OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

FACULTY OF
MATHEMATICS

Simulation, Analysis, and Model Order Reduction for Dynamic Power Network Models

A master thesis submitted by

Frances Weiß

to the
Faculty of Mathematics
Otto-von-Guericke University Magdeburg
in partial fulfillement of the requirements
for the degree of

Master of Science

in

Mathematics

Dr. Sara Grundel
Advisor and 1st reviewer,
Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg

Jun.-Prof. Dr. Jan Heiland
2nd reviewer,
Otto-von-Guericke University Magdeburg
Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg

Dr. Manuel Baumann
Co-advisor,
Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg

Submitted: August 30, 2019

Acknowledgments

First of all, I would like to thank my supervisor, Sara Grundel, and my co-advisor, Manuel Baumann, for their support, time, and patience. With their guidance, I learned a lot of means and tricks to overcome mathematical challenges, and I extended my skill set greatly during the work on this thesis. I would also like to thank Jan Heiland for agreeing to be my second reviewer. There are many other people at the Max-Planck-Institute Magdeburg who deserve credit for giving useful input. Especially Martin Köhler spent a lot of time trying to get me server and MATLAB access.

A thank you to the team at the Very Large Business Applications Lab, too, for being such endearing colleagues. They provided me with with a place to work and even a place to sleep when I needed it.

Finally, I want to express my deepest gratitude to my entire family. Without my extraordinary wife, her love, her encouragement and her belief in me, I would never have been able to accomplish this. Ein unendlich großer Dank geht an meine phänomenalen Eltern. Ihre Liebe, Geduld und Unterstützung haben mir so viel Kraft und Ausdauer geschenkt – unter anderem. Es gibt keine Worte für all das, was sie mir gegeben haben. Ich danke auch meiner Schwester, meinem Bruder, meinem Schwager, meiner Schwägerin, meinen Neffen, meiner Nichte, meiner Omi und Tante für ablenkende Gespräche und Besuche, und Verständnis für verpasste Familienfeiern. Ich habe euch so lieb!

Danke. Thank you.

List of Tables

List of Figures

List of Algorithms

Contents

Glossary and Acronyms

Glossary

A	matrix describing linear part of quadratic system
α	shift parameter to stabilize the A matrix
B	input matrix of quadratic system
C	output matrix of quadratic system
c	cosine values of angles: $c = \cos(\delta)$
coupled oscillator representation	PN represented as a system of coupled oscillators with arbitrary coupling structure; exact coupling structure defined by power network (PN) models, , *see* oscillator
D	damping coefficient characterizing the oscillators
δ	phase angle relating to the internal voltage magnitude $\lvert\mathsf{E}\rvert$ at an internal node
electric circuit representation	PN represented as a system of electric components characterized by electric power laws
E	voltage at an internal node with magnitude $\lvert\mathsf{E}\rvert$
ϵ	absolute L_∞ error
F	one of the parameters determining the inherent frequency
g	used as a super- or subscript indicating values concerning generator nodes
γ	phase shift involved in oscillator coupling
γ^c	cosine values of phase shifts: $\gamma^c = \cos(\gamma)$
γ^s	sine values of phase shifts: $\gamma^s = \sin(\gamma)$
generator node	an element in the PN system which transforms mechanical into electric power
H	matrix describing quadratic part of quadratic system
i	imaginary unit
internal node	an element in the PN system between the constant-magnitude voltage source (with time-dependent phase)

	and the transient reactance x_r'
I	electric current, the flow of electric charge
J	inertia coefficient characterizing the oscillators
K	strength of dynamic coupling between two oscillators
ℓ	used as a super- or subscript indicating values concerning load nodes
λ	eigenvalue
link	an electric connection between two nodes, either a transmission line or a transformer
load node	an element in the PN system which consumes electric power
n	number of nodes in the physical representation of the PN
n_{co}	number of nodes in the coupled oscillator representation of the PN, equals the number of equations in the ODE system after application of one of the PN models
n_{ec}	number of nodes in the electric circuit representation of the PN
n_g	number of generator nodes in the PN
n_ℓ	number of load nodes in the PN
$n_{\mathbf{min}}$	singular value
node	an element in the PN system at which power is injected by a generator or extracted by consumers or a branching point through which power is redistributed
n_ρ	number of equations in the reduced-order model
ν	number of Volterra kernels in the computation of $\mathcal{P}$, $\mathcal{Q}$
$\varphi^\star$	inherent frequency of an oscillator in a PN system
φ_R	reference frequency for a PN system
oscillator	can represent either a generator or a load node
P	active power in the power flow equation
$\mathcal{P}$	reachability Gramian matrix
P_e	electric power consumed by the nongenerator components of the network
ϕ	phase angle relating to the complex voltage magnitude $\vert\mathsf{V}\vert$
physical representation	PN represented with the actual hardware components used to built the network

P_m	net mechanical power input to the generator rotor
power network	generic term for different types of power grid representations
$\mathcal{Q}$	observability Gramian matrix
Q	reactive power in the power flow equation
reference node	an element in the PN system to account for the unknown power losses in the system
$\mathcal{R}$	Cholesky factor of the reachability Gramian $\mathcal{P}$
n_ρ	reduction order of the MOR
s	sine values of angles: $s = \sin(\delta)$
$\mathcal{S}$	Cholesky factor of the observability Gramian $\mathcal{Q}$
σ	singular value
t	used as a superscript indicating values concerning terminal nodes
terminal node	an element in the PN system allowing the circuit to interconnect with its environment and connecting the generator to the rest of the network
ϑ	phase angle of the PN model's admittance matrix
transmission line	an electric coupling between two nodes, *see* link
u	input vector of quadratic system
$\mathcal{V}$	projection matrix of the MOR containing the left-singular vectors
V	complex voltage with magnitude $\lvert\mathsf{V}\rvert$
$\mathcal{W}$	projection matrix of the MOR containing the right-singular vectors
x	state vector with phase angles, their first derivatives, their sin and cos substitutions: $x = [\delta, \omega, s, c]^*$
x_r'	transient reactance, opposition to a change in electric current effective after the damper winding currents have diminished
y	output vector of quadratic system
$\boldsymbol{\mathsf{Y}}$	admittance matrix
Y	complex admittance

Acronyms

BT	Balanced Truncation
CO	coupled oscillator, *see also glossary entry*
EC	electric circuit, *see also glossary entry*

EN	Effective Network
EV	eigenvalue
EVD	Eigenvalue Decomposition
FOM	full-order model
IVP	initial value problem
LHS	left hand side
MOR	Model Order Reduction
ODE	Ordinary Differential Equation
PN	power network, *see also glossary entry*
POD	Proper Orthogonal Decomposition
PTI	Pythagorean Trigonometric Identity
RHS	right hand side
ROM	reduced-order model
SM	Synchronous Motor
SPD	symmetric positive definite
SV	singular value
SVD	Singular Value Decomposition

Symbols

$\mathbb{I}$	identity matrix
$\otimes$	Kronecker product
$\mathbf{1}$	matrix of all ones
$\star$	value of a state variable for a power flow solution
M^*	transpose if $M \in \mathbb{R}^{n\times k}$, complex conjugate transpose if $M \in \mathbb{C}^{n\times k}$,
$\mathbf{0}$	matrix of all zeros

CHAPTER 1

Introduction

Motivation

Dynamic systems are used in a multitude of applications. A lot of them have an increasingly complex structure, e.g., control problems like heating and cooling processes or chemical reactions, or prediction problems like weather forecasts or biological processes. The numerical simulation of these problems attempts to predict or control the system behavior in order to test features or ascertain its conduct over time [Ant05; AS01].

In this thesis, we engage in a particular aspect of PN modeling as an example simulation problem, namely, the *synchronization stability of power generators*. If a single generator runs at a different frequency, the PN will fail to operate properly. A capable PN model must therefore incorporate the dynamics between coupled power generators as part of the network structure. Due to the advent of better computational capabilities and data processing methods researchers have been enabled to study large-scale features like the network structure.

Reasons for asynchronicity might be fluctuations in power demand or the malfunction of system components [NM15a]. The implementation of smart-grids and the insertion of less reliable renewable energy sources, e.g., wind and solar power, will make the networks even more vulnerable to instability [DB12]. Due to these developments, the overall importance of synchronization stability, and a research gap regarding the influence of the network structure on the interconnected system dynamics have made this an area of active research in recent years [DB12; NM15a]. [NM15a] compare three leading models in their paper on power network synchronization, two of which are considered in this thesis. They are all based on a common design, representing the PN's structure as an interlinked 2^{nd}-order system of Ordinary Differential Equations (ODEs) with *nonlinear* dynamics. In their paper, [NM15a] derive the model-dependent parameters which are necessary to find a solution to the ODE systems. However, they do not actually solve these systems. This is the first motivation of this thesis.

Faithfully modeling such large-scale systems requires a considerable number of state variables. The model is therefore often highly complex and its full simulation faces a number of difficulties: Its complexity might be too large to meet storage limitations.

Also, computation results must be produced in reasonable time. Using large-scale models might therefore be infeasible due to limited computational speed. Furthermore, accuracy suffers from the finite precision representation of numbers in computers and the ill-conditioning of large-scale systems [Ant05; AS01]. Although these problems have indeed been mitigated by the rise of computational capabilities in the last years, high-complexity models of dimension up to 10^6 present a challenge, nonetheless.

MOR offers to overcome this dilemma by providing a reduced-order model (ROM) of the original system. A ROM must manage the balancing act of being of low-dimension by downsizing the number of state variables on the one hand, while simultaneously approximating the full-order model (FOM)'s input-output-behavior with an acceptable error and maintain significant system properties on the other hand [BG17].

While MOR theory and methods have been thoroughly studied and used in practice regarding linear systems, the development and adaptation of MOR techniques in the nonlinear context is a work in progress. In general, nonlinear MOR presents challenges which make it more difficult than reducing linear systems. One of the problems is the possibly costly computation, even after the reduction. Solving the nonlinear ROM can be just as costly as the original system, rendering the MOR ineffective. Another issue is the unpredictable behavior of the reduced system. A priory, it is often unclear which (global) properties the ROM will retain from the FOM. Also, nonlinear systems do not provide a canonical form for the ROM. This makes it hard to find an adequate representation and construct useful algorithms [Gu11].

To remedy these challenges, several approaches have been taken. One is to linearize the nonlinear system around an expansion point and subsequently apply linear MOR. This represents the system dynamics well in the vicinity of the expansion point, but fails as the distance grows larger. For this reason, nonlinear dynamics should be reflected in the MOR process. The Proper Orthogonal Decomposition (POD) method in its various incarnations is a popular example which has been successfully applied to general nonlinear systems [BG17; KW19; Pin08; RP03; Vol13]. For several years, there has also been a focus on extending techniques originally designed for linear systems to nonlinear systems with a specific structure. Especially bilinear systems have received much attention, because a lot of biological, physical and physiological processes can be modeled in the form of such systems [BBF14; BGR17; BPK71; Fla12; VG18]. More recently, quadratic-bilinear systems have been studied with regard to MOR. Lifting a nonlinear to a quadratic-bilinear system by adding auxiliary variables to the state vector might not result in a unique representation, but it is an exact process, i.e., there is no added approximation error [BBF14; BG17; Gu11; KW19].

In contrast, transforming nonlinear systems to give them a quadratic structure has been largely disregarded in MOR research so far [BBF14; Che99]. The second motivation of this thesis is thus to lift the nonlinear PN ODE systems to quadratic form, and explore the application of the Balanced Truncation (BT) MOR method to these transformed systems. The method is adapted to the quadratic context beforehand, based on the work by [BG17].

The overall goal of this thesis is to examine the performance of BT applied to the quadratic version of the PN ODE systems. To that end, we solve the systems for the FOM and ROM and vary different reduction parameters to study their influence.

Outline

We approximate our overall goal by first exploring our two main motivations separately in stand-alone chapters.

We begin Chapter 2 by introducing some concepts and terminology of power system analysis. Building on this basis, we derive the dynamic power network models. This includes a characterization and comparison of the model-dependent parameters. At the end, we have two similar nonlinear 2^{nd}-order ODE systems of different sizes.

In Chapter 3, we present the problem setup and idea of MOR in general, and BT in particular, in their linear and nonlinear setting. We introduce the key concepts of reachability and observability, the respective Gramians and how they facilitate the actual process of balancing and truncating, highlighting distinct features of the quadratic case. In addition, we establish the basics of Proper Orthogonal Decomposition (POD), a method we use to compare the performance of quadratic BT.

While Chapters 2 and 3 are independent of each other, we link their topics in Chapter 4. Since the derived PN models lack structure, we bring them into the appropriate quadratic form in a multi-step process. We also look at the necessary assumptions for the application of BT and take care of their satisfaction.

In Chapter 5, we consider a PN test case to evaluate the performance of quadratic BT. We apply the method to the two PN models generated from the test case and compare it to the POD method.

In the last chapter, Chapter 6, we conclude with summarizing and assessing the findings.

CHAPTER 2

Dynamic Power Network Modeling

As mentioned in the introduction, a power network can only operate if all of its generators run at the same frequency. The coupling dynamics between the generators determine their state and therefore the synchronization stability of the system. In consequence, a good network model must properly describe the coupling dynamics of power generators. There are various ways to do this, here we focus on two leading models: the EN model and the Synchronous Motor (SM) model.[1] These models express a PN system containing generator and load nodes as a network of coupled phase oscillators. Therefore, we refer to the EN and SM models as types of *coupled oscillator (CO) models*.

The state of each PN node i is characterized by δ_i, its phase angle, and $\dot{\delta}_i$, its frequency [NM15a] or velocity [WT12]. With $\delta_i = \delta_i(t)$ and $i = 1, \ldots, n_{co}$, the coupling dynamics between oscillators is described by a system of Ordinary Differential Equations (ODEs), the so-called *swing equations*:

$$\frac{2J_i}{\varphi_R}\ddot{\delta}_i + \frac{D_i}{\varphi_R}\dot{\delta}_i = F_i - \sum_{j=1,\, j\neq i} K_{ij}\sin(\delta_i - \delta_j - \gamma_{ij}), \tag{2.1}$$

where the inertia J and damping D constants as well as the reference frequency φ_R are model-independent coefficients determined by the physical structure of the system. The parameters F, K and γ, on the other hand, dependent on the modeling of the system's nodes. While [NM15a] provide the derivation and a comparison of these model-dependent quantities, they do not solve the PN ODE system (2.1). This is one of our main objectives.

The purpose of this chapter is to retrace [NM15a]'s derivation of the parameters F, K, and γ, which are necessary to solve the equations in the ODE system (2.1) for each generator and load node. These two types of nodes differ in their modeling, however. While power generation can be regulated and is predictable to a certain extent, power consumption is affected by a number of different, time-dependent factors like the load type, size and quantity of nodes as well as human activity. The two CO models vary mostly in their description of load nodes, and contingent on this, in the number

[1] In their paper, [NM15a] also discuss a third option, the Structure-Preserving model. It represents loads as first- and generators as second-order ODEs. Because it deviates in its derivation and structure from the other two models, it is disregarded in this work.

of equations n_{co} [NM15a]. We characterize the model-dependent parameters in Section 2.3.3. First, we introduce some power systems terminology and notation in the following Section 2.1, which we use in the context of power network modeling. Then, we address generator node modeling in Section 2.2, before we focus on the load nodes in Section 2.3.

2.1 Basic Concepts of Power System Analysis

A power network is a system consisting of a number of nodes and links, operating to distribute power from generators to consumers. *Power network* is the generic term for different types of power grid representations. First, there is the physical representation, i.e., the actual hardware used to engineer the network. Second, the electric circuit representation describes the network in terms of electric components and electric power laws. Third, the coupled oscillator representation characterizes the PN as a pure system of coupled oscillators, where an oscillator can serve as either a generator or a load node with mathematical identical modeling [NM15a]. An oscillator is an electronic circuit which produces a periodic output, e.g., a sine or square wave [Sne].

A node is the generic term for a point in the network, where power is either injected by a generator or extracted by a consumer or redistributed along branched off transmission lines. We distinguish different kinds of nodes. The simplest distinction is between generator nodes and load nodes. Both are elements in the power network system with the former injecting power into the network by transforming mechanical into electric power and the latter consuming said electric power. A terminal node allows the circuit to interconnect with its environment, and in our case specifically connects the generator to the rest of the network [NM15a; Wil10]. An internal node is a point between the constant-magnitude voltage source and the transient reactance x'_r. [NM15a].

We also operate with two types of complex voltage: When we talk about the voltage at an internal node, we denote it by the internal voltage E. The symbol V indicates the generic complex voltage, but also the terminal voltage at a terminal node if explicitly mentioned.

The links are transmission lines and serve to electrically couple the nodes. Transmission lines have complex-valued impedances. Admittances are the inverse of impedances. Due to this connection, the admittance matrix $\mathbf{Y}$ encodes the structure of the respective network representation [NM15a]. The construction of an admittance matrix needs much less effort than the construction of an impedance matrix. The matrix itself is often sparse for large-scale PN, because each node typically has only a few links to neighboring nodes. Therefore, it is a common approach in power system analysis to use equivalent admittances to express the parameters characterizing the transmission lines [GS94; NM15a]. An admittance matrix denoted by $\mathbf{Y}_0$, i.e., with an additional subscripted 0, represents the physical network structure [MMA+13; NM15a].

Reactance x is the imaginary part of the impedance and quantifies the opposition to a change in electric current I, the flow of electric charge [IL14]. The transient reactance x_r' is the reactance effective after the damper winding[2] currents have diminished [GS94].

We encounter different concepts of power. The active power P makes up the real and the reactive power Q the imaginary parts of the complex power S, which adheres to the formulae

$$S = \mathsf{V}\bar{\mathsf{I}} = \mathsf{P} + \mathsf{i}\mathsf{Q} = |\mathsf{V}||\mathsf{I}|\cos(\phi) + \mathsf{i}|\mathsf{V}||\mathsf{I}|\sin(\phi), \tag{2.2}$$

where V denotes the voltage, $\bar{\mathsf{I}}$ the complex conjugate of the electric current, ϕ the phase angle, and i is the imaginary unit [GS94; IL14; NM15a]. Furthermore, P_m is the net mechanical power input to the generator's rotor. The electric power P_e is the requested power by the nongenerator components of the network [NM15a].

Another context-dependent term we use is the *phase angle*. While δ describes the phase angle at an internal node and always appears in conjunction with the internal voltage magnitude $|\mathsf{E}|$, ϕ represents the phase angle at a terminal or load node and consistently occurs with the voltage magnitude $|\mathsf{V}|$. Which phase angle and voltage (magnitude) representation is used depends on the network component. The phase angle matrix $\boldsymbol{\vartheta}$ is calculated from the respective CO model's admittance matrix $\mathbf{Y}$. The term γ denotes the phase angle shift in the oscillator coupling [NM15a].

Closely connected to angles are frequencies. The first derivative of a phase angle δ is the node's frequency $\dot{\delta}$. We make a distinction between the term φ_R, which is the reference frequency for the PN system and an oscillator's inherent frequency $\varphi^\star$, which is the equilibrium frequency of an oscillator in case there is no coupling to another oscillator [NM15a].

The oscillators are also characterized by the inertia J and damping D coefficients. Just like the reference frequency φ_R, the inertia and damping parameters are given by the physical structure of the system [NM15a].

We refer to two important electric circuit laws. First, *Kirchhoff's current law* says that for any node in a circuit the sum of currents which flow towards that node is identical to the sum of currents which flow away from that node, i.e.,

$$\sum_j \mathsf{I}_j = 0, \tag{2.3}$$

with I being the electric current. Second, we state *Ohm's law* in the following way,

$$\mathsf{I} = \frac{\mathsf{E} - \mathsf{V}}{\mathsf{i}\mathsf{x}_r'}, \tag{2.4}$$

where x_r' is the transient reactance, and E and V stand for the internal and terminal voltage, respectively. Using the admittance matrix $\mathbf{Y}$, we can restate Ohm's law in the

[2] The damper winding in a machine is responsible for the decrease of the mechanical oscillations of the rotor approximately around synchronous speed.

entire network as a matrix-vector product,

$$\mathsf{\mathbf{I}} = \mathsf{\mathbf{Y}}\mathsf{\mathbf{V}}, \tag{2.5}$$

with current vector $\mathsf{\mathbf{I}}$ and voltage vector $\mathsf{\mathbf{V}}$, respectively [IL14; NM15a].

We attempt to avoid confusion and use sans serif letters for those variables and parameters which primarily occur in the power systems context and serif or other special font types for those which primarily occur in the mathematical parts of this thesis. For example, the "P" for the active power P is not the same as the "P" for the matrix $\mathcal{P}$.

In the context of power system analysis, we use bold letters whenever we mean matrices or vectors, e.g., $\mathsf{\mathbf{Y}}$ is the admittance matrix whereas Y is a scalar admittance value, and $\mathsf{\mathbf{V}}$ is the voltage vector whereas V is a scalar voltage value. Whenever g, ℓ are used as a super- or subscript they indicate values concerning generator or load nodes, respectively. A minuscule i used as a superscript, e.g., in g^i, ℓ^i, refers to generator internal and load internal nodes, respectively, and a minuscule t used as a superscript, as in g^t, refers to terminal nodes. The superscripted star $\star$ marks the value of a state variable for a power flow solution, and a bar $\bar{q}$ above a variable q indicates the complex conjugate.

For quick reference, you can also find a concise explanation of these terms in the Glossary and Acronyms chapter, and a list of symbols, beginning on page xiii.

2.2 The Basic Power Network Model for Generator Nodes

The goal is to find computable expression of the model-dependent parameters F, K and γ in the dynamic system (2.1). To that end, we want to identify the quantities necessary to describe the state of a node. In this section, we focus on generator modeling and look at two different representations of a generator to derive two important equations: the *mechanical* representation leads to the so-called *swing equation* and the *electric circuit* representation introduces the *classical model*. On these grounds, we discover the characterizing quantities of a node's state and how to compute them for generators at the end of this section.

The Swing Equation The Mechanical Representation of a Generator

Mechanically, a power generator often consists of a rotor (among other parts), i.e., a rotating mass. Its motions are affected by a mechanical and an electric torque. From these torques, the mechanical and electric power (P_m and P_e, respectively) working on the rotor can be calculated by multiplication with the angular velocity of the rotor, ω (frequently, the approximation $\omega \approx \varphi_R$ is assumed). Usually, both types of torque are positive, so that the mechanical torque corresponds to the generator's power input, and the electric torque corresponds to the power output or to the electric power demanded by the rest of the network. The mechanical power and torque are assumed to be constant, so the electric power and torque determine the rotation's change in velocity, i.e., if the rotor is accelerated ($\mathsf{P}_m > \mathsf{P}_e$), decelerated ($\mathsf{P}_m < \mathsf{P}_e$), or steady ($\mathsf{P}_m = \mathsf{P}_e$) [GS94; NM15a]. This relative motion is expressed in the *swing equation*:

$$\frac{2J}{\varphi_R}\ddot{\delta} + \frac{D}{\varphi_R}\dot{\delta} = \mathsf{P}_m - \mathsf{P}_e, \tag{2.6}$$

the fundamental equation of motion for a generator. The inertia constant J is the kinetic energy of the rotor divided by the rated power. The damping constant D is the combined damping coefficient which - simply put - merges several quantities which decrease or impede the rotor's oscillations. The angle δ is the rotor angle relative to a frame rotating at the reference frequency φ_R [NM15a]. The first and second derivative of δ thus correspond to the velocity (or frequency) and acceleration of the rotor, respectively [GS94]. The systems (2.1) and (2.6) represent a dynamic system for δ.

The swing equation constitutes the linchpin of our further examination. The left hand side (LHS) of the swing equation (2.6) is already in the form of the LHS of the ODE system (2.1). The constants J, D and φ_R are determined by the underlying physical power network. The right hand side (RHS), in contrast, depends on time and the model representation which is employed. The classical model in the next subsection furnishes us with alternative expressions for P_m and P_e.

The Classical Model The Electric Circuit Representation of a Generator

Under realistic conditions, both the mechanical and electric power show nonlinear behavior. The *classical model* incorporates these dynamics and is still relatively simple by making a few very important assumptions concerning the electric circuit[3]:

1. Each generator node is represented as a voltage source.
2. Each voltage source is linked to a terminal by a transient reactance $\mathsf{x}_r' > 0$.
3. Each voltage source has a constant voltage magnitude $|\mathsf{E}|$.
4. Each load node is represented as an impedance.
5. The mechanical power input to each rotor is constant.
6. The phase angle of the voltage source coincides with the mechanical rotor angle.

Constant voltage and mechanical power are sustainable for short time periods. These assumptions limit the complexity and computational costs of the PN system while at the same time depicting the network dynamics authentically [GS94; NM15a]. They are essential to find and compute the model-dependent parameters F, K, and γ, and remain valid throughout this thesis.

[3] In some descriptions of the classical model, damping D is neglected. In [NM15a] however, it is recognized as a parameter which can have conceivable effects on the steady-state stability and can be adjusted to optimize the stability.

An additional assumption in stability studies is the initial synchronization of all generators at the reference frequency φ_R in a steady state. This is legitimate since the system frequency is rigorously supervised and regulated to approximate φ_R in real life under steady-state conditions. This frequency management is done by balancing the mechanical power input to the generators. We then have $\ddot{\delta} = \dot{\delta} = 0$ and $\mathsf{P}_m = \mathsf{P}_e = \mathsf{P}_g^\star$, with $\mathsf{P}_g^\star$ being the active power which a generator is transferring through its terminal. With P_m thus remaining constant, the swing equation and the classical model can be applied in stability analysis.

Under steady and unsteady conditions, the power-angle equation

$$\mathsf{P}_e = \frac{|\mathsf{E}^\star \mathsf{V}|}{\mathsf{x}_r'} \sin(\delta - \phi) \tag{2.7}$$

describes the electric power output. Here, we differentiate between the complex voltages $\mathsf{V}_i = |\mathsf{V}_i| \exp(\mathrm{i}\phi_i)$ at a terminal, and $\mathsf{E}_i = |\mathsf{E}_i^\star| \exp(\mathrm{i}\delta_i)$ at an internal node i, which are both time-dependent. Substituting the new expressions for the mechanical and electric power in equation (2.6), we get

$$\frac{2J}{\varphi_R}\ddot{\delta} + \frac{D}{\varphi_R}\dot{\delta} = \underbrace{\mathsf{P}_g^\star}_{\mathsf{P}_m} - \underbrace{\frac{|\mathsf{E}^\star \mathsf{V}|}{\mathsf{x}_r'} \sin(\delta - \phi)}_{\mathsf{P}_e}, \tag{2.8}$$

which effectually models the dynamic behavior of a generator. The classical model incorporates the nonlinear dynamics by expressing the RHS of (2.6) with a sine term (among others) which we also find on the RHS of the ODE system (2.1).

As already mentioned, the constants J, D and x_r' are known as distinct physical and electrical features of a generator. The parameters $\mathsf{P}_g^\star$ and $|\mathsf{E}^\star|$, however, are contingent on the power flow distribution in a steady-state network. In order to calculate the internal voltage magnitude, we take the complex power equation, (2.2), $\mathsf{P}_g^\star + \mathrm{i}\mathsf{Q}_g^\star = \mathsf{V}\bar{\mathsf{I}}$, substitute I using Ohm's law (2.4), $\mathrm{i}\mathsf{x}_r'\mathsf{I} = \mathsf{E} - \mathsf{V}$, and properties of the complex conjugate:

$$|\mathsf{E}^\star|^2 = \left(\frac{\mathsf{P}_g^\star \mathsf{x}_r'}{|\mathsf{V}^\star|}\right)^2 + \left(|\mathsf{V}^\star| + \frac{\mathsf{Q}_g^\star \mathsf{x}_r'}{|\mathsf{V}^\star|}\right)^2, \tag{2.9}$$

where $|\mathsf{V}^\star|$ is the steady-state terminal voltage and $\mathsf{Q}_g^\star$ is the reactive power which the generator infuses to the terminal. The values of $\mathsf{P}_g^\star$, $\mathsf{Q}_g^\star$ and $|\mathsf{V}^\star|$ yield the state of the terminal node. In consequence, equations (2.8) and (2.9) establish a connection between the terminal and internal nodes as well as the dynamic behavior of the system's generator [NM15a]. From these equations, we can also deduce the characterizing quantities of the dynamic state of a node i: the powers P_i and Q_i, the voltage magnitude $|\mathsf{V}_i|$ or $|\mathsf{E}_i|$, respectively, and the phase angle ϕ_i or δ_i, respectively.

If the classical model is applied in stability analysis, $\mathsf{P}_g^\star$ and $|\mathsf{V}^\star|$ are usually given as input data. Because $|\mathsf{V}|$ is connected to $|\mathsf{E}|$ by the transient reactance x_r', we already have two of the necessary four defining quantities with respect to generators. In the next subsection, we turn to the question of how to obtain the steady-state values for $\mathsf{Q}_g^\star$ as well as $\delta^\star$ and $\phi^\star$. The steady-state phase angles can serve as initial conditions for (2.8), and in consequence as input data for system (2.1) [NM15a].

Getting the Necessary Input The Power Flow Equations

In a network with alternating current, power is a complex number with real (active) and complex (reactive) parts. There are several means to uniquely determine the steady state of an operating PN, the objective of power system analysis. One way is to inspect the power flow distribution through the links of the physical network. Another way is to focus on the nodes of the network. Employing Kirchhoff's laws (2.3), and the restatement of Ohm's law (2.5), the power flow state can be characterized by the *power flow equations*:

$$\text{(active power)} \qquad \mathsf{P}_i = \sum_{j=1}^{n} |\mathsf{V}_i \mathsf{V}_j \mathsf{Y}_{0ij}| \sin(\phi_i - \phi_j - \gamma_{0ij}), \quad i = 1 \dots n, \tag{2.10}$$

$$\text{(reactive power)} \qquad \mathsf{Q}_i = -\sum_{j=1}^{n} |\mathsf{V}_i \mathsf{V}_j \mathsf{Y}_{0ij}| \cos(\phi_i - \phi_j - \gamma_{0ij}), \quad i = 1 \dots n, \tag{2.11}$$

where n is the number of nodes, and $\gamma_{0ij} = \vartheta_{0ij} - \frac{\pi}{2}$ (more on $\boldsymbol{\vartheta}$ below). With (2.10) and (2.11), the steady state of a power flow is likewise dictated by four parameters for each node i: P_i and Q_i, whose sum $\mathsf{P}_i + \mathrm{i}\mathsf{Q}_i$ is the complex power injection at node i, and complex voltage $\mathsf{V}_i = |\mathsf{V}_i| \exp(\mathrm{i}\phi_i)$ with magnitude $|\mathsf{V}_i|$ and phase angle ϕ_i. The (complex-valued) admittance Y_{0ij} is the most interesting quantity in these equations with regard to the two CO models we focus on. It plays a decisive role in each of the model-dependent parameters. The construction of the admittance matrix is where they differ most. In general, the admittances of the physical network are assembled in the *admittance matrix*

$$\mathbf{Y}_0 = (\mathsf{Y}_{0ij}) = \begin{pmatrix} \mathsf{Y}_0^{gg} & \mathsf{Y}_0^{g\ell} \\ \mathsf{Y}_0^{\ell g} & \mathsf{Y}_0^{\ell\ell} \end{pmatrix}, \tag{2.12}$$

where the first n_g rows (columns) are separated from the last n_ℓ rows (columns). The number of generator nodes is denoted by n_g and the number of load nodes is denoted by n_ℓ. The entries are represented in polar form $\mathsf{Y}_{0ij} = |\mathsf{Y}_{0ij}| \exp(\mathrm{i}\vartheta_{0ij})$, where $\boldsymbol{\vartheta}$ is the matrix containing the phase angles of the respective admittances. An off-diagonal element Y_{0ij} is the negative of the admittance between nodes i and $j \neq i$. A diagonal element Y_{0ii} is the sum of all admittances connected to node i. As noted in Section 2.1, two nodes in a PN are connected by a transmission line. It has an impedance, which is represented by an equivalent admittance, the inverse of impedance. The structure of the physical network is thus inscribed in $\mathbf{Y}_0$.

To solve the power flow system (2.10) and (2.11) of $2n$ equations, two real parameters per node are necessary. There are common assumptions made in power system analysis concerning generator and load nodes on which quantities are given. In case i is a generator node, the active power is provided as $\mathsf{P}_i = \mathsf{P}^{\star}_{g,i}$, and the complex voltage magnitude as $|\mathsf{V}_i| = |\mathsf{V}^{\star}_i|$ (or as $|\mathsf{E}_i| = |\mathsf{E}^{\star}_i|$, if i is a generator internal node). Both values are assumed to be constant. It is common practice to define one generator node as a reference node. For this type of node, the phase angle ϕ_i is set to zero, and the value of the active power P_i is left unspecified. This is done to account for the unknown power losses in the system, e.g., in the transmission lines.

In case i is a load node, the active power is given as $\mathsf{P}_i = -\mathsf{P}^{\star}_{\ell,i}$ and the reactive power as $\mathsf{Q}_i = -\mathsf{Q}^{\star}_{\ell,i}$. Supplied with these parameter values and in addition with the admittance matrix $\mathbf{Y}_0$, the power flow equations (2.10) and (2.11) can be solved numerically, e.g., by adequate software tools.

To recap, the dynamic state of a network is identified by the four time-dependent parameters P_i, Q_i, $|\mathsf{V}_i|$ and ϕ_i, for each node i. Hence, we need four equations per node to describe its dynamics. Table 2.1 shows the node type and the respective known and unknown parameter values.

Table 2.1: Known and Unknown Parameter Values

type of node	known	unknown
load	P, Q	$|\mathsf{V}|$, ϕ
reference generator	ϕ, $|\mathsf{V}|$	P, Q
generator (terminal)	P, $|\mathsf{V}|$	Q, ϕ
generator (internal)	P, $|\mathsf{E}|$	Q, δ

The power flow equations (2.10) and (2.11) are valid at each point in time for time-dependent variables, but they only suffice to contribute two equations. This is accurate for both generator and load nodes. Regarding generators, we can replace one of the two given parameters, i.e., the constant power injection and the constant voltage magnitude, with a differential equation, namely the adapted swing equation (2.8) in conjunction with (2.9). Thus, we move from the steady-state power flow equations to a dynamic model, and account for the time-variation of a node's state. The last parameter is fixed by the commonly-made assumption that the reactive power Q_g is identical to its steady-state value $\mathsf{Q}^{\star}_g$ for short time periods. It can therefore be determined by the power flow equations. We justify this by assuming that the generator's reactive power injection into its terminal is invariant for short time intervals. In this manner, the four state variables[4]

[4] The internal voltage $\mathsf{E} = |\mathsf{E}| \exp(\mathrm{i}\delta)$ is linked to the terminal voltage $\mathsf{V} = |\mathsf{V}| \exp(\mathrm{i}\phi)$ by the transient reactance x'_r via Ohm's law (2.4).

$P_g, Q_g, |E|$ and δ can be formulated as functions of time, provided we have an initial condition and know the state variables for all times t at the other nodes [NM15a].

2.3 The Basic Power Network Model for Load Nodes

Finding two additional equations per load node is more difficult, due to the more unpredictable character of loads as mentioned at the beginning of this chapter. We present two approaches addressing the challenge of load modeling in this thesis. But before we focus on their individual concepts, we first make ourselves familiar with the so-called *Kron reduction*, a power engineering technique. Both the EN and SM model use it to remove specific "unnecessary" nodes.

Kron Reduction

If there is no external load or generator connected to a network node, its current injection I is always zero. Nodes of this kind can be removed by Kron reduction. The method is similar to Gaussian elimination, but deletes a redundant row and column completely rather than just making it all zero.

First, we cast Kirchhoff's current law (2.3) into the form of the following nodal-admittance-equations, which is an $(n \times n)$ system,

$$\begin{bmatrix} Y_{11} & \cdots & Y_{1(i-1)} & Y_{1i} & Y_{1(i+1)} & \cdots & Y_{1n} \\ \vdots & \cdots & \cdots & \ddots & \cdots & \cdots & \vdots \\ Y_{(i-1)1} & \cdots & Y_{(i-1)(i-1)} & Y_{(i-1)i} & Y_{(i-1)(i+1)} & \cdots & Y_{(i-1)n} \\ Y_{i1} & \cdots & \cdots & Y_{ii} & \cdots & \cdots & Y_{in} \\ Y_{(i+1)1} & \cdots & Y_{(i+1)(i-1)} & Y_{(i+1)i} & Y_{(i+1)(i+1)} & \cdots & Y_{(i+1)n} \\ \vdots & \cdots & \cdots & \ddots & \cdots & \cdots & \vdots \\ Y_{n1} & \cdots & Y_{n(i-1)} & Y_{ni} & Y_{n(i+1)} & \cdots & Y_{nn} \end{bmatrix} \begin{bmatrix} V_1 \\ \vdots \\ V_{i-1} \\ V_i \\ V_{i+1} \\ \vdots \\ V_n \end{bmatrix} = \begin{bmatrix} I_1 \\ \vdots \\ I_{i-1} \\ 0 \\ I_{i+1} \\ \vdots \\ I_n \end{bmatrix}. \tag{2.13}$$

Node i has zero current injection and is thus eliminated resulting in the $(n-1 \times n-1)$ system,

$$\begin{bmatrix} Y^{(\text{new})}_{11} & \cdots & Y^{(\text{new})}_{1(i-1)} & Y^{(\text{new})}_{1(i+1)} & \cdots & Y^{(\text{new})}_{1n} \\ \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ Y^{(\text{new})}_{(i-1)1} & \cdots & Y^{(\text{new})}_{(i-1)(i-1)} & Y^{(\text{new})}_{(i-1)(i+1)} & \cdots & Y^{(\text{new})}_{(i-1)n} \\ Y^{(\text{new})}_{(i+1)1} & \cdots & Y^{(\text{new})}_{(i+1)(i-1)} & Y^{(\text{new})}_{(i+1)(i+1)} & \cdots & Y^{(\text{new})}_{(i+1)n} \\ \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ Y^{(\text{new})}_{n1} & \cdots & Y^{(\text{new})}_{n(i-1)} & Y_{n(i+1)} & \cdots & Y_{nn} \end{bmatrix} \begin{bmatrix} V_1 \\ \vdots \\ V_{i-1} \\ V_{i+1} \\ \vdots \\ V_n \end{bmatrix} = \begin{bmatrix} I_1 \\ \vdots \\ I_{i-1} \\ I_{i+1} \\ \vdots \\ I_n \end{bmatrix}. \tag{2.14}$$

The voltage and admittance information of the nodes with zero current injection does not get lost during the elimination procedure. In fact, it is incorporated in the remaining nodes. The values for $\mathsf{Y}_{jk}^{(\text{new})}$ are obtained by Algorithm 2.1 [BV00; GS94].

Algorithm 2.1: Kron Reduction (cf. [BV00])

```
Input: nodal-admittance-system as in (2.13) where node $i$ has zero current
       injection.
Output: reduced admittance-nodal-system as in (2.14).
Express $\mathsf{V}_i$ in terms of the other voltages:
    $\mathsf{V}_i = -\sum_{k=1,\, k\neq i}^{n} \frac{\mathsf{Y}_{ik}}{\mathsf{Y}_{ii}} \mathsf{V}_k.$
Substitute $\mathsf{V}_i$ in all equations for $\mathsf{I}_j$, for all $j \neq i$:
    $\mathsf{I}_j = \sum_{k=1,\, k\neq i}^{n} \mathsf{Y}_{jk}\mathsf{V}_k + \mathsf{Y}_{ji}\left(-\sum_{k=1,\, k\neq i}^{n} \frac{\mathsf{Y}_{ik}}{\mathsf{Y}_{ii}}\mathsf{V}_k\right)$
    $\quad = \sum_{k=1,\, k\neq i}^{n} \Bigg(\underbrace{\mathsf{Y}_{jk} - \frac{\mathsf{Y}_{ji}\mathsf{Y}_{ik}}{\mathsf{Y}_{ii}}}_{\mathsf{Y}_{jk}^{(\text{new})}}\Bigg)\mathsf{V}_k.$
Arrange the terms such that the matrix and vectors in (2.14) are generated and
  replace the admittance term with
    $\mathsf{Y}_{jk}^{(\text{new})} = \mathsf{Y}_{jk} - \frac{\mathsf{Y}_{ji}\mathsf{Y}_{ik}}{\mathsf{Y}_{ii}} \qquad j,k = 1,\dots,n \qquad j,k \neq i \qquad (2.15)$
```

Algorithm 2.1 has some noteworthy features [BV00; DB13; GS94]:

- It can be repeated iteratively for all $\mathsf{I}_i = 0$.
- It can destroy sparsity.
- The reduced admittance matrix can be computed directly with equation (2.15).
- Kron reduction and node elimination of a system are interchangeable notions.
- The connectivity of the network is preserved.
- Two nodes j and k are connected in the Kron reduced network if and only if there was a path of intermediate nodes between j and k in the original network and all these intermediate nodes have been eliminated.

We will refer to this procedure in the subsequent sections on the individual model's conception of load nodes.

Preliminaries on the Construction of the Coupled Oscillator Representation

The EN and SM models are indeed very similar. They are mathematically equivalent and take the same substantial steps in load node construction. The assumptions for the classical model on page 9 hold for both models; assumptions $1. - 5.$ are especially important to keep in mind during the construction. We outline the general procedure here and expand on the details in the subsequent sections on the EN and SM models.

Given a PN system in its physical representation, they first transform the network into its electric circuit representation. During this step, they substitute the load nodes with some other type of network component. The EN and SM models differ most in their choice of the electric circuit (EC) component into which each load node should be recast.

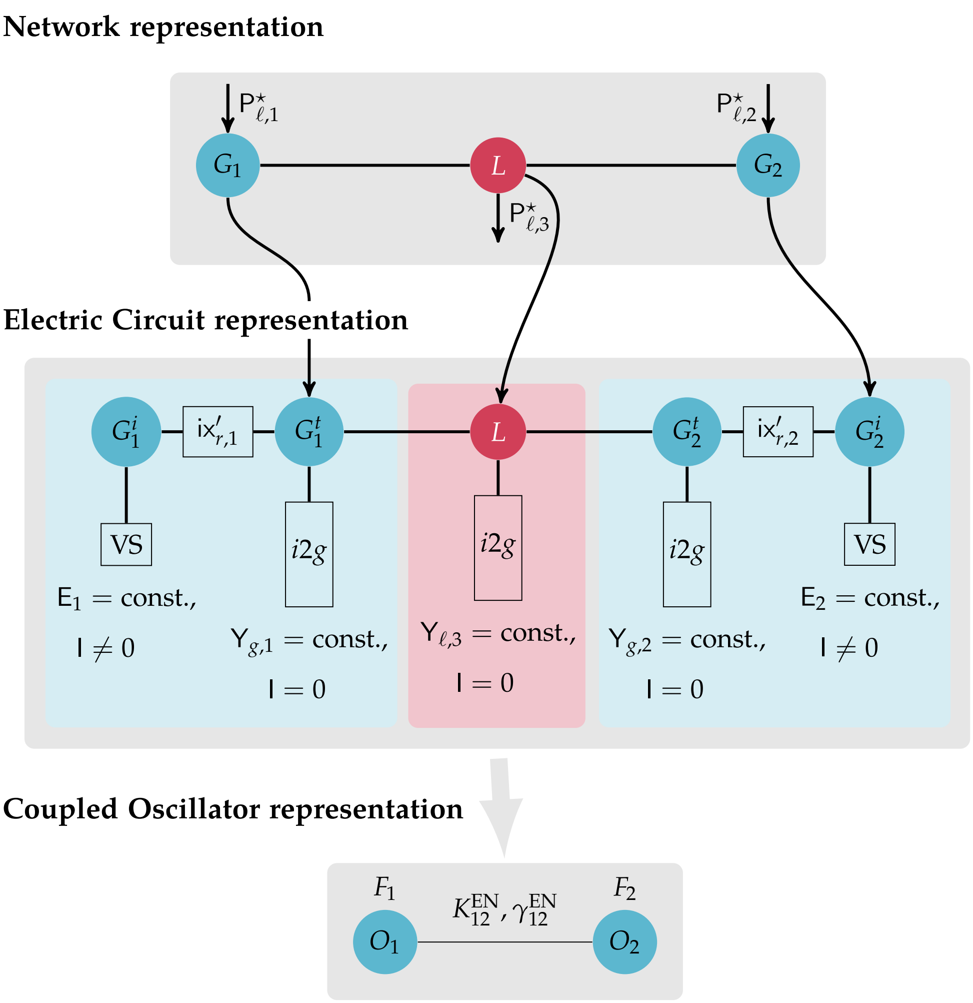
In the process, they expand the network by adding generator internal nodes. The load nodes in the created electric circuit representation of the network are reconstructed as constant impedances to the ground (or equivalent admittances). This implies that these nodes have zero current injection and they are removed via Kron reduction. After the elimination, only the internal nodes are left in the coupled oscillator representation of the network. They all have equivalent mathematical modeling, and depict the generator dynamics as coupled phase oscillators.

2.3.1 Effective Network Model

Assuming constant power demand, the EN model redefines the load nodes in the physical network as constant impedances to the ground, i.e., nodes with zero current injection. All nodes except generator internal nodes are then eliminated through Kron reduction. This can be interpreted as treating the load nodes as transmission lines (which also have impedance), so that the generators are directly linked. The idea is motivated by the observation that the dynamic behavior between two generator nodes nonlinearly depends on the terminal voltage phase in the adapted swing equation (2.8). The equation is merely the mathematical expression of the physical path of transmission lines connecting the generators. The EN model reconstructs this path in such a manner that the coupling between generators can be expressed in a single term depending solely on the generators' state variables [MMA^{+}13; NM15a].

Figure 2.1 depicts an example modeling process for two generator (G_1, G_2) and one load node (L) in the network representation. References to the figure are in parentheses during the following description of the modeling.

Recall that n_g is the number of generator terminal nodes and n_ℓ is the number of load nodes with $n = n_g + n_\ell$ in the physical representation. We start by first enlarging the network. In the electric circuit representation, we add one generator internal node (G_1^i, G_2^i) for each generator terminal node (G_1^t, G_2^t) existing in the physical network. This new kind of node is defined as a point between the internal transient reactance $(\mathrm{i}x_r')$



G:	generator node	P:	power
L:	load node	ix'_r:	transient reactance
O:	oscillator node	E:	internal voltage
VS:	voltage source	Y:	equivalent admittance
$i2g$:	impedance to the ground	I:	current injection

Figure 2.1: Power Network Modeling Process of the EN Model

and the constant voltage source (VS with $|E| = \text{const.}$). The total number of nodes in the electric circuit representation network has now increased to $n_{ec} = 2n_g + n_\ell$. By reordering, the indices $i = 1, \ldots, n_g$ account for the newly created generator internal nodes, $i = n_g + 1, \ldots, 2n_g$ for the generator terminal nodes, and $i = 2n_g + 1, \ldots, n_{ec}$ for the load nodes.

We now return to the admittance matrix $\mathbf{Y}_0$ in (2.12) of the physical network,

$$\mathbf{Y}_0 = (Y_{0ij}) = \begin{pmatrix} Y_0^{gg} & Y_0^{g\ell} \\ Y_0^{\ell g} & Y_0^{\ell\ell} \end{pmatrix},$$

and reconstruct it to fit the electric circuit representation.

In steady state, a load node (L) (and generator terminal nodes (G_1^t, G_2^t) are treated in the same way) uses active and reactive powers $\mathsf{P}_{\ell,i}^{\star}$ and $\mathsf{Q}_{\ell,i}^{\star}$. The node is redefined as a constant impedance to the ground $(i2g)$ and has equivalent admittance

$$\mathsf{Y}_{\ell,i} = \frac{\mathsf{P}_{\ell,i}^{\star} - \mathrm{i}\mathsf{Q}_{\ell,i}^{\star}}{|\mathsf{V}_i^{\star}|^2}.$$

Remember, that constant admittance implies zero current injection $(\mathsf{I} = 0)$. A node's voltage magnitude $|\mathsf{V}_i^{\star}|^2$ can be calculated from the power flow equations (2.10) and (2.11) with $\mathsf{P}_{\ell,i}^{\star}$ and $\mathsf{Q}_{\ell,i}^{\star}$ as input data. Assuming constant power demand, the constant admittance $\mathsf{Y}_{\ell,i}$ can thus be determined. We add these values to the respective diagonal elements of the matrix blocks $\mathbf{Y}_0^{gg}$ and $\mathbf{Y}_0^{\ell\ell}$ to get $\tilde{\mathbf{Y}}_0^{gg}$ and $\tilde{\mathbf{Y}}_0^{\ell\ell}$.

Another admittance matrix we need is the $(n_g \times n_g)$ diagonal matrix $\mathbf{Y}_d^{\mathrm{EN}}$ with the generator transient reactances $(\mathrm{i}x'_{d,1})^{-1}, \dots, (\mathrm{i}x'_{d,n_g})^{-1}$ as its diagonal elements. The purpose of $\mathbf{Y}_d^{\mathrm{EN}}$ is to link the new generator internal (G_1^i, G_2^i) to the preexisting generator terminal nodes (G_1^t, G_2^t).

The admittance matrix $\mathbf{Y}^{EC'} = \mathsf{Y}_{ij}^{EC'}$ of size $(2n_g + n_\ell \times 2n_g + n_\ell)$ is composed of these different kinds of admittance matrices. It comprises the transient reactances along with the equivalent impedances for the generator terminal and load nodes. Therefore, it represents the network structure of the electric circuit representation:

$$\mathbf{Y}^{EC'} = \begin{bmatrix} \mathbf{Y}_d^{\mathrm{EN}} & -\mathbf{Y}_d^{\mathrm{EN}} & \mathbf{0} \\ -\mathbf{Y}_d^{\mathrm{EN}} & \tilde{\mathbf{Y}}_0^{gg} + \mathbf{Y}_d^{\mathrm{EN}} & \mathbf{Y}_0^{g\ell} \\ \mathbf{0} & \mathbf{Y}_0^{\ell g} & \tilde{\mathbf{Y}}_0^{\ell\ell} \end{bmatrix}, \tag{2.16}$$

where $\mathbf{0}$ is the matrix of all zeros of appropriate size.

The last major step in the construction of the EN admittance matrix is to use Kron reduction to eliminate all nodes with zero current injection. Since we just redefined the generator terminal and load nodes as constant impedances to the ground with equivalent constant admittances, all nodes (G_1^t, G_2^t and L) have zero current injection except the generator internal nodes (G_1^i, G_2^i). Let $\mathbf{V}^{g^i}$, $\mathbf{V}^g$ and $\mathbf{V}^\ell$ be the voltage vectors corresponding to the generator internal nodes, generator terminal and load nodes, respectively, and concatenate them to form the voltage vector $\mathbf{V}$. We assume the voltage vector is unique. Similar to the nodal-admittance-equations in (2.13), we write Kirchhoff's current law (2.3) as

$$\begin{bmatrix} \mathbf{Y}_d^{\mathrm{EN}} & -\mathbf{Y}_d^{\mathrm{EN}} & \mathbf{0} \\ -\mathbf{Y}_d^{\mathrm{EN}} & \tilde{\mathbf{Y}}_0^{gg} + \mathbf{Y}_d^{\mathrm{EN}} & \mathbf{Y}_0^{g\ell} \\ \mathbf{0} & \mathbf{Y}_0^{\ell g} & \tilde{\mathbf{Y}}_0^{\ell\ell} \end{bmatrix} \begin{bmatrix} \mathbf{V}^{g^i} \\ \mathbf{V}^g \\ \mathbf{V}^\ell \end{bmatrix} = \begin{bmatrix} \mathbf{I}^{g^i} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}. \tag{2.17}$$

After applying Kron reduction and thereby deleting all nodes relating to $\mathbf{V}^g, \mathbf{V}^\ell$, the system is transformed to $\mathbf{Y}\mathbf{V}^g = \mathbf{I}^{g^i}$. Employing the Kron reduction formula (2.15), the Effective Network admittance matrix $\mathbf{Y}^{\mathrm{EN}} = (\mathsf{Y}_{ij}^{\mathrm{EN}})$ is defined by

$$\mathbf{Y}^{\mathrm{EN}} = \mathbf{Y}'(\mathbb{I} + (\mathbf{Y}_d^{\mathrm{EN}})^{-1}\mathbf{Y}')^{-1}) \qquad \text{with} \qquad \mathbf{Y}' = \tilde{\mathbf{Y}}_0^{gg} - \tilde{\mathbf{Y}}_0^{g\ell}(\tilde{\mathbf{Y}}_0^{\ell\ell})^{-1}\tilde{\mathbf{Y}}_0^{\ell g}, \tag{2.18}$$

where $\mathbb{I}$ is the $(n_g \times n_g)$ identity matrix. The existence of matrix $(\mathbb{I} + (\mathbf{Y}_d^{\mathrm{EN}})^{-1}\mathbf{Y}')^{-1}$ and $\tilde{\mathbf{Y}}_0^{\ell\ell}$ hinges on sufficiently small transient reactances $x'_{d,i}$. The assumption of uniqueness of the voltage vectors guarantees the existence of the inverse of $(\tilde{\mathbf{Y}}_0^{\ell\ell})^{-1}$. The inverse of $\mathbf{Y}_d^{\mathrm{EN}}$ always exists.

The network is now modeled as a system of n_{co} coupled oscillators (O_1, O_2), and the admittance matrix $\mathbf{Y}^{\mathrm{EN}}$ embodies the structure of the coupled oscillator representation of the network. It assists us to find a different form of P_e in the adapted swing equation (2.8). We substitute $\mathbf{Y}_0$ for $\mathbf{Y}^{\mathrm{EN}}$, $|\mathsf{E}_i^\star|$ for $|\mathsf{V}_i|$, δ_i for φ_i in that equation, and obtain an expression which is derived from a power balance equation synonymous to equation (2.10):

$$\mathsf{P}_{e,i} = \sum_{j=1}^{n_g} |\mathsf{E}_i^\star \mathsf{E}_j^\star \mathsf{Y}_{ij}^{\mathrm{EN}}| \cos(\delta_j - \delta_i + \vartheta_{ij}^{\mathrm{EN}}), \tag{2.19}$$

where $\mathsf{Y}_{ij}^{\mathrm{EN}} = |\mathsf{Y}_{ij}^{\mathrm{EN}}| \exp(\mathrm{i}\vartheta_{ij}^{\mathrm{EN}})$. We thus have found a replacement for the coupling term $\frac{|\mathsf{E}^\star \mathsf{V}|}{\mathsf{x}'_r}$ in (2.8). Using equation (2.19), the swing equation (2.6) for each generator can be rewritten such that we have an expression of the same pattern as equation (2.1). Hence, the EN model is defined by $n_{co} = n_g$ ODEs of the form

$$\frac{2J_i}{\varphi_R}\ddot{\delta}_i + \frac{D_i}{\varphi_R}\dot{\delta}_i = F_i^{\mathrm{EN}} - \sum_{j=1,\, j\neq i}^{n_g} K_{ij}^{\mathrm{EN}} \sin(\delta_i - \delta_j - \gamma_{ij}^{\mathrm{EN}}), \quad i = 1, \ldots, n_{co}, \tag{2.20}$$

with

$$\begin{aligned} F_i^{\mathrm{EN}} &= \mathsf{P}_{g,i}^\star - |\mathsf{E}_i^\star|^2 \mathrm{Re}(\mathsf{Y}_{ii}^{\mathrm{EN}}), & K_{ij}^{\mathrm{EN}} &= |\mathsf{E}_i^\star \mathsf{E}_j^\star \mathsf{Y}_{ij}^{\mathrm{EN}}|, \\ \gamma_{ij}^{\mathrm{EN}} &= \vartheta_{ij}^{\mathrm{EN}} - \frac{\pi}{2}, & \mathsf{Y}_{ij}^{\mathrm{EN}} &= |\mathsf{Y}_{ij}^{\mathrm{EN}}| \exp(\mathrm{i}\vartheta_{ij}^{\mathrm{EN}}), \end{aligned} \tag{2.21}$$

and $\vartheta_{ij}^{\mathrm{EN}}$ being the phase angle of the EN model's admittance matrix elements. The internal voltage magnitude $|\mathsf{E}_i^\star|$ is assumed to be constant. Given a steady-state power flow, it is computed with equation (2.9) for each generator [NM15a].

2.3.2 Synchronous Motor Model

The trick of the Synchronous Motor model is to redefine the load nodes as synchronous motors[5]. A motor is essentially a generator with reverse power flow, i.e., it translates electric into mechanical power. For this reason, a load node can be conveniently modeled by the swing equation for generators (2.6) with $\mathsf{P}_m < 0$ and $\mathsf{P}_e < 0$. Figure 2.2 illustrates the modeling process, again for two generator (G_1, G_2) and one load node (L), to which we refer to in parentheses in the modeling description.

Similar to the EN model, to construct the electric circuit representation, therefore define n_g additional internal nodes (G_1^i, G_2^i) for the generators. Because each load node is represented by a synchronous motor, i.e., a generator with negative mechanical power P_m, we also add n_ℓ internal nodes for the loads (L^i). The generator terminal (G_1^t, G_2^t) and load nodes (L) in the physical representation are reconstructed as constant impedances

[5] In the literature, you often encounter the term "synchronous machines".

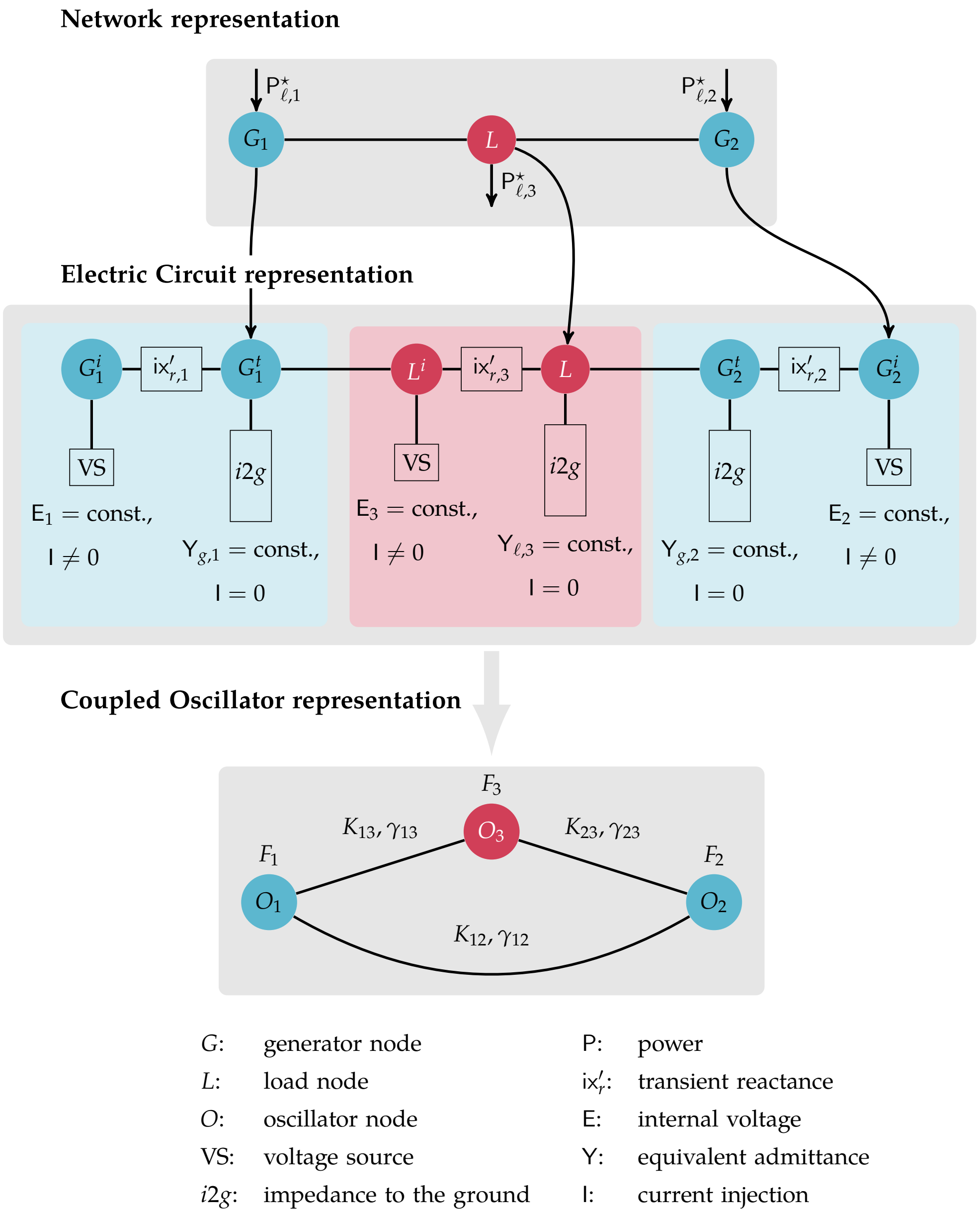


Figure 2.2: Power Network Modeling Process of the SM Model

to the ground ($i2g$) with zero current injection. They are once more equated with transmission lines connecting the new generator and load internal nodes. In consequence, the mathematical formulation of the SM model is identical to the EN model and the derivation of the SM model's admittance matrix is analogous. The SM model's electric circuit representation of the network has a total number of $n_{ec} = 2n_g + 2n_\ell = 2n$ nodes.

Again, we change the node order such that indices $i = 1, \dots, n_g$ indicate the generator internal nodes, $i = n_g + 1, \dots, n$ the load internal nodes, $i = n + 1, \dots, n + n_g$ generator terminal nodes, and $i = n + n_g + 1, \dots, 2n$ load nodes. The diagonal admittance matrix $\mathbf{Y}^{\text{SM}}_d$ is now an $(n \times n)$ matrix with the generator and load transient reactances $(ix'_{d,1})^{-1}, \dots, (ix'_{d,n})^{-1}$ on its diagonal. Parallel to the EN model, $\mathbf{Y}^{\text{SM}}_d$ is responsible for linking the added generator internal and load internal nodes to the generator terminal and load nodes in the underlying physical network.

The admittance matrix $\mathbf{Y}^{EC''} = \mathsf{Y}_{ij}^{EC''}$ of size $(2n \times 2n)$ encodes the network structure in the SM model's electric circuit representation. Just like $\mathbf{Y}^{EC'}$ above, it includes the transient reactances of the internal and the equivalent impedances of the terminal and load nodes:

$$\mathbf{Y}^{EC''} = \begin{bmatrix} \mathbf{Y}_d^{\text{SM}} & -\mathbf{Y}_d^{\text{SM}} \\ -\mathbf{Y}_d^{\text{SM}} & \mathbf{Y}_0 + \mathbf{Y}_d^{\text{SM}} \end{bmatrix}. \tag{2.22}$$

Once more, we perform Kron reduction to delete all generator terminal and load nodes (G_1^t, G_2^t and L) since these have zero current injection. To this end, let $\mathbf{V}^{g^i+\ell^i}$ be the voltage vector corresponding to the internal nodes and $\mathbf{V}^{g+\ell}$ be the voltage vector corresponding to the generator terminal and load nodes. Kirchhoff's current law (2.3) is then written as

$$\begin{bmatrix} \mathbf{Y}_d^{\text{SM}} & -\mathbf{Y}_d^{\text{SM}} \\ -\mathbf{Y}_d^{\text{SM}} & \mathbf{Y}_0 + \mathbf{Y}_d^{\text{SM}} \end{bmatrix} \begin{bmatrix} \mathbf{V}^{g^i+\ell^i} \\ \mathbf{V}^{g+\ell} \end{bmatrix} = \begin{bmatrix} \mathbf{I}^{g^i+\ell^i} \\ \mathbf{0} \end{bmatrix}. \tag{2.23}$$

Kron reduction cuts the system size to $(n \times n)$. We get the Synchronous Motor admittance matrix $\mathbf{Y}^{\text{SM}} = (\mathsf{Y}_{ij}^{\text{SM}})$ by using the Kron reduction formula (2.15):

$$\mathbf{Y}^{\text{SM}} = \mathbf{Y}_d^{\text{SM}}(\mathbb{I} - \mathbf{Y}_d^{\text{SM}}(\mathbf{Y}_0 + \mathbf{Y}_d^{\text{SM}})^{-1}), \tag{2.24}$$

where $\mathbb{I}$ is the $(n \times n)$ identity matrix. The structure of the coupled oscillator representation is now inscribed in the entries of $\mathbf{Y}^{\text{SM}}$, and all nodes are expressed as coupled oscillators (O_1, O_2, O_3). By again replacing $\mathbf{Y}_0$ for $\mathbf{Y}^{\text{SM}}$, $|\mathsf{E}_i^\star|$ for $|\mathsf{V}_i|$, δ_i for φ_i in the adapted swing equation (2.9), we obtain a new statement for the active power identical to (2.19). With $n_{co} = n$, the SM model is given by the ODE system

$$\frac{2J_i}{\varphi_R}\ddot{\delta}_i + \frac{D_i}{\varphi_R}\dot{\delta}_i = F_i^{\text{SM}} - \sum_{j=1,\, j\neq i}^{n} K_{ij}^{\text{SM}} \sin(\delta_i - \delta_j - \gamma_{ij}^{\text{SM}}), \quad i = 1, \dots, n, \tag{2.25}$$

with

$$\begin{aligned} &F_i^{\text{SM}} = \mathsf{P}_{g,i}^\star - |\mathsf{E}_i^\star|^2 \text{Re}(\mathsf{Y}_{ii}^{\text{SM}}), && i = 1, \dots, n_g \text{ for generators,} \\ &F_i^{\text{SM}} = -\mathsf{P}_{g,i}^\star - |\mathsf{E}_i^\star|^2 \text{Re}(\mathsf{Y}_{ii}^{\text{SM}}), && i = n_g + 1, \dots, n \text{ for loads/motors,} \\ &K_{ij}^{\text{SM}} = |\mathsf{E}_i^\star \mathsf{E}_j^\star Y_{ij}^{\text{SM}}|, \quad \gamma_{ij}^{\text{SM}} = \vartheta_{ij}^{\text{SM}} - \frac{\pi}{2}, && \mathsf{Y}_{ij}^{\text{SM}} = |\mathsf{Y}_{ij}^{\text{SM}}| \exp(\mathrm{i}\vartheta_{ij}^{\text{SM}}), \end{aligned} \tag{2.26}$$

and $\vartheta_{ij}^{\text{SM}}$ being the phase angle of the SM model's admittance matrix elements. The constant internal voltage magnitude $|\mathsf{E}_i^\star|$ is likewise computed with equation (2.9) for each generator [NM15a].

2.3.3 Model-Dependent Parameters: Characterization and Comparison

From the derivation above, we see that the CO model equations are ODE systems of swings equations with modeled parameters for the RHSs. The objective of this section is to characterize the model-dependent parameters F, K and γ.

Let the superscript *mod* $\in \{\mathrm{EN}, \mathrm{SM}\}$ indicate model-specific parameters. Because both models recast the PN nodes in the form of coupled oscillators with mathematically identical structure, they both adhere to the following pattern in their definition of the model-dependent parameters:

$$\begin{aligned} F_i^{mod} &= \pm \mathsf{P}^\star_{g/\ell,i} - |\mathsf{E}_i^\star|^2 \mathrm{Re}(\mathsf{Y}_{ii}^{mod}), \qquad & K_{ij}^{mod} &= |\mathsf{E}_i^\star \mathsf{E}_j^\star \mathsf{Y}_{ij}^{mod}|, \\ \gamma_{ij}^{mod} &= \vartheta_{ij}^{mod} - \frac{\pi}{2}, & \mathsf{Y}_{ij}^{mod} &= |\mathsf{Y}_{ij}^{mod}| \exp(j\vartheta_{ij}^{mod}), \end{aligned} \tag{2.27}$$

where $i, j = 1, \ldots, n_{co}$, with $n_{co} = n_g$ for the EN model and $n_{co} = n_g + n_\ell = n$ for the SM model.

Remark 2.1. The two CO models can vary considerably in size depending on the n_g-n_ℓ-ratio. Assuming that every PN system has at least one load node, the SM model is obviously always larger than the EN model. Since a PN system usually has a lot more load nodes than generator nodes, we may expect that both building and reducing the SM model, as well as solving the ODE system requires more time and memory compared to the EN model.

The inherent frequency – the equilibrium frequency of an oscillator in the absence of the coupling term K in (2.1) – is calculated as $\varphi^\star = \varphi_R(1 + \frac{F}{D})$. So, the term model-dependent term F determines an oscillator's inherent frequency along with the network-dependent constant term D. The coupling term K depicts the strength of the dynamic coupling between two oscillators and is always nonnegative [NM15a]. It holds, $K_{ij} \neq 0$ if oscillators i and j are connected, and $K_{ij} = 0$ otherwise. So, it is comparable to an adjacency matrix [FNP08]. The term $K_{ij} \sin(\delta_i - \delta_j - \gamma_{ij})$ represents the power flow along transmission lines [DCB13]. The parameter γ in turn, is the phase shift involved in the oscillator coupling and encodes the coupling structure [NM15a].

On the one hand, each oscillator i is inclined to align with its preferred inherent frequency $\varphi^\star$. On the other hand, the coupling term $K_{ij} \sin(\delta_i - \delta_j - \gamma_{ij})$ drives an oscillator to synchronize with its neighbors. The struggle for supremacy of these two forces determines the dynamic behavior of the PN ODE system (2.1). An equilibrium of the PN network is reached in (2.1), if all oscillators rotate with the same inherent frequency $\varphi^\star{}_i$ and all angles δ_i are aligned. Then we have

$$\dot{\delta}_1 = \dot{\delta}_2 = \cdots = \dot{\delta}_{n_g},$$

i.e., the power network is synchronized.

Different inherent frequencies of the coupled oscillators disturb this equilibrium. The coupling strength K_{ij} also affects these dynamics. If the oscillators are strongly coupled and rotate with approximately equal inherent frequencies, they are more inclined to synchronize. In the reverse case, i.e., the oscillators' inherent frequencies are heavily different and the network is weakly coupled, synchronization is much harder to occur [DCB13].

Based on their research, [NM15a] have implemented an open-source MATLAB toolbox called *pg_sync_models* [NM15b] which computes the model-dependent parameters F, K and γ in the ODE system of swing equations (2.1). It uses another package, MATPOWER [ZM16], to solve the power flow equations and includes a routine to estimate the parameters J, D and x_r' which are not always provided in a power system dataset. By using these toolboxes, we have all the necessary quantities to solve system (2.1).

While the two CO models share the same computational roots, i.e., the swing equation, the classical model, and the power flow equations, it is still interesting to look at both models. In their paper, [NM15a] demonstrate that despite their application to the same network case, their respective CO model representations can feature utterly different parameters F, K, γ. Likewise, applying one model on different datasets exposes dissimilar characteristics of the model parameters. In conclusion, the valid application of these models depends heavily on the intended goal, further assumptions, and power system selection [NM15a].

CHAPTER 3

Model Order Reduction Methods for Dynamic Systems

We begin by introducing the overall problem-setup and idea of Model Order Reduction, before turning our attention to the specifics of the Balanced Truncation and Proper Orthogonal Decomposition. First, we unfold the theory for linear systems, which has been well studied and MOR methods have been fully fleshed out for this case (see, e.g., [Ant05]). Because of the difficulties arising from nonlinear systems, the MOR theory is not as evolved for this type of systems, although important nonlinear-specific concepts have been developed, especially since the '90s (see, e.g., [GM96; Sch93]). So, while primarily looking at the linear case, we point out the relevant differences particular to nonlinear systems.

Throughout this thesis, we only consider continuous (i.e., time t is a real number), time-invariant systems of finite dimension.

3.1 Model Order Reduction

Problem Setup

Consider a dynamic system of (explicit) 1^{st}-order differential equations with the state space $\mathbb{X} \subseteq \mathbb{R}^n$, the input space $\mathbb{U} \subseteq \mathbb{R}^m$ and the output/observation space $\mathbb{Y} \subseteq \mathbb{R}^p$. It is described by state equations of the form

$$\begin{aligned} \dot{x}(t) &= f(x(t)) + g(x(t), u(t)), \\ y(t) &= h(x(t), u(t)), \\ x(0) &= x_0, \end{aligned} \tag{3.1}$$

with variables

$x(t) \in \mathbb{X}$	state vector,
$u(t) \in \mathbb{U}$	input or excitation vector,
$y(t) \in \mathbb{Y}$	output or observation vector.

The (linear or nonlinear) functions $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ model the dynamic behavior of the system, whereas $h : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$ specifies the transformation from the state and input to the output [Ant05; BG17]. In case, (3.1) is a linear system, it can be represented using matrices in the following way:

Definition 3.1 (**State Space Description of (Linear) Dynamic System**, [Ant05]). The state space description of a linear system is composed of matrices[1] $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}$ and $C \in \mathbb{R}^{p\times n}$, such that

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t), \\ x(0) &= x_0, \end{aligned} \tag{3.2}$$

with $x(t) \in \mathbb{R}^n, u(t) \in \mathbb{R}^m$, and $y(t) \in \mathbb{R}^p$, and a given initial value x_0. We concisely denote this system by

$$\Sigma = \left(\begin{array}{c|c} A & B \\ \hline C & \end{array}\right) \qquad \in \mathbb{R}^{(n+p)\times(n+m)}. \tag{3.3}$$

Corresponding to the respective functions, A and B are the state dynamics and input matrices, respectively, and C is the output matrix.

Definition 3.2 (**Complexity, Order**, [Ant05; AS01]). The *complexity*, or *order* of a system (3.1) is defined as the number n of state variables included, i.e., the size of the state vector $x = (x_1, \cdots, x_n)^*$, or the number of state equations, respectively.

Using the matrix exponential defined by

$$e^{Mt} = \mathbb{I}_n + \frac{t}{1!}M + \frac{t^2}{2!}M^2 + \cdots + \frac{t^k}{k!}M^k + \cdots, \tag{3.4}$$

for a matrix $M \in \mathbb{R}^{n\times n}$ and a scalar $t \in \mathbb{R}$, the solution of the state equations in system (3.2) at time $t \geqslant t_0$ is

$$\psi(u; x_0; t) = e^{A(t-t_0)}x_0 + \int_{t_0}^{t} e^{At-\tau}Bu(\tau)d\tau, \tag{3.5}$$

provided an input u, and an initial condition $x(0) = x_0$ at time t_0. The output is accordingly,

$$y(t) = C\psi(u; x(t_0); t) = C\psi(0; x(t_0); t) + C\psi(u; 0; t) \tag{3.6}$$

accordingly, [Ant05].

Idea of Model Order Reduction

In the introduction in Chapter 1, we mentioned some of the challenges posed by large-scale system, e.g., constraints on computation time, storage and accuracy. To circumvent

[1] The output equations can have an additional term $D : \mathbb{R}^m \to \mathbb{R}^p$ mapping u, but since this map is irrelevant in this thesis, we omit it.

these impairments, a simplified model is needed which can be used instead of the original system for simulation or control purposes.

The objective of Model Order Reduction is to find a dynamic system of lower order $n_\rho \ll n$ that can be used instead of the FOM. This reduced-order model can be represented in form of a system of equations relating to system (3.1),

$$\begin{aligned} \dot{\hat{x}}(t) &= \hat{f}(\hat{x}(t)) + \hat{g}(\hat{x}(t), u(t)), \\ \hat{y}(t) &= \hat{h}(\hat{x}(t), u(t)), \\ \hat{x}(0) &= \hat{x}_0, \end{aligned} \tag{3.7}$$

with

$$\hat{f} : \mathbb{R}^{n_\rho} \to \mathbb{R}^{n_\rho}, \quad \hat{g} : \mathbb{R}^{n_\rho} \times \mathbb{R}^m \to \mathbb{R}^{n_\rho}, \quad \hat{h} : \mathbb{R}^{n_\rho} \times \mathbb{R}^m \to \mathbb{R}^p,$$

or, corresponding to (3.2), a system of matrices,

$$\begin{aligned} \dot{\hat{x}}(t) &= \hat{A}\hat{x}(t) + \hat{B}u(t), \\ \hat{y}(t) &= \hat{C}\hat{x}(t), \\ \hat{x}(0) &= \hat{x}_0, \end{aligned} \tag{3.8}$$

with matrices $\hat{A} \in \mathbb{R}^{n_\rho \times n_\rho}$, $\hat{B} \in \mathbb{R}^{n_\rho \times m}$ and $\hat{C} \in \mathbb{R}^{p \times n_\rho}$.

Definition 3.3 (**(Linear) reduced-order model**, [Ant05])**.** We denote the *reduced-order model* of the full-order model Σ in (3.3) by

$$\hat{\Sigma} = \left(\begin{array}{c|c} \hat{A} & \hat{B} \\ \hline \hat{C} & \end{array}\right) \qquad \in \mathbb{R}^{(n_\rho+p)\times(n_\rho+m)}, \tag{3.9}$$

with *reduction order* $n_\rho \ll n$.

Figure 3.1 illustrates the MOR process for linear systems.

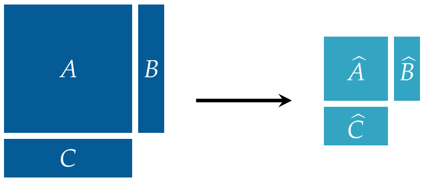


Figure 3.1: Graphical Depiction of MOR Process for Linear Systems

The ROM should approximate the original FOM (3.3) such that it meets certain requirements, for example:[2]: (1) The foremost is, of course, that the ROM is of lower complexity, i.e., $n_\rho \ll n$. (2) The error between the full- and the reduced-order system is small. The existence of a global error bound is also desirable. (3) System properties

[2] A more elaborate list can be found in [Ant05; AS01].

like stability are maintained. (4) The computation itself is stable and efficient [Ant05; AS01; BG17].

The first two requirements considered together imply that the MOR method should provide some decision-making tool or measure to determine which states to discard and which ones to preserve, i.e., which states are either too "important" to cut and/or which states are "not important enough" to keep in order to achieve a small reduction error.

Essential Methods

The most fundamental technique needed in MOR is the state transformation. The intention is to bring the system into a new form such that the MOR method can take advantage of certain structural properties. We require a nonsingular matrix T transforming the state variable

$$\hat{x} = \mathrm{T}x, \tag{3.10}$$

and applied to the system matrices accordingly, system (3.2) is transformed into

$$\begin{aligned} \dot{\hat{x}} &= \underbrace{\mathrm{T}A\mathrm{T}^{-1}}_{\hat{A}}\hat{x} + \underbrace{\mathrm{T}B}_{\hat{B}}\,u, \\ \hat{y} &= \underbrace{C\mathrm{T}^{-1}}_{\hat{C}}\hat{x}. \end{aligned}$$

So, by means of T, we can transform system Σ in (3.3) into an *equivalent system* $\hat{\Sigma}$, (3.9), i.e.,

$$\left(\begin{array}{c|c} \mathrm{T} & \\ \hline & \mathbb{I}_p \end{array}\right) \underbrace{\left(\begin{array}{c|c} A & B \\ \hline C & \end{array}\right)}_{\Sigma} = \underbrace{\left(\begin{array}{c|c} \hat{A} & \hat{B} \\ \hline \hat{C} & \end{array}\right)}_{\hat{\Sigma}} \left(\begin{array}{c|c} \mathrm{T} & \\ \hline & \mathbb{I}_m. \end{array}\right) \tag{3.11}$$

In this thesis, we test the Balanced Truncation (BT) method to reduce the CO model systems. It is a Singular Value Decomposition (SVD)-based MOR approach. Because of the SVD's importance, we briefly revisit it here along with the Cholesky decomposition, another significant decomposition used in BT.

Theorem 3.4 (**Singular Value Decomposition**, [Ant05])**.** *Let $M \in \mathbb{C}^{n\times k}$ with $n \leqslant k$ and $\sigma_i = \sqrt{\lambda_i} \geqslant 0$, where λ_i the i-th eigenvalue (EV) of M^*M. Then, there exists a decomposition*

$$M = V\Sigma W^*, \tag{3.12}$$

where $V = (v_1 \cdots v_n) \in \mathbb{C}^{n\times n}$ and $W = (w_1 \cdots w_k) \in \mathbb{C}^{k\times k}$ are unitary matrices and $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_n)$ is a diagonal matrix with the singular values (SVs) σ_i arranged in decreasing order $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_n \geqslant 0$. The v_i's and w_j's are the left and right singular vectors of M, respectively.

Theorem 3.5 (**Cholesky Decomposition**, [Hig96])**.** *Let $M \in \mathbb{R}^{n \times n}$ be a symmetric positive definite matrix. Then, there exists a unique decomposition*

$$M = R^* R, \tag{3.13}$$

where the Cholesky factor $R \in \mathbb{R}^{n \times n}$ is an upper triangular matrix with positive elements on its diagonal.

3.2 Balanced Truncation for (Linear) Dynamic Models

In this section, we introduce the concepts of *reachability* and *observability*. They give us a measure to determine which states are important in a system. Balanced Truncation uses these concepts as the aforementioned "cut-or-keep" decision-making tool. A number of states in a dynamic system might be difficult to reach while the states which are difficult to observe might be different ones. This difficulty is measured by the *degree of reachability* and *degree of observability*, respectively. BT balances the system first such that the states to truncate are simultaneously hard to reach and observe.

Let $\mathbb{X} \subseteq \mathbb{R}^n$ be the state space throughout this section.

The Reachability Concept

In this subsection, we only consider systems of the form $\Sigma = \left(\begin{array}{c|c} A & B \\ \hline & \end{array}\right)$, since the output matrix C is of no consequence regarding reachability. The concept estimates how much the system's state x can be influenced by the input u.

Definition 3.6 (**Reachability Terminology**, [Ant05])**.**

- A point $\bar{x} \in \mathbb{X}$ is *reachable* from the zero state, if there is an input function $\bar{u}(t)$ of finite energy and a time $\bar{T} < \infty$, such that

  $$\bar{x} = \psi(\bar{u}; 0; \bar{T})$$

  is fulfilled.

- All the reachable states in Σ make up the *reachable subspace* $\mathbb{X}^{\text{reach}} \subseteq \mathbb{X}$. The system Σ is (completely) reachable if $\mathbb{X}^{\text{reach}} = \mathbb{X}$.

- The *reachability matrix* of Σ is defined as

  $$R(A, B) = \begin{bmatrix} B \; AB \; A^2B \; \cdots \; A^{n-1}B \; \cdots \end{bmatrix}. \tag{3.14}$$

Due to the Cayley-Hamilton theorem, the first n terms of the reachability matrix are sufficient to determine its rank and span. This allows us to use the finite reachability matrix $R_n(A, B) = [B \; AB \; A^2B \; \cdots \; A^{n-1}B]$ in computational settings[3].

[3] For properties of the reachability matrix and its connection to the reachability Gramian, see Appendix A.1.

We now introduce the first stepping-stone towards measuring the reachability of a state.

Definition 3.7 (**Finite Reachability Gramian**, [Ant05]). For a time $t < \infty$, the *finite reachability Gramian* is given by

$$\mathcal{P}(t) = \int_0^t e^{A\tau} BB^* e^{A^*\tau} d\tau. \tag{3.15}$$

From the definition follows that the finite reachability Gramian is positive semidefinite, i.e., $\mathcal{P}(t) = \mathcal{P}^*(t) \geqslant 0$.

Theorem 3.8 (**Reachability Conditions**, [Ant05]). *The following statements are equivalent:*

1. *The system Σ containing (A, B) is reachable.*
2. *The reachability matrix $R(A, B)$ has full rank.*
3. *The reachability Gramian $\mathcal{P}(t)$ is positive definite for some $t > 0$.*

The concept of *controllability* is equivalent to the reachability concept, which in a way reverses the latter: Controllability considers how an input u can direct the system from a given nonzero state to the zero state rather than moving the system from the zero state to a specific state.

Definition 3.9 (**Controllable state; controllable subspace**, [Ant05]).

- A (nonzero) point $\bar{x} \in \mathbb{X}$ is *controllable* to the zero state, if there is an input function $\bar{u}(t)$ with finite energy and a time $\bar{T} < \infty$, such that

  $$\psi(\bar{u}; \bar{x}; \bar{T}) = 0$$

  is fulfilled.

- All the reachable states in Σ make up the *controllable subspace* $\mathbb{X}^{\text{contr}} \subseteq \mathbb{X}$. If $\mathbb{X}^{\text{contr}} = \mathbb{X}$, the system Σ is (completely) controllable.

The equivalence of the reachability and controllability concepts is established by

Theorem 3.10 (**Reachability-Controllability Equivalence**, [Ant05]). $\mathbb{X}^{\text{reach}} = \mathbb{X}^{\text{contr}}$.

The Observability Concept

In order to steer and evaluate the dynamics of a system, knowledge of the state variables is necessary. As these data frequently cannot be measured, the state observation problem is concerned with the inference of the state variables $x(T)$ from outputs (or observations) $y(\tau)$, where $\tau \in [T, T + t]$.

Without loss of generality, the observation problem can be simplified by assuming that $T = 0$. We can also assume, that $u(\cdot) = 0$. As we know the input u, for $t \geqslant 0$ the last term in (3.6) is known, too. Thus, the task at hand is to find $x(0)$, given $C\psi(0, x(0), t)$ for $t \geqslant 0$. These assumptions hold throughout this subsection and render input matrix B redundant. Hence, we consider the system $\Sigma = \left(\begin{array}{c|c} A & \\ \hline C & \end{array}\right)$.

Definition 3.11 (**Observability Terminology**, [Ant05])**.**

- A point $\bar{x} \in \mathbb{X}$ is *unobservable* if $y(t) = C\psi(0, \bar{x}, t) = 0$ for all $t \geqslant 0$.
- All the unobservable states in Σ make up the *unobservable subspace* $\mathbb{X}^{\text{unobs}} \subseteq \mathbb{X}$.
- If $\mathbb{X}^{\text{unobs}} = 0$, the system Σ is (completely) observable.
- The *observability matrix* of Σ is defined as

$$O(C, A) = \left[C^* \; A^*C^* \; (A^*)^2C^* \; \cdots \; (A^*)^{n-1}C^* \; \cdots\right]^*. \tag{3.16}$$

Once more, the Cayley-Hamilton theorem permits us to work with the finite observability matrix $O_n(C, A,) = \left[C^* \; A^*C^* \; (A^*)^2C^* \; \cdots \; (A^*)^{n-1}C^*\right]^*$ in computational contexts[4].

We presently define the second stepping-stone towards computing the observability of a system.

Definition 3.12 (**Finite Observability Gramian**, [Ant05])**.** For a time $t < \infty$, the finite *observability Gramian* is given by

$$\mathcal{Q}(t) = \int_0^t e^{A^*\tau}C^*Ce^{A\tau}d\tau. \tag{3.17}$$

Theorem 3.13 (**Observability Conditions**, [Ant05])**.** *The following statements are equivalent:*

1. *The system Σ containing (C, A) is observable.*
2. *The observability matrix $O(C, A)$ has full rank.*
3. *The observability Gramian $\mathcal{Q}(t)$ is positive definite for some $t > 0$.*

Infinite Gramians and Energy Functionals

In this subsection, we derive an important feature of the Gramians which provides the wanted state importance measure. We first introduce one more necessary assumption. To this purpose, we assume $u \equiv 0$ and consider the autonomous system

$$\dot{x}(t) = Ax(t). \tag{3.18}$$

[4] For properties of the observability matrix and its link to the observability Gramian, see Appendix A.1.

Definition 3.14 (**Stable Matrix**, [Ant05; Zha13])**.** A matrix is *stable* if its eigenvalues λ are in the left half of the complex plane, or put differently, $\mathrm{Re}(\lambda) < 0$ for all eigenvalues λ.

A matrix is called *semi-stable* if the real part of every eigenvalue λ is nonpositive, i.e., $\mathrm{Re}(\lambda) \leqslant 0$ for all eigenvalues λ.

Theorem 3.15 relates a stable linear dynamics matrix A to an (asymptotically) stable system Σ.

Theorem 3.15 (**Matrix-based System Stability**, [Ant05])**.** *The system $\dot{x}(t) = Ax(t)$ is*

(1) stable *if and only if for all eigenvalues λ of A it holds* $\mathrm{Re}(\lambda) \leqslant 0$ *and the multiplicity of those eigenvalues with* $\mathrm{Re}(\lambda) = 0$ *is one.*

(2) asymptotically stable *if and only if the matrix A is stable.*

Remark 3.16. We equate stability with asymptotic stability from now on [Ant05]. For a more in-depth treatment of stability, we refer to Appendix A.2.

We now consider a stable system $\Sigma = \left(\begin{array}{c|c} A & B \\ \hline C & \end{array}\right)$. Let $\mathcal{L}_p^n(\mathcal{I})$ be the space of Lebesgue-integrable functions equipped with a p-norm over some finite or infinite interval $\mathcal{I}$, i.e.,

$$\mathcal{L}_p^n(\mathcal{I}) = \{f : \mathcal{I} \to \mathbb{R}^n, ||f||_p < \infty\}, 1 \leqslant p \leqslant \infty. \tag{3.19}$$

The 2-norm of a function f defines its energy

$$||f||^2 = \langle f, f \rangle = \int_0^T f^*(t) f(t) \mathrm{d}t.$$

Our objective is to compute the energy functionals of the system.

Definition 3.17 (**Controllability and Observability Energy Functionals**, [Sch93])**.** The controllability energy functional of a system is defined as

$$L_c(x_0) = \min_{\substack{u \in \mathcal{L}_2(-\infty,0) \\ x(-\infty)=0,\, x(0)=0}} = \frac{1}{2} \int_{-\infty}^0 ||u(t)||^2 \mathrm{d}t. \tag{3.20}$$

It is the minimum amount of energy needed to steer the system from the zero state at time $t = -\infty$ to $x(0) = x_0$.

Likewise, the observability energy functional of a system is defined as

$$L_o(x_0) = \frac{1}{2} \int_0^\infty ||y(t)||^2 \mathrm{d}t, \tag{3.21}$$

where $x(0) = x_0$ and $u(t) = 0$ for $0 \leqslant t < \infty$. It is the amount of energy generated by the initial state x_0 with no input present.

Comparing these functionals with Theorems 3.9 and 3.11, it is apparent, they give us a measure of the *degree of controllability* (or equivalently the *degree of reachability*) and *degree of observability*, respectively [Ant05]. Intuitively, we would like to minimize the energy it takes to control the system and maximize the produced energy we can observe from the system.

We can compute time-dependent values for the energy functionals with finite Gramians, see Proposition A.12 in Appendix A.3. However, it is desirable to have a time-independent measure. Since we assume a stable system with a stable matrix A, the integrals of both types of finite Gramians are bound as we let $t \to \infty$. Thus, we can define infinite versions of the reachability Gramian 3.7 and observability Gramian 3.12. That is for $t = \infty$, we have the following definition.

Definition 3.18 (**Infinite Gramians**, [Ant05])**.**

$$\mathcal{P} = \int_0^\infty e^{A\tau} BB^* e^{A^*\tau} d\tau, \tag{3.22}$$

$$\mathcal{Q} = \int_0^\infty e^{A^*\tau} C^* C e^{A\tau} d\tau. \tag{3.23}$$

To compute them, we solve the so-called *Lyapunov equations*.

Theorem 3.19 (**Lyapunov Equations**, [Ant05])**.** *The infinite Gramians are the unique solutions to the* Lyapunov equations

$$A\mathcal{P} + \mathcal{P}A^* + BB^* = 0, \tag{3.24}$$

$$A^*\mathcal{Q} + \mathcal{Q}A + C^*C = 0. \tag{3.25}$$

Remark 3.20. In Appendix A.2, we present a nice connection between Lyapunov equations and another type of stability based on a special kind of scalar functions, the so-called *Lyapunov functions*.

If we additionally assume (complete) reachability and observability of the system, the Gramians are positive definite and by the properties of the integral, we have

$$\mathcal{P} \geqslant \mathcal{P}(t), \quad \text{and} \quad \mathcal{Q} \geqslant \mathcal{Q}(t), \qquad \text{for all } t.$$

Keeping this observation in mind, we can compute optimized values of the energy functionals with the infinite Gramians as the following Theorem 3.21 states.

Theorem 3.21 (**Computation of Energy Functionals**, [Ant05; Sch93])**.** *Let the infinite Gramians $\mathcal{P}$ and $\mathcal{Q}$ be the unique solutions to the Lyapunov equations in Theorem 3.19. Then,*

$$L_c(x_0) = \frac{1}{2} x_0^* \mathcal{P}^{-1} x_0, \tag{3.26}$$

which is the minimal control energy needed to arrive at the state x_0, and

$$L_o(x_0) = \frac{1}{2} x_0^* \mathcal{Q} x_0, \tag{3.27}$$

which is the maximal observation energy generated by the initial state x_0.

Now we know how to optimize the energy functionals. If we therefore look at the importance of state variables from an energetic point of view, this theorem helps us to distinguish the significant from the insignificant states. In terms of energy, the states which require the most energy to control lie in the span of eigenvectors of $\mathcal{P}$ relating to small eigenvalues. Likewise, the states which cause the least observation energy lie in the eigenvector span of $\mathcal{Q}$ relating to small eigenvalues. The size of these sub-eigenspaces provide the *degree of controllability/reachability* and *degree of observability* [Ant05].

Balancing and Truncating

These degrees guide the decision which states to cut and which to keep. However, as these measures are basis-dependent, it might occur that states which are hard to control are not hard to observe or the other way around. Balanced Truncation makes sure to only omit those states for which the difficulty to control and observe coincide.

Mathematically, this amounts to express $\mathcal{P}$ and $\mathcal{Q}$ in a common basis. It requires a suitable similarity transformation which warrants that

$$\hat{\mathcal{P}} = \mathrm{T}\mathcal{P}\mathrm{T}^*, \hat{\mathcal{Q}} = \mathrm{T}^{-*}\mathcal{Q}\mathrm{T}^{-1} \Rightarrow \hat{\mathcal{P}}\hat{\mathcal{Q}} = \mathrm{T}(\mathcal{P}\mathcal{Q})\mathrm{T}^{-1}. \tag{3.28}$$

The transformation matrix T is then called a *balancing transformation* and its columns are the eigenvectors of the product $\mathcal{P}\mathcal{Q}$. It follows that the product of the Gramians of equivalent systems have the same eigenvalues. This in turn guarantees that the states which are difficult to control are identical to the states which are difficult to observe.

Definition 3.22 (**Balanced System**, [Ant05])**.** The stable, reachable and observable system Σ is *balanced* if $\mathcal{P} = \mathcal{Q}$. If in addition

$$\mathcal{P} = \mathcal{Q} = \Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_n),$$

the system is *principal-axis* balanced.

The values on the diagonal are special system invariants, called the *Hankel singular values* [Ant05]:

Lemma 3.23 (**Hankel Singular Values**, [Ant05])**.** *The* Hankel singular values *of the balanced system Σ are equal to the positive square roots of the eigenvalues (EVs) of the product of the infinite Gramians, that is*

$$\sigma_i(\Sigma) = \sqrt{\lambda_i(\mathcal{P}\mathcal{Q})}, \qquad i = 1, \ldots, n, \tag{3.29}$$

with $\sigma_i > \sigma_{i+1}$.

Evidently, in a balanced system large eigenvalues of the product $\mathcal{P}\mathcal{Q}$ correspond to large Hankel singular values. The decay rate of these SVs indicates an appropriate reduction order n_ρ.

There are several ways to determine a balancing transformation. One approach uses the eigenvalue decomposition. But since this requires the computation of two matrix inverses, we present the square root method here.

First, we decompose both $\mathcal{P}$ and $\mathcal{Q}$ into their respective Cholesky factors,

$$\mathcal{P} = \mathcal{R}^*\mathcal{R} \quad \text{and} \quad \mathcal{Q} = \mathcal{S}^*\mathcal{S}. \tag{3.30}$$

The SVs of the product of the Cholesky factors $\mathcal{R}$ and $\mathcal{S}^*$ are precisely the Hankel SVs. So, we compute the SVD

$$\mathcal{R}\mathcal{S}^* = V\Sigma W^*, \tag{3.31}$$

and get the orthogonal matrices V and W. The balancing matrix is then

$$\mathrm{T} = \Sigma^{-1/2}W^*\mathcal{S} \quad \text{with} \quad \mathrm{T}^{-1} = \mathcal{R}^*V\Sigma^{-1/2}. \tag{3.32}$$

We apply T to system (3.3) as in (3.11) and receive the balanced, equivalent system

$$\left(\begin{array}{c|c} \mathrm{T}A\mathrm{T}^{-1} & \mathrm{T}B \\ \hline C\mathrm{T}^{-1} & \end{array}\right). \tag{3.33}$$

Now for the reduction, we partition $V = \begin{pmatrix} V_{n_\rho} & V_{n-n_\rho} \end{pmatrix}$ and $W = \begin{pmatrix} W_{n_\rho} & W_{n-n_\rho} \end{pmatrix}$, $V_{n_\rho}, W_{n_\rho} \in \mathbb{R}^{n \times n_\rho}$, where n_ρ is the number of the n_ρ largest SVs of Σ, such that

$$\mathcal{R}\mathcal{S}^* = V\Sigma W^* \approx V_{n_\rho}\Sigma_{n_\rho}W_{n_\rho}^*. \tag{3.34}$$

The n_ρ leading columns of V span a low-dimensional space in which the state vector $x(t)$ can be approximated by another vector $\hat{x}(t)$. We set the transformation matrices

$$\mathcal{W}^* = \mathrm{T}_{n_\rho} = \Sigma_{n_\rho}^{-\frac{1}{2}}W_{n_\rho}^*\mathcal{S}, \qquad \mathcal{V} = \mathrm{T}_{n_\rho}^{-1} = \mathcal{R}^*V_{n_\rho}\Sigma_{n_\rho}^{-\frac{1}{2}}, \tag{3.35}$$

where Σ_{n_ρ} is the upper left $n_\rho \times n_\rho$-block of Σ. The state vector can then be approximated by $x \approx \mathcal{V}\hat{x}$ with $\hat{x}(t) \in \mathbb{R}^{n_\rho}$, which relates the ROM to the FOM, and we obtain the ROM

$$\hat{\Sigma} = \left(\begin{array}{c|c} \mathcal{W}^*A\mathcal{V} & \mathcal{W}^*B \\ \hline C\mathcal{V} & \end{array}\right) \tag{3.36}$$

of reduction order $n_\rho \ll n$.

3.3 Balanced Truncation for (Quadratic) Dynamic Models

A lot of concepts and ideas of MOR and BT remain valid under nonlinear conditions, still there are some important peculiarities. Because we lack a holistic nonlinear MOR theory, we focus on those which arise in the context of quadratic systems.

Structural Adaptations to Problem Setup and Balancing Measurement

The problem setup for nonlinear MOR is almost the same, except that at least one of the functions in system (3.1) and one of the maps in system (3.3) is explicitly nonlinear. The system might also have additional functions and corresponding matrix representations which model the nonlinear behavior of the system.

In the following considerations, we assume that the state of the system is the zero state at time $t = 0$, i.e., the initial condition is $x(0) = x_0 = 0$. If $x(0) = x_0 \neq 0$ we can define suitable state variables $\tilde{x} = x - x_0$ and transform the system accordingly, see section 4.3 for the treatment of our specific case, or [BBF14] for linear time-invariant systems.

We base our MOR efforts on nonlinear systems with a quadratic[5] structure[6] Their state space description is analogous to Definition 3.1:

Definition 3.24 (**State Space Description of Quadratic Dynamic System**)**.** Given matrices $A \in \mathbb{R}^{n\times n}, H \in \mathbb{R}^{n\times n^2}, B \in \mathbb{R}^{n\times m}$ and $C \in \mathbb{R}^{p\times n}$, the state space description of a quadratic dynamic system is

$$\begin{aligned} \dot{x}(t) &= Ax(t) + H\big(x(t) \otimes x(t)\big) + Bu(t), \\ y(t) &= Cx(t), \\ x(0) &= 0, \end{aligned} \tag{3.37}$$

with $x(t) \in \mathbb{R}^n, u(t) \in \mathbb{R}^m$, and $y(t) \in \mathbb{R}^p$. The compact notation is

$$\Sigma = \left(\begin{array}{c|c|c} A & H & B \\ \hline C & & \end{array}\right) \in \mathbb{R}^{(n+p)\times(n+n^2+m)}. \tag{3.38}$$

Matrices A and H model the linear and quadratic state dynamics, respectively, B is the input matrix, and C is the output matrix.

Remark 3.25. The symbol $\otimes$ depicts the *Kronecker product*. For the definition and properties, we refer to Appendix B.1

The objective of the quadratic MOR is to find a reduced dynamic system complexity $n_\rho \ll n$ (compare to (3.8)),

$$\begin{aligned} \dot{\hat{x}}(t) &= \hat{A}\hat{x}(t) + \hat{H}\big(\hat{x}(t) \otimes \hat{x}(t)\big) + \hat{B}u(t), \\ \hat{y}(t) &= \hat{C}\hat{x}(t), \\ \hat{x}(0) &= 0, \end{aligned} \tag{3.39}$$

with

$$\hat{A} \in \mathbb{R}^{n_\rho\times n_\rho}, \quad \hat{H} \in \mathbb{R}^{n_\rho\times n_\rho^2}, \quad \hat{B} \in \mathbb{R}^{n_\rho\times m}, \quad \hat{C} \in \mathbb{R}^{p\times n_\rho},$$

[5] This involves the use of the Kronecker product. For its definition and properties, see Appendix B.1.

[6] Our BT approach is based on the paper [BG17]. The authors actually deal with quadratic-bilinear systems, but since the bilinear term is irrelevant in our case, we omit it.

which approximates the quadratic system (3.37) such that it satisfies the aforementioned requirements on Page 26. Figure 3.2 illustrates the MOR process for quadratic systems.

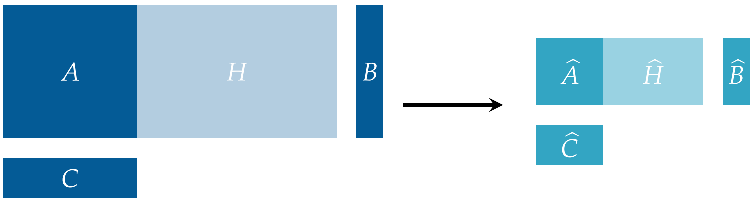


Figure 3.2: Graphical Depiction of MOR Process for Quadratic Systems

Definition 3.26 (**Quadratic Reduced-Order Model**)**.** We denote the *quadratic reduced-order model* of system (3.38) by

$$\hat{\Sigma} = \left(\begin{array}{c|c|c} \hat{A} & \hat{H} & \hat{B} \\ \hat{C} & & \end{array}\right) \in \mathbb{R}^{(n_\rho+p)\times(n_\rho+n_\rho^2+m)} \tag{3.40}$$

with *reduction order* $n_\rho \ll n$ (compare to Definition 3.3).

The basic idea and procedure of Balanced Truncation are identical to the linear case. But just like the problem setup, the key BT-specific concepts need to be modified to the quadratic case. We begin this adaptation process with the energy functionals which provide the measures of where and how much to truncate the system.

The definition of the controllability energy functional in Definition 3.17 holds for the nonlinear case. However, the definition given for the observability energy functional assumes that the system is *zero-state observable*, i.e., if both input and output variables are zero for $t \geqslant 0$, the state variable must remain at zero for all $t \geqslant 0$. This assumption is too strong for nonlinear systems. In [BG17], the authors overcome this restriction by requiring the input to be $\mathcal{L}_2$- and $\mathcal{L}_\infty$-bounded.

Definition 3.27 (**Relaxed Observability Energy Functional**, [BG17])**.** The observability energy functional of a nonlinear system is defined as

$$L_o(x_0) = \max_{\substack{u\in\mathcal{K}_{(\kappa,\beta)} \\ x(0)=x_0,\, x(\infty)=0}} = \frac{1}{2}\int_0^\infty ||y(t)||^2 \mathrm{d}t, \tag{3.41}$$

where $\mathcal{K}_{(\kappa,\beta)} = \{u \in \mathcal{L}_2^m[0,\infty) : ||u||_{\mathcal{L}_2} \leqslant \kappa, ||u||_{\mathcal{L}_\infty} \leqslant \beta\}$.

This definition evaluates the state variables from a state-to-output perspective. A state component would be deemed irrelevant if it has the least share in the observability energy for every conceivable input within the $\mathcal{L}_2$- and $\mathcal{L}_\infty$-bounded ball [BG17; GM96].

In the linear context, the controllability and observability energy functionals can be stated in terms of a plain quadratic formula, as seen in Theorem 3.21. For nonlinear systems, this is not possible, not even with the limitation to quadratic systems. Nevertheless, [BG17] contribute bounds on the energy functionals. The goal of the next subsections of this thesis is to derive and compute the truncated quadratic Gramians which provide the following bounds:

Theorem 3.28 (**Energy Functional Bounds with Truncated Gramians**, [BG17][7])**.** *Consider system* (3.37) *with a stable matrix A. Assume the system to be locally reachable and observable. Let the truncated reachability Gramian $\mathcal{P}_{\mathcal{T}} > 0$ and the observability Gramian $\mathcal{Q}_{\mathcal{T}} > 0$ satisfy their respective Lyapunov equation* (3.48) *and* (3.54)*. Then,*

(1) *there is a neighborhood $\mathcal{K}_c$ of 0, such that for $x \in \mathcal{K}_c$*

$$L_c(x) \geqslant \frac{1}{2} x^* \mathcal{P}_{\mathcal{T}}^{-1} x. \tag{3.42}$$

(2) *Likewise, there is a neighborhood $\mathcal{K}_o$ of 0, such that for $x \in \mathcal{K}_o$*

$$L_o(x) \leqslant \frac{1}{2} x^* \mathcal{Q}_{\mathcal{T}} x. \tag{3.43}$$

In their paper, [BG17] give an example which shows that close to the origin the energy functionals based on the truncated Gramians do not only present a good approximate to the actual energy functionals computed by solving the partial differential equations (see Theorem A.14), but indeed outperform the energy functionals based on the infinite Gramians.

Reachability Gramians

To find an expression for the reachability Gramian, [BG17] make use of the *Volterra series*, a well-established method for bilinear systems (see, e.g., [BGR17; BPK71; Rug81]). They treat the quadratic-bilinear case similarly to this type of nonlinear systems.

The Volterra series restates a nonlinear system in terms of cascaded linear (sub-) systems by a functional series expansion[8] [BPK71; VG18]. There are various ways to construct a Volterra Series representation of a system (see, e.g., [Rug81; VG18]). We merely illustrate the first few steps in the derivation of the series for quadratic systems to give an idea of the procedure and stick to the straightforward approach taken by [BG17] for simplicity.

We start by defining the notation

$$x_{\tau_1,\dots,\tau_\ell}(t) := x(t - \tau_1 - \cdots - \tau_\ell), \qquad u_{\tau_1,\dots,\tau_\ell}(t) := u(t - \tau_1 - \cdots - \tau_\ell),$$

[7] See Theorem A.15 for the version with infinite Gramians.

[8] In particular, a Volterra series expansion is often used to formulate an input-output representation of a nonlinear system in order to linearize the system or to derive transfer functions [BPK71; Fla12; Isi85; Rug81; VG18]. As a we do not seek a different output representation, we focus exclusively on the restatement of the state equation, just like [BG17].

and integrate both sides of the state equation of system (3.37) in the state variables with respect to time. Thus, we get

$$x(t) = \int_0^t e^{A\tau_1} B u_{\tau_1}(t) \mathrm{d}\tau_1 + \int_0^t e^{A\tau_1} H\big(x_{\tau_1}(t) \otimes x_{\tau_1}(t)\big) \mathrm{d}\tau_1. \tag{3.44}$$

As we need the value of $x_{\tau_1}(t)$, we proceed analogously to (3.44), and obtain the same integral solution, but with adapted integration limits and differential, i.e.,

$$x_{\tau_1}(t) = \int_0^{t-\tau_1} e^{A\tau_2} B u_{\tau_1,\tau_2}(t) \mathrm{d}\tau_2 + \int_0^{t-\tau_1} e^{A\tau_2} H\big(x_{\tau_1,\tau_2}(t) \otimes x_{\tau_1,\tau_2}(t)\big) \mathrm{d}\tau_2.$$

So, $x_{\tau_1}(t)$ computes part of the time interval of $x(t)$. Now, we substitute $x_{\tau_1}(t)$ in (3.44), but switch the differential of the second quadratic term from τ_2 to τ_3. Since the integration limit remains unchanged this is of no consequence for the computation of the integral. With this trick, we create a pseudo-linear equation and acquire

$$\begin{aligned}
x(t) = &\int_0^t e^{A\tau_1} B u_{\tau_1}(t) \mathrm{d}\tau_1 \\
&+ \int_0^t e^{A\tau_1} H\Big(\Big[\int_0^{t-\tau_1} e^{A\tau_2} B u_{\tau_1,\tau_2}(t) \mathrm{d}\tau_2 + \int_0^{t-\tau_1} e^{A\tau_2} H\big(x_{\tau_1,\tau_2}(t) \otimes x_{\tau_1,\tau_2}(t)\big) \mathrm{d}\tau_2 \Big] \\
&\otimes \Big[\int_0^{t-\tau_1} e^{A\tau_3} B u_{\tau_1,\tau_3}(t) \mathrm{d}\tau_3 + \int_0^{t-\tau_1} e^{A\tau_3} H\big(x_{\tau_1,\tau_3}(t) \otimes x_{\tau_1,\tau_3}(t)\big) \mathrm{d}\tau_3 \Big] \Big) \mathrm{d}\tau_1 \\
\overset{(\triangleleft)}{=} &\int_0^t e^{A\tau_1} B u_{\tau_1}(t) \mathrm{d}\tau_1 \\
&+ \int_0^t e^{A\tau_1} H\Big(\Big[\int_0^{t-\tau_1} e^{A\tau_2} B u_{\tau_1,\tau_2}(t) \mathrm{d}\tau_2 \otimes \int_0^{t-\tau_1} e^{A\tau_2} B u_{\tau_1,\tau_3}(t) \mathrm{d}\tau_3 \Big] \\
&+ \Big[\int_0^{t-\tau_1} e^{A\tau_2} H\big(x_{\tau_1,\tau_2}(t) \otimes x_{\tau_1,\tau_2}(t)\big) \mathrm{d}\tau_2 \otimes \int_0^{t-\tau_1} e^{A\tau_3} B u_{\tau_1,\tau_3}(t) \mathrm{d}\tau_3 \Big] \\
&+ \Big[\int_0^{t-\tau_1} e^{A\tau_2} B u_{\tau_1,\tau_2}(t) \mathrm{d}\tau_2 \otimes \int_0^{t-\tau_1} e^{A\tau_3} H\big(x_{\tau_1,\tau_3}(t) \otimes x_{\tau_1,\tau_3}(t)\big) \mathrm{d}\tau_3 \Big] \\
&+ \Big[\int_0^{t-\tau_1} e^{A\tau_2} H\big(x_{\tau_1,\tau_2}(t) \otimes x_{\tau_1,\tau_2}(t)\big) \mathrm{d}\tau_2 \otimes \int_0^{t-\tau_1} e^{A\tau_3} H\big(x_{\tau_1,\tau_3}(t) \otimes x_{\tau_1,\tau_3}(t)\big) \mathrm{d}\tau_3 \Big] \Big) \mathrm{d}\tau_1,
\end{aligned}$$

where we used the distributivity property of the Kronecker product in ($\triangleleft$) (3rd property in Theorem B.2). Pulling out the integral and switching the order of integration using

Fubini's theorem, this equals

$$
\begin{aligned}
x(t) = & \int_0^t e^{A\tau_1} B u_{\tau_1}(t) \mathrm{d}\tau_1 \\
& + \int_0^t \int_0^{t-\tau_1} \int_0^{t-\tau_1} e^{A\tau_1} H\Big(\Big[e^{A\tau_2} B u_{\tau_1,\tau_2}(t) \otimes e^{A\tau_3} B u_{\tau_1,\tau_3}(t) \Big] \\
& + \Big[e^{A\tau_2} H\big(x_{\tau_1,\tau_2}(t) \otimes x_{\tau_1,\tau_2}(t)\big) \otimes e^{A\tau_3} B u_{\tau_1,\tau_3}(t) \Big] \\
& + \Big[e^{A\tau_2} B u_{\tau_1,\tau_2}(t) \otimes e^{A\tau_3} H\big(x_{\tau_1,\tau_3}(t) \otimes x_{\tau_1,\tau_3}(t)\big) \Big] \\
& + \Big[e^{A\tau_2} H\big(x_{\tau_1,\tau_2}(t) \otimes x_{\tau_1,\tau_2}(t)\big) \otimes e^{A\tau_3} H\big(x_{\tau_1,\tau_3}(t) \otimes x_{\tau_1,\tau_3}(t)\big) \Big] \Big) \mathrm{d}\tau_1 \mathrm{d}\tau_2 \mathrm{d}\tau_3 \\
\overset{(\triangledown)}{=} & \int_0^t e^{A\tau_1} B u_{\tau_1}(t) \mathrm{d}\tau_1 \\
& + \int_0^t \int_0^{t-\tau_1} \int_0^{t-\tau_1} e^{A\tau_1} H\Big(\Big[(e^{A\tau_2} B \otimes e^{A\tau_3} B)(u_{\tau_1,\tau_2}(t) \otimes u_{\tau_1,\tau_3}(t)) \Big] \\
& + \Big[e^{A\tau_2} B u_{\tau_1,\tau_2}(t) \otimes e^{A\tau_3} H\big(x_{\tau_1,\tau_3}(t) \otimes x_{\tau_1,\tau_3}(t)\big) \Big] \\
& + \Big[e^{A\tau_2} H\big(x_{\tau_1,\tau_2}(t) \otimes x_{\tau_1,\tau_2}(t)\big) \otimes e^{A\tau_3} B u_{\tau_1,\tau_3}(t) \Big] \\
& + \Big[e^{A\tau_2} H\big(x_{\tau_1,\tau_2}(t) \otimes x_{\tau_1,\tau_2}(t)\big) \otimes e^{A\tau_3} H\big(x_{\tau_1,\tau_3}(t) \otimes x_{\tau_1,\tau_3}(t)\big) \Big] \Big) \mathrm{d}\tau_1 \mathrm{d}\tau_2 \mathrm{d}\tau_3,
\end{aligned}
$$

where we used the mixed product property in $(\triangledown)$ (5th property in Theorem B.2). Continuing to substitute for the state variables generates the Volterra series for the quadratic system. From this, the so-called *Volterra kernels* can be identified.

Definition 3.29 (**Reachability Mapping, Volterra Kernels**, [BG17])**.** We define the *reachability mapping* $\bar{P} = [\bar{P}_1, \bar{P}_2, \bar{P}_3, \ldots]$[9] with the $\bar{P}_i$'s being the *Volterra kernels* deduced from the Volterra series:

$$
\begin{aligned}
\bar{P}_1(t_1) &= e^{At_1} B, \\
\bar{P}_3(t_1, t_2, t_3) &= e^{At_3} \big[H[\bar{P}_1(t_1) \otimes \bar{P}_1(t_2)] \big] \\
&= e^{At_3} \big[H[(e^{At_1} B) \otimes (e^{At_2} B)] \big] \\
\bar{P}_5(t_1, \ldots, t_5) &= e^{At_5} \big[H[\bar{P}_1(t_1) \otimes \bar{P}_3(t_2, t_3, t_4), \bar{P}_3(t_1, t_2, t_3) \otimes \bar{P}_1(t_4)] \big], \\
&= e^{At_5} \big[H[(e^{At_1} B) \otimes (e^{At_4} [H(e^{At_2} B \otimes e^{At_3} B)]), \\
&\qquad\qquad (e^{At_3} [H(e^{At_1} B \otimes e^{At_2} B)]) \otimes (e^{At_4} B) \big], \\
&\vdots \\
\bar{P}_i(t_1, \ldots, t_i) &= e^{At_i} \big[H[\bar{P}_1(t_1) \otimes \bar{P}_{i-2}(t_2, \ldots, t_{i-1}), \\
&\qquad \bar{P}_2(t_1, t_2) \otimes \bar{P}_{i-3}(t_3, \ldots, t_{i-1}), \\
&\qquad \ldots, \bar{P}_{i-2}(t_1, \ldots, t_{i-2}) \otimes \bar{P}_1(t_{i-1})] \big], \quad \text{for all } i \geqslant 3,\ i \equiv 1 \bmod 2,
\end{aligned}
$$

and $\bar{P}_i(t_1, \ldots, t_i) = 0$ for all $i \equiv 0 \bmod 2$.

[9] To avoid confusion concerning the indices used in [BG17], we adhere to the numbering given in that paper. So, all the even numbered kernels are 0 here, because the bilinear term is $N = 0$.

Definition 3.30 (**Reachability Gramian**, [BG17])**.** The *reachability Gramian* $\mathcal{P}$ is defined by means of $\bar{P}$ as

$$\mathcal{P} = \sum_{i=1}^{\infty} P_i \qquad \text{with} \qquad P_i = \int_0^{\infty} \cdots \int_0^{\infty} \bar{P}_i(t_1, \ldots, t_i)\bar{P}_i^*(t_1, \ldots, t_i)\mathrm{d}t_1 \cdots \mathrm{d}t_i. \tag{3.45}$$

The summands of the reachability Gramian satisfy a sequence of a special type of quadratic Lyapunov equations. Consider the first summand in Definition 3.30,

$$P_1 = \int_0^{\infty} \bar{P}_1\bar{P}_1^*\mathrm{d}t_1 = \int_0^{\infty} e^{At_1}BB^*e^{A^*t_1}\mathrm{d}t_1.$$

This is just the linear infinite Gramian of Definition 3.18. As stated in Theorem 3.19, under the assumption that A is stable, it is the solution of the Lyapunov equation

$$AP_1 + P_1A^* + BB^* = 0. \tag{3.46}$$

For P_3, we have

$$\begin{aligned}
P_3 &= \int_0^{\infty}\int_0^{\infty}\int_0^{\infty} \bar{P}_3\bar{P}_3^*\mathrm{d}t_1\mathrm{d}t_2\mathrm{d}t_3 \\
&= \int_0^{\infty}\int_0^{\infty}\int_0^{\infty} e^{At_3}\left[H[\bar{P}_1(t_1)\otimes\bar{P}_1(t_2)][\bar{P}_1(t_1)\otimes\bar{P}_1(t_2)]\right]^*H^*e^{A^*t_3}\mathrm{d}t_1\mathrm{d}t_2\mathrm{d}t_3 \\
&= \int_0^{\infty}\int_0^{\infty}\int_0^{\infty} e^{At_3}\left[H[(e^{At_1}B)\otimes(e^{At_2}B)][(e^{At_1}B)\otimes(e^{At_2}B)]^*H^*\right]e^{A^*t_3}\mathrm{d}t_1\mathrm{d}t_2\mathrm{d}t_3 \\
&\overset{(\triangleright)}{=} \int_0^{\infty}\int_0^{\infty}\int_0^{\infty} e^{At_3}\left[H[(e^{At_1}B)\otimes(e^{At_2}B)][(e^{At_1}B)^*\otimes(e^{At_2}B)^*]H^*\right]e^{A^*t_3}\mathrm{d}t_1\mathrm{d}t_2\mathrm{d}t_3 \\
&\overset{(\triangledown)}{=} \int_0^{\infty}\int_0^{\infty}\int_0^{\infty} e^{At_3}\left[H[(e^{At_1}B)(e^{At_1}B)^*]\otimes[(e^{At_2}B)(e^{At_2}B)^*]H^*\right]e^{A^*t_3}\mathrm{d}t_1\mathrm{d}t_2\mathrm{d}t_3 \\
&= \int_0^{\infty}\int_0^{\infty}\int_0^{\infty} e^{At_3}\left[H[(e^{At_1}BB^*e^{A^*t_1})]\otimes[(e^{At_2}BB^*e^{A^*t_2})]H^*\right]e^{A^*t_3}\mathrm{d}t_1\mathrm{d}t_2\mathrm{d}t_3 \\
&= \int_0^{\infty} e^{At_3}\Big[H\underbrace{\Big[\int_0^{\infty}(e^{At_1}BB^*e^{A^*t_1})\mathrm{d}t_1\Big]}_{P_1}\otimes\underbrace{\Big[\int_0^{\infty}(e^{At_2}BB^*e^{A^*t_2})\mathrm{d}t_2\Big]}_{P_1}H^*\Big]e^{A^*t_3}\mathrm{d}t_3 \\
&= \int_0^{\infty} e^{At_3}\left[H(P_1\otimes P_1)H^*\right]e^{A^*t_3}\mathrm{d}t_3,
\end{aligned}$$

where we used the distributivity of (conjugate) transposition in ($\triangleright$) and the mixed product property of the Kronecker product in ($\triangledown$) (the 4th and 5th property in Theorem B.2). So, the term P_3 can be expressed as a function of the preceding term P_1 and is the solution to the Lyapunov equation [BG17],

$$AP_3 + P_3A^* + H(P_1\otimes P_1)H^* = 0. \tag{3.47}$$

Carrying on in the same manner with the succeeding P_i's, we see that each of these multiple integrals can be broken down into parts consisting of its predecessors, i.e.,

$$\begin{aligned}
P_i &= \int_0^\infty \cdots \int_0^\infty \bar{P}_i \bar{P}_i^* \mathrm{d}t_1 \cdots \mathrm{d}t_i \\
&= \int_0^\infty e^{At_i} \Big[H \Big[\Big(\int_0^\infty \bar{P}_1(t_1) \bar{P}_1^*(t_1) \mathrm{d}t_1 \Big) \\
&\qquad\qquad \otimes \Big(\int_0^\infty \cdots \int_0^\infty \bar{P}_{i-2}(t_2, \ldots, t_{i-1}) \bar{P}_{i-2}^*(t_2, \ldots, t_{i-1}) \mathrm{d}t_2 \cdots \mathrm{d}t_{i-1} \Big) \\
&\quad + \cdots + \Big(\int_0^\infty \cdots \int_0^\infty \bar{P}_{i-2}(t_1, \ldots, t_{i-2}) \bar{P}_{i-2}^*(t_1, \ldots, t_{i-2}) \mathrm{d}t_1 \cdots \mathrm{d}t_{i-2} \Big) \\
&\qquad\qquad \otimes \Big(\int_0^\infty \bar{P}_1(t_{i-1}) \bar{P}_1^*(t_{i-1}) \mathrm{d}t_{i-1} \Big) \Big] H^* \Big] e^{A^* t_i} \mathrm{d}t_i
\end{aligned}$$

and solve the Lyapunov equation

$$AP_i + P_i A^* + H(P_1 \otimes P_{i-2} + P_3 \otimes P_{i-3} + \cdots + P_{i-3} \otimes P_3 + P_{i-2} \otimes P_1)H^* = 0. \quad (3.48)$$

When we sum up equations (3.46), (3.47) and (3.48), we have

$$A \underbrace{\sum_{i=1}^\infty P_i}_{\mathcal{P}} + \underbrace{\sum_{i=1}^\infty P_i}_{\mathcal{P}} A^* + H\Big(\underbrace{\sum_{i=1}^\infty P_i}_{\mathcal{P}} \otimes \underbrace{\sum_{i=1}^\infty P_i}_{\mathcal{P}} \Big) H^* + BB^* = 0, \quad (3.49)$$

which implies the following theorem,

Theorem 3.31 (**Reachability Gramian Lyapunov Equation**, [BG17]). *Consider system* (3.37) *and let A be stable. Assuming that the reachability Gramian $\mathcal{P}$ as defined in 3.30 exists, it is the solution of the generalized quadratic Lyapunov equation*

$$A\mathcal{P} + \mathcal{P}A^* + H(\mathcal{P} \otimes \mathcal{P})H^* + BB^* = 0. \quad (3.50)$$

Observability Gramians

The derivation and results for the reachability Gramian are parallel to those which can be obtained for the observability Gramian. We simply state the definitions and conclusions for this Gramian here.

Definition 3.32 (**Observability Mapping, Volterra Kernels**, [BG17]). Let the $\bar{P}_i$'s be as in Definition 3.29, and let $\mathcal{H}^{(2)}$ be the mode-2 unfolding of a tensor constructed from H[10]. Furthermore, let the $\bar{Q}_i$'s be the Volterra kernels deduced from the Volterra series,

[10] For details, see Appendix B.2.

and let $\bar{Q} = [\bar{Q}_1, \bar{Q}_2, \bar{Q}_3, \ldots]$[11] be the observability mapping made up of these kernels:

$$\begin{aligned}
\bar{Q}_1(t_1) &= e^{A^*t_1}C^*,\\
\bar{Q}_3(t_1,t_2,t_3) &= e^{A^*t_3}\big[\mathcal{H}^{(2)}[\bar{P}_1(t_1) \otimes \bar{Q}_1(t_2)]\big],\\
&= e^{A^*t_3}\big[\mathcal{H}^{(2)}[(e^{At_1}B) \otimes (e^{A^*t_2}C^*)]\big]\\
\bar{Q}_5(t_1,\ldots,t_5) &= e^{At_5}\big[\mathcal{H}^{(2)}[\bar{P}_1(t_1) \otimes \bar{Q}_3(t_2,t_3,t_4), \bar{P}_3(t_1,t_2,t_3) \otimes \bar{Q}_1(t_4)]\big],\\
&= e^{At_5}\big[\mathcal{H}^{(2)}[(e^{At_1}B) \otimes (e^{At_4}[\mathcal{H}^{(2)}(e^{A^*t_2}C^* \otimes e^{A^*t_3}C^*)]),\\
&\qquad\qquad (e^{At_3}[H(e^{At_1}B \otimes e^{At_2}B)]) \otimes (e^{A^*t_4}C^*)\big],\\
&\;\;\vdots\\
\bar{Q}_i(t_1,\ldots,t_i) &= e^{At_i}\big[\mathcal{H}^{(2)}[\bar{P}_1(t_1) \otimes \bar{Q}_{i-2}(t_2,\ldots,t_{i-1}),\\
&\qquad \bar{P}_2(t_1,t_2) \otimes \bar{Q}_{i-3}(t_3,\ldots,t_{i-1}),\\
&\qquad \ldots, \bar{P}_{i-2}(t_1,\ldots,t_{i-2}) \otimes \bar{Q}_1(t_{i-1})]\big], \quad \text{for all } i \geqslant 3,\ i \equiv 1 \bmod 2,
\end{aligned}$$

and $\bar{Q}_i(t_1,\ldots,t_i) = 0$ for all $i \equiv 0 \bmod 2$.

Definition 3.33 (**Observability Gramian**, [BG17])**.** We define the *observability Gramian* $\mathcal{Q}$ by means of $\bar{Q}$ as

$$\mathcal{Q} = \sum_{i=1}^{\infty} Q_i \qquad \text{with} \qquad Q_i = \int_0^\infty \cdots \int_0^\infty \bar{Q}_i(t_1,\ldots,t_i)\bar{Q}_i^*(t_1,\ldots,t_i)\mathrm{d}t_1 \cdots \mathrm{d}t_i. \quad (3.51)$$

These infinitely many summands also satisfy a series of a special kind of Lyapunov equation. The first term

$$Q_1 = \int_0^\infty \bar{Q}_1\bar{Q}_1^*\mathrm{d}t_1 = \int_0^\infty e^{A^*t_1}C^*Ce^{At_1}\mathrm{d}t_1$$

is the linear infinite Gramian as given in Definition 3.18, which solves the Lyapunov equation

$$A^*Q_1 + Q_1A + C^*C = 0, \quad (3.52)$$

as stated in Theorem 3.19, assuming that A is stable. For the second nonzero term, we find that

$$\begin{aligned}
Q_3 &= \int_0^\infty \int_0^\infty \int_0^\infty \bar{Q}_3\bar{Q}_3^*\mathrm{d}t_1\mathrm{d}t_2\mathrm{d}t_3\\
&= \int_0^\infty \int_0^\infty \int_0^\infty e^{A^*t_3}\big[\mathcal{H}^{(2)}[\bar{P}_1(t_1) \otimes \bar{Q}_1(t_2)][\bar{P}_1(t_1) \otimes \bar{Q}_1(t_2)]\big]^*(\mathcal{H}^{(2)})^*e^{At_3}\mathrm{d}t_1\mathrm{d}t_2\mathrm{d}t_3\\
&= \int_0^\infty e^{A^*t_3}\big[\mathcal{H}^{(2)}\underbrace{\Big[\int_0^\infty (e^{At_1}BB^*e^{A^*t_1})\mathrm{d}t_1\Big]}_{P_1} \otimes \underbrace{\Big[\int_0^\infty (e^{A^*t_2}C^*Ce^{At_2})\mathrm{d}t_2\Big]}_{Q_1}(\mathcal{H}^{(2)})^*\big]e^{At_3}\mathrm{d}t_3\\
&= \int_0^\infty e^{A^*t_3}\big[\mathcal{H}^{(2)}(P_1 \otimes Q_1)(\mathcal{H}^{(2)})^*\big]e^{At_3}\mathrm{d}t_3.
\end{aligned}$$

So, the term Q_3 is composed of the preceding term Q_1 and P_1 from the reachability part. It is the solution of the Lyapunov equation [BG17],

$$A^*Q_3 + Q_3A + \mathcal{H}^{(2)}(P_1 \otimes Q_1)(\mathcal{H}^{(2)})^* = 0. \quad (3.53)$$

[11] The same reasoning for the kernel numbering applies as given in footnote [9] on page 38.

The succeeding Q_i's take the form

$$\begin{aligned}
Q_i &= \int_0^\infty \cdots \int_0^\infty \bar{Q}_i \bar{Q}_i^* \mathrm{d}t_1 \cdots \mathrm{d}t_i \\
&= \int_0^\infty e^{A^* t_i} \Big[\mathcal{H}^{(2)} \Big[\Big(\int_0^\infty \bar{P}_1(t_1) \bar{P}_1^*(t_1) \mathrm{d}t_1 \Big) \\
&\qquad \otimes \Big(\int_0^\infty \cdots \int_0^\infty \bar{Q}_{i-2}(t_2, \dots, t_{i-1}) \bar{Q}_{i-2}^*(t_2, \dots, t_{i-1}) \mathrm{d}t_2 \cdots \mathrm{d}t_{i-1} \Big) \\
&\quad + \cdots + \Big(\int_0^\infty \cdots \int_0^\infty \bar{P}_{i-2}(t_1, \dots, t_{i-2}) \bar{P}_{i-2}^*(t_1, \dots, t_{i-2}) \mathrm{d}t_1 \cdots \mathrm{d}t_{i-2} \Big) \\
&\qquad \otimes \Big(\int_0^\infty \bar{Q}_1(t_{i-1}) \bar{Q}_1^*(t_{i-1}) \mathrm{d}t_{i-1} \Big) \Big] (\mathcal{H}^{(2)})^* \Big] e^{At_i} \mathrm{d}t_i
\end{aligned}$$

and satisfy the Lyapunov equation

$$A^* Q_i + Q_i A + \mathcal{H}^{(2)}(P_1 \otimes Q_{i-2} + \cdots + P_{i-2} \otimes Q_1)(\mathcal{H}^{(2)})^* = 0. \tag{3.54}$$

Summing up equations (3.52)-(3.54), we have[12]

$$A^* \underbrace{\sum_{i=1}^{\infty} Q_i}_{\mathcal{Q}} + \underbrace{\sum_{i=1}^{\infty} Q_i}_{\mathcal{Q}} A + \mathcal{H}^{(2)} \Big(\underbrace{\sum_{i=1}^{\infty} P_i}_{\mathcal{P}} \otimes \underbrace{\sum_{i=1}^{\infty} Q_i}_{\mathcal{Q}} \Big) (\mathcal{H}^{(2)})^* + C^* C = 0, \tag{3.55}$$

from which we infer the following

Theorem 3.34 (**Observability Gramian Lyapunov Equation**, [BG17])**.** *Consider system* (3.37) *and let A be stable. Assuming that the observability Gramian $\mathcal{Q}$ exists as in Definition* 3.33*, it is the solution of the generalized quadratic Lyapunov equation*

$$A^* \mathcal{Q} + \mathcal{Q} A + \mathcal{H}^{(2)}(\mathcal{P} \otimes \mathcal{Q})(\mathcal{H}^{(2)})^* + C^* C = 0. \tag{3.56}$$

Balancing and Truncating

The MOR via Balanced Truncation of system (3.37) consists of three major steps:

1. the computation of the reachability and observability Gramians,
2. the computation of the balancing transformation matrices,
3. the projection of the reduced system.

The computationally and mathematically most intricate step is the first one. As we have seen in the preceding sections, it involves the solution of two generalized Lyapunov equations, (3.50) and (3.56). Due to their quadratic nature, they are both hard and costly to solve. Their efficient computation is still an ongoing research question [BG17]. Therefore, we resort to truncating the infinite summation in Definitions 3.30 and 3.33:

$$\mathcal{P}_{\mathcal{T}} = \sum_{i=1}^{\nu} P_i, \qquad \mathcal{Q}_{\mathcal{T}} = \sum_{i=1}^{\nu} Q_i, \tag{3.57}$$

[12] Observe that the equation is actually linear in the Q-terms.

where ν is the number of consecutive Volterra kernels used. For $\nu = 1$ and $\nu = 3$, we have already explicitly demonstrated that these truncated Gramians satisfy the Lyapunov equations (3.47) and (3.53).[13]

While [BG17] propose a scheme based on fixed-point iterations which determines low-rank Cholesky factors of the Gramians, our approach stays very close to the derivation of the Gramians. We compute the Volterra kernels directly, add them up to get the Gramians and then determine their respective Cholesky factors.

Before we state the actual algorithm used to compute the Gramians, we investigate the key subroutine responsible for solving the Lyapunov equation

$$\mathbb{A}\mathbb{P} + \mathbb{P}\mathbb{A}^* + \mathbb{B}\mathbb{B}^* = 0,$$

with $\mathbb{A} \in \mathbb{R}^{n\times n}$ stable, $\mathbb{B} \in \mathbb{R}^{n\times m}, \mathbb{P} = \mathbb{P}^* \in \mathbb{R}^{n\times n}$ symmetric, and $(\mathbb{A}, \mathbb{B})$ reachable, i.e., $\mathbb{P} > 0$ and $\mathbb{A}, \mathbb{B}$ are nonsingular.

The method developed by Hammarling [Ham82] computes the Cholesky factor $\mathbb{F}$ of $\mathbb{P}$ instead of $\mathbb{P}$. This has some advantages, e.g., because the Cholesky factor is the desired objective anyway, or because it has a better condition number than $\mathbb{P}$ [Ham82].

The first step is to bring $\mathbb{A}$ into Schur form, i.e., into upper triangular form. Since $\mathbb{A}$ is nonsingular, there exists a unitary matrix $\mathbb{Q}$ such that

$$\mathbb{A} = \mathbb{Q}\check{\mathbb{A}}\mathbb{Q}^*,$$

where $\check{\mathbb{A}}$ is in Schur form. We apply $\mathbb{Q}$ to the other equation matrices

$$\check{\mathbb{P}} = \mathbb{Q}^*\mathbb{P}\mathbb{Q}, \qquad \check{\mathbb{B}} = \mathbb{Q}^*\mathbb{B}\mathbb{Q},$$

and transform the system into an equivalent system in the Schur basis [Ant05; Ham82]. For convenience, we omit the hašek symbol from now on. For the remainder of this subsection, we assume that $\mathbb{A}$ is upper triangular and the Lyapunov equation matrices are in the Schur basis.

The next step is to partition $\mathbb{A}, \mathbb{P}$ and $\mathbb{B}$ such that

$$\mathbb{A} = \begin{pmatrix} \mathbb{A}_{11} & \mathbb{A}_{12} \\ 0 & \mathbb{A}_{22} \end{pmatrix}, \qquad \mathbb{B} = \begin{pmatrix} \mathbb{B}_1 \\ \mathbb{B}_2 \end{pmatrix}, \qquad \mathbb{P} = \begin{pmatrix} \mathbb{P}_{11} & \mathbb{P}_{12} \\ \mathbb{P}_{12}^* & \mathbb{P}_{22} \end{pmatrix}, \tag{3.58}$$

where $\mathbb{A}_{11} \in \mathbb{R}^{k\times k}$ and $\mathbb{A}_{22} \in \mathbb{R}^{(n-k)\times(n-k)}$ are upper triangular, and the other matrix blocks of $\mathbb{A}, \mathbb{P}$ and $\mathbb{B}$ have matching dimensions [Ant05].

Since $\mathbb{P}$ is positive definite, it can be decomposed as $\mathbb{P} = \mathbb{F}\mathbb{F}^*$ with an upper triangular matrix $\mathbb{F}$. To compute the Cholesky factor $\mathbb{F}$, we first perform another transformation

$$\widetilde{\mathbb{A}}\widetilde{\mathbb{P}} + \widetilde{\mathbb{P}}\widetilde{\mathbb{A}}^* + \widetilde{\mathbb{B}}\widetilde{\mathbb{B}}^* = 0, \tag{3.59}$$

[13] In our numerical experiments, we have also run tests with $\nu = 5$, for which we have stated the respective Volterra kernels in Definitions 3.29 and 3.32

where $\widetilde{\mathbb{A}} = \mathrm{T}\mathbb{A}\mathrm{T}^{-1}$, $\widetilde{\mathbb{B}} = \mathrm{T}\mathbb{B}$, $\widetilde{\mathbb{P}} = \mathrm{T}\mathbb{P}\mathrm{T}^{*}$, and

$$\mathrm{T} = \begin{pmatrix} \mathbb{I} & -\mathbb{P}_{12}\mathbb{P}_{22}^{-1} \\ 0 & \mathbb{I} \end{pmatrix}, \qquad \widetilde{\mathbb{A}} = \begin{pmatrix} \mathbb{A}_{11} & \widetilde{\mathbb{A}}_{12} \\ 0 & \mathbb{A}_{22} \end{pmatrix}, \qquad \widetilde{\mathbb{B}} = \begin{pmatrix} \widetilde{\mathbb{B}}_1 \\ \mathbb{B}_2 \end{pmatrix}, \qquad \widetilde{\mathbb{P}} = \begin{pmatrix} \widetilde{\mathbb{P}}_{11} & 0 \\ 0 & \mathbb{P}_{22} \end{pmatrix}, \tag{3.60}$$

with

$$\begin{aligned} \widetilde{\mathbb{A}}_{12} &= \mathbb{A}_{12} - \mathbb{P}_{12}\mathbb{P}_{22}^{-1}\mathbb{A}_{22} + \mathbb{A}_{11}\mathbb{P}_{12}\mathbb{P}_{22}^{-1}, \\ \widetilde{\mathbb{B}}_1 &= \mathbb{B}_1 - \mathbb{P}_{12}\mathbb{P}_{22}^{-1}\mathbb{B}_2, \\ \widetilde{\mathbb{P}}_{11} &= \mathbb{P}_{11} - \mathbb{P}_{12}\mathbb{P}_{22}^{-1}\mathbb{P}_{12}^{*}. \end{aligned} \tag{3.61}$$

With the help of these equations, we can compute the Cholesky factors of $\mathbb{P}$:

$$\mathbb{P} = \underbrace{\begin{pmatrix} \widetilde{\mathbb{P}}_{11}^{1/2} & \mathbb{P}_{12}\mathbb{P}_{22}^{-1/2} \\ 0 & \mathbb{P}_{22}^{1/2} \end{pmatrix}}_{\mathbb{F}} \underbrace{\begin{pmatrix} \widetilde{\mathbb{P}}_{11}^{1/2} & 0 \\ \mathbb{P}_{22}^{-1/2}\mathbb{P}_{12}^{*} & \mathbb{P}_{22}^{1/2} \end{pmatrix}}_{\mathbb{F}^{*}} = \mathbb{F}\mathbb{F}^{*}, \tag{3.62}$$

where $\widetilde{\mathbb{P}}_{11} \in \mathbb{R}^{k\times k}$, $\mathbb{P}_{12} \in \mathbb{R}^{k\times(n-k)}$, and $\mathbb{P}_{22} \in \mathbb{R}^{(n-k)\times(n-k)}$ for $k < n$. We can successively compute the individual blocks of $\mathbb{P}$ and simultaneously determine $\mathbb{F}$ by exploiting triangularity [Ant05].

By comparing equations (3.59), (3.60) and (3.61), we get the equations

$$\begin{aligned} \mathbb{A}_{22}\mathbb{P}_{22} + \mathbb{P}_{22}\mathbb{A}_{22}^{*} + \mathbb{B}_2\mathbb{B}_2^{*} &= 0, \\ \mathbb{A}_{12}\mathbb{P}_{22} - \mathbb{P}_{12}[\mathbb{P}_{22}^{-1}A_{22}\mathbb{P}_{22} + \mathbb{P}_{22}^{-1}\mathbb{B}_2\mathbb{B}_2^{*}] + \mathbb{A}_{11}\mathbb{P}_{12} + \mathbb{B}_1\mathbb{B}_2^{*} &= 0, \\ \mathbb{A}_{11}\widetilde{\mathbb{P}}_{11} + \widetilde{\mathbb{P}}_{11}\mathbb{A}_{11}^{*} + \widetilde{\mathbb{B}}_1\widetilde{\mathbb{B}}_1^{*} &= 0. \end{aligned} \tag{3.63}$$

We go backwards in our computations and set $k = n - 1$ in order to compute the last row, or respectively column, of the Cholesky factors. The first equation in (3.63) is a Lyapunov equation which we solve for $\mathbb{P}_{22}$ and get

$$\mathbb{P}_{22} = -\frac{\mathbb{B}_2\mathbb{B}_2^{*}}{\mathbb{A}_{22} + \mathbb{A}_{22}^{*}} \in \mathbb{R}. \tag{3.64}$$

We plug this into the second equation in (3.63) and thus obtain

$$\mathbb{P}_{12} = (\mathbb{A}_{11} + \mathbb{A}_{22}^{*}\mathbb{I}_{n-1})^{-1}\left(\frac{\mathbb{B}_2}{\mathbb{A}_{22} + \mathbb{A}_{22}^{*}}\mathbb{A}_{12} - \mathbb{B}_1\right)\mathbb{B}_2^{*} \in \mathbb{R}^{n-1}. \tag{3.65}$$

These two steps already suffice to calculate the last column of $\mathbb{F}$ in (3.62). To determine the remaining block $\widetilde{\mathbb{P}}_{11}^{1/2}$, we observe that $\widetilde{\mathbb{B}}_1 = \mathbb{B}_1 - \mathbb{P}_{12}\mathbb{P}_{22}^{-1}\mathbb{B}_2$ from (3.61), and that the last equation in (3.63) is also a Lyapunov equation but of dimension $n-1$. We can now iterate these steps and determine the entire Cholesky factor $\mathbb{F}$ [Ant05; Ham82].

Algorithm 3.1 depicts the computation of the Cholesky factor $\mathbb{F}$ following Hammarling's method as implemented in the MATLAB function lyapchol and in SLICOT [Ham82; iST].

Algorithm 3.1: Lyapunov-Cholesky Routine

```
Input: square matrix $\mathbb{A}$, matrix $\mathbb{B}$.
Output: Cholesky factor $\mathbb{F}$, where $\mathbb{F}^*\mathbb{F} = \mathbb{P}$ is a solution of the Lyapunov
        equation $\mathbb{A}\mathbb{P} + \mathbb{P}\mathbb{A}^* + \mathbb{B}\mathbb{B}^* = 0$.
Compute Schur factorization of $\mathbb{A}$:
    $\mathbb{Q}\mathbb{A}\mathbb{Q}^* := \mathbb{A}$.
Compute QR factorization of $\mathbb{B}$:
    $\widetilde{\mathbb{L}} \begin{bmatrix} \widetilde{\mathbb{M}} \\ 0 \end{bmatrix} := \mathbb{B}$,
where $\widetilde{\mathbb{L}}$ is an $n \times n$ unitary matrix and $\widetilde{\mathbb{M}}$ is a square upper triangular matrix.
Compute $\widetilde{\mathbb{B}} := \widetilde{\mathbb{M}}\mathbb{Q}$ and perform QR factorization:
    $\mathbb{L}\mathbb{M} := \widetilde{\mathbb{B}}$.
Solve the Lyapunov equation $\mathbb{A}(\mathbb{U}^*\mathbb{U}) + (\mathbb{U}^*\mathbb{U})\mathbb{A}^* = \mathbb{M}\mathbb{M}^*$ for $\mathbb{U}$, where $\mathbb{U}$ is
  upper triangular.
Compute QR factorization of $\mathbb{Q}\mathbb{U}$ to obtain Cholesky factor $\mathbb{F}$:
    $\mathbb{Q}\mathbb{F} := \mathbb{U}\mathbb{Q}^*$.
```

We now turn to the algorithm to determine the truncated Gramians. The procedure for the reachability Gramian is given in Algorithm 3.2. It is an immediate implementation of its derivation, but with two amendments: The first is the illustrated computation of the Cholesky factors of the Volterra kernels instead of the kernels themselves in lines 2 and 14-15. The second is the shift of matrix A by a small parameter $\alpha > 0$ in line 1. Due to the system lifting described in Section 4.2.2, the matrix is very sparse with EVs equaling 0, and is therefore not stable. This property is of great significance, e.g., to derive the Gramians, solve the Lyapunov equations in Theorems 3.19, 3.31 and 3.34, and apply the Lyapunov-Cholesky routine in Algorithm 3.1. Therefore, we shift A in Algorithm 3.2 to make it stable.[14]

Remark 3.35. While $H = \widetilde{H}$ is sparse, forming the quadratic product in line 12, and especially the Kronecker product of $P_k \otimes P_j$ in line 9 is not. This causes a substantial computational burden.

We omit stating the algorithm for the observability Gramian here[15] on account of the parallels to the computation of the reachability Gramian. Aside from the exchange of matrix B with matrix C, and the supplementary input of the Volterra kernels $P_1, \ldots, P_\nu$, there is only the additional construction of the mode-2 unfolding to obtain matrix $\mathcal{H}^{(2)}$. This is necessary to compute the Volterra kernels for $\nu \geqslant 3$ as indicated in Definition 3.32, and is demonstrated in Appendix B.2.

This concludes the first major step of reducing system (3.37) by Balanced Trunca-

[14] We go into greater detail on making A stable in Section 4.4.

[15] You can find the Algorithm C.1 in Appendix C.

Algorithm 3.2: Iterative Scheme to Compute Truncated Gramian $\mathcal{P}_\mathcal{T}$

```
Input: quadratic system matrices $A, H, B$, shift parameter $\alpha > 0$, number of
       Volterra kernels $\nu$
Output: Cholesky factor $R$ (upper triangular matrix) of the truncated
        reachability Gramian $P_\mathcal{T}$, Volterra kernels $P_1, \ldots .P_\nu$
Shift matrix $A$ with $\alpha$ to make the matrix stable: $A_\alpha := A - \alpha\mathbb{I}$.
Determine Cholesky factor $R_1$ of Volterra kernel $P_1$ by calling Algorithm 3.1
  with $\mathbb{A} := A_\alpha$, $\mathbb{B} := B$, and compute $P_1 := R_1^* R_1$.
Set $P_2 := 0$.
Compute Cholesky factors $R_3, \ldots, R_\nu$ to determine Volterra kernels $P_3, \ldots P_\nu$:
for $i = 3 : \nu$ do
    Initiate $P_i^{\text{kron}} := 0; \quad j := i - 2$.
    Sum up Kronecker products of previously computed $P_k$.
    for $k = 1 : i - 2$ do
        $P_i^{\text{kron}} := P_i^{\text{kron}} + P_k \otimes P_j$;
        $j := j - 1$;
    end for
    Set $H_i^{\text{kron}} := H \, P_i^{\text{kron}} H^*$.
    Determine Cholesky factor $\mathbb{B}_i$ of $H_i^{\text{kron}}$, i.e., compute $\mathbb{B}_i^* \mathbb{B}_i := H_i^{\text{kron}}$.
    Update Cholesky factor $R_i$ by calling Algorithm 3.1 with
        $\mathbb{A} := A_\alpha$, $\mathbb{B} := \mathbb{B}_i$.
    Compute Volterra kernel $P_i := R_i^* R_i$.
end for
Determine $P_\mathcal{T}$ by summing up Volterra kernels:
$P_\mathcal{T} = 0$;
for $i = 1 : \nu$ do
    $P_\mathcal{T} = P_\mathcal{T} + P_i$.
end for
Determine Cholesky factor $R$ of $P_\mathcal{T}$, i.e., compute $R^* R := P_\mathcal{T}$.
```

tion. The other two, namely, the construction of the balancing transformation matrices and the projection of the reduced-order system in (3.30)–(3.35), are almost exactly the same as in the linear case - only matrix H has to be projected in addition. The state vector approximation, $x \approx \mathcal{V}\hat{x}$, relates the ROM to the FOM again, and we get the ROM,

$$\hat{\Sigma} = \left(\begin{array}{c|c|c} \mathcal{W}^* A \mathcal{V} & \mathcal{W}^* H (\mathcal{V} \otimes \mathcal{V}) & \mathcal{W}^* B \\ \hline C\mathcal{V} & & \end{array}\right) \in \mathbb{R}^{(n_\rho + p) \times (n_\rho + n_\rho^2 + m)}, \tag{3.66}$$

the quadratic version of the reduced linear system in (3.36). All computation steps are put together in Algorithm 3.3. Note that the procedure requires the reduction order n_ρ as an input parameter rather than determining this quantity as part of the procedure.

We have done this to test different values of n_ρ.

Algorithm 3.3: Balanced Truncation Algorithm for Quadratic Systems

```
Input: quadratic system matrices $A, H, B, C$, reduction order $n_\rho$, shift
       parameter $\alpha > 0$, number of Volterra kernels $\nu$.
Output: reduced matrices $\hat{A}, \hat{H}, \hat{B}, \hat{C}$, projection matrices $\mathcal{V}, \mathcal{W}$.
Compute Cholesky factor $R$ of reachability Gramian $\mathcal{P}$ with Algorithm 3.2 and
  Cholesky factor $S$ of observability Gramian $\mathcal{Q}$ with the corresponding
  algorithm, respectively.
Compute the Singular Value Decomposition of $RS^*$:
    $V \Sigma W^* = RS^*$,
    where $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_n)$ contains the singular values in decreasing order.
Construct projection matrices $\mathcal{V}, \mathcal{W}$:
    $\mathcal{V} = R^* V_{n_\rho} \Sigma_{n_\rho}^{-\frac{1}{2}}$ and $\mathcal{W}^* = \Sigma_{n_\rho}^{-\frac{1}{2}} W_{n_\rho}^* S$,
    where $\Sigma_{n_\rho}$ contains the $n_\rho$ largest singular values of $\Sigma$, and $V_{n_\rho}, W_{n_\rho}$ contain
      the corresponding left- and right-singular vectors, respectively.
Compute reduced-order system:
    $\hat{A} = \mathcal{W}^* A \mathcal{V}, \quad \hat{H} = \mathcal{W}^* H (\mathcal{V} \otimes \mathcal{V}), \quad \hat{B} = \mathcal{W}^* B, \quad \hat{C} = C \mathcal{V}.$
```

3.4 Proper Orthogonal Decomposition

We use the well-established Proper Orthogonal Decomposition (POD) method in order to evaluate the validity of our numerical tests with BT. Like BT, it is an SVD-based method. It can be applied to highly complex systems. The method is frequently used in MOR, not only for the reduction of linear but also explicitly for general nonlinear systems [AS01].

To compute the POD of the quadratic system (3.37), we start by fixing the input u and calculate the state trajectory at certain time instances t_k. These are stored in the *snapshot matrix* $\mathcal{X}$ [AS01], i.e.,

$$\mathcal{X} = [x(t_1)\ x(t_2)\ \cdots\ x(t_\zeta)] \in \mathbb{R}^{n \times \zeta}, \tag{3.67}$$

where ζ is the number of snapshots. There is no general rule how large ζ should be. Essentially, it holds that, the larger n, the larger ζ as well, usually $\zeta \gg n$. We then determine the SVD of $\mathcal{X}$ and look at the behavior of the SVs. If they fall off sharply, we can compute a low-order version of the original system, because

$$\mathcal{X} = V \Sigma W^* \approx V_{n_\rho} \Sigma_{n_\rho} W_{n_\rho}^*, \tag{3.68}$$

where n_ρ is the number of the largest SVDs and the corresponding left singular vectors of $\mathcal{X}$, respectively, and $n_\rho \ll n$ as before. Once more, these n_ρ leading columns of V

span a low-dimensional space in which the state vector $x(t)$ can be approximated by another vector $\hat{x}(t)$. We partition $V = \begin{pmatrix} V_{n_\rho} & V_{n-n_\rho} \end{pmatrix}$ and set the projection matrix

$$\mathcal{V} = V_{n_\rho}. \tag{3.69}$$

Because of (3.68), we have $x(t) \approx \mathcal{V}\hat{x}(t)$ with $\hat{x}(t) \in \mathbb{R}^{n_\rho}$. Using this, we can relate the FOM (3.37) to the ROM (3.39). Putting this in terms of the state-space descriptions of quadratic systems (3.38), we set $\mathrm{T} = \mathcal{V}$ to acquire the quadratic ROM (3.40) [Ant05],

$$\hat{\Sigma} = \left(\begin{array}{c|c|c} \mathcal{V}^* A \mathcal{V} & \mathcal{V}^* H(\mathcal{V} \otimes \mathcal{V}) & \mathcal{V}^* B \\ \hline C\mathcal{V} & & \end{array}\right) \in \mathbb{R}^{(n_\rho + p) \times (n_\rho + n_\rho^2 + m)}. \tag{3.70}$$

From the derivation of the POD method one large drawback is apparent: We receive SVs which are not system-invariant, because the constructed reduction is profoundly contingent on the initial fixation of the input function, as well as the number of snapshots and the time instances at which snapshots are taken [AS01].

Algorithm 3.4 illustrates how we implemented the method to reduce the quadratic dynamic system (3.37) to the ROM (3.39). There are more elaborate and/or efficient ways to obtain the projection matrices (e.g., by optimization of the initial input, the collection of snapshots, the automatic determination of an adequate reduction order or error estimation, see, e.g., [Pin08; RP03; Vol13]), but since we only use this method to evaluate the results of Balanced Truncation, we nevertheless adopt a naive approach to compute the POD.

Algorithm 3.4: Proper Orthogonal Decomposition for Quadratic Systems

```
Input: quadratic system matrices $A, H, B, C$, reduction order $n_\rho$, number of
       snapshots $\zeta$, time interval $T$.
Output: reduced matrices $\hat{A}, \hat{H}, \hat{B}, \hat{C}$, projection matrix $\mathcal{V}$.
Create the snapshot matrix $\mathcal{X}$ by solving the state equation in the ODE system
  (3.37) at $\zeta$ different instances with $\frac{T}{\zeta}$ distance.
Compute the Singular Value Decomposition of $\mathcal{X}$:
    $V\Sigma W^* = \mathcal{X}$,
    where $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$ contains the singular values in decreasing order,
      and $V, W$ contain the corresponding left-singular and right-singular
      vectors, respectively.
Truncate and set the projection matrix $\mathcal{V} = V_{n_\rho}$, which contains the $n_\rho$ singular
  vectors corresponding to the $n_\rho$ largest singular values of $\Sigma$.
Compute reduced-order system:
    $\hat{A} = \mathcal{V}^* A \mathcal{V}$,   $\hat{H} = \mathcal{V}^* H(\mathcal{V} \otimes \mathcal{V})$,   $\hat{B} = \mathcal{V}^* B$,   $\hat{C} = C\mathcal{V}$.
```

CHAPTER 4

Mathematical Modeling

So far, we have pursued two independent matters: We have derived the Effective Network and Synchronous Motor approaches modeling the basic network equations (2.1) which govern the nonlinear dynamics of coupled phase oscillators in Chapter 2. In Chapter 3, we have introduced the basics of MOR and Proper Orthogonal Decomposition, and in particular presented the quadratic version of Balanced Truncation in more detail. In this chapter, we connect the CO models with MOR by BT and POD.

As we have seen in the EN ODE system (2.20) and in the SM ODE system (2.25), both CO approaches model the nonlinear system dynamics using ODEs of the form

$$\frac{2J_i}{\varphi_R}\ddot{\delta}_i(t) + \frac{D_i}{\varphi_R}\dot{\delta}_i(t) = F_i - \sum_{j=1,\, j\neq i} K_{ij} \sin\Big(\delta_i(t) - \delta_j(t) - \gamma_{ij}\Big), \tag{4.1}$$

with $i = 1, \ldots, n_{co}$, and n_{co} depending on the CO model's coupled oscillator representation. For convenience, we restate the meaning of the parameters in Table 4.1.

Table 4.1: Coupled Oscillator Model Parameters

model-independent		model-dependent	
φ_R	reference frequency	F_i	determines the i-th oscillator's inherent frequency $\varphi_i^* := \varphi_R(1 + \frac{F_i}{D_i})$ along with D_i
J_i	inertia	K_{ij}	coupling strength between oscillators i and j
D_i	damping	γ_{ij}	phase shift involved in oscillator coupling
		n_{co}	number of nodes in coupled oscillator representation: $n_{co} = n_g$ for EN model $n_{co} = n_g + n_\ell$ for SM model

By adding the initial conditions

$$\delta(0) = \delta_0, \quad \dot{\delta}(0) = \dot{\delta}_0, \tag{4.2}$$

we turn the ODE system into an initial value problem (IVP). Because the RHS of the ODEs is smooth and bounded, it has a unique solution.

Before we can apply quadratic BT, we must first make sure the system (4.1) satisfies the necessary assumptions and requirements stipulated in Section 3.3, i.e, it must have a quadratic structure, it must be an IVP with initial condition $x_0 = 0$, and it must be stable. The application of POD only demands that the system must have quadratic structure and be an IVP.

First, we give the system the appropriate quadratic structure. This is a two-step process. In Section 4.1, we convert the 2^{nd}-order ODE system into a 1^{st}-order ODE system by substituting the state variable δ with an auxiliary variable. The next step is to bring the system into quadratic form and is divided into two parts: In Section 4.2.1, we transform each nonlinear equation into a quadratic equation, again by substitution, and in Section 4.2.2, we aggregate the equations into matrices. Although strictly speaking, the first part of step 2 is redundant for the further considerations of this thesis, it assists in making the transformation of the ODEs into a quadratic system more comprehensible. The process of structuring a system by inserting auxiliary variables and variable transformations is called *lifting* [KW19]. In Section 4.3, we come back to the initial condition (4.2) and discuss its lifting to agree with the quadratic system. At the end of this chapter, in Section 4.4, we lay out the algorithmic application of BT and POD to the CO models.

4.1 Transformation to Nonlinear Dynamic 1st-Order ODE System

This step is necessary not only en route to the quadratic formulation, but also to solve the ODEs with the help of software like MATLAB, which can only handle 1^{st}-order differential equations.

We transform system (4.1) by substituting each 1^{st}-order state variable with an auxiliary variable and add each substitution to the system of equations. Accordingly, defining $\omega_i(t) := \dot{\delta}_i(t)$, we obtain

$$\begin{bmatrix} \dot{\delta}_i(t) \\ \dot{\omega}_i(t) \end{bmatrix} = \begin{bmatrix} \omega_i(t) \\ \frac{\varphi_R}{2J_i} F_i - \frac{\varphi_R}{2J_i} \sum_{j \neq i} K_{ij} \sin\left(\delta_i(t) - \delta_j(t) - \gamma_{ij}\right) - \frac{D_i}{2J_i} \omega_i(t) \end{bmatrix}, \tag{4.3}$$

with $i = 1, \dots, n_{co}$.

So, merely by bringing the equations into computable form, the system size doubles. As mentioned in Remark 2.1, due to the construction of the CO models, this affects the SM model more than the EN model.

4.2 Transformation to Quadratic Dynamic ODE System

4.2.1 Quadratic Dynamic ODE System - Equation version

The nonlinear component of system (4.1) is the sin-function acting on the difference of the state variables δ_i, δ_j and the phase shift γ_{ij}. Turning the nonlinear system without special structure into a system in quadratic form requires another substitution with auxiliary variables. Before we can replace variables, however, we must first break up the sin-term such that each part only depends on one variable or parameter. To achieve this, we make use of the trigonometric addition formulae

$$\begin{aligned}\sin(a \pm b) &= \sin(a)\cos(b) \pm \cos(a)\sin(b),\\ \cos(a \pm b) &= \cos(a)\cos(b) \mp \sin(a)\sin(b).\end{aligned}$$

Setting $a := \delta_i - \delta_j$ and $b := \gamma_{ij}$, and employing the sine addition formula, we get

$$\sin\left((\delta_i - \delta_j) - \gamma_{ij}\right) = \sin(\delta_i - \delta_j)\cos(\gamma_{ij}) - \cos(\delta_i - \delta_j)\sin(\gamma_{ij}). \tag{4.4}$$

Now, setting $a := \delta_i$ and $b := \delta_j$, and then applying the sine and cosine addition formulae simultaneously to the respective parts in equation (4.4), we obtain

$$\sin(\delta_i - \delta_j) = \sin(\delta_i)\cos(\delta_j) - \cos(\delta_i)\sin(\delta_j), \tag{4.5}$$

$$\cos(\delta_i - \delta_j) = \cos(\delta_i)\cos(\delta_j) + \sin(\delta_i)\sin(\delta_j). \tag{4.6}$$

Putting (4.4), (4.5) and (4.6) together, we acquire

$$\begin{aligned}\sin\left(\delta_i - \delta_j - \gamma_{ij}\right) =& \sin(\delta_i)\cos(\delta_j)\cos(\gamma_{ij}) - \cos(\delta_i)\sin(\delta_j)\cos(\gamma_{ij})\\ &- \cos(\delta_i)\cos(\delta_j)\sin(\gamma_{ij}) - \sin(\delta_i)\sin(\delta_j)\sin(\gamma_{ij}).\end{aligned} \tag{4.7}$$

Replacing the $\sin(\delta_i - \delta_j - \gamma_{ij})$ term in the 1st-order system (4.3) with equation (4.7), substituting $s_i(t) := \sin(\delta_i(t))$, $c_i(t) := \cos(\delta_i(t))$, $\gamma_{ij}^s := \sin(\gamma_{ij})$, $\gamma_{ij}^c := \cos(\gamma_{ij})$, and adding the new variables to the system, we get

$$\begin{bmatrix}\dot{\delta}_i(t)\\ \dot{\omega}_i(t)\\ \\ \dot{s}_i(t)\\ \dot{c}_i(t)\end{bmatrix} = \begin{bmatrix}\omega_i(t)\\ \frac{\varphi_R}{2J_i}F_i - \frac{\varphi_R}{2J_i}\sum_{j\neq i}K_{ij}\Big(s_i(t)c_j(t)\gamma_{ij}^c - c_i(t)s_j(t)\gamma_{ij}^c\\ \qquad - c_i(t)c_j(t)\gamma_{ij}^s - s_i(t)s_j(t)\gamma_{ij}^s\Big) - \frac{D_i}{2J_i}\omega_i(t)\\ c_i(t)\omega_i(t)\\ -s_i(t)\omega_i(t)\end{bmatrix}, \tag{4.8}$$

with $i = 1, \ldots, n_{co}$. So, the system complexity has doubled once more. We must now work with $4n_{co}$ state equations, having started with n_{co}. Again, the lifting has a larger effect on the SM model than on the EN model.

4.2.2 Quadratic Dynamic ODE System - Matrix version

Finally, we want to reassemble the separate equations such that we receive a matrix system of the form similar to the one given in (3.37)[1] with $n = 4n_{co}$,

$$\begin{aligned} \dot{x}(t) &= Ax(t) + H\big(x(t) \otimes x(t)\big) + Bu(t), \\ y(t) &= Cx(t), \\ x(0) &= x_0, \end{aligned} \tag{4.9}$$

where $A \in \mathbb{R}^{4n_{co} \times 4n_{co}}$, $H \in \mathbb{R}^{4n_{co} \times (4n_{co})^2}$, $B \in \mathbb{R}^{4n_{co} \times m}$, $C \in \mathbb{R}^{p \times 4n_{co}}$ and $x(t) \in \mathbb{R}^{4n_{co}}$, $u(t) \in \mathbb{R}^m$, and $y(t) \in \mathbb{R}^p$.

In order to achieve this, we split up the equations into their linear, quadratic and input parts, and use them as building blocks for the system matrices. At the end, we specify an output matrix C, which is also necessary to evaluate the MOR performance in Chapter 5.

The Linear Part

The starting point of the construction is the state vector which we define as[2]

$$x = \begin{bmatrix} \delta \\ \omega \\ s \\ c \end{bmatrix}, \quad \text{where} \quad \begin{aligned} \delta &= [\delta_1, \dots, \delta_{n_{co}}]^*, \\ \omega &= [\omega_1, \dots, \omega_{n_{co}}]^*, \\ s &= [s_1, \dots, s_{n_{co}}]^*, \\ c &= [c_1, \dots, c_{n_{co}}]^*. \end{aligned} \tag{4.10}$$

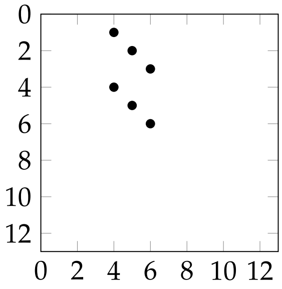
The linear terms of the equation system (4.8) are those which are linear in ω_i. This corresponds to the first n_{co} equations and those terms in the next n_{co} equations with a damping/inertia coefficient. Hence, we define $\boxed{-\frac{D}{2J}} := \operatorname{diag}(\frac{D_1}{2J_1}, \dots, \frac{D_{n_{co}}}{2J_{n_{co}}})$ and form matrix A describing the linear dynamics of the system as

$$A = \begin{bmatrix} 0 & \mathbb{I}_{n_{co}} & 0 & 0 \\ 0 & \boxed{-\frac{D}{2J}} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Figure 4.1 depicts the sparsity pattern of A for an exemplary small PN test system [NM15b; ZM16] modeled by the EN approach. As mentioned at the end of Section 3.3, this matrix is by construction very sparse, has eigenvalues equaling zero and is therefore never stable. This holds independent of the CO model and PN test system. In fact, A in the SM model is usually even sparser than A in the EN model. For example, if we take the same 3-generator/9-nodes system of Figure 4.1, but model it using the SM approach instead, then $n_{co} = 9$ and A is of size (36×36) with only 18 nonzero elements.

[1] The only difference is that we do not (yet) require the system to have a zero initial value. We deal with this requirement for BT in Section 4.3. POD can be applied directly to system (4.9).

[2] For the sake of clarity, we omit the parameter t for the rest of this chapter.



number of nonzero elements: 6

Figure 4.1: Sparsity Pattern of A (EN model, 3-generator/9-nodes system, $n_{co} = 3$)

The Quadratic Part

The quadratic expression of the state vector contains the products of every combination of the state vector elements as they were defined in (4.10), i.e., it is the Kronecker product[3] with itself:

$$x \otimes x = \left[\begin{array}{c|c|c|c} \delta \otimes x & \omega \otimes x & s \otimes x & \underbrace{c \otimes x}_{(4.11)} \end{array} \right]^* .$$

We take a closer look at the last component, $c \otimes x$, to get a better understanding of this vector's architecture. The other three components have a similar structure. Fixing x for the moment, we have

$$c \otimes x = \left[c_1 \otimes x\,, \ldots,\, \underbrace{c_i \otimes x}_{(4.12)}\,, \ldots,\, c_{n_{co}} \otimes x \right]^* . \tag{4.11}$$

Now fixing index i of c and unfixing x, we get

$$c_i \otimes x = \left[c_i\delta_1, \ldots, c_i\delta_{n_{co}},\ c_i\omega_1, \ldots, c_i\omega_{n_{co}},\ c_is_1, \ldots, c_is_{n_{co}},\ c_ic_1, \ldots, c_ic_{n_{co}} \right]^* , \tag{4.12}$$

which is just a $4n_{co}$-sized building block of the entire vector. Doing this with all the $4n_{co}$ elements of the state vector yields a Kronecker product $x \otimes x$ of size $(4n_{co})^2 \times 1$. For convenience, we refer to $x \otimes x$ as the Kronecker state vector.

The matrix H models the nonlinear behavior of the system. We find quadratic components in all except the first n_{co} equations of (4.8). As defined by the Kronecker state vector, these are all the terms containing products of the auxiliary variables ω_i, s_i and c_i with each other. We include their coefficients in H.

Before defining H, we explain some notation. Looking at H in (4.13) below, you can see labels above and on the left side of the matrix. Components or elements of the Kronecker state vector written above (blocks of) the matrix indicate that the associated coefficients are contained in these matrix columns. Components or elements of the state vector derivative found left of the matrix denote to which parts of the derivative H maps to. In this manner, the notation $H(\dot{\omega}, s \otimes x)$ is used to focus on the block of H

[3] For the definition of the Kronecker product, see Appendix B.1.

whose columns contain the coefficients of $s \otimes x$ and whose rows map to the $\dot{\omega}$ part of the state vector derivative. Furthermore, let $\mp\frac{\varphi_R}{2J}K\gamma^{s/c}$ assemble the coefficients corresponding to the $s_i s_j \gamma^s_{ij}$ and $s_i c_j \gamma^c_{ij}$ products in system (4.8), and let $-\frac{\varphi_R}{2J}K\gamma^{c/s}$ likewise collect the $c_i s_j \gamma^c_{ij}$ and $c_i c_j \gamma^s_{ij}$ products. The exponents "s/c" and "c/s" symbolize the order in which the sine and cosine of γ are multiplied with the products. In like manner, "$\mp$" indicates that the term $-s_i s_j \gamma^s_{ij}$ comes before $s_i c_j \gamma^c_{ij}$. The quadratic part of system (4.8) is then represented by matrix H as follows

$$H = \begin{array}{c|ccc|ccc|ccc|ccc|}
 & \multicolumn{3}{c|}{\delta\otimes x} & \multicolumn{3}{c|}{\omega\otimes x} & \multicolumn{3}{c|}{s\otimes x} & \multicolumn{3}{c}{c\otimes x} \\
\dot{\delta} & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\
\dot{\omega} & 0 & \cdots & 0 & 0 & \cdots & 0 & \multicolumn{3}{c|}{\boxed{\cdots \quad \mp\frac{\varphi_R}{2J}K\gamma^{s/c} \quad \cdots}} & \cdots & -\frac{\varphi_R}{2J}K\gamma^{c/s} & \cdots \\
\dot{s} & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & \cdots & \acute{\mathbf{1}} & \cdots \\
\dot{c} & 0 & \cdots & 0 & 0 & \cdots & 0 & \multicolumn{3}{c|}{\boxed{\cdots \quad -\acute{\mathbf{1}} \quad \cdots}} & 0 & \cdots & 0
\end{array}, \tag{4.13}$$

where $\pm\acute{\mathbf{1}}$ symbolizes that "certain" entries in the respective matrix blocks are ± 1. In order to clarify, we first inspect the blue box, i.e., the $H(\dot{c}, s \otimes x)$-block of size $(n_{co} \times 4n_{co}^2)$ you see in (4.14) below. It corresponds to the $H(3n_{co} + 1 : 4n_{co},\ 8n_{co}^2 + 1 : 12n_{co}^2)$ coordinates of H using MATLAB notation. The values under the braces denote the relative column index of the associated $s_i\omega_i$ entry with respect to $s \otimes x$. To know the entry's absolute column index in H, you just have to add $8n_{co}^2$ to the relative index. This number comes from skipping the first $\delta \otimes x$ and $\omega \otimes x$ elements in $x \otimes x$ to arrive at the first entry of $s \otimes x$. Just like $c \otimes x$ in (4.11), each of these components has length $4n_{co}^2$.

The only nonzero elements in $H(\dot{c}, s \otimes x)$ mapping to $\dot{c}$ are the $s_i\omega_i$ products with the same index in system (4.8). We can therefore skip the first n_{co} columns in the $s \otimes x$ part of H corresponding to $s_1 \otimes \delta$. So, the first -1 entry in the block, that assigns $s_1\omega_1$ to $\dot{c}_1$ has the relative column index $n_{co} + 1$. As we move along the Kronecker state vector entries and at the same time along the rows and columns of $H(\dot{c}, s \otimes x)$ to match the corresponding elements, we always have to add the shift of n_{co}. Furthermore, to get from element $s_i\omega_i$ to $s_{i+1}\omega_i$ in the Kronecker state vector, we have an offset of $4n_{co}$ which can be inferred from (4.12). So, moving from row i in $H(\dot{c}, s \otimes x)$ to row $i + 1$, we have to add a shift of $4n_{co}$ to the column index. Plus, we need to add i to hit the right element in the Kronecker state vector. The $H(\dot{c}, s \otimes x)$-block is then

$$\boxed{H(\dot{c}, s \otimes x)} := \begin{array}{c|ccccc|}
 & \cdots & s_1\omega_1 & s_i\omega_i & s_{n_{co}}\omega_{n_{co}} & \cdots \\
\dot{c}_1 & \cdots & \underbrace{\text{-1}}_{n_{co}+1} & & & \cdots \\
\dot{c}_i & \cdots & & \underbrace{\text{-1}}_{n_{co}+(i-1)4n_{co}+i} & & \cdots \\
\dot{c}_{n_{co}} & \cdots & & & \underbrace{\text{-1}}_{4n_{co}^2+n_{co}} & \cdots
\end{array}, \tag{4.14}$$

where $i = 1, \ldots, n_{co}$, and the zero entries are omitted. The $H(\dot{s}, c \otimes x)$-block is analogously constructed.

Next, we address the i-th row of the green-framed box in (4.13), the $H(\dot{\omega}_i, s \otimes x)$-block of size $(1 \times 4n_{co}^2)$. The block matches the $H(n_{co} + i,\ 8n_{co}^2 + 1 : 12n_{co}^2)$ coordinates of H in MATLAB notation. It maps the $s_i s_j$ and $s_i c_j$ products to $\dot{\omega}_i$. Like above, the values under the braces indicate the relative column index of the associated $s_i s_j$ or $s_i c_j$ entry with respect to $s \otimes x$. The absolute column index in H is again attained by adding $8n_{co}^2$ to the relative index.

The nonzero elements in $H(\dot{\omega}_i, s \otimes x)$ mapping to $\dot{\omega}_i$ are the $s_i s_j$ and $s_i c_j$ products, where $i \neq j$ in system (4.8). With the same reasoning as above, we can pass over the first $2n_{co}$ columns in the $s \otimes x$ part of H, skipping the $s \otimes \omega$ component, too. Thus, the first entry in the block (a zero coefficient), which maps $s_1 s_1$ in the Kronecker state vector to $\dot{\omega}_1$, has the relative column index $2n_{co} + 1$. This time, the shift of $2n_{co}$ has to be added to the column index as we advance along the elements in the Kronecker state vector and the columns of $H(\dot{\omega}_i, s \otimes x)$, always passing the δ_i and ω_i elements in $s \otimes x$. To get from entry $s_i s_j$ to $s_{i+1} s_j$, we add another shift of $4n_{co}$ as for the $H(\dot{c}, s \otimes x)$-block. To go from $s_i s_j$ to $s_i c_j$, we have an additional offset of n_{co}.

Defining the shorthand notation $\mp\Gamma_{i,j}^{s/c} = \mp\frac{\varphi_R}{2J_i} K_{i,j} \gamma_{i,j}^{s/c}$, for $i = 1, \ldots, n_{co}$, we have $\boxed{H(\dot{\omega}_i, s \otimes x) :=}$

$$\Big[\cdots \quad \overset{s_i s_1}{\underbrace{-\Gamma_{i,1}^{s}}_{2n_{co}+(i-1)4n_{co}+1}} \quad \cdots \quad \overset{s_i s_i}{\underbrace{0}_{\substack{2n_{co}+(i-1)4n_{co}+i,\\ \Leftrightarrow j=i}}} \quad \cdots \quad \overset{s_i s_j}{\underbrace{-\Gamma_{i,j}^{s}}_{2n_{co}+(i-1)4n_{co}+j}}$$
$$\cdots \quad \overset{s_i c_1}{\underbrace{\Gamma_{i,1}^{c}}_{3n_{co}+(i-1)4n_{co}+1}} \quad \cdots \quad \overset{s_i c_i}{\underbrace{0}_{\substack{3n_{co}+(i-1)4n_{co}+i,\\ \Leftrightarrow j=i}}} \quad \cdots \quad \overset{s_i c_j}{\underbrace{\Gamma_{i,j}^{c}}_{3n_{co}+(i-1)4n_{co}+j}} \quad \cdots\Big],$$

where $j = 1, \ldots, n_{co}$. The block of H corresponding to the $c \otimes x$ components of the Kronecker vector mapping to $\dot{\omega}$ is similarly composed.

The $4n_{co} \times (4n_{co})^2$-sized matrix H is sparse, it only has the coefficients of the sine and cosine substitutes and corresponding ±1's as nonzero entries. Figure 4.2 shows the sparsity pattern of H for the same small PN test case modeled by the EN approach as in Figure 4.1. Again, for every CO model and test system, H features the same sparsity pattern.

Remark 4.1. The matrix H gets large very fast which adds to the computational costs mentioned in Remark 3.35 in connection with the computation of the Volterra kernels.

The Input and Output Parts

Only the constant terms $\frac{\varphi_R}{2J_i} F_i$ have not been dealt with in the equation version of the lifted system (4.8). As the quadratic system (4.9) does not provide for constant terms, we assign these values to the input part of the system which would otherwise be empty.

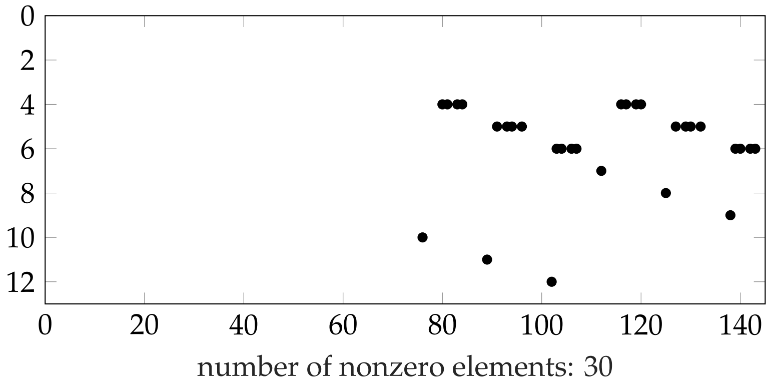


Figure 4.2: Sparsity Pattern of H (EN model, 3-generator/9-nodes system, $n_{co} = 3$)

So, both the input vector u and "matrix" B are almost zero vectors of length $4n_{co}$ with values only in the positions corresponding to $\dot{\omega}$:

$$u = \begin{array}{c} \\ \dot{\omega} \\ \\ \\ \end{array} \begin{bmatrix} 0 \\ \mathbf{1} \\ 0 \\ 0 \end{bmatrix} \in \mathbb{R}^{4n_{co}}, \qquad B^* = \begin{array}{c} \\ \dot{\omega} \\ \\ \\ \end{array} \begin{bmatrix} 0 \\ \boxed{\frac{\varphi_R}{2J} F} \\ 0 \\ 0 \end{bmatrix} \in \mathbb{R}^{4n_{co} \times 1},$$

where $\boxed{\frac{\varphi_R}{2J} F} := \left[\frac{\varphi_R}{2J_1} F_1, \ldots, \frac{\varphi_R}{2J_{n_{co}}} F_{n_{co}} \right]^*$. Note that u is time-invariant here.

Since the coupled oscillator representation does not include an output of any kind, we propose two alternatives to test if the choice makes a difference with respect to the error. Both versions are vectors of size $(1 \times 4n_{co})$.

The first option simply puts out the first oscillator's phase angles as they were computed by solving the ODE system, i.e.,

$$C^{1^{\text{st}}} = \begin{array}{c} \\ y \end{array} \left[\begin{array}{cccc|c|c|c} \delta_1 & \delta_2 & \cdots & \delta_{n_{co}} & \omega & s & c \\ 1 & 0 & \cdots & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array} \right] \in \mathbb{R}^{1 \times 4n_{co}}. \tag{4.15}$$

The second option calculates the arithmetic mean of all the phase angles:

$$C^{\text{am}} = \begin{array}{c} \\ y \end{array} \left[\begin{array}{ccc|ccc|ccc|ccc} \delta_1 & \cdots & \delta_{n_{co}} & \omega_1 & \cdots & \omega_{n_{co}} & s_1 & \cdots & s_{n_{co}} & c_1 & \cdots & c_{n_{co}} \\ \frac{1}{n_{co}} & \cdots & \frac{1}{n_{co}} & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{array} \right] \in \mathbb{R}^{1 \times 4n_{co}}. \tag{4.16}$$

Remark 4.2. Observe that the constructed quadratic representation of the nonlinear PN ODE system (4.1) is not unique, even if we disregard the arbitrary construction of C. To give only one example, since the individual elements in the Kronecker state vector commute, we could have assigned the coefficients corresponding to factor $s_i c_j$ in the $H(\dot{x}, s \otimes x)$-block also to $c_j s_i$ in the $H(\dot{x}, c \otimes x)$-block in the composition of H. The critical point is to only assign every coefficient only once.

Remark 4.3. We implemented all derived representations of the nonlinear PN system (4.1) such that it can be solved in MATLAB. Once you have generated the necessary CO model parameters with the pg_sync_models toolbox [NM15b], the respective functions automatically restructure the model into a general 1st-order nonlinear ODE system, or a quadratic ODE system in equation or matrix representation, respectively. The system can then be solved by built-in MATLAB or self-built ODE solvers, and the quadratic matrix implementation can be reduced by Algorithms 3.3 and 3.4 presented in Section 3.3. You can find the codes in Appendix C.

4.3 Nonzero Initial Values

At the beginning of this chapter, we created the initial value problem by adding $\delta(0) = \delta_0$ and $\dot{\delta}(0) = \dot{\delta_0}$ in (4.2) to the ODE system (4.1). One of the assumptions of Balanced Truncation is an initial value of zero. The state variable x, which we defined in (4.10) in the matrix version of the quadratic system, is a concatenation of vectors constructed from the auxiliary variables in Section 4.2.1. These substitution variables are all functions of δ. So, choosing δ_0 in (4.2) automatically determines the other components of x_0. We set

$$\delta_i(0) = \delta_{0i}, \qquad \omega_{0i} = \dot{\delta_{0i}}, \qquad s_{0i} = \sin(\delta_{0i}), \qquad c_{0i} = \cos(\delta_{0i}),$$

for $i = 1, \dots, n_{co}$. The initial value of the quadratic system (4.9) is thus

$$x_0 = \begin{bmatrix} \delta_0 \\ \omega_0 \\ s_0 \\ c_0 \end{bmatrix}, \quad \text{where} \quad \begin{aligned} \delta_0 &= [\delta_{01}, \dots, \delta_{0n_{co}}]^*, \\ \omega_0 &= [\omega_{01}, \dots, \omega_{0n_{co}}]^*, \\ s_0 &= [s_{01}, \dots, s_{0n_{co}}]^*, \\ c_0 &= [c_{01}, \dots, c_{0n_{co}}]^*. \end{aligned}$$

As $\sin(\delta_{0i})$ and $\cos(\delta_{0i})$ can never simultaneously be zero for the same δ_{0i}, it holds

$$x_0 \neq 0. \tag{4.17}$$

For this reason, we have to adjust the system, so BT can be applied.

Remark 4.4. Not preparing for a nonzero initial value can have unexpected consequences and the result quality can suffer remarkably.

We introduce a new variable $\tilde{x}$ which is zero by definition, i.e., for $x(0) = x_0 \neq 0$, we set

$$\tilde{x} = x - x_0, \qquad \text{and} \qquad \tilde{x}(0) = 0.$$

Using this vector, we transform the entire system in order to obtain an equivalent representation [FKR+04]. We apply the distributivity and associativity as well as the mixed product properties of the Kronecker product (see the 3rd and 6th properties in Theorem B.2), it holds:

$$\begin{aligned} (\tilde{x} + x_0) \otimes (\tilde{x} + x_0) &= (\tilde{x} + x_0) \otimes \tilde{x} + (\tilde{x} + x_0) \otimes x_0 \\ &= \tilde{x} \otimes \tilde{x} + x_0 \otimes \tilde{x} + \tilde{x} \otimes x_0 + x_0 \otimes x_0 \end{aligned} \tag{4.18}$$

and,

$$(\tilde{x} \otimes x_0) + (x_0 \otimes \tilde{x}) = \Big((\mathbb{I} \otimes x_0) + (x_0 \otimes \mathbb{I})\Big)\tilde{x}. \tag{4.19}$$

Using the definition of $\tilde{x}$ to obtain the new representation, we first transform the state equations for which we get

$$\begin{aligned}\dot{\tilde{x}} &= Ax + H(x \otimes x) + Bu \\ &= A(\tilde{x} + x_0) + H\Big((\tilde{x} + x_0) \otimes (\tilde{x} + x_0)\Big) + Bu.\end{aligned}$$

Exploiting the relations (4.18) and (4.19) yields

$$\begin{aligned}\dot{\tilde{x}} &\overset{(4.18)}{=} A\tilde{x} + Ax_0 + H\Big((\mathbb{I} \otimes x_0) + (x_0 \otimes \mathbb{I})\Big)\tilde{x} + H(\tilde{x} \otimes \tilde{x}) + H(x_0 \otimes x_0) + Bu \\ &\overset{(4.19)}{=} \Big(A + H\Big((\mathbb{I} \otimes x_0) + (x_0 \otimes \mathbb{I})\Big)\Big)\tilde{x} + H(\tilde{x} \otimes \tilde{x}) + \Big[B \;\; Ax_0 + H(x_0 \otimes x_0)\Big]\Big[u \;\; 1\Big]^*.\end{aligned}$$

Then, we transform the output equations of the quadratic system,

$$\tilde{y} = Cx = C(\tilde{x} + x_0) = C\tilde{x} + Cx_0. \tag{4.20}$$

Now, we can redefine the matrices and vectors of the transformed quadratic system:

$$\begin{aligned}\widetilde{A} &= A + H\Big((\mathbb{I} \otimes x_0) + (x_0 \otimes \mathbb{I})\Big), \\ \widetilde{B} &= [B \;\; Ax_0 + H(x_0 \otimes x_0)], \\ \widetilde{H} &= H, \\ \widetilde{C} &= C, \\ \tilde{u} &= [u \;\; 1]^*.\end{aligned} \tag{4.21}$$

Once more, Figure 4.3 exemplifies on a small scale what holds for both CO models and all PN test systems: This transformation makes $\widetilde{A}$ less sparse, but the matrix is nonetheless not stable. As the sparsity pattern shows, $\widetilde{A}$ must have EVs equaling 0, i.e., $\widetilde{A}$ is only semi-stable.

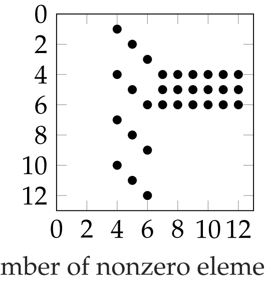


Figure 4.3: Sparsity pattern of $\widetilde{A}$ (EN model, 3-generator/9-nodes system, $n_{co} = 3$)

Remark 4.5. We also use the transformed system (4.21) to solve the FOM, when we compare solutions of ROMs obtained by BT.

4.4 Application of MOR to the Quadratic Dynamic Power Network System

With the transformed matrices (4.21), the CO models are in the required quadratic form (3.37) with zero initial value. However, a basic prerequisite for the application of BT is a stable system as it is necessary in the derivation of the reachability and observability Gramians, the computation of the Lyapunov equations in Theorems 3.19, 3.31 and 3.34, and the Lyapunov-Cholesky routine in Algorithm 3.1. By Theorem 3.15, system stability demands a stable matrix A. But even after the transformation in Section 4.3, $\widetilde{A}$ is only semi-stable and therefore does not satisfy the necessary condition for system stability.

To make $\widetilde{A}$ stable, we subtract a small positive parameter α from its diagonal, i.e., we shift the matrix with $\widetilde{A}_\alpha = \widetilde{A} - \alpha\mathbb{I}$. This step is independent of the nonzero transformation and we only apply it to $\widetilde{A}$ and no other system matrices or vectors. Also, the shift is restricted to the balancing part of BT, i.e., the computation of the Gramians (which subsequently influences the construction of the projection matrices). The actual truncation is applied to the unshifted matrix $\widetilde{A}$. The thus reduced $\hat{A}$ is used to solve the ODE. Furthermore, POD does not assume a stable system, so there is no need to shift A.

The system is now ready for Model Order Reduction. We apply BT to the system

$$\widetilde{\Sigma} = \left(\begin{array}{c|c|c} \widetilde{A} & \widetilde{H} & \widetilde{B} \\ \hline \widetilde{C} & & \end{array}\right), \tag{4.22}$$

with transformed zero initial value, $\tilde{x}(0) = 0$. The subsequent balancing computations depend on the shift α. We therefore mark the respective matrices with a subscripted α. The basis of the BT projection matrices are the Cholesky factors $\mathcal{R}_\alpha$ and $\mathcal{S}_\alpha$ of the truncated reachability and observability Gramians, $\mathcal{P}_{\alpha,\mathcal{T}}$ and $\mathcal{Q}_{\alpha,\mathcal{T}}$, respectively. To compute the Gramians, we determine their Volterra kernels by solving the Lyapunov equations in (3.48) and (3.54) for the transformed system with a stable matrix $\widetilde{A}_\alpha$ with Algorithm 3.2[4] which uses the Lyapunov-Cholesky Algorithm 3.1 as a subroutine.

The parameter ν, i.e., number of Volterra kernels, determines where the infinite sums in Definitions 3.30 and 3.33 are cut off. We then compute the Cholesky factors $\mathcal{P}_{\alpha,\mathcal{T}} = \mathcal{R}_\alpha^*\mathcal{R}_\alpha$ and $\mathcal{Q}_{\alpha,\mathcal{T}} = \mathcal{S}_\alpha^*\mathcal{S}_\alpha$, as in (3.30). The SVD (3.31) of the product of the Cholesky factors $V_\alpha\Sigma_\alpha W_\alpha^* = \mathcal{R}_\alpha\mathcal{S}_\alpha^*$ provide the matrices V_α and W_α consisting of the left- and right-singular vector corresponding to the SVs in Σ_α in decreasing order. With these matrices, we construct the BT projection matrices as in (3.35), $\mathcal{V}_\alpha = R_\alpha^* V_{\alpha,n_\rho}\Sigma_{\alpha,n_\rho}^{-\frac{1}{2}}$ and $\mathcal{W}_\alpha^* = \Sigma_{\alpha,n_\rho}^{-\frac{1}{2}} W_{\alpha,n_\rho}^* S_\alpha$.

If the decline of the SVs is fast, we can approximate the FOM by the ROM via the projection matrices. Specifically, the state vector $\tilde{x}$ is related to the reduced state vector $\hat{x}$ by $\tilde{x} \approx \mathcal{V}\hat{x}$, i.e., $\hat{x}$ is an approximation of $\tilde{x}$ that lives in a low-dimensional space spanned

[4] Plus, we use the counterpart of the $\mathcal{P}_{\alpha,\mathcal{T}}$ algorithm to compute $\mathcal{Q}_{\alpha,\mathcal{T}}$, see Algorithm C.1.

by the columns of $\mathcal{V}_\alpha$. The ROM is then

$$\begin{aligned}\dot{\hat{x}}(t) &= \mathcal{W}_\alpha^* \widetilde{A} \mathcal{V}_\alpha \hat{x}(t) + \mathcal{W}_\alpha^* \widetilde{H}(\mathcal{V}_\alpha \otimes \mathcal{V}_\alpha)\big(\hat{x}(t) \otimes \hat{x}(t)\big) + \mathcal{W}_\alpha^* \widetilde{B} \tilde{u}(t), \\ \hat{y}(t) &= \widetilde{C} \mathcal{V}_\alpha \hat{x}(t), \quad \hat{x}(0) = \mathcal{W}_\alpha^* \tilde{x}(0).\end{aligned} \tag{4.23}$$

As POD is not as demanding as BT, we apply the method to the system

$$\Sigma = \left(\begin{array}{c|c|c} A & H & B \\ \hline C & & \end{array}\right), \tag{4.24}$$

with a given initial value $x(0) = x_0$. The POD projection matrices are based on the snapshot matrix $\mathcal{X}$, (3.67). Its size is determined by the parameter ζ, the number of snapshots taken of the state trajectory. The SVD (3.68) of the snapshot matrix $V\Sigma W^* = \mathcal{X}$, likewise, provides the left- and right-singular vector corresponding to the SVs in Σ in decreasing order. We get the projection matrix $\mathcal{V} = \mathcal{W} = V_{n_\rho}$, as in (3.69).

In analog to BT, the projection of the state vector, $\tilde{x} \approx \mathcal{V}\hat{x}$, relates the FOM to the ROM. Should the SVs fall off rapidly, we can approximate the FOM by the ROM via the projection matrices, i.e.,

$$\begin{aligned}\dot{\hat{x}}(t) &= \mathcal{W}^* A \mathcal{V} \hat{x}(t) + \mathcal{W}^* H(\mathcal{V} \otimes \mathcal{V})\big(\hat{x}(t) \otimes \hat{x}(t)\big) + \mathcal{W}^* B u(t), \\ \hat{y}(t) &= C \mathcal{V} \hat{x}(t), \quad \hat{x}(0) = \mathcal{W}^* x(0).\end{aligned} \tag{4.25}$$

The size of the projection matrices (and by extension the size of the reduced system) depends on the reduction order n_ρ. The value of this parameter is in turn determined by the decline of the respective SVs. The steeper they fall, the smaller n_ρ can be chosen.

The reduced ODE systems (4.23) and (4.25) can now be solved.

The state x has the structure as in (4.10), with sin- and cos-auxiliary variables in the third and fourth component. We want to check, among other things, if the sum of the squares of the respective state elements satisfy the PTI. Before we can measure this, we need to pull the reduced vector back into the high-dimensional lifted space by applying the projection matrix $\mathcal{V}_\alpha$ from the left, $\tilde{x} = \mathcal{V}_\alpha \hat{x}$ for BT and $x = \mathcal{V}\hat{x}$ for POD. The ODE solution determined with the help of POD is now ready for the error analysis. Before we can start the analysis for BT, however, we require to undo the transformation due to the nonzero initial value. To that end, we add Cx_0 to output $\hat{y}$ (see equation (4.20)), and the nonzero initial value x_0 to the pulled-back state vector (see equation (4.17)), i.e., the ODE solution state vector aided by BT is $x = \mathcal{V}_\alpha \hat{x} + x_0$.

Table 4.2 summarizes the features of the two MOR methods.

Table 4.2: Attributes of MOR Methods

	BT	POD
key matrix	reach. Gram. $\mathcal{P}_{\alpha,\mathcal{T}}$, Def. 3.30, obs. Gram. $\mathcal{Q}_{\alpha,\mathcal{T}}$, Def. 3.33	snapshot matrix $\mathcal{X}$ (3.67)
decomposition	Cholesky (3.30), SVD (3.31)	SVD (3.68)
nonzero transformation	yes	no
α-shift of A	yes	no
reduction order	n_ρ	n_ρ
projection matrices	$\mathcal{V}_\alpha, \mathcal{W}_\alpha$ (3.35)	$\mathcal{V} = \mathcal{W}$ (3.69)
state approximation	$\tilde{x} \approx \mathcal{V}_\alpha \hat{x}$	$x \approx \mathcal{V}\hat{x}$
parameters	# Volterra kernels ν, shift α	# snapshots ζ

CHAPTER 5

Numerical Experiments for a Power Network Test Case

We now want to look at the results from the numerical testing[1]. This chapter is divided into two parts. In the first section, we present the test setup and parameters. This includes the choice of the power network test case, the ODE solver, the BT and POD parameters and the evaluation metrics. In the second part, we analyze the results from the numerical computations according to the selected metrics.

5.1 Test Setup and Parameters

Selection of Power Network System

The main test PN system is the *IEEE case57* power flow system which is included in the *MATPOWER* software toolbox [ZM16; ZMT11]. It consists of 57 nodes, thereof 7 generator nodes and 50 load nodes. We select this test case for two reasons: First, it is the largest which is still computable[2] for both the EN and SM model. Second, the ratio of load nodes to generator nodes is high compared to other available test cases. Since we reduce from the electric circuit representation of the power network, it is interesting to see how the reduction methods perform for the two models, considering that the number of oscillators is $n_{co} = 7$ for the EN and $n_{co} = 57$ for the SM model.

Figure 5.1 shows two illustrations of case57. In Figure 5.1a, made with the Python toolbox pandapower[3] [TSS+18], you can see the power flow solution of case57, where the colors indicate the node voltage[4] and the gray squared-in node is the reference generator node. On the right, in Figure 5.1b, is a so-called one-line diagram of the test system [GRA14]. The generators are consecutively numbered $1 - 7$ and represented by the voltage source symbol (the tilde within a circle). All other nodes and numbers depict

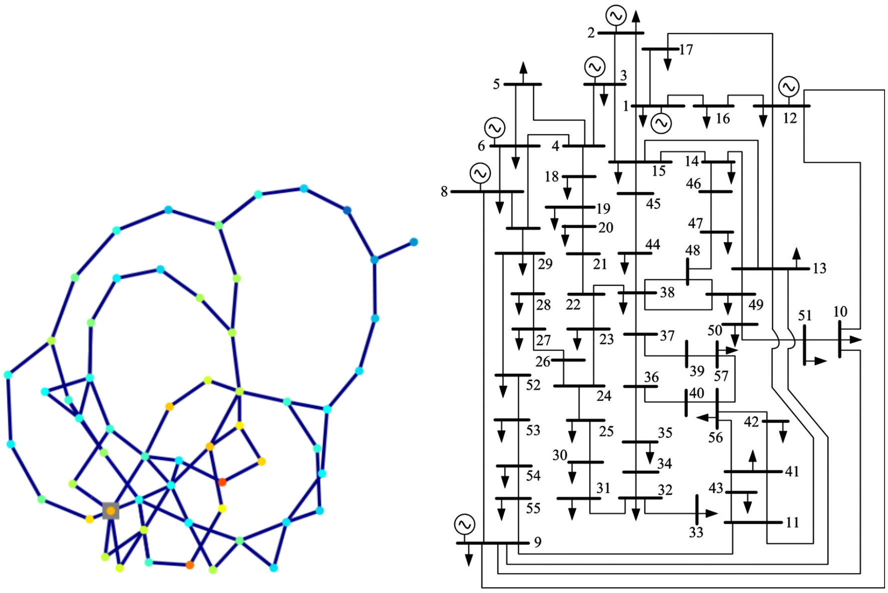
[1] You can find the functions and scripts which generated the data and figures as well as the data themselves in Appendix C

[2] As mentioned in Remark 3.35, forming the Kronecker product of the Volterra kernels, and the quadratic product of $H = \widetilde{H}$ and a factor of the Volterra kernel in Algorithm 3.2 results in large nonsparse matrices which are computationally costly.

[3] The figure was generated by Dr. Manuel Baumann.

[4] Because these values are irrelevant to our analysis, we disregard them here.

load nodes.

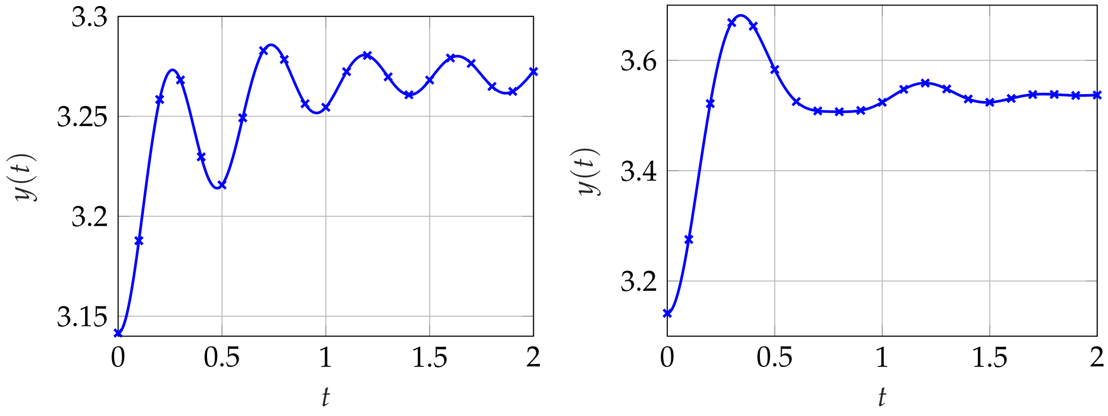


(a) pandapower (b) line diagram

Figure 5.1: IEEE case57 power flow system

For the construction of the coupled oscillator models, we use the MATLAB toolbox *pg_sync_models* [NM15b] provided by [NM15a]. It supplies the model-dependent parameters F_i, K_{ij} and γ_{ij} as well as the model-independent network constants J_i, D_i and φ_R, which are necessary to solve the ODE system.

In Figure 5.2, you see the output-over-time plots for the FOMs of the EN and SM models.



(a) EN model (b) SM model

Figure 5.2: Output of Full-Order Model for case57

Standard Test Parameters[5]

We solve the ODE system with a Runge-Kutta-4 scheme using step size $h = 0.001$ as the standard for the BT tests, and ode45 as the standard for the POD tests[6].

The standard time interval is $T = [0; 2]$, because the assumptions for the CO models are only valid for short time scales[7] [NM15a]. To illustrate the convergence behavior of the ROM output, we also look at a longer time interval of $T = [0; 10]$.

The standard choice for the mathematical CO ODE system representation is the quadratic system in matrix version (4.9).

We test two initial values, where $\delta_0 \in \{0, \pi\}$, and two output matrices: $C^{1^{\text{st}}}$, (4.15), the output of the first oscillator's phase angles, and C^{am}, (4.16), the arithmetic mean of the phase angles. This makes four (δ_0, C) testing pairs. The standard choice is $(\pi, C^{1^{\text{st}}})$.

Table 5.1 provides an overview of those tested parameters that are not influenced by the choice of the CO model.

Table 5.1: Model-Independent MOR Parameters for case57

	standard	also tested
test case PN	case57	–
# generators n_g	7	–
# nodes n	57	–
ODE solver	Runge-Kutta-4	ode45 for POD
time interval T	$[0; 2]$	$[0; 10]$
shift α	0.05	$[0, 0.5]$ with step size 0.01
(δ_0, C)	$(\pi, C^{1^{\text{st}}})$	(π, C^{am}), $(0, C^{1^{\text{st}}})$, $(0, C^{\text{am}})$

The start and step values of the reduction order n_ρ depend on the CO model. There is no standard choice for the latter parameter. Throughout the numerical experiments, we vary n_ρ.

Concerning the BT-specific parameters, there are three main values which can be modified: the number of Volterra kernels ν, and the α shift of matrix A, i.e., A_α. In some instances, the method is tested with different numbers of Volterra kernels, $\nu \in \{1, 3, 5\}$, with the standard choice of $\nu = 3$. We abbreviate BT with $1, 3$ and 5 Volterra kernels with BT1, BT3 and BT5, respectively. We examine α-shifts ranging between 0 and 0.05 with a step size of 0.01, selecting $\alpha = 0.05$ as the standard. As mentioned in Section 4.4, the

[5] When not otherwise mentioned, we tested with the parameters denoted as *standard*.

[6] When we started, we also used the Runge-Kutta-4 scheme for the POD tests. When during testing some re-programming was necessary for the POD method, it was more convenient to switch to ode45. Since the Runge-Kutta-4 scheme is just as accurate as the ode45 solver for the BT method, this has no influence on the results.

[7] See, e.g., the assumptions for the classical model on Page 9.

shift only affects the computation of the projection matrices. The actual ODEs system is solved with the unshifted matrix $\widetilde{A}$.

For our naive implementation of the POD method, there is only one adjustable parameter: the number of snapshots ζ. As discussed in Section 3.4, there is no rule what size ζ should be in general. The goal is to use as few snapshots as possible but as many as necessary[8]. Sample experiments showed that once a certain threshold of snapshots is reached, increasing the number of snapshots does not improve the performance.

In Table 5.2 you can find the test parameters which depend on the the selected CO model.

Table 5.2: Model-Dependent MOR Parameters for case57

	EN model	SM model
red. order n_ρ start	4	8
red. order n_ρ step	4	10
red. order n_ρ end	24	218
size FOM $(4n_{co})$	28	228
size 1^{st}-ord. sys. $n_{\min}$	14	114
# snapshots ζ	981	4333

Evaluation Measures

We examine three measures overall:

(i) output behavior $y(t)$,

(ii) singular values σ,

(iii) L_∞ error.

Considering the first item, this is simply an output-over-time plot. As we stated in Remark 4.5, we used the nonzero transformed system (4.21) to solve the ROM as well as the FOM. So, we plot $\hat{y}(t) = C\hat{x}(t) + Cx_0$ and $y(t) = C\tilde{x}(t) + Cx_0$ for $t \in T$. This direct comparison of the outputs for the FOM and ROM provides an intuition of the approximation quality and helps to assess the L_∞ approximation error, the third evaluation measure.

[8] We employed the formula $\zeta = \beta n_{co}^2 + 1$, where $\beta = 20$ for the EN model, and $\beta = \frac{4}{3}$ for the SM model. This model-dependent factor is doubled, if the POD should fail due to too few snapshots. The snapshot factor β made it easier during programming to scale the number of snapshots ζ. Computing the time step distances $\frac{T}{\zeta}$, then starting to the build the snapshot matrix at $t_0 = 0$ and ending at $t_\zeta = T$ adds 1 to the number of snapshots

Specifically, in Section 5.2, we look at the output-over-time for BT3 with time intervals $T = [0; 2]$ in Figure 5.3, and $T = [0; 10]$ in Figure 5.4, as well as for POD with a time interval of $T = [0; 10]$ in Figure 5.5.

Concerning the second measure, in Section 5.5, we look at the SVs of the Gramians, the Hankel SVs (computed with Algorithm 3.3) in Figure 5.9, and the SVs of the snapshot matrix $\mathcal{X}$ (computed with Algorithm 3.4). We actually examine the $\mathcal{X}$-SVs' behavior not only for the snapshot matrix of the quadratic system (4.9), but in addition the SVs for the snapshot matrix of the 1st-order nonlinear system (4.3)[9] in Figure 5.10.

In our evaluation, the SVs are normalized such that the largest SV is 1, i.e.,

$$\breve{\sigma} = \frac{\sigma}{||\sigma||}.$$

We evaluate two features by measuring the third item, the absolute L_∞ error. First, we examine the distance between the FOM output $y(t)$ and the ROM output $\hat{y}(t)$, that is,

$$\epsilon_\infty = ||y(t) - \hat{y}(t)||_{L_\infty} = \max_{t \in T} |y(t) - \hat{y}(t)|, \tag{5.1}$$

for which we use the MATLAB norm function. We look at this measure in our cross-method comparison of the output error in Section 5.3, for the time interval $T = [0; 2]$ in Figure 5.6 and for the time interval $T = [0; 10]$ in Figure 5.7. We also evaluate the measure for BT3 when we compare the combinations of different initial values and output matrices in Section 5.6 in Figure 5.11. It is used, too, to investigate the influence of the α shift value in Section 5.7, again for BT3 in Figure 5.12.

Second, we also look at the L_∞ error of the *PTI*, i.e., we check if

$$s^2(t) + c^2(t) = \mathbf{1},$$

holds, where $s, c \in \mathbb{R}^{4n_{co}}$ are the sine and cosine components (4.10) of the pulled back and back-transformed state vector $x = \mathcal{V}_\alpha \hat{x} + x_0$ as described at the end of Section 4.4 on Page 61. Using MATLAB's max function, we first compute

$$\epsilon_\infty^{\bowtie} = ||(s^2(t) + c^2(t)) - \mathbf{1}||_{L_\infty} = \max_{t \in T}((s^2(t) + c^2(t)) - \mathbf{1}), \tag{5.2}$$

which gives us an absolute L_∞ error estimate for each oscillator. If $\epsilon_\infty^{\bowtie} \geqslant 10^3$, the error is considered to be $\epsilon_\infty^{\bowtie} = \infty$. We then select the best and worst performing oscillator based on the mean $\epsilon_\infty^{\bowtie}$ value across reduction orders n_ρ. If there are Inf or NaN values, we set them to 10^7, but only for the computation of the mean, such that if there are Inf or NaN values, you can "spot" them by their absence in the plots without distorting the

[9] The unlifted system is only half as large as the quadratic system. However, because this system lacks structure, the nonlinear terms must be estimated by other techniques, termed "hyper-reduction". This provokes a second approximation of the system and can become costly, depending on the degree of nonlinearity in the system. As [KW19] note, the lifted system representation is not unique but exact, i.e., there is no extra approximation necessary. POD can be directly applied to the lifted system and exploit the added structure.

curve. We choose the two oscillators per MOR method with maximum and minimum mean. Investigating the PTI is another means to evaluate the quality of the MOR. An advantage is the identity's independence of the choice of the output matrix. We examine this measure in Section 5.4 for BT1, BT3, BT5 and POD in Figure 5.8.

To improve interpretability, all SVs $\breve{\sigma}$ and both L_∞ errors, ϵ_∞ and $\epsilon_\infty^{\bowtie}$, are plotted on a logarithmic scale.

Before we evaluate the test results, we need to keep two things in mind: First, we quadruple the size of the CO models by lifting them to quadratic form. However, since we want to solve the ODE system by numerical software, we would have to double the original size of n_{co} anyway to transform the 2nd-order ODE systems of size n_{co} to a 1st-order ODE systems. We therefore define

$$n_{\min} = 2n_{co}, \quad \text{with} \quad n_{\min}^{\text{EN}} = 14, \quad \text{and} \quad n_{\min}^{\text{SM}} = 114, \tag{5.3}$$

the minimum size of the system such that it can be solved by numerical software. Second, we simultaneously want a low reduction order n_ρ and a small L_∞ error. Relating this to the first reminder, we may consider a MOR a success, if we have an acceptable L_∞ error and $n_\rho < n_{\min}$ is valid for the associated reduction order.

Furthermore, we point out that the balancing and truncating parts of BT in principle worked in all tests[10]. When we declare that a BT method "failed" or did "not work" or the like in the following sections, we do not mean that the method itself could not be performed, but that the subsequent solution of the ODE system based on the BT reduction failed to produce a computable error, i.e., an error that did not result in `Inf` or `NaN`. The only exception is the examination of the α-shift influence, which we discuss in Section 5.7.

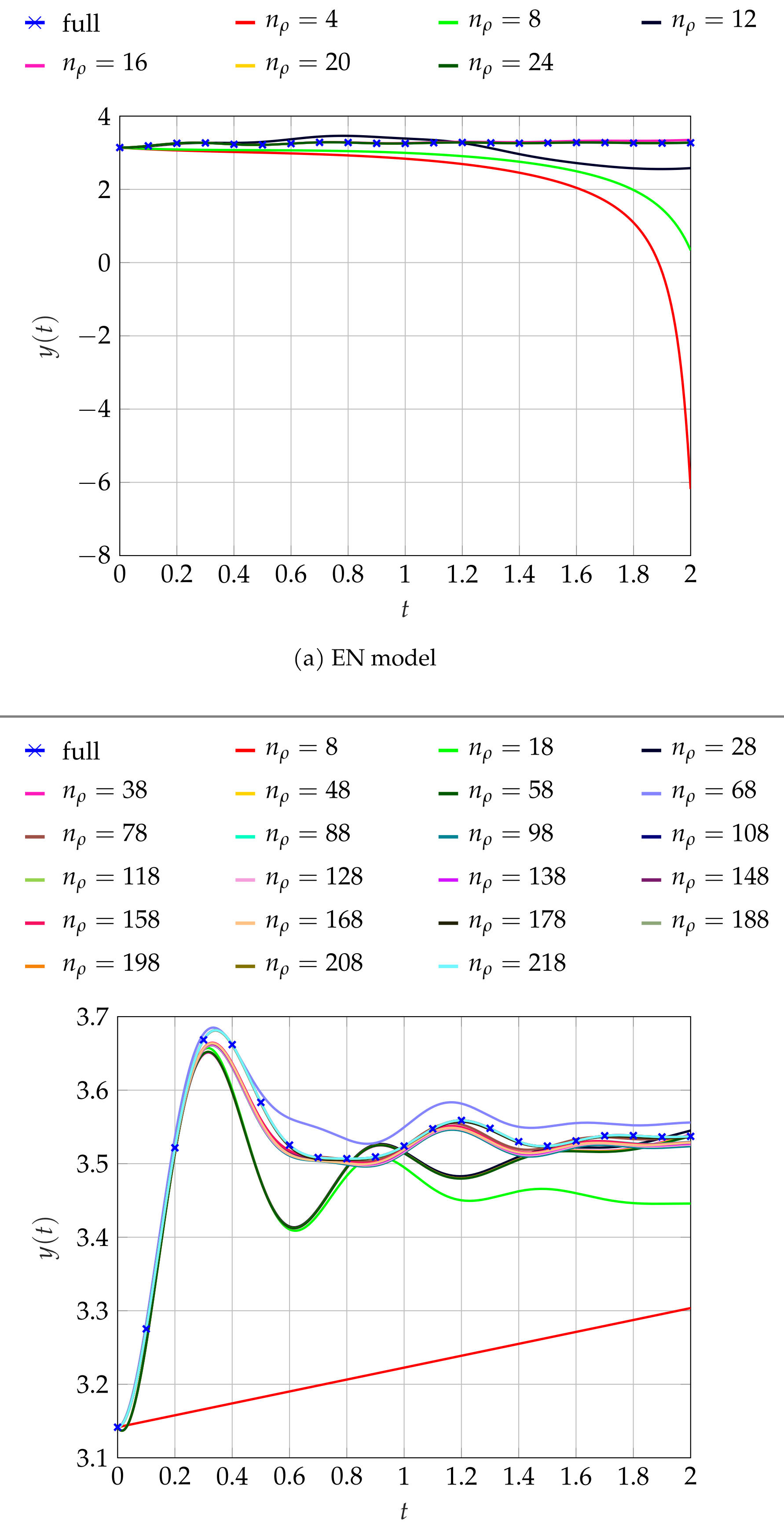
In our analysis of the evaluation items, we always start by examining Balanced Truncation (and its versions, where applicable), first for the EN, second for the SM model, then we make a cross-model comparison, and eventually contrast the findings with the performance of Proper Orthogonal Decomposition (POD).

5.2 Proof-of-Concept: Output-over-Time Behavior

We begin our evaluation by examining the output-over-time for different reduction orders n_ρ in Figures 5.3 and 5.4 for BT3, and Figure 5.5 for POD.

EN Model: We see in Figure 5.3a that the output for the first three reduction order values $n_\rho \in \{4, 8, 12\}$ diverges from the FOM output already in the 2 seconds interval. As we move to the 10 seconds interval in Figure 5.4a, we omit these reduction

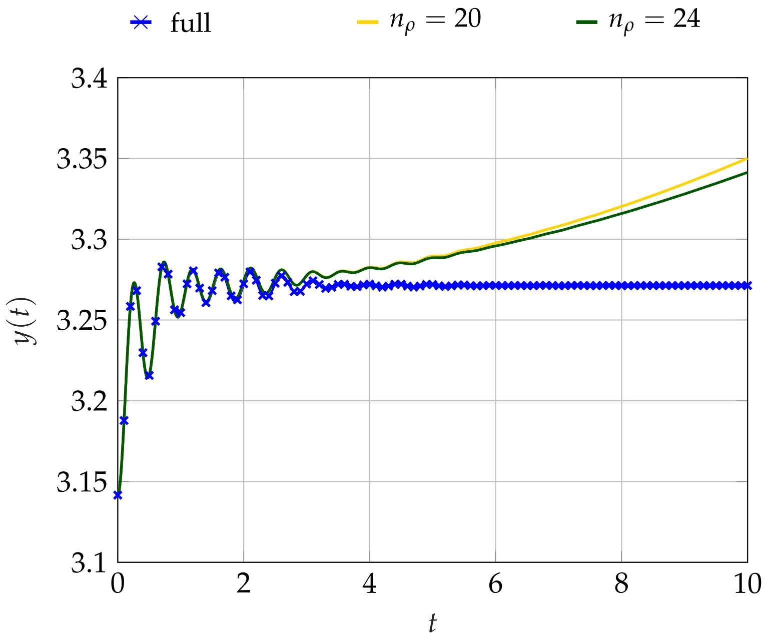
[10] However, whenever the Cholesky factors of the Volterra kernels and/or Gramians were computed with Algorithm 3.2 (and Algorithm C.1), it was almost always necessary to approximate the respective matrix with an symmetric positive definite (SPD) matrix (see Appendix C).



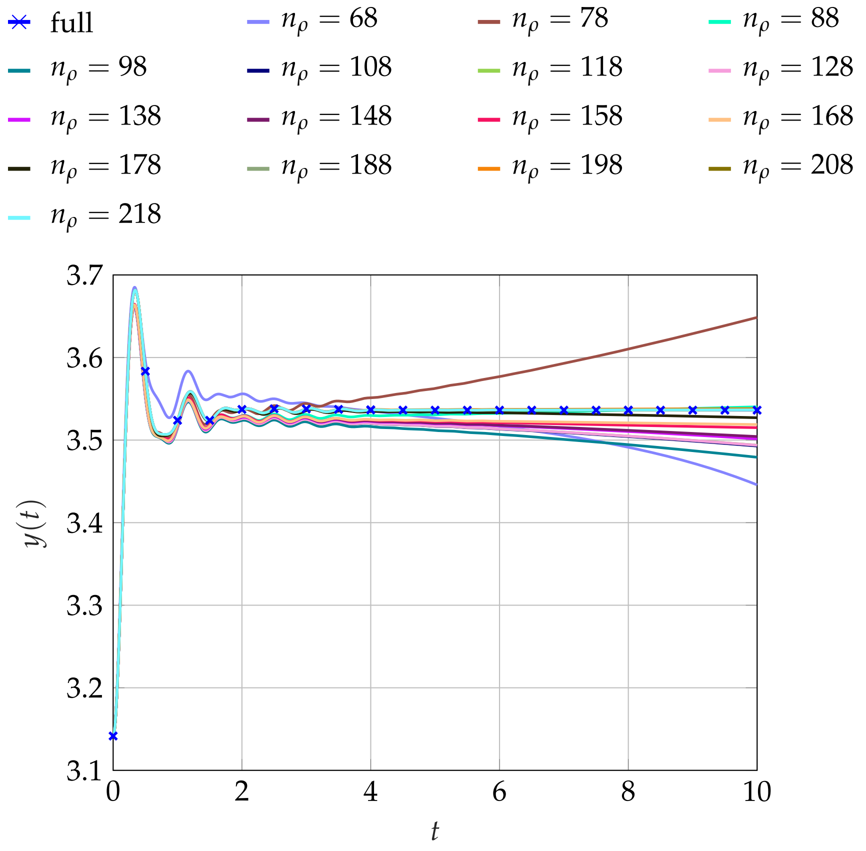
(a) EN model

(b) SM model

Figure 5.3: Output of BT3



(a) EN model



(b) SM model

Figure 5.4: Output of BT3 for 10s Interval

orders and in addition $n_\rho = 16$ to avoid distortion. For almost 3 seconds, the output for the remaining orders stays relatively close to the FOM output. Nonetheless, after that time instance, the output of the EN model for the remaining two reduction orders also branches out.

SM Model: The SM model in Figure 5.3b looks a little more promising. A higher proportion of reduction orders seems to produce output which stays within a short distance of the FOM output. However, some of the lower reduction orders generated interesting output behavior. The output for $n_\rho = 8$ looks linear. The lines relating to $n_\rho \in \{18, 28, 38, 48, 58\}$ demonstrate an almost inverse development between 0.6 and 1.4 seconds, after staying close to the FOM output for the first 0.3 seconds. The latter is also true for the output line of $n_\rho = 68$. Subsequently, it mimics an upwards shifted FOM output.

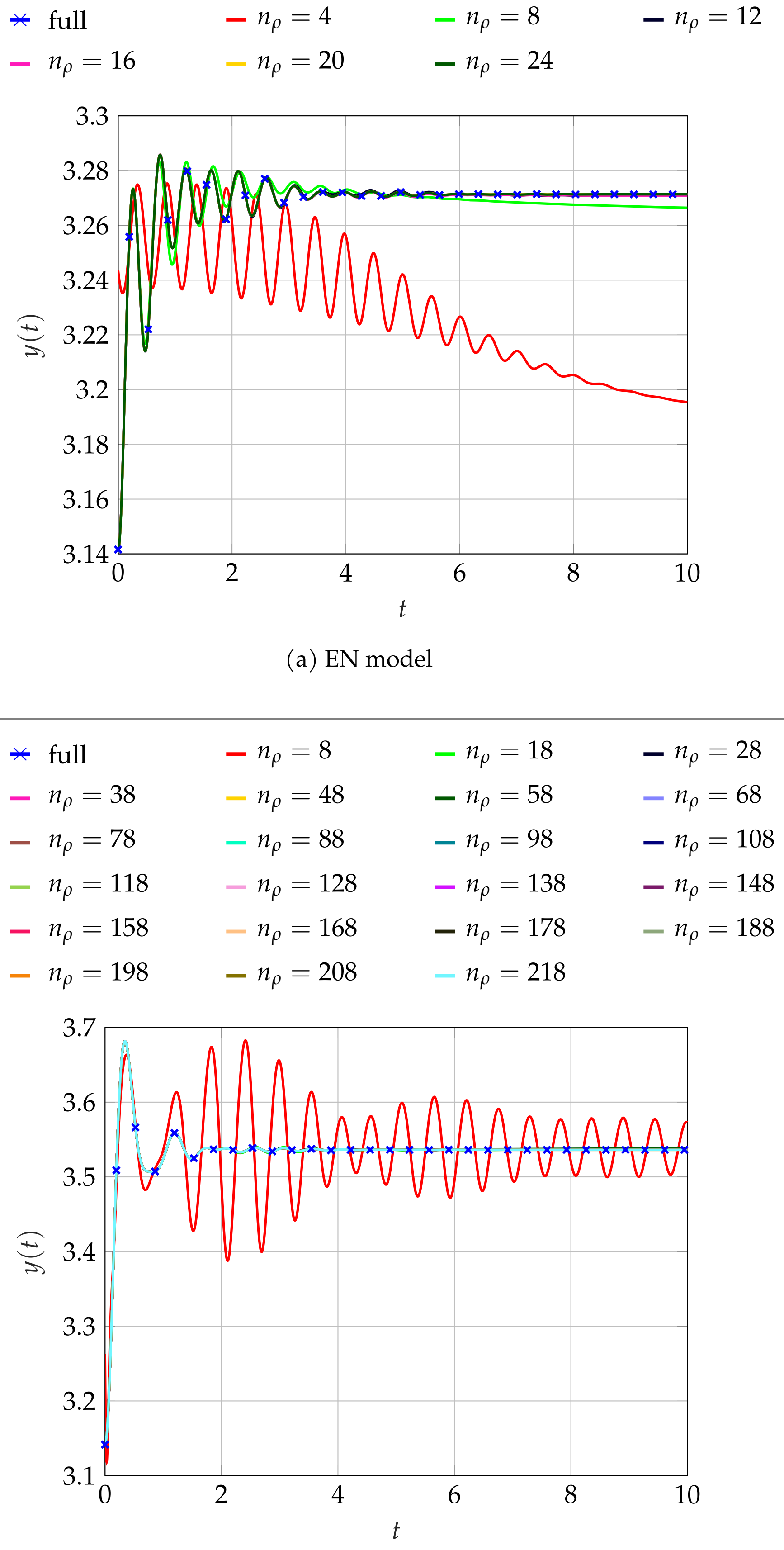
Analogously to the EN model, we eliminate these worst reduction orders to get a clearer picture in Figure 5.4b. After the omission only the outputs relating to $n_\rho \geqslant 68$ are left. In the first 3 seconds the output for these orders keep close to the FOM output; except the output for $n_\rho = 68$, which retains its shifted progression during this time. Until the end of the 10 seconds interval, however, output for $n_\rho = 68$ diverges almost as much as the one for $n_\rho = 78$. In contrast, the output for $n_\rho = 88$ seems to approximate the FOM well, while the output related to $n_\rho \in \{98, 108, 118, 128, 138, 148, 158, 168\}$ deviates noticeably. This leaves only output produced by reduction orders $n_\rho \geqslant 178$ within imperceptible distance of the FOM output.

EN vs. SM Model: From Figure 5.3, we discern that BT3 does not accomplish a good approximation of the FOM for low- to mid-range reduction orders regarding both CO models. In the standard time interval of 2 seconds, the EN model output only stayed close to the FOM for reductions orders $n_\rho \geqslant 16 \geqslant n^{\text{EN}}_{\text{min}} = 14$. The reduction performance of SM is better, considering $n^{\text{SM}}_{\text{min}} = 114$ in the same interval. The four output lines relating to reduction orders between $78 \leqslant n_\rho \leqslant 108$ satisfy $n_\rho \leqslant n^{\text{SM}}_{\text{min}} = 114$ and seem to progress near the FOM output.

We compare the observations concerning the output with the L_∞ output error in Section 5.3, before we draw more conclusions on the performance of BT3.

BT3 vs. POD: We focus immediately on the plot with a time interval of $T = [0; 10]$ in Figure 5.5 and compare the findings thus far to the output approximations achieved by POD. Evidently, this MOR method's output aligns with the FOM output already for reduction orders of $n_\rho \geqslant 12$ for the EN model in Figure 5.5a, with $n_\rho = 8$ also seeming within reasonable distance. POD performs even better for the SM model in Figure 5.5b. Merely for $n_\rho = 8$, the reduced output oscillates with larger distance around the actual output values. All other reduction orders seem to immediately align with the full-order output[11]. So, judging by the output behavior, the POD performance is more accurate

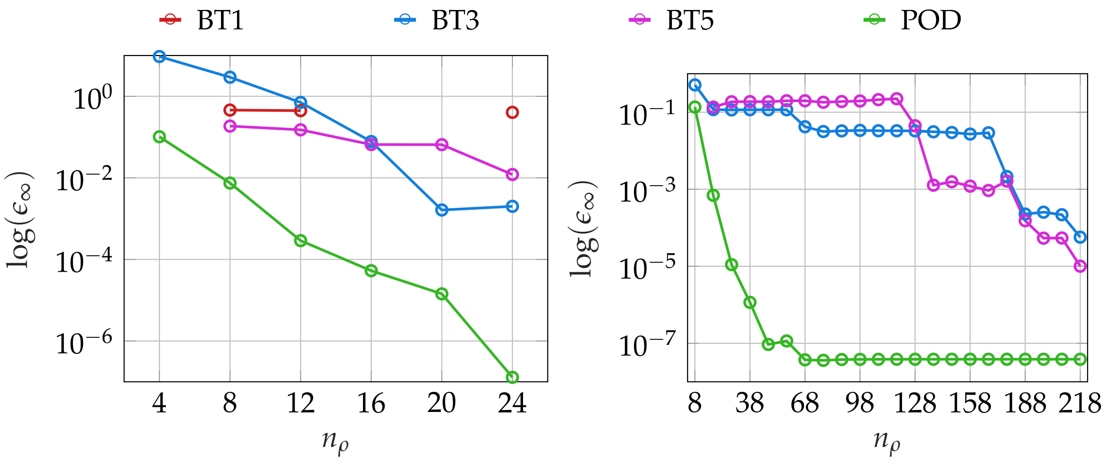
[11] One aspect to point out is that the produced output for $n_\rho = 8$ does not start at $\hat{y}(0) = \pi$ which



(a) EN model

(b) SM model

Figure 5.5: Output of POD for 10s Interval



(a) EN model

(b) SM model

Figure 5.6: L_∞ Output Error for BT1, BT3, BT5, and POD

also for small n_ρ than BT3's.

5.3 Cross-Method-Comparison of Output Error

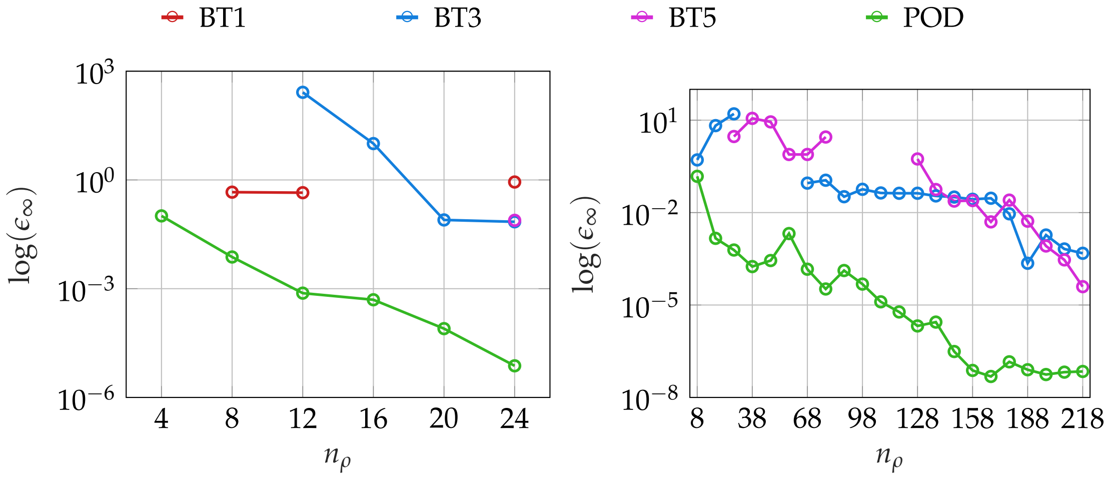
We consider the L_∞ output error as defined in equation (5.1) on a logarithmic scale.

Before we return to the structured analysis, we first use this cross-method comparison to explain the standard choice of BT3 with the help of Figures 5.6 and 5.7. By looking at the plots, $\nu = 3$ is the only Volterra kernel number which produces a measurable L_∞ output error for all tested reduction orders for both CO models in the 2 seconds interval. Estimating the output error fails entirely for the SM ODE system based on the linear BT1, and also the quadratic version with $\nu = 5$ fails for $n_\rho = 8$, see Figure 5.6b. The situation looks similar for the EN model in Figure 5.6a, although the output from BT1 can be measured for $n_\rho \in \{8, 12, 24\}$. The number of noncomputable gaps in the graphs increase for the larger time interval in Figure 5.7. Curiously, the gaps do not only turn up for all n_ρ below a certain threshold, but also appear in between two (mid-range) working reduction orders, e.g, between $n_\rho = 78$ and $n_\rho = 128$ in Figure 5.7b for BT5 in the SM model. In consequence of these findings, we choose the quadratic BT3 method as the most reliable BT option.

EN Model: In Figures 5.6a and 5.7a, we see that BT1 produced measurable errors only for $n_\rho \in \{8, 12, 24\}$. The error is between 10^0 and 10^{-1}, independent of n_ρ and the time interval. Surprisingly, for $n_\rho \in \{8, 12\}$, the linear BT is better than the quadratic BT versions in both time intervals.

While errors could be computed from the application of BT3 in the 2 seconds inter-

pertains both CO models. We tried to remedy this by concatenating the initial state vector with the projection matrix and then performing a QR decomposition to restore orthogonality. As this had no effect, we omitted the procedure afterwards since the value is less than 0.01 off.



(a) EN model

(b) SM model

Figure 5.7: L_∞ Output Error for BT1, BT3, BT5, and POD for 10s Interval

val, it failed for $n_\rho \in \{4, 8\}$ in the longer time frame. Also, except for $n_\rho \in \{20, 24\}$ in the 2 seconds interval, the error is poor with values barely below 10^{-1} or higher. Interestingly, in both time intervals, the error does not vary a lot between $n_\rho = 20$ and $n_\rho = 24$.

Considering BT5, only one error value could be calculated in the 10 seconds interval for the high reduction order $n_\rho = 24$. In the shorter time frame, BT5 was more reliable, but the error curve is shallow, and again, merely for $n_\rho = 24$ an L_∞ error of about 10^{-2} could be computed.

Fixing the reduction order, every BT version produces larger L_∞ errors in the longer time interval.

SM Model: The last statement is also true for the SM model - disregarding that BT1 does not work at all for this model, as we can see in Figure 5.6b and Figure 5.7b. BT3 generates a lower L_∞ error (or at least does not fail) than BT5 for lower reduction orders; its curve meets BT5's first at $n_\rho = 128$ in the 2 seconds interval, and in like manner, between reduction order 138 and 148 in the 10 seconds interval. Whereas BT5 achieves continuously better results than BT3 for $T = [0; 2]$ after their error curves meet, the latter intermediately prevails for $n_\rho \in \{178, 188\}$ in the 10 seconds interval. Staying in this time interval, the error curve of BT5 starts at $n_\rho = 8$ below BT3's and produces output until $n_\rho = 78$. This almost exactly covers the reduction orders for which BT3 fails, i.e., $n_\rho \in \{38, 48, 58\}$.

In the 2 seconds time interval, the L_∞ error based on BT3 noticeably remains almost constant for $18 \leqslant n_\rho \leqslant 58$, again for $68 \leqslant n_\rho \leqslant 168$, and once more for $188 \leqslant n_\rho \leqslant 208$ with errors of about 10^{-1}, just above 10^{-2}, and barely above 10^{-4}, respectively. Regarding BT5, we see output error stagnation at a value of just above 10^{-1} for $18 \leqslant n_\rho \leqslant 118$ in the same interval. BT3's error curve repeats this steadiness for $68 \leqslant n_\rho \leqslant 168$ in the

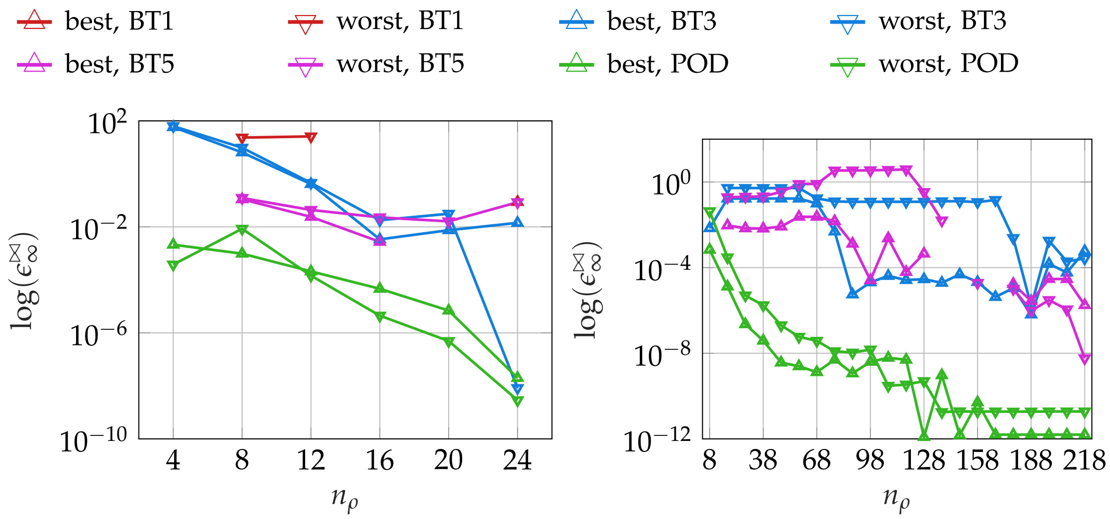
10 seconds time frame with error values a little larger than in the shorter time interval.

EN vs. SM Model: With BT3 and BT5, the reduction accomplishes lower L_∞ errors in the SM model than in the EN model. Even though BT1 worked for some reduction orders in the latter model, the error is so large that the application of this linear version still seems pointless. Taking only the reduction orders into account for which $n_\rho \leqslant n_{\text{min}}^{\text{SM}} = 114$ holds, at least BT3 produces acceptable results in both time frames for the SM model assuming an L_∞ output error of just above 10^{-2} is tolerable.

Output-over-Time vs. Output Error regarding BT3: We interpret the findings of Section 5.2 concerning the output-over-time plots based on the application of BT3 in Figures 5.3 and 5.4 with regard to what we have learned from the L_∞ error plots in Figures 5.6 and 5.7.

Considering the EN model first, the blue L_∞ error curves confirm the diverging behavior for $n_\rho \in \{4, 8, 12\}$ in Figure 5.3a with an L_∞ error above or close to 1 in Figure 5.6a for the 2 seconds time interval. For the two smallest reduction orders regarding the longer time interval, BT3-based ODE solutions failed completely to compute an error $< \infty$ in Figure 5.7a. The outputs for $n_\rho \in \{16, 20, 24\}$ look close to the FOM one and the error curves with values below 10^{-1} for $n_\rho = 16$ and 10^{-3} for the higher reduction orders support this for the 2 seconds time interval. The L_∞ error value of well above 10^1 for $n_\rho = 16$ in the 10 seconds time interval also attests that this reduction order was rightfully omitted in the longer output-over-time plot to prevent distortions. The errors corresponding to $n_\rho \in \{20, 24\}$ in the longer time intervals likewise corroborate the growing deviation of the respective output curves with values approximating 10^{-1}.

The findings for the output-over-time plots match those for the L_∞ error concerning the SM model, too. Considering the 2 seconds time interval in Figure 5.6b, the error for $n_\rho = 8$ borders on 1 and the corresponding output "curve's" straight progression matches this large error in Figure 5.3b. We identified three sets of reduction orders for which the L_∞ output error stagnated in the same time interval. The first set with reduction orders between 18 and 58 result in an L_∞ error of about 10^{-1}. Comparing this with the output-over-time, we see their behavior differs a lot from the FOM curve. For $68 \leqslant n_\rho \leqslant 168$ the error remains almost stationary just above 10^{-2}. This agrees with the output-over-time plot in Figure 5.3b, in which corresponding lines are close together. Reduction order $n_\rho = 68$ produces the aforementioned slightly upwards shifted curve, reflecting the comparatively slightly larger output error. The lines corresponding to the third set of stagnating reduction orders, $188 \leqslant n_\rho \leqslant 208$ can hardly be told apart from the FOM output mirroring the small output error of approximately 10^{-4}. The large error values for the first three reduction orders of above 10^0 and the altogether missing error values in the 10 seconds time frame match the deleted output-over-time lines in Figure 5.4b. The stagnating output error curve corresponding to $68 \leqslant n_\rho \leqslant 168$ agrees with the slowly growing distance between the related output-over-time lines



(a) EN model

(b) SM model

Figure 5.8: L_∞ PTI Error for BT1, BT3, BT5, and POD

and the FOM line. The increase is a little faster for $n_\rho \in \{68, 78, 98\}$, reflecting their slightly larger L_∞ error. A surprise is the absolute minimum error value of about 10^{-4} for $n_\rho = 188$.

In Figure 5.4, we could already see the progressing divergence of the output-over-time lines, especially for the EN model. This indication that reduction by BT3 might be useful for short time intervals, but grows more unreliable with time, is substantiated by the L_∞ output error. In both CO models, the error is either high or not computable for low reduction orders. For mid-range to high reduction orders, we observe the error grows larger in time. Comparing the output error in the 2 seconds interval with the error in the 10 seconds interval for fixed n_ρ value, we notice a greater difference between the errors in the EN than in the SM model.

BT vs. POD: Regardless of CO model and time frame, POD always outperforms the BT approaches. Its green error curves have a much steeper development, especially for the SM model, the method achieves much lower errors, and works for all reduction error in all time intervals. These findings coincide with the findings in Section 5.2.

5.4 Cross-Method-Comparison of Pythagorean Trigonometric Identity Error

In Figure 5.8, we look at the L_∞ PTI errors of the best and worst oscillators as derived in Section 5.1 on Page 67 and the error defined there in equation (5.2). We also examine how this error compares with L_∞ output error of the respective model and MOR approach. Note, that we only consider the standard time interval of $T = [0; 2]$ from now on.

EN Model: First addressing BT1 in Figure 5.8a, determining the L_∞ PTI error was not successful. The worst oscillator offers only two usable errors for the reduction orders of $8, 12$ with a large value of above 10. They are higher than the respective L_∞ output errors in Figure 5.6a. There is also only a single error with a value just below 10^{-1}, at $n_\rho = 24$, belonging to the best oscillator. It is, however, smaller than the corresponding L_∞ output error. BT1 performs worst of the tested MOR methods.

Next considering BT3, this is a good illustration that small differences in values can appear large on a logarithmic scale. We observe that the curves of the best and worst oscillator are nearly identical for the lowest three reduction orders, $n_\rho \in \{4, 8, 12\}$, and within short distance for the next two, $n_\rho \in \{16, 20\}$. That the worst oscillator's error curve does not only steeply decrease in contrast to the best oscillator's curve for $n_\rho = 24$, but it also undercuts the L_∞ output error for the same reduction order. In general, however, the L_∞ PTI error of both curves is worse than the L_∞ output error.

Looking at BT5, its curves of the L_∞ PTI error show a more shallow progression compared to BT3. Nevertheless, the error values are about the same or even lower than BT3's – exempting the missing values for $n_\rho = 4$ (best and worst), and $n_\rho \in \{20, 24\}$ (best only). We observe that BT5's worst and best oscillator error curves are very close for $n_\rho \in \{8, 12, 16\}$ with error values falling from around 10^{-1} to 10^{-3} for the best and from 10^{-1} to 10^{-2} for the worst oscillator. With this reduction order set, the L_∞ PTI errors are smaller than the L_∞ output errors. There are no best values for the two remaining reduction orders, but the L_∞ PTI error values belonging to the worst oscillator remain between 10^{-1} and 10^{-2}, with the one corresponding to $n_\rho = 20$ smaller and the one corresponding to $n_\rho = 24$ larger than the respective L_∞ output error.

SM Model: In Figure 5.8b, we notice that BT1 again does not work at all for this model.

At just about every point, the BT3's worst L_∞ PTI error curve is above the L_∞ output error curve. With the few exceptions of $n_\rho = 8$ and $n_\rho > 188$, this line almost mimics the L_∞ output error curve in Figure 5.6b with a slight upwards shift. We recognize two of the stationary segments, $18 \leqslant n_\rho \leqslant 58$ and $68 \leqslant n_\rho \leqslant 168$, before a rapid decrease in error. Even the minimum at $n_\rho = 188$ exists. With an error value of about 10^{-6}, it is much lower than its output error counterpart. The L_∞ PTI error curve created by the best oscillator remains close to the worst one until $n_\rho = 68$ and then rapidly falls off, before likewise steadying for $98 \leqslant n_\rho \leqslant 158$ with error values between 10^{-5} and 10^{-4}. Following a slow decrease in error, this curve reaches its minimum at $n_\rho = 188$, too. Afterwards, it approximates the line belonging to the worst oscillator.

While BT5 only failed for $n_\rho = 8$ concerning the L_∞ output error in Figure 5.6b, the L_∞ PTI curves have gaps for higher reduction orders. Considering the purple curve of the worst oscillator, its L_∞ PTI errors are much larger than the corresponding L_∞ output errors until the first break in the curve. Yet, after the break, the L_∞ PTI error values are much lower compared to the respective L_∞ output errors. They are also better than the best oscillator's values. Except for the break, the best oscillator's PTI performance, however, offers error values well below those of the related parts of the L_∞ output error curve.

EN vs. SM Model: In parallel to the L_∞ output error, the EN model admits L_∞ PTI error values computed with BT1 in contrast to the SM model, but they are still scarce and large. The reduction by BT3 and BT5 in the SM model results in smaller L_∞ PTI errors for the best oscillators, too, compared to the other CO model. Their best and worst error curves have a greater distance than in the EN model, especially for mid-range reduction orders.

Again, solely allowing for reduction orders with $n_\rho \leqslant n_{\min}^{\mathrm{EN}} = 14$ for the EN model, and $n_\rho \leqslant n_{\min}^{\mathrm{SM}} = 114$ for the SM model, respectively, BT5 alone might be considered to provide a tolerable L_∞ PTI error with values between 10^{-1} and 10^{-2} for both best and worst oscillator. Regarding the SM model, only the best oscillators' error curves have smaller error values than 10^{-1}. BT5's best oscillator curve satisfies this error threshold for all reduction orders $18 \leqslant n_\rho \leqslant 108 \leqslant n_{\min}^{\mathrm{SM}} = 114$, while BT3's error values comply within the range $78 \leqslant n_\rho \leqslant 108 \leqslant n_{\min}^{\mathrm{SM}} = 114$

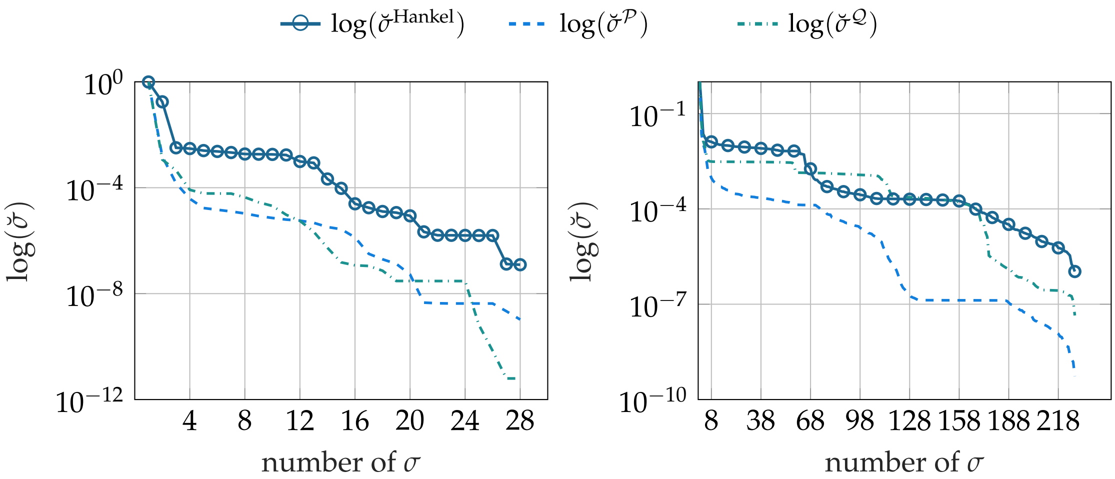
BT vs. POD: Once again, POD outperforms BT. For most of the reduction orders in both CO models, there is a huge distance between the L_∞ PTI errors produced by the MOR reduction and the BT approaches. In addition, all of POD errors are small, even for $n_\rho = 4$ in the EN model, and $n_\rho = 8$ for the SM model, the values are comparatively low.

5.5 Evolution of the Singular Values

In this section, we first look at the BT3-related SVs in Figure 5.9. We then compare the SVs of POD's snapshot matrices for the lifted quadratic system (4.9) and the transformed 1st-order ODE system (4.3) in Figure 5.10. Afterwards, we contrast BT3 and POD.

EN vs. SM Model: As characterized in Definition 3.22 and stipulated in Lemma 3.23, the Hankel SVs and the SVs of the reachability and observability Gramians, $\mathcal{P}$ and $\mathcal{Q}$, should be equal in a (principal-axis) balanced system. Figure 5.9 depicts that this does not hold for both CO models in our case. In contrast, the three SV curves in each plot show a pronounced difference. A common trait is the sharp decrease for the first SVs, and the subsequent preliminary stagnation. This first stationary segment (although on different levels for the different SV types) sets in before our first tested reduction order value $n_\rho = 4$ for EN, and $n_\rho = 8$ for SM model. So, testing even lower orders would probably not have been successful.

Focusing on the Hankel SV curve, we observe that the first stagnation ends at around the 12th Hankel SV in the EN model, and shortly before the 68th Hankel SV in the SM



(a) EN model

(b) SM model

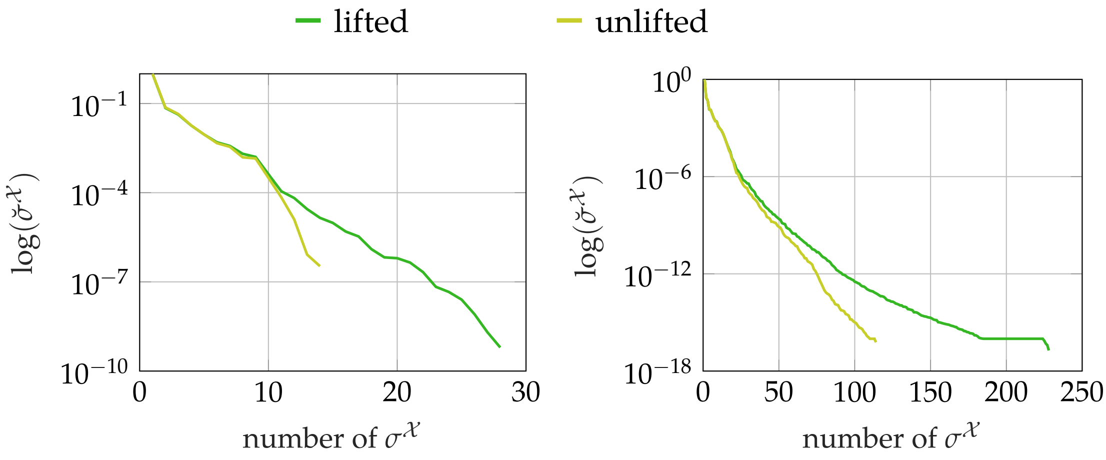
Figure 5.9: Hankel SVs, and SVs of Gramians $\mathcal{P}$, $\mathcal{Q}$

model. The former CO model has two additional stationary phases between the 16th and 20th, and the 21st and 26th Hankel SV. The SM model has only one additional stagnant segment starting around the 68th and ending at circa the 158th Hankel SV. The remainder of the Hankel SV curve slowly decreases until Hankel SV 218, and then falls of sharply. If we highlighted the Hankel SV numbers which are also in our set of tested reduction orders, the evolution of the Hankel SVs mimics the progress of the L_∞ output error in Figure 5.6 for both CO models. Especially, the L_∞ output error stagnation intervals for the SM model are retraceable.

Taking also the SVs of the Gramians into consideration, we see that for the most part, the Hankel SVs are larger than or equal to the SVs of the Gramians and they do not decrease as much. Whereas the SV curves of the Gramians cross several times in the EN model, the line belonging to the observability Gramian is always above the reachability one in the SM model. In general, the SVs in the EN model are smaller compared to the other model.

Lifted vs. Unlifted ODE System: Figure 5.10 depicts the decline of the SVs of the snapshot matrices for the lifted quadratic ODE system (4.9) and the unlifted 1st-order nonlinear ODE system (4.3). The two respective curves in the EN and SM model plots fall off fast, even more so for first SVs in the latter model. The lines are actually aligned for the prime 10 SVs, and the 25 SVs, respectively. After that, the curves belonging to the unlifted system accelerates its decrease. Nonetheless, the SVs belonging to the lifted quadratic system decay rapidly, too. This implies that the projection matrices computed from the SVD of the unlifted and lifted snapshot matrix, respectively, span the same or at least a very similar low-dimensional subspace in which the state vector can be approximated. So, POD does not suffer from the system lifting.

We can also spot the parallel progression of the quadratic system's snapshot SV



(a) EN model

(b) SM model

Figure 5.10: Singular Values of POD Snapshot Matrices for Lifted Quadratic and Unlifted Nonlinear System

curve and the L_∞ output error as well as the L_∞ PTI error curves in Figure 5.6 and Figure 5.8, respectively.

BT3 vs. POD: Focusing on the $\mathcal{X}$-SVs of the quadratic system, we see that their descent is much more rapid and lower compared to the fall-off of the BT3-related SVs. This explains the better performance of POD across reduction orders, CO models, and time frames. Especially juxtaposing the Hankel SVs with the $\mathcal{X}$-SVs illustrates the former's comparatively shallow decrease

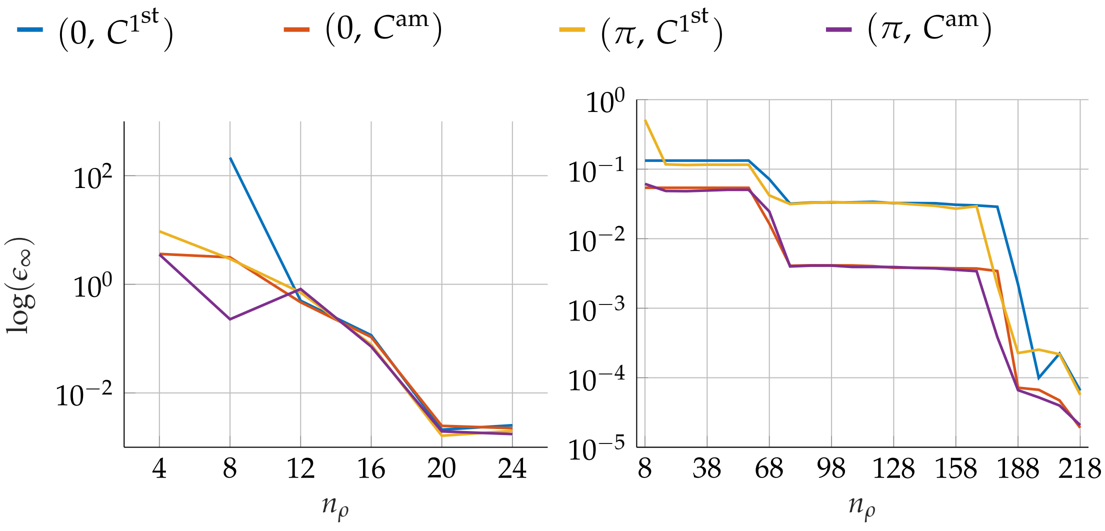
5.6 Influence of Initial Value & Output Matrix on Output Error

In Figure 5.11, we inspect the L_∞ output error for four combinations, i.e., four pairs (δ_0, C) of two initial values and two output matrices as mentioned in Section 5.1 and summarized in Table 5.1 on Page 65.

EN Model: For $n_\rho = 4$, the BT reduction does not work at all for the pair $(0, C^{1^{\text{st}}})$, but the L_∞ output errors for the other combinations are comparatively close. The pair $(0, C^{1^{\text{st}}})$ negatively diverges, and the pair $(0, C^{1^{\text{st}}})$ starts with a large error value of 10^2 at $n_\rho = 8$, while having a very similar (large) error. The error curves of all pairs are almost aligned for reduction orders $n_\rho \geqslant 12$.

SM Model: Regarding the SM model, two features stand out: First, both pairs including output matrix C^{am} almost always perform better than the other two. Second, there is almost no improvement in the output error in the reduction order sets we identified in Section 5.3 for all combinations.

EN vs. SM Model: Although the curves develop a little erratically for $n_\rho < 12$ in the EN model, and the two curves relating to the $(\delta_0, C^{\text{am}})$ pairs have a negative shift, the L_∞ output error behavior is very similar in both CO models in general. Considering



(a) EN model

(b) SM model

Figure 5.11: L_∞ Output Error for (δ_0, C) pairs of Initial Values and Output Matrices

our previous findings, this supports the assertion that the evolution of the Hankel SVs is more important to the BT reduction result than the decision on a particular initial value or output matrix.

We chose the pair $(\pi, C^{1^{st}})$ as standard, because the output of one oscillator is easier to interprete than the average of all, and we wanted an initial value that is also nonzero for the 1st-order ODE system.

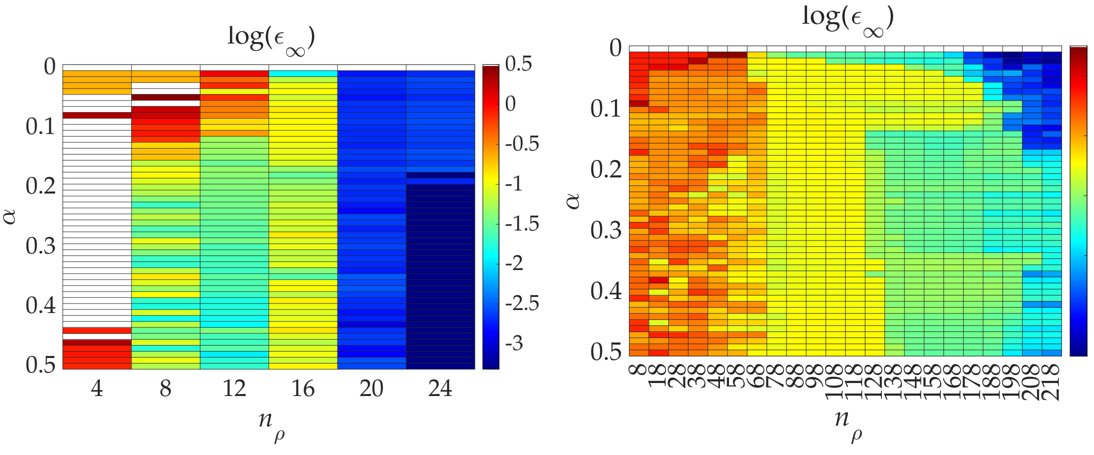
5.7 Influence of α-Shift on Output Error

In this section, we examine to what extent the choice of the α-shift to make matrix $\widetilde{A}$ stable affects the L_∞ output error performance of BT. To that end, we examine the heatmaps in Figure 5.12, depicting the error value on a color scale for α values between 0 and 0.5 with a step size of 0.01 and the usual reduction orders.

EN Model: For $\alpha = 0$, BT3 itself does not work. Without a shift, $\widetilde{A}$ violated the requirement for a stable matrix in Definition 3.14, and therefore the assumption for a stable system in Theorem 3.15, that all its EVs must be in the left half of the complex plane. In consequence, neither the Lyapunov-Cholesky routine in Algorithm 3.1 nor the Gramians in Algorithm 3.2 and Algorithm C.1 could be computed.

This was not the reason for the other white cells in the heatmap for reduction orders $n_\rho = 8$ and especially $n_\rho = 4$. Here, the error computation resulted in NaN.

An evident finding is that for high reduction orders $n_\rho \in \{20, 24\}$, all tested α values resulted in output errors of $10^{-2.5}$ or smaller. In contrast, discovering a pattern between α value and output error for the other reduction orders is more difficult. There is no n_ρ for which the output error progresses downwards on the color scale along with in-



(a) EN model

(b) SM model

Figure 5.12: Impact of Different A_α on L_∞ Output Error

creasing alpha, or vice versa. There are also merely a few α values for which the output error improves with increasing the reduction order. Order $n_\rho = 16$ is especially interesting, because it is more sensitive to the shift values than reduction orders $n_\rho = 8$ and $n_\rho = 12$. It results in worse L_∞ output errors for many of the α values compared to the lower reduction orders. It actually works best with $\alpha = 0.01$, a value which produces the worst error for $n_\rho = 12$. Disregarding the two highest and the lowest reduction orders $n_\rho \in \{4, 20, 24\}$, $n_\rho = 12$ shows the most coherent behavior, moving from red to light blue with increasing α. But here, too, there are output error values disturbing the color flow, e.g., for $\alpha = 0.05$. These anomalies occur for all reduction orders with different frequency, quantity, and significance.

SM Model: Shifting $\widetilde{A}$ is likewise absolutely necessary to perform the balancing part of BT3 concerning this model, for the same reason and reasoning as above. In contrast to the other CO model, L_∞ output errors can be computed for all other shift and reduction order values.

The error progression appears a little more comprehensible. While for this CO model, too, color distribution follows no stringent pattern, moving from left to right along reduction orders, there is a general tendency to encounter lower output errors. Going from top to bottom along α values, however, it is harder to spot a pattern. For reduction orders $n_\rho \geqslant 68$, very low α shifts work better than larger ones, especially for $n_\rho \geqslant 178$, where the output error seems to be inversely proportional to the shift value. It looks as if the exact shift value does not significantly affect the output error for $78 \leqslant n_\rho \leqslant 198$ after passing a certain α threshold. In between $128 \leqslant n_\rho \leqslant 198$, there is a second threshold as we can see a yellow area squeezed in between very small and mid-range α values.

For lower reduction orders $n_\rho \leqslant 58$, the L_∞ output error is for the most part worse, and the error pattern is also more inconsistent than for higher orders. Both findings can be inferred from the color distribution.

EN vs. SM Model: Both models need a minimum α shift to make BT work. They also have in common that for a number of reduction orders, very small shifts produce better results than larger ones. An interpretation is that, depending on the system size and the reduction order, on the one hand, α must be large enough to make $\widetilde{A}$ stable, and on the other hand, it must not be too large such that the matrix is not shifted too far from the original.

However, this does not explain the unclear relation between the value of α and the L_∞ output error, which is more pronounced in the EN model. To make out some pattern, we have to factor in the reduction order. This indicates that the success of BT3 depends more on the reduction order than on the shift.

Due to this arbitrary behavior of the shift, we picked the standard value of $\alpha = 0.05$ for the numerical experiments, reasoning that it was large enough to make $\widetilde{A}$ stable, but small enough to keep it close to the original.

CHAPTER 6

Summary and Conclusion

The main contribution of this thesis is the application and examination of an only recently proposed MOR method – Balanced Truncation adapted from the linear to the quadratic(-bilinear) case – to a problem of practical, real-world importance – the solution of a nonlinear ODE system which models PN dynamics.

We used quadratic lifting of the system, an approach lacking research especially in the context of MOR. We contributed implementations of the PN ODE models given the model-dependent parameters, such that they can be solved via MATLAB. This includes the transformed lifted quadratic matrix ODE representation to which quadratic BT (and POD) can be applied. We derived and implemented the adapted quadratic version of BT. Its performance was examined by testing the influence of specific, selectable reduction parameters, like the reduction order or the stabilizing shift, and pitched it against the well-established POD method.

From these numerical experiments, we draw these main conclusions and propose some conjectures from our analysis of the CO model MOR by BT:

The application of BT1, i.e., linear BT, to a quadratic system does not give good approximations. While this is probably foreseeable, it might come as a surprise that BT5 overall does not perform as good and is not as reliable as BT3, especially for lower reduction orders. Considering that the computation of the Volterra kernels is very costly, it is valuable to know that adding more kernels does not necessarily help the performance.

The output from the reduced solutions grows unstable for longer time intervals; the magnitude of divergence seems to depend on the underlying CO model or the system size. The output divergence and error is greater for the smaller EN model than for the SM model in any case. Since some of the necessary assumptions to derive the CO models' parameters are only valid for short time periods, anyway, this might not be an issue in the reduction of power networks.

In general, the reduction worked better for the latter CO model. This raises the question whether BT is better suited for larger systems, or whether the CO model representation is the crucial factor. It would be interesting to represent other power flow cases as EN and SM models in quadratic structure, and study if the reduction by BT reproduces the findings in a cross-CO-model-test-case comparison. The implemented functions and scripts are ready to test different, small power flow cases. The code to determine the Gramians could also be optimized to study larger systems, e.g., by parallelization of the bottle-neck Kronecker product, or better memory allocation.

Experiments with varying initial value, output matrix, and shift indicate that the reduction performance depends more on the reduction order and Hankel SV evolution than on the specific values of x_0, C and α. At all events, the decrease of the Hankel SVs provides the most reliable measure to predict how successful the BT reduction will be. The comparison of output error and Hankel SVs also reveals the link between these SVs and the reduction order.

Overall, the reduction performance of BT is not very satisfactory. This is especially evident whenever we measure it against POD. However, POD has the disadvantage of depending on a fixed input and the snapshot instances. So, BT might have its advantages here, because once the projection matrices are computed, we can just reuse them. Testing modified static or time-dependent inputs would give some insights.

In conclusion, it is too soon for a final verdict on quadratic Balanced Truncation. This thesis provides some preliminary results and lays the ground for further experiments.

APPENDIX A

More on Gramians, Stability and Energy Functionals for Balanced Truncation

A.1 Linear Reachability and Observability Concepts

We consider the system $\Sigma = \left(\begin{array}{c|c} A & B \\ \hline & \end{array}\right)$ with $A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times m}$.

Theorem A.1 (**Reachability properties**, [Ant05]).

(1) *The reachability space* $\mathbb{X}^{\text{reach}} \subseteq \mathbb{X}$ *is a linear space. It holds*

$$\mathbb{X}^{\text{reach}} = \operatorname{im}(R(A, B)).$$

(2) $A\mathbb{X}^{\text{reach}} \subseteq \mathbb{X}^{\text{reach}}$, *i.e., the reachable subspace is A-invariant.*

(3) Σ *is (completely) reachable if and only if* $\operatorname{rank}(R(A, B)) = n$.

(4) *Reachability is independent of the system basis.*

Proposition A.2 (**Relation of Reachability Matrix and Gramian**, [Ant05]). *For all times* $t > 0$, *the column space of the reachability Gramian generates the whole reachability subspace, i.e.,*

$$\operatorname{im}(\mathcal{P}(t)) = \operatorname{im}(R(A, B)).$$

With respect to the observability concept, we consider the system $\Sigma = \left(\begin{array}{c|c} A & \\ \hline C & \end{array}\right)$ with $A \in \mathbb{R}^{n \times n}, C \in \mathbb{R}^{p \times n}$.

Theorem A.3 (**Observability properties**, [Ant05]).

(1) *The unobservability subspace* $\mathbb{X}^{\text{unobs}} \subseteq \mathbb{X}$ *is a linear space. It holds*

$$\mathbb{X}^{\text{unobs}} = \ker(O(C, A)) = \{x \in \mathbb{X} \mid CA^{i-1}x = 0, i > 0\}.$$

(2) $A\mathbb{X}^{\text{unobs}} \subseteq \mathbb{X}^{\text{unobs}}$, *i.e., the unobservable subspace is A-invariant.*

(3) Σ *is (completely) observable if and only if* $\operatorname{rank}(O(C, A)) = n$.

(4) *Observability is independent of the system basis.*

Proposition A.4 (**Relation of Observability Matrix and Gramian**, [Ant05]). *For all times $t > 0$, it holds*

$$\ker(\mathcal{Q}(t)) = \ker(O(C, A)). \tag{A.1}$$

Employing the duality principle in linear systems, we can show a link between the reachable subspace of a system Σ and the unobservable subspace of its dual system Σ^*. To that end, we now consider the system (3.3), $\Sigma = \left(\begin{array}{c|c} A & B \\ \hline C & \end{array}\right)$, and define its *dual system* as

$$\Sigma^* = \left(\begin{array}{c|c} -A^* & -C^* \\ \hline B^* & \end{array}\right) \in \mathbb{R}^{(n+m)\times(n+p)},$$

with A^*, B^* and C^* being the respective dual maps of A, B and C. Their corresponding matrix representations are the complex transposed matrices A, B and C. The dual system switches the parts of input and output variables, i.e., $-C^*$ determines the input, B^* the output and $-A^*$ the dynamics of the system. The dual system Σ^* is also the adjoint of Σ with respect to the usual inner product $\mathcal{L}_2(\mathbb{R})$.

Theorem A.5 (**Duality Principle**, [Ant05]). *It holds:*

$$(\mathbb{X}_{\Sigma}^{\text{reach}})^{\perp} = \mathbb{X}_{\Sigma^*}^{\text{unobs}}.$$

The system Σ is reachable if and only if its dual Σ^* is observable.

A.2 Stability

In this section, we consider the autonomous system

$$\dot{x}(t) = Ax(t). \tag{A.2}$$

We present additional definitions and properties concerning stability in order to provide further information on energy functionals below in Appendices A.3 and A.4.

Definition A.6 (**Equilibrium point**, [Lue79]). We say $\bar{x}$ is an *equilibrium point* of system (A.2) if $A\bar{x}(t) = 0$ for all $t \geqslant t_0$. In other words, after the state vector has reached the point $\bar{x}$, it will stay in that position for all time.

Definition A.7 (**(Asymptotically) Stable Equilibrium Point**, [Lue79]). Let $\mathcal{K}_{\bar{x},R}$ be the ball of radius R with equilibrium point $\bar{x}$ at its center.

(i) An equilibrium point $\bar{x}$ is called *stable* if for every $R > 0$ there is an $0 < r < R$ such that if $x(0) \in \mathcal{K}_{\bar{x},r}$ then $x(t) \in \mathcal{K}_{\bar{x},R}$ for all $t > 0$.

(ii) An equilibrium point $\bar{x}$ is called *asymptotically stable* if in addition to being stable there is an $\bar{R} > 0$ such that if $x(0) \in \mathcal{K}_{\bar{x},\bar{R}}$ then $x(t) \to \bar{x}$ for $t \to \infty$.

Aside from the matrix-based system stability in Theorem 3.15, there is also the so-called *Lyapunov-based stability*, based on a special kind of scalar functions:

Definition A.8 (**Lyapunov Functions**, [BG17; Lue79])**.** A *Lyapunov function* over a ball $\mathcal{K}_{0,r}$ is a continuous function $\mathcal{F} : \mathbb{R}^n \to \mathbb{R}$ with continuous first partial derivatives if it holds

(i) $\mathcal{F}(0) = 0$ and $\mathcal{F}(x) > 0$ for $x \in \mathcal{K}_{0,r} \setminus \{0\}$,

(ii) $\frac{d}{dt}\mathcal{F}(x) \leqslant 0$ for all $x \in \mathcal{K}_{0,r}$.

The first condition attests that, within $\mathcal{K}_{0,r}$, the function $\mathcal{F}(x)$ is positive definite and has a unique minimum at $x = 0$; the second condition means that the gradient of the function is negative semidefinite and ensures that the function values are nonincreasing along all state trajectories within the ball.

Theorem A.9 (**Lyapunov Function Stability**, [BG17; Lue79])**.** *If there exists a Lyapunov function over a ball $\mathcal{K}_{0,r}$, then $\bar{x} = 0$ is a stable equilibrium point. If the gradient of the Lyapunov function is negative definite for every state vector in $\mathcal{K}_{0,r} \setminus \{0\}$, then $\bar{x} = 0$ is an asymptotically stable equilibrium point.*

Usually, it is challenging to construct a Lyapunov function, especially for general nonlinear systems [Lue79], but under linear, autonomous conditions as in system (A.2), we have the well-established quadratic function at our disposal [Ant05]:

$$\mathcal{F}(x) = x^* P x, \quad P = P^* \in \mathbb{R}^{n \times n},$$

and differentiate the equation to see which properties the function terms must have to meet the conditions of Definition A.8:

$$\begin{aligned}
\frac{d}{dt}\mathcal{F}(x) &= \dot{x}^* P x + x^* P \dot{x} \\
&= x^* A^* P x + x^* P A x \\
&= x^* \underbrace{(A^* P + PA)}_{=Z} x \\
&= x^* Z x.
\end{aligned}$$

So, to find a suitable P, we need to solve the quadratic Lyapunov function

$$A^* P + PA = Z. \tag{A.3}$$

The following theorem establishes the necessary properties of A and Z to ensure the existence of a unique solution P and the desired system stability behavior.

Theorem A.10 (**Lyapunov Equation Stability for Linear Systems**, [Ant05])**.**

(1) *If A is stable, then there exists a unique solution $P \in \mathbb{R}^{n \times n}$ of* (A.3) *for all $Z \in \mathbb{R}^{n \times n}$.*

(2) *If $Z = Z^*$, then also $P = P^*$.*

(3) *If Z is negative semidefinite and the pair (A, Z) is observable, then P is positive definite.*

(4) *If Z is negative semidefinite and P is positive definite and in addition Z and P satisfy* (A.3), *then the system* (A.2) *is stable.*

(5) *If* (4) *holds and furthermore* (A, Z) *is observable, then the system* (A.2) *is asymptotically stable.*

Considering quadratic systems, the previously given Definitions A.6 and A.8 remain valid except that we replace the linear autonomous system (A.2) with the quadratic autonomous system

$$\dot{x}(t) = Ax(t) + H\big(x(t) \otimes x(t)\big), \tag{A.4}$$

where $\bar{x} = 0$ is a stable equilibrium point [BG17].

The Lyapunov method of determining stability is so powerful that we can apply it to the nonlinear case as well. Even though the analysis of equilibrium points and the construction of Lyapunov functions is more difficult for nonlinear systems, [BG17] provide the following results (adapted by us for the quadratic case):

Theorem A.11 (**Lyapunov Function Stability for Quadratic Systems**, [BG17]). *Consider the quadratic system* (3.38) *and let A be stable. Furthermore, let the reduced-order system* (3.40) *be obtained by performing Algorithm 3.3, where low-rank approximations of the truncated Gramians are determined by an iterative scheme. Let P_1 and Q_1 be the solutions to their respective linear Lyapunov equations* (3.46) *and* (3.52). *We define*

$$r = \frac{\sigma_{min(\mathcal{V}^*\mathcal{G}\mathcal{V})}}{2||\Sigma_{n_\rho}||||\hat{H}||} \quad \text{and} \quad \mathcal{G} = \mathcal{H}^{(2)}(P_1 \otimes Q_1)(\mathcal{H}^{(2)})^* + C^*C.$$

Then it holds for the Lyapunov function $\mathcal{F}(\hat{x}) = \hat{x}^*\hat{\Sigma}\hat{x}$ that

$$\mathcal{F}(\hat{x}) > 0 \quad \text{and} \quad \frac{d}{dt}\mathcal{F}(\hat{x}) < 0, \quad \text{for all } \hat{x} \in \mathcal{K}_{0,r} \setminus \{0\}, \tag{A.5}$$

i.e., $\bar{x} = 0$ is an asymptotically stable equilibrium point.

A.3 Energy Functionals for Linear Balanced Truncation

In this section, we consider a stable system $\Sigma = \left(\begin{array}{c|c} A & B \\ \hline C & \end{array}\right)$ with $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}$ and $C \in \mathbb{R}^{p\times n}$.

Proposition A.12 (**Computation of Energy Functionals with finite Gramians**, [Ant05]). *Let the system Σ be reachable and observable and let the finite Gramians $\mathcal{P}(t), \mathcal{Q}(t)$ be as defined in 3.7 and 3.12, respectively. Also, let $x = \int_0^t e^{A^*(t-\tau)}Bu\,d\tau$ and $v \in$* im$(\mathcal{P}(t))$ *(see Theorem A.2). Then,*

$$||u||^2 = x^*\mathcal{P}^{-1}(t)x, \tag{A.6}$$

where $u(t) = -B^*e^{A^*(t-\tau)}v$ is the minimal input energy necessary to move the system to the state x. Furthermore,

$$||y||^2 = x_0^*\mathcal{Q}(t)x_0 \tag{A.7}$$

is the output energy generated by the initial state x_0 at time t.

From the definition of the Gramians, it is clear that

$$\mathcal{P}(t_2) \geqslant \mathcal{P}(t_1) \quad \text{and} \quad \mathcal{Q}(t_2) \geqslant \mathcal{Q}(t_1) \quad \text{for} \quad t_2 \geqslant t_1,$$

i.e., the finite Gramians grow larger in time [Ant05]. This also means, using them to determine the energy functionals is time-dependent.

Remark A.13. The existence, or respectively finiteness, of the energy functionals in Definition 3.17 is not always clear. If the system is not stable, L_o can be infinite. By convention, $L_c(x_0)$ is infinite if x_0 cannot be reached from 0. We therefore assume that both energy functionals are finite from now on [Sch93].

A.4 Energy Functionals for Quadratic Balanced Truncation

We present theorems which are based on the relaxed Definition 3.27 of the observability energy functional.

Theorem A.14 (**Computation of Energy Functionals of General Nonlinear Systems**, [BG17; GM96])**.** *Consider the nonlinear version of system* (3.1) *with* $\bar{x} = 0$ *as an asymptotically stable equilibrium in two neighborhoods* $\mathcal{K}_o$ *and* $\mathcal{K}_c$ *of* 0*. Assume the system is controllable and observable, and* L_o *and* L_c *are smooth functions of* x*.*

(1) *For all* $x \in \mathcal{K}_c$*, the controllability energy functional* $L_c(x)$ *can be uniquely determined by the partial differential equation*

$$\frac{\partial L_c}{\partial x} f(x) + f(x) \frac{\partial L_c}{\partial x} + \frac{\partial L_c}{\partial x} g(x) g^*(x) \frac{\partial^* L_c}{\partial x} = 0, \quad L_o(0) = 0, \tag{A.8}$$

under the assumption there exists a smooth solution $\bar{L}_c$ *on* $\mathcal{K}_c$ *and* 0 *is an asymptotically stable equilibrium of* $-(f(x) + g(x)g^*(x)\frac{\partial^* L_c}{\partial x})$ *on* $\mathcal{K}_c$*.*

(2) *Likewise, for all* $x \in \mathcal{K}_o$*, the observability energy functional* $L_o(x)$ *can be uniquely determined by the partial differential equation*

$$\frac{\partial L_o}{\partial x} f(x) + \frac{1}{2} h^*(x) h(x) - \frac{1}{2} \mu^{-1} \frac{\partial L_o}{\partial x} g(x) g^*(x) \frac{\partial^* L_o}{\partial x} = 0, \quad L_o(0) = -\frac{1}{2}\mu, \tag{A.9}$$

under the assumption there exists a smooth solution $\bar{L}_o$ *on* $\mathcal{K}_o$ *and* 0 *is an asymptotically stable equilibrium of* $\bar{f}(x) := (f(x) - \mu^{-1} g(x) g^*(x) \frac{\partial^* L_o}{\partial x})$ *on* $\mathcal{K}_o$ *with* $\dot{\psi} = \bar{f}(\psi)$, $\psi(0) = x$ *and* $\mu := -||g^*(\psi) \frac{\partial^* \bar{L}_o}{\partial x}(\psi)||_{\mathcal{L}_2} \in \mathbb{R}$.

For the rest of this section, we look at the quadratic system $\Sigma = \left(\begin{array}{c|c|c} A & H & B \\ \hline C & & \end{array}\right)$, with matrices $A \in \mathbb{R}^{n \times n}, H \in \mathbb{R}^{n \times n^2}, B \in \mathbb{R}^{n \times m}$ and $C \in \mathbb{R}^{p \times n}$.

With the next theorem, [BG17] contribute bounds on the energy functionals using the quadratic form of the constructed infinite Gramians. They derive the bounds computed with truncated Gramians in Theorem 3.28 from it.

Theorem A.15 (**Energy Functional Bounds with infinite Gramians**, [BG17])**.**

(1) *Let the system be controllable and A be stable. Furthermore, let the reachability Gramian $\mathcal{P} > 0$ uniquely satisfy the generalized Lyapunov equation* (3.50)*. Also, using Theorem A.14, let the controllability energy functional $L_c(x)$ be the solution of*

$$\frac{\partial L_c}{\partial x}(Ax + H(x \otimes x)) + (Ax + H(x \otimes x))^* \frac{\partial^* L_c}{\partial x} + \frac{\partial L_c}{\partial x} BB^* \frac{\partial^* L_c}{\partial x} = 0. \tag{A.10}$$

Then, there is a neighborhood $\mathcal{K}_c$ of 0*, such that for $x \in \mathcal{K}_c$*

$$L_c(x) \geqslant \frac{1}{2} x^* \mathcal{P}^{-1} x. \tag{A.11}$$

(2) *Let $B \equiv 0$ and x_0 be an initial condition. Let the reachability Gramian $\mathcal{P} > 0$ and the observability Gramian $\mathcal{Q} \geqslant 0$ uniquely satisfy their respective generalized Lyapunov equation* (3.50) *or* (3.56)*. Let $L_o(x)$ be the observability energy functional as defined in Definition 3.27. Then, there is a neighborhood $\mathcal{K}_o$ of* 0*, such that for $x \in \mathcal{K}_o$*

$$L_o(x) \leqslant \frac{1}{2} x^* \mathcal{Q} x. \tag{A.12}$$

It follows, that the proposed Gramians provide local estimates of the energy functionals near the origin.

A strong assumption in Theorems A.15 and 3.28 is the positive definiteness of $\mathcal{P}$. Since this condition often cannot be guaranteed, [BG17] offer a direct connection of the infinite and truncated Gramians to the reachability and observability of the system. In consequence, this ties the proposed Gramians to the classical idea of balanced truncation.

Theorem A.16 (**Reachability and Observability with infinite Gramians**, [BG17])**.** *Consider the system* (3.37)*.*

(1) *Let the reachability Gramian $\mathcal{P} > 0$ satisfy the generalized Lyapunov equation* (3.50)*. Let the system be driven from the zero state to x_0. If $x_0 \notin \operatorname{im}(\mathcal{P})$, then $L_c(x) = \infty$ for all inputs u.*

(2) *Let $B \equiv 0$ and $x(0) = x_0$ be an initial condition. Let the reachability Gramian $\mathcal{P} > 0$ and the observability Gramian $\mathcal{Q} \geqslant 0$ satisfy their respective generalized Lyapunov equation* (3.50) *or* (3.56)*. If $x_0 \in \ker(\mathcal{Q})$, then the observability energy functional $L_o(x_0) = 0$.*

It follows from the first part of the theorem, that a state variable x_0 which is not in the image of $\mathcal{P}$ cannot be reached from 0. Conversely, all the states which lie in $\ker(\mathcal{P})$ are uncontrollable. Likewise, the second part asserts that after the elimination of the uncontrollable states (by assuming positive definiteness of $\mathcal{P}$), the states which are contained in $\ker(\mathcal{Q})$ are unobservable. Accordingly, states that are in one of the Gramians' kernels are judged unimportant regarding the system's dynamics. This coupled with the bounds given in Theorem A.15 correspond with the idea of linear balanced truncation of removing those states with low degrees of reachability and observability, again measured by the SVs of the proposed Gramians.

Corollary A.17 (**Reachability and Observability with truncated Gramians**, [BG17]). *Consider the system* (3.37).

(1) *Let the truncated reachability Gramian $\mathcal{P}_{\mathcal{T}}$ and the truncated observability Gramian $\mathcal{Q}_{\mathcal{T}}$ satisfy the respective Lyapunov equation* (3.48) *and* (3.54). *Let the system be driven from the zero state to $x_0 \notin \mathrm{im}(\mathcal{P}_{\mathcal{T}})$. Then, $L_c(x) = \infty$ for all inputs u.*

(2) *Let the quadratic system be locally controllable around* 0*, and assume that the pair (A, B) is controllable. Then for $B \equiv 0$ and $x(0) = x_0$, it holds that if $x_0 \in \ker(\mathcal{Q}_{\mathcal{T}})$, then $L_o(x_0) = 0$.*

APPENDIX B

Multi-Linear Algebra Operations

B.1 The Kronecker Product

Let $\mathbb{K}$ be either $\mathbb{R}$ or $\mathbb{C}$. For a more in depth treatment of the Kronecker product, see [Pol14; ZD13].

Definition B.1 (**Kronecker Product**, [ZD13]). Let $M = [m_{ij}] \in \mathbb{K}^{k\times\ell}$ and $N \in \mathbb{K}^{m\times n}$ be matrices. The *Kronecker product* of M and N is the $(km \times \ell n)$ matrix:

$$\begin{aligned} M \otimes N &= \begin{bmatrix} m_{ij}N \end{bmatrix} \\ &= \begin{bmatrix} m_{1,1}N & m_{1,2}N & \cdots & m_{1,\ell}N \\ m_{2,1}N & m_{2,2}N & \cdots & m_{2,\ell}N \\ \vdots & \vdots & \ddots & \vdots \\ m_{k,1}N & m_{k,2}N & \cdots & m_{k,\ell}N \end{bmatrix}. \end{aligned}$$

Theorem B.2 (**Kronecker Product Properties**, [ZD13]). *Let $M \in \mathbb{K}^{k\times\ell}$, $K \in \mathbb{K}^{\ell\times p}$ and $N \in \mathbb{K}^{m\times n}$, $L \in \mathbb{K}^{n\times r}$ be matrices, and μ a scalar. The following properties hold for the Kronecker product:*

(1) *In general, noncommutativity:*

$$M \otimes N \neq N \otimes M.$$

(2) *Linearity in both terms:*

$$M \otimes (\mu N) = \mu(M \otimes N) = (\mu M) \otimes N.$$

(3) *Distributivity and associativity:*

$$\begin{aligned} (M + N) \otimes K &= (M \otimes K) + (N \otimes K), \\ M \otimes (N + K) &= (M \otimes N) + (M \otimes K), \\ (M \otimes N) \otimes K &= M \otimes (N \otimes K). \end{aligned}$$

(4) *Distributivity of (conjugate) transposition:*

$$(M \otimes N)^* = M^* \otimes N^*.$$

(5) *Mixed product with appropriately sized arbitrary matrices:*

$$(M \otimes N)(K \otimes L) = MK \otimes NL.$$

(6) *Mixed product with appropriately sized identity matrix:*

$$(M \otimes N) = (M \otimes \mathbb{I}_m)(\mathbb{I}_\ell \otimes N) = (\mathbb{I}_k \otimes N)(M \otimes \mathbb{I}_n).$$

Observe that $(M \otimes \mathbb{I}_n)(\mathbb{I}_\ell \otimes N)$ *commutes for square matrices* $M \in \mathbb{K}^{\ell \times \ell}$ *and* $N \in \mathbb{K}^{n \times n}$.

B.2 Tensor Unfolding

For the computation of the observability Gramian as in Definition 3.33, and by extension its Volterra kernels as in Definition 3.32, the Lyapunov equations in (3.53), (3.54) and in Theorem 3.34 must be solved. This requires the matrix $\mathcal{H}^{(2)}$. We introduced this matrix as the mode-2 unfolding of a tensor created from H in the Section 3.3. Now, we give the mathematical construction of this matrix. Since we only provide a brief, informal treatment of tensor theory and restrict ourselves to the necessities of our case, we refer to [KB09] for a detailed overview on the subject.

A *tensor* is an array of multiple dimensions. Their number determines its *order* or *mode*. So far in this thesis, we have encountered 1- and 2-order tensors, i.e., vectors and matrices [KB09]. To construct $\mathcal{H}^{(2)}$, we must briefly enter the 3rd dimension. To that end, let us specifically consider the 3rd-order tensor $\mathcal{H}$ with elements η_{ijk}. To get a one-dimensional *fiber* of $\mathcal{H}$, we fix every index but one. Accordingly, $\mathcal{H}$ has three different kinds of fibers: the mode-1 fibers $\eta_{:jk}$ (the columns), the mode-2 fiber $\eta_{i:k}$ (the rows), and the mode-3 fibers $\eta_{ij:}$ (the tubes). Tensor *slices* are two-dimensional, obtained by fixing every index but two. That leaves only one free index for $\mathcal{H}$ with which we can access its horizontal ($\mathcal{H}_{i::}$), lateral ($\mathcal{H}_{:j:}$) and frontal ($\mathcal{H}_{::k}$) slices. The *mode-m unfolding* of a tensor repositions its mode-m fibers to become the columns of a matrix [KB09]. It is denoted by $\mathcal{H}^{(m)}$. For example, let $\mathcal{H} \in \mathbb{R}^{4\times3\times2}$ with slices

$$\mathcal{H}_{::1} = \begin{bmatrix} \eta_{1,1,1} & \eta_{1,2,1} & \eta_{1,3,1} \\ \eta_{2,1,1} & \eta_{2,2,1} & \eta_{2,3,1} \\ \eta_{3,1,1} & \eta_{3,2,1} & \eta_{3,3,1} \\ \eta_{4,1,1} & \eta_{4,2,1} & \eta_{4,3,1} \end{bmatrix}, \qquad \mathcal{H}_{::2} = \begin{bmatrix} \eta_{1,1,2} & \eta_{1,2,2} & \eta_{1,3,2} \\ \eta_{2,1,2} & \eta_{2,2,2} & \eta_{2,3,2} \\ \eta_{3,1,2} & \eta_{3,2,2} & \eta_{3,3,2} \\ \eta_{4,1,2} & \eta_{4,2,2} & \eta_{4,3,2} \end{bmatrix},$$

then we have the following mode-m unfoldings:

$$\mathcal{H}^{(1)} = \begin{bmatrix} \eta_{1,1,1} & \eta_{1,2,1} & \eta_{1,3,1} & \eta_{1,1,2} & \eta_{1,2,2} & \eta_{1,3,2} \\ \eta_{2,1,1} & \eta_{2,2,1} & \eta_{2,3,1} & \eta_{2,1,2} & \eta_{2,2,2} & \eta_{2,3,2} \\ \eta_{3,1,1} & \eta_{3,2,1} & \eta_{3,3,1} & \eta_{3,1,2} & \eta_{3,2,2} & \eta_{3,3,2} \\ \eta_{4,1,1} & \eta_{4,2,1} & \eta_{4,3,1} & \eta_{4,1,2} & \eta_{4,2,2} & \eta_{4,3,2} \end{bmatrix},$$

$$\mathcal{H}^{(2)} = \begin{bmatrix} \eta_{1,1,1} & \eta_{2,1,1} & \eta_{3,1,1} & \eta_{4,1,1} & \eta_{1,1,2} & \eta_{2,1,2} & \eta_{3,1,2} & \eta_{4,1,2} \\ \eta_{1,2,1} & \eta_{2,2,1} & \eta_{3,2,1} & \eta_{4,2,1} & \eta_{1,2,2} & \eta_{2,2,2} & \eta_{3,2,2} & \eta_{4,2,2} \\ \eta_{1,3,1} & \eta_{2,3,1} & \eta_{3,3,1} & \eta_{4,3,1} & \eta_{1,3,2} & \eta_{2,3,2} & \eta_{3,3,2} & \eta_{4,3,2} \end{bmatrix},$$

and

$$\mathcal{H}^{(3)} = \begin{bmatrix} \eta_{1,1,1} & \eta_{2,1,1} & \eta_{3,1,1} & \eta_{4,1,1} & \eta_{1,2,1} & \eta_{2,2,1} & \eta_{3,2,1} & \eta_{4,2,1} & \eta_{1,3,1} & \eta_{2,3,1} & \eta_{3,3,1} & \eta_{4,3,1} \\ \eta_{1,1,2} & \eta_{2,1,2} & \eta_{3,1,2} & \eta_{4,1,2} & \eta_{1,2,2} & \eta_{2,2,2} & \eta_{3,2,2} & \eta_{4,2,2} & \eta_{1,3,2} & \eta_{2,3,2} & \eta_{3,3,2} & \eta_{4,3,2} \end{bmatrix}.$$

Matrix H in system (3.37) is a Hessian of size $(n^2 \times n)$. It can be interpreted as the mode-1 unfolding of the 3rd-order tensor $\mathcal{H}$ of size $(n \times n \times n)$ [BG17], i.e., $H = \mathcal{H}^{(1)}$. The matrix $\mathcal{H}^{(2)}$ is the mode-2 unfolding of $\mathcal{H}$. In practice, the tensor $\mathcal{H}$ is not formed in order to construct its mode-2 unfolding. Instead, $\mathcal{H}^{(2)}$ is composed by rearranging and concatenating the columns of H (or $\mathcal{H}^{(1)}$ respectively), into rows as implied by the example above.

[BG17] derive $\mathcal{H}^{(2)}$ in the process of determining of the observability Gramian $\mathcal{Q}$. They take advantage of the duality principle between reachable and observable subspace, similar to the linear case in Appendix A.1, to acquire the dual quadratic-bilinear ODE system to system (3.37) (including the bilinear term). Then, they exploit properties of the tensor product and unfolding to simplify the quadratic term in the dual system. This leads them to the creation of $\mathcal{H}^{(2)}$.

Figure B.1 shows sparsity pattern of $\mathcal{H}^{(2)}$.

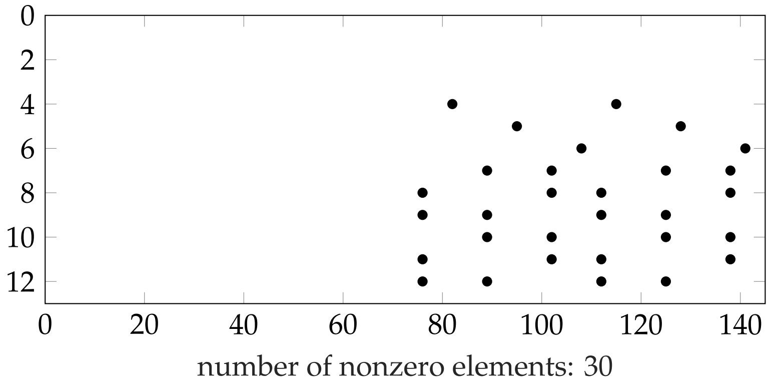


Figure B.1: Sparsity Pattern of $\mathcal{H}^{(2)}$ (EN model, 3-generator/9-nodes system, $n_{co} = 3$)

APPENDIX C

Notes on Codes

Codes & Data

At the moment, this section is yet incomplete. In the printed version of the thesis, you will also find the readme document from the git repository. It includes a guide and instructions of how to handle the files, and a table listing the script which generated the data and the script which created the plot for each figure in Chapter 5.

Under the following link, you can find the code used for the numerical experiments inChapter 5. This encompasses the scripts to generate the data for case57, the data themselves, and the main and auxiliary functions to .

`https://gitlab.mpi-magdeburg.mpg.de/grundel/Strommodelle/tree/master`

Remarks on Algorithms & Algorithm to Compute the Truncated Observability Gramian

In our description of the algorithms in Sections 3.3 and 3.4, and Appendix C, we use the MATLAB functions $[V, \Sigma, W] = \mathsf{svd}(M)$, and $R = \mathsf{chol}(M)$, or $[R, \mathsf{flag}] = \mathsf{chol}(M)$, respectively.

It is imperative to mention that the function chol is very sensitive concerning the positive definiteness of M. If the EVs are too close to zero, MATLAB will not compute the Cholesky decomposition and return with an error. As this happened very often during the computations, we employ the following strategy whenever we have to find the Cholesky factor of a matrix:

1. We run $[R, \mathsf{flag}] = \mathsf{chol}(M)$.

2. If flag equals zero, the algorithm just continues with the next major step.

3. If flag does not equal zero, we apply the auxiliary function $\mathsf{makeSPD}(.)$ to M in order to find a positive definite matrix $\tilde{M}$ that is close to M.

   This function is adapted from [Mar12] and based on the iterative spectral method. [Mar12] tested different algorithms with the aim to approximate a nonpositive definite matrix with a positive definite one. We choose the iterative spectral method, because it was deemed almost as good as the most accurate algorithm (in the Frobenius norm), but is considerably faster. We changed one significant aspect of the algorithm, though. Instead of basing the decision if a matrix is positive definite on the eigenvalues of the matrix, we instead run the chol function and evaluated the flag value in each iteration. This resulted in a much more reliable performance.

4. Finally, we run $R = \mathsf{chol}(\tilde{M})$ and the algorithm continues the next step with the Cholesky factor of this approximate matrix.

Algorithm C.1: Iterative Scheme to Compute Truncated Gramian $\mathcal{Q}_{\mathcal{T}}$

```
Input: quadratic system matrices $A, H, C$, Volterra kernels $P_1, \dots, P_v$, shift
       $\alpha > 0$, number of Volterra kernels $v$
Output: Cholesky factor $S$ (upper triangular matrix) of the truncated
        observability Gramian $Q_{\mathcal{T}}$.
Compute the matrix $\mathcal{H}^{(2)}$.
for $j = 1 : 4n_{co} : 16n_{co}^2$ do
    $\mathcal{H}^{(2)}(:, j : j + 4n_{co} - 1) := H(:, j : j + 4n_{co} - 1)$.
end for
Shift matrix $A$ with $\alpha$ to make the matrix stable: $A_\alpha := A - \alpha\mathbb{I}$.
Determine Cholesky factor $S_1$ of first Volterra kernel $Q_1$ by calling Algorithm 3.1 with $\mathbb{A} := A_\alpha$, $\mathbb{B} := C$ and compute $Q_1 := S_1^* S_1$.
Set second Volterra kernel $Q_2 := 0$.
Compute Cholesky factors $S_3, \dots, S_v$ to determine Volterra kernels $Q_3, \dots Q_v$:
for $i = 3 : v$ do
    $Q_i^{\mathrm{kron}} := 0$ $j := i - 2$;
    Sum up Kronecker products of previously computed $P_k$ and $Q_j$.
    for $k = 1 : i - 2$ do
        $Q_i^{\mathrm{kron}} := Q_i^{\mathrm{kron}} + P_k \otimes Q_j$;
        $j := j - 1$;
    end for
    Set $\mathcal{H}_i^{\mathrm{kron}} := \mathcal{H}^{(2)} Q_i^{\mathrm{kron}} (\mathcal{H}^{(2)})^*$.
    Compute Cholesky factor $\mathbb{B}_i$ of $\mathcal{H}_i^{\mathrm{kron}}$.
    Update Cholesky factor $S_i$ by calling Algorithm 3.1 with
        $\mathbb{A} := A_\alpha$, $\mathbb{B} := \mathbb{B}_i$.
    Compute Volterra kernel $Q_i := S_i^* S_i$.
end for
Determine $Q_{\mathcal{T}}$ by summing up Volterra kernels:
$Q_{\mathcal{T}} = 0$;
for $i = 1 : v$ do
    $Q_{\mathcal{T}} = Q_{\mathcal{T}} + Q_i$.
end for
Compute Cholesky factor $S$ of $Q_{\mathcal{T}}$.
```

Bibliography

[Ant05] A. C. Antoulas. *Approximation of Large-Scale Dynamical Systems*. Advances in Design and Control Series. Society for Industrial and Applied Mathematics, Philadelphia, 2005.

[AS01] A. C. Antoulas and D. C. Sorensen. Approximation of large-scale dynamical systems: an overview. *International Journal of Applied Mathematics and Computer Science*, 11(5):1093–1121, 2001.

[BBF14] 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*, 21:331–358, 2014.

[BG17] P. Benner and P. Goyal. Balanced truncation model order reduction for quadratic-bilinear control systems, 2017. arXiv: `1705.00160`.

[BGR17] P. Benner, P. Goyal, and M. Redman. *Truncated gramians for bilinear systems and their advantages in model order reduction*. In *Model Reduction of Parametrized Systems*. P. Benner, M. Ohlberger, A.T. Patera, G. Rozza, and K. Urban, editors. Volume 17. MS & A Modeling, Simulation and Applications. Springer International Publishing, 2017. Chapter 18, pages 285–300.

[BPK71] C. Bruni, G. Di Pillo, and G. Koch. On the mathematical models of bilinear systems. *Ricerche Di Automatica*, 2(1):11–26, January 1971.

[BV00] A. R. Bergen and V. Vittal. *Power Systems Analysis*. Prentice Hall, New Jersey, 2000.

[Che99] Y. Chen. *Model Order Reduction for Nonlinear Systems*. Master's Thesis in Mathematics, Massachusetts Institute of Technology, 1999.

[DB12] F. Dörfler and F. Bullo. Synchronization and transient stability in power networks and non-uniform kuramoto oscillators. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 60(1):150–163, 2012.

[DB13] F. Dörfler and F. Bullo. Kron reduction of graphs with applications to electrical networks. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 60(1):150–163, 2013.

[DCB13] F. Dörer, M. Chertkov, and F. Bullo. Synchronization in complex oscillator networks and smart grids. *Proceedings of the National Academy of Sciences*, 110(6):2005–2010, 2013.

[FKR+04] L. Feng, D. Koziol, E. B. Rudnyi, and J. G. Korvink. Model order reduction for scanning electrochemical microscope: the treatment of nonzero initial condition. In volume 3, pages 1236–1239, Piscataway, NJ. IEEE, October 2004.

[Fla12] G. M. Flagg. *Interpolation Methods for the Model Reduction of Bilinear Systems*. Dissertation in Mathematics, Virginia Polytechnic Institute and State University, 2012.

[FNP08] G. Filatrella, A. H. Nielsen, and N. F. Pedersen. Analysis of a power grid using a kuramoto-like model. *The European Physical Journal B*, 61(4):485–491, 2008.

[GM96] W. S. Gray and J. Mesko. Controllability and observability functions for model reduction of nonlinear systems. *Proceedings of the Conference on Information Science and Systems*:1244–1249, 1996.

[GRA14] O. Gomez, M. A. Rios, and G. Anders. Reliability-based phasor measurement unit placement in power systems considering transmission line outages and channel limits. *IET Generation, Transmission & Distribution*, 8:121–130(9), 1, January 2014.

[GS94] J. J. Grainger and W. D. Stevenson Jr. *Power System Analysis*. Series in Electrical and Computer Engineering. McGraw-Hill Co., Singapore, 1994.

[Gu11] C. Gu. *Model Order Reduction of Nonlinear Dynamical Systems*. Dissertation in Electrical Engineering and Computer Science, University of California, Berkeley, 2011.

[Ham82] S. J. Hammarling. Numerical solution of the stable, non-negative definite lyapunov equation. *IMA Journal of Numerical Analysis*, 2:303–323, 1982.

[Hig96] N. J. Higham. *Accuracy and Stability of Numerical Algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, 1996.

[IL14] R. Idema and D. J.P. Lahaye. *Computational Methods in Power System Analysis*. Atlantis Studies in Scientific Computing in Electromagnetics. Atlantis Press, Paris, 2014.

[Isi85] A. Isidori. *Nonlinear Control Systems: An Introduction*. M. Thoma, editor, volume 72 of *Lecture Notes in Control and Information Sciences*. Springer-Verlag, Berlin, Heidelberg, New York, Tokyo, 1985.

[iST] SLICOT Subroutine Library in Systems and Control Theory. Sb03od. Solution of stable continuous- or discrete-time Lyapunov equations (Cholesky factor). Niconet e.V. URL: `http://slicot.org/objects/software/shared/doc/SB03OD.html` (visited on 09/03/2018).

[KB09] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

[KW19] B. Kramer and K. E. Willcox. Nonlinear model order reduction via lifting transformations and proper orthogonal decomposition. *AIAA Journal*, 57(6):2297–2307, 2019.

[Lue79] D. G. Luenberger. *Introduction to Dynamic Systems: Theory, Models, and Applications*. J. Wiley & Sons, New York, Chichester, Brisbane, Toronto, 1979.

[Mar12] S. Marée. *Correcting Non Positive Denite Correlation Matrices*. Bachelor's Thesis in Applied Mathematics, TU Delft, 2012.

[MMA$^+$13] A. E. Motter, S. A. Myers, M. Anghel, and T. Nishikawa. Spontaneous synchrony in power-grid networks. *Nature Physics*, 9:2005–2010, 2013.

[NM15a] T. Nishikawa and A. E. Motter. Comparative analysis of existing models for power-grid synchronization. *New Journal of Physics*, 17, January 2015.

[NM15b] T. Nishikawa and A. E. Motter. Power-grid synchronization models. Computing parameters for network dynamics models. January 2015. URL: `https://sourceforge.net/projects/pg-sync-models/` (visited on 01/23/2018).

[Pin08] R. Pinnau. *Model reduction via proper orthogonal decomposition*. In *Model Order Reduction: Theory, Research Aspects and Applications*. W. H. A. Schilders and H. A. van der Vorst and J. Rommes, editors. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pages 95–109.

[Pol14] D. Pollock. On Kronecker Products, Tensor Products and Matrix Differential Calculus. Discussion Papers in Economics 14/02, Division of Economics, School of Business, University of Leicester, 2014.

[RP03] M. Rathinam and L. Petzold. A new look at proper orthogonal decomposition. *SIAM Journal on Numerical Analysis*, 41(5):1893–1925, 2003.

[Rug81] W. J. Rugh. *Nonlinear Systems Theory*. Johns Hopkins University Press, Baltimore, 1981. web version prepared in 2002.

[Sch93] J.M.A. Scherpen. Balancing for nonlinear systems. *Systems & Control Letters*, 21(2):143–153, 1993.

[Sne] M. Snelgrove. Oscillator. McGraw-Hill Education. URL: `https://doi.org/10.1036/1097-8542.477900` (visited on 02/15/2019).

[TSS$^+$18] L. Thurner, A. Scheidler, F. Schäfer, J. Menke, J. Dollichon, F. Meier, S. Meinecke, and M. Braun. Pandapower an open-source python tool for convenient modeling, analysis, and optimization of electric power systems. *IEEE Transactions on Power Systems*, 33(6):6510–6521, November 2018. URL: `http://www.pandapower.org/` (visited on 08/12/2019).

[VG18] M. Cruz Varona and R. Gebhart. Impulse response of bilinear systems based on volterra series representation, 2018. arXiv: `1812.05360`.

[Vol13] S. Volkwein. Proper orthogonal decomposition: theory and reduced-order modelling, Lecture Notes, 2013.

[Wil10] J. C. Willems. *Ports and terminals*. In *Perspectives in Mathematical System Theory, Control, and Signal Processing: A Festschrift in Honor of Yutaka Yamamoto on the Occasion of his 60th Birthday*. J. C. Willems, S. Hara, Y. Ohta, and H. Fujioka, editors. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pages 27–36.

[WT12] D. Witthaut and M. Timme. Braess's paradox in oscillator networks, desynchronization and power outage. *New Journal of Physics*, 14(8):083036, 2012.

[ZD13] H. Zhang and F. Ding. On the kronecker products and their applications. *Journal of Applied Mathematics*, 2013:Article ID 296185, 2013.

[Zha13] X. Zhan. *Matrix Theory*, volume 2013 of *Graduate Studies in Mathematics Series*. American Mathematical Society, Providence, Rhode Island, 2013.

[ZM16] R. D. Zimmerman and C. E. Murillo-Sanchez. Matpower. Version 6.0. January 2016. URL: `https://matpower.org` (visited on 01/23/2018).

[ZMT11] R. D. Zimmerman, C. E. Murillo-Sanchez, 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, February 2011.

Statutory Declarations

I, Frances Weiß, declare that I have developed and written the enclosed Master Thesis completely by myself, and have not used sources or means without declaration in the text. Any thoughts from others or literal quotations are clearly marked. The Master Thesis was not used in the same or in a similar version to achieve an academic grading.

Magdeburg, August 30, 2019