Model Boundary Approximation Method as a Unifying Framework for Balanced Truncation and Singular Perturbation Approximation

Philip E. Paré, David Grimsman, Alma T. Wilson, Mark K. Transtrum and Sean Warnick

Abstract—We show that two widely accepted model reduction techniques, Balanced Truncation and Balanced Singular Perturbation Approximation, can be derived as limiting approximations of a carefully constructed parameterization of Linear Time Invariant (LTI) systems by employing the Model Boundary Approximation Method (MBAM), a recent development in the Physics literature [1]. This unifying framework of these popular model reduction techniques shows that Balanced Truncation and Balanced Singular Perturbation Approximation each correspond to a particular boundary point on a manifold, the “model manifold,” which is associated with the specific choice of model parameterization and initial condition, and is embedded in a sample space of measured outputs, which can be chosen arbitrarily, provided that the number of samples exceeds the number of parameters. We also show that MBAM provides a novel way to interpolate between Balanced Truncation and Balanced Singular Perturbation Approximation, by exploring the set of approximations on the boundary of the manifold between the elements that correspond to the two model reduction techniques; this allows for alternative approximations of a given system to be found that may be better under certain conditions. The work herein suggests similar types of approximations may be obtainable in topologically similar places (i.e. on certain boundaries) on the model manifold of nonlinear systems if analogous parameterizations can be achieved, therefore extending these widely accepted model reduction techniques to nonlinear systems.

I. INTRODUCTION

Modern systems theorists are studying and engineering systems that are larger and more complex than ever before [2]. Typical examples of these complex systems include economic networks [3], biological systems [4]–[6], and the Internet [7]. Due to the unprecedented size of these systems, simplified models are necessary to reason about them effectively [8], [9]. Specifically, we detail four important motivations for building simplified approximations of large scale complicated systems, and then provide an overview of how this work relates singular perturbation and balanced truncated methods.

Philip E. Paré is at KTH Royal Institute of Technology and can be reached at philip.e.pare@gmail.com. David Grimsman is at the University of California at Santa Barbara and can be reached at davidgrimsman@gmail.com, Alma T. Wilson, Mark K. Transtrum and Sean C. Warnick are at Brigham Young University, Provo, Utah, and can be reached at alba_wilzod@yahoo.com, mktranstrum@byu.edu, and sean.warnick@gmail.com, respectively. This material is based on research sponsored by the Department of Homeland Security (DHS) Science and Technology Directorate, Homeland Security Advanced Research Projects Agency (HSARPA), Cyber Security Division (DHS S&T/HSARPA/CSD), LRBAA 12-07 via contract number HSHQDC-13-C-B0052. All material in this paper represents the position of the authors and not necessarily that of DHS.

A. Motivation

When attempting to learn a system from limited data, large models cannot be identified, making simplified models necessary. First principles models typically have many parameters that must be tuned correctly for the model to reflect the behavior of a real system. Examples are everywhere, from agronomy and biochemical reaction networks to swarms of autonomous flying robots or power systems. Using data to learn the correct values of parameters is the purview of system identification, and a rich theory has developed quantifying when data is informative enough to accurately estimate parameter values [10]–[15]. Typically, however, there is much less data than needed to learn all the parameters in a first-principles model, so simplifying the model to yield one with fewer parameters can help identify the system from data.

Second, the need for simplified models arises when designing controllers for complex systems. The complexity of an optimal controller often mirrors that of the system being controlled; therefore a complex system may often suggest the need for complicated controllers. Nevertheless, when engineering such complicated systems is unreasonable, designing controllers for simplified approximations can lead to acceptable trade-offs between complexity and performance degradation.

Third, simplified models can be an important link between macro-scale and micro-scale models. Generative models, for example, often detail micro-scale phenomena, such as consumer-choice models or models of a single neuron or molecular organization, etc., and then hypothesize that macro-scale behavior, such as consumer demand or regions of the brain or material properties, etc., is the aggregation of a large number of micro-scale instances [14]–[16]. The complexity of modeling a macro-scale model composed of thousands or millions of micro-scale instances, however, can be unwieldy, and such models almost never exhibit behavioral complexity commensurate with the descriptive complexity of the model as an aggregate of many micro-scale models. Thus, a systematic technique for developing simplified descriptions of macro-scale models that resemble, in a principled way, the aggregate of micro-scale phenomena can be critical in such applications.

Fourth, understanding the resilience and vulnerability of large-scale critical infrastructures demands techniques for modeling the attack surface of complex cyber-physical-human systems. The attack surface is typically a simplified model of the system that highlights the exposed variables and dynamics strongly affected by or observed from them [17]–[22]. Thus,
the techniques discussed here offer the possibility of modeling
the attack surface of large-scale cyber-physical-human systems
and contribute to a science of system security.

B. Overview

Two important model reduction techniques for linear sys-
tems include Balanced Truncation and singular perturbation
methods. Each of these approaches focuses on a particular
aspect of the system to preserve.

Balanced Truncation (BT) was first proposed in [23] and has
been explored for continuous and discrete time [24]. A clear
presentation is provided in [25]. The basic idea is that a change
of basis is used to order the states equally observable and
controllable, and to order them from most to least controllable
and observable, and then the least controllable and observable
states are truncated; see [9] for a survey of projection-based
reduction techniques, including balanced truncation, for pa-
rameterized dynamical systems. A complete explanation of BT
is presented in Section II-A.

Perturbation theory is a well studied area and has a rich
background in linear operator theory and the controls field
[26]–[28], and it has seen renewed interest in recent years
[29]–[31]. It is commonly applied in the context of well-
separated time scales. In this case the ratio of time-scales
identifies an explicit “small” parameter in which a series
expansion can be computed. The theory has also been applied
to balanced realizations, which we will refer to as Balanced
Singular Perturbation Approximation (BSPA) [32]–[35]. In
[33], an alternative parameterization of the transfer function
is used to provide a different unification of BT and BSPA using
the “generalized singular perturbational approximation.” This
parameterization is quite different than the one proposed here
and does not lead to the insight that BT and BSPA occur on
the boundary of a particular manifold representing the original
model class. A complete explanation of BSPA is presented in
Section II-B. The Manifold Boundary Approximation Method,
introduced in [1], is an algorithm that, given a parameterized
model class, systematically generates a “representative” subset
of the class parameterized by fewer parameters than the
original set of models. The process is enabled by the fact
that a parameterized model can be interpreted as a mapping
between a parameter space and prediction space. The set of
all possible models generated by varying the parameters over
all values typically corresponds to a manifold with the model
mapping acting as a coordinate chart. To make this abstract
idea more concrete, consider a model class with \( N \) parameters
that is sampled at \( M \) different points (times or frequencies). In
this case, the model defines a mapping from \( \mathbb{R}^N \) to \( \mathbb{R}^M \) whose
image is an \( N \) dimensional submanifold of \( \mathbb{R}^M \). We illustrate
this conceptually for several simple models in this manuscript.
Each point on the manifold corresponds to the sample of a
different set of parameters. For practical problems, explicitly
constructing the entire manifold would be impractical, but such
a manifold can be explored using computational differential
geometry. Systematic study of model manifolds from a variety
of classes has revealed that they are typically bounded and
that the boundary consists of a hierarchical cell complex,
i.e., a hierarchy of faces, edges, corners, etc., similar to a
high-dimensional polygon. Each boundary cell is a manifold
that corresponds to a model class with fewer parameters. The
“best” approximation of this model can be found by locating
the closest boundary point with the desired level of complexity,
where closest depends on your metric of choice. MBAM will
be formalized in Section III.

In this paper we explore the problem of system approxi-
imation and identify an intrinsic structure to the problem by
proposing an appropriate parameterization for LTI systems that
enables us to show that BT and BSPA can be derived by
applying MBAM to it. Given this insight, and since MBAM
does not depend on linearity, only on a given parameterization,
we conjecture that MBAM may provide a framework for the
systematic model reduction of more general (nonlinear) sys-
tems. In [36], Andersson et al. provide bounds for model
reduction on systems that include uncertainties or nonlinear-
ities, modeled as a delta block, assuming the delta block can
be modeled using integral quadratic constraints (IQC). The
authors also show that BT and BSPA appear as natural limits
of the proposed IQC model, but do not provide any insight as
to how to interpolate between the two. While the model in [36]
is more general than the model considered herein, we believe
that the ideas proposed in this work can be applied to a broader
classes of models, independent of the IQC assumption.

In Section II we review Balanced Truncation and Balanced
Singular Perturbation Approximation. We present MBAM in
Section III. In Section IV we present a parameterization of
LTI systems, and then in Section V we use it to derive BT
and BSPA using MBAM. In Section VI we present a simple
example that illustrates the result and how MBAM can give
insight into model reduction of linear systems, allowing a way
to interpolate between BT and BSPA. A condensed version of
the work is given in [29]. Further contributions of this paper
include: 1) the inclusion of the complete proofs; 2) the addition
of several illustrative examples; 3) a clearer, more in-depth
presentation of MBAM; and 4) an extensive discussion of the
parameterization of LTI systems.

II. MODEL REDUCTION OF LTI SYSTEMS

In this work we consider LTI systems described by,
\[
\dot{x}(t) = Ax(t) + Bu(t)
\]
\[
y(t) = Cx(t) + Du(t)
\]
where \( x(t) \in \mathbb{R}^n, y(t) \in \mathbb{R}^p, \) and \( u(t) \in \mathbb{R}^m \). We will assume
the system described by (1) is minimal and stable, and note
that it generates a unique transfer function given by:
\[
G(s) = (sI - A)^{-1}B + D
\]
which is a \( p \times m \) matrix of rational functions in the complex
variable \( s \in \mathbb{C} \). Note that while (1) uniquely specifies (2),
there are many equivalent realizations of the form (1) for any
given input-output map (2).

Model reduction is an important, well–studied problem.
Although there are many approaches to model reduction, this
work focuses on two types of model reduction for LTI systems:
BT and BSPA. We will review these two methods and their
accompanying literature.
A. Balanced Truncation

Consider a stable system given by a bounded map (2), and let some (arbitrary) minimal realization of it, (1), be known. In general, the description (1) may not reveal any particular insight about the system, compared with any other realization of the system, and all such minimal realizations can be found by changing coordinates of the state variable, i.e. by rewriting (1) in terms of a new basis of the state space obtained by multiplying \( x \) by any invertible matrix, \( \bar{x} = Tx \), and considering the system in terms of \( \bar{x} \); we call \( T \) a state transformation.

Nevertheless, while an arbitrary realization may be no more useful than any other, some particular realizations are more useful for particular types of analysis than others. For example, modal realizations change basis of the state space to reveal the eigenstructure of each dynamic mode of the system. Any such “particular” realization is called \textit{canonical} when there is a well-defined process for constructing it, i.e. when such a realization always exists, and when it is unique for every system \( G \). Thus, a canonical realization is in one-to-one correspondence with any given input-output map (2).

It turns out that there is a canonical realization of LTI systems that has a property we call \textit{balanced}. That is to say, although there may be multiple realizations with the property of being balanced, we can specify a fixed procedure that will repeatedly construct such a realization for any system \( G \). We will represent this realization by:

\[
\begin{align*}
\dot{\bar{x}}(t) &= \bar{A}\bar{x}(t) + \bar{B}u(t) \\
y(t) &= \bar{C}\bar{x}(t) + \bar{D}u(t)
\end{align*}
\]

where \( \bar{A} \) satisfies the Lyapunov equations:

\[
\bar{A}^T X + X \bar{A} = -\bar{C}^T \bar{C} \quad \text{and} \quad \bar{A}X + X \bar{A}^T = -\bar{B}\bar{B}^T
\]

with \( X = \text{diag}(\theta_1, \ldots, \theta_n) \) and the \( \theta_i \)'s are the Hankel singular values (HSV's) of the system [25]. These Lyapunov equations are solved by the controllability and observability Gramians; since they are solved by the same diagonal matrix, \( X \), we call the realization \textit{balanced}, that is, each state in this basis is as observable as it is controllable, quantified by the corresponding \( \theta_i \). The HSVs are ordered from largest to the smallest in magnitude, and consequently the states are ordered from the most controllable/observable to the least.

Once balanced, one may partition the states into two sets, a highly controllable/observable set and a low controllable/observable set. Partitioning the state matrices commensurate with this partition of the states then yields:

\[
\begin{align*}
\bar{A} &= \begin{bmatrix} \bar{A}_{11} & \bar{A}_{12} \\ \bar{A}_{21} & \bar{A}_{22} \end{bmatrix}, \\
\bar{B} &= \begin{bmatrix} \bar{B}_1 \\ \bar{B}_2 \end{bmatrix}, \\
\bar{C} &= \begin{bmatrix} \bar{C}_1 \\ \bar{C}_2 \end{bmatrix}, \quad \text{and} \\
\bar{x} &= \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix}
\end{align*}
\]

where \( \bar{A}_{11} \in \mathbb{R}^{(n-k) \times (n-k)}, \bar{A}_{22} \in \mathbb{R}^{k \times k}, \) and the rest of the blocks are the appropriate dimensions. BT of \( k \) states projects the system to become:

\[
\begin{align*}
\dot{\bar{x}}_1(t) &= \bar{A}_{11}\bar{x}_1(t) + \bar{B}_1u(t) \\
y(t) &= \bar{C}_1\bar{x}_1(t) + \bar{D}u(t)
\end{align*}
\]

Note that stability is preserved under truncation [25]. The transfer function for the truncated system becomes:

\[
\bar{G}_r(s) = \bar{C}_1(sI - \bar{A}_{11})^{-1}\bar{B}_1 + \bar{D}.
\]

Using the H-infinity norm, if \( \theta_{n-k+1} < \theta_{n-k} \) and \( \theta_{n-k+1} = \cdots = \theta_n \),

\[
\|G - \bar{G}_r\|_\infty \leq 2\theta_{n-k+1}.
\]

If \( \theta_{n-k+1} < \theta_{n-k} \) and \( \theta_{n-k+1} > \cdots > \theta_n \),

\[
\|G - \bar{G}_r\|_\infty \leq 2 \sum_{i=n-k+1}^n \theta_i.
\]

Proofs of these error bound can be found in [25, 37–39]. BT has been extended to several different classes of systems including time-varying, multidimensional, and uncertain systems [40–42], and many structure preserving model reduction techniques, with varying definitions, have been considered as extensions of BT [43–47]. Extensions have also been developed for nonlinear systems [48–50].

B. Balanced Singular Perturbation Approximation

Given a partitioned balanced realization characterized as in (5), the reduced system given by BSPA is

\[
\begin{align*}
\dot{\hat{x}}_1(t) &= \bar{A}_{11}\hat{x}_1(t) + \bar{B}_1u(t) \\
y(t) &= \bar{C}_1\hat{x}_1(t) + \bar{D}u(t)
\end{align*}
\]

where

\[
\begin{align*}
\hat{A} &= \bar{A}_{11} - \bar{A}_{12}\bar{A}_{22}^{-1}\bar{A}_{21}, \\
\hat{B} &= \bar{B}_1 - \bar{A}_{12}\bar{A}_{22}^{-1}\bar{B}_2, \\
\hat{C} &= \bar{C}_1 - \bar{C}_2\bar{A}_{22}^{-1}\bar{A}_{21}, \quad \text{and} \\
\hat{D} &= \bar{D} - \bar{C}_2\bar{A}_{22}^{-1}\bar{B}_2.
\end{align*}
\]

The matrix \( \hat{A} \) is the Schur complement of \( \bar{A} \), which we will denote by \( \bar{A} / \bar{A}_{22} \). Note, similarly to BT, stability is preserved under the BSPA process [35].

Define the transfer function of the BSPA system to be

\[
\bar{G}_r(s) = \bar{C}_1(sI - \hat{A})^{-1}\hat{B} + \hat{D}.
\]

Then, identically to BT, if \( \theta_{n-k+1} < \theta_{n-k} \) and \( \theta_{n-k+1} = \cdots = \theta_n \),

\[
\|G - \bar{G}_r\|_\infty \leq 2\theta_{n-k+1}.
\]

If \( \theta_{n-k+1} < \theta_{n-k} \) and \( \theta_{n-k+1} > \cdots > \theta_n \),

\[
\|G - \bar{G}_r\|_\infty \leq 2 \sum_{i=n-k+1}^n \theta_i.
\]

These bounds are proven in [35].

In many respects, BT and BSPA are complementary types of approximations. They are both derived from the same block partition of a balanced realization. They share the same
error bounds, illustrated in \([6,7]\) and \([9,10]\). It is well known that BT typically provides better approximations at high frequencies while BSPA works well at low frequencies \([35]\). In what follows, we will see how these similarities allow us to unify both approximations as limiting approximations of a balanced system.

### III. Manifold Boundary Approximation Method

The Manifold Boundary Approximation Method, i.e. MBAM, was originally described in \([1]\), but we present an overview here for completeness. The idea is that a parameterized set of models, as a manifold embedded in the space of possible predictions (i.e. embedded in data space). As such, it is natural to interpret a model class, or a continuous manifold, as being on the boundary of the model class. For a given model manifold \(\mathcal{M}(D)\), we denote the closure of the manifold as \(\bar{\mathcal{M}}(D)\).

**Definition 1.** A model mapping of the set of systems in \([11,13]\) sampled at fixed times is defined as the map \(\mathcal{M} : D \subset \mathcal{P} \rightarrow \mathcal{D}\), where \(D = \mathbb{R}^{M \times p}\) is the data or prediction space, with \(N < M\).

In this work, we assume that \(f\), \(g\), and \(h\) in \([11,12]\), and \([13]\) are such that \(\mathcal{M}\) is smooth. Furthermore, note that in Definition 1 we have assumed that \(N < M\), with no bound on \(M\). It is possible for the data or prediction space to be infinite dimensional. In many cases, however, we sample the output at specific times, making the data space finite dimensional. Note that while other methods, such as \([48]\), sample system data to build an empirical controllability Gramian, the sampling here embeds the model manifold in an appropriately large data space. Moreover, the topological properties of the manifold are largely invariant to details of the sampling \([52]\). Thus, the model manifold becomes the central object to represent the parameterized model class describing the system.

We use the notation \(\mathcal{M}(D)\) to denote the image of \(D\), or the set of points in \(\mathbb{R}^{M \times p}\) that are mapped from \(D\); this set of points is a manifold embedded in \(\mathcal{D}\), called the model manifold. Each point on the manifold, then, corresponds to a particular choice of parameter, \(p \in \mathcal{P}\), which specifies a given system in the model class. For a given model manifold \(\mathcal{M}(D)\), we denote the closure of the manifold as \(\bar{\mathcal{M}}(D)\).

**Definition 2.** Given a model mapping \(\mathcal{M} : D \subset \mathcal{P} \rightarrow \mathcal{D}\), a point \(d \in \mathcal{M}(D)\) is interior if there exists an open neighborhood of \(\mathcal{M}(D)\) centered at \(d\). If \(d\) is not interior then it is a boundary point.

The set of boundary points of the manifold closure defines the boundary of the manifold, denoted by \(\partial\mathcal{M}(D)\).

**Definition 3.** Given a model mapping \(\mathcal{M} : D \subset \mathcal{P} \rightarrow \mathcal{D}\), a model mapping \(\mathcal{M}\) is a \(k\)-order manifold boundary approximation of \(\mathcal{M}\) if:

1. \(\mathcal{M} : D \subset \mathcal{P} = \mathbb{R}^{N-k} \rightarrow \mathcal{D}\) and
2. \(\mathcal{M}(D) \subset \partial\mathcal{M}(D)\).

That is to say, a \(k\)-order model boundary approximation is itself a manifold defined as being on the boundary of the model manifold and having \(k\) less parameters than the original model manifold. A model boundary approximation method, then, is a systematic process for choosing a \(k\)-order approximation to the model mapping on the boundary of the model manifold.

Note that in the case where the output is sampled, one needs to collect more samples than parameters to ensure that the model manifold is embedded in a large enough space. In general, though, as long as there are enough sample points, the topological features of the resulting model manifold are invariant to the specific times when the samples are taken as long as they are decided randomly—see \([52]\) for details. Once a model manifold is constructed, however, we then turn our attention to the specific details of choosing a \(k\)-order approximation.

To develop such an MBAM method, we consider the sensitivity of such models to their parameters by taking partial derivatives as follows:

\[
\frac{d}{dt} \frac{\partial x}{\partial p} = \frac{\partial f}{\partial x} \frac{\partial x}{\partial p} + \frac{\partial f}{\partial z} \frac{\partial z}{\partial p} + \frac{\partial f}{\partial p} \quad (14)
\]

\[
0 = \frac{\partial g}{\partial x} \frac{\partial x}{\partial p} + \frac{\partial g}{\partial z} \frac{\partial z}{\partial p} + \frac{\partial g}{\partial p} \quad (15)
\]

\[
\frac{\partial y}{\partial p} = \frac{\partial h}{\partial x} \frac{\partial x}{\partial p} + \frac{\partial h}{\partial z} \frac{\partial z}{\partial p} + \frac{\partial h}{\partial p} \quad (16)
\]

Note that these equations are linear in terms of sensitivities, but the matrices involved do vary with time. We see that \([14-16]\) are derived by differentiating \([11,13]\) with respect to the parameters and applying the chain rule to account for the implicit dependence of the dynamic, algebraic, and measured variables on the parameters. We also use the second order sensitivities in the sequel, the equations for which can be derived similarly. The problem of calculating parametric sensitivities for dynamical systems is well-known, with a long history \([53]\). However, deriving expressions for the first and second order sensitivities by hand can be tedious and error prone (particularly for large models). We therefore use...
Previous work suggests that model manifolds for real, physical systems tend to be “thin,” in the data space where they are embedded; that is to say, measured variables tend to be considerably less sensitive to some parameter combinations than others [1], [56], [57]. MBAM exploits this “sloppy” character of such model classes by approximating this “thin” manifold by its boundary. We identify the boundary by numerically computing a geodesic on the model manifold and using the results from this calculation to identify an approximate model. This “boundary” model will have fewer parameters than the original, and it approximates the original model manifold the same way a long edge of a ribbon approximates the ribbon.

Geodesics are the analogs of straight lines generalized to curved surfaces, and we compute them numerically as the solution to a second order ordinary differential equation

\[
\frac{d^2p^i}{d\tau^2} = \sum_{j,k} \Gamma_{jk}^i \frac{dp^j}{d\tau} \frac{dp^k}{d\tau}; \quad \Gamma_{jk}^i = \sum_{l,m} (I^{-1})^{il} \frac{\partial y_m}{\partial p_k} \frac{\partial^2 y_m}{\partial p_j \partial p_k}
\]

where \( \Gamma \) are the Christoffel symbols [57]–[59], containing curvature information about the mapping between parameter space and data space. These are expressed in terms of the first and second order parametric sensitivities [14]–[16] and \( I \) is the Fisher Information Matrix (FIM) for the measurement process. The geodesic is parameterized by \( \tau \), which is proportional to the arc length of the geodesic as measured on the model manifold, i.e., in data space. Solving (17) gives a parameterized curve \( p(\tau) \) in parameter space that is used to reveal a limiting behavior in the model as we demonstrate below.

Equation (17) is an ordinary differential equation that we solve as an initial value problem. Here, we take the models nominal, or “true” parameter values as the starting point of the geodesic. The initial “velocity” is taken to be the least sensitive parameter direction as measured by the FIM. In this way, the MBAM procedure can be summarized as a five step algorithm.

**Step a**: The least sensitive parameter combination is identified from an eigenvalue decomposition of the FIM which becomes the initial geodesic velocity, \( \partial p / \partial \tau \). The geodesic acceleration \( \partial^2 p / \partial \tau^2 \) is given by (17). If \( \sum_i (\partial p^i / \partial \tau)(\partial^2 p^i / \partial \tau^2) < 0 \), then reverse the initial velocity: \( \partial p / \partial \tau \rightarrow -\partial p / \partial \tau \). This heuristic resolves the ambiguous direction associated with the eigenvalue.

**Step b**: A geodesic on the model manifold is constructed by numerically solving (17) using the nominal or “true” parameter values and the velocity \( \partial p / \partial \tau \) calculated in Step a as initial conditions. The limiting behavior of this curve identifies the boundary of the model manifold.

**Step c**: Having found the edge of the model manifold, the corresponding lower-dimensional model class is identified as an approximation to the original model class. By inspecting which values of the parameter vector become infinite, we identify the boundary as a limiting approximation in the model class. We evaluate this limit to construct the approximate model class parameterized by fewer parameters.

**Step d**: We choose parameter values of this approximate model class by fitting the approximate model to the behavior of the original model using, for example, least squares regression.

**Step e**: The procedure is repeated until the reduced model is unable to approximate quantities of interest within the desired tolerance.

Note that there can be many model boundary approximation methods, depending on which boundary one chooses as an approximation of the model manifold. A common approach is to choose a metric and then select the boundary that is closest to a given point in the data space. In the original presentation of MBAM [1], an information distance was imposed on the data space defined by the FIM, which defines a Riemannian metric on the parameter space [58], [59]. From this Riemannian metric, computational differential geometry can be used to identify candidate boundary approximations [1]. Nonetheless, MBAM is a topological operation in general and is agnostic to the actual choice of metric [52].

### A. Michaelis Menten Example

The MBAM concept is best illustrated by example. Consider the dynamics of the Michaelis Menten Reaction (MMR),

\[
\dot{x}(t) = \frac{-\rho_1 x(t)}{\rho_2 + x(t)}
\]

where \( \rho_i \) indicates a parameter in the dynamic equations characterizing the family of models under consideration [60]–[62]. The parameters \( \rho_i \) are only physically relevant for positive values; we therefore restrict our attention to the domain \( D \) as the positive quadrant of the parameter space, that is, where \( \rho_i \geq 0 \forall i \). Assuming a fixed initial condition, \( x_0 = 1 \), and varying \( \rho_1 \) and \( \rho_2 \) among all possible elements of the domain,
(a) Potential time series for the MMR model. The red, cyan, and magenta curves are calculated by \((\rho_1, \rho_2)\) values of \((1, 1)\), \((3, 1)\), and \((1, 3)\) respectively.

(b) Visualizing the model’s parameter dependence. Color corresponds to the sum of squares of the the change in the model’s prediction at each time point. The red, cyan, and magenta points correspond to the curves of the same color in Fig. 2a. The green and blue curves are geodesic paths in opposite directions on the model manifold.

(c) A view of the Model Manifold for the MMR model. Colored points correspond to the point of the same color in Fig. 2b and a trajectory of the same color in Fig. 2a. The geodesic curves in Fig. 2b appear as short curves of the same color across the most narrow part of the manifold.

Fig. 2: MBAM works by sampling outputs at particular times for a variety of parameter values (Fig. 2a), constructing geodesics in parameter space (Fig. 2b), and interpreting the limiting case of these curves as portions of the manifold boundary in “data space” (Fig. 2c).

Consider now three observations of this system at time points, \((t_1, t_2, t_3)\), indicated by the vertical dotted lines in Fig. 2a. To visualize the sensitivity of the model to variations in the parameters, it is common to construct an “error function” in parameter space as in Fig. 2b. An alternative approach is to recognize that the space of all possible experimental data at these three time points forms a three-dimensional “data space.” All possible model predictions, for different values of \(\rho_1\) and \(\rho_2\), correspond to a two-dimensional subset of this space. This two-dimensional surface is the model manifold characterized as in Definition 1. The manifold for the MMR model illustrated in Fig. 2c is bounded, a feature shown to be common among model manifolds with many more parameters than can be estimated from available data [56], [57]. The existence of these boundaries is the crucial element that enables the manifold boundary approximation, where each boundary corresponds to a different model reduction of the original model in (18). The “top” boundary reached by the blue geodesic curve, corresponds to the limit where \(\rho_1, \rho_2 \to \infty\), and the ratio \(\rho_1/\rho_2\) becomes the new parameter. The “bottom” boundary reached by the green geodesic curve corresponds to the limit where \(\rho_2 \to 0\), leaving \(\rho_1\) as the sole parameter. These two different limiting approximations each lead to two distinct reduced models of the form:

\[
\dot{x}(t) = -\frac{\rho_1}{\rho_2} x(t), \quad \text{and} \quad (19)
\]

\[
\dot{x}(t) = -\rho_1, \quad (20)
\]

respectively. A well–known interpretation of the MMR in biochemistry is that it describes a saturable reaction. That is, at low concentrations (small values of \(x\)), the reaction rate is nearly first order (i.e., linear in \(x\)), while at high concentrations the rate saturates into a zeroth order (i.e., constant with respect to \(x\)). Notice how these two limits are naturally identified as the boundaries of the model manifold in Fig. 2c and together bound the space of predictions that can be reached by the model.

To be clear, the use of MBAM to approximate the MMR system here reduced the set of systems from a two dimensional set to a one dimensional set, but it did not reduce the dynamical order of the system, which started as a first order system and remained, in each limiting case, a first order system. This is because parameter reduction does not necessarily lead to model-order reduction. The next example will demonstrate a case where parameter reduction does reduce the order of the model. Although the dynamical order was not reduced in the example above, the reduced models are simpler. While the original model exhibited nonlinear dynamics, the approximate models in (19) and (20) are linear and constant, respectively. Note that code for computing each step of the MBAM process can be found at [63].
B. Second-Order LTI Example

As a second example, consider the set of systems described by:

\[
\begin{bmatrix}
\dot{x}_1(t) \\
\dot{x}_2(t)
\end{bmatrix} =
\begin{bmatrix}
-\rho_1 & 0 \\
0 & -\rho_2
\end{bmatrix}
\begin{bmatrix}
x_1(t) \\
x_2(t)
\end{bmatrix}
\]

\[
y = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}
\]

where \( t, \rho_1, \rho_2 \in \mathbb{R}^+ \). For any choice of \( \rho_1 \) and \( \rho_2 \), the behavior of the corresponding system is the set of functions

\[
y(t) = e^{-\rho_1 t} x_1(0) + e^{-\rho_2 t} x_2(0).
\]

When we fix the initial condition, say to \( x(0) = [1 \ 1]^T \), this set is completely parameterized by the parameters \( \rho_1 \) and \( \rho_2 \). One way to characterize the behavior of this set of systems is to associate each choice of parameters with an observation vector given by \( v = [y(t_1) \ y(t_2) \ y(t_3)]^T \), where the observation times \( t_1, t_2, \) and \( t_3 \) are fixed. Spanning over all choices of parameters \( \rho_1 \) and \( \rho_2 \), the observation vectors \( v(\rho_1, \rho_2) \) then sweep out a two-dimensional manifold embedded in \( \mathbb{R}^3 \), shown in Fig. 3b-3c. This is the model manifold for this set of systems, and we note that although its size and shape somewhat depend on the observation times \( t_1, t_2, \) and \( t_3 \), its characteristic topological features (number of cusps and edges and the relationships among them) are determined solely by the relationship between the parameters, \( \rho_1 \) and \( \rho_2 \), and observations, \( y(t) \). That is to say, regardless of how complicated the underlying dynamics are, an \( N \)-parameter system generates an \( N \)-dimensional manifold embedded in a larger data space (a fact that therefore requires the number of observation points to be larger than the number of parameters), and the relationship between these cusps and edges is fixed, regardless of when we make the observations or whether we make more than \( n \) observations, etc. [22].

This invariance of the key topological features of the model manifold allows us to consider a graphical representation among these features, shown in Fig. 4. This graph encodes the relationship between the model manifold surface in Fig. 3 and its boundaries, and between its boundaries and their endpoints. Each of these objects are themselves a manifold of lower dimension, and this hierarchy of lower dimensional manifolds forms a partial ordering that reveals different ways the fully parameterized model can be approximated by representations of lower complexity.

Thus, for example, each point on the model manifold, \( M \), in Fig. 3 is specified by choosing particular values for both of the parameters \( \rho_1 \) and \( \rho_2 \); the fact that the model manifold is two-dimensional results from the fact that two parameters must be specified. When one of these parameters is set to one of its limiting values, zero or infinity, the resulting set of models parameterized by the other parameter form a boundary on the model manifold, a boundary of dimension one (since only one parameter is left free). Thus one edge is formed when one of the parameters (without loss of generality, assume it is \( \rho_1 \)) is fixed at zero, yielding the remaining behaviors \( y(t) = 1 + e^{-\rho_2 t} \) represented by the one-dimensional manifold, the edge \( e_{AB} \). Another edge is formed when \( \rho_2 \) approaches
infinity, yielding \( y(t) = e^{-\rho_1 t} \) and represented by the edge \( e_{BC} \). Finally, the last boundary of \( M \) is formed when \( \rho_1 \) is fixed to the same value as \( \rho_2 \), yielding \( y(t) = 2e^{-\rho_2 t} \), and is represented by the edge \( e_{AC} \).

The three points where these edges intersect form zero-dimensional manifolds and represent systems where both parameters are fixed at one of the limiting values. At point A, \( \rho_1 = \rho_2 = 0 \) and the output \( y(t) \equiv 2 \). At point B, one of the parameters (\( \rho_1 \) in our case above) is zero while the other approaches infinity, with \( y(t) \equiv 1 \). At point C both parameters approach infinity and \( y(t) \equiv 0 \).

The partial ordering among these models of decreasing complexity is captured by the Hasse diagram, shown in Fig. 3. This diagram captures the relationship between model complexity and different types of approximations. So, for example, \( e_{AB} \), with behavior \( y(t) = 1 + e^{-\rho_2 t} \), is an approximation that considers that the unmodeled mode of the system is much faster than the other; it models the fast dynamics of the system. On the other hand, \( e_{BC} \), with behavior \( y(t) = e^{-\rho_1 t} \), is an approximation that considers that the unmodeled mode of the system is much faster than the other and has already reached equilibrium; it models the slow dynamics of the system. Finally, \( e_{AC} \) describes the situation where both decay rates are comparable; it models the dynamics of a symmetric system.

Each of these three types of approximations capture a different idealized behavior of every model in the set of systems parameterized by \( \rho_1 \) and \( \rho_2 \). Likewise, each of these approximations are themselves approximated at their endpoints by one of three extreme behaviors A, B, or C. These behaviors are themselves stationary, describing the regimes of behavior characterizing these systems. In this example, A represents a "high" value, B a "medium" value, and C a "low" value, but in other contexts they might represent different phenotypes or behavior regimes such as "healthy" and "sick," etc.

These examples all illustrate the underlying mechanism driving MBAM techniques—that of looking for approximations on the boundary of the model manifold. Current research is exploring efficient algorithms for conducting such a search, as well as application of the ideas to large scale systems, but this work reports how both BT and BSPA, and a host of hybrid techniques, become special cases of MBAM methods when applied to LTI systems.

IV. PARAMETERIZATIONS OF LTI SYSTEMS

The previous examples show how MBAM can reduce the number of parameters used to describe a set of models. The reduced parameter descriptions for the first example, given by (19) and (20), parameterize a subset of the original model class, but this subset is chosen to be “representative” of the original class in the sense that it is a set of models on the boundary of the original set. We can parameterize this set of boundary models with fewer parameters than the original set, hence MBAM is a parameter-reduction technique.

Parameter reduction is not necessarily model order reduction, however. The example in Section III-A shows that MBAM can reduce the number of parameters without changing the order of the system, while the example in Section III-B shows that depending on the parameterization the order can be reduced. The interest of this work is to use MBAM to perform model order reduction on general LTI systems. This requires a careful examination of the parameters of such systems, so that reduced-order models generated by BT or BSPA can be shown to be instances of reduced-order models generated by MBAM.

In this section we present and classify the parameters of general LTI systems, and the following sections will leverage this parameterization to prove this relationship between these two model order reduction techniques and provide a framework to interpolate between them.

A. Parameterizations of Transfer Functions

Consider the LTI system in (1), which has \( m \) inputs and \( p \) outputs. The input-output behavior of such a system is characterized by

\[
Y(s) = G(s)U(s)
\]

where \( U(s) \) and \( Y(s) \) are the Laplace transforms of the input and output signals, respectively, and \( G(s) \) is given by

\[
G(s) = C(sI - A)^{-1}B + D. \tag{21}
\]

In this context, the operator \( G \) is the transfer function of the LTI system, and shows what the output of the system will be given any input. Thus \( G \) is a \( p \times m \) matrix, where each entry is a proper polynomial of the Laplace variable \( s \in \mathbb{C} \), which we will write as

\[
G(s) = \begin{bmatrix}
\nu_{11}(s) & \cdots & \nu_{1m}(s) \\
\delta(s) & \ddots & \delta(s) \\
\vdots & \ddots & \vdots \\
\nu_{p1}(s) & \cdots & \nu_{pm}(s) \\
\delta(s) & \cdots & \delta(s)
\end{bmatrix}. \tag{22}
\]

Without loss of generality, we may consider that each entry has a common denominator \( \delta(s) \), with degree \( N \), and the degree
A minimal list of parameters that define the set of all transfer functions of degree $N$ with $m$ inputs and $p$ outputs. This parameterization is shown in (22).

**Gilbert’s Transfer Function Parameterization**

| Name   | # Parameters | # Occurrences | Type     |
|--------|--------------|---------------|----------|
| $\delta(s)$ | $N$ | 1 | Identifiable |
| $\nu_{ij}(s)$ | $N+1$ | pm | Identifiable |
| Total | $Npm + N + pm$ | | |

(b) A minimal list of parameters that define the set of all transfer functions of degree $N$ with $m$ inputs and $p$ outputs, with the constraint that the poles of the system have geometric multiplicity one. This parameterization is shown in (24).

**Example 1 Transfer Function Parameterization**

| Name   | # Parameters | # Occurrences | Type     |
|--------|--------------|---------------|----------|
| $\lambda_i$ | 1 | $N$ | Identifiable |
| $K_i$ | pm | $N$ | Identifiable |
| $D$ | pm | 1 | Identifiable |
| Total | $Npm + N + pm$ | | |

(c) A minimal list of parameters that define the set of all transfer functions of degree $N$ with $m$ inputs and $p$ outputs, with the constraint that the system poles all have algebraic multiplicity one. This scenario is described in Example 1.

Fig. 5: Parameter counts for three different sets of transfer functions.

of every polynomial $\nu_{ij}(s)$ is equal to $N$. The matrix $G$ can be rewritten as

\[
G(s) = \frac{\nu(s)}{\delta(s)}
\]

where $\nu(s) = [\nu_{ij}(s)]$. We assume that $\delta(s)$ is monic (the leading coefficient is one), since if it were not, in each entry, both $\delta(s)$ and $\nu_{ij}(s)$ could be divided by the leading coefficient in $\delta(s)$ without changing the behavior of $G(s)$. Thus, each $\nu_{ij}(s)$ is determined by the $N + 1$ coefficients; therefore $\nu(s)$ is completely characterized by $pm(N + 1)$ coefficients. Adding in the $N$ coefficients of $\delta(s)$ (not $N + 1$ because $\delta(s)$ is monic), it follows that $G(s)$ is completely characterized by $Npm + N + pm$ coefficients (see Fig. 5c).

These coefficients provide a way to minimally parameterize the set of all $p \times m$ LTI operators of degree $N$. The parameterization is minimal in that if any coefficient is perturbed, then this perturbation will uniquely change the behavior of the LTI system, and no coordinated changes among the other parameters can produce the same change in the input-output behavior. Moreover, all LTI systems of size $p \times m$ with degree $N$ have such a parameterization. In this context we conclude that a $p \times m$ transfer function $G(s)$ with degree $N$ is uniquely characterized by $Npm + N + pm$ parameters.

Another way to minimally parameterize the set of $p \times m$ degree $N$ transfer functions is to employ a partial fraction expansion on each element $\frac{\nu_{ij}(s)}{\delta(s)}$. For systems where every pole has a geometric multiplicity of one, the transfer function then becomes

\[
G(s) = D + \sum_{i=1}^{N} \frac{K_i}{s - \lambda_i}
\]

where $\lambda_i \in \mathbb{C}$ are the roots of $\delta(s)$, and $D, K_i \in \mathbb{R}^{p \times m}$. The matrix $D$ is a matrix of the leading coefficients for each $\nu_{ij}$ and is equal to $D$ in (1) for any state space realization of $G(s)$; indeed, every state realization of $G(s)$ will have the same $D$ matrix. The $K_i$ matrices contain the scalar numerators corresponding to $s - \lambda_i$ in the partial fraction expansion. In this representation (called Gilbert’s realization [64]), the parameters are $D$ ($pm$ entries), $\lambda_i$’s ($N$ elements of the spectrum), and $K_i$ ($pm$ entries for each $i = 1, ..., N$). Thus we see that the number of parameters remains the same as the previous parameterization: $Npm + N + pm$ (see Fig. 5b). Nevertheless, this particular parameterization is often more descriptive of the actual system dynamics: not only are the input-output dynamics clearly defined, but as we will see in Section IV-B, each $\lambda_i$ is a mode of the system and the $D$ matrix represents how the inputs directly affect the outputs.

**Example 1.** Consider the representation in (24) where $K_i$ are all rank one matrices; this scenario arises when the poles of the system have algebraic multiplicity of one (i.e., they are distinct). For a particular $i$, instead of $K_i$ being parameterized by $pm$ parameters, here many of the parameters of $K_i$ are dependent on the other parameters, in order to ensure that the rank one constraint is met. Let $c_i \in \mathbb{R}^p$ and $b_i \in \mathbb{R}^m$ for all $i$, where $c_i b_i^T = K_i$. We can assume that the first entry of $c_i$ is one, if not, we could divide $c_i$ by its first entry and multiply $b_i$ by the same, thus leaving $K_i$ unchanged. Under this circumstance, each $K_i$ now only requires $p + m - 1$ entries to be specified in order to completely characterize it. Additionally, now any perturbations in these entries will uniquely change $K_i$. Therefore, these $p + m - 1$ entries are the parameters of $K_i$. Thus for this example, the total number of parameters for $G(s)$ is $Np + Nm + pm$ (see Fig. 5c).

We note that both parameterizations of the set of all $p \times m$ transfer functions of degree $N$, (23) and (24), have the same number of parameters. Generically, almost any element of this set is uniquely identified by a choice of these parameters, and sufficiently rich input-output data enables the identification of the parameters specifying the system generating the data. As a result, in either case, we call these $Npm + N + pm$ parameters identifiable. This definition of parameter identifiability is consistent with the definition of structurally identifiable [65].

**B. Parameterizations of State Realizations**

In the previous section we saw that the set of all transfer functions of a given size and degree was efficiently parameterized by a particular number of identifiable parameters, and we demonstrated two distinct ways of doing this. In this section we extend these ideas to state realizations.
Recall that the standard form for state realizations is given in [1], and that we have assumed minimality. The set of all state realizations with \( m \) inputs and \( p \) outputs of order \( n \) is parameterized by four matrices: \( A \in \mathbb{R}^{n \times n} \), \( B \in \mathbb{R}^{n \times m} \), \( C \in \mathbb{R}^{p \times n} \), and \( D \in \mathbb{R}^{p \times m} \). Counting each entry in each matrix as a parameter yields a total of
\[
n^2 + nm + np + pm
\]
parameters for the set.

In the generic case, \( A \) is diagonalizable with no repeated eigenvalues. This implies that there exists a transformation \( T \) such that \( x = T \hat{x} \) and \( T^{-1}AT = \hat{A} \) is diagonal. Likewise, \( T^{-1}B = \hat{B}, CT = \hat{C}, \) and \( D = \hat{D} \). In this case, by (21),
\[
G(s) = \hat{C}(sI - \hat{A})^{-1}\hat{B} + \hat{D}
\]
\[
= \hat{D} + [\hat{c}_1 \cdots \hat{c}_n]
\begin{bmatrix}
\frac{1}{s - \lambda_1} & \cdots & 0 \\
0 & \ddots & 0 \\
0 & \cdots & \frac{1}{s - \lambda_n}
\end{bmatrix}
\begin{bmatrix}
\hat{b}_1^T \\
\vdots \\
\hat{b}_n^T
\end{bmatrix}
\]
\[
= \hat{D} + \sum_{i=1}^{n} \hat{c}_i \hat{b}_i^T
\]

Each term \( \hat{c}_i \hat{b}_i^T \) is a rank one matrix (being the outer product of two vectors), thus it matches the transfer function parameterization given in (24), satisfying the constraint given in Example 1 that each \( K_i \) is rank one. Example 1 shows that there are \( Nm + Np + pm \) parameters in the set of such transfer functions, where \( N \) is the degree of \( \delta(s) \). Since \( A \) has no repeated eigenvalues, it follows from (26) that \( N = n \). This implies that the set of corresponding transfer functions has \( nm + np + pm \) parameters. Therefore, by (25), the set of corresponding state realizations has \( n^2 \) more parameters.

This discrepancy in parameters is not surprising, since it is well-known that any transfer function will have infinitely many corresponding state realizations. Further, it is also well-known that any two state realizations of a transfer function are related by an \( n \times n \) state transformation matrix \( T \) as described prior to (26). Therefore, once a \( G \) has been specified with \( nm + np + pm \) parameters, \( n^2 \) additional parameters are required to specify an instance of \( (A, B, C, D) \). We say that these \( n^2 \) parameters are structural, since they do not affect the system dynamics and are not identifiable from input-output data, but they do affect the internal structure of the system.

The matrices given in (1) are one way to parameterize the set of state realizations. In this parameterization, it is clear from (24) that \( D \) can be identified directly from knowing \( G \), so its entries are also identifiable parameters. That is to say, every state space realization of \( G \) has the same \( D \) matrix. However, the rest of the parameters cannot be identified simply by knowing \( G \): one must also know \( n^2 \) additional parameters (fixing the structure, in a sense, by fixing the basis of the state space). For instance, if we fix \( G \), there are many admissible choices for \( B \) and \( C \), but it could be the case that if \( B \) were known, \( C \) could be determined from data. It also is the case that the reverse is true: fixing \( C \) would allow \( B \) to be determined from data. Therefore, we refer to the entries of \( B \) and \( C \) as conditionally identifiable: once a certain number of them have been fixed, the rest can be determined from data (see Fig. 7a). An illustrative example is also shown in Fig. 8.

Conditionally identifiable parameters are not ideal when performing MBAM, because it is not clear whether or how the structure or dynamics will be affected. Therefore, it is important to find a state space parameterization which has only identifiable and structural parameters.

C. Parameterization Using the Balanced Realization

When performing MBAM on a state realization, it is important to know which parameters will affect the dynamics of the system (identifiable) and which will affect the implementation of the dynamics (structural), since either may be fixed, depending on the application. Parameterizations that partition the parameters into identifiable and structural without the need for conditionally identifiable parameters are useful when applying MBAM so that unintended parameters are not eliminated. Unique canonical realizations allow one to accomplish this goal. All parameters from a unique canonical realization can be identified from data, since the canonical realization fixes the structure. Then, any realization can be parameterized using the parameters of the canonical realization as the identifiable parameters and the transformation matrix \( T \) between the two realizations as the structural parameters.

We illustrate this point by employing the balanced realization. Consider a minimal, stable system as in (1). It is well known that there exists a state transformation from these matrices to an input–output equivalent balanced realization \((\bar{A}, \bar{B}, \bar{C}, \bar{D})\) as in (3), where \( X \) is the diagonal matrix of HSVs \((\theta_i)'s\) satisfying (4). Notice the following simple statement is true for balanced realizations.

**Lemma 1.** Given a balanced realization \((\bar{A}, \bar{B}, \bar{C}, \bar{D})\), \( \text{diag}(BB^T) = \text{diag}(CTC) \).

**Proof.** Since \((\bar{A}, \bar{B}, \bar{C}, \bar{D})\) is balanced, the observability and controllability Gramians are equal with the HSVs, \( \theta_1, \ldots, \theta_n \), on the diagonal. The diagonals of the Lyapunov equations give
\[
\begin{align*}
a_{ii}\theta_i + \theta_i a_{ii} &= -(\bar{C}^T \bar{C})_{ii} \\
a_{ii}\theta_i + \theta_i a_{ii} &= -(BB^T)_{ii}
\end{align*}
\]
where the subscript \( ii \) indicates the \( i \)th diagonal entry of the matrix and \( a_{ii} \) is the \( i \)th diagonal entry of the \( \bar{A} \) matrix. Therefore \( \text{diag}(BB^T) = \text{diag}(CTC) \). \( \square \)

Note that this lemma is also a result of Theorem 1 in [67].

Let the common diagonal entries of \( \bar{B} \) and \( \bar{C} \) be denoted by \( r_1, \ldots, r_n \). We can then write the \( \bar{B} \) matrix as
\[
\bar{B} = \begin{bmatrix}
\beta_1^T \\
\vdots \\
\beta_n^T
\end{bmatrix}
\]
where the \( \beta_i \)’s are a collection of normalized column vectors in \( \mathbb{R}^m \) satisfying \( \beta_i^T \beta_i = 1 \), for all \( i = 1, \ldots, n \). Denoting the \( n \times m \) matrix whose rows correspond to \( \beta_i^T \) as \( \beta^T \) and
Fig. 6: A SISO first order transfer function has three identifiable parameters, while all minimal realizations of any such transfer function have four parameters, some of which are identifiable, conditionally identifiable, or structural. Consider this example, $G(s)$, where $d = 0$, $k = 1$, and $\lambda = 3$. All realizations of this transfer function have $a = 3$ and $d = 0$, so these two of the four state parameters are identifiable, leaving the other parameters, $b$ and $c$, to be conditionally identifiable (i.e. identifiable conditioned on a particular choice of other parameters). Figure (a) shows the hyperbolic set of possible $(b, c)$ combinations given $a = 3$, and Figures (b) and (c) show how a different realization $(a, b, c)$ combination) is selected from the allowed set if, for example, one knew a priori that $b = 3$ (Figure (b)) or $c = 4$ (Figure (c)). In the case of fixing $b = 3$ (or, respectively $c = 4$), $b$ would be the structural parameter (known a priori, specifying a particular basis for the state space realization), enabling the remaining parameter $c$ (or $b$, respectively) to become identifiable [66].

introducing $R = \text{diag}(r_1, \ldots, r_n)$, it follows that $\bar{B} = R\beta^T$. Clearly by construction $\text{diag}(B\bar{B}^T) = (r_1^2, \ldots, r_n^2)$.

Similarly, we can write $\bar{C}$ as

$$\bar{C} = \begin{bmatrix} r_1\gamma_1 & \ldots & r_n\gamma_n \end{bmatrix}$$

where the $\gamma_i$’s are a collection of normalized column vectors in $\mathbb{R}^p$ satisfying $\gamma_i^T\gamma_i = 1$, for all $i = 1, \ldots, n$. We write $\bar{C} = \gamma R$; note that $\text{diag}(\bar{C}^T\bar{C}) = (r_1^2, \ldots, r_n^2) = \text{diag}(B\bar{B}^T)$, consistent with Lemma 1.

Plugging $r_i^2$ into (27) gives

$$\bar{a}_{ii} = -\frac{r_i^2}{2\theta_i}$$

(28)

for the diagonal elements of the $\bar{A}$ matrix. From the off-diagonals of the Lyapunov equations we find,

$$\bar{a}_{ij} = r_ir_j\alpha_{ij},$$

(29)

for $i \neq j$, where

$$\alpha_{ij} = \frac{\theta_j(\beta^T\beta)_{ij} - \theta_i(\gamma^T\gamma)_{ij}}{\theta_i^2 - \theta_j^2}.$$

Leveraging these properties, we see that a balanced realization is specified by $(\theta, \beta, \gamma, R, D)$ as follows:

$$\begin{align*}
\dot{\bar{x}}(t) &= \bar{A}\bar{x}(t) + \begin{bmatrix} r_1\beta_1^T \\ \vdots \\ r_n\beta_n^T \end{bmatrix} u(t) \\
y(t) &= \begin{bmatrix} r_1\gamma_1 & \ldots & r_n\gamma_n \end{bmatrix} \bar{x}(t) + Du(t)
\end{align*}$$

(30)

where $\bar{A}$ is defined in (28) and (29). We know that $\theta$ and $R$ both contain $n$ parameters and that $D$ contains $pm$ parameters. While $\beta$ is an $n \times m$ matrix, since $\beta_i^T\beta_i = 1$, specifying $n - 1$ entries in $\beta_i$ fixes the magnitude of the final entry. Thus $\beta$ carries only $n(m - 1)$ parameters. Likewise, $\gamma$ carries $n(p - 1)$. It follows that any realization can be specified by these parameters along with an additional transformation matrix $T$ that dictates the change of basis from the balanced realization. Using this balanced parameterization, we again have $n^2 + nm + np + pm$ parameters (see Fig. 7), but now each parameter is clearly labeled as either identifiable or structural. Since any realization can be uniquely described by $T$, its transformation matrix to the balanced realization, we know that there are $n^2$ structural parameters. As discussed, the remaining $nm + np + pm$ parameters match the same number as $G(s)$ in Example 1 and therefore are all identifiable from data. As a result this parameterization enables one, when using MBAM to reduce the number of parameters, to be deliberate as to how the structure and/or dynamics are affected.

V. Unification of Model Reduction Techniques

Given the parameterization of LTI systems of the previous section, we are equipped to apply the algorithm summarized in Figure 1. In our experience solving the geodesic equation on multiple LTI systems, we empirically observed that the boundary approximations always take one of two forms when reducing the order of the system. This empirical observation suggests that it may not be necessary to sample the model predictions and solve Eq. 17. In other words, it is not necessary
Multiplying through by \( \theta_n \) gives
\[
\theta_n \dot{x}_n(t) = \theta_n \sum_{i=1}^{n-1} r_i r_n \alpha_{ni} \dot{x}_i(t) - \frac{r_n^2 \dot{x}_n(t)}{2\theta_n} + \theta_n r_n \sum_{i=1}^{m} \beta_{ni} u_i(t).
\]

Performing an MBAM approximation by taking the limit \( \theta_n \to 0 \) gives that \( \ddot{x}_n = 0 \). Plugging this back into the dynamics of the rest of the system, i.e. \( \dot{x}_i(t), \ i < n \), gives
\[
\dot{x}_i(t) = -\frac{r_i^2 \dot{x}_i(t)}{2} + \sum_{j=1,j\neq i}^{n-1} r_i r_j \alpha_{ij} \dot{x}_j(t) + \sum_{j=1}^{m} \beta_{ij} u_j(t),
\]
which is equivalent to BT of one state. Clearly also having \( x_n = 0 \) gives that
\[
y(t) = \begin{bmatrix} r_1 \gamma_1 & \cdots & r_{n-1} \gamma_{n-1} \\ \vdots & & \vdots \\ x_{n-1}(t) & & x_{n-1}(t) \end{bmatrix}.
\]

Iterating this MBAM approximation \( k \) more times, always choosing the smallest HSV, completes the proof.

**Theorem 2.** Consider a balanced realization, as in (30), where \( r_1, \ldots, r_n \) are parameters and the Hankel singular values, \( \beta, \gamma \), and \( D \) are fixed. Balanced Truncation of \( k \) states is equivalent to \( k \) iterations of MBAM taking the relevant \( r_i \)'s \( \to 0 \).

**Proof.** Consider the equation for \( \dot{x}_n(t) \) in (31). Performing an MBAM approximation by taking the limit \( r_n \to 0 \) gives that \( \dot{x}_n = 0 \). Plugging this back into the dynamics of the rest of the system, i.e. \( \dot{x}_i(t), \ i < n \), gives
\[
\dot{x}_i(t) = -\frac{r_i^2 \dot{x}_i(t)}{2} + \sum_{j=1,j\neq i}^{n-1} r_i r_j \alpha_{ij} \dot{x}_j(t) + \sum_{j=1}^{m} \beta_{ij} u_j(t).
\]

Also, when \( r_n = 0 \),
\[
y(t) = \begin{bmatrix} r_1 \gamma_1 & \cdots & r_{n-1} \gamma_{n-1} \\ \vdots & & \vdots \\ x_{n-1}(t) \end{bmatrix} \ddot{x}(t).
\]

This gives BT of one state. Iterating this limit \( k \) times, always choosing the \( r_i \) with the largest subscript, completes the proof.

Note that Theorems 1 and 2 indicate two different paths along the model manifold that both converge to the same point on the boundary. These paths are found by taking different limits of different parameters, but the common limit point is the Balanced Truncation approximation for the system.

**B. Singular Perturbation Approximation from MBAM**

In a similar manner BSPA can be derived by applying MBAM to the balanced parameterization.

**Theorem 3.** Consider a balanced realization, as in (30), where \( r_1, \ldots, r_n \) are parameters and the Hankel singular values, \( \beta, \gamma \), and \( D \) are fixed. Balanced Singular Perturbation Approximation of \( k \) states is equivalent to \( k \) iterations of MBAM taking the relevant \( r_i \)'s \( \to \infty \).
Proof. We will prove this by induction, starting with the one state case. Dividing (31) by \( r_n \) gives
\[
\frac{1}{r_n} \dot{x}_n(t) = \frac{1}{r_n} \sum_{i=1}^{n-1} r_i \alpha_{ni} \dot{x}_i(t) - \frac{x_n(t)}{2\theta_n} + \frac{1}{r_n} \sum_{i=1}^{m} \beta_{ni} u_i(t).
\]
Letting \( r_n \to \infty \) then yields
\[
0 = -\frac{x_n(t)}{2\theta_n}
\]
driving
\[
\dot{x}_n \to 0. \tag{32}
\]
Dividing (31) by \( r_n \) gives
\[
\frac{1}{r_n} (\dot{x}_n(t)) = \sum_{i=1}^{n-1} r_i \alpha_{ni} \dot{x}_i(t) - \frac{r_n \dot{x}_n(t)}{2\theta_n} + \sum_{i=1}^{m} \beta_{ni} u_i(t).
\]
Taking the limit as \( r_n \to \infty \) with \( r_n \dot{x}_n \) remaining finite, since, by (32), \( \dot{x}_n \to 0 \), gives
\[
r_n \dot{x}_n(t) = 2\theta_n \sum_{i=1}^{n-1} r_i \alpha_{ni} \dot{x}_i(t) + 2\theta_n \sum_{i=1}^{m} \beta_{ni} u_i(t). \tag{33}
\]
Furthermore, in the remaining equations for \( \dot{x}_i(t), i = 1, \ldots, n - 1, \) and \( y_i(t), i = 1, \ldots, p, \) we find that \( r_n \) and \( \dot{x}_n \) always appear in the combination \( r_n \dot{x}_n \). Therefore, this limit is a well-defined boundary approximation for this parameterization. Plugging (33) into the rest of the system, i.e. \( \dot{x}_i(t), i < n \), gives
\[
\dot{x}_i(t) = \sum_{j=1,j\neq i}^{n-1} (r_j \alpha_{ij} + 2\theta_n r_j \alpha_{in} \alpha_{nj}) \dot{x}_j(t)
\]
\[+ \left( 2\theta_n r_i \alpha_{in} \alpha_{ni} - \frac{r_i}{2\theta_i} \right) \dot{x}_i(t) + \sum_{j=1}^{m} (r_i \beta_{ij} + 2\theta_n r_i \alpha_{in} \beta_{nj}) u_j(t) \tag{34}
\]
\[= \sum_{j=1}^{n-1} (\alpha_{ij} + \frac{\alpha_{in} \alpha_{nj}}{\alpha_{in}}) \dot{x}_j(t) + \sum_{j=1}^{m} (\beta_{ij} - \frac{\alpha_{in} \beta_{nj}}{\alpha_{in}}) u_j(t),
\]
which is the system in (8) for \( k = 1 \), that is, BSPA of one state. By similarly plugging (33) into \( \hat{C} \) and \( \hat{D}, \hat{C} \) and \( \hat{D} \) from (8) for \( k = 1 \) appear.

Now for the inductive step, assume BSPA of \( k \) states is equivalent to \( k \) MBAM approximations, giving the system in (8), denoted by \( \hat{A}, \hat{B}, \hat{C}, \hat{D} \). Let the system matrices that result from BSPA of \( k + 1 \) states be denoted by \( \hat{A}^{k+1}, \hat{B}^{k+1}, \hat{C}^{k+1}, \hat{D}^{k+1} \). Partition \( \hat{A} \) such that
\[
\hat{A} = \begin{bmatrix} \hat{A}_{k+1,1}^{T} & \hat{A}_{k+1,1}^{T} \end{bmatrix} = \begin{bmatrix} \hat{A}_{k+1,1} \hat{A}_{k+1,1}^{T} \end{bmatrix}
\]
where \( \hat{A}_{k+1,1} \in \mathbb{R}^{k \times (k+1)} \) and \( \hat{A}_{k,1} \in \mathbb{R}^{k \times k} \); therefore, using the Schur complement notation, \( \hat{A}^{k+1} = \hat{A} / \hat{A}_{k+1,1} \) and \( \hat{A} = \hat{A} / \hat{A}_{k,1} \). By the Quotient Formula for Schur complements (Theorem 1.4 in (68)),
\[
\hat{A}^{k+1} = \hat{A} / \hat{A}_{k+1,1} = (\hat{A} / \hat{A}_{k,1}) / (\hat{a}_{k+1,1} / \hat{a}_{k,1}) \tag{35}
\]
where
\[
\hat{A}_{k+1,1} = \begin{bmatrix} \hat{a}_{k+1,1}^{T} & \hat{a}_{k+1,1} \end{bmatrix} \hat{A}_{k+1,1}^{T}
\]
with \( \hat{A}_{k,1} \in \mathbb{R}^{k \times k} \) and \( \hat{a}_{k+1,1} \in \mathbb{R}^{k \times k} \). This means that \( \hat{A}^{k+1} \) resulting from BSPA of \( k + 1 \) states (the left hand side of (35)) is equivalent to the analogous part in (34) (the right hand side of (35)).

Now we will show the same holds for \( \hat{B}^{k+1} \), which is slightly more laborious since we cannot appeal to Schur complement properties. Partitioning \( \hat{A}_{k,1} \) as
\[
\hat{A}_{k,1} = \begin{bmatrix} \hat{A}_{1,1} \hat{A}_{1,1}^{T} \end{bmatrix}
\]
with \( \hat{a}_{k+1,1} \) defined in (36) allows us to partition \( \hat{A}_{1,1} \) as
\[
\hat{A}_{1,1} = \begin{bmatrix} \hat{a}_{1,1} \hat{A}_{1,1} \end{bmatrix}
\]
where \( \hat{a}_{k+1,1} \in \mathbb{R}^{n-(k+1)} \). Using the above, (8), and (36), and partitioning \( \hat{B}_{k+1} \) as
\[
\hat{B}_{k+1} = \begin{bmatrix} \hat{b}_{k+1} \hat{B}_{k} \end{bmatrix}
\]
with \( \hat{b}_{k+1} \in \mathbb{R}^{n-k \times m} \) and \( \hat{b}_{k+1} \in \mathbb{R}^{m} \), we have
\[
\hat{B}^{k+1} = \hat{B}_{1} - \begin{bmatrix} \hat{a}_{k+1,1} \hat{A}_{1,1} \end{bmatrix} \hat{A}^{k+1} + \begin{bmatrix} \hat{b}_{k+1} \hat{B}_{k} \end{bmatrix}
\]
where
\[
\hat{a}_{k+1,1} \hat{A}_{1,1} \hat{A}^{k+1} + \begin{bmatrix} \hat{b}_{k+1} \hat{B}_{k} \end{bmatrix}
\]
\[= \begin{bmatrix} \hat{a}_{k+1,1} + \hat{A}_{1,1} \hat{A}^{k+1} \end{bmatrix} \hat{b}_{k+1} + \frac{1}{c} (\hat{a}_{k+1,1} - \hat{A}_{1,1} \hat{A}^{k+1} \hat{a}_{k+1,1}) d
\]
\[
\hat{A} = \begin{bmatrix} \hat{a}_{n-k,n-k} \hat{a}_{n-k,n-k} \end{bmatrix} \quad \hat{B} = \begin{bmatrix} \hat{b}_{n-k,n-k} \end{bmatrix}
\]
with \( \hat{a}_{n-k,n-k} \in \mathbb{R}^{k \times k} \) and \( \hat{b}_{n-k,n-k} \in \mathbb{R}^{m} \). Note that \( c \) is equal to \( \hat{a}_{n-k,n-k}, \hat{A}_{1,1} \hat{A}^{k+1} \hat{a}_{k+1,1} = \hat{a}_{n-k,n-k} \), and \( d = \hat{b}_{n-k,n-k} \). Also note that \( \hat{B}_{1} - \hat{A}_{1,1} \hat{A}^{k+1} \hat{b}_{k+1} = \hat{B}_{1} \). Therefore, (37) becomes
\[
\hat{B}^{k+1} = \hat{B}_{1} - \frac{1}{\hat{a}_{n-k,n-k}} \hat{a}_{n-k,n-k} \hat{b}_{n-k,n-k}
\]
Similar expressions for \( \hat{C}^{k+1} \) and \( \hat{D}^{k+1} \) can be found in an analogous way. Therefore (35), (38), and the analogous expressions for \( \hat{C}^{k+1} \) and \( \hat{D}^{k+1} \) can be obtained, similar to (35), by performing an iteration of MBAM on \( \hat{A}, \hat{B}, \hat{C}, \hat{D} \) by taking \( r_{n-k} \to \infty \), thus completing the inductive step. Therefore, by induction, it holds for all \( k \).

The resulting BSPA system is Hurwitz and is still a balanced realization with the \( n-1 \) HSVs, the same as the largest \( n-1 \)
HSVs of the original system \([\Sigma_2]\).

**Note:** In (34) if \(\theta_n\), which is clearly no longer a HSV, is equal to its original value, then the reduced system is BSPA. However, if \(\theta_n\) is set to zero, then the resulting reduction is BT. Therefore, Theorem 3 offers a whole new class of reduced systems ranging between BSPA and BT, and given a metric or goal, the optimal reduction can be found.

### VI. ILLUSTRATIVE EXAMPLE

Consider the balanced parameterization for state realizations, presented in Section IV-C with \(x \in \mathbb{R}^2\),

\[
\begin{align*}
    \dot{x}(t) &= \begin{bmatrix} -r_2^2 \theta_2 & 0 \\
                        0 & -r_2^2 \theta_2 \end{bmatrix} \overline{x}(t) + \begin{bmatrix} r_1 \\
                        r_2 \end{bmatrix} u(t) \\
    y(t) &= \begin{bmatrix} r_1 \\
                        r_2 \end{bmatrix} \overline{x}(t) + du(t)
\end{align*}
\]

where \(\theta_1 = r_1 = 1\) and \(\theta_2\) and \(r_2\) are the free parameters (while enforcing \(\theta_2 \in [0, \theta_2]\)), and we set \(d = 0\) and \(\beta_1 = \beta_2 = \gamma_1 = \gamma_2 = 1\). We can calculate the frequency responses of the different systems by varying the free parameters \(\theta_2\) and \(r_2\). See Fig. 8 for some sample frequency responses. We then use the frequencies \((\theta_1, \theta_2, \theta_3)\) to create the manifold in the data space. The model manifold, in Figures 8b–8d, has two boundaries. The cyan boundary, at the top of the manifold, corresponds to the parameterizations where \(r_2 \to \infty\) (we use one hundred since it is two orders of magnitude larger than one). The red boundary corresponds to parameterizations where \(\theta_1 = \theta_2\). The point at which the two boundaries meet at the bottom of the manifold, the magenta dot, is BT, corresponding to the parameterization where \(\theta_2\) and/or \(r_2 \to 0\).

Consider the system given by the parameter values \((\theta_2, r_2) = (0.7, 8)\), depicted on the manifold by the black triangle. It is clear that BT (the magenta dot) is not going to be a good approximation of the system by the distance between the two systems on the manifold. However, BSPA denoted by the green "+" symbol is a very close approximation. Consider another system given by the parameter values \((\theta_2, r_2) = (0.01, 0.8)\), depicted on the manifold by the black circle. The reduced model given by BSPA, shown as a green "x", is still fairly close to the true system; however, it is clear that BT (the magenta dot) is much closer.

Although, as was explained in Section IV, BT and BSPA have the same a priori \(H_\infty\) error bound, BT typically gives better results at high frequencies while BSPA excels at low frequencies \([33]\). The optimal approximation requires one to identify a metric customized to a context of interest. As a concrete example, returning to the system given by the parameter values \((\theta_2, r_2) = (0.01, 0.8)\) (the green circle), the cyan curve connecting the black "x" to the magenta dot represents a family of candidate reduced models that interpolate between BT and BSPA. The optimal reduced model relative to the three frequencies sampled, using the two-norm as the metric as measured in the data-space, is the point on this cyan curve that is closest to the green dot. It is identified numerically as \((\theta_2, r_2) = (5.3958 \times 10^{-4}, 0.8)\), depicted by the purple diamond. Clearly this point is much better than the green circle, as is evident from the resulting magnitude plots.

**Fig. 8:** Magnitude plot and the resulting manifold for the two–state model.
closer to the original system than either BSPA or BT and therefore a better approximation for the chosen metric of the two-norm.

Recall that the model manifold, in Figures 8b–8d, is constructed from the sampled frequencies ($\hat{s}_1$, $\hat{s}_2$, $\hat{s}_3$). Different samplings will give different manifolds, and therefore could provide better reduced models for different samplings. If a system designer had a certain frequency of importance $s^*$, a manifold could be constructed using sampled frequencies ($\hat{s}_1$, $\hat{s}_2$, $\hat{s}_3$) around $s^*$, and find the best approximation of the system by finding the closest boundary point to the original system.

VII. Conclusion

This paper demonstrates how both Balanced Truncation and Singular Perturbation Approximations can be viewed as a type of manifold boundary approximation. The key idea unifying these different techniques is to choose a canonical parameterization for any given system that partitions its parameters into identifiable and structural sets, eliminating conditionally identifiable parameters. Application of the Manifold Boundary Approximation Method, an information geometry technique that requires neither linearity nor time invariance, then illustrates that each technique finds approximations on different boundaries of the system’s model manifold embedded in an appropriate data space. Depending on the choice of metric, this manifold can give insight into which approximation is the best for a given application and give alternative approximations interpolating between BT and BSPA.

As we saw in Section III nothing about the MBAM approximation requires a linear model class, indeed, the Michaelis-Menten Reaction example was nonlinear. This fact, combined with the promising results for linear systems, suggests that by building an analogous parameterization for certain classes of nonlinear systems, similar types of approximations may be obtainable. This framework, then, focuses attention on obtaining good parameterizations of nonlinear systems in order to recover approximations with desirable qualities similar to those of BT and BSPA, questions that can be explored in future work. The work in [43–50] explores extending balanced truncation to nonlinear systems, providing one possible reduced model. Alternatively, after the cost of constructing a parameterization, the MBAM approach, as illustrated in Section VII would provide a whole class of possible reduced models, extending BT and BSPA to nonlinear systems and providing a spectrum of reduced-order systems interpolating between the two.

References

[1] M. K. Transtrum and P. Qiu, “Model reduction by manifold boundaries,” Phys. Rev. Lett., vol. 113, Aug 2014.
[2] S. H. Strogatz, “Exploring complex networks,” Nature, vol. 410, no. 6825, pp. 268–276, 2001.
[3] M. O. Jackson and A. Wolfinsky, “A strategic model of social and economic networks,” Journal of economic theory, vol. 71, no. 1, 1996.
[4] L. H. Hartwell, J. J. Hopfield, S. Leibler, and A. W. Murray, “From molecular to modular cell biology,” Nature, vol. 402, 1999.
[5] U. S. Bhalla and R. Iyengar, “Emergent properties of networks of biological signaling pathways,” Science, vol. 283, no. 5400, 1999.
[6] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A.-L. Barabási, “The large-scale organization of metabolic networks,” Nature, vol. 407, no. 6804, pp. 651–654, 2000.
[7] A. Broder, R. Kumar, P. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener, “Graph structure in the web,” Computer Networks, vol. 33, 2000.
[8] A. C. Antoulas, Approximation of large-scale dynamical systems. SIAM, 2005.
[9] P. Benner, S. Gugercin, and K. Willcox, “A survey of projection-based model reduction methods for parametric dynamical systems,” SIAM review, vol. 57, no. 4, pp. 483–531, 2015.
[10] B. Moore, “Persistence of excitation in extended least squares,” Automatic Control, IEEE Transactions on, vol. 28, no. 1, 1983.
[11] R. R. Bitmead, “Persistence of excitation conditions and the convergence of adaptive schemes,” Information Theory, IEEE Transactions on, vol. 30, no. 2, pp. 183–191, 1984.
[12] N. Shimkin and A. Feuer, “Persistence of excitation in continuous-time systems,” Systems & control letters, vol. 9, no. 3, 1987.
[13] T. Katayama, Subspace methods for system identification. Springer Science & Business Media, 2005.
[14] D. Uzunbay, P. Nath, J. J. Plata, G. L. Hart, I. Takeuchi, M. B. Nardelli, M. Fornari, and S. Curtarolo, “First principles thermodynamical modeling of the binodal and spinodal curves in lead chalcogenides,” Physical Chemistry Chemical Physics, vol. 18, no. 6, pp. 5005–5011, 2016.
[15] A. Frangi, G. Diion, and R. Sepulchin, “Modelling the modulation of neuronal bursting: a singularity theory approach,” SIAM Journal on Applied Dynamical Systems, vol. 13, no. 2, pp. 798–829, 2014.
[16] J. M. Epstein, Generative social science: Studies in agent-based computational modeling. Princeton University Press, 2006.
[17] V. Chetty, N. Woodbury, E. Vaziripour, and S. Warnick, “Vulnerability analysis for distributed and coordinated destabilization attacks,” in Conference on Decision and Control, Los Angeles, 2014.
[18] A. Cox, S. Roy, and S. Warnick, “A science of system security,” in Conference on Decision and Control, Los Angeles, CA, 2014, pp. 487–492.
[19] D. Grimsman, V. Chetty, N. Woodbury, E. Vaziripour, S. Roy, D. Zapalac, and S. Warnick, “A case study of a systematic attack design method for critical infrastructure cyber-physical systems,” in American Control Conference, Boston, MA, 2016.
[20] A. Teixeira, K. C. Sou, H. Sandberg, and K. H. Johansson, “Secure control systems: A quantitative risk management approach,” IEEE Control Systems, vol. 35, no. 1, pp. 24–45, 2015.
[21] Q. Zhu and T. Basar, “Game-theoretic methods for robustness, security, and resilience of cyberphysical control systems: games-in-games principle for optimal cross-layer resilient control systems,” IEEE Control Systems, vol. 35, no. 1, pp. 46–65, 2015.
[22] S. Amin, G. A. Schwartz, A. A. Cardenas, and S. S. Sastry, “Game-theoretic models of electricity theft detection in smart utility networks: Providing new capabilities with advanced metering infrastructure,” IEEE Control Systems, vol. 35, no. 1, pp. 66–81, 2015.
[23] B. Moore, “Principal component analysis in linear systems: Controllability, observability, and model reduction,” IEEE Transactions on Automatic Control, vol. 26, no. 1, pp. 17–32, 1981.
[24] L. Pernebo and L. Silverman, “Model reduction via balanced state space representations,” IEEE Transactions on Automatic Control, vol. 27, no. 2, pp. 382–387, 1982.
[25] G. E. P. Box and G. M. Jenkins, Time series analysis: Forecasting and control. Holden-Day, Oakland, CA, 1970.
[26] J. M. Epstein, Generative social science: Studies in agent-based computational modeling. Princeton University Press, 2006.
[27] T. Kato, Perturbation Theory for Linear Operators. Springer, 1966.
[28] P. G. Kokotovic, H. K. Khalil, and J. O’Reilly, Singular perturbation methods in control: analysis and design. Singapore, 1999.
[29] P. E. Paré, A. T. Wilson, M. K. Transtrum, and S. C. Warnick, “A unified view of balanced truncation and singular perturbation approximations,” in American Control Conference (ACC), 2015. IEEE, 2015, pp. 1989–1994.
[30] H. Jardón-Kojakhmetov, H. W. Broer, and R. Roussarie, “Analysis of a slow–fast system near a cusp singularity,” Journal of Differential Equations, vol. 260, no. 4, pp. 3785–3843, 2016.
[31] H. Jardón-Kojakhmetov, J. Scherpen, and D. del Puerto-Flores, “Stabilization of slow-fast systems at fold points,” arXiv preprint arXiv:1704.05654, 2017.
[32] K. Fernando and H. Nicholson, “Singular perturbation model reduction of balanced systems,” IEEE Transactions on Automatic Control, vol. 27, no. 2, 1982.
33. “Singular perturbational model reduction in the frequency domain,” IEEE Transactions on Automatic Control, vol. 27, 1982.
34. “Singular perturbational approximations for discrete-time balanced systems,” IEEE Transactions on Automatic Control, vol. 28, no. 2, pp. 240–242, 1983.
35. Y. Liu and B. D. Anderson, “Singular perturbation approximation of balanced systems,” International Journal of Control, vol. 50, no. 4, pp. 1379–1405, 1989.
36. L. Andersson, A. Rantzer, and C. Beck, “Model comparison and simplification,” International Journal of Robust and Nonlinear Control, vol. 9, pp. 157–181, 1999.
37. K. Glover, “All optimal Hankel-norm approximations of linear multivariable systems and their H∞-error bounds,” International Journal of Control, vol. 39, no. 6, 1984.
38. D. F. Enns, “Model reduction with balanced realizations: An error bound and a frequency weighted generalization,” in Proc. 23rd IEEE Conference on Decision and Control (CDC), 1984.
39. D. Enns, “Model reduction for control system design,” Ph. D Dissertation, Department of Aeronautics and Astronautics, Stanford university, 1984.
40. C. L. Beck, J. Doyle, and K. Glover, “Model reduction of multidimensional and uncertain systems,” IEEE Transactions on Automatic Control, vol. 41, 1996.
41. S. Lall and C. Beck, “Model reduction of complex systems in the linear-fractional framework,” in Computer aided control system design. IEEE, 1999, pp. 34–39.
42. C. Beck, “Coprime factors reduction methods for linear parameter varying and uncertain systems,” Systems & control letters, vol. 55, no. 3, pp. 199–213, 2006.
43. C. S. Hsu, “Decomposition of large scale systems—a numerical approach,” in Proc. American Control Conference (ACC), 1983.
44. K. Zhou, C. D’Souza, and J. R. Cloutier, “Structurally balanced controller order reduction with guaranteed closed loop performance,” Systems & control letters, vol. 24, no. 4, pp. 235–242, 1995.
45. A. Vandendorpe and P. Van Dooren, “On model reduction of interconnected systems,” in Proceedings International Symposium Math. Th. New. Syst., Belgium, 2004.
46. L. Li and F. Paganini, “Structured coprime factor model reduction based on LMIs,” Automatica, vol. 41, 2005.
47. H. Sandberg and R. M. Murray, “Model reduction of interconnected linear systems,” Optimal Control Applications and Methods, vol. 30, no. 3, pp. 225–245, 2009.
48. J. M. Scherpen, “Balancing for nonlinear systems,” Systems & Control Letters, vol. 21, no. 2, 1993.
49. ——, “H∞ balancing for nonlinear systems,” International Journal of Robust and Nonlinear Control, vol. 6, 1996.
50. S. Lall, J. E. Marsden, and S. Glavas˘ki, “A subspace approach to balanced truncation for model reduction of nonlinear control systems,” International Journal of Robust and Nonlinear Control, vol. 12, 2002.
51. V. Chetty and S. Warnick, “Meanings and applications of structure in networks of dynamic systems,” in Principles of Cyber-Physical Systems, S. Roy and S. K. Das, Eds. Oxford: Cambridge University Press, arXiv:1406.1844 [cs.SY].
52. M. Transtrum, G. Hart, and P. Qiu, “Information topology identifies emergent model classes,” arXiv, Sept 2014.
53. J. R. Leis and M. A. Kramer, “The simultaneous solution and sensitivity analysis of systems described by ordinary differential equations,” ACM Transactions on Mathematical Software (TOMS), vol. 14, no. 1, pp. 45–60, 1988.
54. L. B. Rall, “Automatic differentiation: Techniques and applications,” 1981.
55. H. M. Bucker, G. Corliss, P. Hovland, U. Naumann, and B. Norris, Automatic differentiation: applications, theory, and implementations. Springer Science & Business Media, 2006, vol. 50.
56. M. Transtrum, B. B. Machta, and J. P. Sethna, “Why are nonlinear fits to data so challenging?” Physical Review Letters, vol. 104, 2010.
57. ——, “Geometry of nonlinear least squares with applications to sloppy models and optimization,” Physical Review E, 2011.
58. M. Spivak, A comprehensive introduction to differential geometry. Publish or Perish, 1979.
59. C. Misner, K. Thorne, and J. Wheeler, Gravitation. WH Freeman and Company, 1973.
60. L. Michaelis and M. M. L. Menten, “Die kinetik der invertinwirkung,” Biochem Z, vol. 49, pp. 333–369, 1913.
61. E. Briggs and J. B. S. Haldane, “A note on the kinetics of enzyme action,” Biochem Journal, vol. 19, pp. 338–339, 1925.
62. B. P. English, W. Min, A. M. van Oijen, K. T. Lee, G. Luo, H. Sun, B. J. Cherayil, S. Kou, and X. S. Xie, “Ever-fluctuating single enzyme molecules: Michaelis-menten equation revisited,” Nature chemical biology, vol. 2, no. 2, pp. 87–94, 2005.
63. M. Transtrum, “MBAM GitHub repository.” [Online]. Available: https://github.com/mktranstrum/MBAM
64. E. G. Gilbert, “Controllability and observability in multivariable control systems,” Journal of the Society for Industrial and Applied Mathematics, Series A: Control, vol. 1, no. 2, pp. 128–151, 1963.
65. R. Bellman and K. J. Åström, “On structural identifiability,” Mathematical biosciences, vol. 7, no. 3–4, pp. 329–339, 1970.
66. P. E. Paré, “Necessary and Sufficient Conditions for State-Space Network Realization,” Master’s thesis, Brigham Young University, Provo, UT, 2014.
67. S. C. Smith and J. Fisher, “On generating random systems: a gramician approach,” in American Control Conference, 2003. Proceedings of the 2003, vol. 3, IEEE, 2003, pp. 2743–2748.
68. F. Zhang, The Schur complement and its applications. Springer, 2006.

Philip E. Paré received his B.S. in Mathematics with University Honors and his M.S. in Computer Science from Brigham Young University, Provo, UT, in 2012 and 2014, respectively, and his Ph.D. in Electrical and Computer Engineering from the University of Illinois at Urbana-Champaign, Urbana, IL in 2018. He was the recipient of the 2017-2018 Robert T. Chien Memorial Award for excellence in research and named a 2017-2018 College of Engineering Mavis Future Faculty Fellow. His research interests include the modeling and control of dynamic networked systems, model reduction techniques, and time-varying systems.

David Grimsman is currently a Ph.D. candidate in the ECE Department at the University of California at Santa Barbara, where he is a recipient of the NSF Network Science IGERT Fellowship. He received a B.S. in Electrical Engineering from Brigham Young University in 2006 as a Heritage Scholar, and after some work in industry returned to academia to earn an M.S. in Computer Science from Brigham Young University in 2016. His research interests include distributed control, network systems, game theory, optimization, and algorithm design.

Alma Wilson is a postdoctoral fellow in the CS Department at Brigham Young University. He received a B.Sc. in Physics and Mathematics from the University of Auckland, where he was a New Zealand University Junior Scholar and a Ngarimu V.C. Scholar, and he subsequently took a Ph.D. from Brigham Young University in 2000. After work in industry, with stints in Arizona, Maryland, Australia, and Utah, he returned to academic life. He is interested in the role of extensive properties in control theory, information theory, quantum theory, and thermodynamics, and he is particularly fond of constructing pedagogically advantageous representations of familiar STEM material.
Mark K. Transtrum received his M.S. and Ph.D. degrees in physics from Cornell University in 2010 and 2011 respectively. Previously, he received his B.S. in Physics from Brigham Young University in 2006. After graduate school, he studied computational biology as a Postdoctoral Fellow at MD Anderson Cancer Center. Since 2013, he has been on the faculty of the Department of Physics & Astronomy at Brigham Young University. His work primarily uses information theory to study properties of mathematical models drawn from a variety of complex systems including power systems, systems biology, materials science, and neuroscience.

Sean Warnick has been with the faculty of the Computer Science Department at Brigham Young University since 2003. He received his Ph.D. and S.M. degrees from the Massachusetts Institute of Technology in 1995 and 2003, respectively, and his B.S.E. from Arizona State University in 1993. He attended ASU on scholarship from the Flinn Foundation, graduated summa cum laude with honors, and was named the Outstanding Graduate of the College of Engineering and Applied Sciences. He has also held visiting positions from Cambridge University (2006), the University of Maryland at College Park (2008), and the University of Luxembourg’s Centre for Systems Biomedicine (2014). Sean was named the Distinguished Visiting Professor by the National Security Agency three years in a row, 2008-2010, for his work with their Summer Program for Operations Research Technology, and he has consulted with various companies. Sean’s research interests focus on representations of complex networks of dynamic systems.