

The second example is a 3×3 power grid in free space. The grid consists of copper wires with width $2\mu m$ and thickness $2\mu m$, and resulted in an order $N = 1566$ unstable model, created using the same EMQS solver . Using Algorithm 2 we were able to create an order $q = 10$ stable reduced model by constructing V to match moments and selecting $U_0 = V$. For this example the Galerkin projection $U = V$ resulted in an unstable model. Figure 3 plots the real part of the impedance of the original system, the stabilized reduced model, and an unstable reduced model created using a Galerkin projection.

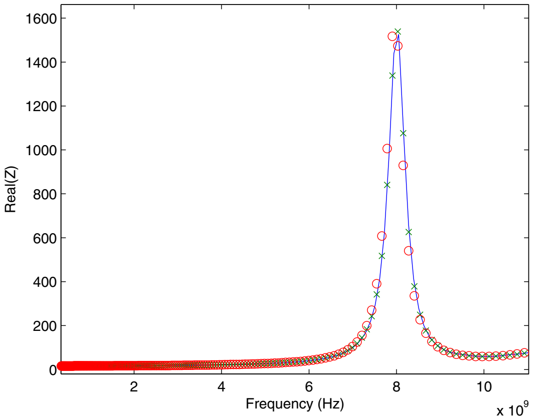

Fig. 3. Real part of the impedance for a 3×3 power grid. The original model (solid blue line) is unstable and has order $N = 1566$. An unstable reduced model created through Galerkin projection (green crosses) of order $q = 10$ is comparable in accuracy to a stable reduced model (red circles) of order $q = 10$ created by our proposed method.

As a reminder we would like to point out that although the frequency-domain results of the unstable reduced models in Figure 3 look nice, it is possible that certain inputs, or simply numerical noise in the simulator, could excite the unstable modes in the model resulting in unphysical behavior. For instance a non-zero initial condition could cause the solution to explode, which cannot happen for a passive system such as a power grid.

D. Linearized Systems

The final example considered is a nonlinear Microelectromechanical System (MEMS) switch [25]. The need to efficiently construct stabilizing projections for indefinite linearized models arises in small-signal analysis of analog circuits, and when using nonlinear projection functions to create stable nonlinear reduced models of nonlinear systems [7]. The nonlinear model is obtained from the discretization of a pair of nonlinear PDEs, and the linear model is obtained by linearizing at the equilibrium state. For this system the matrices are neither symmetric nor definite. The original model has order $N = 1680$, and we were able to create an accurate stable reduced model with order $q = 12$.

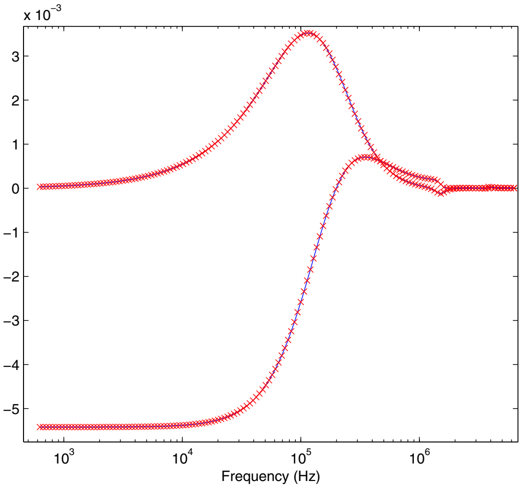

Fig. 4. Real and imaginary transfer function parts of a linearized model of a nonlinear MEMS device. The original stable model (solid blue line) has order $N = 1680$, while the reduced stable model (red crosses) has order $q = 12$. A Galerkin projection on this indefinite system does not preserve stability.

For this example a Galerkin projection does not preserve stability, and constructing a stability-preserving projection by solving a Lyapunov equation for the full system was 10 times slower than our proposed algorithm.

VI. Conclusion

In this work we have presented a new technique for efficiently generating stable reduced models from indefinite or unstable systems. This was achieved by formulating a system of linear matrix inequalities whose solution is a stabilizing projection matrix. Typically we have chosen the right-projection matrix to guarantee accuracy by matching moments, and solved for a left-projection matrix which ensures stability, enabling the proposed algorithms to integrate smoothly with an existing moment-matching framework. With the addition of one extra linear constraint the proposed method can also guarantee passivity. Additionally, we have proposed in Algorithm 2 a method for generating stabilizing projection matrices with a computational complexity dependent only on the order of the reduced-order system, and completely independent of the size of the original large-order system. Our proposed method has also been shown to generate stable reduced models reliably from unstable models that describe stable physical systems, and creates reduced models that are as accurate as models created with traditional projection approaches such as Galerkin projection.

References

[1] Eric Grimme. *Krylov Projection Methods for Model Reduction*. PhD thesis, Coordinated-Science Laboratory, University of Illinois at Urbana-Champaign, Urbana-Champaign, IL, 1997.

[2] Zhaojun Bai, Peter Feldmann, and Roland W. Freund. Stable and passive reduced order models based on partial pade approximation via the lanczos process. Technical Report Numerical Analysis Manuscript No.97-3-10, Bell Laboratories, Lucent Technologies, Murray Hill, New Jersey, October 1997.

[3] A. Odabasioglu, M. Celik, and L. T. Pileggi. PRIMA: passive reduced-order interconnect macromodeling algorithm. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 17(8):645–654, August 1998.

[4] K. J. Kerns, I. L. Wemple, and A. T. Yang. Stable and efficient reduction of substrate model networks using congruence transforms. In *Proc. of the IEEE/ACM International Conference on Computer-Aided Design*, pages 207 – 214, San Jose, CA, November 1995.

[5] L. M. Silveira, M. Kamon, I. Elfadel, and J. K. White. Coordinate-transformed arnoldi algorithm for generating guarantee stable reduced-order models of RLC. *Computer Methods in Applied Mechanics and Engineering*, 169(3-4):377–89, February 1999.

[6] Freund. Krylov-subspace methods for reduced-order modeling in circuit simulation. In *J. Comput. Appl. Math*, volume 123, pages 395–421, 2000.

[7] B. Bond and L. Daniel. Stabilizing schemes for piecewise-linear reduced order models via projection and weighting functions. In *Proc. of the IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA, 2007.

[8] Bruce Moore. Principal Component Analysis in Linear Systems: Controllability, Observability, and Model Reduction. *IEEE Transactions on Automatic Control*, AC-26(1):17–32, February 1981.

[9] J. R. Phillips, L. Daniel, and M. Silveira. Guaranteed passive balancing transformations for model order reduction. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 22(8):1027–41, Aug 2003.

[10]

[11] M. Vidyagasar. *Nonlinear Systems Analysis*. Prentice Hall, New York, 1978.

[12] S. Sastry. *Nonlinear Systems: Analysis, Stability, and Control*. Springer, New York, 1999.

[13] T. Stykel. Stability and inertia theorems for generalized lyapunov equations. *Linear Algebra and its Applications, volume=*.

[14] J. C. Willems. Dissipative dynamical systems. *Arch. Rational Mechanics and Analysis*, 45:321–393, 1972.

[15] Anders Rantzer. On the Kalman-Yakubovich-Popov lemma. *Systems and Control Letters*, 27(5), January 1996.

[16] R. Freund and F. Jarre. An extension of the positive real lemma to descriptor systems. Technical Report Numerical Analysis Manuscript No.00-3-09, Bell Laboratories, Lucent Technologies, Murray Hill, New Jersey, December 2000.

[17] P. Feldmann and R. W. Freund. Efficient linear circuit analysis by Padé approximation via the Lanczos process. In *Proceedings of EURO-DAC '94 with EURO-VHDL '94, Grenoble, France*, pages 170–175. IEEE Computer Society Press, 1994.

[18] R. W. Freund and P. Feldmann. The SyMPVL algorithm and its applications to interconnect simulation. In *Proceedings of the 1997 International Conference on Simulation of Semiconductor Processes and Devices*, pages 113–116. IEEE, 1997.

[19] K. Willcox and J. Peraire. Balanced model reduction via the proper orthogonal decomposition. In *Proceedings of the 15th AIAA Computational Fluid Dynamics Conference*, Anaheim, CA, June 2001.

[20] Joel Phillips and L. Miguel Silveira. Poor man's tbr: A simple model reduction scheme. In *DATE '04: Proceedings of the conference on Design, automation and test in Europe*, page 20938, Washington, DC, USA, 2004. IEEE Computer Society.

[21] J.F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11–12:625–653, 1999.

[22] J. Lfberg. Yalmip : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[23] Tarek Moselhy, Xin Hu, and Luca Daniel. pfft in fastmaxwell: a fast impedance extraction solver for 3d conductor structures over substrate. In *DATE '07: Proceedings of the conference on Design, automation and test in Europe*, pages 1194–1199, San Jose, CA, USA, 2007. EDA Consortium.

[24] Fastmaxwell. *http://www.rle.mit.edu/cpg/fastmaxwell*.

[25] S.D. Senturia E. Hung, Y.-J. Yahg. Low-order models for fast dynamical simulation of mems microstructures. In *Proceedings of IEEE International Conference on Solid State Sensors and Actuators*, volume 2, pages 1101–1104, 1997.