Model Reduction of Power Networks

Bita Safaee

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Mechanical Engineering

Serkan Gugercin, Co-chair
Pablo Tarazaga, Co-chair
Pinar Acar
Christopher Beattie
Andrew J. Kurdila

May 10, 2022
Blacksburg, Virginia

Keywords: Model Reduction, Power Networks, Parametric Model Reduction, $\mathcal{H}_2$ Model Reduction, Data Driven Modeling

Model Reduction of Power Networks

Bita Safaee

(ABSTRACT)

A power grid network is an interconnected network of coupled devices that generate, transmit and distribute power to consumers. These complex and usually large-scale systems have high dimensional models that are computationally expensive to simulate especially in real time applications, stability analysis, and control design. Model order reduction (MOR) tackles this issue by approximating these high dimensional models with reduced high-fidelity representations. When the internal description of the models is not available, the reduced representations are constructed by data. In this dissertation, we investigate four problems regarding the MOR and data-driven modeling of the power networks model, particularly the swing equations.

We first develop a parametric MOR approach for linearized parametric swing equations that preserves the physically-meaningful second-order structure of the swing equations dynamics. Parameters in the model correspond to variations in operating conditions. We employ a global basis approach to develop the parametric reduced model. We obtain these local bases by $\mathcal{H}_2$-based interpolatory model reduction and then concatenate them to form a global basis. We develop a framework to enrich this global basis based on a residue analysis to ensure bounded $\mathcal{H}_2$ and $\mathcal{H}_\infty$ errors over the entire parameter domain.

Then, we focus on nonlinear power grid networks and develop a structure-preserving system-theoretic model reduction framework. First, to perform an intermediate model reduction step, we convert the original nonlinear system to an equivalent quadratic nonlinear model via a lifting transformation. Then, we employ the $\mathcal{H}_2$-based model reduction approach,

Quadratic Iterative Rational Krylov Algorithm (Q-IRKA). Using a special subspace structure of the model reduction bases resulting from Q-IRKA and the structure of the underlying power network model, we form our final reduction basis that yields a reduced model of the same second-order structure as the original model.

Next, we focus on a data-driven modeling framework for power network dynamics by applying the Lift and Learn approach. Once again, with the help of the lifting transformation, we lift the snapshot data resulting from the simulation of the original nonlinear swing equations such that the resulting lifted-data corresponds to a quadratic nonlinearity. We then, project the lifted data onto a lower dimensional basis via a singular value decomposition. By employing a least-squares measure, we fit the reduced quadratic matrices to this reduced lifted data. Moreover, we investigate various regularization approaches.

Finally, inspired by the second-order sparse identification of nonlinear dynamics (SINDY) method, we propose a structure-preserving data-driven system identification method for the nonlinear swing equations. Using the special structure on the right-hand-side of power systems dynamics, we choose functions in the SINDY library of terms, and enforce sparsity in the SINDY output of coefficients.

Throughout the dissertation, we use various power network models to illustrate the effectiveness of our approaches.

Model Reduction of Power Networks

Bita Safaee

(GENERAL AUDIENCE ABSTRACT)

Power grid networks are interconnected networks of devices responsible for delivering electricity to consumers, e.g., houses and industries for their daily needs. There exist mathematical models representing power networks dynamics that are generally nonlinear but can also be simplified by linear dynamics. Usually, these models are complex and large-scale and therefore take a long time to simulate. Hence, obtaining models of much smaller dimension that can capture the behavior of the original systems with an acceptable accuracy is a necessity. In this dissertation, we focus on approximation of power networks model through the swing equations. First, we study the linear parametric power network model whose operating conditions depend on parameters. We develop an algorithm to replace the original model with a model of smaller dimension and the ability to perform in different operating conditions. Second, given an explicit representation of the nonlinear power network model, we approximate the original model with a model of the same structure but smaller dimension. In the cases where the mathematical models are not available but only time-domain data resulting from simulation of the model is at hand, we apply an already developed framework to infer a model of a small dimension and a specific nonlinear structure: quadratic dynamics. In addition, we develop a framework to identify the nonlinear dynamics while maintaining their original physically-meaningful structure.

Dedication

To my wonderful parents, Minoo and Ali.

To my beloved husband, Payam.

Acknowledgments

I would like to express my sincere gratitude to my advisor, Dr. Serkan Gugercin without whom this thesis would be impossible. I will be forever grateful for his understanding, patience and dedication to mentor me with his vast knowledge and encouragement. Without doubt, he exemplifies the qualities of an incredible advisor. I would like to thank Dr. Vasileios Kekatos for his valuable discussions and deep insights in the power systems area. Equal in measure is my gratitude for my committee members Dr. Pablo Tarazaga, Dr. Christopher Beattie, Dr. Pinar Acar and Dr. Andrew J. Kurdila for their valuable feedback and insights. In addition, I would like to give a heartfelt thank you to my family and friends for their supports and friendship. Especially, to my mom for believing in me, her unconditional love and supports in every possible way, and to my husband for making my life wonderful and brighter every single day.

Contents

List of Figures

List of Tables

List of Abbreviations

$\circ$	Hadamard product
$\mathbf{1}_n$	Vector of dimension n with all entries equal to one
$\mathcal{H}^{(2)}$	mode-2 matricization
$\widetilde{\otimes}$	Permuted form of Kronecker product
$\widehat{\otimes}$	Kronecker product without the redundant/repeated terms
$\otimes$	Kronecker product
σ	Frequency interpolation point
orth	Orthonormal basis
tr$(\cdot)$	Trace of the argument
vec$(\cdot)$	Vectorization of a matrix
e_i	ith unit vector
$H(s)$	Original model (single variable)
$H(s,p)$	Original model (two variables)
$H_r(s)$	Reduced model of $H(s)$
$H_r(s,p)$	Reduced model of $H(s,p)$
I_n	Identity matrix of dimension $n \times n$

DEIM Discrete Empirical Interpolation Method

EN Effective Network

MIMO Multi-Input/Multi-Output

MOR Model Order Reduction

p Parameter vector

PCA Principal Component Analysis

PMOR Parametric MOR

POD Proper Orthogonal Decomposition

Q-IRKA Quadratic Iterative Rational Krylov Algorithm

QBT Quadratic Balanced Truncation

ROM Reduced Order Model

s Frequency variable

SISO Single-Input/Single-Output

SM Synchronous Machine

SOR-IRKA Second Order IRKA

SVD Singular Value Decomposition

Chapter 1

Introduction

Power grid networks are networks of electrically coupled devices that play a fundamental role in transferring power from generators to consumers. A power network grid can be viewed as a network of nodes and links [90]. A node (bus) is a point where power is injected by a generator (generator node) or consumed by a load (load node) or even redistributed through branches of the transmission lines. A link is considered as a transmission line, i.e., an electrical connection between the two nodes. In Figure 1.1, an example of a power grid networks, the IEEE 14-bus system which includes 14 buses, 5 generators, and 11 loads, is depicted.

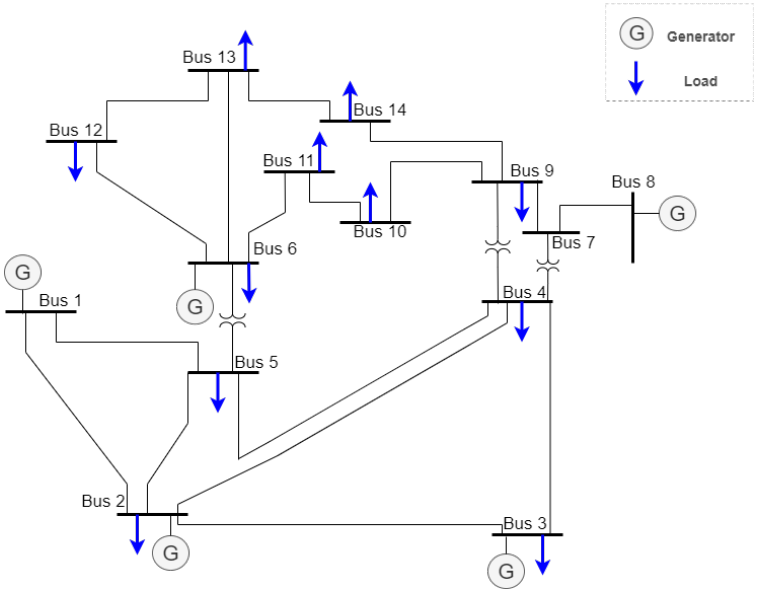


Figure 1.1: IEEE 14-bus

In the following sections, we introduce some of the leading models of power network grids as well as some concepts and terminology used in power systems analysis.

1.1 Power grid network model

The coupled dynamics of power generators plays an important role in determining the synchronization stability of power generators, i.e., the power network fails to properly operate if a single generator operates at a different frequency. Therefore, a good network model capable of describing the coupled dynamics of power generators is of a great importance. There are three most commonly used models for describing a dynamical model of a power grid network: *synchronous motor* (SM), *effective network* (EN) and *structure-preserving* (SP). Each model is described as a network of n coupled oscillators (generators or loads) whose dynamics at the ith node is expressed by the nonlinear second-order equation [90]

$$\frac{2J_i}{\omega_R}\ddot{\delta}_i + \frac{D_i}{\omega_R}\dot{\delta}_i + \sum_{\substack{j=1 \\ j\neq i}}^{n} K_{ij}\sin(\delta_i - \delta_j - \gamma_{ij}) = B_i, \tag{1.1}$$

where δ_i is the angle of rotation for the ith oscillator; J_i and D_i are inertia and damping constants, respectively; ω_R is the angular frequency for the system; $K_{ij} \geq 0$ is dynamical coupling between oscillator i and j; and γ_{ij} is the phase shift in this coupling. Constants B_i, K_{ij} and γ_{ij} are computed by solving the power flow equations and applying Kron reduction [90]. The three models (SM, EN, and SP) mainly differ by the way they model loads which leads to different values and interpretations of the parameters B_i, K_{ij} and γ_{ij}. The EN model assumes loads as constant impedances instead of oscillators. The SP model represents loads as first-order oscillators ($J_i = 0$) while the SM model represent loads as synchronous motors that are expressed as second-order oscillators. In this thesis, we focus on the EN and SM

models in [90] that will be explained in more detail in Sections 1.1.1 and 1.1.2 .

Define the state-vector

$$\delta = [\delta_1 \ \delta_2 \ \dots \ \delta_n]^T \in \mathbb{R}^n.$$

Then, the dynamics of a network of n coupled oscillators, given in (1.1) for the ith node, can be described in the system form

$$\mathcal{M}\ddot{\delta}(t) + \mathcal{D}\dot{\delta}(t) + f(\delta) = \mathcal{B}u(t), \tag{1.2}$$

$$y(t) = \mathcal{C}\delta(t), \tag{1.3}$$

where the matrices $\mathcal{B} \in \mathbb{R}^n$, $\mathcal{M} \in \mathbb{R}^{n\times n}$, and $\mathcal{D} \in \mathbb{R}^{n\times n}$ are defined as

$$\begin{aligned} \mathcal{B} &= \begin{bmatrix} B_1, & B_2, & \dots, & B_n \end{bmatrix}^T, \\ \mathcal{M} &= \operatorname{diag}\left(\frac{2J_1}{\omega_R}, \frac{2J_2}{\omega_R}, \ \dots, \frac{2J_n}{\omega_R} \right), \\ \mathcal{D} &= \operatorname{diag}\left(\frac{D_1}{\omega_R}, \frac{D_2}{\omega_R}, \dots, \frac{D_n}{\omega_R} \right); \end{aligned} \tag{1.4}$$

$u(t) = 1$ for $t > 0$ is input to the dynamics; and the nonlinear map $f : \mathbb{R}^n \to \mathbb{R}^n$ is defined such that its ith component is

$$(f(\delta))_i = \sum_{\substack{j=1 \\ j\neq i}}^{n} K_{ij} \sin(\delta_i - \delta_j - \gamma_{ij}), \quad i = 1, 2, \dots, n \ . \tag{1.5}$$

The variable $y(t) \in \mathbb{R}^p$ in (1.3) corresponds to the output (quantity of interest) of the networks dynamics, described by the state-to-output matrix $\mathcal{C} \in \mathbb{R}^{p\times n}$.

1.1.1 Effective network model

Before we go into the details of the EN model, we need to explain some basic concepts of a power network.

Recall that a power network grid is a network of nodes and links where links represent the transmission lines. Transmission lines in power system analysis are modeled by the so-called π model shown in the grey box of Figure 1.2. We refer to the admittance $\mathsf{j}\frac{b_{ij}}{2}$ as the shunt admittance to the ground, where j is the imaginary unit, i.e., $\mathsf{j}^2 = -1$. For more details, we refer the reader to [90] and the references therein.

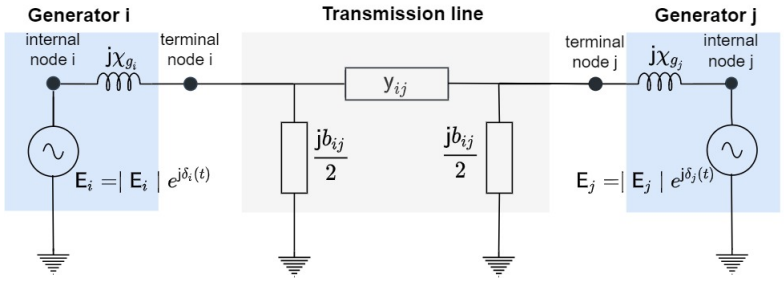


Figure 1.2: Electric circuit representation of a network with two generators

The structure of the physical network is represented by the admittance matrix Y in (1.6), where the off-diagonal element $\mathsf{Y}(i,j)$ is the negative of the admittance y_{ij} between nodes i and $j \neq i$, and the diagonal element $\mathsf{Y}(i,i)$ is the summation of all the connected admittances to node i including the shunt admittance to the ground.

$$\mathsf{Y}(i,j) := \begin{cases} -\mathsf{y}_{ij}, & \text{if } \exists \text{ line}(i,j), \\ \sum_{k\neq i} \mathsf{y}_{ik} + \mathsf{j}\frac{b_{ik}}{2}, & \text{if } j = i, \\ 0, & \text{otherwise} \end{cases} . \tag{1.6}$$

The sparse and symmetric matrix $\mathsf{Y}(i,j)$ can be represented in the polar coordinate as

$\mathsf{Y}(i,j) = \left|\mathsf{Y}(i,j)\right| e^{\mathsf{j}\alpha_{ij}}$ with magnitude $\left|\mathsf{Y}(i,j)\right|$ and phase angle α_{ij}.

In a network with alternating current, power at node i is a complex number with real and complex parts given by

$$\mathsf{S}_i = \mathsf{V}_i \mathsf{I}_i^* = \mathsf{P}_i + \mathsf{j}\mathsf{Q}_i, \tag{1.7}$$

where I_i^* denotes the complex conjugate of the current I_i, $\mathsf{V}_i = |\mathsf{V}_i| e^{\mathsf{j}\phi_i}$ is the complex voltage (with magnitude $|\mathsf{V}_i|$ and phase angle ϕ_i), and P_i and Q_i are, respectively, the active power and the reactive power at node i. Let n be the number of the nodes. Then, using the admittance matrix Y in (1.6) and *Ohm's law*, we can write the current at node i as

$$\mathsf{I}_i = \sum_{j=1}^{n} \mathsf{Y}(i,j)\mathsf{V}_j \ . \tag{1.8}$$

Plugging (1.8) into (1.7) yields

$$\begin{aligned} \mathsf{S}_i &= \mathsf{V}_i \sum_{j=1}^{n} \big(\mathsf{Y}(i,j)\big)^* \mathsf{V}_j^* = \sum_{j=1}^{n} \left|\mathsf{Y}(i,j)\mathsf{V}_j\mathsf{V}_i\right| e^{\mathsf{j}(\phi_i - \phi_j - \alpha_{ij})} \\ &= \sum_{j=1}^{n} \left|\mathsf{Y}(i,j)\mathsf{V}_j\mathsf{V}_i\right| (\sin(\phi_i - \phi_j - \gamma_{ij}) - \mathsf{j}\cos(\phi_i - \phi_j - \gamma_{ij})) \\ &= \mathsf{P}_i + \mathsf{j}\mathsf{Q}_i, \end{aligned} \tag{1.9}$$

where $\gamma_{ij} := \alpha_{ij} - \frac{\pi}{2}$, and

$$\mathsf{P}_i = \sum_{j=1}^{n} \left|\mathsf{Y}(i,j)\mathsf{V}_j\mathsf{V}_i\right| \sin(\phi_i - \phi_j - \gamma_{ij}), \quad i = 1, \ldots, n \quad \text{and} \tag{1.10}$$

$$\mathsf{Q}_i = -\sum_{j=1}^{n} \left|\mathsf{Y}(i,j)\mathsf{V}_j\mathsf{V}_i\right| \cos(\phi_i - \phi_j - \gamma_{ij}) \ , \quad i = 1, \ldots, n \tag{1.11}$$

are called the power flow equations that are fundamental for steady state power systems analysis. As one can see from (1.10)-(1.11), there are a set of $2n$ nonlinear equations to solve for $4n$ unknowns P_i, Q_i, $|\mathsf{V}_i|$ and ϕ_i for $i = 1, \ldots, n$. To overcome this issue, we assume that two of the four real unknowns per node are given. This assumption depends on the type of the nodes: if node i is a generator node, we assume P_i and $|\mathsf{V}_i|$ are given as constants $\mathsf{P}_i = \mathsf{P}^\star_{g_i}$ and $|\mathsf{V}^\star_i|$. If node i is a load node, we assume $\mathsf{P}_i = -\mathsf{P}^\star_{l_i}$ and $\mathsf{Q}_i = -\mathsf{Q}^\star_{l_i}$. Given these assumptions and the admittance matrix Y, the power flow equations (1.10)-(1.11) can be numerically solved by a range of the available softwares such as MATHPOWER [133] and Power System Toolbox (PST) [48].

The equation of motion for a generator i results from the Newton's second law and can be expressed by the so-called swing equation [90]

$$\frac{2J_i}{\omega_R}\ddot{\delta}_i + \frac{D_i}{\omega_R}\dot{\delta}_i = P_{m_i} - P_{e_i}, \tag{1.12}$$

where P_{m_i} (in per unit (p.u.)) is the mechanical input power to the generator and it is assumed to be constant, i.e., $P_{m_i} = P^\star_g$, and P_{e_i} (in p.u.) is the output electrical power demanded by the network.

Now going back to our problem, the EN model or the network-reduced model defines loads as constant impedances to the ground, i.e., nodes without current injections. Hence, it reduces the physical network to an equivalent network of interactions between generators by eliminating all nodes except the generators' internal node via Kron reduction. Towards this goal, let n_g and n_l denote the number of generator internal nodes and load nodes, respectively. Recall the admittance matrix Y in (1.6) for the physical network and decomposed it into four block matrices resulted from separating the columns (rows) of the generators nodes and

the columns (rows) of the load nodes.

$$\mathsf{Y} = \begin{bmatrix} \mathsf{Y}_{11} & \mathsf{Y}_{12} \\ \mathsf{Y}_{21} & \mathsf{Y}_{22} \end{bmatrix} \in \mathbb{C}^{(n_g+n_l)\times(n_g+n_l)}, \tag{1.13}$$

where $\mathsf{Y}_{11} \in \mathbb{C}^{n_g \times n_g}$, $\mathsf{Y}_{12} \in \mathbb{C}^{n_g \times n_l}$,$\mathsf{Y}_{21} \in \mathbb{C}^{n_l \times n_g}$ and $\mathsf{Y}_{22} \in \mathbb{C}^{n_l \times n_l}$. Also, let us define additional n_g nodes, namely the generators terminal nodes, as points between the generators internal nodes and transient reactances (see Figure 1.2) with the corresponding admittance Y_g formed as

$$\mathsf{Y}_g = \mathrm{diag}(\frac{1}{\mathsf{j}\chi_{g_1}}, \ldots, \frac{1}{\mathsf{j}\chi_{g_{n_g}}}) \in \mathbb{C}^{n_g \times n_g} \ , \tag{1.14}$$

where χ_{g_i} is the transient reactance for generator i. In a steady state, the consumed reactive power $\mathsf{Q}^\star_{l_i}$ and the active power at $\mathsf{P}^\star_{l_i}$ at a load node i (and a generator terminal node) is represented by a constant impedance to the ground with an equivalent admittance

$$\mathsf{Y}_{l_i} = \frac{\mathsf{P}^\star_{l_i} - \mathsf{j}\mathsf{Q}^\star_{l_i}}{|\mathsf{V}^\star_i|^2} \ , \tag{1.15}$$

where the voltage magnitude $|\mathsf{V}^\star_i|$ is obtained by plugging $\mathsf{P}^\star_{l_i}$ and $\mathsf{Q}^\star_{l_i}$ into the power flow equations (1.10)-(1.11). The constant Y_{l_i} is then added to the diagonal elements of of the admittances Y_{11} and Y_{22}. We denote the resulted admittances as $\hat{\mathsf{Y}}_{11}$ and $\hat{\mathsf{Y}}_{22}$. Note that in this representation, we assumed a constant power demand, which is a valid assumption on a short time scale. Now we are ready to form an augmented admittance matrix $\hat{\mathsf{Y}}$ which also contains the admittance of the transient reactances between the generators' internal and

terminal nodes, and the equivalent impenadces for the loads

$$\hat{\mathsf{Y}} := \begin{bmatrix} \mathsf{Y}_g & -\mathsf{Y}_g & 0_{n_g \times n_l} \\ -\mathsf{Y}_g & \hat{\mathsf{Y}}_{11} + \mathsf{Y}_g & \mathsf{Y}_{12} \\ 0_{n_l \times n_g} & \mathsf{Y}_{21} & \hat{\mathsf{Y}}_{22} \end{bmatrix} \in \mathbb{C}^{(2n_g+n_l)\times(2n_g+n_l)} \ . \tag{1.16}$$

Now let $\hat{\mathsf{V}} \in \mathbb{C}^{2n_g+n_l}$ be a unique voltage vector including voltages for all nodes (the generators internal and terminal loads and load nodes) as

$$\hat{\mathsf{V}} = \begin{bmatrix} \mathsf{V}_{int} \\ \mathsf{V}_t \\ \mathsf{V}_l \end{bmatrix} \ , \tag{1.17}$$

where $\mathsf{V}_{int} \in \mathbb{C}^{n_g}$ and $\mathsf{V}_t \in \mathbb{C}^{n_g}$ denote the voltage vectors of generators' internal nodes and generators' terminal nodes, respectively, and $\mathsf{V}_l \in \mathbb{C}^{n_l}$ is the voltage vector of load nodes. Since loads are defined as constant impedances to the ground, only generators' internal nodes have non-zero injection currents, which we denote by I_{int}. Knowing theses facts, we can form the current vector $\hat{\mathsf{I}}$ as

$$\hat{\mathsf{I}} = \begin{bmatrix} \mathsf{I}_{int} \\ \mathsf{I}_t \\ \mathsf{I}_l \end{bmatrix} = \begin{bmatrix} \mathsf{I}_{int} \\ 0_{n_g \times 1} \\ 0_{n_l \times 1} \end{bmatrix} . \tag{1.18}$$

From *Kirchhoff's current law*, we can form $\hat{\mathsf{I}} = \hat{\mathsf{Y}}\hat{\mathsf{V}}$ as

$$\begin{bmatrix} \mathsf{I}_{int} \\ 0_{n_g \times 1} \\ 0_{n_l \times 1} \end{bmatrix} = \begin{bmatrix} \mathsf{Y}_g & -\mathsf{Y}_g & 0_{n_g \times n_l} \\ -\mathsf{Y}_g & \hat{\mathsf{Y}}_{11} + \mathsf{Y}_g & \mathsf{Y}_{12} \\ 0_{n_l \times n_g} & \mathsf{Y}_{21} & \hat{\mathsf{Y}}_{22} \end{bmatrix} \begin{bmatrix} \mathsf{V}_{int} \\ \mathsf{V}_t \\ \mathsf{V}_l \end{bmatrix} . \tag{1.19}$$

Using the Kron reduction, i.e., reducing (1.19) to $\mathsf{I}_{int} = \mathsf{Y}^{EN}\mathsf{V}_{int}$ by eliminating V_t and V_l, we obtain the $n_g \times n_g$ admittance matrix of the effective network Y^{EN} as

$$\mathsf{Y}^{EN} := \tilde{\mathsf{Y}}(I_{n_g} + \mathsf{Y}_g^{-1}\tilde{\mathsf{Y}})^{-1} \quad \text{where} \quad \tilde{\mathsf{Y}} := \hat{\mathsf{Y}}_{11} - \mathsf{Y}_{12}\hat{\mathsf{Y}}_{22}^{-1}\mathsf{Y}_{21} \tag{1.20}$$

where $I_{n_g} \in \mathbb{R}^{n_g \times n_g}$ denotes the identity matrix. Note that Y_g is always invertible [90]. In addition, the existence of $\hat{\mathsf{Y}}_{22}^{-1}$ follows from the assumption of uniqueness of $\mathbb{V}$ [90]. Therefore, the existence of Y^{EN} depends on χ_{g_i}, for $i = 1, \ldots, n_g$, to be small. Due to the symmetry property of Y^{EN}, one can view the effective network as an electrically equivalent network where every distinct pair of the generator internal nodes i and j are connected by an effective admittance $-\mathsf{Y}^{EN}(i,j)$. We should note that all quantities must be expressed in p.u. with respect to the system base. Using Y^{EN}, the electrical output power P_{e_i} (1.12) is obtained by the so-called power angle equation as

$$\begin{aligned} P_{e_i} &= \sum_{j=1}^{n_g} \left|\mathsf{E}_i^\star \mathsf{E}_j^\star \mathsf{Y}^{EN}(i,j)\right| \cos(\delta_i - \delta_j + \alpha_{ij}) \\ &= |\mathsf{E}_i^*|^2 \mathsf{G}^{EN}(i,i) + \sum_{\substack{j=1 \\ j\neq i}}^{n_g} \left|\mathsf{E}_i^\star \mathsf{E}_j^\star \mathsf{Y}^{EN}(i,j)\right| \cos(\delta_i - \delta_j + \alpha_{ij}), \end{aligned} \tag{1.21}$$

where $\mathsf{Y}^{EN}(i,j) = \left|\mathsf{Y}^{EN}(i,j)\right| e^{\mathrm{j}\alpha_{ij}^{EN}}$, and G^{EN} is the conductance, i.e., the real part of the complex admittance Y^{EN}. By plugging (1.21) into the swing equation, we obtain the EN model with the same form in (1.1) as

$$\begin{aligned} &\frac{2J_i}{\omega_R}\ddot{\delta}_i + \frac{D_i}{\omega_R}\dot{\delta}_i + \sum_{\substack{j=1 \\ j\neq i}}^{n_g} K_{ij}^{EN} \sin(\delta_i - \delta_j - \gamma_{ij}^{EN}) = B_i^{EN}, \quad i = 1, \ldots, n_g \\ &B_i^{EN} := \mathsf{P}_{g,i}^* - |\mathsf{E}_i^*|^2 \mathsf{G}^{EN}(i,j), \quad K_{ij}^{EN} := \left|\mathsf{E}_i^* \mathsf{E}_j^* \mathsf{Y}^{EN}(i,j)\right|, \quad \gamma_{ij}^{EN} := \alpha_{ij}^{EN} - \frac{\pi}{2}. \end{aligned} \tag{1.22}$$

The steady-state internal voltage magnitude E_i^* for each generator is calculated as

$$(\mathsf{E}_i^*)^2 = \left(\frac{\mathsf{P}_{g,i}^* \chi_{g_i}}{|V_{t_i}{}^*|}\right)^2 + \left(|V_{t_i}{}^*| + \frac{\mathsf{Q}_{g,i}^* \chi_{g_i}}{|V_{t_i}^*|}\right)^2, \quad i = 1, \ldots, n_g. \tag{1.23}$$

1.1.2 Synchronous motor model

In the SM model, loads are modeled as synchronous motors. A synchronous motor is the same electrical device as a generator, but it translates electrical power into mechanical power, as opposed to a generator. Therefore, in the SM model, a load node i is exactly modeled by the swing equation (1.12) with $P_{m_i}, P_{e_i} < 0$. As in the previous case, to find the SM model representation, let n_g denote the number of generator internal nodes and n_l denote the number of synchronous motor internal nodes. Similar to the EN model where we defined additional n_g nodes for the generators terminal nodes, we define additional n_l nodes to represent the synchronous motors terminal nodes which makes the total number of nodes $2(n_l + n_g)$.

For simplicity of the notation, let $n = n_l + n_g$. Then, the admittance matrix corresponding to the transient reactances of both generators and synchronous motors can be formed similar to (1.14) as

$$\tilde{\mathsf{Y}}_g = \text{diag}\left(\frac{1}{\mathsf{j}\chi_{g_1}}, \ldots, \frac{1}{\mathsf{j}\chi_{g_n}}\right) \in \mathbb{C}^{n \times n} . \tag{1.24}$$

Also, (1.16) will be replaced by

$$\tilde{\mathsf{Y}} := \begin{bmatrix} \tilde{\mathsf{Y}}_g & -\tilde{\mathsf{Y}}_g \\ -\tilde{\mathsf{Y}}_g & \mathsf{Y} + \tilde{\mathsf{Y}}_g \end{bmatrix} \in \mathbb{C}^{2n \times 2n} , \tag{1.25}$$

where Y is the admittance matrix representing the physical network as in (1.13). Now let

$$\tilde{V} = \begin{bmatrix} \tilde{\mathsf{V}}_{in} \\ \tilde{\mathsf{V}}_t \end{bmatrix} \in \mathbb{C}^{2n} \tag{1.26}$$

be the voltage vector including the voltages for both generators and motors internal nodes $\tilde{\mathsf{V}}_{in} \in \mathbb{C}^n$ and terminal nodes $\tilde{\mathsf{V}}_t \in \mathbb{C}^n$. Following the same argument as in the EN model and using the *Kirchhoff's current law*, we obtain

$$\begin{bmatrix} \tilde{\mathsf{I}}_{in} \\ 0 \end{bmatrix} = \begin{bmatrix} \tilde{\mathsf{Y}}_g & -\tilde{\mathsf{Y}}_g \\ -\tilde{\mathsf{Y}}_g & \mathsf{Y} + \tilde{\mathsf{Y}}_g \end{bmatrix} \begin{bmatrix} \tilde{\mathsf{V}}_{in} \\ \tilde{\mathsf{V}}_t \end{bmatrix}. \tag{1.27}$$

As in the previous case, we use Kron reduction to remove all the terminal nodes of generators and motors and reduced (1.27) to $\tilde{\mathsf{I}}_{in} = \mathsf{Y}^{SM}\tilde{V}_{in}$, with the $n \times n$ admittance matrix Y^{SM} defined as

$$\mathsf{Y}^{SM} = \tilde{\mathsf{Y}}_g(I_n - (\mathsf{Y} + \tilde{\mathsf{Y}}_g)^{-1}\tilde{\mathsf{Y}}_g). \tag{1.28}$$

Using Y^{SM}, the electrical output power P_{e_i} (1.12) is now formed as

$$\begin{aligned} P_{e_i} &= \sum_{j=1}^{n} \left| \mathsf{E}_i^\star \mathsf{E}_j^\star \mathsf{Y}^{SM}(i,j) \right| \cos(\delta_i - \delta_j + \alpha_{ij}) \\ &= |\mathsf{E}_i^*|^2 \mathsf{G}^{SM}(i,i) + \sum_{\substack{j=1 \\ j \neq i}}^{n} \left| \mathsf{E}_i^\star \mathsf{E}_j^\star \mathsf{Y}^{SM}(i,j) \right| \cos(\delta_i - \delta_j + \alpha_{ij}). \end{aligned} \tag{1.29}$$

Once again, by plugging (1.29) into the swing equation, we obtain the SM model that has

the same form in (1.1) as

$$\frac{2J_i}{\omega_R}\ddot{\delta}_i + \frac{D_i}{\omega_R}\dot{\delta}_i + \sum_{\substack{j=1\\ j\neq i}}^{n} K_{ij}^{SM}\sin(\delta_i - \delta_j - \gamma_{ij}^{SM}) = B_i^{SM}, \quad i = 1,\ldots,n \tag{1.30}$$

$$\mathsf{Y}^{SM}(i,j) = \left|\mathsf{Y}^{SM}(i,j)\right| e^{\mathrm{j}\alpha_{ij}^{SM}}, \quad K_{ij}^{SM} := \left|\mathsf{E}_i^*\mathsf{E}_j^* Y^{SM}(i,j)\right|, \quad \gamma_{ij}^{SM} := \alpha_{ij}^{SM} - \frac{\pi}{2},$$

where

$$B_i^{SM} := \mathsf{P}_{g,i}^* - |\mathsf{E}_i^*|^2\mathsf{G}^{SM}(i,i) \quad \text{for generators} \quad i = 1,\ldots,n_g \quad \text{and}$$
$$B_i^{SM} := -\mathsf{P}_{l,i}^* - |\mathsf{E}_i^*|^2\mathsf{G}^{SM}(i,i) \quad \text{for loads} \quad i = n_g+1,\ldots,n.$$

The steady-state internal voltage magnitude E_i^* for each generator/motor is calculated in the same way as in (1.23).

1.1.3 Linearized network swing model

Recall that a power network is represented by a connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with buses as nodes $\mathcal{V} = \{1,\ldots,n\}$ and transmission lines as edges $\mathcal{E} \subseteq \mathcal{V}\times\mathcal{V}$ [29, 99]. Here, we present another representation of modeling the power networks dynamics in the literature where the gnereators' transient reactances are ignored, and it is assumed that each bus hosts a generator. We can model the active power P_{ij} flowing from bus (node) i to bus j along the transmission line $(i,j) \in \mathcal{E}$ as

$$P_{ij} = \frac{E_i E_j}{\chi_{ij}}\sin(\delta_i - \delta_j), \tag{1.31}$$

where δ_i is the phase angle, E_i is the peak voltage magnitude, and $\chi_{ij} > 0$ is the line reactance. This model ignores the line resistances. The swing equation for a single generator

i results from Newton's second law and is given by

$$M_i\ddot{\delta}_i + D_i\dot{\delta}_i = P_i^{mech} - P_i^{elec}, \quad i \in \{1, \dots, n\}, \tag{1.32}$$

where $M_i > 0$ is the rotor moment of inertia, $D_i > 0$ is a damping constant, and P_i^{mech} and P_i^{elec} are the input mechanical power and output electrical power for the i^{th} generator, respectively. Plugging (1.31) into (1.32) leads to the swing equations of an electric power grid [27, 91, 112]

$$M_i\ddot{\delta}_i + D_i\dot{\delta}_i + \sum_{j\in\mathcal{V}_i} \frac{E_iE_j}{\chi_{ij}} \sin(\delta_i - \delta_j) = P_i^{mech} - P_i^{load} = P_i^{net}\ , \qquad \forall i \in \mathcal{V}, \tag{1.33}$$

where the set $\mathcal{V}_i \in \mathcal{V}$ refers to those buses connected to bus i in $\mathcal{G}$, P^{load} corresponds to the portion of the electric power consumed at bus i and P_i^{net} is the net power input at bus i.

Assuming small angle differences ($\delta_i - \delta_j \simeq 0$) and unity voltage magnitudes ($E_i = 1$), we can rewrite (1.31) as

$$P_{ij} \simeq b_{ij}(\delta_i - \delta_j), \tag{1.34}$$

where $b_{ij} = \frac{1}{\chi_{ij}}$ is the suseptance between the nodes $(i, j) \in \mathcal{E}$.

Define $\delta = [\delta_1, \delta_2, \dots, \delta_n]^T \in \mathbb{R}^n$. Then, the original dynamics in (1.33) can be linearized as

$$\Sigma := \begin{cases} M\ddot{\delta}(t) + D\dot{\delta}(t) + L\delta(t) = Bu(t), \\ y(t) = C\delta(t), \end{cases} \tag{1.35}$$

where $M > 0$ and $D > 0$ are the diagonal matrices of inertia and damping coefficients

defined as

$$M = \text{diag}(M_1, M_2, \ldots, M_n) \in \mathbb{R}^{n\times n} \quad \text{and} \quad D = \text{diag}(D_1, D_2, \ldots, D_n) \in \mathbb{R}^{n\times n}, \tag{1.36}$$

and $L \in \mathbb{R}^{n\times n}$ is the susceptance Laplacian matrix whose (i,j)th entry is given by

$$L(i,j) := \begin{cases} -b_{ij}, & \text{if } (i,j) \in \mathcal{E}, \\ \sum_{(i,j)\in\mathcal{E}} b_{ij}, & \text{if } j = i, \\ 0 & \text{otherwise} \end{cases}, \tag{1.37}$$

and has the following properties: Matrix L has strictly positive diagonal entries while the off-diagonal entries are negative or zero. It is positive semi-definite ($L = L^T \geq 0$) and has a simple zero eigenvalue with the corresponding eigenvector of ones $\mathbf{1}_n \in \mathbb{R}^n$ ($L\mathbf{1}_n = 0_{n\times 1}$). Moreover

$$u = [P_1^{net} \ \ldots \ P_n^{net}]^T \in \mathbb{R}^n \quad \text{and} \quad B = I_n \in \mathbb{R}^{n\times n}, \tag{1.38}$$

where $I_n \in \mathbb{R}^{n\times n}$ is the identity matrix, and $C \in \mathbb{R}^{p\times n}$ yields the output of the system.

By defining the new state variable

$$x = \begin{bmatrix} \delta \\ \dot{\delta} \end{bmatrix} \in \mathbb{R}^{2n},$$

one can equivalently represent the second-order dynamic (1.35) in its first-order form

$$\begin{aligned} \dot{x}(t) &= \mathcal{A}x(t) + \mathcal{B}u(t), \\ y(t) &= \mathcal{C}x(t) \end{aligned} \tag{1.39}$$

with

$$\mathcal{A} = \begin{bmatrix} 0_{n\times n} & I_n \\ -M^{-1}L & -M^{-1}D \end{bmatrix} \in \mathbb{R}^{(2n)\times(2n)}, \quad \mathcal{B} = \begin{bmatrix} 0_{n\times n} \\ M^{-1}B \end{bmatrix} \in \mathbb{R}^{(2n)\times n}, \quad \mathcal{C} = \begin{bmatrix} C & 0_{p\times n} \end{bmatrix} \in \mathbb{R}^{p\times(2n)}.$$

Since M and D are positive definite, and $L = L^T$ is positive semi-definite with the aforementioned properties, $\mathcal{A}$ has one eigenvalue at zero and $2n-1$ eigenvalues in the left-half plane [43, 45]. Thus (1.35) is a stable dynamical system, not asymptotically stable [45].

1.1.4 Linearized parametric model

In practice, the matrix L is not constant due to variations, for example, in peak voltage magnitudes E_i, reactances, and equilibria; see, e.g., [72]. Thus, to allow variations, in this section we will view E_i as a parameter that can vary and write it simply as p_i to obtain the parametric power network model

$$M_i\ddot{\delta}_i(t;p)+D_i\dot{\delta}_i(t;p) + \sum_{j\in\nu_i} \frac{p_i p_j}{\chi_{ij}} \sin(\delta_i(t;p) - \delta_j(t;p)) = P_i^{net} \ , \qquad \forall i \in \mathcal{V} \tag{1.40}$$

with the corresponding linear model

$$\begin{aligned} M\ddot{\delta}(t;p) + D\dot{\delta}(t;p) + L(p)\delta(t;p) &= Bu(t) \\ y(t;p) &= C\delta(t;p), \end{aligned} \tag{1.41}$$

where

$$p = [p_1 \ p_2 \ \dots \ p_n]^T \in \Omega \subseteq \mathbb{R}^n$$

is the parameter vector, the matrix $L(p)$ will now vary with p, and allows for variation in operating conditions. The parametric matrix $L(p)$ can be written as

$$L(p) = \mathcal{P}L\mathcal{P}, \tag{1.42}$$

where

$$\mathcal{P} = \text{diag}(p) = \text{diag}(p_1, \ldots, p_n) \in \mathbb{R}^{n\times n} \tag{1.43}$$

is diagonal and L is as defined in (1.37). Note that $p_i = 1$ for $i = 1, \ldots, n$ recovers the non-parametric problem. We will allow p_i's vary around this nominal value, i.e., $p_i \in (1-\alpha, 1+\alpha)$ where $0 < \alpha < 1$; thus $\mathcal{P}$ stays invertible for every $p \in \Omega$. Choosing, e.g., $\alpha = 0.15$, corresponds to allowing a 15% variation in peak voltage magnitudes.

1.2 Contributions

Power systems are complex and large-scale systems in operation. Simulation of these high dimensional power system models are computationally expensive and demand unmanageable levels of storage in dynamic simulation or trajectory sensitivity analysis. Hence, reducing the order of these models is of a great importance especially for real time applications, stability analysis, and control design [46]. Model order reduction (MOR) tackles this predicament by constructing a substantially lower dimensional representation of the original high dimensional model that provides a high-fidelity approximation of the input/output behaviour of the original model [3, 4, 5, 5, 12, 22, 37]. For instance, recall the nonlinear power network model

(1.2)-(1.3) of order n

$$\mathcal{M}\ddot{\delta}(t) + \mathcal{D}\dot{\delta}(t) + f(\delta) = \mathcal{B}u(t), \tag{1.2}$$

$$y(t) = \mathcal{C}\delta(t), \tag{1.3}$$

Using MOR we aim to replace (1.2)-(1.3) with a lower dimensional model with $r \ll n$

$$\mathcal{M}_r\ddot{\delta}_r(t) + \mathcal{D}_r\dot{\delta}_r(t) + f_r(\delta_r) = \mathcal{B}_r u(t),$$

$$y_r(t) = \mathcal{C}_r\delta_r(t),$$

where $\mathcal{M}_r, \mathcal{D}_r \in \mathbb{R}^{r\times r}$; $\mathcal{B}_r \in \mathbb{R}^r$; $f_r : \mathbb{R}^r \to \mathbb{R}^r$; and $\mathcal{C}_r \in \mathbb{R}^{p\times r}$. The goal is that the reduced output $y_r(t)$ approximates the original output $y(t)$ with high fidelity, i.e., $y(t) \approx y_r(t)$. In this thesis, we investigate MOR for both the nonlinear swing equations and its linearized parametric model.

In a case where the internal description of a system is not accessible and only some measurement data is available, data-driven modeling frameworks come into play [32, 53, 75, 78, 81, 102, 107, 113, 126]. In our setting, data corresponds to K input trajectory data as well as the state snapshot data, i.e., simulation data of the original nonlinear swing equations (1.2) as

$$Z = \begin{bmatrix} \delta(t_0) & \delta(t_1) & \dots & \delta(t_{K-1}) \end{bmatrix} \in \mathbb{R}^{n\times K}$$

$$\dot{Z} = \begin{bmatrix} \dot{\delta}(t_0) & \dot{\delta}(t_1) & \dots & \dot{\delta}(t_{K-1}) \end{bmatrix} \in \mathbb{R}^{n\times K}$$

$$U = \begin{bmatrix} u(t_0) & u(t_1) & \dots & u(t_{K-1}) \end{bmatrix} \in \mathbb{R}^{1\times K}.$$

MOR and data-driven modeling have a long history in power systems [31, 39, 47, 124, 124,

130]. For a comparative study of MOR techniques in power systems, see, e.g. [2, 28, 46]. In this thesis, we develop model reduction frameworks and data driven modeling tailored for power grid swing equations.

Main contributions and organization of the thesis

Our main contributions are:

1. We develop a structure-preserving parametric model reduction approach for parametrized linearized swing equations where parametrization corresponds to variations in operating conditions. (Chapter 2)

2. We develop a structure-preserving system-theoretic model reduction framework for nonlinear power grid networks. (Chapter 3 and Chapter 4)

3. We develop a non-intrusive data-driven modeling framework for power network dynamics using the Lift and Learn approach of [102]. (Chapter 5)

4. We develop a structure-preserving data-driven system identification framework for nonlinear power grid networks. (Chapter 6)

The rest of this dissertation is organized as follows:

- Chapter 2: This chapter presents the first major contribution of this dissertation. We develop a structure-preserving parametric model reduction approach for linearized swing equations using a global basis framework and interpolatory $\mathcal{H}_2$ model reduction. We establish the subspace conditions for the reduction basis so that the error system is an $\mathcal{H}_2$ and $\mathcal{H}_\infty$ function over the entire parameter space.

- Chapter 3: We present a quadratic representation for the power network dynamics via a lifting transformation. Then, we investigate the effect of this lifting transformation

on the equilibrium points.

- Chapter 4: We develop the second major contribution of this thesis, namely $\mathcal{H}_2$-based structure-preserving MOR for power network dynamics (StrH2). We employ the $\mathcal{H}_2$-based model reduction approach, Quadratic Iterative Rational Krylov Algorithm (Q-IRKA) [23], as an intermediate model reduction step. Exploiting the structure of the underlying power network model, we show that the model reduction bases resulting from Q-IRKA has a special subspace structure, which allows us to effectively construct the final model reduction basis. This final basis is then applied in the original nonlinear structure to yield a reduced model that preserves the physically meaningful (second-order) structure of the original model.

- Chapter 5: This chapter presents the third major contribution of this dissertation. We start by reviewing the Lift and Learn method [102] for quadratic models. Then, we employ this method to learn a reduced quadratic model for the nonlinear swing equations using time domain data.

- Chapter 6: This chapter present the last major contribution of this dissertation. We develop a data-driven system identification method for the second-order nonlinear model (1.1). Our framework is inspired by the second-order SINDY [38], where we chose the nonlinear functions in the SINDY library of terms based on the structure of the swing equations. Unlike the regular SINDY method [32], we enforced sparsity in the SINDY output of coefficients by employing a mask matrix.

- Chapter 7: We present a summary of our work and future research outlooks.

Chapter 2

Structure-preserving Model Reduction of Parametric Power Networks

Power networks are naturally modeled as second-order dynamical systems [45, 79, 91, 112]. There is a plethora of model reduction approaches for second-order dynamical systems, see, e.g., [9], [30], [13], [118], [85], [104], [37], for model reduction of general second-order systems, and see, e.g., [71], [45], [44], [87], [129] with a focus on network dynamics.

In this chapter, we focus on parametrically varying power networks where the parameter variations correspond to different operation conditions. This leads to the parametric model reduction (PMOR) framework [21, 22, 64, 103]. The goal is to find a parametric reduced model that can approximate the original model with acceptable accuracy over a wide range of parameters. PMOR eliminates the need for performing a separate reduction for each parameter value (operating condition) and therefore plays an important rule in control, design, optimization and uncertainty quantification. In Section 2.1, we employ a global basis approach to form our parametric reduced-order structure-preserving (second-order) power network model. The model reduction basis is constructed by concatenation of local bases for selected parameter samples. We obtain the local bases using second-order interpolatory $\mathcal{H}_2$-optimal methods [121, 127]. In Section 2.2, we establish a subspace condition on the

model reduction basis to guarantee bounded $\mathcal{H}_2$ and $\mathcal{H}_\infty$ error norms for whole parameter domain. Then, we explain algorithmic implications. Section 2.3 illustrates the feasibility of our approach via numerical examples.

Much of the material in this chapter has already appeared in the published manuscript [108].

2.1 Structure-preserving parametric reduced models for linearized swing equations

Recall the linear parametric model from Section 1.1.4

$$\begin{aligned} M\ddot{\delta}(t;p) + D\dot{\delta}(t;p) + L(p)\delta(t;p) &= Bu(t) \\ y(t;p) &= C\delta(t;p), \end{aligned} \tag{1.41}$$

where $p = [p_1\ p_2\ \ldots\ p_n]^T \in \Omega \subseteq \mathbb{R}^n$ denotes the parameter vector. We seek to develop a reduction framework such that not only it preserves the structure of (1.41), but also the parametric reduced model serves with acceptable accuracy as a surrogate model over diverse operating conditions. Since it is crucial that the reduced model preserves the physically-meaningful second-order structure as in (1.35), instead of transferring the second-order dynamics to the first-order form, as in (1.39), and applying model reduction there, we will directly reduce the second-order dynamics (1.41). In other words, our goal is to find a reduced parametric system

$$\begin{aligned} M_r\ddot{\delta}_r(t;p) + D_r\dot{\delta}_r(t;p) + L_r(p)\delta_r(t;p) &= B_r u(t) \\ y_r(t;p) &= C_r\delta_r(t;p), \end{aligned} \tag{2.1}$$

where $\delta_r(t;p) \in \mathbb{R}^r$, M_r, $L_r(p)$, $D_r \in \mathbb{R}^{r\times r}$, $B \in \mathbb{R}^{r\times n}$, and $C \in \mathbb{R}^{p\times r}$ with $r \ll n$ such that the $y_r(t;p) \approx y(t;p)$ for a wide range of inputs $u(t)$ over the parameter range of interest.

Since M and D are symmetric positive definite, and $L(p)$ is symmetric positive semi-definite, one should preserve these structures in the reduced model. We achieve this using Galerkin projection where the original state δ is approximated as $\delta \approx V\delta_r$: construct a model reduction basis $V \in \mathbb{R}^{n\times r}$ and the reduced-order matrices in (2.1) using

$$\begin{aligned} &M_r = V^T M V,\ D_r = V^T D V,\ L_r(p) = V^T L(p) V, \\ &B_r = V^T B, \text{ and } C_r = CV. \end{aligned} \tag{2.2}$$

Accuracy of the structure-preserving reduced model (2.1) with the form (2.2) clearly depends on the choice of V. We describe this choice next.

2.1.1 Interpolatory model reduction bases

There are numerous ways to choose the model reduction basis V for reducing parametric dynamical systems; see, for example, [5, 21, 22, 64, 103] and the references therein. For the parametric structured second-order dynamical system (1.41), we will employ the structure-preserving parametric interpolatory model reduction framework from [6], which extended the interpolatory model reduction framework for parametric systems [11] to the structured setting. For recent extensions of structured interpolatory model reduction to special classes of nonlinear systems, see [25, 26]. For a comprehensive study of interpolatory model reduction, we refer the reader to [5].

To begin with, we need to obtain the frequency domain representation of the parametric original and the reduced order models by Laplace transform. Let $Y(s,p)$, $Y_r(s,p)$ and $U(s)$

be the Laplace transforms of the $y(t;p)$, $y_r(t;p)$ and $u(t)$, respectively. By taking the Laplace transform of (1.41) and (2.1), respectively, we obtain

$$\begin{aligned} Y(s,p) &= (C(s^2M + sD + L(p))^{-1}B)U(s) = H(s,p)U(s) \\ \text{and} \quad Y_r(s,p) &= (C_r(s^2M_r + sD_r + L_r(p))^{-1}B_r)U(s) = H_r(s,p)U(s), \end{aligned}$$

where the transfer functions of the full-order parametric model (1.41) and reduced one (2.1) are, respectively,

$$H(s,p) = C(s^2M + sD + L(p))^{-1}B, \tag{2.3}$$

$$\text{and} \quad H_r(s,p) = C_r(s^2M_r + sD_r + L_r(p))^{-1}B_r. \tag{2.4}$$

Note that both $H(s,p)$ and $H_r(s,p)$ are $p \times n$ matrix-valued rational functions in s. The goal, in parametric interpolatory model reduction, is to choose V such that $H_r(s,p)$ interpolates $H(s,p)$ at selected points in the frequency s and parameter p. For a parametric single input/single output (SISO) case with m parameter samples $\{p^{(1)}, \ldots, p^{(m)}\}$ ($p^{(i)}$ is a parameter point of interest) and r_i frequency interpolation points $\{\sigma_1^{(i)}, \ldots, \sigma_{r_i}^{(i)}\} \in \mathbb{C}$, this implies

$$H(\sigma_j^{(i)}, p^{(i)}) = H_r(\sigma_j^{(i)}, p^{(i)}) \; ; \quad j = 1, 2, \ldots, r_i \quad \text{and} \quad i = 1, 2, \ldots, m. \tag{2.5}$$

However, for a multi-input/multi-output (MIMO) system like (1.41), enforcing the interpolation condition (2.5) for every interpolation point and parameter sample may require a large reduced order r. To overcome this issue, one can enforce interpolation only along the

selected directions. Then, the goal is to construct V such that

$$H(\sigma_j^{(i)}, p^{(i)})b_j^{(i)} = H_r(\sigma_j^{(i)}, p^{(i)})b_j^{(i)} \; ; \quad j = 1, 2, \ldots, r_i \quad \text{and} \quad i = 1, 2, \ldots, m, \tag{2.6}$$

where $\{b_1^{(i)}, \ldots, b_{r_i}^{(i)}\} \in \mathbb{C}^n$ are tangent directions for the parameter sample $p^{(i)}$. Define

$$\mathcal{K}(s, p) = s^2 M + sD + L(p)$$

so that

$$H(s, p) = C\mathcal{K}^{-1}(s, p)B.$$

For $i = 1, 2, \ldots, m$, construct the *local* interpolation basis $V^{(i)} \in \mathbb{C}^{n \times r_i}$ corresponding to the parameter sample $p^{(i)}$ using

$$V^{(i)} = \left[\mathcal{K}(\sigma_1^{(i)}, p^{(i)})^{-1}Bb_1^{(i)}, \ldots, \mathcal{K}(\sigma_{r_i}^{(i)}, p^{(i)})^{-1}Bb_{r_i}^{(i)}\right] \tag{2.7}$$

and concatenate the local bases to construct the *global* basis:

$$V = \text{orth}\left(\begin{bmatrix} V^{(1)} & V^{(2)} & \ldots V^{(m)} \end{bmatrix}\right) \in \mathbb{R}^{n \times r}, \tag{2.8}$$

where "orth" refers to an orthogonal basis so that $V^T V = I_r$. Realness of V is guaranteed by choosing the interpolation points and tangent directions in conjugate pairs. Then, the reduced model (2.1) obtained as in (2.2) using V from (2.8) satisfies the interpolation conditions (2.6); see [5, 6].

Quality of the reduced model will depend on the choice of interpolation points and tangent directions. We choose them, and thus the local bases $V^{(i)}$, using interpolatory optimal $\mathcal{H}_2$ model reduction. In other words, for every $p^{(i)}$, we construct the local basis $V^{(i)}$ to

minimize/reduce the $\mathcal{H}_2$-distance

$$\|H(\cdot,p^{(i)})-H_r(\cdot,p^{(i)})\|_{\mathcal{H}_2}=\Big(\frac{1}{2\pi}\int_{-\infty}^{\infty}\|H(\mathrm{j}\omega,p^{(i)})-H_r(\mathrm{j}\omega,p^{(i)})\|_F^2 d\omega\Big)^{\frac{1}{2}}, \tag{2.9}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Note that the $\mathcal{H}_2$ error norm (2.9) is related to the $\mathcal{L}_\infty$ norm of the output error

$$\|y(\cdot,p^{(i)})-y_r(\cdot,p^{(i)})\|_{\mathcal{L}_\infty}=\sup_{t>0}\|y(\cdot,p^{(i)})-y_r(\cdot,p^{(i)})\|_\infty$$

in the time domain as

$$\|y(\cdot,p^{(i)})-y_r(\cdot,p^{(i)})\|_{\mathcal{L}_\infty}\leq\|H(\cdot,p^{(i)})-H_r(\cdot,p^{(i)})\|_{\mathcal{H}_2}\ \|u\|_{\mathcal{L}_2},$$

which means that by using $\mathcal{H}_2$-based model reductions, we can minimize the output error in the $\mathcal{L}_\infty$ norm. Optimal $\mathcal{H}_2$ model reduction is a heavily studied topic, see for example, [1, 33, 36, 92] and the references therein. In the case of non-parametric unstructured linear dynamical systems, i.e., $H_r(s)=C_r(sI_r-A_r)^{-1}B_r$, the optimal reduced model in the $\mathcal{H}_2$-norm is a bitangential Hermite interpolant to $H(s)$ at the mirror images of the reduced poles [6, 61]. This means that given the first order stable model

$$\begin{aligned}\dot{x}(t)&=\mathcal{A}x(t)+\mathcal{B}u(t),\\ y(t)&=\mathcal{C}x(t)\end{aligned}$$

with l inputs and p outputs, and the corresponding transfer function $H(s)=\mathcal{C}(sI_n-\mathcal{A})^{-1}\mathcal{B}$, let $H_r(s)$ be the best rth order rational approximation of H with respect to the $\mathcal{H}_2$ norm.

Then, for $k = 1, \ldots, r$

$$\begin{aligned} H(-\lambda_k)r_k &= H_r(-\lambda_k)r_k, \\ \ell_k^T H(-\lambda_k) &= \ell_k^T H_r(-\lambda_k) \quad \text{and} \\ \ell_k^T H'(-\lambda_k)r_k &= \ell_k^T H_r'(-\lambda_k)r_k, \end{aligned} \tag{2.10}$$

where $H'(-\lambda_k)$ indicates the first derivative of H at $s = -\lambda_k$, $\lambda_k \in \mathbb{C}$ for $k = 1, \ldots, r$ are the simple poles of H_r, and $\ell_k \in \mathbb{C}^p$ and $r_k \in \mathbb{C}^l$ are residue directions ($\ell_k r_k^T$ is a rank-1 matrix) resulting from the pole residue expansion of H_r as

$$H_r(s) = \mathcal{C}_r(sI_r - \mathcal{A}_r)^{-1}\mathcal{B}_r = \sum_{k=1}^{r} \frac{\ell_k r_k^T}{s - \lambda_k} \ .$$

Due to the lack of the prior knowledge of the reduced system poles, the model reduction algorithm must be performed in an iterative way. The Iterative Rational Krylov Algorithm (IRKA) [61] and it variants, e.g., [16, 67, 128], have been successfully applied in this setting to construct optimal interpolation points and directions. Since we require the reduced-model to have the second-order form, we employ the structured version of IRKA, namely the Second Order IRKA (SOR-IRKA) [121, 127] to construct interpolation points $\{\sigma_1^{(i)}, \ldots, \sigma_{r_i}^{(i)}\}$ and $\{b_1^{(i)}, \ldots, b_{r_i}^{(i)}\} \in \mathbb{C}^n$ and thus the local bases $V^{(i)}$. A brief sketch of SOR-IRKA is given in Algorithm 1. SOR-IRKA produces a reduced-model that satisfies only a subset of optimal interpolation conditions at the cost of preserving structure. This is due to the fact that in SOR-IRKA instead of directly assigning the r_i interpolation points as the mirror images of the r_i poles out of the $2r_i$ poles of the reduced model $H_{r_i}(s, p^{(i)}) = \mathcal{C}_{2r_i}(sI_{2r_i} - \mathcal{A}_{2r_i}(p^{(i)}))^{-1}\mathcal{B}_{2r_i}$ in Step (2) of Algorithm 1, we use an intermediate model reduction step to reduce the dimension of the reduced model $H_{r_i}(s, p^{(i)})$ from $2r_i$ to r_i and assign the mirror images of the resulting r_i poles as the r_i interpolation points. In other words, the r_i interpolation

points $\{\sigma_1^{(i)}, \ldots, \sigma_{r_i}^{(i)}\}$ are not the mirror images of the reduced model $H_{r_i}(s, p^{(i)})$, yet result from the key characteristic of the $2r_i$ poles of the system. Since the underlying system has a pole at zero in our case, we will modify SOR-IRKA further. This will be explained in detail in Section 2.2.2. For other work on $\mathcal{H}_2$-based model reduction of second-order systems, see, e.g., [15, 87, 129].

Remark 2.1. The intermediate model reduction technique in Step (2) of Algorithm 1 can also be done via balanced truncation [60].

Remark 2.2. As opposed to developing locally optimal $\mathcal{H}_2$ model reduction bases $V^{(i)}$ and concatenating them to construct the global basis V, following [11] one could introduce a composite error measure ($\mathcal{L}_2$ error in the parameter space and $\mathcal{H}_2$ error in the frequency domain). Then, one can try to construct V directly to minimize this composite measure. We refer the reader to [11] and more recent works [58, 68, 69] in this direction for the unstructured setting.

2.2 Matching the parametric residue corresponding to the pole at zero

Since $L(p) = \mathcal{P}L\mathcal{P}$ and $L\mathbf{1}_n = 0_{n\times 1}$ where $\mathbf{1}_n \in \mathbb{R}^{n\times 1}$ is the vector of ones, we obtain

$$L(p)\mathcal{P}^{-1}\mathbf{1}_n = \mathcal{P}L\mathbf{1}_n = 0_{n\times 1}.$$

Therefore, for every $p \in \Omega$, $L(p)$ has a simple zero eigenvalue with the eigenvector $v = \mathcal{P}^{-1}\mathbf{1}_n$, and consequently $H(s, p)$ has a simple pole at zero for every p. This means that $H(s, p)$ is not an $\mathcal{H}_2$-function. However, we can still perform an $\mathcal{H}_2$-based model reduction on $H(s, p)$

Algorithm 1 SOR-IRKA [127]

Input: Full order model (1.41) at the parameter sample $p^{(i)}$ (M, D, $L(p^{(i)})$, B, C)

Output: Local base $V^{(i)}$

- Initialize interpolation points $\{\sigma_1^{(i)}, \ldots, \sigma_{r_i}^{(i)}\}$ and tangent directions $\{b_1^{(i)}, \ldots, b_{r_i}^{(i)}\}$.
- Form the orthogonal base $V^{(i)} = \left[\mathcal{K}(\sigma_1^{(i)}, p^{(i)})^{-1} B b_1^{(i)}, \ldots, \mathcal{K}(\sigma_{r_i}^{(i)}, p^{(i)})^{-1} B b_{r_i}^{(i)}\right] \in \mathbb{C}^{n \times r_i}$.
- While (not converged):

  1. $M_{r_i} = {V^{(i)}}^T M V^{(i)}, \quad D_{r_i} = {V^{(i)}}^T D V^{(i)}, \quad L_{r_i}(p^{(i)}) = {V^{(i)}}^T L(p^{(i)}) V^{(i)},$
  $B_{r_i} = {V^{(i)}}^T B, \quad C_{r_i} = C V^{(i)}$

  2. Convert the second-order reduced model of dimension r_i found in Step (1) to its associated first order model of dimension $2r_i$ as done in (1.39).

$$\begin{aligned} \dot{x}(t) &= \mathcal{A}_{2r_i} x(t) + \mathcal{B}_{2r_i} u(t), \\ y(t) &= \mathcal{C}_{2r_i} x(t) \end{aligned}$$

  3. *Intermediate step:* Reduce the first order model of dimension $2r_i$ in Step (2) by (first-order) IRKA [61] to a reduced order model of dimension r_i, namely $H_{r_i}(s, p^{(i)}) = \mathcal{C}_{r_i}(sI - \mathcal{A}_{r_i}(p^{(i)}))^{-1} \mathcal{B}_{r_i}$.

  4. Compute the eigenvalue decomposition of $\mathcal{A}_{r_i}(p^{(i)}) = X \Lambda_{r_i} Y^T$, where Λ_{r_i} is the matrix of eigenvalues, and X and Y ($Y^T X = I_r$) are the right and left eigenvectors of $\mathcal{A}_{r_i}(p^{(i)})$, respectively.

  5. Update the interpolation points and tangent directions

$$\sigma_i^{(i)} \leftarrow -\lambda_i(\mathcal{A}_{r_i}(p^{(i)})) \quad \text{and} \quad \left(b_i^{(i)}\right)^T \leftarrow e_i^T Y \mathcal{B}_{r_i} \quad \text{for } i = 1, ..., r_i,$$

  where $\lambda_i(.)$ denotes the eigenvalue of the quantity of interest, and e_i is the unit vector.

  6. Update the local basis

$$V^{(i)} = \left[\mathcal{K}(\sigma_1^{(i)}, p^{(i)})^{-1} B b_1^{(i)}, \ldots, \mathcal{K}(\sigma_{r_i}^{(i)}, p^{(i)})^{-1} B b_{r_i}^{(i)}\right]. \tag{2.7}$$

as long as we guarantee that the error system, i.e., $H(s,p) - H_r(s,p)$, stays an $\mathcal{H}_2$-function for every p. This issue has been studied in the *non-parametric* case. [45] achieves a bounded $\mathcal{H}_2$ error norm in model reduction of second order networks where the Galerkin projection is obtained via clustering techniques. In a more recent work, [129] splits a non-parametric second order network with proportional damping into an asymptotically stable system and an average subsystem containing the zero eigenvalue. Then, the asymptotically stable system is reduced via interpolatory techniques and then re-combined with the average system leads to a reduced model with bounded (and small) $\mathcal{H}_2$ error. We also refer the reader to, e.g., [74, 87, 88, 89] for the first-order dynamics case.

In reducing the *parametric* second-order model (1.41), we need to enforce that $H_r(s,p)$ retains the zero eigenvalue and its *parametric* residue *for every* $p \in \Omega$ so that the error stays bounded over the whole domain. Next, we establish the subspace conditions on the model reduction basis V to achieve this goal.

2.2.1 Subspace conditions for matching the parametric residue

For a given a parameter, the next result establishes the conditions on V to match the residue at zero.

Theorem 2.3. *Given the parametric full-order model (1.41), let the parametric reduced model (2.1) be obtained as in (2.2). Let $\hat{p} \in \Omega$ be a parameter of interest. Define $\hat{\mathcal{P}} = diag(\hat{p})$ and $\hat{v} = \hat{\mathcal{P}}^{-1}\mathbf{1}_n$. Then for $\hat{p} \in \Omega$, the reduced model $H_r(s,\hat{p})$ retains the simple pole of $H(s,\hat{p})$ at $s = 0$ and its corresponding parameter-dependent residue if $\hat{v} \in span(V)$.*

Proof. First, we show that $L_r(\hat{p})$ has a simple zero eigenvalue.

Using $\hat{v} \in \text{span}(V)$, write V as

$$V = \begin{bmatrix} V_1 & \hat{v} \end{bmatrix}, \tag{2.11}$$

where $V_1 \in \mathbb{R}^{n\times(r-1)}$ and $\hat{v} \notin \text{span}(V_1)$. Then, using (2.11) and the fact $L(\hat{p})\hat{v} = 0$, we obtain

$$\begin{aligned} L_r(\hat{p}) = V^T L(\hat{p}) V = \begin{bmatrix} V_1^T \\ \hat{v}^T \end{bmatrix} L(\hat{p}) \begin{bmatrix} V_1 & \hat{v} \end{bmatrix} &= \begin{bmatrix} {V_1}^T L(\hat{p}) V_1 & {V_1}^T L(\hat{p})\hat{v} \\ \hat{v}^T L(\hat{p}) V_1 & \hat{v}^T L(\hat{p})\hat{v} \end{bmatrix} \\ &= \begin{bmatrix} V_1^T L(\hat{p}) V_1 & 0_{(r-1)\times 1} \\ 0_{1\times(r-1)} & 0 \end{bmatrix}. \end{aligned} \tag{2.12}$$

Since $\hat{v} \notin \text{span}(V_1)$, $L_r(\hat{p})$ has only one simple zero eigenvalue. Moreover, since M and D are positive definite and model reduction is performed via a Galerkin projection as in (2.2), all the other poles of $H_r(s, \hat{p})$ have negative real parts except for this simple pole at zero.

Now we need to show that the parametrically varying residues of $H(s, \hat{p})$ and $H_r(s, \hat{p})$ corresponding to the pole at zero match. To find the residue of $H(s, \hat{p})$, we follow an analysis inspired by [45]. Transform the second-order dynamic (1.41) to its equivalent first-order form

$$\begin{aligned} \dot{x}(t; p) &= \mathcal{A}(p)x(t; p) + \mathcal{B}u(t), \\ y(t; p) &= \mathcal{C}x(t; p), \end{aligned}$$

where

$$\mathcal{A}(p) = \begin{bmatrix} 0_{n\times n} & I_n \\ -M^{-1}L(p) & -M^{-1}D \end{bmatrix}, \quad \mathcal{B} = \begin{bmatrix} 0_{n\times n} \\ M^{-1}B \end{bmatrix}, \quad \text{and} \quad \mathcal{C} = \begin{bmatrix} C & 0_{p\times n} \end{bmatrix}. \tag{2.13}$$

Let $\mathcal{A}(p)$ have the Jordan decomposition

$$\mathcal{A}(p) = Q\Lambda Q^{-1} = \begin{bmatrix} q_1 & Q_2 \end{bmatrix} \begin{bmatrix} 0 & \\ & \bar{\Lambda} \end{bmatrix} \begin{bmatrix} \tilde{q}_1^T \\ \tilde{Q}_2^T \end{bmatrix}, \tag{2.14}$$

where the Jordan block $\bar{\Lambda} \in \mathbb{C}^{(2n-1)\times(2n-1)}$ contains the eigenvalues with negative real parts, and $q_1 \in \mathbb{R}^{2n}$ and $\tilde{q}_1 \in \mathbb{R}^{2n}$ are, respectively, the right and left eigenvectors corresponding to the zero eigenvalue such that

$$\mathcal{A}^T(p)\tilde{q}_1 = 0_{2n\times 1}, \tag{2.15}$$

$$\mathcal{A}(p)q_1 = 0_{2n\times 1}, \tag{2.16}$$

$$\tilde{q}_1^T q_1 = 1. \tag{2.17}$$

We note that this decomposition is parameter dependent but to simplify the notation, we write, e.g., Q instead of $Q(p)$. Let

$$q_1 = \begin{bmatrix} q_{11} \\ q_{12} \end{bmatrix} \quad \text{and} \quad \tilde{q}_1 = \begin{bmatrix} \tilde{q}_{11} \\ \tilde{q}_{12} \end{bmatrix}, \tag{2.18}$$

where $\tilde{q}_{1j}$, $q_{1j} \in \mathbb{R}^n$ for $j = 1, 2$. Calculating (2.16) at $p = \hat{p}$ using (2.18) and (2.13) leads to

$$\mathcal{A}(\hat{p})q_1 = \begin{bmatrix} 0_{n\times n} & I_n \\ -M^{-1}L(\hat{p}) & -M^{-1}D \end{bmatrix} \begin{bmatrix} q_{11} \\ q_{12} \end{bmatrix} = \begin{bmatrix} 0_{n\times 1} \\ 0_{n\times 1} \end{bmatrix}, \tag{2.19}$$

which by the fact $L(\hat{p})\hat{v} = 0$ yields

$$q_1 = \begin{bmatrix} q_{11} \\ 0_{n\times 1} \end{bmatrix} = \begin{bmatrix} \hat{v} \\ 0_{n\times 1} \end{bmatrix}. \tag{2.20}$$

Similarly, form (2.15) using (2.18) and (2.13):

$$\mathcal{A}^T(\hat{p})\tilde{q}_1 = \begin{bmatrix} 0_{n\times n} & -L(\hat{p})M^{-1} \\ I_n & -DM^{-1} \end{bmatrix} \begin{bmatrix} \tilde{q}_{11} \\ \tilde{q}_{12} \end{bmatrix} = \begin{bmatrix} 0_{n\times 1} \\ 0_{n\times 1} \end{bmatrix}, \tag{2.21}$$

which leads to

$$-L(\hat{p})M^{-1}\tilde{q}_{12} = 0 \quad \text{and} \tag{2.22}$$

$$\tilde{q}_{11} - DM^{-1}\tilde{q}_{12} = 0. \tag{2.23}$$

Then, by using $L(\hat{p})\hat{v} = 0$ and (2.17), we obtain

$$\tilde{q}_1 = \begin{bmatrix} \tilde{q}_{11} \\ \tilde{q}_{12} \end{bmatrix} = \frac{1}{\alpha_D} \begin{bmatrix} D\hat{v} \\ M\hat{v} \end{bmatrix}, \tag{2.24}$$

where $\alpha_D = \hat{v}^T D\hat{v}$. Using (2.14), we write

$$\begin{aligned} H(s,\hat{p}) = \mathcal{C}(sI_{2n} - \mathcal{A}(\hat{p}))^{-1}\mathcal{B} &= \mathcal{C}Q(sI_{2n} - \Lambda)^{-1}Q^{-1}\mathcal{B} \\ &= \frac{(\mathcal{C}q_1)(\tilde{q}_1^T\mathcal{B})}{s} + \mathcal{C}Q_2(sI_{2n-1} - \bar{\Lambda})^{-1}\tilde{Q}_2\mathcal{B} \\ &= \frac{\phi_0}{s} + \mathcal{C}Q_2(sI_{2n-1} - \bar{\Lambda})^{-1}\tilde{Q}_2\mathcal{B}. \end{aligned} \tag{2.25}$$

where $\phi_0 = (\mathcal{C}q_1)(\tilde{q}_1^T\mathcal{B})$ is the residue of $H(s,\hat{p})$ for the pole at zero. Then, substituting q_1

and $\tilde{q}_1$ from (2.20) and (2.24), and $\mathcal{C}$ and $\mathcal{B}$ from (2.13) into $\phi_0 = (\mathcal{C}q_1)(\tilde{q}_1^T\mathcal{B})$ yields

$$\begin{aligned}\phi_0 &= \alpha_D^{-1}\mathcal{C}\begin{bmatrix}\hat{v}\\ 0_{n\times 1}\end{bmatrix}\begin{bmatrix}\hat{v}^TD & \hat{v}^TM\end{bmatrix}\mathcal{B} = \alpha_D^{-1}\mathcal{C}\begin{bmatrix}\hat{v}\hat{v}^TD & \hat{v}\hat{v}^TM\\ 0_{n\times n} & 0_{n\times n}\end{bmatrix}\mathcal{B}\\ &= \alpha_D^{-1}\begin{bmatrix}C & 0_{p\times n}\end{bmatrix}\begin{bmatrix}\hat{v}\hat{v}^TD & \hat{v}\hat{v}^TM\\ 0_{n\times n} & 0_{n\times n}\end{bmatrix}\begin{bmatrix}0_{n\times n}\\ M^{-1}B\end{bmatrix}\\ &= \alpha_D^{-1}C\hat{v}\hat{v}^TB.\end{aligned} \tag{2.26}$$

Following similar steps, the residue of the reduced system $H_r(s,\hat{p})$ corresponding to the pole at $s=0$ is obtained as

$$\phi_{0_r} = \alpha_{D_r}^{-1}CVV^T\hat{v}\hat{v}^TVV^TB, \tag{2.27}$$

where $\alpha_{D_r} = \hat{v}^TVD_rV^T\hat{v}$. Since VV^T is an orthogonal projector (i.e., $(VV^T)^2 = VV^T$ and $(VV^T)^T = VV^T$) , when $\hat{v} \in span(V)$, we have

$$VV^T\hat{v} = \hat{v}\ . \tag{2.28}$$

Then, plug (2.28) into (2.27) to obtain

$$\begin{aligned}\phi_{0_r} = \alpha_{D_r}^{-1}C\hat{v}\hat{v}^TB &= (\hat{v}^TVD_rV^T\hat{v})^{-1}C\hat{v}\hat{v}^TB\\ &= (\hat{v}^TVV^TDVV^T\hat{v})^{-1}C\hat{v}\hat{v}^TB\\ &= (\hat{v}^TD\hat{v})^{-1}C\hat{v}\hat{v}^TB = \alpha_D^{-1}C\hat{v}\hat{v}^TB = \phi_0.\end{aligned} \tag{2.29}$$

□

Theorem 2.3 establishes that if $\hat{v} = \hat{\mathcal{P}}^{-1}\mathbf{1}_n \in \text{span}(V)$, for that parameter value $\hat{p}$, the

residues of $H(s,\hat{p})$ and $H_r(s,\hat{p})$ match for the pole at $s=0$. This means that

$$\begin{aligned} H(s,\hat{p}) - H_r(s,\hat{p}) &= \mathcal{C}(sI_{2n} - \mathcal{A}(\hat{p}))^{-1}\mathcal{B} - \mathcal{C}_r(sI_r - \mathcal{A}_r(\hat{p}))^{-1}\mathcal{B}_r \\ &= \frac{\phi_0}{s} + H_a(s,\hat{p}) - \left(\frac{\phi_{r_0}}{s} + H_{a_r}(s,\hat{p})\right) \\ &= H_a(s,\hat{p}) - H_{a_r}(s,\hat{p}), \end{aligned} \tag{2.30}$$

where the transfer functions

$$H_a(s,\hat{p}) = \mathcal{C}Q_2(sI_{2n-1} - \bar{\Lambda})^{-1}\tilde{Q}_2\mathcal{B} \tag{2.31}$$

$$\text{and} \quad H_{a_r}(s,\hat{p}) = \mathcal{C}_r Q_{2_r}(sI_{r-1} - \bar{\Lambda}_r)^{-1}\tilde{Q}_{2_r}\mathcal{B}_r \tag{2.32}$$

are asymptotically stable. Therefore, the error system (2.30) is asymptotically stable at $\hat{p}$. We write this result as a corollary.

Corollary 2.4. *Assume the set-up of Theorem 2.3. Then, the error system $H(s,\hat{p}) - H_r(s,\hat{p})$ is asymptotically stable, and has bounded $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norms.*

2.2.2 Algorithmic Implications

Theorem 2.3 and Corollary 2.4 hint at how to construct V so that the error system is asymptotically stable at a parameter value of interest. As stated in Section 2.1.1, for the parameter samples $p^{(i)}$ for $i = 1,\ldots,m$, we will construct the local bases $V^{(i)}$ in (2.7) via SOR-IRKA to have local $\mathcal{H}_2$ optimality. However, we will modify SOR-IRKA by taking into consideration that $H(s,p)$ has a pole at zero for every p, i.e., $H(s,p)$ is not an $\mathcal{H}_2$ function. SOR-IRKA is an iterative algorithm that corrects the interpolation points in every step. Due to the pole at zero, SOR-IRKA might drive one of the interpolation points to zero as it should

so that the pole and residue at zero are matched. This will require computing the vector $\mathcal{K}(0, p^{(i)})^{-1} B b_0^{(i)}$ in (2.7) of Algorithm 1. However, due to the pole at zero, $\mathcal{K}(0, p^{(i)}) = L(p^{(i)})$ is not invertible. Therefore, inspired by Theorem 2.3, in SOR-IRKA in Algorithm 1, we will replace vector $\mathcal{K}(0, p^{(i)})^{-1} B b_0^{(i)}$ with the zero eigenvector of $L(p^{(i)})$, i.e., $v^{(i)}$, and rewrite (2.7) as

$$V^{(i)} = \left[\mathcal{K}(\sigma_1^{(i)}, p^{(i)})^{-1} B b_1^{(i)}, \ldots, v^{(i)}, \ldots, \mathcal{K}(\sigma_{r_i}^{(i)}, p^{(i)})^{-1} B b_{r_i}^{(i)} \right]. \tag{2.33}$$

Thus the span of $V^{(i)}$ will contain this eigenvector. Hence, once the global basis V is constructed as in (2.8), Theorem 2.3 will guarantee that the error system $H(s,p) - H_r(s,p)$ in (2.30) is asymptotically stable for the *sampled parameter values* $p^{(i)}$ for $i = 1, \ldots, m$.

To use $H_r(s,p)$ for an unsampled parameter value $\hat{p}$ and to still guarantee bounded error, we compute $\hat{v} = \hat{\mathcal{P}}^{-1} \mathbf{1}_n$, construct the new basis $\widehat{V} = \begin{bmatrix} V & \hat{v} \end{bmatrix}$, and obtain $H_r(s, \hat{p})$ as in (2.2), now using $\widehat{V}$. Theorem 2.3 will then guarantee a bounded error at $\hat{p}$ as well.

The reduction step (2.2) does not need to be applied from scratch for every new $\hat{p}$. For the new basis $\widehat{V}$, consider

$$\widehat{M}_r : \ \widehat{M}_r = \widehat{V}^T M \widehat{V} = \begin{bmatrix} V^T \\ \hat{v}^T \end{bmatrix} M \begin{bmatrix} V & \hat{v} \end{bmatrix} = \begin{bmatrix} V^T M V & V^T M \hat{v} \\ \hat{v}^T M V & \hat{v}^T M \hat{v} \end{bmatrix}.$$

The terms $V^T M V$, $V^T M$ and $M V$ are calculated only once in the offline stage using V, and only the vector $M\hat{v}$ needs computing for a new parameter $\hat{p}$. The situation is similar for the other reduced quantities except for $\widehat{L}_r(p)$ due to the nonaffine parametrization of $L(p) = \mathcal{P} L \mathcal{P}$. To overcome this issue, an affine parametric approximation of $L(p)$ via, for example, DEIM [21] as we will see in Section 4.4.5, can be used to achieve efficient online computations.

2.2.3 Smaller number of parameters

Now we assume that $L(p)$ is parametrized with a smaller number of parameters. Let $p = [p_1\ p_2\ \cdots\ p_\nu]^T \in \Omega_\nu \subseteq \mathbb{R}^\nu$ and consider the parametrization

$$L(p) = \mathcal{P} L \mathcal{P} \quad \text{with} \quad \mathcal{P} = \begin{bmatrix} p_1 I_{n_1} & & \\ & \ddots & \\ & & p_\nu I_{n_\nu} \end{bmatrix}, \tag{2.34}$$

where $n_1 + \cdots + n_\nu = n$ and $\nu < n$. This can be viewed as some of the peak voltage magnitudes E_i varying together. This structure will drastically simplify the algorithmic considerations from Section 2.2.2. In (2.34) we can also set some p_i's to 1 to allow variations only in a subset set E_i's.

Proposition 2.5. *Consider the parametrization in (2.34). Define*

$$\mathbf{e}_k = \begin{bmatrix} 0_{(n_1+\cdots+n_{k-1})\times 1} \\ \mathbf{1}_{n_k} \\ 0_{(n_{k+1}+\cdots+n_\nu)\times 1} \end{bmatrix} \in \mathbb{R}^n, \qquad k = 1, 2, \ldots, \nu\ . \tag{2.35}$$

If $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_\nu\} \in$ *span*(V)*, then* $H_r(s,p)$ *retains the simple pole at zero and its corresponding parameter-dependent residue of* $H(s,p)$ *for every* $p \in \Omega_\nu$*.*

Proof. For any $\hat{p} \in \Omega_\nu$, the eigenvector of $L(\hat{p})$ corresponding to the zero eigenvalue can be

written as

$$\begin{aligned}\hat{v} = \hat{\mathcal{P}}^{-1}\mathbf{1}_n &= \begin{bmatrix} \frac{1}{p_1}I_{n_1} & & \\ & \ddots & \\ & & \frac{1}{p_\nu}I_{n_\nu} \end{bmatrix} \begin{bmatrix} \mathbf{1}_{n_1} \\ \vdots \\ \mathbf{1}_{n_\nu} \end{bmatrix} = \begin{bmatrix} \frac{1}{p_1}\mathbf{1}_{n_1} \\ \vdots \\ \frac{1}{p_\nu}\mathbf{1}_{n_\nu} \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{p_1}\mathbf{1}_{n_1} \\ 0_{(n-n_1)\times 1} \end{bmatrix} + \cdots + \begin{bmatrix} 0_{(n-n_\nu)\times 1} \\ \frac{1}{p_\nu}\mathbf{1}_{n_\nu} \end{bmatrix} \\ &= \frac{1}{p_1}\mathbf{e}_1 + \cdots + \frac{1}{p_\nu}\mathbf{e}_\nu \ .\end{aligned} \tag{2.36}$$

Therefore, if $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_\nu\} \in \text{span}(V)$, we have $\hat{v} \in \text{span}(V)$ for *every* $\hat{p} \in \Omega_\nu$ and the desired result follows from Theorem 2.3. □

Proposition 2.5 shows that for the parametrization (2.34), adding ν vectors to span(V) suffices to match the residue at $s = 0$ for *every* $p \in \Omega_\nu$ and augmenting the basis by a new vector for a given $\hat{p}$ as in Section 2.2.2 is no longer necessary. A *fixed* global basis V satisfying $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_\nu\} \in \text{span}(V)$ does the job for every $p \in \Omega_\nu$. Note that one needs ν to be modest so that the reduced dimension stays modest.

2.2.3.1 Algorithmic details for implementing Proposition 2.5

The global basis V in Proposition 2.5 can result from any model reduction method of choice. As long as the vectors $\{\mathbf{e}_1, \ldots, \mathbf{e}_\nu\}$ are added to its span, the result will hold.

We will form V as in (2.8) where the local bases result from the modified implementation of SOR-IRKA as described in Section 2.2.2. Given the parameter samples $p^{(i)}$ for $i = 1, \ldots, m$, let $v^{(i)}$ denote the eigenvector of $L(p^{(i)})$ corresponding to the zero eigenvalue. Our SOR-IRKA implementation will provide that $\{v^{(1)}, \ldots, v^{(m)}\} \in \text{span}(V)$. As shown in the proof

of Proposition 2.5, for any $\hat{p} \in \Omega_\nu$, $\hat{v} = \hat{\mathcal{P}}^{-1}\mathbf{1}_n$ is spanned by ν vectors. We will choose $m \geq \nu$ different parameter samples, obtaining a linearly independent set $\{v^{(1)}, \ldots, v^{(m)}\}$. Since these vectors are in the span of V, we will automatically satisfy the subspace condition in Proposition 2.5. Therefore, our construction of V via modified SOR-IRKA with $m \geq \nu$ parameter samples will guarantee bounded $\mathcal{H}_2$ and $\mathcal{H}_\infty$ error for every $p \in \Omega_\nu$ (including the unsampled parameters) without explicitly adding the vectors $\{\mathbf{e}_1, \ldots, \mathbf{e}_\nu\}$ to the reduction basis V.

2.3 Numerical results

We use a linearized model of 2736-bus Polish network [132] with $n = 2736$. We focus on a single-input single-output model with $B = C^T = [1 \quad 0 \quad \cdots \quad 0]^T \in \mathbb{R}^{n\times 1}$ and allow 15% variation in peak voltage magnitudes, i.e., $0.85 \leq p_i \leq 1.15$ in $L(p)$. Recall that $p_i = 1$ corresponds to the non-parametric unity voltage magnitude case ($E_i = 1$).

2.3.1 Case 1: single parameter

We start with the simple case

$$\mathcal{P} = \begin{bmatrix} p & \\ & I_{n-1} \end{bmatrix},$$

where p is a scalar; i.e., only E_1 in (1.40) is varying. We randomly pick two parameter values $p^{(i)}$ for $i = 1, 2$ as in Table 2.1, apply one-sided SOR-IRKA to obtain local bases $V^{(1)} \in \mathbb{R}^{n\times k_1}$

Table 2.1: Sample sets for case 1

Sample set	p
$p^{(1)}$	0.8536
$p^{(2)}$	1.1412

and $V^{(2)} \in \mathbb{R}^{n\times k_2}$ with $k_1 = k_2 = 20$. SOR-IRKA is modified based on Section 2.2.3.1 so that both $V^{(1)}$ and $V^{(2)}$ contain the eigenvectors of $L(p^{(1)})$ and $L(p^{(2)})$ corresponding to the zero eigenvalue in their span. Then, the global basis $V \in \mathbb{R}^{n\times r}$ is obtained by finding an orthogonal basis for $[V^{(1)} \;\; V^{(2)}]$ as in (2.8), leading to a final reduced model $H_r(s,p)$ with $r = 40$. Due to Proposition 2.5, $H_r(s,p)$ retains the residue at $s = 0$ and thus the error system $H(s,p) - H_r(s,p)$ is bounded in the $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norms for every $p \in [0.85\ 1.15]$. To illustrate the accuracy of $H_r(s,p)$, in Figure 2.1 we show the relative $\mathcal{H}_\infty$ error defined as

$$e(t) = \frac{\|H(s,p) - H_r(s,p)\|_\infty}{\|H_a(s,p)\|_\infty} \tag{2.37}$$

over the full parameter range. As the figure illustrates, the structure-preserving reduced model $H_r(s,p)$ is a high fidelity approximation to $H(s,p)$ model with a maximum relative error less than 5×10^{-3} over the full parameter range.

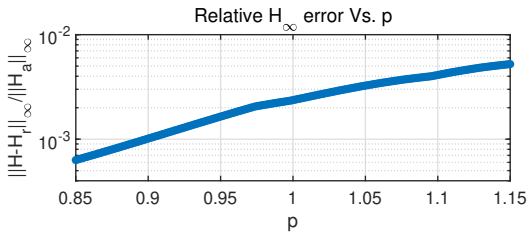


Figure 2.1: Example 2.3.1: Relative $\mathcal{H}_\infty$ error over the parameter domain

2.3.2 Case 2: two parameters

We consider a parametrization with $\nu = 2$ parameters p_1 and p_2 as

$$\mathcal{P} = \begin{bmatrix} p_1 I_{\frac{n}{2}} & \\ & p_2 I_{\frac{n}{2}} \end{bmatrix}.$$

We pick two random samples as in Table 2.2, and apply the modified SOR-IRKA to obtain

Table 2.2: Sample sets for case 2

Sample set	p_1	p_2
$p^{(1)}$	0.9572	0.93399
$p^{(2)}$	1.0304	0.9522

local bases $V^{(1)} \in \mathbb{R}^{n\times 20}$ and $V^{(2)} \in \mathbb{R}^{n\times 20}$. An orthogonalization of $[V^{(1)}\ V^{(2)}]$ leads to the global basis $V \in \mathbb{R}^{n\times 40}$, thus a reduced model $H_r(s,p)$ with $r = 40$. Due to Proposition 2.5 and the discussion in Section 2.2.3.1, $H_r(s,p)$ matches the residue at $s = 0$ and provides bounded $\mathcal{H}_2$ and $\mathcal{H}_\infty$ error throughout the whole domain $[p_1,\ p_2] \in \Omega_2 = [0.85\ 1.15] \times [0.85\ 1.15]$. Once again, to illustrate the accuracy of $H_r(s,p)$, in Figure 2.2 we show the relative $\mathcal{H}_\infty$ error in (2.37) over the full parameter space. As the figure illustrates, the structure-preserving reduced model $H_r(s,p)$ is a high fidelity approximation to $H(s,p)$ over the full parameter space with a maximum relative error less than 1.5×10^{-2}.

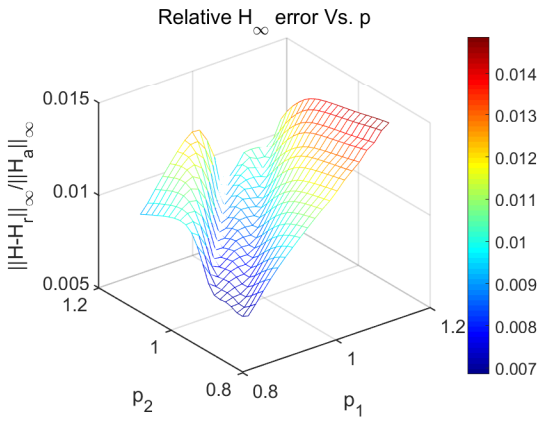


Figure 2.2: Example 2.3.2: Relative $\mathcal{H}_\infty$ error over the parameter domain

2.3.3 Case 3: four parameters

We now consider the four parameter case, p_1, p_2, p_3 and p_4 such that

$$\mathcal{P} = \text{diag}(p_1 I_{\frac{n}{4}},\ p_2 I_{\frac{n}{4}},\ p_3 I_{\frac{n}{4}},\ p_4 I_{\frac{n}{4}}).$$

We randomly pick four parameter sample sets as in Table 2.3. Using these samples, we

Table 2.3: Sample sets for case 3

Sample set	p_1	p_2	p_3	p_4
$p^{(1)}$	1.0967	0.8541	0.9399	0.887
$p^{(2)}$	0.9399	0.9146	1.0377	1.0459
$p^{(3)}$	0.9522	1.0713	0.9399	0.9572
$p^{(4)}$	1.0801	0.9399	1.0377	1.1029

apply the modified SOR-IRKA to obtain the local bases $V^{(i)} \in \mathbb{R}^{n\times 20}$; $i = \{1, 2, 3, 4\}$ and a parametric reduced model of order $r = 80$ ($V \in \mathbb{R}^{n\times 80}$). As in the previous example, the reduced model guarantees bounded error over the whole parameter space. To show the approximation quality, we pick 200 random samples in the four-dimensional parameter space, and depict the resulting relative $\mathcal{H}_\infty$ error in Figure 2.3, showing a maximum relative error less than 10^{-2} over this sample set.

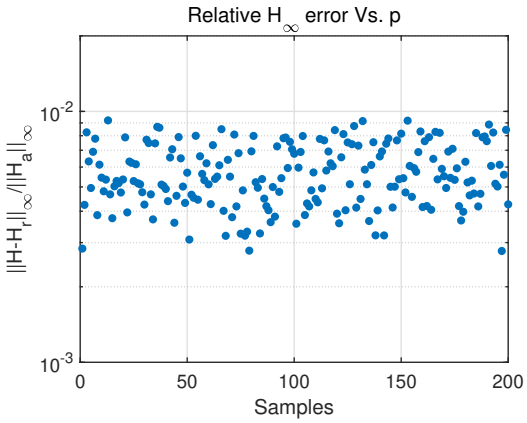


Figure 2.3: Example 2.3.3: Relative $\mathcal{H}_\infty$ error over 200 samples.

2.3.4 Adaptive parameter sampling

Instead of random sampling of the parameter sample sets, one can adaptively choose them via a greedy sampling method [21, 34, 57, 123]. Greedy sampling method can improve the

approximation error; however, it might be computationally expensive when the parameter space is large. This method adaptively chooses sample sets $p^{(j)}$ by finding the location in the predetermined parameter space where the estimate of the error in the reduced model is maximum. In other word, let $p^{(1)}$ be the current parameter sample set with the corresponding local basis $V^{(1)}$. Then, the next parameter sample set is chosen such that the relative $\mathcal{H}_\infty$ error between the parametric full order and reduced models in (2.37), is maximized. In other words,

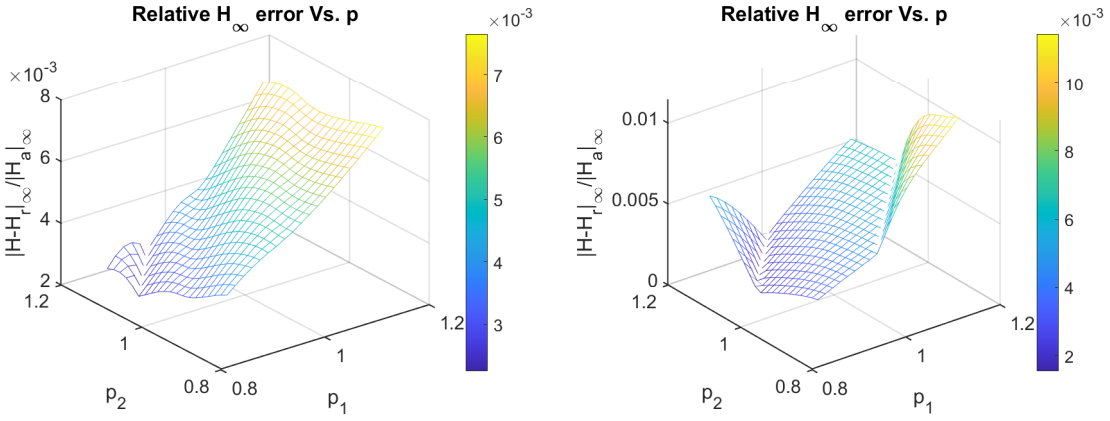
$$p^{(2)} = \arg\max_{p} \frac{\|H(s,p) - H_r(s,p)\|_\infty}{\|H_a(s,p)\|_\infty} \ .$$

The corresponding local basis is then added to the current local basis $V^{(1)}$ as in (2.8) to form the parametric reduced model (2.1). These steps are repeated until we achieve a pre-specified error tolerance , i.e., desired approximation error. In most cases, to avoid evaluating the full model at each parameter set (especially for large parameter space), instead of using the actual reduced model error, one can use an approximation of that in finding the worst-case parameter value [21].

In this section, we apply the greedy parameter selection to our second example in Section 2.3.2. For a tolerance of 10^{-2}, the greedy algorithm stops after two iterations with the corresponding parameter sample sets shown in Table 2.4. This leads to a parametric reduced model $H_r(s,p)$ of order $r = 60$ with a maximum relative error less than 7.65×10^{-3} over the parameter domain (See Figure 2.4a). Although the greedy algorithm led to a higher order reduced model, one can observe from comparing Figures 2.2 and 2.4b that even with a single iteration of the greedy algorithm, the maximum relative error of the parametric reduced model has decreased from 1.5×10^{-2} in Figure 2.2 to 1.15×10^{-2} in Figure 2.4b.

Table 2.4: Sample sets for case 2

Greedy iteration	Sample set	p_1	p_2
	$p^{(0)}$	0.85	0.85
1	$p^{(1)}$	1.05	1.15
2	$p^{(2)}$	1.15	0.85



(a) Two iterations of greedy algorithm (b) Single iteration of greedy algorithm

Figure 2.4: Example 2.3.2: Relative $\mathcal{H}_\infty$ error over the parameter domain

2.4 Summary

We have developed a structure-preserving parametric model reduction approach for linearized swing equations using a global basis framework and interpolatory $\mathcal{H}_2$ model reduction. We have established the subspace conditions for the reduction basis so that the error system is an $\mathcal{H}_2$ and $\mathcal{H}_\infty$ function over the entire parameter space. Two examples have been used to illustrate the the efficiency of our approach.

Chapter 3

Converting Power Networks Dynamics into Quadratic Dynamics

In this chapter, we drive a quadratic representation for the power network dynamics

$$\mathcal{M}\ddot{\delta}(t) + \mathcal{D}\dot{\delta}(t) + f(\delta) = \mathcal{B}u(t), \tag{1.2}$$

$$y(t) = \mathcal{C}\delta(t), \tag{1.3}$$

by introducing a lifting transformation. These lifting transformation and quadratic dynamics play an important role in the model reduction frameworks of Chapters 4 and 5. We also investigate the effect of the lifting transformation on the equilibrium points.

Most of the analysis in this chapter are from the preprint [110].

3.1 Quadratic representation

A large class of nonlinear systems, such as those with smooth nonlinearities, e.g., exponential, trigonometric, can indeed be represented as quadratic-bilinear systems by introducing new variables (in a lifting map) [83], [59], [19], [106], [102]. Converting a nonlinear model to its equivalent quadratic form provides an opportunity to perform input-independent model reduction techniques where a system-theoretic norm is defined; see, e.g., [18, 20, 23]. Although

this transformation is not unique in most cases, it is exact, i.e., there is no approximation error [20]. Second-order model (1.1) inherently contains a quadratic nonlinearity in $f(\delta)$ and the nonlinear system (1.2) can be lifted to a quadratic dynamic.

Towards this goal, expand the nonlinearity $\sin(\delta_i - \delta_j - \gamma_{ij})$ using the trigonometric identity

$$\begin{aligned}\sin(\delta_i - \delta_j - \gamma_{ij}) = &(\sin(\delta_i)\cos(\delta_j) - \cos(\delta_i)\sin(\delta_j))\cos(\gamma_{ij}) \\ &-(\cos(\delta_i)\cos(\delta_j) + \sin(\delta_i)\sin(\delta_j))\sin(\gamma_{ij}).\end{aligned} \tag{3.1}$$

Note that equation (3.1) is quadratic in $\sin(\delta_i)$ and $\cos(\delta_i)$. This directly hints at choosing $\sin(\delta)$ and $\cos(\delta)$ as auxiliary variables in a new state vector to convert the nonlinear dynamics to a quadratic one. This is precisely what has been done in [86], [106] to find a quadratic representation for (1.2)-(1.3). However, since the transformation, the so-called lifting, is not unique and we have a slightly different form in the resulting state-space form, we include its derivation here for completeness.

Lemma 3.1. *Given the second-order dynamics (1.2)-(1.3), define the new lifted state vector $q(t) \in \mathbb{R}^{4n}$ as*

$$q(t) = \begin{bmatrix}\delta(t)^T & \dot{\delta}(t)^T & \sin(\delta(t))^T & \cos(\delta(t))^T\end{bmatrix}^T. \tag{3.2}$$

For the new state-variable $q(t)$ in (3.2), the original network dynamics (1.2)-(1.3) can be written exactly as a quadratic nonlinear system of the form

$$\begin{aligned}E\dot{q}(t) &= Aq(t) + H(q(t) \otimes q(t)) + Bu(t) \\ y(t) &= Cq(t),\end{aligned} \tag{3.3}$$

where $E, A \in \mathbb{R}^{4n\times 4n}$, $B \in \mathbb{R}^{4n}$, $C \in \mathbb{R}^{p\times 4n}$, $H \in \mathbb{R}^{4n\times(4n)^2}$, and $\otimes$ denotes the Kronecker

product.

Proof. The proof is constructive, i.e., it will show how the matrices E, A, H, B and C in (3.3) will be constructed. Using the new state vector $q(t)$ in (3.2), we can directly rewrite the original output $y(t) = \mathcal{C}\delta(t)$ as

$$y(t) = Cq(t) \quad \text{with} \quad C = \begin{bmatrix} \mathcal{C} & 0_{p\times n} & 0_{p\times n} & 0_{p\times n} \end{bmatrix} \in \mathbb{R}^{p\times 4n}. \tag{3.4}$$

To simplify the notation, define

$$q_1 = \delta(t), \quad q_2 = \dot{\delta}(t), \quad q_3 = \sin(\delta(t)), \quad \text{and} \quad q_4 = \cos(\delta(t)) \tag{3.5}$$

so that

$$q(t) = \begin{bmatrix} \delta(t) \\ \dot{\delta(t)} \\ \sin(\delta(t)) \\ \cos(\delta(t)) \end{bmatrix} = \begin{bmatrix} q_1(t) \\ q_2(t) \\ q_3(t) \\ q_4(t) \end{bmatrix}. \tag{3.6}$$

Then, it follows from (3.5) and (1.2) that

$$\dot{q}_1 = q_2, \tag{3.7}$$

$$\mathcal{M}\dot{q}_2 = -\mathcal{D}q_2 - f(q_1) + \mathcal{B}u, \tag{3.8}$$

$$\dot{q}_3 = q_2 \circ q_4, \tag{3.9}$$

$$\dot{q}_4 = -q_2 \circ q_3, \tag{3.10}$$

where $\circ$ denotes the Hadamard product.

We start by writing $\dot{q}_3$ in (3.9) and $\dot{q}_4$ in (3.10) as

$$\dot{q}_3 = \Phi(q_2 \otimes q_4) \quad \text{and} \quad \dot{q}_4 = -\Phi(q_2 \otimes q_3) \tag{3.11}$$

where

$$\Phi = \begin{bmatrix} \Phi_1 & \Phi_2 & \dots & \Phi_n \end{bmatrix} \in \mathbb{R}^{n \times n^2} \tag{3.12}$$

with

$$\Phi_k = e_k \, e_k^T \in \mathbb{R}^{n \times n} \quad \text{for} \quad k = 1, \dots, n,$$

where e_k is the k^{th} column of the identity matrix I_n.

Next, we write the nonlinearity $f(q_1)$ in (3.8) as a quadratic term. Towards this goal, we use the expansion (3.1) in (1.5) to obtain

$$\begin{aligned} \big(f(q_1)\big)_i = &\sum_{\substack{j=1 \\ j \neq i}}^{n} K_{ij} \left((q_3)_i (q_4)_j - (q_4)_i (q_3)_j \right) \cos(\gamma_{ij}) - \sum_{\substack{j=1 \\ j \neq i}}^{n} K_{ij} ((q_4)_i (q_4)_j) \sin(\gamma_{ij}) \\ &- \sum_{\substack{j=1 \\ j \neq i}}^{n} K_{ij} \left((q_3)_i (q_3)_j \right) \sin(\gamma_{ij}). \end{aligned} \tag{3.13}$$

Using (3.13), we can write the vector $f(q_1)$ compactly as

$$f(q_1) = -\left(Z(q_3 \otimes q_4) + \Psi(q_4 \otimes q_4) + \Psi(q_3 \otimes q_3) \right), \tag{3.14}$$

where

$$Z = \begin{bmatrix} Z_1 & Z_2 & \dots & Z_n \end{bmatrix} \in \mathbb{R}^{n \times n^2}$$

consists of n block matrices $Z_k \in \mathbb{R}^{n \times n}$ defined as

$$Z_k(i,j) = \begin{cases} K_{kj} \cos(\gamma_{kj}), & i = j = 1, ..., n, \ j \neq k \\ -K_{kj} \cos(\gamma_{kj}), & i = k, j = 1, ..., n, \ j \neq k \\ 0 & \text{otherwise} \end{cases}. \tag{3.15}$$

Similarly,

$$\Psi = \begin{bmatrix} \Psi_1 & \Psi_2 & \dots & \Psi_n \end{bmatrix} \in \mathbb{R}^{n \times n^2}$$

and for the k^{th} block of Ψ denoted by $\Psi_k \in \mathbb{R}^{n \times n}$, we have

$$\Psi_k(i,j) = \begin{cases} -\frac{1}{2} K_{kj} \sin(\gamma_{kj}), & i = j = 1, ..., n, \ j \neq k \\ -\frac{1}{2} K_{kj} \sin(\gamma_{kj}), & i = k, j = 1, ..., n, \ j \neq k \\ 0 & \text{otherwise} \end{cases}. \tag{3.16}$$

Using (3.11) and (3.14), we now rewrite (3.7)-(3.10) as

$$\dot{q}_1 = q_2, \tag{3.17}$$

$$\mathcal{M}\dot{q}_2 = -\mathcal{D}q_2 + Z(q_3 \otimes q_4) + \Psi(q_4 \otimes q_4) + \Psi(q_3 \otimes q_3) + \mathcal{B}u, \tag{3.18}$$

$$\dot{q}_3 = \Phi(q_2 \otimes q_4), \tag{3.19}$$

$$\dot{q}_4 = -\Phi(q_2 \otimes q_3). \tag{3.20}$$

Note that the new formulation of the dynamics in (3.17)-(3.20) only contains linear and

quadratic terms in q and the input mapping is linear. The output mapping C is already established in (3.4). Therefore, there exist matrices E, A, H, B and C such that the original power network dynamics can be written as a quadratic nonlinear system as in (3.3), thus proving the desired result. However, to give a constructive proof, we continue to show how the state-space matrices in (3.3) can be explicitly constructed.

It immediately follows from (3.17)-(3.20) that $E \in \mathbb{R}^{4n\times 4n}$, $A \in \mathbb{R}^{4n\times 4n}$ and $B \in \mathbb{R}^{4n}$ in (3.3) (corresponding to the linear parts of the dynamics) are given by

$$E = \text{blkdiag}(I_n, \mathcal{M}, I_n, I_n), \quad B = \begin{bmatrix} 0_{n\times 1} \\ \mathcal{B} \\ 0_{n\times 1} \\ 0_{n\times 1} \end{bmatrix} \quad \text{and} \quad A = \begin{bmatrix} 0_{n\times n} & I_n & 0_{n\times n} & 0_{n\times n} \\ 0_{n\times n} & -\mathcal{D} & 0_{n\times n} & 0_{n\times n} \\ 0_{n\times n} & 0_{n\times n} & 0_{n\times n} & 0_{n\times n} \\ 0_{n\times n} & 0_{n\times n} & 0_{n\times n} & 0_{n\times n} \end{bmatrix}, \tag{3.21}$$

where $I_n \in \mathbb{R}^{n\times n}$ denotes the identity matrix.

To show how H is constructed, we first define a revised version of the Kronecker product as

$$q \,\widetilde{\otimes}\, q = \begin{bmatrix} q_1 \,\widetilde{\otimes}\, q \\ q_2 \,\widetilde{\otimes}\, q \\ q_3 \,\widetilde{\otimes}\, q \\ q_4 \,\widetilde{\otimes}\, q \end{bmatrix} \quad \text{where} \quad q_i \,\widetilde{\otimes}\, q = \begin{bmatrix} q_i \otimes q_1 \\ q_i \otimes q_2 \\ q_i \otimes q_3 \\ q_i \otimes q_4 \end{bmatrix}, \quad i = 1, \ldots, 4. \tag{3.22}$$

Clearly, (3.22) is simply a permuted form of the regular Kronecker product. It is introduced here to make the derivation of H easier. The quadratic terms in (3.18)-(3.20), i.e., $Z(q_3 \otimes q_4) + \Psi(q_4 \otimes q_4) + \Psi(q_3 \otimes q_3), \Phi(q_2 \otimes q_4)$ and $-\Phi(q_2 \otimes q_3)$ can be written as

$$\tilde{H}(q \,\widetilde{\otimes}\, q), \quad \text{where} \quad \tilde{H} \in \mathbb{R}^{4n\times(4n)^2}. \tag{3.23}$$

Decompose $\tilde{H}$ into four sub-matrices:

$$\tilde{H} = \begin{bmatrix} \tilde{H}_1 & \tilde{H}_2 & \tilde{H}_3 & \tilde{H}_4 \end{bmatrix}, \tag{3.24}$$

where $\tilde{H}_k \in \mathbb{R}^{4n \times 4n^2}$ for $k = 1, 2, 3, 4$. The first submatrix $\tilde{H}_1$ corresponds to the first block of (3.22), i.e,

$$q_1 \tilde{\otimes} q = \begin{bmatrix} q_1 \otimes q_1; & q_1 \otimes q_2; & q_1 \otimes q_3; & q_1 \otimes q_4 \end{bmatrix}.$$

There is no $q_1 \otimes q_i$ term in (3.17)-(3.20). Thus, we set

$$\tilde{H}_1 = 0_{4n \times 4n^2}. \tag{3.25}$$

Similarly, the matrix $\tilde{H}_2$ corresponds to the second block of (3.22), i.e., $q_2 \tilde{\otimes} q$. We can see from (3.17)-(3.20) that two terms, namely $q_2 \otimes q_3$ and $q_2 \otimes q_4$, exist in (3.19) and (3.20), respectively. One can rewrite (3.19) as

$$\dot{q}_3 = \frac{\Phi}{2}(q_2 \otimes q_4) + \frac{\Phi}{2}(q_4 \otimes q_2)$$

and allocate the first term, i.e., $\frac{\Phi}{2}(q_2 \otimes q_4)$ in $\tilde{H}(q \,\widetilde{\otimes}\, q)$ to $\tilde{H}_2$. More specifically and by using the MATLAB notation for row and column indices, we set the

$$\tilde{H}_2(2n + 1 : 3n, 3n^2 + 1 : 4n^2) = \frac{\Phi}{2}.$$

(The second term in $\dot{q}_3$, i.e.,$\frac{\Phi}{2}(q_4 \otimes q_2)$) will be matched by $\tilde{H}_4$ below).

Similarly, we can write (3.20) as

$$\dot{q}_4 = -\frac{\Phi}{2}(q_2 \otimes q_3) - \frac{\Phi}{2}(q_3 \otimes q_2)$$

and allocate the first term to $\tilde{H}_2$ by setting

$$\tilde{H}_2(3n+1:4n, 2n^2+1:3n^2) = -\frac{\Phi}{2}$$

(the second term will be matched via $\tilde{H}_3$). Thus, we set

$$\tilde{H}_2 = \begin{bmatrix} 0_{n\times n^2} & 0_{n\times n^2} & 0_{n\times n^2} & 0_{n\times n^2} \\ 0_{n\times n^2} & 0_{n\times n^2} & 0_{n\times n^2} & 0_{n\times n^2} \\ 0_{n\times n^2} & 0_{n\times n^2} & 0_{n\times n^2} & \frac{\Phi}{2} \\ 0_{n\times n^2} & 0_{n\times n^2} & \frac{-\Phi}{2} & 0_{n\times n^2} \end{bmatrix}. \tag{3.26}$$

Following similar arguments, the matrix $\tilde{H}_3$ corresponds to the third block of (3.22), i.e, $q_3\tilde{\otimes}q$. According to (3.17)-(3.20), one can see that two terms $q_3 \otimes q_3$ and $q_3 \otimes q_4$ exist in (3.18). By rewriting $\dot{q}_2$ in (3.18), we obtain

$$\dot{q}_2 = -\mathcal{D}q_2 + \frac{Z}{2}(q_3 \otimes q_4) + \Psi(q_3 \otimes q_3) - \frac{Z}{2}(q_4 \otimes q_3) + \Psi(q_4 \otimes q_4) + \mathcal{B}u,$$

where the two terms $\frac{Z}{2}(q_3\otimes q_4)+\Psi(q_3\otimes q_3)$ will be match by $\tilde{H}_3$, and $-\frac{Z}{2}(q_4\otimes q_3)+\Psi(q_4\otimes q_4)$ will be match by $\tilde{H}_4$ such that

$$\tilde{H}_3 = \begin{bmatrix} 0_{n\times n^2} & 0_{n\times n^2} & 0_{n\times n^2} & 0_{n\times n^2} \\ 0_{n\times n^2} & 0_{n\times n^2} & \Psi & \frac{Z}{2} \\ 0_{n\times n^2} & 0_{n\times n^2} & 0_{n\times n^2} & 0_{n\times n^2} \\ 0_{n\times n^2} & \frac{-\Phi}{2} & 0_{n\times n^2} & 0_{n\times n^2} \end{bmatrix} \quad \text{and} \quad \tilde{H}_4 = \begin{bmatrix} 0_{n\times n^2} & 0_{n\times n^2} & 0_{n\times n^2} & 0_{n\times n^2} \\ 0_{n\times n^2} & 0_{n\times n^2} & \frac{-Z}{2} & \Psi \\ 0_{n\times n^2} & \frac{\Phi}{2} & 0_{n\times n^2} & 0_{n\times n^2} \\ 0_{n\times n^2} & 0_{n\times n^2} & 0_{n\times n^2} & 0_{n\times n^2} \end{bmatrix}. \tag{3.27}$$

Since (3.22) is a permuted form of $(q \otimes q)$, there exists a permutation matrix $P_q \in \mathbb{R}^{4n^2\times 4n^2}$ such that

$$(q \,\widetilde{\otimes}\, q) = P^T (q \otimes q), \tag{3.28}$$

where $P = \text{blkdiag}\,(P_q, P_q, P_q, P_q)$. Then, H in (3.3) is

$$H = \tilde{H} P^T. \tag{3.29}$$

□

3.2 Equilibrium analysis

So far we have represented the power network dynamics in two formats: as second-order dynamics in (1.2) and as quadratic dynamics in (3.3). In this section, we investigate the effect of the lifting transformation (3.2) on the equilibrium points.

To find the equilibrium points of (1.2), first we transform it to a first-order form by defining a state vector

$$\mathcal{X}(t) = \begin{bmatrix} \mathcal{X}_1(t) \\ \mathcal{X}_2(t) \end{bmatrix} = \begin{bmatrix} \delta(t) \\ \dot{\delta}(t) \end{bmatrix} \in \mathbb{R}^{2n}$$

and recalling that $u(t) = 1$ to obtain

$$\dot{\mathcal{X}}(t) = \begin{bmatrix} 0_{n\times n} & I_n \\ 0_{n\times n} & -\mathcal{M}^{-1}\mathcal{D} \end{bmatrix} \mathcal{X}(t) + \begin{bmatrix} 0_{n\times 1} \\ -\mathcal{M}^{-1} f(\mathcal{X}_1) \end{bmatrix} + \begin{bmatrix} 0_{n\times 1} \\ \mathcal{M}^{-1}\mathcal{B} \end{bmatrix}. \tag{3.30}$$

Let $\mathcal{X}^\star = \begin{bmatrix} \mathcal{X}_1^\star \\ \mathcal{X}_2^\star \end{bmatrix}$ be an equilibrium point of (3.30). By setting $\dot{\mathcal{X}}(t) = 0$, we obtain

$$\mathcal{X}_2^\star = 0,$$
$$-\mathcal{M}^{-1}\mathcal{D}\mathcal{X}_2^\star - \mathcal{M}^{-1}f(\mathcal{X}_1^\star) + \mathcal{M}^{-1}\mathcal{B} = 0$$

that leads to

$$\mathcal{X}_2^\star = 0,$$
$$\mathcal{M}^{-1}(-f(\mathcal{X}_1^\star) + \mathcal{B}) = 0 \rightarrow f(\mathcal{X}_1^\star) = \mathcal{B}$$

Hence, $\mathcal{X}^\star$ satisfies

$$\mathcal{X}^\star = \begin{bmatrix} \mathcal{X}_1^\star \\ \mathcal{X}_2^\star \end{bmatrix} = \begin{bmatrix} \delta^\star \\ 0_{n\times 1} \end{bmatrix} \quad \text{where} \quad f(\delta^\star) = \mathcal{B}. \tag{3.31}$$

Lemma 3.2. *Let* $\mathcal{X}^\star = \begin{bmatrix} \delta^\star \\ 0_{n\times 1} \end{bmatrix}$ *be an equilibrium point of the original dynamics* (3.30). *Then,*

$$q^\star = \begin{bmatrix} q_1^\star \\ q_2^\star \\ q_3^\star \\ q_4^\star \end{bmatrix} = \begin{bmatrix} \delta^\star \\ 0_{n\times 1} \\ \sin(\delta^\star) \\ \cos(\delta^\star) \end{bmatrix} \tag{3.32}$$

is an equilibrium point of (3.3). *Similarly, let*

$$q^\star = \begin{bmatrix} q_1^\star \\ q_2^\star \\ q_3^\star \\ q_4^\star \end{bmatrix} \tag{3.33}$$

be an equilibrium point of (3.3). *Then,* $q_2^\star = 0$ *and* $\mathcal{X}^\star = \begin{bmatrix} q_1^\star \\ 0_{n\times 1} \end{bmatrix}$ *is an equilibrium point of* (3.30).

Proof. Rewrite (3.30) as

$$\dot{\mathcal{X}} = \mathcal{F}(\mathcal{X}), \tag{3.34}$$

where $\mathcal{F} : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$. If $\mathcal{X}^\star$ in (3.31) is an equilibrium of (3.34), then we have

$$\mathcal{F}(\mathcal{X}^\star) = 0. \tag{3.35}$$

Now, let us write (3.2) as

$$q = \mathcal{T}(\mathcal{X}), \tag{3.36}$$

where $\mathcal{T}$ is the quadratic lifting in (3.34), given by

$$\mathcal{T} : \mathcal{X} = \begin{pmatrix} \mathcal{X}_1 \\ \mathcal{X}_2 \end{pmatrix} \rightarrow q = \begin{pmatrix} \mathcal{X}_1 \\ \mathcal{X}_2 \\ \sin(\mathcal{X}_1) \\ \cos(\mathcal{X}_1) \end{pmatrix}. \tag{3.37}$$

Differentiate (3.36) to obtain

$$\dot{q} = J_{\mathcal{T}} \dot{\mathcal{X}} = J_{\mathcal{T}} \mathcal{F}(\mathcal{X}) := \mathcal{G}(q), \tag{3.38}$$

where the Jacobian J_T is given by

$$J_{\mathcal{T}} = \begin{bmatrix} I_n & 0_{n\times n} \\ 0_{n\times n} & I_n \\ \text{diag}(\cos(\mathcal{X}_1)) & 0_{n\times n} \\ -\text{diag}(\sin(\mathcal{X}_1)) & 0_{n\times n} \end{bmatrix} \in \mathbb{R}^{4n\times 2n}. \tag{3.39}$$

Insert (3.35) into (3.38) to obtain

$$\dot{q} = \mathcal{G}(q^\star) = J_{\mathcal{T}} \mathcal{F}(\mathcal{X}^\star) = 0, \tag{3.40}$$

which shows that (3.32) is an equilibrium of (3.3).

Now let $q^\star$ in (3.33) be an equilibrium of (3.3). Then from (3.38), we obtain

$$\mathcal{G}(q^\star) = J_{\mathcal{T}} \mathcal{F}(\mathcal{X}^\star) = 0. \tag{3.41}$$

Since the Jacobian (3.39) has full column rank, we have,

$$\mathcal{F}(\mathcal{X}^\star) = 0, \tag{3.42}$$

which by (3.34) yields that (3.31) is an equilibrium of (3.30). The fact that $q_2^\star = 0$ follows from the second row block of

$$\mathcal{F}(\mathcal{X}^\star) = \begin{bmatrix} 0_{n\times n} & I_n \\ 0_{n\times n} & -\mathcal{M}^{-1}\mathcal{D} \end{bmatrix} \begin{bmatrix} \mathcal{X}_1^\star \\ \mathcal{X}_2^\star \end{bmatrix} + \begin{bmatrix} 0_{n\times 1} \\ -\mathcal{M}^{-1} f(\mathcal{X}_1^\star) \end{bmatrix} + \begin{bmatrix} 0_{n\times 1} \\ \mathcal{M}^{-1}\mathcal{B} \end{bmatrix} = 0, \tag{3.43}$$

which yields to $\mathcal{X}_2^\star = 0$. □

Chapter 4

Structure-Preserving Model Reduction for Nonlinear Power Networks

As we briefly discussed in Chapter 3, for systems with quadratic nonlinearities, effective, and in some cases optimal, systems-theoretic model reductions methods already exist. Therefore, in obtaining high-fidelity reduced models, we can significantly benefit from a quadratic representation of the nonlinear swing dynamics. In this chapter, we seek to develop a nonlinear structure-preserving model reduction approach such that the reduced model preserves the physically-meaningful second-order structure, i.e., we seek to obtain a reduced second-order model, not a reduced quadratic model. Therefore, by exploiting the inherent structure of the model reduction bases obtained for the quadratic model, we form the final reduction base $\mathcal{V}$ to construct the final structure preserving reduced model (4.3). This is achieved in Sections 4.2– 4.3.

Much of the material of this chapter has already appeared in the submitted manuscript [110].

4.1 Structure-preserving model reduction approach

Recall the nonlinear dynamics

$$\mathcal{M}\ddot{\delta}(t) + \mathcal{D}\dot{\delta}(t) + f(\delta) = \mathcal{B}u(t), \tag{1.2}$$

$$y(t) = \mathcal{C}\delta(t). \tag{1.3}$$

As previously stated, we seek to develop a nonlinear structure-preserving model reduction approach. In other words, we directly reduce the second-order dynamics (1.2)-(1.3) to obtain a reduced second-order system with dimension $r \ll n$ such that it has the form

$$\mathcal{M}_r\ddot{\delta}_r(t) + \mathcal{D}_r\dot{\delta}_r(t) + f_r(\delta_r) = \mathcal{B}_r u(t), \tag{4.1}$$

$$y_r(t) = \mathcal{C}_r\delta_r(t). \tag{4.2}$$

where $\delta_r \in \mathbb{R}^r$ is the reduced state; $\mathcal{M}_r, \mathcal{D}_r \in \mathbb{R}^{r\times r}$; $\mathcal{B}_r \in \mathbb{R}^r$; $f_r : \mathbb{R}^r \to \mathbb{R}^r$; and $\mathcal{C}_r \in \mathbb{R}^{p\times r}$. The goal of this structure-preserving model reduction process is that the reduced output $y_r(t)$ approximates the original output $y(t)$ with high fidelity.

Given the full-order dynamics (1.2)-(1.3), the most common approach for constructing the reduced model (4.1)-(4.2) is the Petrov-Galerkin projection framework: Construct two model reduction bases $\mathcal{W}, \mathcal{V} \in \mathbb{R}^{n\times r}$ such that $\delta(t) \approx \mathcal{V}\delta_r(t)$. Insert this approximation of $\delta(t)$ into (1.2)

$$\mathcal{M}\mathcal{V}\ddot{\delta}_r(t) + \mathcal{D}\mathcal{V}\dot{\delta}_r(t) + f(\mathcal{V}\delta_r(t)) = \mathcal{B}u(t)$$

and enforce a Petrov-Galerkin condition on the residual

$$\mathcal{W}^T\Big(\mathcal{M}\mathcal{V}\ddot{\delta}_r(t)+\mathcal{D}\mathcal{V}\dot{\delta}_r(t)+f(\mathcal{V}\delta_r(t))-\mathcal{B}u(t)\Big)=0$$

to obtain the reduced matrices in (4.1) as

$$\begin{aligned}\mathcal{M}_r &= \mathcal{W}^T\mathcal{M}\mathcal{V},\ \mathcal{D}_r=\mathcal{W}^T\mathcal{D}\mathcal{V},\ f_r(\delta_r)=\mathcal{W}^T f(\mathcal{V}\delta_r),\\ \mathcal{B}_r &= \mathcal{W}^T\mathcal{B},\ \text{and}\ \mathcal{C}_r=\mathcal{C}\mathcal{V}.\end{aligned}\tag{4.3}$$

Note that although the term $f_r(\delta_r)\in\mathbb{R}^r$ in (4.3) lies in r dimensional subspace, it requires evaluating the original nonlinear function f at $\mathcal{V}\delta_r\in\mathbb{R}^n$. Hence, it is usually further approximated to resolve this lifting bottleneck [41] that will be addressed in Section 4.4.5. The reduced model (4.1) should preserve the symmetry and positive definiteness of $\mathcal{M}$ and $\mathcal{D}$. In our framework, we achieve this goal by setting $\mathcal{W}=\mathcal{V}$. Therefore, we obtain the reduced model via a Galerkin projection. However, unlike the usual Galerkin projection formulation for model reduction where the subspace $\mathcal{V}$ (and thus $\mathcal{W}$) only contains information from the dynamics equation (1.2) and ignores the output equation (1.3), by taking advantage of the special structure arising in the network dynamics, our one-sided Galerkin projection will contain information from both (1.2) and (1.3). These issues are discussed in detail in Section 4.3.

Obviously, the accuracy of the structure-preserving reduced model (4.1) depends on the proper choice of $\mathcal{V}$. For general nonlinearities, POD [65], [80] is usually the method of choice. However, for special classes of nonlinear systems such as bilinear, quadratic and quadratic-bilinear systems, one can indeed use well-established system theoretical methods; see, e.g., [18, 20, 23]. Recall that we have represented the original nonlinear swing dynamics (1.2)-(1.3) as an equivalent quadratic dynamical system (3.3) in Chapter 3. In the following

section, we will investigate the model reduction problem for those systems in more details.

4.2 Model reduction for quadratic systems

Consider the quadratic dynamical system

$$\Sigma := \begin{cases} E\dot{x}(t) = Ax(t) + H(x(t) \otimes x(t)) + Bu(t) \\ y(t) = Cx(t), \quad x(0) = 0 \end{cases}, \tag{4.4}$$

where E and $A \in \mathbb{R}^{N\times N}$, $B \in \mathbb{R}^{N\times l}$, $C \in \mathbb{R}^{p\times N}$ and $H \in \mathbb{R}^{N\times N^2}$. The matrix H can be viewed as the mode-1 matricization of a third-order tensor $\mathcal{H} \in \mathbb{R}^{N\times N\times N}$. In other words, let $\mathcal{H}_i \in \mathbb{R}^{N\times N}$ for $i = 1, \ldots, N$ be the frontal slices of $\mathcal{H}$. Then,

$$H = \mathcal{H}^{(1)} = \begin{bmatrix} \mathcal{H}_1 & \mathcal{H}_2 & \ldots & \mathcal{H}_N \end{bmatrix}, \tag{4.5}$$

where we introduce the notation $\mathcal{H}^{(1)}$ to denote the mode-1 matricization. Similarly, for the mode-2 matricization $\mathcal{H}^{(2)}$, we have

$$\mathcal{H}^{(2)} = \begin{bmatrix} \mathcal{H}_1^T & \mathcal{H}_2^T & \ldots & \mathcal{H}_N^T \end{bmatrix} \in \mathbb{R}^{N\times N^2}. \tag{4.6}$$

For the cases where the state-space dimension N is large, the goal is to find a reduced quadratic system

$$\Sigma_r := \begin{cases} E_r\dot{x_r}(t) = A_r x_r(t) + H_r(x_r(t) \otimes x_r(t)) + B_r u(t) \\ y_r(t) = C_r x_r(t), \quad x_r(0) = 0 \end{cases}, \tag{4.7}$$

where $E_r, A_r \in \mathbb{R}^{r_q \times r_q}$ $H_r \in \mathbb{R}^{r_q \times r_q^2}$, $C_r \in \mathbb{R}^{p \times r_q}$ and $B_r \in \mathbb{R}^{r_q \times l}$ with $r_q \ll N$ such that $y(t) \approx y_r(t)$ for a wide range of inputs. We construct the reduce system (4.7) via projection: Construct two model reduction bases $V, W \in \mathbb{R}^{N \times r_q}$ such that the reduced matrices in (4.7) are given by

$$\begin{aligned} &E_r = W^T E V,\ A_r = W^T A V,\ H_r = W^T H(V \otimes V), \\ &B_r = W^T B,\ C_r = CV. \end{aligned} \tag{4.8}$$

Note that in practice it is undesirable to explicitly formulate $V \otimes V$ for large scale systems in the reduced matrix H_r. There are efficient methods as proposed in [19] and [23] to avoid this computational issue. The former is based on the properties of tensor matricizations, and the latter uses a particular sparsity structure of H.

The quality of the reduced system (4.7) depends on the choice of the model reduction bases V and W. There are various model reduction techniques to determine theses bases including trajectory-based method such as POD (see, e.g.,[65], [80]) and trajectory piece-wise linear method [105]; and input-independent systems theoretic methods for quadratic systems such as balanced truncation [20], [106], interpolatory projections [17], [8], [35], [97], [98] and $\mathcal{H}_2$-quasi-optimal model reduction [23]. In this chapter, we employ the $\mathcal{H}_2$-based approach in [23].

4.2.1 $\mathcal{H}_2$-based model reduction of quadratic nonlinearities

Consider the original quadratic dynamics (4.4) with an asymptotically stable matrix $E^{-1}A$, i.e., eigenvalues of $E^{-1}A$ have negative real parts . The output $y(t)$ of the quadratic dynamics

(4.4) can be written as [23]

$$\begin{aligned} y(t) = &\int_0^t Ce^{E^{-1}At_1}E^{-1}Bu(t-t_1)dt_1 \\ &+ \int_0^t \int_0^{t-t_3} \int_0^{t-t_3} Ce^{E^{-1}At_3}E^{-1}H(e^{E^{-1}At_2}E^{-1}B \otimes e^{E^{-1}At_1}E^{-1}B) \\ &\times u(t-t_2-t_3) \otimes u(t-t_1-t_2)dt_1dt_2dt_3 + \dots \end{aligned} \tag{4.9}$$

Using the three leading Volterra kernels of (4.9)

$$\begin{aligned} & h_1(t_1) = Ce^{E^{-1}At_1}E^{-1}B, \quad h_2(t_1,t_2) = 0 \\ \text{and} \quad & h_3(t_1,t_2,t_3) = Ce^{E^{-1}At_3}E^{-1}H(e^{E^{-1}At_2}E^{-1}B \otimes e^{E^{-1}At_1}E^{-1}B) \end{aligned}$$

define a truncated $\mathcal{H}_2$ norm for (4.4) denoted by $\|\Sigma\|_{\mathcal{H}_2}(\tau)$ as [23]:

$$\|\Sigma\|^2_{\mathcal{H}_2^{(\mathcal{T})}} := \text{tr}\left(\sum_{i=1}^{3} \int_0^\infty \cdots \int_0^\infty h_i(t_1,\dots,t_i)h_i^T(t_1,\dots,t_i)dt_1\cdots dt_i\right). \tag{4.10}$$

where $\text{tr}(\cdot)$ denotes the trace of the argument. Note that when $H = 0$ (in other words, when the underlying system is linear), only the first kernel $h_1(t_1)$ remains and accordingly (4.10) boils down to the classical $\mathcal{H}_2$-norm of an asymptotically stable linear system. We know that the $\mathcal{H}_2$ norm of the linear systems are computed using reachability and observability gramians. Similarly, the truncated $\mathcal{H}_2$-norm for the quadratic systems in (4.10) can be computed via the truncated reachability gramian P_T and observability gramian Q_T. When $E^{-1}A$ is asymptotically stable, theses gramians are unique and positive semi-definite solutions are

the following Lyaponov equations [23, 56]

$$AP_TE^T + EP_TA^T + H(P_l \otimes P_l)H^T + BB^T = 0 \tag{4.11}$$

$$A^TE^{-T}Q_T + Q_TE^{-1}A + \mathcal{H}_E^{(2)}(P_l \otimes Q_l)(\hat{\mathcal{H}}_E{}^{(2)})^T + C^TC = 0, \tag{4.12}$$

where $\mathcal{H}_E^{(2)}$ is the mode-2 matricization of $E^{-1}H$, and P_l and Q_l are the unique solutions to the Lyapunov equations

$$AP_lE^T + EP_lA^T + BB^T = 0 \tag{4.13}$$

$$A^TE^{-T}Q_l + Q_lE^{-1}A + C^TC = 0\ .$$

Then (4.10) can be written as [23]

$$\|\Sigma\|^2_{\mathcal{H}_2^{(T)}} = \operatorname{tr}(CP_TE^TC^T). \tag{4.14}$$

The goal is now to construct V and W such that the truncated $\mathcal{H}_2$ error norm $\|\Sigma-\Sigma_r\|_{\mathcal{H}_2}(\tau)$ is approximately minimized. This can be achieved by the numerically efficient model reduction algorithm Q-IRKA [23, 56]. Q-IRKA generates quadratic reduced models that approximately satisfy the optimality conditions resulting from the truncated $\mathcal{H}_2$ error norm $\|\Sigma-\Sigma_r\|_{\mathcal{H}_2}(\tau)$. A brief sketch of (Two-sided) Q-IRKA is shown in Algorithm 2. In some settings, to retain the positive definiteness and/or symmetry of the original realization in the reduced order matrices (4.8), one sided version of Q-IRKA is applied where W is set to V as given in Algorithm 3. The proposed approach to structure preserving model reduction of power networks as outlined in the next section will employ Q-IRKA as an intermediate step. Note that the reduced models from Q-IRKA is quadratic and does not preserve the original second-order structure (1.2) and thus it cannot be directly used in a structure preserving setting.

In the next section, we will prove that the model reduction bases V and W resulting from applying Q-IRKA to the quadratic form of power network have special subspace structure, which will then allow us to develop our structure-preserving model reduction algorithm.

4.2.2 Modifications to perform Q-IRKA for power network dynamics: zero initial conditions and asymptotic stability

Recall the state vector $q(t)$ in (3.2). Since $\sin(\delta(t))$ and $\cos(\delta(t))$ can not be simultaneously zero, the reformulation (3.3) as a quadratic model will always have a nonzero initial condition. Although some MOR approaches for linear dynamical systems can handle non-zero initial condition [14, 63], most system theoretic methods to model reduction assumes a zero initial condition. Hence, to have a zero initial condition, a natural approach is to consider a shifted state vector that allows the use of the reduction methods developed for zero initial condition [12]. Towards this goal, we define a shifted state vector $x = q - q_0$, such that in the new state x, we have $x(0) = 0$. Following the approach in [125], we write

$$\begin{aligned} q \otimes q = (x + q_0) \otimes (x + q_0) &= x \otimes x + (x \otimes q_0 + q_0 \otimes x) + q_0 \otimes q_0 \\ &= x \otimes x + ((I_{4n} \otimes q_0) + (q_0 \otimes I_{4n}))x + q_0 \otimes q_0. \end{aligned} \tag{4.15}$$

By substituting (4.15) in (3.3), we obtain

$$\begin{aligned} E\dot{x}(t) &= \tilde{A}x(t) + H(x(t) \otimes x(t)) + \tilde{B}\tilde{u}(t) \\ y(t) &= Cx(t), \quad x(0) = 0, \end{aligned} \tag{4.16}$$

Algorithm 2 Two-sided Q-IRKA

Input: The system matrices: E, A, H, B, C

Output: The reduced matrices: E_r, A_r, H_r, B_r, C_r, and the reduction bases: V, W

- Symmetrize the Hessian H, then transform it to a third-order $N \times N \times N$ tensor and determine its mode-2 matricization $\mathcal{H}^{(2)}$ as (4.6) (for details see, e.g., [77], [23])
- Make an initial guess for the reduced matrices E_r, A_r, H_r, B_r, C_r
- while (not converged)
  1. Perform the spectral decomposition of the pair E_r and A_r, i.e., $A_r R = E_r R \Lambda$ and define:
  $$\hat{H} = (E_r R)^{-1} H_r (R \otimes R), \ \hat{B} = (E_r R)^{-1} B_r, \ \hat{C} = C_r R$$
  2. Compute mode-2 matricization $\hat{\mathcal{H}}^{(2)}$
  3. Solve for V_1 and V_2:
  $$\begin{aligned} -EV_1\Lambda - AV_1 &= B\hat{B}^T \\ -EV_2\Lambda - AV_2 &= H(V_1 \otimes V_1)\hat{H}^T \end{aligned}$$
  4. Solve for W_1 and W_2:
  $$\begin{aligned} -E^T W_1\Lambda - A^T W_1 &= C^T\hat{C} \\ -E^T W_2\Lambda - A^T W_2 &= \mathcal{H}^{(2)}(V_1 \otimes W_1)(\hat{\mathcal{H}}^{(2)})^T \end{aligned}$$
  5. Compute V and W:
  $$V \coloneqq V_1 + V_2 \quad \text{and} \quad W \coloneqq W_1 + W_2$$
  6. Create an orthogonal basis for each V and W
  7. Determine the reduced matrices:
  $$\begin{aligned} E_r &= W^T E V, \quad A_r = W^T A V \\ H_r &= W^T H(V \otimes V), \quad B_r = W^T B, \quad C_r = CV \end{aligned}$$

Algorithm 3 One-sided Q-IRKA

Input: The system matrices: E, A, H, B, C

Output: The reduced matrices: E_r, A_r, H_r, B_r, C_r, and the reduction bases: V, W

- Apply Algorithm 2 by replacing Step 4 with

$$W_1 = V_1 \quad \text{and} \quad W_2 = V_2$$

where

$$\begin{aligned} \tilde{A} &= A + H\left((I_{4n} \otimes q_0) + (q_0 \otimes I_{4n})\right), \\ \tilde{B} &= \begin{bmatrix} B & Aq_0 + H\ (q_0 \otimes q_0) \end{bmatrix}, \quad \text{and} \quad \tilde{u} = [u \ \ 1]^T. \end{aligned} \tag{4.17}$$

Given the dynamical system (4.17) with zero initial conditions, one can apply a wide range of available methods, as mentioned at the beginning of Section 4.2, to construct a quadratic reduced model. However, as in the linear case, optimal-$\mathcal{H}_2$ model reduction for quadratic system requires asymptotic stability of $E^{-1}A$ [23]. Recall the quadratic dynamics (4.16) and $\tilde{A}$ in (4.17).

$$\tilde{A} = A + H\left((I_{4n} \otimes q_0) + (q_0 \otimes I_{4n})\right).$$

Also recall that the first n columns of A in (3.21) are zero. As we will see in the proof of Theorem 4.1, the first n columns of $H\left((I_{4n} \otimes q_0) + (q_0 \otimes I_{4n})\right)$ are zero as well. Thus, the first n columns of $\tilde{A}$ are zero. Hence, $\tilde{A}$ have zero eigenvalues and therefore $E^{-1}\tilde{A}$ is not asymptotically stable [125]. To overcome this stability issue in the power network setting, we replace $\tilde{A}$ in (4.16) by the shifted matrix

$$\tilde{A}_\eta = \tilde{A} - \eta E, \tag{4.18}$$

where $\eta > 0$ is real and small, so that all the eigenvalues of the pair $\tilde{A}_\eta$ and E have negative real part. Therefore, as the final modification, we replace (4.16) by

$$\begin{aligned} E\dot{x}(t) &= \tilde{A}_\eta x(t) + H(x(t) \otimes x(t)) + \tilde{B}\tilde{u}(t) \\ y(t) &= Cx(t), \quad x(0) = 0. \end{aligned} \tag{4.19}$$

The quadratic dynamical system in (4.19) has now asymptotically stable linear part and have zero initial conditions. Therefore, it has the structure to perform $\mathcal{H}_2$ based model reduction using Q-IRKA. However, we re-emphasize that the reduced-model obtained from applying Q-IRKA to (4.19) is not structure-preserving; it does not inherit the second-order structure. Therefore, we employ Q-IRKA only to obtain potential reduction subspace information. In the next section where we present the proposed framework, we describe how we benefit from and how we use the subspaces resulting from Q-IRKA to obtain a reduced second-order model in (4.1). We also show that Q-IRKA subspaces applied to (4.19) has special properties due to the underlying power network structure.

4.3 The proposed approach for model reduction of power networks

Recall that our aim is to perform structure-preserving model reduction of the second-order dynamics (1.2) using the reduction bases $\mathcal{V}, \mathcal{W} \in \mathbb{R}^{n \times r}$ to obtain the reduced second-order model (4.1)-(4.3). In order to keep the symmetry of the underlying dynamics, we will perform a Galerkin projection by choosing $\mathcal{W} = \mathcal{V}$. Converting the nonlinear second-order dynamics to the quadratic form has allowed us to use (optimal) systems-theoretic techniques to apply, such as Q-IRKA. However, as pointed out in the previous section, we cannot simply use the output of Q-IRKA as the reduced model since it does not preserve the second-order

structure. However, we would like to employ the model reduction bases V and W resulting from Q-IRKA in constructing $\mathcal{V}$ since they are high-fidelity model reduction bases. In this section, by analyzing and exploiting the structure of the underlying power network dynamics, we show how to efficiently construct $\mathcal{V}$ using V and W from Q-IRKA.

4.3.1 Structures arising in Q-IRKA in the power network setting

Let $V \in \mathbb{R}^{4n\times r_q}$ and $W \in \mathbb{R}^{4n\times r_q}$ be the model reduction bases obtained from applying Q-IRKA to (4.19). Let $V_T \in \mathbb{R}^{n\times r_q}$ and $W_T \in \mathbb{R}^{n\times r_q}$ denote leading n rows of V and W. Recall that due to (3.2), the leading n rows of $q(t)$, the state of quadratic dynamic (4.19), corresponds to the original (second-order) state $\delta(t)$. Therefore, one can consider choosing $\mathcal{V} = V_T$ and $\mathcal{W} = W_T$ in (4.3). However, in addition to this being a Petrov-Galerkin projection and thus not preserving the symmetry, we will show in the next result that W_T has a particular subspace structure that one can exploit in constructing the structured reduced model.

Theorem 4.1. *Let V and W be obtained by applying Q-IRKA as in Algorithm 2 to the quadratised power network dynamics (4.19) resulting with a reduced quadratic system with asymptotically stable linear part. Let $W_T \in \mathbb{R}^{n\times r_q}$ denote the first n rows of W. Then*

$$\mathit{Range}(W_T) \subseteq \mathit{Range}(\mathcal{C}^T). \tag{4.20}$$

In addition, when $p \leq r_q$, we have $\mathit{Range}(\mathcal{C}^T) = \mathit{Range}(W_T)$.

Before we prove Theorem 4.1, we establish a symmetry property of H that will be used in the proof.

Lemma 4.2. *Let H in (4.19) be the mode-1 matricization of the third order tensor $\mathcal{H} \in$*

$\mathbb{R}^{4n\times 4n\times 4n}$, *Then, the tensor* $\mathcal{H}$ *is symmetric, i.e.,*

$$H(\nu\otimes\rho)=H(\rho\otimes\nu)\qquad \textit{for every}\quad \nu,\rho\in\mathbb{R}^{4n\times 1}. \tag{4.21}$$

Proof. Recall (3.28) and (3.29), and rewrite $H(\rho\otimes\nu)$ as

$$H(\rho\otimes\nu)=\tilde{H}P^T(\rho\otimes\nu)=\tilde{H}(\rho\,\widetilde{\otimes}\,\nu),$$

where

$$\rho=\begin{bmatrix}\rho_1\\ \rho_2\\ \rho_3\\ \rho_4\end{bmatrix}\quad\text{and}\quad \nu=\begin{bmatrix}\nu_1\\ \nu_2\\ \nu_3\\ \nu_4\end{bmatrix}\quad;\ \rho_i,\nu_i\in\mathbb{R}^{n\times 1}.$$

Using (3.22) and (3.24) we obtain

$$H(\rho\otimes\nu)=\left[\tilde{H}_2(\rho_2\,\widetilde{\otimes}\,\nu)+\tilde{H}_3(\rho_3\,\widetilde{\otimes}\,\nu)+\tilde{H}_4(\rho_4\,\widetilde{\otimes}\,\nu)\right]. \tag{4.22}$$

Let

$$\mathcal{I}_1=\begin{bmatrix}I_n & 0_{n\times n} & 0_{n\times n} & 0_{n\times n}\end{bmatrix}\in\mathbb{R}^{n\times 4n}. \tag{4.23}$$

Multiply (4.22) from the right with (4.23) to pick the first n rows:

$$\mathcal{I}_1H(\rho\otimes\nu)=\begin{bmatrix}I_n & 0_{n\times n} & 0_{n\times n} & 0_{n\times n}\end{bmatrix}\left[\tilde{H}_2(\rho_2\,\widetilde{\otimes}\,\nu)+\tilde{H}_3(\rho_3\,\widetilde{\otimes}\,\nu)+\tilde{H}_4(\rho_4\,\widetilde{\otimes}\,\nu)\right]. \tag{4.24}$$

Since from (3.26) and (3.27) the first n rows of $\tilde{H}_2, \tilde{H}_3$ and $\tilde{H}_4$ are zero, we have

$$\mathcal{I}_1 H(\rho \otimes \nu) = 0_{n\times 1}. \tag{4.25}$$

Similarly,

$$H(\nu \otimes \rho) = \left[\tilde{H}_2(\nu_2 \,\widetilde{\otimes}\, \rho) + \tilde{H}_3(\nu_3 \,\widetilde{\otimes}\, \rho) + \tilde{H}_4(\nu_4 \,\widetilde{\otimes}\, \rho)\right] \tag{4.26}$$

and

$$\mathcal{I}_1 H(\nu \otimes \rho) = \begin{bmatrix} I_n & 0_{n\times n} & 0_{n\times n} & 0_{n\times n} \end{bmatrix} \left[\tilde{H}_2(\nu_2 \,\widetilde{\otimes}\, \rho) + \tilde{H}_3(\nu_3 \,\widetilde{\otimes}\, \rho) + \tilde{H}_4(\nu_4 \,\widetilde{\otimes}\, \rho)\right] = 0_{n\times 1}. \tag{4.27}$$

Therefore, we have

$$\mathcal{I}_1 H(\rho \otimes \nu) = \mathcal{I}_1 H(\nu \otimes \rho). \tag{4.28}$$

Now let

$$\mathcal{I}_2 = \begin{bmatrix} 0_{n\times n} & I_n & 0_{n\times n} & 0_{n\times n} \end{bmatrix} \in \mathbb{R}^{n\times 4n} \tag{4.29}$$

and multiply (4.22) from the right with (4.29):

$$\begin{aligned} \mathcal{I}_2 H(\rho \otimes \nu) &= \begin{bmatrix} 0_{n\times n} & I_n & 0_{n\times n} & 0_{n\times n} \end{bmatrix} \left[\tilde{H}_2(\rho_2 \,\widetilde{\otimes}\, \nu) + \tilde{H}_3(\rho_3 \,\widetilde{\otimes}\, \nu) + \tilde{H}_4(\rho_4 \,\widetilde{\otimes}\, \nu)\right] \\ &= \begin{bmatrix} 0_{n\times n^2} & 0_{n\times n^2} & \Psi & \frac{Z}{2} \end{bmatrix} (\rho_3 \,\widetilde{\otimes}\, \nu) + \begin{bmatrix} 0_{n\times n^2} & 0_{n\times n^2} & -\frac{Z}{2} & \Psi \end{bmatrix} (\rho_4 \,\widetilde{\otimes}\, \nu) \\ &= \Psi(\rho_3 \otimes \nu_3) + \frac{Z}{2}(\rho_3 \otimes \nu_4) - \frac{Z}{2}(\rho_4 \otimes \nu_3) + \Psi(\rho_4 \otimes \nu_4). \end{aligned} \tag{4.30}$$

Let $S \in \mathbb{R}^{n^2 \times n^2}$ be the permutation matrix so that

$$\rho \otimes \nu = S(\nu \otimes \rho), \tag{4.31}$$

where S is partitioned as

$$S = \begin{bmatrix} S_1 & S_2 & \dots S_n \end{bmatrix}. \tag{4.32}$$

Each $S_j \in \mathbb{R}^{n^2 \times n}$, for $j = 1, \dots, n$, is defined as

$$S_j = \begin{bmatrix} e_j & e_{n+j} & e_{2n+j} & \dots & e_{n(n-1)+j} \end{bmatrix}, \tag{4.33}$$

where $e_j \in \mathbb{R}^{n^2 \times 1}$ denotes the jth unit vector, i.e., the jth column of the identity matrix $I_{n^2} \in \mathbb{R}^{n^2 \times n^2}$. Insert (4.31) into (4.30) to obtain

$$\mathcal{I}_2 H(\rho \otimes \nu) = \Psi S(\nu_3 \otimes \rho_3) + \frac{Z}{2} S(\nu_3 \otimes \rho_4) - \frac{Z}{2} S(\nu_4 \otimes \rho_3) + \Psi S(\nu_4 \otimes \rho_4). \tag{4.34}$$

Use (4.32) to obtain

$$\Psi S = \begin{bmatrix} \Psi S_1 & \Psi S_2 & \dots \Psi S_n \end{bmatrix}, \tag{4.35}$$

where, for $j = 1, \dots, n$, we have

$$\begin{aligned} \Psi S_j &= \Psi \begin{bmatrix} e_j & e_{n+j} & e_{2n+j} & \dots & e_{n(n-1)+j} \end{bmatrix} \\ &= \begin{bmatrix} \Psi_1(:, j) & \Psi_2(:, j) & \Psi_3(:, j) & \dots & \Psi_n(:, j) \end{bmatrix}, \end{aligned} \tag{4.36}$$

and where $\Psi_1(:, j)$ corresponds to the jth column of Ψ_1. Using (3.16), for the j column of

Ψ_k, we write

$$\Psi_k(:,j) = \begin{cases} -\frac{1}{2}K_{kj}\sin(\gamma_{kj}), & i = j \neq k \\ -\frac{1}{2}K_{kj}\sin(\gamma_{kj}), & i = k,\ j \neq k\,. \\ 0 & \text{otherwise} \end{cases} \tag{4.37}$$

Since for the coupling between the oscillators i and j we have $K_{ij} = K_{ji}$ and $\gamma_{ij} = \gamma_{ji}$, it follows from (4.37) that

$$\Psi_k(:,j) = \Psi_j(:,k)\ . \tag{4.38}$$

Then using (4.38) to rewrite (4.36) yields

$$\begin{aligned} \Psi S_j &= \begin{bmatrix} \Psi_1(:,j) & \Psi_2(:,j) & \Psi_3(:,j) & \dots & \Psi_n(:,j) \end{bmatrix} \\ &= \begin{bmatrix} \Psi_j(:,1) & \Psi_j(:,2) & \Psi_j(:,3) & \dots & \Psi_j(:,n) \end{bmatrix} = \Psi_j. \end{aligned} \tag{4.39}$$

Inserting (4.39) into (4.35) for $j = 1, \dots, n$, then, yields

$$\Psi S = \begin{bmatrix} \Psi S_1 & \Psi S_2 & \dots \Psi S_n \end{bmatrix} = \begin{bmatrix} \Psi_1 & \Psi_2 & \dots & \Psi_n \end{bmatrix} = \Psi. \tag{4.40}$$

Similarly,

$$ZS = \begin{bmatrix} ZS_1 & ZS_2 & \dots ZS_n \end{bmatrix}, \tag{4.41}$$

where, for $j = 1, \dots, n$, we have

$$ZS_j = \begin{bmatrix} Z_1(:,j) & Z_2(:,j) & Z_3(:,j) & \dots & Z_n(:,j) \end{bmatrix}. \tag{4.42}$$

According to (3.15) for the j column of Z_k, we have

$$Z_k(:,j) = \begin{cases} K_{kj}\cos(\gamma_{kj}), & i = j \neq k \\ -K_{kj}\cos(\gamma_{kj}), & i = k,\ j \neq k \\ 0 & \text{otherwise} \end{cases}. \tag{4.43}$$

Knowing the facts that $K_{ij} = K_{ji}$ and $\gamma_{ij} = \gamma_{ji}$, lead to

$$Z_k(:,j) = -Z_j(:,k)\ . \tag{4.44}$$

Then using (4.44) to rewrite (4.42) yields

$$\begin{aligned} ZS_j &= \begin{bmatrix} Z_1(:,j) & Z_2(:,j) & Z_3(:,j) & \dots & Z_n(:,j) \end{bmatrix} \\ &= \begin{bmatrix} -Z_j(:,1) & -Z_j(:,2) & -Z_j(:,3) & \dots & -Z_j(:,n) \end{bmatrix} = -Z_j. \end{aligned} \tag{4.45}$$

Inserting (4.45) into (4.41) for $j = 1, \dots, n$, then, yields

$$ZS = \begin{bmatrix} ZS_1 & ZS_2 & \dots ZS_n \end{bmatrix} = \begin{bmatrix} -Z_1 & -Z_2 & \dots & -Z_n \end{bmatrix} = -Z. \tag{4.46}$$

Using (4.40) and (4.46), we rewrite (4.30) as

$$\begin{aligned} \mathcal{I}_2 H(\rho \otimes \nu) =& \Psi(\rho_3 \otimes \nu_3) + \frac{Z}{2}(\rho_3 \otimes \nu_4) - \frac{Z}{2}(\rho_4 \otimes \nu_3) + \Psi(\rho_4 \otimes \nu_4) \\ =& \Psi S(\rho_3 \otimes \nu_3) - \frac{ZS}{2}(\rho_3 \otimes \nu_4) + \frac{ZS}{2}(\rho_4 \otimes \nu_3) + \Psi S(\rho_4 \otimes \nu_4) \\ =& \Psi(\nu_3 \otimes \rho_3) - \frac{Z}{2}(\nu_4 \otimes \rho_3) + \frac{Z}{2}(\nu_3 \otimes \rho_4) + \Psi(\nu_4 \otimes \rho_4) = \mathcal{I}_2 H(\nu \otimes \rho). \end{aligned} \tag{4.47}$$

Recall the definition of Φ in (3.11). Let

$$\mathcal{I}_3 = \begin{bmatrix} 0_{n\times n} & 0_{n\times n} & I_n & 0_{n\times n} \end{bmatrix} \in \mathbb{R}^{n\times 4n} \tag{4.48}$$

and follow a similar strategy by multiplying (4.48) from the left with (4.22) to obtain

$$\begin{aligned} \mathcal{I}_3 H(\rho\otimes\nu) &= \begin{bmatrix} 0_{n\times n^2} & 0_{n\times n^2} & 0_{n\times n^2} & \frac{\Phi}{2} \end{bmatrix} (\rho_2 \mathbin{\widetilde{\otimes}} \nu) + \begin{bmatrix} 0_{n\times n^2} & \frac{\Phi}{2} & 0_{n\times n^2} & 0_{n\times n^2} \end{bmatrix} (\rho_4 \mathbin{\widetilde{\otimes}} \nu) \\ &= \frac{1}{2}\begin{bmatrix} \Phi(\rho_2\otimes\nu_4) \end{bmatrix} + \frac{1}{2}\begin{bmatrix} \Phi(\rho_4\otimes\nu_2) \end{bmatrix} \\ &= \frac{1}{2}\begin{bmatrix} (\rho_2)_1\Phi_1\nu_4 + \cdots + (\rho_2)_n\Phi_n\nu_4 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} (\rho_4)_1\Phi_1\nu_2 + \cdots + (\rho_4)_n\Phi_n\nu_2 \end{bmatrix} \\ &= \frac{1}{2}\begin{bmatrix} (\rho_2)_1(\nu_4)_1 e_1 + \cdots + (\rho_2)_n(\nu_4)_n e_n \end{bmatrix} + \frac{1}{2}\begin{bmatrix} (\rho_4)_1(\nu_2)_1 e_1 + \cdots + (\rho_4)_n(\nu_2)_n e_n \end{bmatrix}, \end{aligned} \tag{4.49}$$

where $(\rho_2)_j \in \mathbb{R}$ refers to the jth entry of the vector ρ_2, and the unit vector $e_i \in \mathbb{R}^{n\times 1}$ corresponds to the ith column of the identity matrix $I_n \in \mathbb{R}^{n\times n}$. By rearranging (4.49) we obtain

$$\begin{aligned} \mathcal{I}_3 H(\rho\otimes\nu) &= \frac{1}{2}\begin{bmatrix} (\nu_4)_1(\rho_2)_1 e_1 + \cdots + (\nu_4)_n(\rho_2)_n e_n \end{bmatrix} + \frac{1}{2}\begin{bmatrix} (\nu_2)_1(\rho_4)_1 e_1 + \cdots + (\nu_2)_n(\rho_4)_n e_n \end{bmatrix} \\ &= \frac{1}{2}\begin{bmatrix} \Phi(\nu_4\otimes\rho_2) \end{bmatrix} + \frac{1}{2}\begin{bmatrix} \Phi(\nu_2\otimes\rho_4) \end{bmatrix} = \mathcal{I}_3 H(\nu\otimes\rho). \end{aligned} \tag{4.50}$$

Following similar steps and by letting $\mathcal{I}_4 = \begin{bmatrix} 0_{n\times n} & 0_{n\times n} & 0_{n\times n} & I_n \end{bmatrix} \in \mathbb{R}^{n\times 4n}$, we obtain

$$\begin{aligned}
\mathcal{I}_4 H(\rho\otimes\nu) &= -\frac{1}{2}\left[\Phi(\rho_2\otimes\nu_3)\right] - \frac{1}{2}\left[\Phi(\rho_3\otimes\nu_2)\right] \\
&= -\frac{1}{2}\left[(\rho_2)_1\Phi_1\nu_3 + \cdots + (\rho_2)_n\Phi_n\nu_3\right] - \frac{1}{2}\left[(\rho_3)_1\Phi_1\nu_2 + \cdots + (\rho_3)_n\Phi_n\nu_2\right] \\
&= -\frac{1}{2}\left[(\rho_2)_1(\nu_3)_1 e_1 + \cdots + (\rho_2)_n(\nu_3)_n e_n\right] - \frac{1}{2}\left[(\rho_3)_1(\nu_2)_1 e_1 + \cdots + (\rho_3)_n(\nu_2)_n e_n\right] \\
&= -\frac{1}{2}\left[(\nu_3)_1(\rho_2)_1 e_1 + \cdots + (\nu_3)_n(\rho_2)_n e_n\right] - \frac{1}{2}\left[(\nu_2)_1(\rho_3)_1 e_1 + \cdots + (\nu_2)_n(\rho_3)_n e_n\right] \\
&= -\frac{1}{2}\left[\Phi(\nu_3\otimes\rho_2)\right] - \frac{1}{2}\left[\Phi(\nu_2\otimes\rho_3)\right] = \mathcal{I}_4 H(\nu\otimes\rho),
\end{aligned} \tag{4.51}$$

Combining (4.28), (4.47), (4.50), and (4.51), we conclude (4.21), i.e., $\mathcal{H}$ is symmetric. □

Now we are ready to prove Theorem 4.1.

Proof. (*of Theorem 4.1*) We start by analyzing the first Sylvester equation involving W_1 in Step 7 of Algorithm 2:

$$-E^T W_1 \Lambda - \tilde{A}_\eta^T W_1 = C^T \hat{C}, \tag{4.52}$$

where $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_r) \in \mathbb{R}^{r_q\times r_q}$ contains the eigenvalues of $E_r^{-1}A_r$, $\hat{C} \in \mathbb{R}^{p\times r_q}$ and

$$\tilde{A}_\eta = \tilde{A} - \eta E = A + H\left((I_{4n}\otimes q_0) + (q_0\otimes I_{4n})\right) - \eta E. \tag{4.53}$$

Recall $H \in \mathbb{R}^{4n\times(4n)^2}$ in (3.29):

$$H = \tilde{H}P^T = \begin{bmatrix} \tilde{H}_1 P_q^T & \tilde{H}_2 P_q^T & \tilde{H}_3 P_q^T & \tilde{H}_4 P_q^T \end{bmatrix} = \begin{bmatrix} 0_{4n\times 4n^2} & \tilde{H}_2 P_q^T & \tilde{H}_3 P_q^T & \tilde{H}_4 P_q^T \end{bmatrix}. \tag{4.54}$$

Multiply $H\left((I_{4n} \otimes q_0) + (q_0 \otimes I_{4n})\right)$ by the first unit vector $e_1 \in \mathbb{R}^{4n\times 1}$ from the right to obtain

$$h_1 = H\left((I_{4n} \otimes q_0) + (q_0 \otimes I_{4n})\right) e_1 = H\left((e_1 \otimes q_0) + (q_0 \otimes e_1)\right), \tag{4.55}$$

where $h_1 \in \mathbb{R}^{4n\times 1}$ denotes the first column of $H\left((I_{4n} \otimes q_0) + (q_0 \otimes I_{4n})\right)$. Now, we apply Lemma 4.2 to (4.55) and use (4.54) to obtain

$$h_1 = H\left((e_1 \otimes q_0) + (e_1 \otimes q_0)\right) = 2H(e_1 \otimes q_0) = 2H \begin{bmatrix} q_0 \\ 0_{4n(4n-1)} \end{bmatrix} = 0_{4n\times 1}. \tag{4.56}$$

Similarly, for the second column of (4.54), we have

$$h_2 = 2H(e_2 \otimes q_0) = 2H \begin{bmatrix} 0_{4n} \\ q_0 \\ 0_{4n(4n-2)} \end{bmatrix} = 0_{4n\times 1}\ ; \tag{4.57}$$

and for the kth column of (4.54), we obtain

$$h_k = 2H(e_k \otimes q_0) = 2H \begin{bmatrix} 0_{4n(n-1)} \\ q_0 \\ 0_{12n^2} \end{bmatrix} = 0_{4n\times 1} \quad ; k = 2, 3, \ldots, n. \tag{4.58}$$

Therefore, (4.56) and (4.58) reveal that the first n columns of $H\left((I_{4n} \otimes q_0) + (q_0 \otimes I_{4n})\right)$ are

zero. Using this fact, and (3.21), we obtain the first n columns of (4.53) as

$$\tilde{A}_\eta(:, 1:n) = A(:, 1:n) - \eta E(:, 1:n) = \begin{bmatrix} -\eta I_n \\ 0_{n\times n} \\ 0_{n\times n} \\ 0_{n\times n} \end{bmatrix} \in \mathbb{R}^{4n\times n}, \tag{4.59}$$

where $Y(:, 1:n)$ refers to the first n columns of the matrix Y. Now let $W_1 \in \mathbb{R}^{4n\times r_q}$ be decomposed as

$$W_1 = \begin{bmatrix} W_{11} \\ W_{12} \\ W_{13} \\ W_{14} \end{bmatrix}, \quad \text{where} \quad W_{1i} \in \mathbb{R}^{n\times r_q} \quad ; i = 1, 2, 3, 4 \tag{4.60}$$

Recall (3.4) and (3.21),

$$y(t) = Cq(t) \quad \text{with} \quad C = \begin{bmatrix} \mathcal{C} & 0_{p\times n} & 0_{p\times n} & 0_{p\times n} \end{bmatrix} \in \mathbb{R}^{p\times 4n}. \tag{3.4}$$

$$E = \text{blkdiag}(I_n, \mathcal{M}, I_n, I_n), \quad B = \begin{bmatrix} 0_{n\times 1} \\ \mathcal{B} \\ 0_{n\times 1} \\ 0_{n\times 1} \end{bmatrix} \quad \text{and} \quad A = \begin{bmatrix} 0_{n\times n} & I_n & 0_{n\times n} & 0_{n\times n} \\ 0_{n\times n} & -\mathcal{D} & 0_{n\times n} & 0_{n\times n} \\ 0_{n\times n} & 0_{n\times n} & 0_{n\times n} & 0_{n\times n} \\ 0_{n\times n} & 0_{n\times n} & 0_{n\times n} & 0_{n\times n} \end{bmatrix}, \tag{3.21}$$

and use (4.59) and (4.60) to obtain the first n rows of (4.52)

$$\begin{bmatrix} -I_n & 0_{n\times n} & 0_{n\times n} & 0_{n\times n} \end{bmatrix} \begin{bmatrix} W_{11} \\ W_{12} \\ W_{13} \\ W_{14} \end{bmatrix} \Lambda + \begin{bmatrix} -\eta I_n & 0_{n\times n} & 0_{n\times n} & 0_{n\times n} \end{bmatrix} \begin{bmatrix} W_{11} \\ W_{12} \\ W_{13} \\ W_{14} \end{bmatrix} = \mathcal{C}^T \hat{C}, \tag{4.61}$$

which yields,

$$W_{11}(\eta I_{r_q} - \Lambda) = \mathcal{C}^T \hat{C}. \tag{4.62}$$

Recall that η is a positive real number and the reduced order poles are assumed in the left-half plane. Therefore, the matrix $\eta I_{r_q} - \Lambda$ is invertible and

$$W_{11} = \mathcal{C}^T \hat{C}(\eta I_{r_q} - \Lambda)^{-1}. \tag{4.63}$$

Similarly, let

$$W_2 = \begin{bmatrix} W_{21} \\ W_{22} \\ W_{23} \\ W_{24} \end{bmatrix}, \quad \text{where} \quad W_{2i} \in \mathbb{R}^{n\times r_q} \quad ; i = 1, 2, 3, 4 \tag{4.64}$$

and solve for the first $n \times r_q$ block of $W_2 \in \mathbb{R}^{4n\times r_q}$ denoted by $W_{21} \in \mathbb{R}^{n\times r_q}$ in the second Sylvester equation in Step 7 of Algorithm 2:

$$-E^T W_2 \Lambda - \tilde{A}_\eta^T W_2 = \mathcal{H}^{(2)}(V_1 \otimes W_1)(\hat{\mathcal{H}}^{(2)})^T. \tag{4.65}$$

Next, we will prove that the first n rows of $\mathcal{H}^{(2)}$ are zero. Towards this goal, recall (4.5) and

let H be partitioned as

$$H = \mathcal{H}^{(1)} = \begin{bmatrix} \mathcal{H}_1 & \mathcal{H}_2 & \dots & \mathcal{H}_{4n} \end{bmatrix}, \tag{4.66}$$

where $\mathcal{H}_i \in \mathbb{R}^{4n\times 4n}$ for $i = 1, 2, \dots, 4n$. From (4.54), we know that the leading $4n^2$ columns are zero. Therefore, we write

$$H = \mathcal{H}^{(1)} = \begin{bmatrix} 0_{4n\times 4n^2} & \mathcal{H}_{n+1} & \dots & \mathcal{H}_{4n} \end{bmatrix}, \tag{4.67}$$

i.e., $\mathcal{H}_i = 0_{4n\times 4n}$ for $i = 1, 2, \dots, n$. Recall that $\mathcal{H}$ is symmetric due to Lemma 4.2. This means that

$$\mathcal{H}^{(2)} = \mathcal{H}^{(3)} \in \mathbb{R}^{4n\times (4n)^2}, \tag{4.68}$$

where

$$\mathcal{H}^{(3)} = \begin{bmatrix} \text{vec}(\mathcal{H}_1) & \text{vec}(\mathcal{H}_2) & \dots & \text{vec}(\mathcal{H}_{4n}) \end{bmatrix}^T \tag{4.69}$$

denotes the mode-3 matricization, and vec$(\cdot)$ stands for vectorization of a matrix. Since $\mathcal{H}_i = 0_{4n\times 4n}$ for $i = 1, 2, \dots, n$, we obtain

$$\mathcal{H}^{(3)} = \begin{bmatrix} 0_{16n^2\times n} & \text{vec}(\mathcal{H}_{n+1}) \dots & \text{vec}(\mathcal{H}_{4n}) \end{bmatrix}^T. \tag{4.70}$$

Due to (4.68), the first n rows of $\mathcal{H}^{(2)}$ are the first n columns of $\left(\mathcal{H}^{(3)}\right)^T$. Then, (4.70) implies that $\mathcal{H}^{(2)}(1:n,:) = 0_{n\times(4n)^2}$. Using this fact in (4.65) yields

$$W_{21}(\eta I_{r_q} - \Lambda) = 0_{n\times r_q}. \tag{4.71}$$

Since $\eta I_{r_q} - \Lambda$ is invertible, we obtain

$$W_{21} = 0_{n \times r_q} \ . \tag{4.72}$$

Recall W in Step 8 of Algorithm (2) defined as $W = W_1 + W_2$. Then, for the first n rows of W, i.e., for W_T, we obtain

$$W_T = W_{11} + W_{21} = W_{11} = \mathcal{C}^T \hat{C} (\eta I_{r_q} - \Lambda)^{-1} \in \mathbb{R}^{n \times r_q}. \tag{4.73}$$

Hence, Range$(W_T) \subseteq$ Range$(\mathcal{C}^T)$, which proves (4.20). In addition, when $p \leq r_q$, if $\varkappa \in$ Range$(\mathcal{C}^T)$, there exist a vector ζ such that

$$\varkappa = \mathcal{C}^T \zeta.$$

By using the right-inverse of $\hat{C}$, namely $\hat{C}^+$, where $\hat{C}\hat{C}^+ = I_p$, we can write

$$\varkappa = \mathcal{C}^T \zeta = \mathcal{C}^T \hat{C} \hat{C}^+ \zeta = \mathcal{C}^T \hat{C} \tilde{\zeta},$$

where $\tilde{\zeta} = \hat{C}^+ \zeta$. Using the fact that $(\eta I_{r_q} - \Lambda)$ is invertible, we can write

$$\begin{aligned} \varkappa &= \mathcal{C}^T \hat{C} (\eta I_{r_q} - \Lambda)^{-1} (\eta I_{r_q} - \Lambda) \tilde{\zeta} \\ &= W_T (\eta I_{r_q} - \Lambda) \tilde{\zeta} \end{aligned}$$

that yields Range$(\mathcal{C}^T) \subseteq$ Range(W_T), completing the proof of the theorem.

□

We emphasize that the structure of the range of W_T due to Q-IRKA as shown in Theorem

4.1 is specific to the underlying power network dynamics and its quadratic form. We are not aware of any other models that yield such a form. This result will be crucial in forming the proposed algorithm next.

4.3.2 Proposed structure-preserving model reduction algorithm

Theorem 4.1 states that whenever $p \leq r_q$ (which will be the common situation), W_T is rank-deficient. Therefore, even though this means that W_T is not suitable to be used as a model reduction space as is, we will use it to our advantage.

As noted earlier, we will perform Galerkin projection. In other words, in (4.3) we will use $\mathcal{W} = \mathcal{V}$. One option is to simply ignore W_T and set $\mathcal{V} = V_T$. However, this is not preferred in an input-output based systems-theoretic model reduction setting since the information in W_T related to the output is fully ignored. Fortunately, due to the subspace result in Theorem 4.1, $(\text{span}(W_T) \subseteq \text{span}(\mathcal{C}^T))$, we can include both the input-to-state subspace information (V_T) and the state-to-output subspace information (W_T) in one subspace $\mathcal{V}$ by choosing in (4.3) as

$$\mathcal{V} = \text{orth}\left(\begin{bmatrix} V_T & \mathcal{C}^T \end{bmatrix}\right) \in \mathbb{R}^{n \times r}, \tag{4.74}$$

where "orth" refers to forming an orthonormal basis so that $\mathcal{V}^T\mathcal{V} = I_r$. Then, we choose $\mathcal{W} = \mathcal{V}$ and perform a Galerkin projection to compute the reduced second-order structure-preserving model (4.1)-(4.3). This means that despite performing a one-sided Galerkin, by exploiting Theorem 4.1 and using the reduction basis $\mathcal{V}$ in (4.74), we are able to also incorporate the output information in the reduction framework, as the standard in systems theoretic model reduction. We note that for the column dimension of $\mathcal{V}$ in (4.74), i.e., r, we have $r = \min(r_q + p, 2r_q)$.

Now, we finally summarize our proposed approach, an $\mathcal{H}_2$-based Structure-preserving MOR method for Power Network Dynamics (StrH2), in Algorithm 4. We start by converting the original dynamics to the equivalent quadratic form in Step 1. Then, in Step 2, we *either* apply two-sided Q-IRKA (Algorithm 2) *or* one-sided Q-IRKA (Algorithm 3) to the resulting quadratic-system to construct V. The resulting versions of the proposed method will be labeled as StrH2-A and StrH2-B, respectively. Step 3 constructs the final model reduction basis using the analysis of Theorem 4.1 and Step 4 constructs the structured reduced-model.

We note that in (4.74), the W-subspace information is not needed since it is already contained in W. However, StrH2-A version of Algorithm 4, in Step 2, still uses two-sided Q-IRKA, which computes W during the iteration unlike in one-sided Q-IRKA where the W-subspace is never computed. One might think that since the W-subspace information is replaced by $\mathcal{C}^T$ in the end, the two options StrH2-A and StrH2-B should produce equivalent result. This is not the case at all since the V-subspace resulting from one-sided and two-sided Q-IRKA implementations are completely different; the former never used the output information. We will see in next section that this distinction does indeed impact the final reduced model.

4.4 Numerical experiments

We test the proposed approach on two models: (i) the SM model of the 39-bus New England test system and (ii) IEEE 118 bus system, included in the MATPOWER software toolbox [131],[133]. We focus on a single-output system ($p = 1$) by choosing the arithmetic mean of all phase angles as the output as in [106, 109]. Models are generated by MATPOWER and MATLAB toolbox `pg-sync-models` [90] that provide all the necessary parameters to form (1.2) including J_i, D_i, ω_R, K_{ij}, γ_{ij} and B_i. To illustrate the effectiveness of our approach, we compare it with two other approaches, namely a structure-preserving formulation of balanced

Algorithm 4 $\mathcal{H}_2$-based Structure-preserving MOR for Power Network Dynamics (StrH2)

Input: Second-order model (1.2)-(1.3)

$$\begin{aligned} \mathcal{M}\ddot{\delta}(t) + \mathcal{D}\dot{\delta}(t) + f(\delta) &= \mathcal{B}u(t) \\ y(t) &= \mathcal{C}\delta(t) \end{aligned}$$

Output: Reduced second-order model (4.1)-(4.2)

$$\begin{aligned} \mathcal{M}_r\ddot{\delta}_r(t) + \mathcal{D}_r\dot{\delta}_r(t) + f_r(\delta_r) &= \mathcal{B}_r u(t) \\ y_r(t) &= \mathcal{C}_r\delta_r(t) \end{aligned}$$

- Transform the second-order dynamic (1.2) into the quadratic dynamics (4.19) with asymptotically stable linear part:

$$\begin{aligned} E\dot{x}(t) &= \tilde{A}_\eta x(t) + H(x(t) \otimes x(t)) + \tilde{B}\tilde{u}(t) \\ y(t) &= Cx(t), \quad x(0) = 0. \end{aligned} \tag{4.19}$$

- For the quadratic dynamics (4.19), apply

  Option A: Algorithm 2

  or to construct V.

  Option B: Algorithm 3

- Form the final model reduction basis $\mathcal{V}$ using the leading n rows of V and the output matrix $\mathcal{C}$ as in (4.74):

$$\mathcal{V} = \operatorname{orth}\left(\begin{bmatrix} V_T & \mathcal{C}^T \end{bmatrix}\right)$$

- Compute the reduced matrices as in (4.3):

$$\begin{aligned} &\mathcal{M}_r = \mathcal{V}^T\mathcal{M}\mathcal{V}, \ \mathcal{D}_r = \mathcal{V}^T\mathcal{D}\mathcal{V}, \ f_r(\delta_r) = \mathcal{V}^T f(\mathcal{V}\delta_r), \\ &\mathcal{B}_r = \mathcal{V}^T\mathcal{B}, \text{ and } \mathcal{C}_r = \mathcal{C}\mathcal{V}. \end{aligned}$$

truncation (Str-QBT) for power systems and POD. Before we go to numerical examples, we will briefly explain these methods in the following sections.

Also, to test the accuracy of the reduced models, we will measure the time-domain error between the true output $y(t)$ and the reduced-model outputs $y_r(t)$ over a time interval $t \in [0, T]$. Towards this goal, define the $\mathcal{L}_\infty(T)$ norm of $y(t)$ as

$$\|y\|_{\mathcal{L}_\infty(T)} = \max_{t\in[0,T]} \mid y(t) \mid . \tag{4.75}$$

The corresponding *relative* $\mathcal{L}_\infty$ output error is defined as

$$\|e\|_{\mathcal{L}_\infty(T)} = \frac{\|y - y_r\|_{\mathcal{L}_\infty(T)}}{\|y\|_{\mathcal{L}_\infty(T)}} . \tag{4.76}$$

4.4.1 Structure-preserving formulation of balanced truncation

We follow the approach of [106] in finding V and W. Recall (4.19), (4.11) and (4.12), and let $\tilde{P}_T$ and $\tilde{Q}_T$ be, respectively, the truncated reachability and observability gramians of the quadratic system (4.16) satisfying the following Lyapunov equations:

$$\tilde{A}_\eta \tilde{P}_T E^T + E\tilde{P}_T \tilde{A}_\eta^T + H(\tilde{P}_l \otimes \tilde{P}_l)H^T + \tilde{B}\tilde{B}^T = 0 \tag{4.77}$$

$$\tilde{A}_\eta^T E^{-T} \tilde{Q}_T + \tilde{Q}_T E^{-1} \tilde{A}_\eta + \mathcal{H}_E^{(2)}(\tilde{P}_l \otimes \tilde{Q}_l)(\hat{\mathcal{H}_E}^{(2)})^T + C^T C = 0, \tag{4.78}$$

where $\mathcal{H}_E^{(2)}$ is the mode-2 matricization of $E^{-1}H$, and $\tilde{P}_l$ and $\tilde{Q}_l$ are the solutions to the Lyapunov equations

$$\tilde{A}_\eta \tilde{P}_l E^T + E\tilde{P}_l \tilde{A}_\eta^T + \tilde{B}\tilde{B}^T = 0 \tag{4.79}$$

$$\tilde{A}_\eta^T E^{-T} \tilde{Q}_l + \tilde{Q}_l E^{-1} \tilde{A}_\eta + C^T C = 0 . \tag{4.80}$$

Then, form the reduction bases V and W as

$$V = \text{blkdiag}\{V_b, V_b, V_b, V_b\} \quad \text{and} \quad W = \text{blkdiag}\{W_b, W_b, W_b, W_b\}, \tag{4.81}$$

with V_b and $W_b \in \mathbb{R}^{n \times r_q}$ being constructed as

$$V_b = R_b^T \hat{U}(:, 1 : r_q)(\hat{\Sigma}(:, 1 : r_q))^{-\frac{1}{2}} \quad \text{and} \quad W_b = S_b^T \hat{V}(:, 1 : r_q)(\hat{\Sigma}(:, 1 : r_q))^{-\frac{1}{2}}, \tag{4.82}$$

where $R_b S_b^T = \hat{U}\hat{\Sigma}\hat{V}^T$ is the SVD of $R_b S_b^T$. The matrix R_b is the Cholesky factor of the second $n \times n$ block of the truncated reachability $\tilde{P}_T$ in (4.77), namely $\tilde{P}_{T_b} = R_b^T R_b$. Similarly, S_b is the Cholesky factor of the second $n \times n$ block of the observability gramians $\tilde{Q}_T$ in (4.78), namely $\tilde{Q}_{T_b} = S_b^T S_b$ [20], [106]. Literature [106] performs the reduction on (4.19) using (4.81) to obtain the quadratic reduced model

$$\begin{aligned} E_r \dot{x}_r(t) &= \tilde{A}_{\eta_r} x_r(t) + H_r(x_r(t) \otimes x_r(t)) + \tilde{B}_r \tilde{u}(t) \\ y_r(t) &= C_r x_r(t), \quad x_r(0) = 0 \end{aligned}$$

However, as opposed to [106], which performs the reduction on the quadratic dynamics (4.19) and constructs a quadratic reduced model, we directly find the reduced second order dynamic (4.1) by extracting $\mathcal{V}$ and $\mathcal{W}$ from V and W. Since the second $n \times n$ blocks of the truncated reachability and observability gramians of the quadratic system correspond to the second order dynamic (1.2), we pick in (4.3)

$$\mathcal{V} = V_b \quad \text{and} \quad \mathcal{W} = W_b \ ,$$

and directly perform the reduction on the second order dynamic (1.2), (1.3) to obtain the reduced second order dynamic (4.1)-(4.2):

$$W_b^T \mathcal{M} V_b \ \ddot{\delta}_r(t) + W_b{}^T \mathcal{D} V_b \ \dot{\delta}_r(t) + W_b{}^T f(V_b \delta_r) = W_b{}^T \mathcal{B} \ u(t),$$
$$y_r(t) = \mathcal{C} V_b \ \delta_r(t)$$

4.4.2 Proper Orthogonal Decomposition

POD [40, 65, 80] or also known as Principal Component Analysis (PCA) in statistics [73] is a method which enables us to extract the most importnt information from a data set. In our setting, this data set is the state snapshot matrix $\Delta \in \mathbb{R}^{n \times L}$ that is formed by simulating the dynamical system (1.2) for a given time interval

$$\Delta = \begin{bmatrix} \delta(t_0) & \delta(t_1) & \dots & \delta(t_{L-1}) \end{bmatrix} \in \mathbb{R}^{n \times L} \ .$$

Then, we compute the SVD of the state snapshot Δ as

$$\Delta = U_\Delta \Sigma_\Delta V_\Delta^T$$

and we pick the reduction basis $\mathcal{V} = \mathcal{W}$ in (4.3) as

$$\mathcal{V} = U_\Delta(:, 1:r),$$

where $U_\Delta(:, 1:r)$ denotes the r leading left singular vectors of Δ, i.e., the leading r columns of U_Δ. Then, form the reduced model (4.1)-(4.2) as

$$\mathcal{V}^T\mathcal{M}\mathcal{V}\ \ddot{\delta}_r(t) + \mathcal{V}^T\mathcal{D}\mathcal{V}\ \dot{\delta}_r(t) + \mathcal{V}^T f(\mathcal{V}\delta_r) = \mathcal{V}^T\mathcal{B}\ u(t),$$
$$y_r(t) = \mathcal{C}\mathcal{V}\ \delta_r(t).$$

Since the reduction base $\mathcal{V}$ is formed by the state snapshot Δ that depends on the simulation of the system, particularly the training input, POD generated reduced models could behave differently for different inputs.

4.4.3 Example 1: New England test system

The model has order $n = 39$ in the original second-order coordinates (1.2). We choose $\eta = 10^{-3}$ in (4.18). We apply both formulations of StrH2 in Algorithm 4 (SthH2-A and StrH2-B), Str-QBT, and POD; and reduce the order to $r = 2, \ldots 25$. In Figure 4.1, the relative $\mathcal{L}_\infty$ error $\|e\|_{\mathcal{L}_\infty(T)}$ in (4.76) with $T = 10$ vs. reduction order r is depicted for each method. The figure shows that for the majority of r values, StrH2-A yields the lowest relative $\mathcal{L}_\infty$ error. We note that this is indeed the best case scenario for POD since the reduced model is tested with the same input that is used to train POD.

For a more detailed comparison, the output of the original and the reduced order models obtained by different methods for $r = 23$, (at which the relative error due to StrH2-A drops below 1%), are depicted in Figure 4.2. The corresponding absolute error, $\mid y(t) - y_r(t) \mid$, plots are shown in Figure 4.3. These two figures support the earlier observation that StrH2-A approximates the original model with a better accuracy compared to POD and Str-QBT.

To investigate the robustness of the algorithms to variations in the input, we slightly perturb

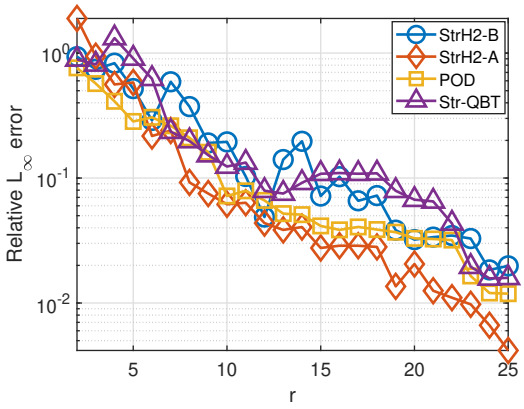


Figure 4.1: Relative $\mathcal{L}_\infty$ error vs. reduction order

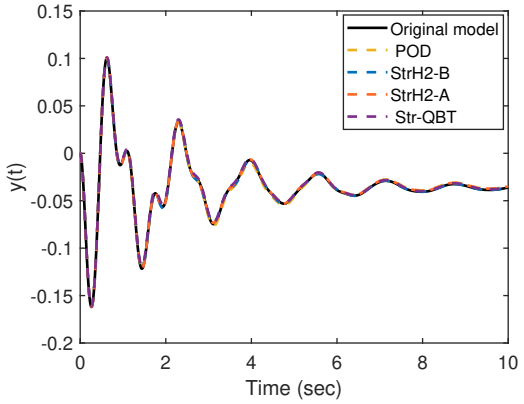


Figure 4.2: Original model and the reduced order models output vs. time for $r = 23$

the input and repeat the same model reduction procedures as before. One might view this perturbation to the input as if the operation conditions are slightly varied. We perturb the input only by 0.1%, the maximum amount that we can add to the input such that the system output remains stable. Note that the choice of input has never entered the proposed algorithm or Str-QBT; however it is used to train the POD model. Figures 4.4 shows the relative error as r varies for this slightly varied input. As in the previous case, StrH2-A outperforms the other methods. We also observe that the relative errors for the POD reduced models stagnate after $r = 13$, i.e. not much further improvements, while StrH2-A and StrH2-B, and Str-QBT approximate the original model with almost the same accuracy as the earlier case. This is expected since the input never entered the model reduction process

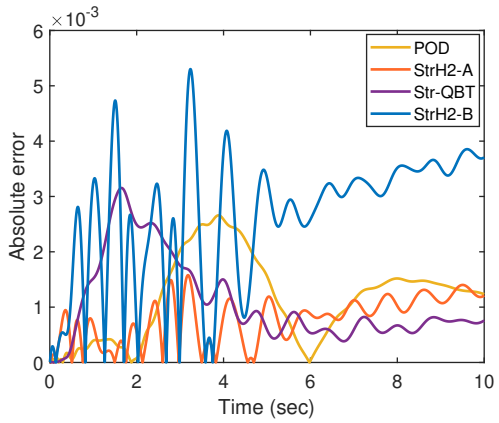


Figure 4.3: Absolute error vs. time for $r = 23$

of the StrH2-A, StrH2-B, or Str-QBT. Thus, for this *small* input variation case, we except those reduced models to provide accurate approximations independent of the input selection as in the linear case. As before, outputs of the original and the reduced order models and

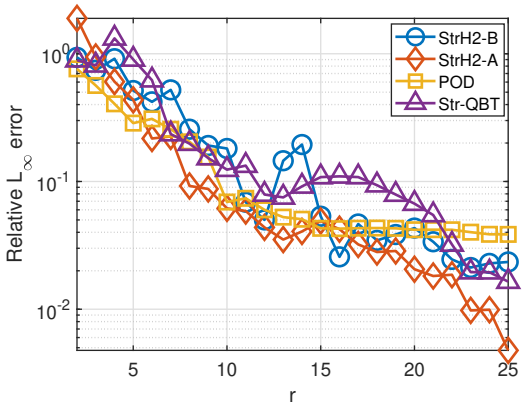


Figure 4.4: Relative $\mathcal{L}_\infty$ error vs. reduction order

the absolute output errors, for $r = 23$, are depicted in Figures 4.5 and 4.6, respectively. While POD has the largest absolute error, the absolute error of StrH2-A is less than 0.14%.

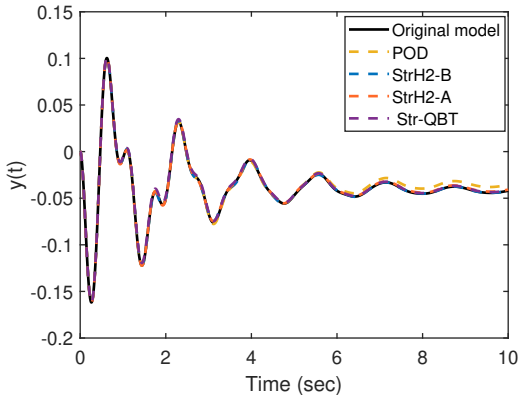


Figure 4.5: Original model and the reduced order models output vs. time for $r = 23$

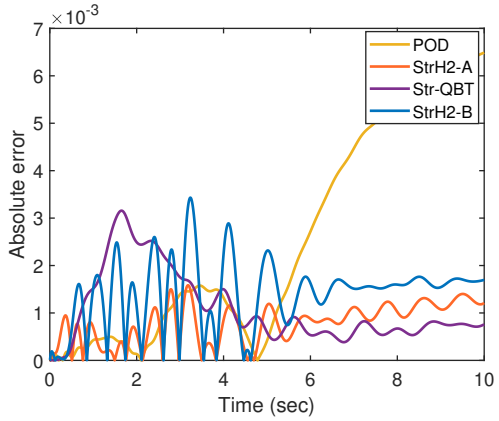


Figure 4.6: Absolute error vs. time for $r = 23$

4.4.4 Example 2: IEEE 118 bus

Here, the second-order model (1.2) is of order $n = 118$. We choose $\eta = 10^{-2}$ and as in the previous example, apply StrH2-A, StrH2-B, and POD for $r = 2, \dots 10$ and over $T = [0\ 3]$. Str-QBT has been ignored in this example due to its poor performance. The relative $\mathcal{L}_\infty$ output error $\|e(t)\|_{\mathcal{L}_\infty(T)}$ vs. reduction r is depicted in Figure 4.7. According to the figure, although POD yields the lower $\mathcal{L}_\infty$ error, StrH2-A has an acceptable accuracy as POD does. We re-emphasize that this is the best case scenario for POD since the same input used to train POD is also used to test the reduced model. For a more detailed comparison, the output of the original and the reduced order models obtained by StrH2-A, StrH2-B and POD for $r = 10$ with the corresponding absolute error, $\mid y(t) - y_r(t) \mid$, plots are illustrated in

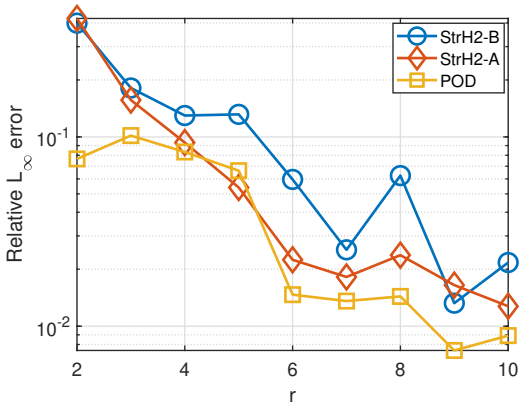
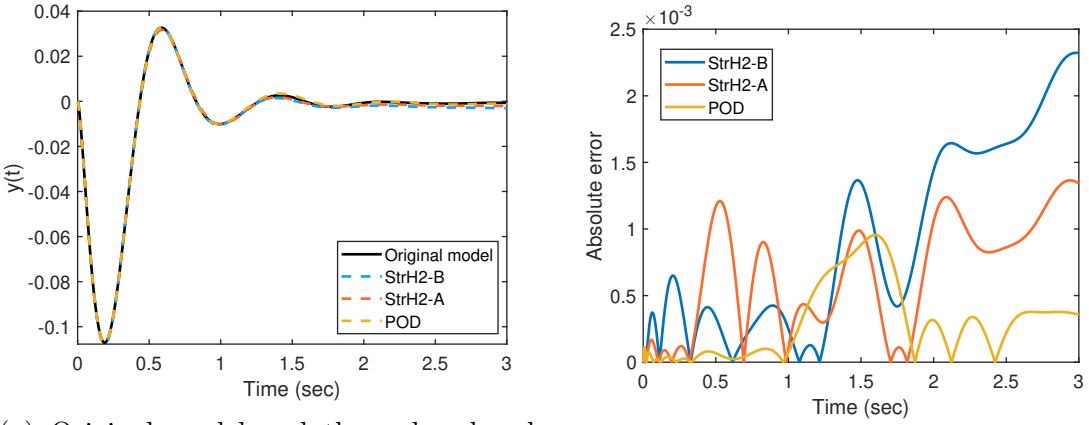


Figure 4.7: Relative $\mathcal{L}_\infty$ error vs. reduction order

Figures 4.8a and 4.8b, respectively. According to the figures, although POD has the lowest absolute error over $T = [0\ 3]$, the absolute error of StrH2-A has an acceptable accuracy of less than 0.14%.



(a) Original model and the reduced order models output vs. time

(b) Absolute error vs. time

Figure 4.8: Comparison of the original and the reduced models for $r = 10$.

As in the previous example, to test the robustness of the algorithms in the presence of a variation in the input, we slightly perturb the input u by 0.5%. As one can observe in Figure 4.9, both StrH2-A and StrH2-B outperform POD. Also as in the previous example, the relative errors for the POD reduced models receive not much further improvements after

$r = 3$, while both StrH2 formulations approximate the original model with similar accuracy as in the earlier case.

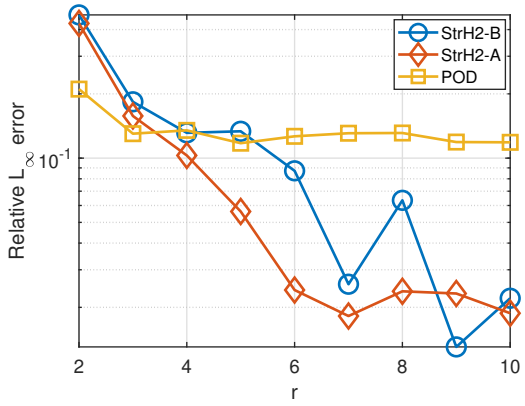
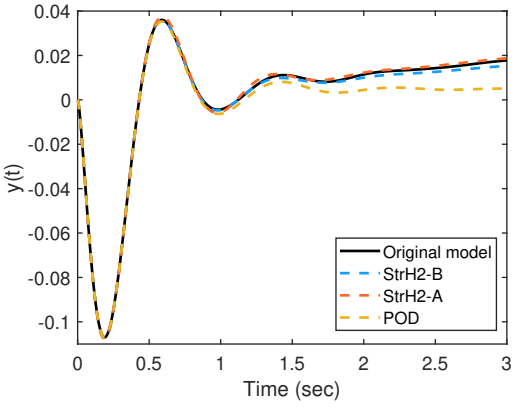
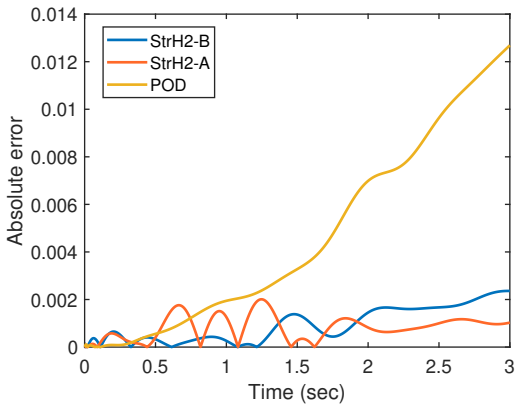


Figure 4.9: Relative $\mathcal{L}_\infty$ error vs. reduction order

As before, outputs of the original and the reduced order models and the absolute output errors, for $r = 10$, are shown in Figures 4.10a and 4.10b, respectively. While POD has the largest absolute error (less than 1.26%), the absolute errors of StrH2-A and StrH2-B are, respectively, less than 0.2% and 0.24%.



(a) Original model and the reduced order models output vs. time



(b) Absolute error vs. time

Figure 4.10: Comparison of the original and the reduced models for $r = 10$.

4.4.5 The need for (D)EIM

Even though the reduced order nonlinear term

$$f_r(\delta_r) = \mathcal{V}^T f(\mathcal{V}\delta_r) \tag{4.83}$$

in (4.3) lies in an r-dimensional subspace, it has a computational cost that depends on the full order dimension n since it requires evaluating the original nonlinear function f at $\mathcal{V}\delta_r \in \mathbb{R}^n$. This is known as the lifting bottleneck that may make the simulation of the reduced model as costly as that of the original. One way to tackle this issue is to approximate the nonlinearity $f_r(\delta_r)$ via (Discrete) Empirical Interpolation Method (DEIM) [10, 41] as

$$f(\mathcal{V}\delta_r) \approx P_{\text{DEIM}} f(\mathcal{V}\delta_r), \tag{4.84}$$

where

$$P_{\text{DEIM}} = U_m(\mathcal{S}^T U_m)^{-1}\mathcal{S}^T.$$

Matrix $U_m \in \mathbb{R}^{n\times m}$ with $m \ll n$ and $U_m^T U_m = I_m$ is a POD basis for the nonlinear snapshot F obtained from the original full model

$$F = \begin{bmatrix} f(\delta(t_1)) & \dots & f(\delta(t_k)) \end{bmatrix},$$

and $\mathcal{S} \in \mathbb{R}^{n\times m}$ is a row selection operator, obtained by picking m columns of the $n \times n$ identity matrix, such that $\mathcal{S}^T U_m$ is invertible. Then, the DEIM approximation to $f_r(\delta_r)$ is

given by

$$f_r(\delta_r) \approx \mathcal{V}^T U_m (\mathcal{S}^T U_m)^{-1} \mathcal{S}^T f(\mathcal{V}\delta_r) \tag{4.85}$$

and in the final reduced model (4.3), the reduced nonlinear term is replaced by the DEIM approximation (4.85). Clearly, the effectiveness of the DEIM approximation depends on the selection operator $\mathcal{S}$ and this is done by picking $\mathcal{S}$ such that $\|(\mathcal{S}^T U_m)^{-1}\|_2$ is as small as possible in the DEIM error bound [10, 41, 49, 50]

$$\|f - P_{\text{DEIM}} f\| \leq \|(\mathcal{S}^T U_m)^{-1}\| \|e_{\text{opt}}\|, \tag{4.86}$$

where $e_{\text{opt}} = (I_n - U_m U_m^T) f$.

There are other variations and extensions to DEIM such as QDEIM [50], adaptive DEIM [93, 117], DEIM-CUR [115], localized DEIM [95], and DEIM in the presence of noise [96]. In the next chapter, we replace F with the nonlinear snapshot $F(\mathcal{V})$ that depends on the reduction base $\mathcal{V}$ and is defined as

$$F(\mathcal{V}) = \begin{bmatrix} f(\mathcal{V}\delta_r(t_1)) & \dots & f(\mathcal{V}\delta_r(t_k)) \end{bmatrix},$$

which requires the simulation of the reduced model without DEIM. Also, we use the QDEIM selection operator from [50] that uses QR factorization of U_m^T with column pivoting to determine $\mathcal{S}$. This leads to a sharper error bound for (4.86).

4.4.5.1 Example: IEEE 118 bus

We use the IEEE 118 bus system to employ QDEIM in our framework. We vary the reduction order from $r = 2$ to $r = 10$ with increments of one and choose the corresponding QDEIM

dimension m based on the singular value decays of the nonlinear snapshots using a relative truncation tolerance of 1×10^{-6}. The corresponding m value for every r is given in Table 4.1. In Figure 4.11, we illustrate the performance of StrH2-A with and without QDEIM. As the figure depicts, QDEIM-based reduced model is able to match the high-fidelity behavior of the reduced model without it (almost replicates) while at the same decreasing the computational cost arising from the lifting bottleneck.

Table 4.1: QDEIM dimension vs. reduced order

r	2	3	4	5	6	7	8	9	10
m	6	9	12	13	16	17	18	19	20

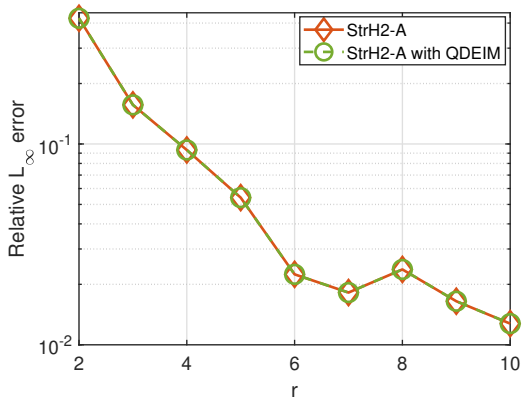


Figure 4.11: Performance of the StrH2-A with QDEIM

4.5 Summary

We have developed a structure-preserving system-theoretic model reduction technique for nonlinear power grid networks. We used a lifting transformation to convert the original nonlinear model to its equivalent quadratic system. With the help of this quadratic representation, we are able to employ Q-IRKA as an intermediate stage of our model reduction framework. We have shown that the model reduction bases obtained by Q-IRKA has a

specific subspace structure that can be exploited to construct the desired reduction basis for reducing the original nonlinear model. This reduction basis has led to a reduced model that preserved the physically meaningful second-order structure of the original model. We have illustrated the effectiveness of our proposed approach via two numerical examples.

Chapter 5

Data-driven Modeling of Power Networks

Recall that in Section (4.2) of Chapter 4, we showed that we can find a reduced quadratic model as in (4.7) for (3.3) via projection. However, obtaining the reduced order matrices in (4.8) requires the knowledge of the full order model, i.e., the full order matrices $\mathcal{M}$, $\mathcal{D}$, $\mathcal{B}$ and $f(\delta)$ in (1.2) that may not always be available or easy to derive. In some cases, even though the reduced order matrices can no longer be obtained via intrusive projection-based model reduction as in (4.8), they can be inferred from data. Recently, non-intrusive model reduction methods (data-driven methods) have received great attention. These methods learn a model based on data and without explicitly having access to the full order model operators. Various methods have been used to construct a reduced model from data. While some approaches are based on the frequency-domain data (see, e.g., [5, 7, 51, 55, 62, 70, 76, 82]) the others use time-domain data (see, e.g., [53, 75, 78, 81, 102, 107, 113, 126]).

In this chapter, we aim to learn a quadratic reduced model for nonlinear swing equations modeling power network dynamics and employ the Lift and Learn approach [102]. We apply the previously defined lifting map (3.37) in Section 3.2 to data obtained by evaluating the swing equations. The lifted data is projected onto a lower dimensional basis. Then, lower dimensional quadratic matrix operators are fitted to this reduced lifted data by a least-squares operator inference procedure.

The remainder of this chapter is organized as follows: In Section 5.1, we review the Lift and Learn approach of [102]. Section 5.2 presents the proposed data-driven framework where we apply the Lift and Learn approach to learn a low dimensional quadratic model for swing equations. Section 5.3 illustrates the feasibility of our approach via numerical examples. Finally, in Section 5.3.3, we employ the regularization strategy of [84] to improve the predictive power of the inferred quadratic reduced models.

Much of the material of this chapter has appeared in the published manuscript [109].

5.1 Lift and Learn method for quadratic models

The Lift and Learn approach [102] is a powerful data-driven approach that uses the simulation data from the original nonlinear model (without access to its full-order state-space representation) to learn a quadratic reduced-order approximation to it. First, the approach collects state trajectory data of the original nonlinear model. Next, it lifts the data by a proper problem-dependent mapping to a quadratic model, and project the lifted data onto a low-dimensional basis via singular value decomposition (SVD). Then, it fits the reduced quadratic operators to the data by the least-squares operator inference procedure [94].

To be more precise, consider the following nonlinear dynamical system of n ordinary differential equations

$$\dot{z} = f(z, u), \tag{5.1}$$

where $z \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^l$ is the input, and $f(z, u) : \mathbb{R}^n \times \mathbb{R}^l \to \mathbb{R}^n$ is a nonlinear function. For the dynamics (5.1), collect K state snapshot data and input trajectory data

at the time samples t_k for $k = 0, \ldots, K-1$:

$$Z = \begin{bmatrix} z(t_0) & z(t_1) & \ldots & z(t_{K-1}) \end{bmatrix} \in \mathbb{R}^{n \times K}, \tag{5.2}$$
$$U = \begin{bmatrix} u(t_0) & u(t_1) & \ldots & u(t_{K-1}) \end{bmatrix} \in \mathbb{R}^{l \times K}.$$

Define a lifting map $\mathcal{T} : \mathbb{R}^n \to \mathbb{R}^N$

$$\mathcal{T} : z \to x \tag{5.3}$$

such that in the lifted-state x, the dynamics (5.1) can be written exactly as a quadratic model (5.4), i.e.,

$$\begin{aligned} \dot{x}(t) &= Ax(t) + H(x(t) \otimes x(t)) + Bu(t), \\ y(t) &= Cx(t), \end{aligned} \tag{5.4}$$

similar to what we did in (3.2) and (3.3) for the power network dynamics. Note that (3.3) can be written in the form of (5.4) by multiplying the dynamics of (3.3) with E^{-1}. Then, apply this lifting map on each column of state snapshot (5.2) to form the lifted snapshot:

$$X = \begin{bmatrix} x(t_0) & x(t_1) & \ldots & x(t_{K-1}) \end{bmatrix} \in \mathbb{R}^{N \times K}. \tag{5.5}$$

Compute the economy-size singular value decomposition (SVD) of the lifted snapshot X:

$$X = \Phi \Sigma \Psi^T,$$

where $\Sigma = \mathsf{diag}(\sigma_1, \sigma_2, \ldots, \sigma_K) \in \mathbb{R}^{K \times K}$ is diagonal with the singular values $\{\sigma_i\}$ of X on

the diagonal, and $\Phi \in \mathbb{R}^{N\times K}$ and $\Psi \in \mathbb{R}^{K\times K}$ have orthonormal columns. (Here we assumed $N \geq K$; the $N \leq K$ case follows similarly.) Based on the decay of $\{\sigma_i\}$, choose a truncation index r and use the leading r columns of Φ, denoted by Φ_r, to construct the reduced lifted snapshot matrix $X_r \in \mathbb{R}^{r\times K}$

$$X_r = \Phi_r^T X. \tag{5.6}$$

The goal is to learn a reduced quadratic approximation, as

$$\begin{aligned} \dot{x}_r(t) &= A_r x_r(t) + H_r(x_r(t) \otimes x_r(t)) + B_r u(t) \\ y_r(t) &= C_r x_r(t), \end{aligned} \tag{5.7}$$

to the original nonlinear dynamics (5.1). It is clear from (5.7) that to infer the reduced operators A_r, H_r and B_r, the reduced time derivative of state snapshot $\dot{X}_r$ is also required in addition to the reduced state snapshot X_r in (5.6). Time derivative snapshot $\dot{X} \in \mathbb{R}^{4n\times K}$ can be obtained from the state snapshot (5.5) using a time derivative approximation [24], e.g., the forward difference approximation. Then, similar to (5.6), [102] obtains the reduced time derivative snapshot $\dot{X}_r \in \mathbb{R}^{r\times K}$ as

$$\dot{X}_r = \Phi_r^T \dot{X}. \tag{5.8}$$

5.1.1 Least-squares operator inference procedure

Given K reduced state snapshot (5.6), its corresponding reduced time derivative (5.8) and the input snapshot in (5.2), operator inference approach in [94] formulates the following least-squares minimization problem to obtain the reduced order matrices $A_r \in \mathbb{R}^{r\times r}$, $H_r \in \mathbb{R}^{r\times r^2}$,

$B_r \in \mathbb{R}^{r\times l}$ appearing in (5.7):

$$\min_{A_r,H_r,B_r} \frac{1}{K}\left\| X_r^T A_r^T + (X_r \otimes X_r)^T H_r^T + U^T B_r^T - \dot{X}_r^T \right\|_F^2. \tag{5.9}$$

The least-squares problem (5.9) is linear in the unknown variables A_r, H_r and B_r. Hence, the minimization problem (5.9) can be transformed into solving the linear least-squares problem [94], [102]

$$\min_{\mathsf{X}\in\mathbb{R}^{(r+l+\frac{r^2+r}{2})\times r}} \|\mathsf{AX}-\mathsf{B}\|_F^2, \tag{5.10}$$

with

$$\mathsf{A} = \begin{bmatrix} X_r^T & U^T & \hat{X}_r^T \end{bmatrix} \in \mathbb{R}^{K\times(r+l+\frac{r^2+r}{2})}, \quad \mathsf{X} = \begin{bmatrix} A_r^T \\ B_r^T \\ \hat{H}_r^T \end{bmatrix}, \quad \text{and} \quad \mathsf{B} = \dot{X}_r^T, \tag{5.11}$$

where $\hat{X}_r = (X_r \mathbin{\widehat{\otimes}} X_r) \in \mathbb{R}^{\frac{r^2+r}{2}\times K}$ is constructed as

$$X_r \mathbin{\widehat{\otimes}} X_r = \begin{bmatrix} x_{r_1} \mathbin{\widehat{\otimes}} x_{r_1} & x_{r_2} \mathbin{\widehat{\otimes}} x_{r_2} & \dots & x_{r_K} \mathbin{\widehat{\otimes}} x_{r_K} \end{bmatrix}, \tag{5.12}$$

where x_{r_i} is the ith column of X_r and $\widehat{\otimes}$ denotes the Kronecker product $\otimes$ without the redundant/repeated terms [102]. For example, for $x = [x_1 \ x_2]^T$, the standard Kronecker product yields $x \otimes x = [x_1^2 \ x_1x_2 \ x_2x_1 \ x_2^2]^T$ while by removing the repeated term x_2x_1, we have $x \mathbin{\widehat{\otimes}} x = [x_1^2 \ x_1x_2 \ x_2^2]^T$.

Similarly, $\hat{H}_r \in \mathbb{R}^{r\times\frac{r^2+r}{2}}$ is the form of H_r without redundancy such that we can construct $H_r \in \mathbb{R}^{r\times r^2}$ from $\hat{H}_r$ by splitting the values corresponding to the quadratic cross terms across

the redundant terms. For example, for $r = 3$,

$$\hat{H}_r = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \end{bmatrix} \text{ and } H_r = \begin{bmatrix} h_{11} & \frac{h_{12}}{2} & \frac{h_{12}}{2} & h_{13} \\ h_{21} & \frac{h_{22}}{2} & \frac{h_{22}}{2} & h_{23} \end{bmatrix}.$$

The least-squares (5.10) problem has a unique solution for $K \geq (r + \frac{r^2+r}{2} + l)$ if A has a full column rank [54]. In addition, it can be expressed as r independent least-squares problems as

$$\min_{\mathsf{X}} \|\mathsf{A}\mathsf{X}(:, i) - \mathsf{B}(:, i)\|_2^2 \;\; ; \;\; i = 1, \ldots, r, \tag{5.13}$$

where $\mathsf{X}(:, i)$ is the MATLAB notation referring to the ith column of X.

In the following section, we show how the Lift and Learn approach of [102] can be applied for data-driven modeling of nonlinear swing equations in power networks.

5.2 Learning power networks model from data

Recall the second-order dynamics of power grid networks given in (1.2), which we repeat here:

$$\mathcal{M}\ddot{\delta}(t) + \mathcal{D}\dot{\delta}(t) + f(\delta) = \mathcal{B}u(t), \tag{1.2}$$

$$y(t) = \mathcal{C}\delta(t). \tag{1.3}$$

For this nonlinear model (1.2), our goal in this section is to infer a reduced quadratic model approximation that preserves the quadratic structure, as in (5.7)

$$\begin{aligned} \dot{x}_r(t) &= A_r x_r(t) + H_r(x_r(t) \otimes x_r(t)) + B_r u(t), \\ y_r(t) &= C_r x_r(t), \end{aligned} \tag{5.7}$$

using the data trajectories of δ, $\dot{\delta}$ and input u (without access to the full order operators $\mathcal{M}$, $\mathcal{D}$, $f(\delta)$, $\mathcal{B}$) by employing the Lift and Learn approach [102] reviewed in Section 5.1. In other words, the goal is to fit quadratic reduced order matrices A_r, H_r and B_r to the reduced snapshot data.

For the nonlinear model (1.2) under investigation, let Z, $\dot{Z}$ and U be, respectively, the K snapshots of δ, $\dot{\delta}$ and input at time instances t_k; $k = 0, \ldots, K-1$:

$$\begin{aligned} Z &= \begin{bmatrix} \delta(t_0) & \delta(t_1) & \ldots & \delta(t_{K-1}) \end{bmatrix} \in \mathbb{R}^{n \times K}, \\ \dot{Z} &= \begin{bmatrix} \dot{\delta}(t_0) & \dot{\delta}(t_1) & \ldots & \dot{\delta}(t_{K-1}) \end{bmatrix} \in \mathbb{R}^{n \times K}, \\ U &= \begin{bmatrix} u(t_0) & u(t_1) & \ldots & u(t_{K-1}) \end{bmatrix} \in \mathbb{R}^{1 \times K}. \end{aligned} \tag{5.14}$$

Recall that in (1.2), $u(t) = 1$, thus in this setting, we have

$$U = \begin{bmatrix} 1 & 1 & \ldots & 1 \end{bmatrix} \in \mathbb{R}^{1 \times K}. \tag{5.15}$$

Also recall the lifting map $\mathcal{T} : \mathbb{R}^{2n} \to \mathbb{R}^{4n}$ in (3.37)

$$\mathcal{T} : \begin{pmatrix} \delta(t) \\ \dot{\delta}(t) \end{pmatrix} \to \begin{pmatrix} \delta(t) \\ \dot{\delta}(t) \\ \sin(\delta(t)) \\ \cos(\delta(t)) \end{pmatrix} = x(t). \tag{3.37}$$

(we called $x(t)$ in Chapter 3 as $q(t)$). Then, using $\mathcal{T}$ in (3.37), the lifted data matrix $X \in \mathbb{R}^{4n \times K}$ is obtained as

$$X = \begin{bmatrix} \delta(t_0) & \delta(t_1) & \dots & \delta(t_{K-1}) \\ \dot{\delta}(t_0) & \dot{\delta}(t_1) & \dots & \dot{\delta}(t_{K-1}) \\ \sin(\delta(t_0)) & \sin(\delta(t_1)) & \dots & \sin(\delta(t_{K-1})) \\ \cos(\delta(t_0)) & \cos(\delta(t_1)) & \dots & \cos(\delta(t_{K-1})) \end{bmatrix} = \begin{bmatrix} x(t_0) & x(t_1) & \dots & x(t_{K-1}) \end{bmatrix}. \tag{5.16}$$

The time derivative data snapshot $\dot{X}$ is also computed as explained in Section 5.1. Next, we compute the SVD basis Φ_r for the lifted state data X as in (5.16):

$$X = \Phi \Sigma \Psi^T, \quad \text{where} \quad \Phi_r = \Phi(:, 1:r)$$

to construct the reduced lifted state data X_r as in (5.6)

$$X_r = \Phi_r^T X \tag{5.6}$$

and the reduced lifted time derivative data $\dot{X}_r$ in (5.8) via projection.

$$\dot{X}_r = \Phi_r^T \dot{X}. \tag{5.8}$$

Then, by solving the least-squares minimization

$$\min_{A_r,H_r,B_r} \frac{1}{K}\left\| X_r^T A_r^T + (X_r \otimes X_r)^T H_r^T + U^T B_r^T - \dot{X}_r^T \right\|_F^2. \tag{5.9}$$

using the reduced lifted data and input snapshot in (5.14), we infer matrices A_r, H_r and B_r. The reduced matrix C_r is calculated as

$$C_r = C\Phi_r,$$

where C yields the output of choice. The resulting data-driven modeling approach for non-linear power networks (1.2) via Lift and Learn is summarized in Algorithm (5).

In situations where the coefficient matrix A in (5.11) is rank-deficient, [119] proposes to regularize the least-squares problem (5.13) with an $\mathcal{L}_2$ regularization as

$$\min_{\mathsf{X}} \|\mathsf{A}\mathsf{X}(:,i) - \mathsf{B}(:,i)\|_2^2 + \mu\|\mathsf{X}(:,i)\|_2^2, \quad \text{for } i = 1,\ldots,r \tag{5.17}$$

where $\mu > 0$ is the regularization tuning parameter that controls a trade-off between solutions that fit the data well and solutions with a small norm. Regularization avoids over-fitting and improves the conditioning of the problem as well as the stability of the reduced order model. As we discuss in the next section, for the power network models we have studied, we have frequently encountered this situation in our numerical examples and had to employ the regularization process in our implementation.

Algorithm 5 Lift and Learn for Power Network Models

Input: Snapshot data and input trajectory for the original nonlinear model (1.2)

Output: A_r, H_r, B_r and C_r

1. Collect the snapshot data Z, $\dot{Z}$, and U (5.14) from the power network model (1.2).

$$Z = \begin{bmatrix} \delta(t_0) & \delta(t_1) & \dots & \delta(t_{K-1}) \end{bmatrix}$$
$$\dot{Z} = \begin{bmatrix} \dot{\delta}(t_0) & \dot{\delta}(t_1) & \dots & \dot{\delta}(t_{K-1}) \end{bmatrix}$$
$$U = \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}$$

2. Use the lifting map (3.37) to find the lifted state snapshot X (5.16).

$$X = \begin{bmatrix} \delta(t_0) & \delta(t_1) & \dots & \delta(t_{K-1}) \\ \dot{\delta}(t_0) & \dot{\delta}(t_1) & \dots & \dot{\delta}(t_{K-1}) \\ \sin(\delta(t_0)) & \sin(\delta(t_1)) & \dots & \sin(\delta(t_{K-1})) \\ \cos(\delta(t_0)) & \cos(\delta(t_1)) & \dots & \cos(\delta(t_{K-1})) \end{bmatrix}$$

3. Compute the SVD basis Φ_r for the lifted state data.

$$X = \Phi \Sigma \Psi^T, \quad \Phi_r = \Phi(:, 1:r)$$

4. Compute the reduced lifted state data (5.6) and the reduced lifted time derivative data (5.8) via projection.

$$X_r = \Phi_r^T X \tag{5.6}$$
$$\dot{X}_r = \Phi_r^T \dot{X}. \tag{5.8}$$

5. Solve least-squares minimization (5.9) using the reduced lifted data and input snapshot in (5.14) to infer matrices A_r, H_r and B_r as in (5.7).

$$\min_{A_r, H_r, B_r} \frac{1}{K} \left\| X_r^T A_r^T + (X_r \otimes X_r)^T H_r^T + U^T B_r^T - \dot{X}_r^T \right\|_F^2. \tag{5.9}$$

6. Using the desired output of choice (C is given), calculate $C_r = C\Phi_r$.

5.3 Numerical example

The two test systems we investigate are the SM model of the IEEE 118 bus system introduced in Section 4.4 with $n = 118$ and the EN model of IEEE 300 with $n = 69$, included in the MATPOWER software toolbox [131], [133]. We focus on the single-output system ($p = 1$); thus $\mathcal{C} \in \mathbb{R}^{1\times n}$. We choose the output, quantity of interest, $y(t)$ as the arithmetic mean of all phase angles $\delta(t)$ as done in [106]. In both cases, we obtain the data via a numerical simulation with the time step size $\Delta t = 10^{-3}$ and the regularization tuning parameter $\mu = 10^{-3}$. The inferred reduced order r is chosen based on the singular value decay of the snapshot data X with a relative tolerance of tol $= 1.5 \times 10^{-4}$. In our simulations, we have employed the operator inference source code provided in [101].

5.3.1 Example 1: IEEE 118 bus

We collect the data snapshots for the time interval $T = [0\ 3]$ seconds. With a step size of $\Delta t = 10^{-3}$, this leads to the snapshot matrix $X \in \mathbb{R}^{472\times 3001}$. Based on the singular value decay of X as shown in Figure 5.1 and the relative truncation tolerance of 1.5×10^{-4}, we choose $r = 23$ and form the projection basis $\Phi_r \in \mathbb{R}^{472\times 23}$. Based on the reduced lifted data

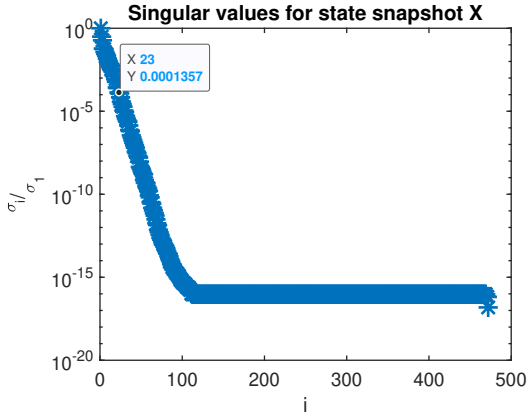


Figure 5.1: Singular values for state snapshot X

$X_r \in \mathbb{R}^{23\times 3001}$ and $\dot{X}_r \in \mathbb{R}^{23\times 3001}$, and the input snapshot $U = \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}$, resulting coefficient matrix $\mathsf{A} \in \mathbb{R}^{3001\times 300}$ is rank-deficient with $\text{rank}(\mathsf{A}) = 82 < 300$. Therefore, we solve the regularized least-squares problem (5.17) with $\mu = 10^{-3}$. Using Algorithm 5, we find the data-driven quadratic reduced matrices $A_r \in \mathbb{R}^{23\times 23}$, $H_r \in \mathbb{R}^{23\times (23)^2}$ and $B_r \in \mathbb{R}^{23}$ in (5.7). To test the accuracy of the inferred model, we compare full-order model output $y(t)$ with the reduced quadratic output $y_r(t)$ calculated as

$$y_r(t) = \begin{bmatrix} \mathcal{C} & 0 & 0 & 0 \end{bmatrix} \Phi_r x_r(t)$$

in Figure 5.2. As the figure illustrates, the data-driven reduced quadratic model of order $r = 23$, obtained without access to original power network dynamics, accurately approximates the full model output. Recall the $\mathcal{L}_\infty(T)$ norm of the output $y(t)$ in (4.75) which we repeat

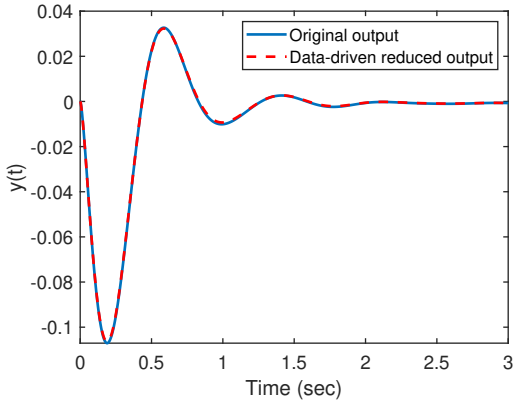


Figure 5.2: Comparison of original output and the data-driven reduced output

here

$$\|y\|_{\mathcal{L}_\infty(T)} = \max_{t\in T} \mid y(t) \mid . \tag{4.75}$$

The relative output error

$$e(t) = \frac{\mid y(t) - y_r(t) \mid}{\|y\|_{\mathcal{L}_\infty(T)}} \tag{5.18}$$

for $T = [0\ 3]$ is shown in Figure 5.3. Figure 5.3 illustrates that, the learned model achieves a relative $\mathcal{L}_\infty$ error $\|e\|_{\mathcal{L}_\infty(T)}$ of less than 0.9% with a reduced order $r = 23$.

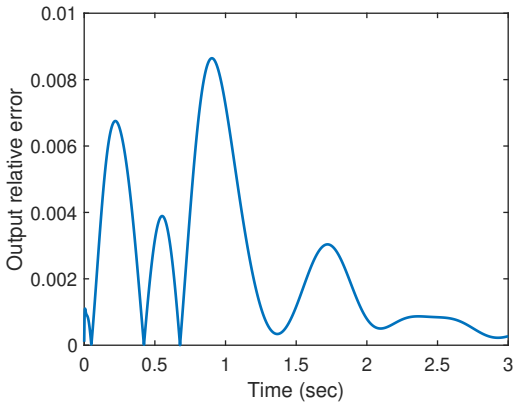


Figure 5.3: Relative $\mathcal{L}_\infty$ error vs. time

5.3.2 Example 2: IEEE 300

In this example, we use EN model of IEEE 300 with $n = 69$. We collect the data snapshots for the time interval $T = [0\ 10]$ and obtain the snapshot matrix $X \in \mathbb{R}^{276\times 10001}$. Based on the singular value decay depicted in Figure 5.4, we choose $r = 46$. As in the previous example, the coefficient matrix $\mathsf{A} \in \mathbb{R}^{10001\times 1128}$ is rank-deficient ($\text{rank}(\mathsf{A}) = 221 < 1128$). Hence, we solve (5.17) with $\mu = 10^{-3}$ to infer the reduced operators $A_r \in \mathbb{R}^{46\times 46}$, $H_r \in \mathbb{R}^{46\times 46^2}$, $B_r \in \mathbb{R}^{46}$.

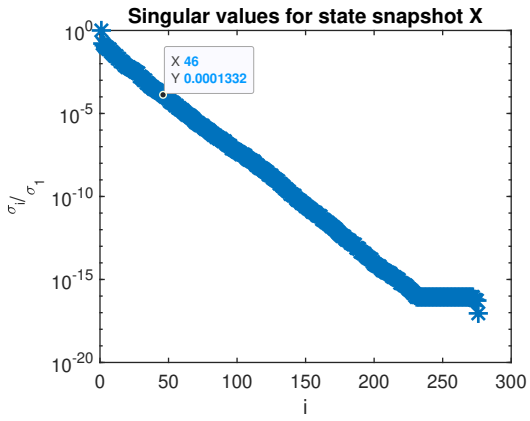


Figure 5.4: Singular values for state snapshot X

The outputs of the full-order and the reduced quadratic models are shown in Figure 5.5, once again illustrating an accurate match of the power network output via the learned model.

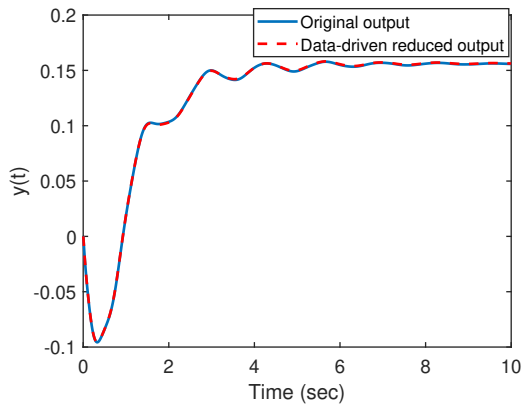


Figure 5.5: Comparison of original output and the data-driven reduced output

Figure 5.6 illustrates relative output error over the simulation time. According to the figure, the reduced model successfully approximate the full model with a relative $L_\infty(T)$ output error less than 0.46% over the time-interval $T = [0\ 10]$ seconds.

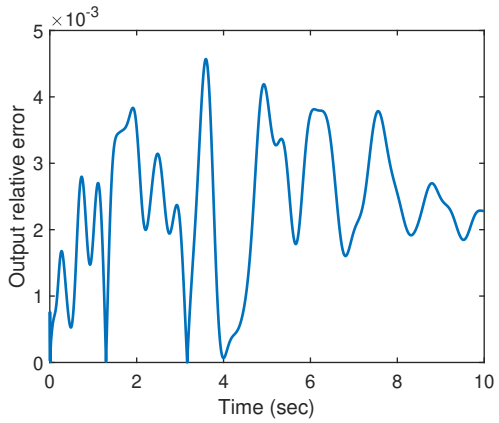


Figure 5.6: Relative $\mathcal{L}_\infty$ error vs. time

5.3.3 Operator inference with regularization selection

Recall that in Section 5.2, we used an $\mathcal{L}_2$ regularizer $\mu I_{r+l+\frac{r^2+r}{2}}$ in (5.10) as

$$\min_{\mathsf{X}\in\mathbb{R}^{(r+l+\frac{r^2+r}{2})\times r}} \|\mathsf{A}\mathsf{X}-\mathsf{B}\|_F^2 + \mu\|\mathsf{X}\|_F^2 \tag{5.19}$$

to improve the conditioning of the problem, where $\mu > 0$ is a regularization parameter and I is the identity matrix. The unique minimizer of the least squares problem (5.19) satisfies

$$\mathsf{X} = (\mathsf{A}^T\mathsf{A} + \mu^2 I_{r+l+\frac{r^2+r}{2}})^{-1}\mathsf{A}^T\mathsf{B},$$

where $(\mathsf{A}^T\mathsf{A}+\mu^2 I_{r+l+\frac{r^2+r}{2}})$ is symmetric positive definite [84]. The regularizer in (5.19) equally penalizes each entry of X, i.e., the inferred reduced operators A_r, B_r and $\hat{H}_r$. However, the entries of the quadratic operator $\hat{H}_r$ have a different scaling compared to the entries of A_r, B_r [84]. Therefore, the learned reduced models obtained by (5.19) may sometimes have poor predictive performance outside the training time interval $T = [0\ t_{K-1}]$ (see, e.g., Figure 5.7). To solve this overfitting issue, [84] proposes a block diagonal regularizer of the form

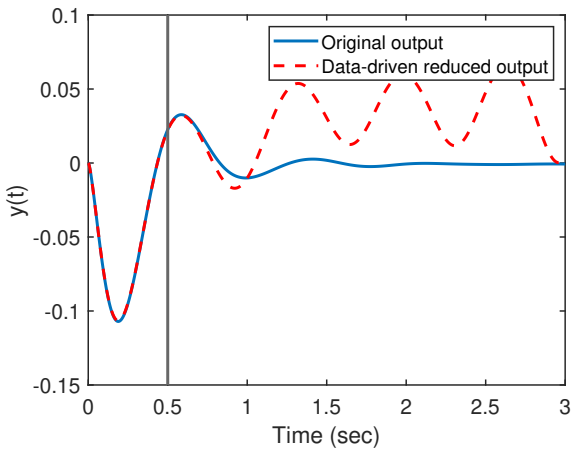


Figure 5.7: Example 1 with training time $T = [0\ 0.5]$ ($X \in \mathbb{R}^{472\times 501}$) has a poor performance for the prediction period [0.7 3]. The training and prediction periods are separated by the vertical black lines.

$$\Upsilon = \text{blkdiag}(\mu_1 I_{r+l}\,, \mu_2 I_{\frac{r^2+r}{2}}) \in \mathbb{R}^{(r+l+\frac{r^2+r}{2})\times(r+l+\frac{r^2+r}{2})}, \tag{5.20}$$

with $\mu_1, \mu_2 > 0$ such that μ_1 penalizes the entries of A_r and B_r , and μ_2 penalizes the entries of $\hat{H}_r$. In other words, using (5.20), the least squares problem (5.19) is replaced by

$$\min_{\mathsf{X}\in\mathbb{R}^{(r+l+\frac{r^2+r}{2})\times r}} \|\mathsf{AX}-\mathsf{B}\|_F^2 + \mu_1(\|A_r\|_F^2 + \|B_r\|_F^2) + \mu_2\|\hat{H}_r\|_F^2\ . \tag{5.21}$$

The parameters μ_1 and μ_2 are chosen such that the resulting learned reduced model minimizes some error metric over $T = [0\ t_{K-1}]$ keeping the integrated POD coefficients bounded over $T_f = [0\ t_f]$, where $t_f > t_{K-1}$. In other words, we want each entries of the reduced quadratic state $x_r(t) = [x_{r_1}(t)\ \ \ldots\ \ x_{r_r}(t)]^T$ in (5.7) to satisfy

$$\left|x_{r_i}(t)\right| \leq \tau, \quad i = 1, \ldots, r \ \text{ and } \ T_f = [0\ t_f], \tag{5.22}$$

for some $\tau > 0$. Using (5.22) and the fact that $x(t) = \Phi_r x_r(t)$, one can write

$$\left|x_i(t)\right| = \left|\sum_{j=1}^{r} \big(\Phi(i,j)\big) x_{r_i}(t)\right| \leq \sum_{j=1}^{r} \left|\Phi(i,j)\right| \left|x_{r_i}(t)\right| \leq \tau \sum_{j=1}^{r} \left|\Phi(i,j)\right| \quad i = 1, \ldots, n, \quad (5.23)$$

which means that bounding the magnitude of entries of the reduced state $x_r(t)$ ensures a bound on the magnitude of entries of the original high dimensional state $x(t)$ [84]. A detailed description of the regularized operator inference selection procedure [84] for power network models is given in Algorithm 6, where τ is chosen proportional to the maximum absolute entry of X_r [84], i.e.,

$$\tau = \beta \max_{i,j} |X_r(i,j)|, \quad (5.24)$$

where $\beta \geq 1$. According to the algorithm, as before, first we collect the snapshot data Z, $\dot{Z}$, and U (5.14) from the power network model (1.2) and use the lifting map (3.37) to find the lifted state snapshot X (5.16). Second, we compute the SVD basis $\Phi_r = \Phi(:, 1:r)$ for the lifted state data $X = \Phi \Sigma \Psi^T$. Third, we compute the reduced lifted state data X_r in (5.6) and the reduced lifted time derivative data $\dot{X}_r$ in (5.8) via projection. Next, we solve least-squares minimization (5.21) using the reduced lifted data and input snapshot in (5.14), and an initial value for (μ_1, μ_2) to infer matrices A_r, H_r and B_r as in (5.7).

$$\min_{\mathsf{X} \in \mathbb{R}^{(r+l+\frac{r^2+r}{2}) \times r}} \|\mathsf{A}\mathsf{X} - \mathsf{B}\|_F^2 + \mu_1(\|A_r\|_F^2 + \|B_r\|_F^2) + \mu_2 \|\hat{H}_r\|_F^2 \,. \quad (5.21)$$

Then, we simulate the corresponding quadratic dynamic for $T_f = [0 \; t_f]$

$$\dot{x}_r(t) = A_r x_r(t) + H_r(x_r(t) \otimes x_r(t)) + B_r u(t), \quad (5.7)$$

and gather its snapshot for $T_f = [0\ t_f]$, namely X_{r_f}:

$$X_{r_f} = \begin{bmatrix} x_r(t_0) & x_r(t_1) & \dots & x_r(t_{K_f-1}) \end{bmatrix}.$$

If $\max_{i,j} |X_{r_f}(i,j)| <= \beta \max_{i,j} |X_r(i,j)|$, we compute and store the training error

$$e = \|X_r - X_{r_f}(:, 1:K)\|. \tag{5.25}$$

We repeat this process for other candidate values for (μ_1, μ_2) and choose the optimal value $(\mu_1^\star, \mu_2^\star)$ as the one that leads to smaller error (5.25). Finally, using $(\mu_1^\star, \mu_2^\star)$, we solve least-squares minimization (5.21) and infer matrices A_r, H_r and B_r.

In the next two sections, we repeat our numerical examples for our two case studies in Section 5.3, the SM model of IEEE bus system and the EN model of IEEE 300. As before, the output is the arithmetic mean of all phase angles. Data is obtain via numerical simulation with the step size $\Delta = 10^{-3}$, and r is obtained based on the relative tolerance of $\text{tol} = 1.5 \times 10^{-4}$ for the singular value decay of the snapshot data X. The only difference is that we solve for the reduced quadratic operators A_r, H_r and B_r using Algorithm 6 where instead of solving the minimization problem in (5.19) with a single random regularization parameter, we solve (5.21) with two optimal regularization parameters.

5.3.3.1 Example 1: IEEE 118 bus

We obtain two sets of data over two training time intervals $T^{(1)} = [0\ 0.5]$ and $T^{(2)} = [0\ 0.7]$. Figures 5.8a and 5.8b depict the singular value decay for the state snapshot X for $T^{(1)} = [0\ 0.5]$ and $T^{(2)} = [0\ 0.7]$, respectively. Based on the figures, we choose, respectively, $r^{(1)} = 12$ and $r^{(2)} = 13$. Let $T_f = [0\ 3]$ and $\beta = 1$ in Algorithm 6. We infer the reduced operators

Algorithm 6 Operator inference with regularization selection for Power Network Models

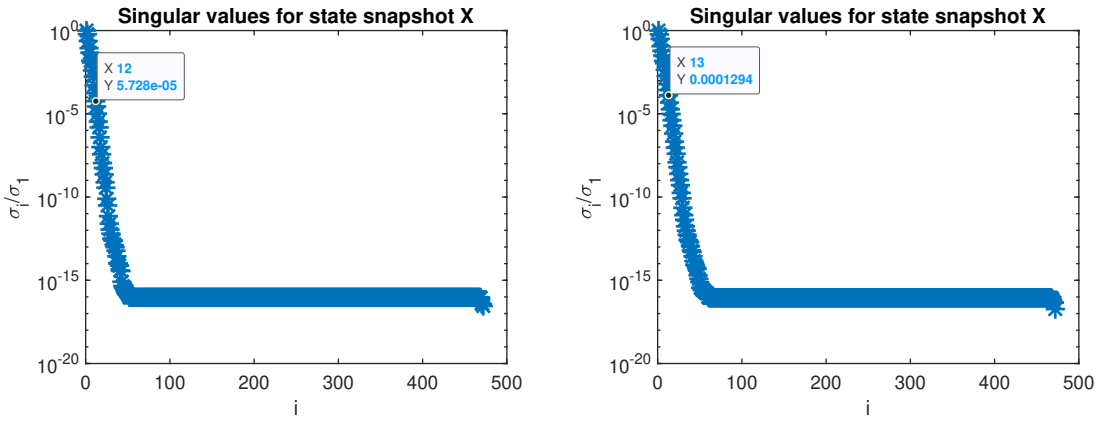
Input: Snapshot data and input trajectory for the original nonlinear model (1.2) over $T = [0\ t_{K-1}]$, β in (5.24)

Output: A_r, H_r and B_r of the predictive quadratic model

- Follow steps (1)-(4) of Algorithm 5.
- Define a desired two-dimensional grid to generate a set including pairs of μ_1 and μ_2, namely S_μ
- For each (μ_1, μ_2) in the set S_μ:
  - Solve (5.21) with regularizer Υ in (5.20) to obtain A_r, H_r and B_r
  - Simulate (5.7) with the quadratic operators obtained in the previous step for $T_f = [0\ t_f]$ and gather the corresponding reduced quadratic snapshot X_{r_f}
  - If $\max_{i,j} |X_{r_f}(i,j)| <= \tau$ compute the training error $\|X_r - X_{r_f}(:, 1:K)\|$ and store it in the vector e_μ
- Find the optimal regularizer $\Upsilon^\star$

$$\Upsilon^\star = \underset{(\mu_1,\mu_2)\in S_\mu}{\arg\min}\ e_\mu$$

- Solve (5.21) with the optimal regularizer $\Upsilon^\star$ and obtain A_r, H_r and B_r



(a) Training time $T^{(1)} = [0\ .5]$ (b) Training time $T^{(2)} = [0\ .7]$

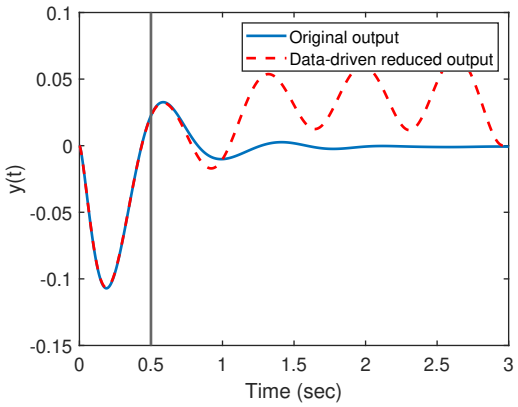
Figure 5.8: Singular values for state snapshot X

$A_r \in \mathbb{R}^{r^{(i)}\times r^{(i)}}$, $H_r \in \mathbb{R}^{r^{(i)}\times (r^{(i)})^2}$, $B_r \in \mathbb{R}^{r^{(i)}}$ for $i = 1, 2$ first with Algorithm 5 combined with the regularization tuning parameter $\mu = 10^{-3}$, and second using Algorithm 6 where S_μ includes 20×20 combinations of $(\mu_1, \mu_2) \in [10^{-9}, 10^{-6}] \times [10^{-5}, 10^{-1}]$. The regularization selection strategy in Algorithm 6 yields the optimal regularization parameters for $T^{(1)} = [0\ 0.5]$ and $T^{(2)} = [0\ 0.7]$ as $(\mu_1, \mu_2) = (10^{-6}, 10^{-3})$ and $(\mu_1, \mu_2) = (6.158 \times 10^{-9}, 2.636 \times 10^{-5})$, respectively. To investigate the effectiveness of Algorithm 6 in generating a predictive reduced quadratic model, we plot the obtained reduced quadratic output $y_r(t)$ and relative output error over $T_f = [0\ 3]$ in Figures 5.9 and 5.10, respectively. According to these figures, we gain the following insights: First, the inferred reduced quadratic models obtained by Algorithm 5 combined with the $\mathcal{L}_2$ regularizer with $\mu = 10^{-3}$ are able to reconstruct the training data but have a poor performance in the prediction regime. Second, by using Algorithm 6, the inferred reduced quadratic models not only capture the output throughout the training time, but also have a comparatively better performance in approximation the original model in the prediction phase (the relative $\mathcal{L}_\infty$ error $\|e\|_{\mathcal{L}_\infty(T_f)}$ has decreased from 20.33% in 5.9c (5.10c) to 3.067% in 5.9d (5.10d)). Although $\|e\|_{\mathcal{L}_\infty(T_f)}$ has decreased from 62.53% in 5.9a (5.10a) to 10.93% in 5.9b (5.10b), the inferred model struggles to appropriately

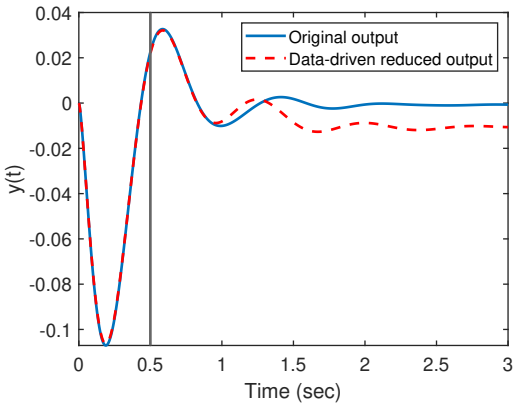
predict the output beyond 1.25 seconds which indicates the importance of training time on the behavior in the prediction period. Once again, comparing Figures 5.9b (5.10b) and 5.9d (5.10d), one can observe that $\|e\|_{\mathcal{L}_\infty(T_f)}$ has decreased from 10.93% to 3.067% which implies that as the training time increases, the inferred models prediction power increases.

5.3.3.2 Example 2: IEEE 300

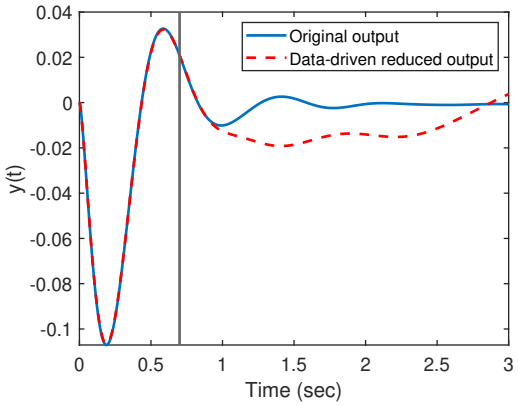
Once again, we obtain two sets of data over two training time intervals $T^{(1)} = [0\ 2.1]$ and $T^{(2)} = [0\ 2.5]$. Based on the singular value decay for the state snapshot X depicted in Figure 5.11a for $T^{(1)}$ and Figure 5.11b for $T^{(2)}$, we choose, respectively, $r^{(1)} = 23$ and $r^{(2)} = 25$. Then, we infer the reduced operators using both Algorithm 5 with $\mathcal{L}_2$ regularization parameter $\mu = 10^{-3}$, and Algorithm 6. Let $T_f = [0\ 8]$ and $\beta = 1$. The regularization selection approach in Algorithm 6 leads to the optimal regularization parameters for $T^{(1)}$ as $(\mu_1, \mu_2) = (10^{-12}, 1.291 \times 10^{-4})$ among 10×10 combinations of $(\mu_1, \mu_2) \in [10^{-12}, 10^{-11}] \times [10^{-5}, 10^{-3}]$, and for $T^{(2)}$ as $(\mu_1, \mu_2) = (10^{-3}, 10^{-2})$ among 10×10 combinations of $(\mu_1, \mu_2) \in [10^{-4}, 10^{-3}] \times [10^{-3}, 10^{-2}]$. Figures 5.9 and 5.10 illustrate the reduced quadratic output $y_r(t)$ and relative output error over $T_f = [0\ 8]$, respectively. According to Figures 5.12a and 5.13c, we observe that by increasing the training time (here by 0.4 seconds) even by using Algorithm 5 and a single $\mathcal{L}_2$ regularization parameter, one can improve the performance of the reduced inferred model beyond the training time. To have a more detailed comparison, according to Figures 5.13a and 5.13c, the relative $\mathcal{L}_\infty$ error $\|e\|_{\mathcal{L}_\infty(T_f)}$ has decreased from 10.27% to 3.7%. As Figures 5.13c and 5.13d illustrate, $\|e\|_{\mathcal{L}_\infty(T_f)}$ has decreased from 3.7% to 2.7%. Similarly, by observing Figures 5.13a and 5.13b, one can see a decrease in $\|e\|_{\mathcal{L}_\infty(T_f)}$ from 10.27% to 5.12%. These observations support the earlier observation in the previous example that the Algorithm 6 yields a better approximation of the original model in the prediction regime.



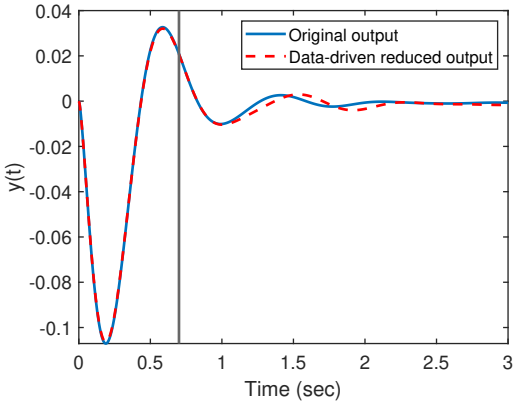
(a) $\mu = 10^{-3}$ and $r^{(1)} = 12$



(b) $(\mu_1, \mu_2) = (10^{-6}, 10^{-3})$ and $r^{(1)} = 12$

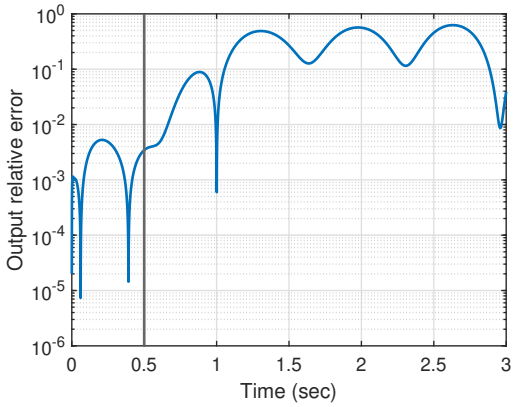


(c) $\mu = 10^{-3}$ and $r^{(2)} = 13$

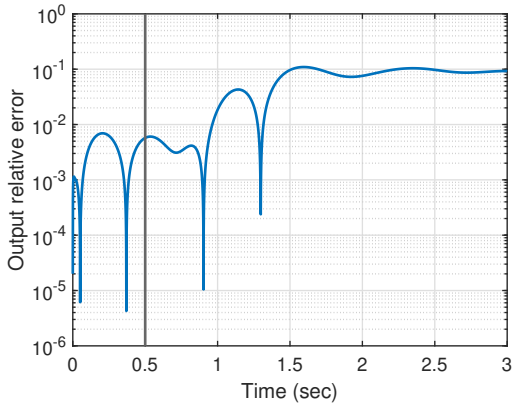


(d) $(\mu_1, \mu_2) = (6.158 \times 10^{-9}, 2.636 \times 10^{-5})$ and $r^{(2)} = 13$

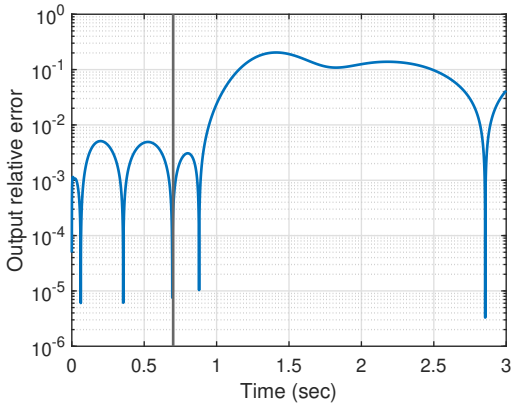
Figure 5.9: Comparison of original output and the data-driven reduced output. The training and prediction periods are separated by the vertical black lines.



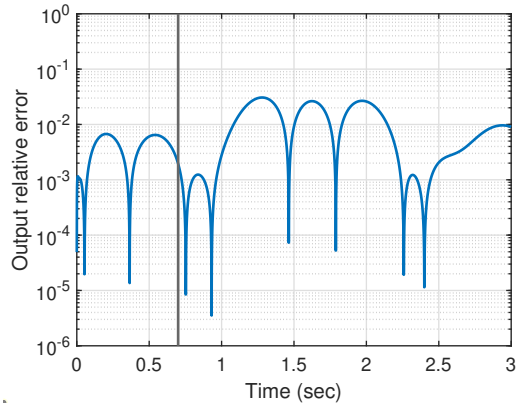
(a) $\mu = 10^{-3}$ and $r^{(1)} = 12$



(b) $(\mu_1, \mu_2) = (10^{-6}, 10^{-3})$ and $r^{(1)} = 12$

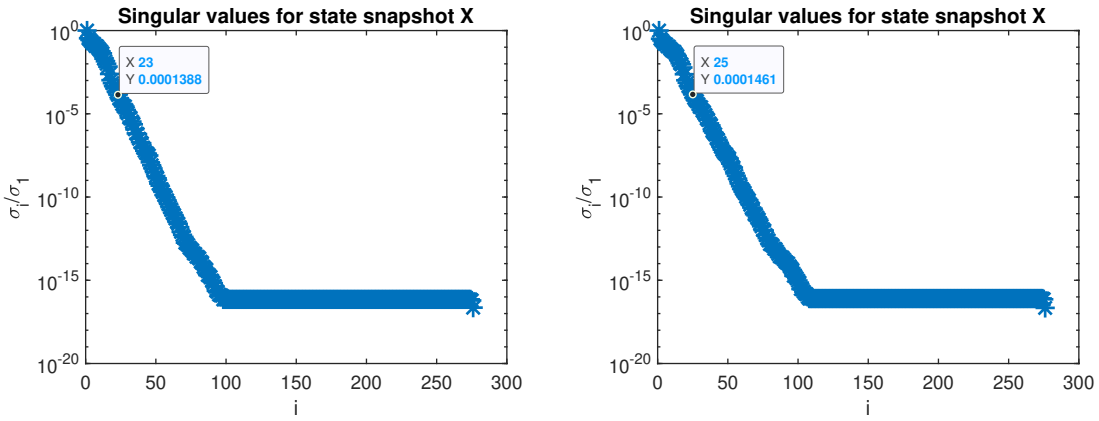


(c) $\mu = 10^{-3}$ and $r^{(2)} = 13$



(d) $(\mu_1, \mu_2) = (6.158 \times 10^{-9}, 2.636 \times 10^{-5})$ and $r^{(2)} = 13$

Figure 5.10: Relative $\mathcal{L}_\infty$ error vs. time

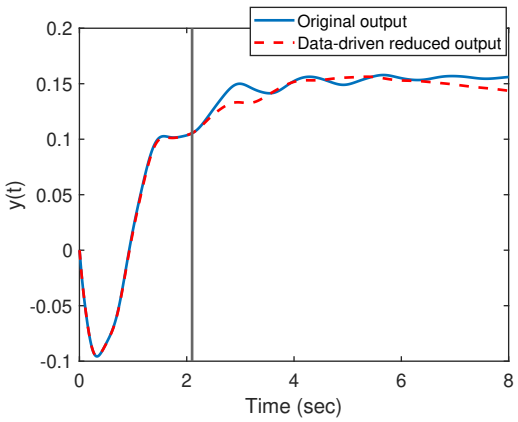


(a) Training time $T^{(1)} = [0\ 2.1]$ (b) Training time $T^{(2)} = [0\ 2.5]$

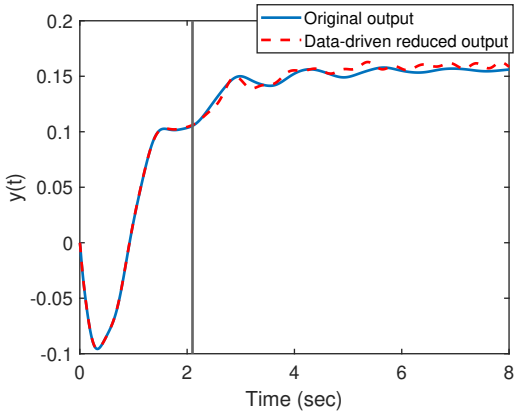
Figure 5.11: Singular values for state snapshot X

5.4 Summary

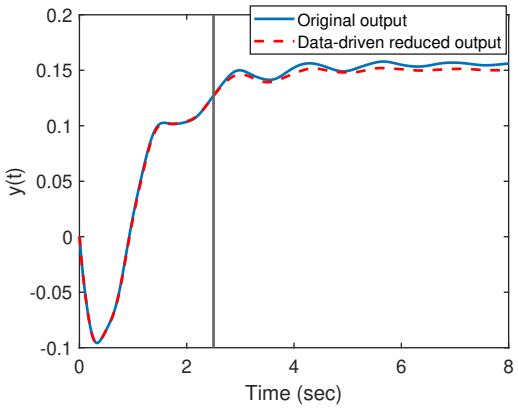
We have illustrated the application of a data driven model reduction approach, the so called Lift and Learn method, to power grid networks. The non-intrusive nature of this methods enables us to infer a quadratic reduced model for the nonlinear swing equations using time domain data. We also investigated the effect of two regularization strategies on the predictive power of the inferred quadratic reduced models. Two examples have been used to demonstrate the the efficiency of our approach.



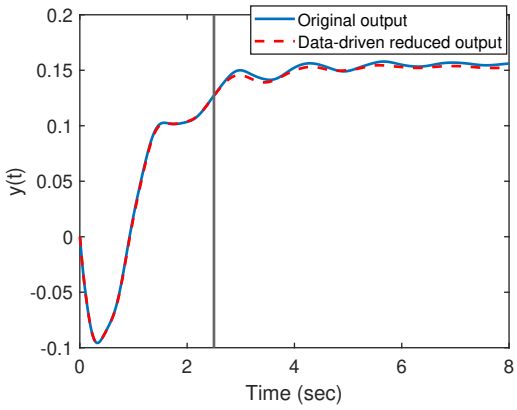
(a) $\mu = 10^{-3}$ and $r^{(1)} = 23$



(b) $(\mu_1, \mu_2) = (10^{-12}, 1.291 \times 10^{-4})$ and $r^{(1)} = 23$

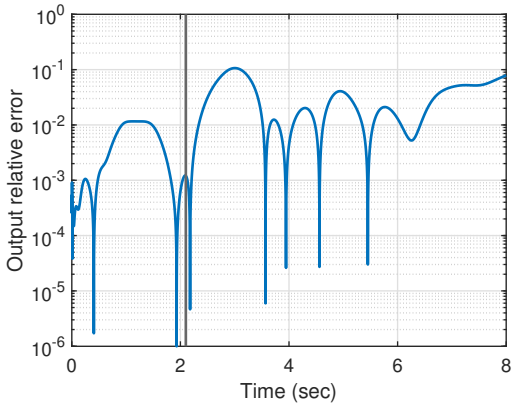


(c) $\mu = 10^{-3}$ and $r^{(2)} = 25$

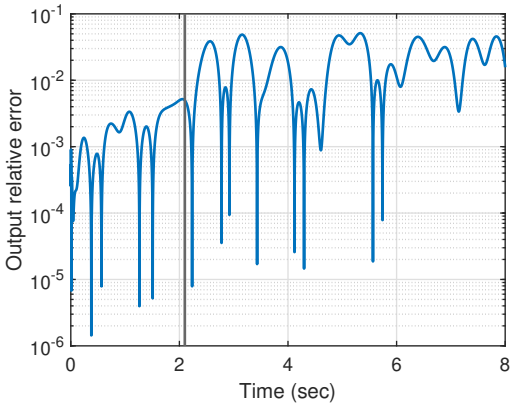


(d) $(\mu_1, \mu_2) = (10^{-3}, 10^{-2})$ and $r^{(2)} = 25$

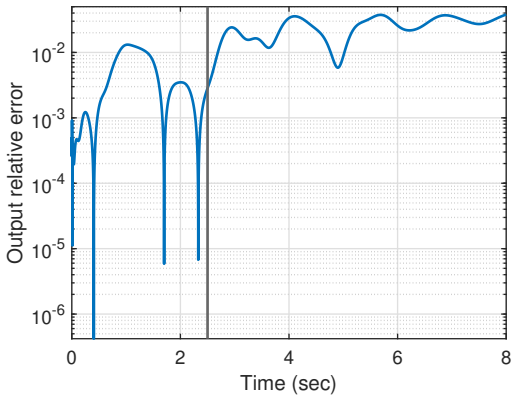
Figure 5.12: Comparison of original output and the data-driven reduced output



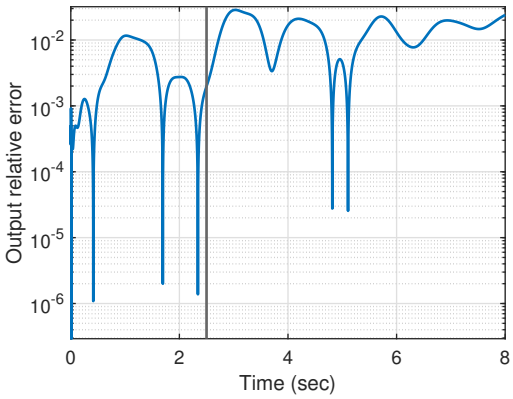
(a) $\mu = 10^{-3}$ and $r^{(1)} = 23$



(b) $(\mu_1, \mu_2) = (10^{-12}, 1.291 \times 10^{-4})$ and $r^{(1)} = 23$



(c) $\mu = 10^{-3}$ and $r^{(2)} = 25$



(d) $(\mu_1, \mu_2) = (10^{-3}, 10^{-2})$ and $r^{(2)} = 25$

Figure 5.13: Relative $\mathcal{L}_\infty$ error vs. time

Chapter 6

Structure-preserving Data-driven Modeling of Power Networks

Dynamical system modeling in power systems has received great attention. There exists a plethora techniques for discovering governing equations from time series data including sparse identification of nonlinear dynamics (SINDY) method [32, 66], dynamic mode decomposition [42, 81, 100, 113, 122, 126], equation-free modeling [52], symbolic regression [114], system identification methods [53, 75], methods based on lifting transformations [78, 102], Koopman operator [107], etc. In power network setting, [111, 116] proposes a data-driven system identification method for synchronous generators using an extended version of the SINDY method. Artificial neural networks [31] and genetic algorithm [130] have been employed for identification of power system load dynamics.

In this chapter, we propose a data-driven system identification method for the second-order nonlinear model (1.1). The method is inspired by the second-order SINDY method [38] where the first-order derivative of the state is also incorporated in the SINDY library function to predict the second-order derivative of the state. Using a special structure of power systems dynamics, we choose the nonlinear functions in the SINDY library of terms, and enforce sparsity in the SINDY output of coefficients. Therefore, there would be no need for using sparsity promoting techniques such as $\mathcal{L}_1$ regularization [120] or sequential threshold least square algorithm [32].

6.1 Problem formulation

Recall (1.1), which we repeat here

$$\frac{2J_i}{\omega_R}\ddot{\delta}_i + \frac{D_i}{\omega_R}\dot{\delta}_i + \sum_{\substack{j=1\\ j\neq i}}^{n} K_{ij}\sin(\delta_i - \delta_j - \gamma_{ij}) = B_i, \quad i = 1,\dots,n. \tag{1.1}$$

Let $m_i = \frac{2J_i}{\omega_R}$ and $d_i = \frac{D_i}{\omega_R}$. Using (3.1), rewrite (1.1), $i = 1,\dots,n$, as

$$\begin{aligned}\ddot{\delta}_i =& -\frac{d_i}{m_i}\dot{\delta}_i - \sum_{\substack{j=1\\ j\neq i}}^{n}\frac{K_{ij}}{m_i}\Big((\sin(\delta_i-\delta_j)\cos(\gamma_{ij}) - \cos(\delta_i-\delta_j)\sin(\gamma_{ij})\Big) + \frac{B_i}{m_i}\\ =& -\frac{d_i}{m_i}\dot{\delta}_i - \sum_{\substack{j=1\\ j\neq i}}^{n}\frac{K_{ij}\cos(\gamma_{ij})}{m_i}\sin(\delta_i-\delta_j) + \sum_{\substack{j=1\\ j\neq i}}^{n}\frac{K_{ij}\sin(\gamma_{ij})}{m_i}\cos(\delta_i-\delta_j) + \frac{B_i}{m_i}.\end{aligned} \tag{6.1}$$

As an example, let $n = 3$. Then, (6.1) becomes

$$\begin{aligned}\ddot{\delta}_1 =& -\frac{d_1}{m_1}\dot{\delta}_1 - \frac{K_{12}\cos(\gamma_{12})}{m_1}\sin(\delta_1-\delta_2) + \frac{K_{12}\sin(\gamma_{12})}{m_1}\cos(\delta_1-\delta_2)\\ &- \frac{K_{13}\cos(\gamma_{13})}{m_1}\sin(\delta_1-\delta_3) + \frac{K_{13}\sin(\gamma_{13})}{m_1}\cos(\delta_1-\delta_3) + \frac{B_1}{m_1}\end{aligned} \tag{6.2}$$

$$\begin{aligned}\ddot{\delta}_2 =& -\frac{d_2}{m_2}\dot{\delta}_2 - \frac{K_{12}\cos(\gamma_{12})}{m_2}\sin(\delta_2-\delta_1) + \frac{K_{12}\sin(\gamma_{12})}{m_2}\cos(\delta_2-\delta_1)\\ &- \frac{K_{23}\cos(\gamma_{23})}{m_2}\sin(\delta_2-\delta_3) + \frac{K_{23}\sin(\gamma_{23})}{m_2}\cos(\delta_2-\delta_3) + \frac{B_2}{m_2}\end{aligned} \tag{6.3}$$

$$\begin{aligned}\ddot{\delta}_3 =& -\frac{d_3}{m_3}\dot{\delta}_3 - \frac{K_{13}\cos(\gamma_{13})}{m_3}\sin(\delta_3-\delta_1) + \frac{K_{13}\sin(\gamma_{13})}{m_3}\cos(\delta_3-\delta_1)\\ &- \frac{K_{23}\cos(\gamma_{23})}{m_3}\sin(\delta_3-\delta_2) + \frac{K_{23}\sin(\gamma_{23})}{m_3}\cos(\delta_3-\delta_2) + \frac{B_3}{m_3}\ .\end{aligned} \tag{6.4}$$

Recall (1.2). For this simple example, our goal is to identify the coefficients on the right hand side of (6.2)-(6.4) shown in Table 6.1 from the data trajectories of δ, $\dot{\delta}$, $\ddot{\delta}$, and input

u.

Table 6.1: Coefficients to be identified

$\ddot{\delta}_1$	$\ddot{\delta}_2$	$\ddot{\delta}_3$
$\frac{B_1}{m_1}$	$\frac{B_2}{m_2}$	$\frac{B_3}{m_3}$
$-\frac{d_1}{m_1}$	$-\frac{d_2}{m_2}$	$-\frac{d_3}{m_3}$
$-\frac{K_{12}\cos(\gamma_{12})}{m_1}$	$\frac{K_{12}\cos(\gamma_{12})}{m_2}$	$\frac{K_{13}\cos(\gamma_{12})}{m_3}$
$\frac{K_{12}\sin(\gamma_{12})}{m_1}$	$\frac{K_{12}\sin(\gamma_{12})}{m_2}$	$\frac{K_{13}\sin(\gamma_{12})}{m_3}$
$-\frac{K_{13}\cos(\gamma_{12})}{m_1}$	$-\frac{K_{23}\cos(\gamma_{23})}{m_2}$	$\frac{K_{23}\cos(\gamma_{23})}{m_3}$
$\frac{K_{13}\sin(\gamma_{12})}{m_1}$	$\frac{K_{23}\sin(\gamma_{23})}{m_2}$	$\frac{K_{23}\sin(\gamma_{23})}{m_3}$

Towards this goal, let $\delta(t) = [\delta_1(t)\ \delta_2(t)\ \ldots,\ \delta_n(t)]^T \in \mathbb{R}^n$ denote the state at time t. We collect K state snapshot data $\delta(t_k)$, its first and second time derivatives ($\dot{\delta}(t_k)$ and $\ddot{\delta}(t_k)$), and input trajectory data $u(t_k)$ at the time samples t_k for $k = 0, \ldots, K-1$, and form the corresponding snapshots, respectively, as

$$\Delta = \begin{bmatrix} \delta^T(t_0) \\ \delta^T(t_1) \\ \vdots \\ \delta^T(t_{K-1}) \end{bmatrix} = \begin{bmatrix} \delta_1(t_0) & \delta_2(t_0) & \ldots & \delta_n(t_0) \\ \delta_1(t_1) & \delta_2(t_1) & \ldots & \delta_n(t_1) \\ \vdots & \vdots & \ldots & \vdots \\ \delta_1(t_{K-1}) & \delta_2(t_{K-1}) & \ldots & \delta_n(t_{K-1}) \end{bmatrix} \in \mathbb{R}^{K\times n}, \tag{6.5}$$

$$\dot{\Delta} = \begin{bmatrix} \dot{\delta}^T(t_0) \\ \dot{\delta}^T(t_1) \\ \vdots \\ \dot{\delta}^T(t_{K-1}) \end{bmatrix} = \begin{bmatrix} \dot{\delta}_1(t_0) & \dot{\delta}_2(t_0) & \ldots & \dot{\delta}_n(t_0) \\ \dot{\delta}_1(t_1) & \dot{\delta}_2(t_1) & \ldots & \dot{\delta}_n(t_1) \\ \vdots & \vdots & \ldots & \vdots \\ \dot{\delta}_1(t_{K-1}) & \dot{\delta}_2(t_{K-1}) & \ldots & \dot{\delta}_n(t_{K-1}) \end{bmatrix} \in \mathbb{R}^{K\times n}, \tag{6.6}$$

$$\ddot{\Delta} = \begin{bmatrix} \ddot{\delta}^T(t_0) \\ \ddot{\delta}^T(t_1) \\ \vdots \\ \ddot{\delta}^T(t_{K-1}) \end{bmatrix} = \begin{bmatrix} \ddot{\delta}_1(t_0) & \ddot{\delta}_2(t_0) & \dots & \ddot{\delta}_n(t_0) \\ \ddot{\delta}_1(t_1) & \ddot{\delta}_2(t_1) & \dots & \ddot{\delta}_n(t_1) \\ \vdots & \vdots & \dots & \vdots \\ \ddot{\delta}_1(t_{K-1}) & \ddot{\delta}_2(t_{K-1}) & \dots & \ddot{\delta}_n(t_{K-1}) \end{bmatrix} \in \mathbb{R}^{K\times n}, \tag{6.7}$$

and

$$U = \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix} \in \mathbb{R}^{1\times K}. \tag{5.15}$$

The right hand side of (6.1) does not explicitly contain δ while it includes sin and cos terms of δ. Knowing the facts that $\sin(\delta_i - \delta_j) = -\sin(\delta_j - \delta_i)$ and $\cos(\delta_i - \delta_j) = \cos(\delta_j - \delta_i)$, we form another snapshot matrix, namely Δ^{sc} using the state snapshot Δ. To simplify the notation, let

$$\sin\left(\delta_i(t_k) - \delta_j(t_k)\right) = s_{ij}(t_k)$$
$$\text{and } \cos\left(\delta_i(t_k) - \delta_j(t_k)\right) = c_{ij}(t_k).$$

Then, define

$$\Delta^{sc} = \begin{bmatrix} \Delta_1^{sc} & \Delta_2^{sc} & \dots & \Delta_{n-1}^{sc} \end{bmatrix} \in \mathbb{R}^{K\times n(n-1)}, \tag{6.8}$$

where, for $i = 1, \dots, n-1$,

$$\Delta_i^{sc} = \begin{bmatrix} s_{i(i+1)}(t_0) & c_{i(i+1)}(t_0) & \dots & s_{in}(t_0) & c_{in}(t_0) \\ s_{i(i+1)}(t_1) & c_{i(i+1)}(t_1) & \dots & s_{in}(t_1) & c_{in}(t_1) \\ \vdots & \vdots & \dots & \vdots & \\ s_{i(i+1)}(t_{K-1}) & c_{i(i+1)}(t_{K-1}) & \dots & s_{in}(t_{K-1}) & c_{in}(t_{K-1}) \end{bmatrix}. \tag{6.9}$$

Next, using (6.6), (5.15) and (6.8), we construct the *library function* $\Theta(U, \dot{\Delta}, \Delta)$

$$\Theta(U, \dot{\Delta}, \Delta) = \begin{bmatrix} U^T & \dot{\Delta} & \Delta^{sc} \end{bmatrix} \in \mathbb{R}^{K \times (1+n^2)}, \tag{6.10}$$

where $K \gg 1 + n^2$. Now, we can write the dynamic model (6.1) in the form of

$$\ddot{\Delta} = \Theta(U, \dot{\Delta}, \Delta)\,(\Gamma \circ \Xi), \tag{6.11}$$

where Ξ is a sparse coefficients matrix that includes the unknown coefficients on the right hand-side of (6.1) and is defined as

$$\Xi = \begin{bmatrix} \xi_1 & \xi_2 & \dots & \xi_n \end{bmatrix} \in \mathbb{R}^{(1+n^2)\times n} \quad \text{with} \quad \xi_i = \begin{bmatrix} \xi_{i_1} \\ \xi_{i_2} \\ \vdots \\ \xi_{i_{1+n^2}} \end{bmatrix} \in \mathbb{R}^{(1+n^2)}. \tag{6.12}$$

Note that each vector ξ_i includes the coefficients on the right hand side of $\ddot{\delta}_i$ for $i = 1, \dots, n$. The matrix $\Gamma \in \mathbb{R}^{(1+n^2)}$ is a matrix of zeros and ones that enforces a special form of sparsity in Ξ. For instance, for the simple example (6.2)-(6.4) with $n = 3$, Γ becomes

$$\Gamma = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}^T \in \mathbb{R}^{10\times 3},$$

which creates sparsity in Ξ as

$$\Xi = \begin{bmatrix} \xi_1 & \xi_2 & \xi_3 \end{bmatrix} = \begin{bmatrix} \xi_{1_1} & \xi_{1_2} & 0 & 0 & \xi_{1_5} & \xi_{1_6} & \xi_{1_7} & \xi_{1_8} & 0 & 0 \\ \xi_{2_1} & 0 & \xi_{2_3} & 0 & \xi_{2_5} & \xi_{2_6} & 0 & 0 & \xi_{2_9} & \xi_{2_{10}} \\ \xi_{3_1} & 0 & 0 & \xi_{3_4} & 0 & 0 & \xi_{3_7} & \xi_{3_8} & \xi_{3_9} & \xi_{3_{10}} \end{bmatrix}^T \in \mathbb{R}^{10\times 3}.$$

The over-determined problem (6.11), can be solved via the least-squares problem

$$\min_{\Xi\in\mathbb{R}^{(1+n^2)\times n}} \left\| \ddot{\Delta} - \Theta(U, \dot{\Delta}, \Delta)\,(\Gamma \circ \Xi) \right\|_F^2. \tag{6.13}$$

Then, the non-zero entries of each column of Ξ in (6.11) (vector ξ_i) can be calculated using a separate sparse regression algorithm as

$$\xi_i(\kappa_i) = \underset{\xi_i(\kappa_i)}{\arg\min} \left\| \ddot{\Delta}(:, i) - \Big(\Theta(U, \dot{\Delta}, \Delta)(:, \kappa_i)\Big)\xi_i(\kappa_i) \right\|_2^2 \quad \text{for} \quad i = 1, \ldots, n, \tag{6.14}$$

where κ_i represents an array of logical values for the ith column of Γ. Once the sparse matrix Ξ is determined, we can form (6.1)

$$\ddot{\delta}_i(t) = \theta(u, \dot{\delta}, \delta)\,\xi_i\ , \qquad \text{for} \quad i = 1, \ldots, n, \tag{6.15}$$

with

$$\theta(u, \dot{\delta}, \delta) = \begin{bmatrix} 1 & \dot{\delta}^T(t) & \delta_1^{sc}(t) & \delta_2^{sc}(t) & \ldots & \delta_{n-1}^{sc}(t) \end{bmatrix} \in \mathbb{R}^{1\times(1+n^2)}, \tag{6.16}$$

where for $i = 1, \ldots, n-1$

$$\delta_i^{sc}(t) = \begin{bmatrix} \sin(\delta_i(t) - \delta_{i+1}(t)) & \cos(\delta_i(t) - \delta_{i+1}(t)) & \ldots & \sin(\delta_i(t) - \delta_n(t)) & \cos(\delta_i(t) - \delta_n(t)) \end{bmatrix}.$$

Note that $\theta(u, \dot{\delta}, \delta)$ at $t = t_k$ corresponds to the $(k+1)$th row of $\Theta(U, \dot{\Delta}, \Delta)$ for $k = 0, \ldots, K-1$. A brief sketch of our approach is given in Algorithm 7.

Algorithm 7 Identification of Power network models

- Collect K snapshot of the state snapshot Δ in (6.5), the first and second derivatives of the state snapshot Δ in (6.6) and (6.7) and the input trajectory as in (5.15).
- Construct the nonlinear snapshot Δ^{sc} in (6.8).
- Construct the library function $\Theta(U, \Delta, \dot{\Delta})$ in (6.10)

$$\Theta(U, \dot{\Delta}, \Delta) = \begin{bmatrix} U^T & \dot{\Delta} & \Delta^{sc} \end{bmatrix}. \tag{6.10}$$

- Solve the sparse regression algorithm

$$\xi_i(\kappa_i) = \underset{\xi_i(\kappa_i)}{\arg\min} \left\| \ddot{\Delta}(:, i) - \Big(\Theta(U, \dot{\Delta}, \Delta)(:, \kappa_i)\Big)\xi_i(\kappa_i) \right\|_2^2 \quad \text{for} \quad i = 1, \ldots, n. \tag{6.14}$$

- Using ξ_i and (6.16), form (6.1)

$$\ddot{\delta}_i(t) = \theta(u, \dot{\delta}, \delta)\, \xi_i \ , \quad \text{for} \quad i = 1, \ldots, n \tag{6.15}$$

To summarize our results, we present the following lemma.

Lemma 6.1. *Given the nonlinear state snapshot data Δ^{sc} in (6.8), $\dot{\Delta}$ in (6.6), $\ddot{\Delta}$ in (6.7), input snapshot U in (5.15), we can identify the coefficients on the right hand side of (6.1) as*

$$\xi_i(\kappa_i) = \underset{\xi_i(\kappa_i)}{\arg\min} \left\| \ddot{\Delta}(:, i) - \Big(\Theta(U, \dot{\Delta}, \Delta)(:, \kappa_i)\Big)\xi_i(\kappa_i) \right\|_2^2 \quad \textit{for} \quad i = 1, \ldots, n \tag{6.14}$$

and form (6.1) for $i = 1, \ldots, n$ as

$$\ddot{\delta}_i(t) = \theta(u, \dot{\delta}, \delta)\, \xi_i \ , \tag{6.15}$$

where $\theta(u, \dot{\delta}, \delta)$ is defined in (6.16).

6.2 Numerical examples

The two test systems we investigate are the simple EN model of "test-system-3gen" with $n = 3$ and the SM model of the New England test system with $n = 39$, provided by MATPOWER and MATLAB toolbox `pg-sync-models` [90] . We focus on the single-output system with $y(t)$ chosen as the arithmetic mean of all phase angles $\delta(t)$. In both cases, we obtain the data via a numerical simulation with the time step size $\Delta t = 10^{-3}$. The second order derivative (6.7) is calculated from (6.6) by using the second-order forward difference approximation.

6.2.1 Example 1: 3-generator system

In this example, we have a model of dimension $n = 3$ as in (6.2)-(6.4). Using data, our aim is to identify the coefficients shown in Table 6.1 from the data. We collect K snapshot of the state snapshot $\Delta \in \mathbb{R}^{K\times 3}$ as in (6.5):

$$\Delta = \begin{bmatrix} \delta_1(t_0) & \delta_2(t_0) & \delta_3(t_0) \\ \delta_1(t_1) & \delta_2(t_1) & \delta_3(t_1) \\ \vdots & \vdots & \vdots \\ \delta_1(t_{K-1}) & \delta_2(t_{K-1}) & \delta_3(t_{K-1}) \end{bmatrix} \in \mathbb{R}^{K\times 3}$$

and form the nonlinear snapshot Δ^{sc} as in (6.8):

$$\Delta^{sc} = \begin{bmatrix} \Delta_1^{sc} & \Delta_2^{sc} \end{bmatrix}$$
$$= \begin{bmatrix} s_{12}(t_0) & c_{12}(t_0) & s_{13}(t_0) & c_{13}(t_0) & s_{23}(t_0) & c_{23}(t_0) \\ s_{12}(t_1) & c_{12}(t_1) & s_{13}(t_1) & c_{13}(t_1) & s_{23}(t_1) & c_{23}(t_1) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \\ s_{12}(t_{K-1}) & c_{12}(t_{K-1}) & s_{13}(t_{K-1}) & c_{13}(t_{K-1}) & s_{23}(t_{K-1}) & c_{23}(t_{K-1}) \end{bmatrix} \in \mathbb{R}^{K\times 6}.$$

Also, we collect K snapshot of the first and second derivatives of the state snapshot Δ as in (6.6) and (6.7):

$$\dot{\Delta} = \begin{bmatrix} \dot{\delta}_1(t_0) & \dot{\delta}_2(t_0) & \dot{\delta}_3(t_0) \\ \dot{\delta}_1(t_1) & \dot{\delta}_2(t_1) & \dot{\delta}_3(t_1) \\ \vdots & \vdots & \vdots \\ \dot{\delta}_1(t_{K-1}) & \dot{\delta}_2(t_{K-1}) & \dot{\delta}_3(t_{K-1}) \end{bmatrix} \in \mathbb{R}^{K\times 3}, \quad \ddot{\Delta} = \begin{bmatrix} \ddot{\delta}_1(t_0) & \ddot{\delta}_2(t_0) & \ddot{\delta}_3(t_0) \\ \ddot{\delta}_1(t_1) & \ddot{\delta}_2(t_1) & \ddot{\delta}_3(t_1) \\ \vdots & \vdots & \vdots \\ \ddot{\delta}_1(t_{K-1}) & \ddot{\delta}_2(t_{K-1}) & \ddot{\delta}_3(t_{K-1}) \end{bmatrix} \in \mathbb{R}^{K\times 3}$$

and input trajectory as in (5.15):

$$U^T = \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix} = \mathbf{1}_K \in \mathbb{R}^K.$$

Following Algorithm (6.14), we form the library function

$$\Theta(U, \dot{\Delta}, \Delta) = \begin{bmatrix} \mathbf{1}_K & \dot{\Delta} & \Delta^{sc} \end{bmatrix} \in \mathbb{R}^{K\times 10}$$

and solve for

$$\Xi = \begin{bmatrix} \xi_1 & \xi_2 & \xi_3 \end{bmatrix} \in \mathbb{R}^{10\times 3} \quad \text{with} \quad \xi_i \in \mathbb{R}^{10\times 1}$$

with $\Gamma \in \mathbb{R}^{10\times 3}$ formed as

$$\Gamma = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}^T .$$

Note that each vector $\xi_i \in \mathbb{R}^{10\times 1}$ contains the terms in the right hand side of $\ddot{\delta}_i$ for $i = 1, 2, 3$. For example, ξ_1 includes the term in the first column of Table 6.1. We collect the data snapshots for the time interval $T = [0\ .5]$ seconds. With a step size of $\Delta t = 10^{-3}$ this leads to the snapshot matrix $\Delta \in \mathbb{R}^{501\times 3}$. The true and identified values for Ξ are shown in Tables 6.2 and 6.3, respectively.

Table 6.2: True value for Ξ

	ξ_1	ξ_2	ξ_3
1	$\frac{B_1}{m_1} = -1.8145$	$\frac{B_2}{m_2} = 34.3647$	$\frac{B_3}{m_3} = 35.2895$
$\dot{\delta}_1$	$-\frac{d_1}{m_1} = -1.0575$	0	0
$\dot{\delta}_2$	0	$-\frac{d_2}{m_2} = -3.9063$	0
$\dot{\delta}_3$	0	0	$-\frac{d_3}{m_3} = -8.3056$
$\sin(\delta_1 - \delta_2)$	$-\frac{K_{12}\cos(\gamma_{12})}{m_1} = -13.3868$	$\frac{K_{12}\cos(\gamma_{12})}{m_2} = 49.4474$	0
$\cos(\delta_1 - \delta_2)$	$\frac{K_{12}\sin(\gamma_{12})}{m_1} = -2.5404$	$\frac{K_{12}\sin(\gamma_{12})}{m_2} = -9.3836$	0
$\sin(\delta_1 - \delta_3)$	$-\frac{K_{13}\cos(\gamma_{12})}{m_1} = -10.5013$	0	$\frac{K_{13}\cos(\gamma_{12})}{m_3} = 82.4752$
$\cos(\delta_1 - \delta_3)$	$\frac{K_{13}\sin(\gamma_{12})}{m_1} = -1.7958$	0	$\frac{K_{13}\sin(\gamma_{12})}{m_3} = -14.1042$
$\sin(\delta_2 - \delta_3)$	0	$-\frac{K_{23}\cos(\gamma_{23})}{m_2} = -34.2217$	$\frac{K_{23}\cos(\gamma_{23})}{m_3} = 72.7636$
$\cos(\delta_2 - \delta_3)$	0	$\frac{K_{23}\sin(\gamma_{23})}{m_2} = -6.7086$	$\frac{K_{23}\sin(\gamma_{23})}{m_3} = -14.2641$

As these tables show, the proposed method has efficiently identified Ξ with a high accuracy. According to the obtained Ξ in Table 6.3, we form our dynamics (6.1)

$$\begin{aligned} \ddot{\delta}_i(t) &= \theta(u, \dot{\delta}, \delta)\, \xi_i \\ &= \begin{bmatrix} 1 & \dot{\delta}_1 & \dot{\delta}_2 & \dot{\delta}_3 & s(\delta_{12}) & c(\delta_{12}) & s(\delta_{13}) & c(\delta_{13}) & s(\delta_{23}) & c(\delta_{23}) \end{bmatrix} \xi_i, \quad \text{for } i = 1, 2, 3 \end{aligned}$$

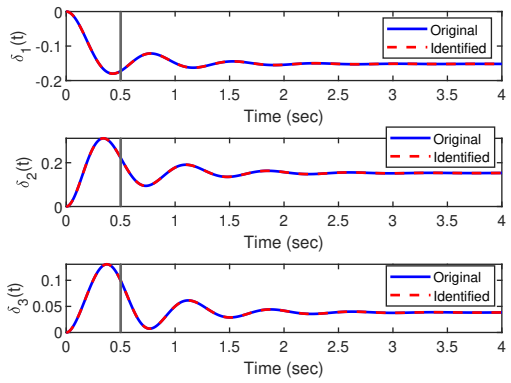
Table 6.3: Identified value for Ξ

	ξ_1	ξ_2	ξ_3
1	$\frac{B_1}{m_1} = -1.8155$	$\frac{B_2}{m_2} = 34.3759$	$\frac{B_3}{m_3} = 35.2787$
$\dot{\delta}_1$	$-\frac{d_1}{m_1} = -1.0576$	0	0
$\dot{\delta}_2$	0	$-\frac{d_2}{m_2} = -3.9064$	0
$\dot{\delta}_3$	0	0	$-\frac{d_3}{m_3} = -8.3057$
$\sin(\delta_1 - \delta_2)$	$-\frac{K_{12}\cos(\gamma_{12})}{m_1} = -13.3869$	$\frac{K_{12}\cos(\gamma_{12})}{m_2} = 49.4484$	0
$\cos(\delta_1 - \delta_2)$	$\frac{K_{12}\sin(\gamma_{12})}{m_1} = -2.5404$	$\frac{K_{12}\sin(\gamma_{12})}{m_2} = -9.3853$	0
$\sin(\delta_1 - \delta_3)$	$-\frac{K_{13}\cos(\gamma_{12})}{m_1} = -10.5018$	0	$\frac{K_{13}\cos(\gamma_{12})}{m_3} = 82.4748$
$\cos(\delta_1 - \delta_3)$	$\frac{K_{13}\sin(\gamma_{12})}{m_1} = -1.7950$	0	$\frac{K_{13}\sin(\gamma_{12})}{m_3} = -14.0976$
$\sin(\delta_2 - \delta_3)$	0	$-\frac{K_{23}\cos(\gamma_{23})}{m_2} = -34.2244$	$\frac{K_{23}\cos(\gamma_{23})}{m_3} = 72.7650$
$\cos(\delta_2 - \delta_3)$	0	$\frac{K_{23}\sin(\gamma_{23})}{m_2} = -6.7176$	$\frac{K_{23}\sin(\gamma_{23})}{m_3} = -14.2599$

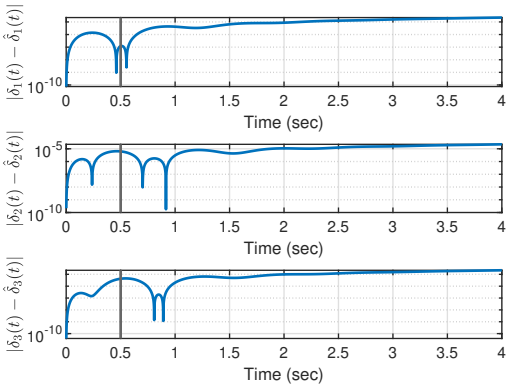
which yields

$$\begin{aligned}
\ddot{\delta}_1 =& -1.0576\dot{\delta}_1 - 13.3869\sin(\delta_1 - \delta_2) - 2.5404\cos(\delta_1 - \delta_2) \\
& -10.5018\sin(\delta_1 - \delta_3) - 1.795\cos(\delta_1 - \delta_3) - 1.8155 \\
\ddot{\delta}_2 =& -3.9064\dot{\delta}_2 + 49.4484\sin(\delta_1 - \delta_2) - 9.3853\cos(\delta_1 - \delta_2) \\
& -34.2244\sin(\delta_2 - \delta_3) - 6.7176\cos(\delta_2 - \delta_3) + 34.3759 \\
\ddot{\delta}_3 =& -8.3057\dot{\delta}_3 + 82.4748\sin(\delta_1 - \delta_3) - 14.0976\cos(\delta_1 - \delta_3) \\
& + 72.765\sin(\delta_2 - \delta_3) - 14.2599\cos(\delta_2 - \delta_3) + 35.2787
\end{aligned}$$

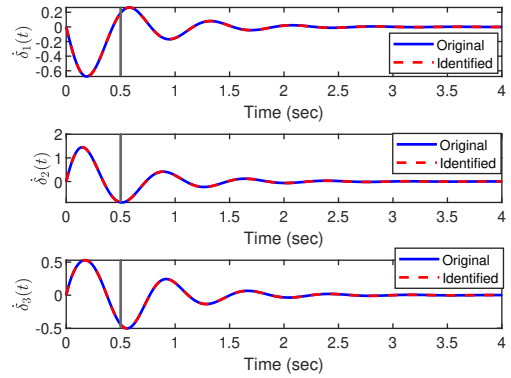
The state δ and its time derivative $\dot{\delta}$ of the original and the identified models and their corresponding absolute error over $T_f = [0\ 4]$ are illustrated in Figure 6.1. According to the figure, the identified model has successfully captured the original model behavior. To have a more detailed comparison, we compare the original model output $y(t)$ with the output of the identified model, namely $\hat{y}(t)$, and the corresponding relative output error in Figure 6.2.



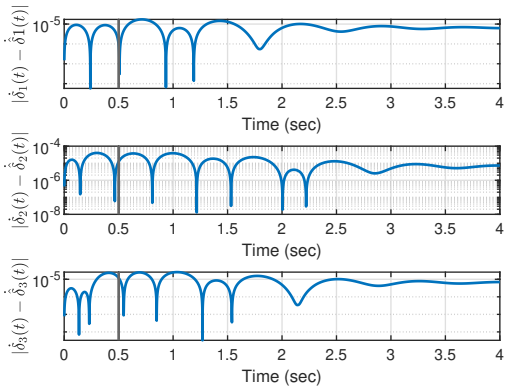
(a) $\delta(t)$ vs. time



(b) Absolute error vs. time

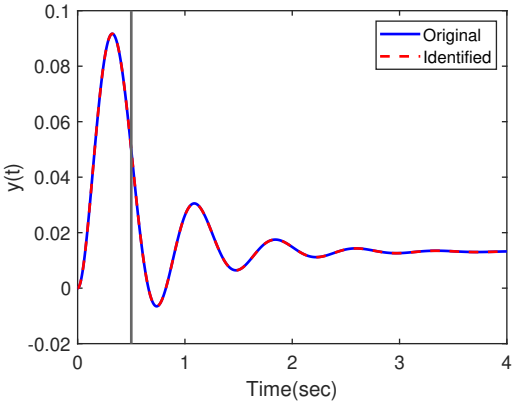


(c) $\dot{\delta}(t)$ vs. time

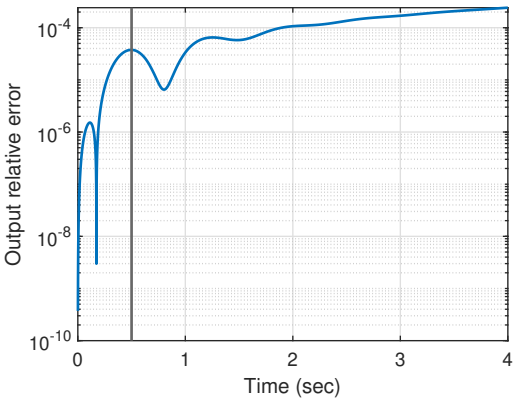


(d) Absolute error vs. time

Figure 6.1: Comparison of the original and the identified system
The training and prediction periods are separated by The vertical black lines.



(a) Output $y(t)$ vs. time



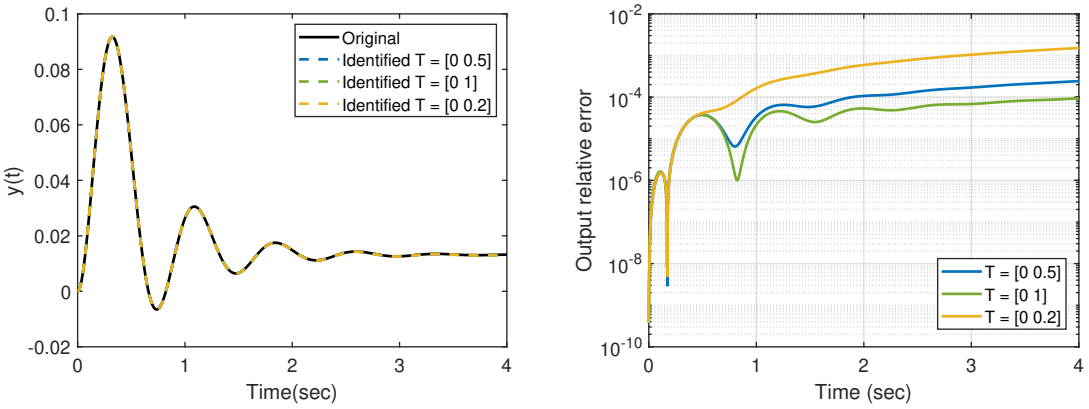
(b) Relative $\mathcal{L}_\infty(T_f)$ error vs. time

Figure 6.2: Comparison of the original and the identified outputs.
The training and prediction periods are separated by The vertical black lines.

According to Figure 6.2, the identified model approximates the original model with a relative $\mathcal{L}_\infty(T_f)$ output error

$$\|e(t)\|_{\mathcal{L}_\infty(T_f)} = \max_{t\in T_f} \frac{|\, y(t) - y_r(t)\,|}{\|y\|_{\mathcal{L}_\infty(T_f)}}, \quad T_f = [0\ 4]$$

less than 0.0243% over the time-interval $T_f = [0\ 4]$ (3.5 seconds beyond the training time). To illustrate whether this approach can avoid overfitting the training time, we plot the original model output $y(t)$ and the output of the identified model $\hat{y}(t)$, and the corresponding relative output for different values of the training time T. As figure shows, even with smaller amount of data ($T = [0\ 0.2]$), one can achieve a relative $\mathcal{L}_\infty(T_f)$ output error less than 0.1513% over $T_f = [0\ 4]$.



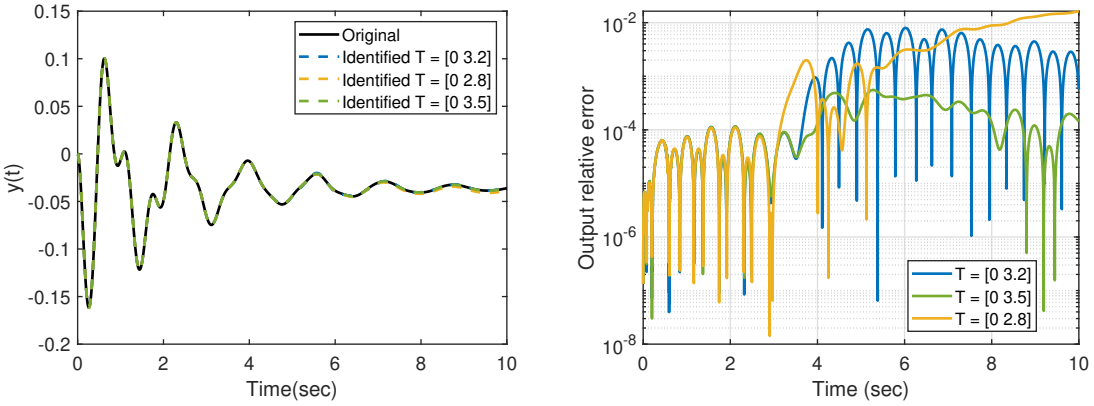
(a) Output $y(t)$ vs. time (b) Relative $\mathcal{L}_\infty(T_f)$ error vs. time

Figure 6.3: Comparison of the original and the identified outputs for different T

6.2.2 Example 2: New England test system

Recall the SM model of New England test system in 4.4 with $n = 39$ in the original second-order coordinates (1.2). We collect the data snapshots Δ, $\dot{\Delta}$, $\ddot{\Delta}$ and U for three time intervals $T = [0\ 2.8]$, $T = [0\ 3.2]$ and $T = [0\ 3.5]$, and follow Algorithm 7 to identify (6.1) for

$i = 1, 2, \ldots, 39$. The original model output $y(t)$ and the output of the identified models for the aforementioned time intervals, and the corresponding relative output error for $T_f = [0\ 10]$ are depicted in Figure 6.4. As Figure 6.4a illustrates, the outputs of the identified models have successfully captured the original model output $y(t)$ even beyond their training time intervals. According to Figure 6.4b, for the smallest time interval $T = [0\ 2.8]$, the relative $\mathcal{L}_\infty(T_f)$ output error $\|e(t)\|_{\mathcal{L}_\infty(T_f)}$ is less than 1.46% for $T_f = [0\ 10]$ (7.2 seconds beyond the training time). This supports our previous observation about the immunity of the proposed method towards overfitting the training time. In addition, we can observe that one can improve the performance of the model by slightly increasing the training time. To be more specific, the $\|e(t)\|_{\mathcal{L}_\infty(T_f)}$ has decreased from 1.46% to 0.0554% when the training time is increased by only 0.7 seconds.



(a) Output $y(t)$ vs. time

(b) Relative $\mathcal{L}_\infty(T_f)$ error vs. time

Figure 6.4: Comparison of the original and the identified outputs for different T

6.3 Summary

Inspired by the second-order SINDY method, we proposed a data-driven system identification method for the second-order nonlinear model (1.1). Exploiting the structure of the swing

equations, we picked the nonlinear functions in the SINDY library of terms. We enforced sparsity in the SINDY output of coefficients by employing a mask matrix. We showed that our proposed method is able to identify accurate models even with small amount of data and seems to be immune to overfitting. We have illustrated the effectiveness of our proposed approach via two numerical examples.

Chapter 7

Conclusions and Outlook

In this dissertation, we have explored model reduction and data-driven modeling for power networks.

As the first major contribution, we developed a structure-preserving parametric model reduction approach for linearized swing equations using a global basis framework and second-order interpolatory $\mathcal{H}_2$ model reduction. We discussed that since the full-order dynamics has a pole at zero with a parametrically varying residue, the parametric reduced model needs to retain this residue in order to have bounded $\mathcal{H}_2$ and $\mathcal{H}_\infty$ error norms for whole parameter domain. Based on a detailed residue analysis, we established the subspace conditions on the model reduction basis to guarantee this property and explain the algorithmic implications. We illustrated the effectiveness of the proposed method through two numerical examples.

As the second main contribution, we designed a structure-preserving and input-independent MOR framework for a special class of power grid networks. We obtained the reduction bases using the idea of lifting the nonlinear dynamics to quadratic models and employing $\mathcal{H}_2$-quasi-optimal model reduction for quadratic systems (Q-IRKA). To be more specific, we obtained the reduction basis for the second-order nonlinear power network grids model with the help of the reduction bases obtained by Q-IRKA and their particular subspace structure. Then, we applied this reduction basis on the second-order nonlinear power network grids model to form a structure-preserving reduced model. We used two numerical example to illustrate the effectiveness of our approach, and compared our results with POD and Q-BT.

As the third major contribution, we developed a non-intrusive data-driven modeling framework for power network dynamics using the Lift and Learn approach. We applied a lifting map to the snapshot data obtained from the original nonlinear swing equations describing the underlying power network such that the lifted data corresponds to quadratic nonlinearity. Then, we projected the lifted data onto a lower dimensional basis and fitted the reduced quadratic matrices to this reduced lifted data via a least-squares measure. We also investigated two regularization approaches. We illustrated the effectiveness of the proposed approach by two power network models.

As our final and fourth contribution, We developed a data-driven system identification method inspired by the second-order SINDY for the second-order nonlinear model (1.1). Unlike the regular SINDY method, we limit the choice of nonlinear functions in the SINDY library of terms based on the structure of the swing equations, and enforced sparsity in the SINDY output matrix of coefficients by employing a mask matrix. We illustrated the effectiveness of the proposed approach via numerical examples.

There are various interesting future directions to pursue. In Chapter 2, we use a global basis approach to construct the reduction base V by concatenating the locally optimal $\mathcal{H}_2$ model reduction bases. That would be interesting to directly construct V via the recent composite $\mathcal{H}_2 \times \mathcal{L}_2$-optimal basis constructions [58, 68, 69]. Extensions to the nonlinear parametric setting is also an important topic to consider.

In Chapter 5, the learned model is a reduced quadratic system and thus does not preserve the original second-order structure of the swing equations. Another interesting research direction to pursue would be learning a reduced-structured model. In addition, the data for our data-driven approach has been obtained via numerical simulation. One can test the robustness of the approach on the noisy real measurements, such as Phasor Measurement Unit data.

Bibliography

[1] B. Anić, C. Beattie, S. Gugercin, and A. C. Antoulas. Interpolatory weighted-h2 model reduction. *Automatica*, 49(5):1275–1280, 2013. ISSN 0005-1098. doi: https://doi.org/10.1016/j.automatica.2013.01.040. URL https://www.sciencedirect.com/science/article/pii/S0005109813000411.

[2] U. D. Annakkage, N. K. C. Nair, Y. Liang, A. M. Gole, V. Dinavahi, B. Gustavsen, T. Noda, H. Ghasemi, A. Monti, M. Matar, R. Iravani, and J. A. Martinez. Dynamic system equivalents: A survey of available techniques. *IEEE Transactions on Power Delivery*, 27(1):411–420, 2012. doi: 10.1109/TPWRD.2011.2167351.

[3] A. C. Antoulas. *Approximation of large-scale dynamical systems*. SIAM, 2005.

[4] A. C. Antoulas, D. C. Sorensen, and S. Gugercin. A survey of model reduction methods for large-scale systems. *Structured Matrices in Operator Theory, Numerical Analysis, Control, Signal and Image Processing, Contemporary Mathematics, AMS publications*, 280:193–219, 2001.

[5] A. C. Antoulas, C. Beattie, and S. Güğercin. *Interpolatory methods for model reduction*. Computational Science and Engineering 21. SIAM, Philadelphia, 2020.

[6] A.C. Antoulas, C.A. Beattie, and S. Gugercin. Interpolatory model reduction of large-scale dynamical systems. In J. Mohammadpour and K. Grigoriadis, editors, *Efficient Modeling and Control of Large-Scale Systems*, pages 2–58. Springer-Verlag, 2010.

[7] A. Astolfi. Model reduction by moment matching for linear and nonlinear systems. *IEEE Transactions on Automatic Control*, 55(10):2321–2336, 2010.

[8] Z. Bai. Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems. *Applied Numerical Mathematics*, 43(1):9 – 44, 2002. ISSN 0168-9274. doi: https://doi.org/10.1016/S0168-9274(02)00116-2. URL http://www.sciencedirect.com/science/article/pii/S0168927402001162. 19th Dundee Biennial Conference on Numerical Analysis.

[9] Z. Bai and Y. Su. Dimension reduction of large-scale second-order dynamical systems via a second-order arnoldi method. *SIAM Journal on Scientific Computing*, 26(5):1692–1709, 2005. doi: 10.1137/040605552. URL https://doi.org/10.1137/040605552.

[10] M. Barrault, N. C. Nguyen, Y. Maday, and A. T. Patera. An empirical interpolation method: Application to efficient reduced-basis discretization of partial differential equations. *C. R. Acad. Sci. Paris, Série I.*, 339:667–672, 2004.

[11] U. Baur, C. Beattie, P. Benner, and S. Gugercin. Interpolatory projection methods for parameterized model reduction. *SIAM Journal on Scientific Computing*, 33(5):2489–2518, 2011. doi: 10.1137/090776925. URL https://doi.org/10.1137/090776925.

[12] U. Baur, P. Benner, and L. Feng. Model order reduction for linear and nonlinear systems: a system-theoretic perspective. *Archives of Computational Methods in Engineering*, 21(4):331–358, 2014.

[13] C. Beattie and S. Gugercin. Interpolatory projection methods for structure-preserving model reduction. *Systems & Control Letters*, 58(3):225 – 232, 2009. ISSN 0167-6911. doi: https://doi.org/10.1016/j.sysconle.2008.10.016. URL http://www.sciencedirect.com/science/article/pii/S0167691108001850.

[14] C. Beattie, S. Gugercin, and V. Mehrmann. Model reduction for systems with inhomogeneous initial conditions. *Systems & Control Letters*, 99:99 – 106, 2017.

ISSN 0167-6911. doi: https://doi.org/10.1016/j.sysconle.2016.11.007. URL http://www.sciencedirect.com/science/article/pii/S0167691116301864.

[15] C.A. Beattie and P. Benner. $\mathcal{H}_2$-optimality conditions for structured dynamical systems. *Preprint MPIMD/14-18, Max Planck Institute Magdeburg, Germany*, 2014.

[16] C.A. Beattie and S. Gugercin. Realization-independent $\mathcal{H}_2$-approximation. In *Proceedings of 51st IEEE Conference on Decision and Control*, pages 4953 – 4958, 2012.

[17] P. Benner and T. Breiten. Two-sided moment matching methods for nonlinear model reduction. Preprint MPIMD/12-12, Max Planck Institute Magdeburg, June 2012. Available from http://www.mpi-magdeburg.mpg.de/preprints/.

[18] P. Benner and T. Breiten. Interpolation-based $\mathcal{H}_2$-model reduction of bilinear control systems. *SIAM Journal on Matrix Analysis and Applications*, 33(3):859–885, 2012. doi: 10.1137/110836742. URL https://doi.org/10.1137/110836742.

[19] P. Benner and T. Breiten. Two-sided projection methods for nonlinear model order reduction. *SIAM Journal on Scientific Computing*, 37(2):B239–B260, 2015. doi: 10.1137/14097255X. URL https://doi.org/10.1137/14097255X.

[20] P. Benner and P. Goyal. Balanced truncation model order reduction for quadratic-bilinear control systems. e-print 1705.00160, arXiv, 2017. URL https://arxiv.org/abs/1705.00160. math.OC.

[21] P. Benner, S. Gugercin, and K. Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Review*, 57(4):483–531, 2015. doi: 10.1137/130932715. URL https://doi.org/10.1137/130932715.

[22] P. Benner, A. Cohen, M. Ohlberger, and K. Willcox. *Model Reduction and Approx-*

imation: Theory and Algorithms. Computational Science and Engineering, SIAM Publications, Philadelphia, PA, 2017.

[23] P. Benner, P. Goyal, and S. Gugercin. $\mathcal{H}_2$-quasi-optimal model order reduction for quadratic-bilinear control systems. *SIAM Journal on Matrix Analysis and Applications*, 39(2):983–1032, 2018.

[24] P. Benner, P. Goyal, B. Kramer, B. Peherstorfer, and K. Willcox. Operator inference for non-intrusive model reduction of systems with non-polynomial nonlinear terms. *Computer Methods in Applied Mechanics and Engineering*, 372:113433, 2020. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2020.113433. URL https://www.sciencedirect.com/science/article/pii/S0045782520306186.

[25] P. Benner, S. Gugercin, and S. W. R. Werner. Structure-preserving interpolation of bilinear control systems. *Advances in Computational Mathematics*, 47(3):1–38, 2021.

[26] P. Benner, S. Gugercin, and S. W. R. Werner. Structure-preserving interpolation for model reduction of parametric bilinear systems. *Automatica*, 132:109799, 2021.

[27] A. R. Bergen and D. J. Hill. A structure preserving model for power system stability analysis. *IEEE Transactions on Power Apparatus and Systems*, PAS-100(1):25–35, 1981.

[28] B. Besselink, U. Tabak, A. Lutowska, N. van de Wouw, H. Nijmeijer, D.J. Rixen, M.E. Hochstenbach, and W.H.A. Schilders. A comparison of model reduction techniques from structural dynamics, numerical mathematics and systems and control. *Journal of Sound and Vibration*, 332(19):4403–4422, 2013. ISSN 0022-460X. doi: https://doi.org/10.1016/j.jsv.2013.03.025. URL https://www.sciencedirect.com/science/article/pii/S0022460X1300285X.

[29] S. Bhela, D. Deka, H. Nagarajan, and V. Kekatos. Designing power grid topologies for minimizing network disturbances: An exact milp formulation. In *2019 American Control Conference (ACC)*, pages 1949–1956, 2019. doi: 10.23919/ACC.2019.8814435.

[30] T. Bonin, H. Faßbender, A. Soppa, and M. Zaeh. A fully adaptive rational global arnoldi method for the model-order reduction of second-order mimo systems with proportional damping. *Mathematics and Computers in Simulation*, 122:1 – 19, 2016. ISSN 0378-4754. doi: https://doi.org/10.1016/j.matcom.2015.08.017. URL http://www.sciencedirect.com/science/article/pii/S0378475415001822.

[31] M. Bostanci, J. Koplowitz, and C.W. Taylor. Identification of power system load dynamics using artificial neural networks. *IEEE Transactions on Power Systems*, 12 (4):1468–1473, 1997. doi: 10.1109/59.627843.

[32] S. L. Brunton, J. L. Proctor, and J. N. Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016. doi: 10.1073/pnas.1517384113. URL https://www.pnas.org/doi/abs/10.1073/pnas.1517384113.

[33] A.E. Bryson Jr and A. Carrier. Second-order algorithm for optimal model order reduction. *Journal of Guidance, Control, and Dynamics*, 13(5):887–892, 1990.

[34] T. Bui-Thanh, K. Willcox, and O. Ghattas. Parametric reduced-order models for probabilistic analysis of unsteady aerodynamic applications. *AIAA Journal*, 46(10): 2520–2529, 2008. doi: 10.2514/1.35850. URL https://doi.org/10.2514/1.35850.

[35] Gu C. A projection based nonlinear model order reduction approach using quadratic linear representation of nonlinear systems. *IEEE Transactions On Computer Aided Design of Integrated Circuits and Systems*, 30(9):1307–1320, 2011.

[36] A. Castagnotto and B. Lohmann. A new framework for $\mathcal{H}_2$-optimal model reduction. *Mathematical and Computer Modelling of Dynamical Systems*, 24(3):236–257, 2018. doi: 10.1080/13873954.2018.1464030. URL https://doi.org/10.1080/13873954.2018.1464030.

[37] V. Chahlaoui, K. A. Gallivan, A. Vandendorpe, and P. Van Dooren. Model reduction of second-order systems. In P. Benner, V. Mehrmann, and D.C. Sorensen, editors, *Dimension Reduction of Large-Scale Systems*, volume 45 of *Lecture Notes in Computational Science and Engineering*, pages 149–172. Springer-Verlag, Berlin/Heidelberg, Germany, 2005.

[38] K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451, 2019. doi: 10.1073/pnas.1906995116. URL https://www.pnas.org/doi/abs/10.1073/pnas.1906995116.

[39] D. Chaniotis and M. A. Pai. Model reduction in power systems using krylov subspace methods. *IEEE Transactions on Power Systems*, 20(2):888–894, 2005.

[40] A. Chatterjee. An introduction to the proper orthogonal decomposition. *Current Science*, 78(7):808–817, 2000. ISSN 00113891. URL http://www.jstor.org/stable/24103957.

[41] S. Chaturantabut and D. C. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, 2010. doi: 10.1137/090766498. URL https://doi.org/10.1137/090766498.

[42] K. K. Chen, J. H. Tu, and C. W. Rowley. Variants of dynamic mode decomposition: boundary condition, koopman, and fourier analyses. *Journal of nonlinear science*, 22 (6):887–915, 2012. doi: 10.1007/s00332-012-9130-9.

[43] X. Cheng, Y. Kawano, and J. M. A. Scherpen. Graph structure-preserving model reduction of linear network systems. In *2016 European Control Conference (ECC)*, pages 1970–1975, 2016. doi: 10.1109/ECC.2016.7810580.

[44] X. Cheng, J. M. A. Scherpen, and Y. Kawano. Model reduction of second-order network systems using graph clustering. In *2016 IEEE 55th Conference on Decision and Control*, pages 7471–7476, 2016.

[45] X. Cheng, Y. Kawano, and J. M. A. Scherpen. Reduction of second-order network systems with structure preservation. *IEEE Transactions on Automatic Control*, 62 (10):5026–5038, 2017.

[46] J. H. Chow. *Power system coherency and model reduction*, volume 84. Springer, 2013.

[47] J. H. Chow and J. J. Sanchez-Gasca. *Power System Coherency and Model Reduction*, pages 531–562. John Wiley & Sons, 2020.

[48] J.H. Chow and K.W. Cheung. A toolbox for power system dynamics and control engineering education and research. *IEEE Transactions on Power Systems*, 7(4):1559–1564, 1992. doi: 10.1109/59.207380.

[49] Z. Drmavc and A. Saibaba. The discrete empirical interpolation method: Canonical structure and formulation in weighted inner product spaces. *SIAM Journal on Matrix Analysis and Applications*, 39(3):1152–1180, 2018. doi: 10.1137/17M1129635. URL https://doi.org/10.1137/17M1129635.

[50] Z. Drmač and S. Gugercin. A new selection operator for the discrete empirical interpolation method—improved a priori error bound and extensions. *SIAM Journal on Scientific Computing*, 38(2):A631–A648, 2016. doi: 10.1137/15M1019271. URL https://doi.org/10.1137/15M1019271.

[51] Z. Drmač, S. Gugercin, and C. Beattie. Quadrature-based vector fitting for discretized $\mathcal{H}_2$ approximation. *SIAM Journal on Scientific Computing*, 37(2):A625–A652, 2015. doi: 10.1137/140961511. URL https://doi.org/10.1137/140961511.

[52] C. W. Gear, J. M. Hyman, P. G Kevrekidid, I. G. Kevrekidis, O. Runborg, and C. Theodoropoulos. Equation-Free, Coarse-Grained Multiscale Computation: Enabling Mocroscopic Simulators to Perform System-Level Analysis. *Communications in Mathematical Sciences*, 1(4):715 – 762, 2003. doi: cms/1119655353. URL https://doi.org/.

[53] F. Giri and E.-W. Bai, editors. *Block-oriented Nonlinear System Identification*. Lecture Notes in Control and Information Sciences. Springer-Verlag, London, 2010.

[54] G. H. Golub and C. F. Van Loan. *Matrix computations*. JHU press, 2013.

[55] I. V. Gosea and A. C. Antoulas. Data-driven model order reduction of quadratic-bilinear systems. *Numerical Linear Algebra with Applications*, 25(6):e2200, 2018. doi: https://doi.org/10.1002/nla.2200. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nla.2200. e2200 nla.2200.

[56] P. K. Goyal. *System-Theoretic Model Order Reduction for Bilinear and Quadratic-Bilinear Control Systems*. PhD thesis, Otto-von-Guericke-Universität, Magdeburg, 2018.

[57] M. A. Grepl and A. T. Patera. A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 39(1):157–181, 2005. doi: 10.1051/m2an:2005006.

[58] A. R. Grimm. *Parametric Dynamical Systems: Transient Analysis and Data Driven Modeling*. PhD thesis, Virginia Tech, 2018.

[59] C. Gu. Qlmor: A projection-based nonlinear model order reduction approach using quadratic-linear representation of nonlinear systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 30(9):1307–1320, 2011.

[60] S. Gugercin and A.s C. Antoulas. A survey of model reduction by balanced truncation and some new results. *International Journal of Control*, 77(8):748–766, 2004. doi: 10.1080/00207170410001713448. URL https://doi.org/10.1080/00207170410001713448.

[61] S. Gugercin, A.C. Antoulas, and C.A. Beattie. $\mathcal{H}_2$ model reduction for large-scale linear dynamical systems. *SIAM Journal on Matrix Analysis and Applications*, 30(2): 609–638, 2008.

[62] B. Gustavsen and A. Semlyen. Rational approximation of frequency domain responses by vector fitting. *IEEE Transactions on Power Delivery*, 14(3):1052–1061, 1999. doi: 10.1109/61.772353.

[63] M. Heinkenschloss, T. Reis, and A.C. Antoulas. Balanced truncation model reduction for systems with inhomogeneous initial conditions. *Automatica*, 47(3):559–564, 2011. ISSN 0005-1098. doi: https://doi.org/10.1016/j.automatica.2010.12.002. URL https://www.sciencedirect.com/science/article/pii/S0005109810004905.

[64] J. S. Hesthaven, G. Rozza, and B. Stamm. *Certified reduced basis methods for parametrized partial differential equations*. Springer Briefs in Mathematics. Springer, Switzerland, 2016. ISBN 978-3-319-22469-5.

[65] M. Hinze and S. Volkwein. Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: Error estimates and suboptimal control. In P. Benner, D. C. Sorensen, and V. Mehrmann, editors, *Dimension Reduction of Large-Scale Systems*, pages 261–306. Springer Berlin Heidelberg, 2005. ISBN 978-3-540-27909-9.

[66] M. Hoffmann, C. Fröhner, and F. Noé. Reactive sindy: Discovering governing reactions from concentration data. *The Journal of Chemical Physics*, 150(2):025101, 2019. doi: 10.1063/1.5066099. URL https://doi.org/10.1063/1.5066099.

[67] J. Hokanson and C. Magruder. $\mathcal{H}_2$-optimal model reduction using projected nonlinear least squares. (1811.11962), 2018. URL https://arxiv.org/abs/1811.11962.

[68] M. Hund, P. Mlinarić, and J. Saak. An $\mathcal{H}_2 \otimes \mathcal{L}_2$-optimal model order reduction approach for parametric linear time-invariant systems. *PAMM*, 18(1):e201800084, 2018. doi: https://doi.org/10.1002/pamm.201800084. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/pamm.201800084.

[69] M. Hund, T. Mitchell, P. Mlinarić, and J. Saak. Optimization-based parametric model order reduction via $\mathcal{H}_2 \otimes \mathcal{L}_2$ first-order necessary conditions. *arXiv preprint arXiv:2103.03136*, 2021.

[70] A. C. Ionita and A. C. Antoulas. Data-driven parametrized model reduction in the loewner framework. *SIAM Journal on Scientific Computing*, 36(3):A984–A1007, 2014. doi: 10.1137/130914619. URL https://doi.org/10.1137/130914619.

[71] T. Ishizaki and J. Imura. Clustered model reduction of interconnected second-order systems. *Nonlinear Theory and Its Applications, IEICE*, 6(1):26–37, 2015. doi: 10.1587/nolta.6.26.

[72] T. Ishizaki, A. Chakrabortty, and J. Imura. Graph-theoretic analysis of power systems. *Proceedings of the IEEE*, 106(5):931–952, 2018. doi: 10.1109/JPROC.2018.2812298.

[73] I. T. Jolliffe and J. Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical*

and Engineering Sciences, 374(2065):20150202, 2016. doi: 10.1098/rsta.2015.0202. URL https://royalsocietypublishing.org/doi/abs/10.1098/rsta.2015.0202.

[74] H.-J. Jongsma, P. Mlinarić, S. Grundel, P. Benner, and H.L. Trentelman. Model reduction of linear multi-agent systems by clustering with $\mathcal{H}_2$ and $\mathcal{H}_\infty$ error bounds. *Mathematics of Control, Signals, and Systems*, 30(1):6, 2018.

[75] A. Juditsky, H. Hjalmarsson, A. Benveniste, B. Delyon, L. Ljung, J. Sjöberg, and Q. Zhang. Nonlinear black-box models in system identification: Mathematical foundations. *Automatica*, 31(12):1725 – 1750, 1995.

[76] P. Kergus, F. Demourant, and C. Poussot-Vassal. Identification of parametric models in the frequency-domain through the subspace framework under lmi constraints. *International Journal of Control*, 93(8):1879–1890, 2020.

[77] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009. doi: 10.1137/07070111X. URL https://doi.org/10.1137/07070111X.

[78] B. Kramer and K. E. Willcox. Nonlinear model order reduction via lifting transformations and proper orthogonal decomposition. *AIAA Journal*, 57(6):2297–2307, 2019. doi: 10.2514/1.J057791. URL https://doi.org/10.2514/1.J057791.

[79] P. Kundur. *Power System Stability and Control*. McGraw-Hill, New York, NY, 1994.

[80] K. Kunisch and S. Volkwein. Proper orthogonal decomposition for optimality systems. *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, 42(1):1–23, 2008. doi: 10.1051/m2an:2007054. URL http://www.numdam.org/item/M2AN_2008__42_1_1_0.

[81] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor. *Dynamic Mode Decomposition*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2016. doi: 10.1137/1.9781611974508. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611974508.

[82] A. J. Mayo and A. C. Antoulas. A framework for the solution of the generalized realization problem. *Linear Algebra and its Applications*, 425(2):634–662, 2007. ISSN 0024-3795. doi: https://doi.org/10.1016/j.laa.2007.03.008. URL https://www.sciencedirect.com/science/article/pii/S0024379507001280. Special Issue in honor of Paul Fuhrmann.

[83] G. P. McCormick. Computability of global solutions to factorable nonconvex programs: Part i—convex underestimating problems. *Mathematical programming*, 10(1):147–175, 1976.

[84] S. A. McQuarrie, C. Huang, and K. E. Willcox. Data-driven reduced-order models via regularised operator inference for a single-injector combustion process. *Journal of the Royal Society of New Zealand*, 51(2):194–211, 2021. doi: 10.1080/03036758.2020.1863237. URL https://doi.org/10.1080/03036758.2020.1863237.

[85] D. G. Meyer and S. Srinivasan. Balancing and model reduction for second-order form linear systems. *IEEE Transactions on Automatic Control*, 41(11):1632–1644, 1996.

[86] M. Baumann S. Grundel P. Mlinarić. Complexity reduction for power flow simulations, 2019. URL http://manuelbaumann.de/projects/fepset19.pdf.

[87] P. Mlinarić. *Structure-Preserving Model Order Reduction for Network Systems*. PhD thesis, Otto-von-Guericke-Universität Magdeburg, 2020.

[88] P. Mlinarić, S. Grundel, and P. Benner. Efficient model order reduction for multi-agent systems using QR decomposition-based clustering. In *Proceedings of 54th IEEE Conference on Decision and Control*, pages 4794–4799, 2015.

[89] N. Monshizadeh, H.L. Trentelman, and M.K. Camlibel. Projection-based model reduction of multi-agent systems using graph partitions. *IEEE Transactions on Control of Network Systems*, 1(2):145–154, 2014.

[90] T. Nishikawa and A. E Motter. Comparative analysis of existing models for power-grid synchronization. *New Journal of Physics*, 17(1):015012, Jan 2015. doi: 10.1088/1367-2630/17/1/015012. URL https://doi.org/10.1088.

[91] T. Nishikawa and A.E. Motter. Comparative analysis of existing models for power-grid synchronization. *New Journal of Physics*, 17(1):015012, 2015.

[92] H. K. F. Panzer, S. Jaensch, T. Wolf, and B. Lohmann. A greedy rational krylov method for $\mathcal{H}_2$-pseudooptimal model order reduction with preservation of stability. In *2013 American Control Conference*, pages 5512–5517, 2013. doi: 10.1109/ACC.2013.6580700.

[93] B. Peherstorfer and K. Willcox. Online adaptive model reduction for nonlinear systems via low-rank updates. *SIAM Journal on Scientific Computing*, 37(4):A2123–A2150, 2015. doi: 10.1137/140989169. URL https://doi.org/10.1137/140989169.

[94] B. Peherstorfer and K. Willcox. Data-driven operator inference for nonintrusive projection-based model reduction. *Computer Methods in Applied Mechanics and Engineering*, 306:196 – 215, 2016. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2016.03.025. URL http://www.sciencedirect.com/science/article/pii/S0045782516301104.

[95] B. Peherstorfer, D. Butnaru, K. Willcox, and H. Bungartz. Localized discrete empirical interpolation method. *SIAM Journal on Scientific Computing*, 36(1):A168–A192, 2014. doi: 10.1137/130924408. URL https://doi.org/10.1137/130924408.

[96] Benjamin Peherstorfer, Zlatko Drmač, and Serkan Gugercin. Stability of discrete empirical interpolation and gappy proper orthogonal decomposition with randomized and deterministic sampling points. *SIAM Journal on Scientific Computing*, 42(5): A2837–A2864, 2020. doi: 10.1137/19M1307391. URL https://doi.org/10.1137/19M1307391.

[97] J. R. Phillips. Projection frameworks for model reduction of weakly nonlinear systems. In *Proceedings of the 37th Annual Design Automation Conference*, DAC '00, page 184–189, New York, NY, USA, 2000. Association for Computing Machinery. ISBN 1581131879. doi: 10.1145/337292.337380. URL https://doi.org/10.1145/337292.337380.

[98] J. R. Phillips. Projection-based approaches for model reduction of weakly nonlinear, time-varying systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 22(2):171–187, 2003.

[99] B. K. Poolla, S. Bolognani, and F. Dörfler. Optimal placement of virtual inertia in power grids. *IEEE Transactions on Automatic Control*, 62(12):6209–6220, 2017.

[100] J. L. Proctor, S. L. Brunton, and J. N. Kutz. Dynamic mode decomposition with control. *SIAM Journal on Applied Dynamical Systems*, 15(1):142–161, 2016. doi: 10.1137/15M1013857. URL https://doi.org/10.1137/15M1013857.

[101] E. Qian. Operator inference. https://github.com/elizqian/operator-inference, 2019.

[102] E. Qian, B. Kramer, B. Peherstorfer, and K. Willcox. Lift & learn: Physics-informed machine learning for large-scale nonlinear dynamical systems. *Physica D: Nonlinear Phenomena*, 406:132401, 2020. ISSN 0167-2789. doi: https://doi.org/10.1016/j.physd.2020.132401. URL http://www.sciencedirect.com/science/article/pii/S0167278919307651.

[103] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced Basis Methods for Partial Differential Equations: An Introduction*. R. UNITEXT. Springer Cham, 2016.

[104] T. Reis and T. Stykel. Balanced truncation model reduction of second-order systems. *Mathematical and Computer Modelling of Dynamical Systems*, 14(5):391–406, 2008. doi: 10.1080/13873950701844170. URL https://doi.org/10.1080/13873950701844170.

[105] M. J. Rewienski. *A Trajectory Piecewise-Linear Approach to Model Order Reduction of Nonlinear Dynamical Systems*. PhD thesis, Massachusetts Institute of Technology, 2003.

[106] T. K.S. Ritschel, F. Weiß, M. Baumann, and S. Grundel. Nonlinear model reduction of dynamical power grid models using quadratization and balanced truncation. *at - Automatisierungstechnik*, 68(12):1022–1034, 2020. doi: doi:10.1515/auto-2020-0070. URL https://doi.org/10.1515/auto-2020-0070.

[107] C. W. Rowley., I. Mezi, S. Bagheri, P. Schlatter, and Dan S. H. Spectral analysis of nonlinear flows. *Journal of Fluid Mechanics*, 641:115–127, December 2009. ISSN 0022-1120. doi: 10.1017/S0022112009992059.

[108] B. Safaee and S. Gugercin. Structure-preserving model reduction of parametric power networks. In *2021 American Control Conference (ACC)*, pages 1824–1829, 2021. doi: 10.23919/ACC50511.2021.9483168.

[109] B. Safaee and S. Gugercin. Data-driven modeling of power networks. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 4236–4241, 2021. doi: 10.1109/CDC45484.2021.9683100.

[110] B. Safaee and S. Gugercin. Structure-preserving model reduction for nonlinear power grid network. e-print 2203.09021, arXiv, 2022. URL https://arxiv.org/abs/2203.09021. eess.SY.

[111] A. T. Sarić, A. A. Sarić, M. K. Transtrum, and A. M. Stanković. Symbolic regression for data-driven dynamic model refinement in power systems. *IEEE Transactions on Power Systems*, 36(3):2390–2402, 2021. doi: 10.1109/TPWRS.2020.3033261.

[112] P. W. Sauer and M.A. Pai. *Power system dynamics and stability*, volume 101. Wiley Online Library, 1998.

[113] P. J. Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656:5–28, 2010. doi: 10.1017/S0022112010001217.

[114] M. Schmidt and H. Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85, 2009. doi: 10.1126/science.1165893. URL https://www.science.org/doi/abs/10.1126/science.1165893.

[115] D. C. Sorensen and M. Embree. A deim induced cur factorization. *SIAM Journal on Scientific Computing*, 38(3):A1454–A1482, 2016. doi: 10.1137/140978430. URL https://doi.org/10.1137/140978430.

[116] A. M. Stanković, A. A. Sarić, A. T. Sarić, and M. K. Transtrum. Data-driven symbolic regression for identification of nonlinear dynamics in power systems. In *2020 IEEE Power Energy Society General Meeting (PESGM)*, pages 1–5, 2020. doi: 10.1109/PESGM41954.2020.9281935.

[117] R. Stefanescu and A. Sandu. A goal-oriented adaptive discrete empirical interpolation method. e-print 1901.05343, arXiv, 2019. URL https://arxiv.org/abs/1901.05343. math.NA.

[118] T. Su and R. R. Craig. Model reduction and control of flexible structures using krylov vectors. *Journal of Guidance, Control, and Dynamics*, 14(2):260–267, 1991. doi: 10.2514/3.20636. URL https://doi.org/10.2514/3.20636.

[119] R. Swischuk, B. Kramer, C. Huang, and K. Willcox. Learning physics-based reduced-order models for a single-injector combustion process. *AIAA Journal*, 58(6):2658–2672, 2020. doi: 10.2514/1.J058943. URL https://doi.org/10.2514/1.J058943.

[120] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996. doi: https://doi.org/10.1111/j.2517-6161.1996.tb02080.x. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1996.tb02080.x.

[121] Z. Tomljanović, C. Beattie, and S. Gugercin. Damping optimization of parameter dependent mechanical systems by rational interpolation. *Advances in Computational Mathematics*, pages 1–24, 2018.

[122] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz. On dynamic mode decomposition: Theory and applications. *Journal of Computational Dynamics*, 1(2):391–421, 2014.

[123] K. Veroy, C. Prud'homme, D. Rovas, and A. Patera. *A Posteriori Error Bounds for Reduced-Basis Approximation of Parametrized Noncoercive and Nonlinear Elliptic Partial Differential Equations*, page 3847. doi: 10.2514/6.2003-3847. URL https://arc.aiaa.org/doi/abs/10.2514/6.2003-3847.

[124] H. Weber and E. Welfonder. Dynamic model reduction for the modal analysis of frequency and power oscillations in large power systems. In A.J. CALVAER, editor, *Power Systems: Modelling and Control Applications*, IFAC Symposia Series, pages 233–239. Pergamon, Oxford, 1989. ISBN 978-0-08-036135-2. doi: https://doi.org/10.1016/B978-0-08-036135-2.50043-0. URL http://www.sciencedirect.com/science/article/pii/B9780080361352500430.

[125] F. Weiß. *Simulation, analysis, and model order reduction for dynamic power network models*. PhD thesis, Master's thesis, Otto-von-Guericke-Universität Magdeburg, 2019.

[126] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley. A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015. doi: 10.1007/s00332-015-9258-5. Communicated by Oliver Junge.

[127] S. A. Wyatt. *Issues in interpolatory model reduction: Inexact solves, second-order systems and DAEs*. PhD thesis, Virginia Tech, 2012.

[128] Y. Xu and T. Zeng. Optimal $\mathcal{H}_2$ model reduction for large scale MIMO systems via tangential interpolation. *International Journal of Numerical Analysis and Modeling*, 8 (1):174–188, 2011.

[129] L. Yu, X. Cheng, J. M. A. Scherpen, and E. Gort. $\mathcal{H}_2$ sub-optimal model reduction for second-order network systems. In *2019 IEEE 58th Conference on Decision and Control*, pages 5062–5067, 2019.

[130] S. Z. Zhu, Z. Y. Dong, K. P. Wong, and Z.H. Wang. Power system dynamic load identification and stability. In *PowerCon 2000. 2000 International Conference on Power System Technology. Proceedings (Cat. No.00EX409)*, volume 1, pages 13–18 vol.1, 2000. doi: 10.1109/ICPST.2000.900024.

[131] R. D. Zimmerman and C. E. Murillo-Sánchez. Matpower 6.0 user's manual. *Power Systems Engineering Research Center*, 9, 2016.

[132] R. D Zimmerman and C. E Murillo-Sánchez. Matpower 6.0 user manual. *Power Systems Engineering Research Center*, 9, 2016.

[133] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas. Matpower: Steady-state operations, planning, and analysis tools for power systems research and education. *IEEE Transactions on Power Systems*, 26(1):12–19, 2010.

Appendices

Appendix A

Permutation matrix

The permutation matrix $P_q \in \mathbb{R}^{4n^2 \times 4n^2}$ included in P in (3.28) is generated using Algorithm 8.

Algorithm 8 MATLAB Pseudo-code for P_q

$I = \text{speye}(4n^2)$, $P_q = \text{sparse}(4n^2, 4n^2)$
$i = 1$, $j = 1$, iter $= 1$
while iter $\leq n$ **do**
  $P_q(:, (i-1)n+1 : i.n) = I(:, j : n + (j-1))$
  $P_q(:, (i-1+n)n+1 : (i+n)n) = I(:, j+n : 2n + (j-1))$
  $P_q(:, (i-1+2n)n+1 : (i+2n)n) = I(:, j+2n : 3n + (j-1))$
  $P_q(:, (i-1+3n)n+1 : (i+3n)n) = I(:, j+3n : 4n + (j-1))$
  $j = j + 4n$
  $i = i + 1$
  iter $=$ iter $+ 1$
end while