Combined State and Parameter Reduction

for Nonlinear Systems with an Application in Neuroscience

Christian Himpe
aus Unna

Inauguraldissertation
zur Erlangung des Doktorgrades der Naturwissenschaften im Fachbereich Mathematik und Informatik der Mathematisch-Naturwissenschaftlichen Fakultät der Westfälischen Wilhelms-Universität Münster.

- 2016 -

Dekan: Prof. Dr. Martin Stein
Erster Gutachter: Prof. Dr. Mario Ohlberger
Zweiter Gutachter: Prof. Dr. Peter Benner

Tag der mündlichen Prüfung: 20.06.2016
Tag der Promotion: 20.06.2016

Abstract

This work investigates two complementary methods for combined state and parameter reduction of nonlinear systems. First, a system-theoretic approach using empirical gramians, and second, an iterative method utilizing the greedy algorithm. The presented methods are applied in the context of connectivity analysis of functional neuroimaging data, in which these nonlinear model reduction techniques are demonstrated to accelerate the solution of many-query problems enabling the data-driven exploration of more complex neuronal networks, for example in the human brain.

Zusammenfassung

Diese Arbeit untersucht zwei komplementäre Methoden für die kombinierte Zustands- und Parameterreduktion von nichtlinearen Systemen. Einerseits wird ein systemtheoretischer Ansatz untersucht, der empirische Gram-Matrizen benutzt, andererseits eine iterative Methode, die den Greedy-Algorithmus verwendet. Die vorgestellten Methoden werden auf die Konnektivitätsanalyse aus funktionellen neurobildgebenden Verfahren angewandt, für welche gezeigt wird, dass diese nichtlinearen Modellreduktions-Techniken oft zu wiederholende Lösungen beschleunigen und damit eine datengetriebene Erforschung von komplexeren neuronalen Netzwerken erlauben, zum Beispiel im menschlichen Gehirn.

Categorization

Cat: Mathematics / Model Reduction
DDC: 518 (Numerical Analysis)
LCC: QA (Mathematics)

MSC: 93B99 (Controllability, Observability, and System Structure)
ACM: G.4 (Mathematical Software)
PACS: 02.30.Yy (Control Theory)

Keywords: Model Reduction, Model Order Reduction, Combined Reduction, Controllability, Observability, Cross Gramian, Greedy Sampling

Acknowledgements

I thank:

- Mario Ohlberger for his scientific guidance during the preparation of this work,
- Andreas Buhr, René Milk, Stephan Rave and Felix Schindler for countless scientific and technical discussion which lead to new ideas and insights,
- Felix Luca and Andreas Nüßing for our calm office collective,
- the whole Workgroup Ohlberger, as well as the other workgroups of the institute for Applied Mathematics Münster for a great work atmosphere,
- Ulrike Baur, Martin Köhler, Jens Saak and the Computational Methods in Systems and Control Theory group at the MPI Magdeburg for a seamless cooperation,
- Thomas Knösche and the MEG and EEG - Cortical Networks and Cognitive Functions group at the MPI Leipzig for valuable discussions,
- Athanasios Antoulas for numerous discussions on model reduction,
- Stephan Volkwein for the in-depth explanations on optimization topics,
- Anne Schindler for proof reading,
- the MORwiki community for the (ongoing) effort of compiling this splendid online resource on model order reduction,
- my parents and grandparents for encouraging me to pursue mathematics,
- and my wife Nina-Claire Himpe for her love and steadfast support.

This thesis was supported by the Deutsche Forschungsgemeinschaft, DFG EXC 1003 Cells in Motion - Cluster of Excellence, Münster, Germany as well as by the Center for Developing Mathematics in Interaction, DEMAIN, Münster, Germany.

Contents

List of Figures and Tables

List of Symbols

$\imath$ – Imaginary unit $\sqrt{-1}$

Re – Real part of complex number

$\dot{x}$ – Newton's notation of the time derivative $\frac{\partial x}{\partial t}$

C^0 – Space of continuous functions

C^1 – Space of continuously differentiable functions

ess sup – Essential supremum

$|Q|$ – Cardinality of the set Q

$\langle \cdot, \cdot \rangle$ – Euclidean inner product

$\langle \cdot, \cdot \rangle_X$ – Inner product with respect to $X = X^\intercal > 0$

$*$ – Convolution operator

$\circ$ – Composition operator

$\odot$ – Hadamard product (component-wise product)

$\otimes$ – Kronecker product (tensor product)

vec – Vectorization operator

0_N – Zero matrix of dimension $N \times N$

$\mathbb{1}_N$ – Unit Matrix of dimension $N \times N$

$\delta_{i,j}^{m \times n}$ – $m \times n$ dimensional Kronecker matrix, with non-zero element (i, j)

$\vec{1}_N$ – Vector of dimension N with each element set to one

$\chi_{[a,b)}(t)$ – Indicator function on the half open interval $[a, b)$

$\delta(t)$ – Dirac delta distribution

$\varsigma(x)$ – Sigmoid Function

$\bar{\varsigma}(x)$ – Sigmoid-like function (y-shifted and scaled sigmoid function)

$\mathrm{tr}(A)$ – Trace of a matrix or operator A

im – Image space

$\ker$ – Null space

A^+ – Moore-Penrose inverse to A

$\lambda_i(A)$ – i-th eigenvalue of A

$\sigma_i(A)$ – i-th singular value of A

$\mathscr{U}_{[a,b]}$ – Uniform random distribution on $[a, b]$

$\mathscr{N}_{m,v}$ – Normal random distribution with mean m and covariance v

$\ln \mathscr{N}_{m,v}$ – Log-Normal random distribution with location m and scale v

$\mathbb{E}[X]$ - Mean / Expected value

A_{i*} – i-th row of A

A_{*j} – j-th column of A

diag – Main diagonal of a matrix

orth – Orthogonal part of a matrix

$\operatorname{pod}_m(x)$ – first m (dominant) POD modes of x

$\mathscr{C}$ – Controllability operator

$\mathscr{O}$ – Observability operator

$\|\cdot\|_H$ – Hankel-norm

$\|\cdot\|_{\mathrm{L}_p}$ – Lebesgue L_p-norm

$\|\cdot\|_{\ell_p}$ – Discrete Lebesgue ℓ_p-norm

$\|\cdot\|_{\mathrm{L}_p \otimes \mathrm{L}_q}$ – Joint Lebesgue $\mathrm{L}_p \otimes \mathrm{L}_q$-norm

$\|\cdot\|_{\ell_p \otimes \ell_q}$ – Discrete joint Lebesgue $\ell_p \otimes \ell_q$-norm

$\|\cdot\|_{\mathscr{H}_p}$ – Hardy $\mathscr{H}_p$-norm

$\|\cdot\|_*$ – Trace-norm

$\|\cdot\|_F$ – Frobenius-norm

1. Introduction

Contents

Ever more detailed mathematical models for numerical simulations of natural and technical processes demand increasing computing power. Combined reduction is a type of model reduction that allows to decrease the computational complexity while retaining the model's functionality even for nonlinear relations.

1.1. Motivation

The human brain, with its estimated 10^{11} neurons[1], embodies one of the most complex systems known to mankind. This enormous network of neurons can process information to a degree enabling abstract thought and sophisticated planning.
Functional neuroimaging data reveals downstream physical effects of the brain's inner workings, yet not the networked neuronal activity itself. An inverse problem to reconstruct the connectivity of a hidden (neuronal) network from neuroimaging data is formed, which is a computationally challenging task even for a few network(ed) nodes. Hence, for large-scale networks surrogate models with similar dynamic behavior, yet of reduced dimensionality, are sought. Model reduction enables the computation of such smaller approximate systems. The general underlying mathematical model is an input-output system, given by a time-invariant, parametrized, possibly nonlinear, control system:

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t), \theta), \\ y(t) &= g(x(t), u(t), \theta), \end{aligned} \tag{1.1}$$

with the state x representing neuronal activity among the considered nodes (brain regions). The change of neuronal activity over time $\dot{x}$ is described by the vector field f, that depends on x, the external input u and the parameters θ; the latter embody, for example, the network connectivity between the individual nodes. In this setting the output functional g then transforms the neuronal activity x to the measurable output y. Essentially, this model maps inputs u to outputs y, as illustrated in Figure 1.1.

[1] See for example: [54, Ch. 3].

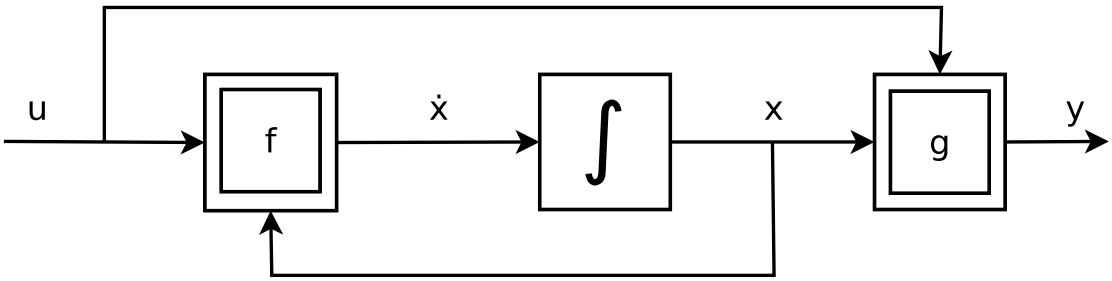


Figure 1.1.: *A schematic input-output system.*

In experiments with known input u, output data y_d is recorded. The inverse problem is constituted by estimating the parameters θ such that the model's output y matches the recorded output data y_d. This inversion of measured data to deduce the underlying connectivity can be computationally infeasible for complex networks, due to high-dimensional state- and parameter-spaces.

1.2. Aim

In settings with a high-dimensional state-space $\dim(x(t)) \gg 1$ and a high-dimensional parameter-space $\dim(\theta) \gg 1$, repeated simulations for different locations in the parameter-space, as in estimation or optimization of θ, may be computationally costly. The presented combined state and parameter reduction methods are targeted at such many-query or related situations, for example in uncertainty quantification. By reduction of the state-space, the solution for a certain location in the parameter-space is accelerated, while by reduction of the parameter-space, the overall number of required solutions is reduced.
In this context an application in neuroscience is investigated: To infer connectivity between multiple regions of the brain, the parameters of a network model are optimized to match the model's output to neuroimaging recordings. For networks with many nodes, the parameter inference of the associated large-scale model presents a use case for these combined (state and parameter) reduction methods.

1.3. Dual Approach

This work explores two methods for nonlinear model order reduction, more specifically the combined state and parameter reduction [112] of nonlinear control systems. First, a system-theoretic ansatz derived from **balanced truncation** is presented, which assesses system properties like controllability and observability encoded in (gramian) matrices for linear systems, and extends to nonlinear systems by the use of empirical gramian matrices. Second, an optimization-based approach based on the **greedy algorithm** is investigated that assembles iteratively a low-dimensional base from locally optimal base components, and also extends to nonlinear systems.

1.4. Literature Review

In this section major works on the explored topics of model order reduction are outlined, which are the basis for the methods developed in the following chapters, first, on the gramian-based methods, then, on the optimization-based methods. In addition, related methods are mentioned and an overall classification is given.

1.4.1. Gramian-Based Model Reduction

Gramian-based model reduction originates in [168] by MOORE, introducing the balanced truncation method, which proposes the balancing of a system in terms of **controllability**[2] and **observability**[3] [132] (encoded in gramian matrices) and truncating the least important states in this view. ENNS [61] and GLOVER [86] discovered global error bounds of the resulting reduced order model, while an efficient balancing procedure was devised in [146] and the related concept of balanced gains was introduced in [131, 49].
In [63, 64, 65, 66, 67, 68, 69], FERNANDO and NICHOLSON introduced the cross gramian, combining controllability and observability into a single matrix. The scope of the cross gramian concept was expanded in [145, 1, 195] and also revisited by SORENSEN and ANTOULAS in [209, 210]. Another notable work on the cross gramian is [4] by ALDHAHERI. Numerical methods for the computation of the cross gramian encompass, for example, the utilization of the matrix sign function [14] by BAUR and BENNER or a variant of the alternating direction implicit iteration [188] by SAAK, BENNER and KÜRSCHNER.
The concept of nonlinear balancing was formally introduced by SCHERPEN in [189, 190]. Based on the former, a nonlinear cross gramian is assessed in [123, 124, 125, 126, 127, 191, 79] mainly by IONESCU, FUJIMOTO and SCHERPEN. An alternative ansatz for gramian-based model reduction of nonlinear systems are empirical gramians. Empirical (controllability and observability) gramians were introduced by LALL, MARSDEN and GLAVASKI [142, 143], which were further developed (among others) in [97, 98, 99, 100, 102] by HAHN and EDGAR and in [44, 45, 43] by CONDON and IVANOV. The empirical cross gramian for Single-Input-Single-Output systems was introduced by STREIF *et al.* [214, 215] and generalized to symmetric Multiple-Input-Multiple-Output systems by the author and OHLBERGER in [110]. Moreover, parametric model order reduction using the (empirical) cross gramian is presented in [113] and an extension of the cross gramian to non-symmetric systems is introduced in [117], both by the author and OHLBERGER.
Using empirical gramians for system identification is demonstrated from a controllability point of view in [217, 218] by SUN and HAHN. From an observability point of view, empirical-gramian-based parameter identification is explored in [200, 201, 203] by SINGH and HAHN as well as in [83, 82] by GEFFEN and applied in [58]. A first step in cross-gramian-based system identification is made in [214]. In [110], the empirical cross gramian is extended by the author and OHLBERGER to enable cross-gramian-based parameter identification and thus concurrent combined state and parameter reduction[4].

[2] The effectiveness of the input u in driving the state x, see Section 3.2.1.
[3] The effectiveness of the output y in characterizing the state x, see Section 3.2.2.
[4] This approach is assessed in [91] by GRESSER.

1.4.2. Optimization-Based Model Reduction

An alternative approach is motivated by large-scale (Bayesian) inverse problems for which model reduction is employed in order to enable or accelerate the approximation of a solution [75]. The optimization-based approach investigated in this work is based upon work by LIEBERMAN, WILLCOX and GHATTAS in [151]. This method combines parameter-space reduction and state-space reduction targeted at (Bayesian) inverse problems. The parameter-space reduction utilizes the greedy algorithm, described in [225] by VEROY, PRUD'HOMME, ROVAS and PATERA, to obtain a reduced order parameter sub-space [31]. For the state-space reduction in the context of inverse problems, various approaches have been explored: The use of proper orthogonal decomposition for optimal control is investigated by KUNISCH and VOLKWEIN in [140], and by NGUYEN and WILLCOX for Bayesian inversion in [170]. A goal-oriented approach is introduced in [229] by WILLCOX *et al.*, which is used in [30] by BUI-THANH *et al.* for the reduced order optimization.
Another related method is the Hessian-matrix-based model reduction method targeted at input-free systems, introduced by BASHIR *et al.* [12, 13] and discussed in [149]. In [154], this Hessian-based approach is connected to the system-theoretic concept of observability and a generalization to nonlinear systems is presented in [80] by GALBALLY *et al.* A low-rank approximation of the Hessian is developed in [71] by FLATH *et al.*
In [114], the author and OHLBERGER extend the original approach from [151] by incorporating the observed data and accelerating the reduced order model assembly using a randomized approach.
In a wider sense, the low-rank approximation used in [32] by BUI-THANH *et al.* is a related ansatz, and the inference for prediction method introduced in [150, 153] by LIEBERMAN connects the system-theoretic balanced truncation with Bayesian inversion.

1.4.3. Related work

Both, the empirical-gramian-based and the optimization-based model reduction methods are data-driven, and loosely related to the method of snapshots [205] by SIROVICH utilizing proper orthogonal decomposition (POD) [141]. More closely related to the empirical gramian method is balanced POD [228] introduced by WILLCOX and further developed by ROWLEY in [184]. Remotely related approaches are the Loewner framework by ANTOULAS and IONITA [128] as well as the utilization of the impulse response gramian and the system's gram matrix by SREERAM and AGATHOKLIS [212].
Apart from the reduction of (nonlinear) state-space systems, model reduction is also applicable to partial differential equations[5] (PDE), such as the Navier-Stokes equation [9]. In the realm of PDEs, the reduced basis method (RBM) [186], is employed in the context of parametric model order reduction for evolution equations in [96] by HAASDONK and OHLBERGER. Furthermore, the optimization-based approach is related to the field of PDE-constrained optimization [22].
An overview of related model reduction methods and associated software implementations is given for example in [19, 7, 8, 17, 20, 23].

[5] The methods investigated in this work are restricted to spatially discretized instationary PDEs.

1.4.4. Synoptic View

The presented methods are applicable to first and second order nonlinear parametric systems, and even though the focus in this work is on input-output systems governed by ODEs (1.1), the presented methods also apply to discretized (instationary) PDEs. In the context of other model reduction techniques, the presented methods can be classified as follows: While the gramian-based combined reduction is a balancing method which extends the (algebraic) system gramians to the empirical gramians with focus on the cross gramian, the optimization-based combined reduction joins a greedy parameter-space reduction with an energy-based state-space reduction and thus is related to the class of POD-based methods.
A layout of the interrelations of the mentioned methods is given in Figure 1.2. It should also be noted that both methods for combined reduction produce globally reduced order models. Neither the underlying system (state-space) nor the associated parameter-space is partitioned, which would yield locally reduced models.

1.4.5. Neuroscientific Application

The modeling of neuronal networks and their reconstruction from functional neuroimaging data in terms of causal interactions between different regions of the brain is a major research objective in the neurosciences. To this end the **dynamic causal modelling** framework, introduced in [77] by FRISTON *et al.* can be employed, which models the hidden underlying neuronal network by a dynamic system and the measurements by an attached output functional. Together, this amounts to a control system model (1.1), which can be evaluated with system-theoretic tools as in [169] by MORAN *et al.* The parametrized network connectivity is then estimated by Bayesian inference from time series data constrained by the model.

1.5. Overview

This work is composed of two main parts. The first part sets up the mathematical theory and develops the combined state and parameter reduction methods; the second part summarizes the software realization, introduces the neuroscientific application and assesses the computational results.

1.5.1. Outline

The subsequent content is structured as follows: Chapter 2 reviews control systems and the properties important to model reduction methods. In Chapter 3 gramian-based state reduction, parameter reduction and combined state and parameter reduction are presented. Chapter 4 describes the optimization-based combined reduction. The implementation of the presented approaches is discussed in Chapter 5. An outline of network-based connectivity inference, the target application, is given in Chapter 6. Numerical experiments and their results are evaluated and discussed in Chapter 7. Finally, a conclusion and an outlook are given in Chapter 8.

1.5.2. Published Results

This work is based upon peer-reviewed research articles and proceedings by the author. The empirical-cross-gramian-based combined reduction in Chapter 3 is originally introduced in the article [110], and the use of empirical gramians for parametric systems is presented in the proceedings [113]. Furthermore, the non-symmetric cross gramian is proposed in the article [117]. Chapter 4 presents the optimization-based combined reduction of instationary systems which is based on the article [114]. The empirical gramian framework, described in Chapter 5, is initially described in the article [109]. Finally, the combined reduction for network models and neuroimaging models, analyzed in Chapter 7, is devised in the proceedings [111] and [112] respectively.

1.5.3. Key Contributions

In brief, the following contributions are featured:

- Nonlinear (Cross-)Gramian-based combined state and parameter reduction.
- Nonlinear (Greedy-)Optimization-based combined state and parameter reduction.
- Reusable software implementation for both methods.
- Application of the combined reduction in systems neuroscience.

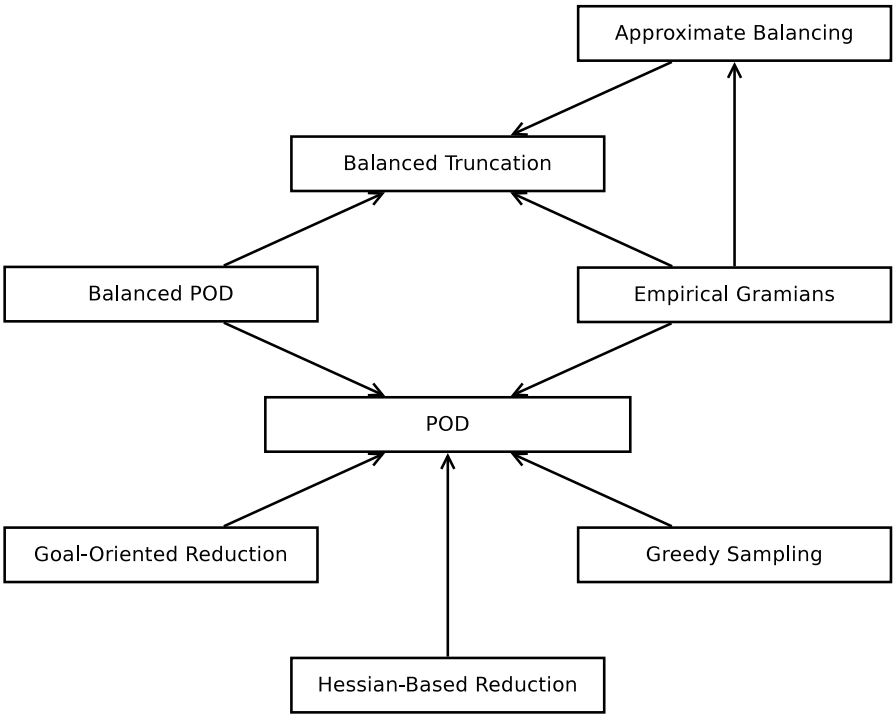


Figure 1.2.: *Interrelations of referenced model reduction methods.*

2. Preliminaries

Contents

In this chapter an outline of dynamical systems, control systems and model reduction is provided to establish notation and terminology. Since the theory for the employed approaches in nonlinear model order reduction is based on results from linear system theory, this summary focuses on the linear setting.

2.1. Dynamical Systems

First, a minimal introduction to dynamical systems is provided, as a foundation to the enclosing concept of control systems, which is presented in the next Section 2.2.
A continuous dynamical system with finite dimensional state-space $N < \infty$ can be described by an ordinary differential equation (ODE) [147, Ch. 6],

$$\dot{x}(t) = f(t, x(t)), \tag{2.1}$$

with $t \in \mathbb{R}^{>0}$, the state $x : \mathbb{R} \to \mathbb{R}^N$ and a vector field $f : \mathbb{R} \times \mathbb{R}^N \to \mathbb{R}^N$. An ODE together with an initial condition $x_0 = x(t_0)$ forms an initial value problem (IVP). By the theorem of Picard-Lindelöf (see for example [94]) an IVP has a unique solution, if f is Lipschitz-continuous. The image of the solution $x(t)$ is called orbit or **trajectory**.
A time-invariant[6] homogeneous linear dynamical system has a time independent linear vector field $f(t, x(t)) = f(x(t))$, given by a transition matrix $A \in \mathbb{R}^{N \times N}$:

$$\dot{x}(t) = Ax(t). \tag{2.2}$$

In this homogeneous linear setting, the solution for the IVP is given by:

$$x(t) = \mathrm{e}^{At} x_0,$$

with the fundamental solution operator L,

$$L(\cdot)(t) = \mathrm{e}^{At}. \tag{2.3}$$

[6] Also known as: autonomous (system).

For an inhomogeneous linear system, with the continuous function $b : \mathbb{R} \to \mathbb{R}^N$, $b \in \mathrm{C}^0$,

$$\dot{x}(t) = Ax(t) + b(t),$$

the solution is given by the sum of the homogeneous solution and an inhomogeneous solution, the latter is a convolution of the fundamental solution (2.3) with b,

$$x(t) = L(x_0)(t) + (L * b)(t) = \mathrm{e}^{At}\, x_0 + \int_0^t \mathrm{e}^{A\tau}\, b(\tau)\, \mathrm{d}\tau,$$

which is a consequence of Duhamel's principle [148, Ch. 5.1.1].
Usually, a solution to an IVP is additionally sought to be stable. Stability can roughly be defined by $\|x(t)\| < \infty \;\; \forall\, t > 0$; yet, a more precise definition requires a steady-state[7] $\bar{x}$ of the dynamical system, which is a root of the vector field f:

$$f(\bar{x}) = 0,$$

thus, $\bar{x} := x(t_0) \Rightarrow x(t > t_0) = \bar{x}$ holds. A solution $x(t)$ to (2.1) is called (Lyapunov) **stable** if, given a steady-state $\bar{x}$, for all $\epsilon > 0$ there exists a $\delta > 0$ such that:

$$\|x(t) - \bar{x}\| < \epsilon \;\; \forall\, t \geq 0, \;\; \forall\, x_0 \;:\; \|x_0 - \bar{x}\| < \delta.$$

Furthermore, a solution $x(t)$ is called **asymptotically stable** if for all $\epsilon > 0$ **and all** $\delta > 0$ there exists a $T > 0$ such that:

$$\|x(t) - \bar{x}\| < \epsilon \;\; \forall\, t \geq T, \;\; \forall\, x_0 \;:\; \|x_0 - \bar{x}\| < \delta.$$

If additionally there exist constants $c_1, c_2 \in \mathbb{R}^{>0}$ such that:

$$\|x(t) - \bar{x}\| < c_1\, \mathrm{e}^{-c_2 t}\, \|x_0 - \bar{x}\|, \;\; \forall\, t \geq 0,$$

a solution $x(t)$ is called **exponentially stable**.
Linear systems (2.2) are exponentially stable (which implies asymptotic stability) if and only if the real parts of all eigenvalues of A are negative:

$$\mathrm{Re}(\lambda_i(A)) < 0, \;\; \forall\, i = 1, \ldots, N; \tag{2.4}$$

in this case the (transition) matrix A is called a Hurwitz matrix.
For a nonlinear vector field, the system's stability can be assessed locally in a neighborhood around a steady-state by linearization. If the vector field $f \in \mathrm{C}^1$ is continuously differentiable, a local **linearization** of the vector field at the steady-state $\bar{x}$ is given by the Jacobian of f:

$$A = \frac{\partial f}{\partial x}(\bar{x})$$
$$\Rightarrow \dot{x}_l(t) = Ax_l(t) \approx f(x(t)) : \|x(t) - \bar{x}\| < \epsilon.$$

Then, the trajectory $x(t)$ is locally exponentially stable near $\bar{x}$ if $x_l(t)$ is exponentially stable.

[7] Also known as: equilibrium point.

2.2. Control Systems

This section defines control systems, establishes notation and summarizes some attributes of linear control systems. Since the methods developed in Chapter 3 and Chapter 4 are mainly concerned with the time-domain evaluation of control systems, frequency-domain related concepts are omitted if not explicitly required.
First, a general definition of a control system in state-space formulation is given.

Definition 2.1 (Control System)
*A **control system** consists of a dynamical system and an output function. The dynamical system has a vector field $f : \mathbb{R} \times \mathbb{R}^N \times \mathbb{R}^M \times \mathbb{R}^P \to \mathbb{R}^N$ depending on time t, the state $x : \mathbb{R} \to \mathbb{R}^N$, input $u : \mathbb{R} \to \mathbb{R}^M$ and parameter $\theta \in \mathbb{R}^P$. The output functional $g : \mathbb{R} \times \mathbb{R}^N \times \mathbb{R}^M \times \mathbb{R}^P \to \mathbb{R}^O$ also depends on time t, the state $x(t)$, input $u(t)$ and parameter θ,*

$$\begin{aligned} \dot{x}(t) &= f(t, x(t), u(t), \theta), \\ y(t) &= g(t, x(t), u(t), \theta). \end{aligned} \tag{2.5}$$

Extracted from the state $x(t)$, the output $y(t)$ of the output functional g represents some quantity of interest (QoI), for example measurements by a small set of sensors. Additionally, a unique solution of $y(t)$ requires an initial condition $x_0 \in \mathbb{R}^N$,

$$x(0) = x_0.$$

For the dimensions of the control system components, the following notation is used consistently throughout this work:

- Input: $M := \dim(u(t)) < \infty$,
- State: $N := \dim(x(t)) < \infty$,
- Output: $O := \dim(y(t)) < \infty$,
- Parameter: $P := \dim(\theta) < \infty$.

A special class of control systems are linear control systems.

Definition 2.2 (Linear Control System)
*A **linear control system** is a control system with a linear vector field and a linear output functional. The vector field is a sum of a linear transformation $A \in \mathbb{R}^{N \times N}$ of the state $x(t)$, and a linear transformation $B \in \mathbb{R}^{N \times M}$ of the input $u(t)$. The output functional is also a sum of a linear transformation $C \in \mathbb{R}^{O \times N}$ of the state $x(t)$ and a linear transformation $D \in \mathbb{R}^{O \times M}$ of the input $u(t)$,*

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

Unless noted otherwise, linear **time-invariant** (LTI) systems are considered, thus:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t). \end{aligned} \tag{2.6}$$

The following names are used for the components of linear control systems:

- System Matrix: $A \in \mathbb{R}^{N\times N}$,
- Input Matrix: $B \in \mathbb{R}^{N\times M}$,
- Output Matrix: $C \in \mathbb{R}^{O\times N}$,
- Feedforward Matrix[8]: $D \in \mathbb{R}^{O\times M}$.

Yet, for the remainder of this work we will consider a trivial feedforward matrix:

$$D = 0,$$

since it is not affected by the presented model reduction methods. Following [6], a shorthand notation of a linear control system is given by a block matrix $\Sigma(A, B, C)$:

$$\Sigma = \left(\begin{array}{c|c} A & B \\ \hline C & 0 \end{array}\right).$$

Some required attributes of linear control systems are summarized next.
An input function $u(t) = \delta(t) \in \mathrm{L}_2$ with $\int_0^\infty \delta(t)\,\mathrm{d}t = 1$ is called impulse input and the **impulse response matrix**,

$$\tilde{g}(t) = C\,\mathrm{e}^{At}B,$$

is the fundamental solution to an LTI system. The operator mapping impulse input to the output of a linear control system is called impulse response.

Definition 2.3 (Impulse Response)
*The **impulse response** of a linear system is given by:*

$$g(t) = \begin{cases} C\,\mathrm{e}^{At}B & t \geq 0 \\ 0 & t < 0 \end{cases}.$$

For negative time the impulse response $g(t)$ is set to zero to ensure causality [105, Ch. 3]. The convolution of the impulse response with an input function $u(t)$ then yields the output $y(t)$ for a zero initial state $x_0 = 0$,

$$y(t) = (g * u)(t) = \int_0^\infty C\,\mathrm{e}^{A(t-\tau)}Bu(\tau)\,\mathrm{d}\tau, \tag{2.7}$$

which leads to the concept of input-output stability. A linear system, which for bounded input $\|u\|_{\mathrm{L}_\infty} < \infty$ generates bounded output $\|y\|_{\mathrm{L}_\infty} < \infty$, is called **bounded-input-bounded-output** (BIBO) stable. As a consequence of Young's inequality (Appendix A.1), this is the case if the L_1-norm of the impulse response (2.19) exists, i.e. $\|g\|_{\mathrm{L}_1} < \infty$:

$$\|y\|_{\mathrm{L}_\infty} = \|g * u\|_{\mathrm{L}_\infty} \leq \|g\|_{\mathrm{L}_1}\|u\|_{\mathrm{L}_\infty}. \tag{2.8}$$

[8] Also known as: feedthrough matrix.

The associated system running backward in time $\hat{t} := -t$ is called adjoint system Σ^* and has the impulse response $g^*(\hat{t}) = -B^\intercal \mathrm{e}^{A^\intercal \hat{t}}\, C^\intercal$, thus it has the following state-space description [6, Ch. 5.2].

Definition 2.4 (Adjoint System)
*The **adjoint system** to a linear control system is given by:*

$$\begin{aligned} \dot{z}(\hat{t}) &= -A^\intercal z(\hat{t}) - C^\intercal u^*(\hat{t}), \\ y^*(\hat{t}) &= \ B^\intercal z(\hat{t}). \end{aligned}$$

The control system (2.6) is represented in the time-domain; a Laplace transformation of this state-space representation of the impulse response leads to a frequency-space representation of the control system: the transfer function.

Definition 2.5 (Transfer Function)
*A linear system's **transfer function** for a frequency s is given by:*

$$G(s) = C(s\,\mathbb{1}_N - A)^{-1}B.$$

The **system gain** S corresponds to the trace of the transfer function at zero frequency $s = 0$:

$$S := -\operatorname{tr}(G(0)) = -\operatorname{tr}(CA^{-1}B),$$

and quantifies the output amplification for constant input $u(t) = 1$.
Lastly, the following concept of symmetric (linear) control systems is essential for the cross-gramian-based model reduction presented in Chapter 3.

Definition 2.6 (Symmetric System)
*A linear control system is called **symmetric** if the system's transfer function is symmetric[9] [73]:*

$$G(s) = (G(s))^\intercal.$$

A single-input-single-output (SISO) system is always symmetric[10], since the associated transfer function is scalar and thus symmetric. A multiple-input-multiple-output (MIMO) system is symmetric if a symmetric matrix $J \in \mathbb{R}^{N\times N}$, $J = J^\intercal$ exists, such that

$$\begin{aligned} AJ &= JA^\intercal, \\ B &= JC^\intercal. \end{aligned} \tag{2.9}$$

In the special case of $J = \mathbb{1}_N$, such a system is called **state-space symmetric** and one has:

$$\begin{aligned} A &= A^\intercal, \\ B &= C^\intercal. \end{aligned} \tag{2.10}$$

[9] Equivalently, the symmetry of the impulse response $g(t) = g(t)^\intercal$, or the Markov parameter $CA^kB = (CA^kB)^\intercal, \forall k \in \mathbb{N}$, indicates a symmetric linear system.

[10] This holds also for nonlinear (SISO) systems, with the concept of a gradient system as nonlinear generalization to symmetric systems [191].

2.3. Model Reduction

Model reduction or **model order reduction** (MOR) can be defined as the algorithmic computation of a surrogate model, which exhibits similar dynamics but is of lower dimensionality compared to the original system. A model reduction method provides a **reduced order model** (ROM) of smaller dimension than the original **full order model** (FOM), and is expected to exhibit a small error compared to the original model's dynamics, or to conserve additional properties such as stability, passivity or conservation of energy. More recently, the scope of state-space reduction is broadened to parametric systems. **Parametric model order reduction** (pMOR) targets reduced order models that also preserve a parameter dependency of the FOM. In this work, model reduction encompasses the reduction of the state-space dimension (in the presence of parameters) and also a reduction of the parameter-space dimension $P = \dim(\theta)$. For the dimensions of the reduced order model's components, the following notation is used:

- Reduced State: $x_r(t)$, with: $n := \dim(x_r(t))$,
- Reduced Parameter: θ_r, with: $p := \dim(\theta_r)$.

For dynamical systems, reduced order modelling involves finding a low-dimensional subspace to the state-space,

$$\dim(x_r(t)) \ll \dim(x(t)),$$

that contains the dominant evolution of the trajectory $x(t)$,

$$\|x - x_r\| \ll 1.$$

In the context of control systems, model reduction also refers to the reduction of the state-space dimension $N = \dim(x(t))$, but with respect to the output dynamics $y(t)$. The dimension of the state-space is usually large compared to the dimension of the inputs and outputs:

$$\begin{aligned}\dim(x(t)) &\gg 1,\\ \dim(u(t)) &\ll \dim(x(t)),\\ \dim(y(t)) &\ll \dim(x(t)).\end{aligned}$$

For the evaluation of these input-output systems, the mapping from inputs to outputs $u \mapsto y$ is of prominent interest. It is then reasonable to ask, if for this mapping from the low-dimensional inputs $u(t)$ to the low-dimensional outputs $y(t)$, a low-dimensional approximate state trajectory $x_r(t)$,

$$\dim(x_r(t)) \ll \dim(x(t)),$$

may still yield acceptable output trajectories $y_r(t)$ compared to $y(t)$,

$$\|y - y_r\| \ll 1,$$

in a suitable norm.

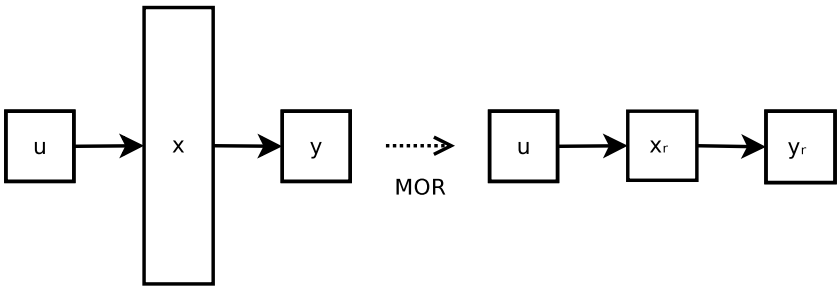


Figure 2.1.: *A cartoon illustration of model reduction for an input-output system.*

While the order reduction of a dynamical system aims to match the ROM for the whole state-space trajectory, a reduced order control system approximates the lower-dimensional output trajectory. Hence, model reduction methods for control system models take advantage of the presence of an output function. Schematically, the model reduction process is depicted in Figure 2.1.
pMOR aims to reduce the state-space in a manner such that the ROM remains valid over the whole (admissible) parameter-space Θ:

$$\|y(\theta) - y_r(\theta)\| \ll 1.$$

If the parameter-space dimension is large,

$$\dim(\theta) \gg 1,$$

an optimization or inference on the parameters constraint by the parametrized model becomes computationally expensive. A low-dimensional approximate parameter θ_r,

$$\dim(\theta_r) \ll \dim(\theta),$$

obtained from a parameter-space reduction, with a reconstruction $\theta \approx \rho(\theta_r)$, may still provide workable results,

$$\|y(\theta) - y(\rho(\theta_r))\| \ll 1,$$

yet for a lower computational cost.
Finally, the **combined state and parameter reduction** aims to concurrently reduce the state- and parameter-space dimensions. This combined reduction integrates the parametric state-space reduction with the parameter-space reduction to yield ROMs which approximate a highly parametrized FOM:

$$\|y(\theta) - y_r(\theta_r)\| \ll 1.$$

In a many-query setting like optimization, where the complexity depends on the parameter-space dimension, the combined reduction can accelerate these repeated evaluations.

2.3.1. Reduced Order Model

Generally, a reduced order model can be formulated based on a reduced state $x_r(t) \in \mathbb{R}^n$, a reduced vector field f_r, a reduced output functional g_r a and reduced parameter θ_r yielding the reduced order output $y_r(t)$:

$$\begin{aligned} \dot{x}_r(t) &= f_r(x_r(t), u(t), \theta_r), \\ y_r(t) &= g_r(x_r(t), u(t), \theta_r), \\ x_r(0) &= x_{r,0}. \end{aligned}$$

For the investigated methods, the input $u(t)$ remains unchanged and the reduced order output $y_r(t)$ retains the original model's dimension,

$$\dim(y_r(t)) = \dim(y(t)).$$

2.3.2. Projection-Based Model Reduction

A projection is an idempotent linear mapping. In this work, reduced order models are constructed by projecting the states (and parameters) to low-dimensional linear subspaces, which capture the dominant features of the enclosing spaces, defined by the underlying (control) system model, with respect to a some norm. Furthermore, it is assumed that all spaces considered for reduction are Hilbert spaces.
Projection-based model reduction is founded upon the approximation:

$$x_r(t) = V x(t),$$

using a reducing truncated projection V and a reconstructing truncated projection U,

$$x(t) \approx U x_r(t).$$

An oblique projection with bi-orthogonal matrices $U \in \mathbb{R}^{N \times n}$ and $V \in \mathbb{R}^{n \times N}$,

$$VU = \mathbb{1}_n,$$

is called **Petrov-Galerkin projection**. An orthogonal projection with an orthogonal matrix representation $U \in \mathbb{R}^{N \times n}$,

$$U^\intercal U = \mathbb{1}_n,$$

and $V := U^\intercal$ is called **Galerkin projection**[11], which can be seen as a special case of a Petrov-Galerkin projection.
For parametrized systems two variants of the projection-based approach can be considered [23, Sec. 4]; either several projections are constructed which act as local bases for a specific parameter subspace[12], or a single fixed projection representing a global base covering the whole parameter-space. In the work at hand the latter, global approach is exclusively considered.

[11] Also known as: Ritz-Galerkin projection or Galjorkin projection.
[12] See for example [172].

State Reduction

A framework for the projection-based state-space dimension reduction of a general control system near a steady-state $\bar{x}$ is given by [161]:

$$\begin{aligned} f_r &:= V f(\bar{x} + U x_r(t), u(t), \theta), \\ g_r &:= \quad g(\bar{x} + U x_r(t), u(t), \theta), \\ x_{0,r} &:= V(x_0 - \bar{x}), \end{aligned} \tag{2.11}$$

and will be used in the nonlinear model order reduction (nMOR) setting.
In the case of linear systems (with zero steady-state) the projection matrices can be directly applied to the control system components A, B, C and initial state x_0:

$$\begin{aligned} A_r &:= VAU, \\ B_r &:= VB, \\ C_r &:= CU, \\ x_{0,r} &:= V x_0. \end{aligned} \tag{2.12}$$

Parameter Reduction

For the parameter reduction, a Galerkin projection is employed to confine the parameters to their dominant subspace and with the inverse projection, the parameter is approximately reconstructed:

$$\begin{aligned} \theta_r &= \Pi\theta \\ \Rightarrow \theta &\approx \Pi^\intercal \theta_r. \end{aligned} \tag{2.13}$$

Combined Reduction

Finally, the combined state and parameter reduction is a combination of the state reduction from (2.11) and the parameter reduction from (2.13):

$$\begin{aligned} f_r &= V f(\bar{x} + U x_r(t), u(t), \Pi^\intercal \theta_r), \\ g_r &= \quad g(\bar{x} + U x_r(t), u(t), \Pi^\intercal \theta_r), \\ x_{0,r} &= V(x_0 - \bar{x}), \\ \theta_r &= \Pi\theta. \end{aligned} \tag{2.14}$$

At the heart of this work are two approaches for this combined reduction of state- and parameter-spaces applicable to nonlinear models. The combined reduction of states and parameters provides a ROM that can be especially useful for many-query settings such as optimization and inverse problems. While the parameter reduction reduces the complexity of the task itself, the state reduction accelerates the simulations for the individual locations in the parameter-space. The task of obtaining the state and parameter projections (numerically) is investigated in the following chapters.

2.4. Reduced Order Model Quality

The quality of a reduced order model is assessed by two families of norms [223, 62] that evaluate the model reduction error of the reduced compared to the original system. First, the signal norms, which quantify the output error signal $y_e := y - y_r$ between y and y_r in the time-domain,

$$\varepsilon_{\text{Time}} = \|y - y_r\| = \|y_e\|,$$

are considered and second, the system norms, which measure the error transfer function $G_e := G - G_r$ between G and G_r in the frequency-domain,

$$\varepsilon_{\text{Frequency}} = \|G - G_r\| = \|G_e\|.$$

2.4.1. Time-Domain Norms

In the time-domain, the Lebesgue norms $\|\cdot\|_{\mathrm{L}_p}$ for vector-valued signals are utilized. These L_p-norms measure certain properties of a time-series and are the basic tool for quantification of error signals. Of interest here are L_p-norms for $p \in \{1, 2, \infty\}$. The L_1-norm, or action, of a vector signal is defined by

$$\|y\|_{\mathrm{L}_1} := \int_0^\infty \|y(t)\|_1 \,\mathrm{d}t = \int_0^\infty \sum_{j=1}^{O} |y_i(t)| \,\mathrm{d}t,$$

the L_2-norm, or energy, of a vector signal is given by

$$\|y\|_{\mathrm{L}_2} := \sqrt{\int_0^\infty \|y(t)\|_2^2 \,\mathrm{d}t} = \sqrt{\int_0^\infty \sum_{j=1}^{O} y_i^2(t) \,\mathrm{d}t},$$

and the L_∞-norm of a vector signal in the time-domain is the peak component over time:

$$\|y\|_{\mathrm{L}_\infty} := \operatorname*{ess\,sup}_{t\in[0,\infty)} \|y(t)\|_\infty = \operatorname*{ess\,sup}_{t\in[0,\infty)} \max_j |y_j(t)|.$$

The previously introduced time-domain norms are related through Hölder's inequality:

$$\|y\|_{\mathrm{L}_2}^2 \leq \|y\|_{\mathrm{L}_1} \|y\|_{\mathrm{L}_\infty}.$$

For discrete output trajectories y_h over T time steps of width Δt, discrete variants of the time-domain norms $\|\cdot\|_{\ell_p}$ are given by:

$$\begin{aligned}
\|y_h\|_{\ell_1} &:= \Delta t \sum_{i=1}^{T} \sum_{j=1}^{O} |y_{h,j}(t_i)|, \\
\|y_h\|_{\ell_2} &:= \sqrt{\Delta t \sum_{i=1}^{T} \sum_{j=1}^{O} (y_{h,j}(t_i))^2}, \\
\|y_h\|_{\ell_\infty} &:= \max_{t_i} \max_j |y_{h,j}(t_i)|.
\end{aligned} \tag{2.15}$$

2.4.2. Joint State and Parameter Norms

Errors for parametric model order reduction and combined state and parameter reduction are assessed in joint state- and parameter-space norms. The employed norms are related to the joint measures introduced in [16, Sec. 5.1]. These joint norms can be constructed as a composition of a parameter-space L_q-norm "after" the time-domain L_p-norm [18]. Thus, for all elements of the parameter-space, the output is measured in the L_p-norm, then over all output norms the L_q-norm yields the joint state and parameter norm:

$$\|y(\theta)\|_{\mathrm{L}_p \otimes \mathrm{L}_q} := (\|\cdot\|_{\mathrm{L}_q} \circ \|\cdot\|_{\mathrm{L}_p})(y(\theta)).$$

In the scope of this work the following four joint state and parameter norms are utilized:

$$\begin{aligned}
\|y(\theta)\|_{\mathrm{L}_1 \otimes \mathrm{L}_2} &= \sqrt{\int_\Theta \|y(\theta)\|_{\mathrm{L}_1}^2 \,\mathrm{d}\theta}, \\
\|y(\theta)\|_{\mathrm{L}_2 \otimes \mathrm{L}_2} &= \sqrt{\int_\Theta \|y(\theta)\|_{\mathrm{L}_2}^2 \,\mathrm{d}\theta}, \\
\|y(\theta)\|_{\mathrm{L}_\infty \otimes \mathrm{L}_2} &= \sqrt{\int_\Theta \|y(\theta)\|_{\mathrm{L}_\infty}^2 \,\mathrm{d}\theta}, \\
\|y(\theta)\|_{\mathrm{L}_2 \otimes \mathrm{L}_\infty} &= \sup_{\theta \in \Theta} \|y(\theta)\|_{\mathrm{L}_2}.
\end{aligned}$$

Practically, the joint norm cannot be computed for all elements of the parameter-space. Hence, some representative discrete subspace $\widetilde{\Theta}$ of the parameter-space has to be selected, either, a predefined grid or a number of random samples with suitable statistics. This leads to the discrete variants of the previous four joint norms:

$$\|y_h(\theta)\|_{\ell_1 \otimes \ell_2} = \sqrt{\sum_{k=1}^{|\widetilde{\Theta}|} (\Delta t \sum_{i=1}^{T} \sum_{j=1}^{O} |y_{h,j}(t_i; \tilde{\theta}_k)|)^2}, \tag{2.16a}$$

$$\|y_h(\theta)\|_{\ell_2 \otimes \ell_2} = \sqrt{\sum_{k=1}^{|\widetilde{\Theta}|} \Delta t^2 \sum_{i=1}^{T} \sum_{j=1}^{O} (y_{h,j}(t_i; \tilde{\theta}_k))^2}, \tag{2.16b}$$

$$\|y_h(\theta)\|_{\ell_\infty \otimes \ell_2} = \sqrt{\sum_{k=1}^{|\widetilde{\Theta}|} (\max_{t_i} \max_{j} |y_{h,j}(t_i; \tilde{\theta}_k)|)^2}, \tag{2.16c}$$

$$\|y_h(\theta)\|_{\ell_2 \otimes \ell_\infty} = \max_{\tilde{\theta} \in \widetilde{\Theta}} \sqrt{\Delta t \sum_{i=1}^{T} \sum_{j=1}^{O} (y_{h,j}(t_i; \tilde{\theta}))^2}, \tag{2.16d}$$

which are computable numerically.

2.4.3. Frequency-Domain Norms

In the frequency-domain, the Hardy-norms $\|\cdot\|_{\mathscr{H}_p}$ of the transfer function for $p \in \{2, \infty\}$ are employed. The $\mathscr{H}_p$-norms measure the magnitude of the frequency response, and are also called system norms. Generally, the Hardy-norms are the Schatten norms of the transfer function constrained to positive frequencies $\omega > 0$; hence, the $\mathscr{H}_2$-norm is defined as:

$$\|G\|_{\mathscr{H}_2} := \sqrt{\int_0^\infty \operatorname{tr}(G(-\imath\,\omega)^* G(\imath\,\omega))\,\mathrm{d}\omega}.$$

As a result of Parseval's theorem [6, Ch. 5.5.1], the $\mathscr{H}_2$-norm is equal to the L_2-norm of the impulse response,

$$\|G\|_{\mathscr{H}_2} = \|g\|_{\mathrm{L}_2},$$

and, by Young's inequality (Appendix A.1), a bound for the L_∞-norm in the time-domain:

$$\begin{aligned} \|y\|_{\mathrm{L}_\infty} &= \|g * u\|_{\mathrm{L}_\infty} \\ &\leq \|g\|_{\mathrm{L}_2} \|u\|_{\mathrm{L}_2} = \|G\|_{\mathscr{H}_2} \|u\|_{\mathrm{L}_2}. \end{aligned} \tag{2.17}$$

The $\mathscr{H}_\infty$-norm is defined as the maximum singular value of the transfer function G,

$$\begin{aligned} \|G\|_{\mathscr{H}_\infty} &:= \sup_{\omega \in \mathbb{R}^{>0}} \sigma_{\max}(G(\imath\,\omega)) \\ &= \sup_{\|u\|_{\mathrm{L}_2}=1} \|y\|_2, \end{aligned}$$

which is equivalent to the L_2-gain of the system. This leads to a bound for the L_2-norm in the time-domain,

$$\|y\|_{\mathrm{L}_2} \leq \|G\|_{\mathscr{H}_\infty} \|u\|_{\mathrm{L}_2}. \tag{2.18}$$

As presented in [16], compound frequency- and parameter-space norms could also be used such as $\|\cdot\|_{\mathscr{H}_2 \otimes \mathrm{L}_2}$, but are not considered in the work at hand.
For nonlinear systems only the time-domain norms can be applied directly, while the frequency-domain norms would require a linearization of the system; nonetheless, especially in the case of the gramian-based state-space reduction, the previous norms can be utilized for error bounds and error indicators.
Lastly, the system L_1-norm is presented in this context:

$$\|g\|_{\mathrm{L}_1} := \max_i \sum_{j=1}^{O} \Big(\int_0^\infty |g(t)|\,\mathrm{d}t \Big)_{ij} = \| \int_0^\infty |g(t)|\,\mathrm{d}t \|_\infty, \tag{2.19}$$

which corresponds to the L_∞-gain of the system (2.8) and leads to a time-domain bound in the L_1-norm (Appendix A.1):

$$\|y\|_{\mathrm{L}_1} = \|g * u\|_{\mathrm{L}_1} \leq \|g\|_{\mathrm{L}_1} \|u\|_{\mathrm{L}_1}.$$

2.5. Principal Axis Transformation

A fundamental mathematical building block of model reduction is the availability of a transformation which allows to sort a model's components by their importance. For the subspace of symmetric matrices the existence of such a principal axis transformation is guaranteed by the following theorem.

Theorem 2.7 (Principal Axis Theorem)
For every real symmetric matrix there exists an orthogonal base of eigenvectors.

Proof. See for example [84, Ch. 3.8.7]. □

An eigendecomposition of a symmetric matrix yields the eigenvectors as dominant "directions", which can be sorted by their associated eigenvalues[13] and at the same time embody an orthogonal basis, and thus uncorrelated elements, over the Hilbert space $\mathbb{R}^N$ equipped with the Euclidean inner product. Since this result is only valid for symmetric matrices, a generalization to arbitrary matrices is mentioned next.

2.5.1. Singular Value Decomposition

For a non-symmetric matrix $X \in \mathbb{R}^{N\times M}$ the product with its transpose $XX^\intercal$ is symmetric, to which the previous Theorem 2.7 can be applied. The same holds for $X^\intercal X$. A matrix factorization that yields the (orthogonal) eigenvectors of $XX^\intercal$ and $X^\intercal X$ is the singular value decomposition (SVD).

Theorem 2.8 (Singular Value Decomposition)
For a matrix $X \in \mathbb{R}^{N\times M}$ there exists orthogonal matrices $U \in \mathbb{R}^{N\times N}$, $V \in \mathbb{R}^{M\times M}$ and a diagonal matrix $D \in \mathbb{R}^{N\times M}$ such that:

$$X = UDV^\intercal,$$

with the diagonal entries of D corresponding to the singular values σ of X:

$$D_{ii} = \sigma_i = \sqrt{\lambda_i(XX^\intercal)},$$

thus the dyadic decomposition with columns u_i of U and v_i of V:

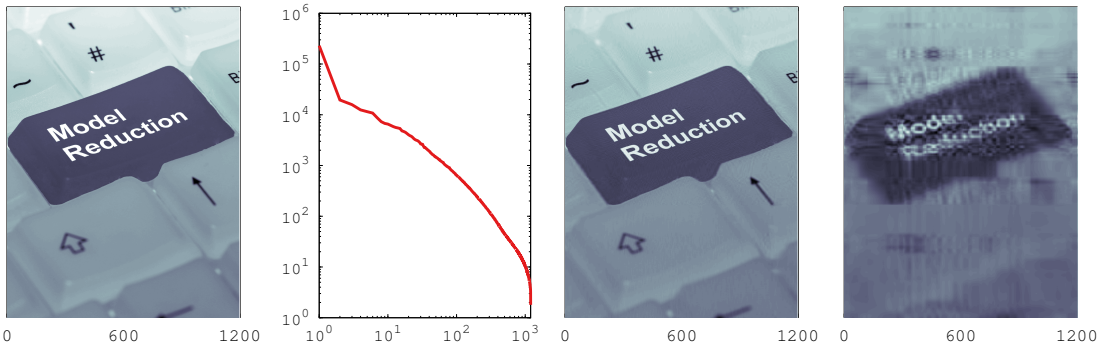
$$X = \sum_{i=1}^{N} u_i \sigma_i v_i^\intercal, \tag{2.20}$$

*is a **singular value decomposition**.*

Proof. See for example [6, Ch. 3.2.1]. □

The eigenvectors of $XX^\intercal$ are thus given by the left singular vectors u_i and those of $X^\intercal X$ by the right singular vectors v_i to which the singular values $\sigma_i > 0$, as the square roots of the eigenvalues of $XX^\intercal$, correspond.

[13] The eigenvalues of a real symmetric matrix are real.



(a) *Original Image*[15]. (b) *Singular Values.* (c) *Rank 100.* (d) *Rank 10.*

Figure 2.2.: *SVD-based dimension reduction of a digital image of dimension* 1200×1600.

Dimension Reduction

Without loss of generality the left and right singular vectors u_i, v_i are usually assumed to be sorted by their corresponding singular values σ_i in descending order. The truncation of the sum (2.20) to $n < N$ summands results in a low rank approximation $\widetilde{X}$ to X, excluding the components with small singular values. This SVD-based dimension reduction[15] is illustrated in Figure 2.2, by compressing a grayscale image.

Proper Orthogonal Decomposition

Given a dataset $X = [x_1, \ldots, x_M] \in \mathbb{R}^{N \times M}$, $x_i \in \mathbb{R}^N$, an eigendecomposition of the associated autocorrelation matrix $R := \mathbb{E}[XX^\intercal]$,

$$R\lambda_i = \lambda_i w_i \Leftrightarrow X = W \Lambda W^{-1},$$

yields the (finite dimensional) proper orthogonal decomposition (POD) of X, which is equivalent to the SVD of X and is also called Karhunen-Lóeve transformation. POD is a basic tool to construct Galerkin projections as the left singular vectors of a discrete time-series [205] obtained from numerical simulations and is also called method of snapshots. Balanced POD is a variant of this method taking into account adjoint information [120].

Principal Component Analysis

The eigendecomposition of the covariance matrix $C := \mathbb{E}[(X - \mathbb{E}[X])^\intercal (X - \mathbb{E}[X])]$ of the aforementioned dataset X,

$$C\lambda_i = \lambda_i w_i \Leftrightarrow (X - \mathbb{E}[X]) = W \Lambda W^{-1},$$

results in the principal component analysis (PCA) of X. This method is also called empirical eigenfunctions or empirical orthogonal functions. A nonlinear extension to the PCA is the kernel principal component analysis (KPCA) developed in [193].

[15] See `code/ch2/dimred.m` in the supplementary source code archive (Appendix B.1).

[15] Image by: Nina-Claire Himpe, 2015; licensed under CC-BY.

3. Gramian-Based Combined Reduction

Contents

Gramian-based model reduction is a system-theoretic approach, developed originally in [168] for the state-space reduction of linear control systems. The nonlinear state-space reduction, parameter-space reduction, and consequently the combined reduction, are based upon the theory for linear state reduction. Central to this approach are two attributes of control systems, **controllability** and **observability**; the associated gramian matrices are instrumental to this SVD-based model reduction technique. Special focus is placed on the (empirical) cross gramian matrix, which has theoretical and numerical favorable properties.

3.1. State Reduction

Gramian-based state-space reduction aims to confine the dynamics of a system to a low-dimensional subspace $\mathbb{R}^n \subset \mathbb{R}^N, n \ll N$ based on its input-output behavior [17]. The output $y(t)$ of a control system (2.6) at a time $t \in \mathbb{R}^{>0}$, for an initial state x_0 and a squarely integrable input function $u(t)$ is given by:

$$y(t) = C\,\mathrm{e}^{At}\,x_0 + \int_0^\infty C\,\mathrm{e}^{A(t-\tau)}\,Bu(\tau)\,\mathrm{d}\tau.$$

More generally, a stable control system maps the input or control $u \in \mathrm{L}_2^M[0,\infty)$ to the output $y \in \mathrm{L}_2^O[0,\infty)$. For a linear control system with zero initial state, this mapping from inputs to outputs is described by an operator S,

$$u : \mathrm{L}_2^M[0,\infty) \xrightarrow{\;S\;} y : \mathrm{L}_2^O[0,\infty),$$

which is given by a convolution of the impulse response g with the input function u as in (2.7) for a zero initial state,

$$y(t) = S(u)(t) = (g * u)(t) = \int_0^\infty g(t-\tau)u(\tau)\,\mathrm{d}\tau, \tag{3.1}$$

Yet, the convolution operator[16] S cannot be assumed to be of finite rank [6, Ch. 5].

[16] Also known as: evolution operator.

The composition of the operator S with a "time-flip"-operator [90] $F : [0, \infty) \to (-\infty, 0]$, $F(u)(t) = u(-t)$,

$$H := S \circ F, \tag{3.2}$$

yields the related **Hankel operator**,

$$\begin{aligned} H(u)(t) &= \int_{-\infty}^{0} g(t-\tau)u(\tau)\,\mathrm{d}\tau \\ &= \int_{-\infty}^{0} C\,\mathrm{e}^{A(t-\tau)}Bu(\tau)\,\mathrm{d}\tau \\ &= C\,\mathrm{e}^{At}\int_{0}^{\infty}\mathrm{e}^{A\tau}Bu(-\tau)\,\mathrm{d}\tau, \end{aligned}$$

which maps past inputs[17] $F(u)$ to future outputs y [52]. This means, the current state $x(0)$ is the result of the past inputs $F(u)$,

$$x(0) = \int_{\infty}^{0}\mathrm{e}^{-A\tau}Bu(\tau)\,\mathrm{d}\tau;$$

and the future outputs $y(t)$ for the current state $x(0)$ are given by:

$$y(t) = C\,\mathrm{e}^{At}\,x(0).$$

Thus, if the underlying system is stable, the Hankel operator is decomposable into a mapping from (past) inputs to states, $\mathscr{C} : \mathrm{L}_2(-\infty, 0] \to \mathbb{R}^N$, and a mapping from states to (future) outputs, $\mathscr{O} : \mathbb{R}^N \to \mathrm{L}_2[0, \infty)$, see Figure 3.1.

$$\begin{array}{ccc} \mathrm{L}_2^M(-\infty, 0] & \xrightarrow{\;H\;} & \mathrm{L}_2^O[0, \infty) \\ \mathscr{C}\searrow & & \nearrow\mathscr{O} \\ & \mathbb{R}^N & \end{array}$$

Figure 3.1.: *Commutative diagram illustrating the action of the Hankel operator mapping past inputs to future outputs.*

A stable system matrix A implies the L_2-stability of the system and thus $\|G\|_{\mathscr{H}_\infty} < \infty$. Then, following [74, Ch. 5, Thm. 1], the composition of the surjective operator $\mathscr{C}$ with the injective operator $\mathscr{O}$,

$$\mathscr{O} \circ \mathscr{C}\ (= H),$$

has finite rank N and hence the Hankel operator has finite rank less or equal to N.

[17] This assumes future inputs are zero.

Hankel Singular Values

Given a finite-dimensional system $N < \infty$, the Hankel operator exposes properties which provide the foundation for gramian-based model reduction. For a stable system, the associated Hankel operator is compact due to its finite rank. Since compact operators are bounded, they admit a singular value decomposition; and the Hankel operator is a Hilbert-Schmidt operator as its Frobenius norm, the finite sum of squared singular values, is finite:

$$\begin{aligned}\|H\|_F &= \operatorname{tr}(H^*H) \\ &= \sum_{i=1}^{N} \sigma_i^2(H) < \infty.\end{aligned}$$

The singular values of the Hankel operator $\sigma_i(H)$ are a system invariant (independent from the state-space coordinate system). These **Hankel singular value**s (HSV) classify the states by importance in terms of energy transfer to the overall system[18]. This measure of coherence between system inputs and outputs motivates model reduction by excluding states with corresponding small HSVs which transfer the least energy. Additionally, the Hankel operator is also a nuclear operator, since its trace norm is finite:

$$\begin{aligned}\|H\|_* &= \operatorname{tr}((H^*H)^{\frac{1}{2}}) \\ &= \sum_{i=1}^{N} \sigma_i(H) < \infty.\end{aligned}$$

This property will be important in Section 3.2.4, since it enables the computation of upper bounds for the model reduction error.

Hankel Norm

The Hankel operator induces a norm which can provide a lower error bound for the reduced system. For a linear state-space system $\Sigma(A, B, C)$, the **Hankel norm** is defined as the maximum singular value[19] of the Hankel operator,

$$\|\Sigma\|_H := \max_{i=1\ldots N} (\sigma_i(H)).$$

As the following gramian-based model reduction approach is based upon these HSVs, a ROM of rank n given by $\Sigma_r(A_r, B_r, C_r)$ would exhibit at least an error ε_{MOR}:

$$\varepsilon_{MOR} \geq \sigma_{n+1}(H) = \|\Sigma - \Sigma_r\|_H.$$

This can be seen as a lower bound on the model reduction error [86, 87] for the subsequent gramian-based methods, which is the result of the famous Adamjan-Arov-Krein theorem [2, Thm. 1.1].

[18] HSVs are without loss of generality assumed to be sorted in descending order $\sigma_1 \geq \cdots \geq \sigma_N$.
[19] Also known as: Schatten-∞ norm.

3.2. System Gramians

Essential information on a control system can be encoded in certain gramian matrices. A gramian matrix[20] W to a given matrix $V \in \mathbb{R}^{N\times M}$ is defined as [84, Ch. 8.6.1]:

$$W := VV^\mathsf{T},$$

which is symmetric and positive semi-definite and thus the eigenvalues of W are real and non-negative, $\lambda(W) \in \mathbb{R}^{\geq 0}$. For $V = \begin{pmatrix} v_1 & \dots & v_N \end{pmatrix}^\mathsf{T}$, the gramian matrix can also be written as the matrix of all inner products of row vectors $v_i \in \mathbb{R}^M$,

$$W_{ij} = \langle v_i, v_j \rangle.$$

Generally, a gramian matrix can be computed for a given set of vector-valued functions.

Definition 3.1 (Gramian Matrix)
For a set of functions $\{v_i \in \mathrm{L}_2^N[0,\infty)\}_{i=1\dots M}$ *an associated* ***gramian matrix*** $W \in \mathbb{R}^{N\times N}$ *is defined as the inner product of all combinations of the set's elements:*

$$W_{ij} := \langle v_i, v_j \rangle.$$

Condensing the set of vector-valued functions $\{v_i\}_{i=1\dots M}$ into a matrix-valued function $V(t) = \begin{pmatrix} v_1(t) & \dots & v_M(t) \end{pmatrix}$ yields the following representation of a gramian matrix:

$$W = \int_0^\infty V(t)V^\mathsf{T}(t)\,\mathrm{d}t.$$

The gramian matrices associated with control systems, the so-called **system gramians**, encode information on attributes of the states. These system gramians enable the model order reduction since they relate to the HSVs through the previously introduced operators $\mathscr{C}$ and $\mathscr{O}$. Following, three system gramians[21] are presented, namely:

- the controllability gramian,
- the observability gramian,
- and the cross gramian.

While the controllability gramian and the observability gramian are self-adjoint, the cross gramian is generally not a gramian matrix in the sense of Definition 3.1, but rather a cross-covariance matrix, yet it was introduced under the label "gramian" [65] and is designated throughout this work as "cross gramian" to avoid confusion. Over the course of this chapter the cross gramian is recurringly of particular interest and thus investigated more closely in the linear system theory context of this section.

[20] Also known as: grammian or Gram matrix.

[21] The impulse response gramian and the system's gram matrix [212] can also be considered system gramians but are not investigated in this work.

3.2.1. Controllability Gramian

Controllability quantifies how well the states $x(t)$ of a (control) system are driven by external input $u(t)$. More specifically, a controllable system guarantees that from any location in the state-space a steady-state $\bar{x}$ is attained in finite time. A stronger concept to controllability is reachability, which certifies that from a steady-state any location in the state-space can be reached in finite time. A weaker concept to controllability is stabilizability and warrants that a steady-state is reached asymptotically.

Definition 3.2 (Controllability, Reachability, Stabilizability)

- ***Controllability**: A system is called controllable if for any state $\tilde{x} \in \mathbb{R}^N$ there exists an input function $u : [0, \mathscr{T}] \to \mathbb{R}^M$, $\mathscr{T} < \infty$, such that $x(0) = \tilde{x}$ and $x(\mathscr{T}) = \bar{x}$.*
- ***Reachability**: A system is called reachable if for any state $\tilde{x} \in \mathbb{R}^N$ there exists an input function $u : [0, \mathscr{T}] \to \mathbb{R}^M$, $\mathscr{T} < \infty$, such that $x(0) = \bar{x}$ and $x(\mathscr{T}) = \tilde{x}$.*
- ***Stabilizability**: A system is called stabilizable if all uncontrollable subsystems are (asymptotically) stable.*

Reachability implies controllability [105, Ch. 11] which in turn implies stabilizability [105, Ch. 14] and for continuous LTI systems controllability and reachability are equivalent [6, Thm. 4.18]. The associated operator to evaluate controllability maps past inputs to current states. Hence, the **controllability operator** $\mathscr{C} : \mathrm{L}_2^M(-\infty, 0] \to \mathbb{R}^N$ is defined as:

$$\mathscr{C}(u) := \int_{-\infty}^{0} \mathrm{e}^{-At} B u(t)\,\mathrm{d}t = \int_{0}^{\infty} \mathrm{e}^{At} B u(-t)\,\mathrm{d}t. \tag{3.3}$$

The adjoint operator $\mathscr{C}^* : \mathbb{R}^N \to \mathrm{L}_2^M(-\infty, 0]$ is then given by:

$$\mathscr{C}^*(z)(t) = B^\mathsf{T} \mathrm{e}^{A^\mathsf{T} t} z.$$

Now, a gramian matrix for $\mathscr{C}$ can be defined.

Definition 3.3 (Controllability Gramian)
*For an asymptotically stable LTI system the **controllability gramian** [22] W_C is given by:*

$$W_C := \mathscr{C} \circ \mathscr{C}^* = \int_{0}^{\infty} \mathrm{e}^{At} B B^\mathsf{T} \mathrm{e}^{A^\mathsf{T} t}\,\mathrm{d}t.$$

If the controllability gramian W_C has full rank, the associated system is controllable. The system gramian W_C can be computed as the solution to a matrix equation.

Lemma 3.4 (Controllability Gramian)
The controllability gramian satisfies the Lyapunov equation:

$$AW_C + W_C A^\mathsf{T} = -BB^\mathsf{T}.$$

Proof. See proof of Lemma 3.9

[22] This work uses the term "controllability" instead of "reachability" due to the use in [168, 65, 142].

3.2.2. Observability Gramian

Observability quantifies how well the output $y(t)$ of a control system reflects the state $x(t)$. Essentially, reconstructability[23] of a system guarantees that any initial state x_0 in the state-space trajectory can be determined from the output in finite time. A stronger concept to reconstructability is observability, which certifies that any state can be determined from previous output after finite time. A weaker concept to reconstructability is detectability and warrants that the initial state is retrieved asymptotically.

Definition 3.5 (Reconstructability, Observability, Detectability)

- ***Reconstructability**: A system is called reconstructable if an initial state x_0 is uniquely determined by the output $y(t) \in \mathbb{R}^O$ on a finite time interval $[0, \mathcal{T}]$.*
- ***Observability**: A system is called observable if any state $x(\mathcal{T})$ is uniquely determined by the output $y(t) \in \mathbb{R}^O$ on a finite time interval $[0, \mathcal{T}]$.*
- ***Detectability**: A system is called detectable if all unobservable subsystems are (asymptotically) stable.*

Observability implies reconstructability [105, Ch. 15] which in turn implies detectability [105, Ch. 16] and for continuous LTI systems observability and reconstructability are equivalent [6, Thm. 4.18]. The associated operator to evaluate observability maps current states to future outputs. Hence, the **observability operator** $\mathcal{O} : \mathbb{R}^N \to \mathrm{L}_2^O$ is defined as:

$$\mathcal{O}(x_0)(t) := C\,\mathrm{e}^{At}\,x_0. \tag{3.4}$$

The adjoint operator $\mathcal{O}^* : \mathrm{L}_2^O \to \mathbb{R}^N$ is then given by:

$$\mathcal{O}^*(y) = \int_0^\infty \mathrm{e}^{A^\intercal t}\,C^\intercal y(t)\,\mathrm{d}t.$$

Again, also a gramian matrix for $\mathcal{O}$ can be defined.

Definition 3.6 (Observability Gramian)
*For an asymptotically stable LTI system the **observability gramian** W_O is given by:*

$$W_O := \mathcal{O}^* \circ \mathcal{O} = \int_0^\infty \mathrm{e}^{A^\intercal t}\,C^\intercal C\,\mathrm{e}^{At}\,\mathrm{d}t.$$

If the observability gramian W_O has full rank, the associated system is observable. The system gramian W_O can also be computed as the solution to a matrix equation.

Lemma 3.7 (Observability Gramian)
The observability gramian satisfies the Lyapunov equation:

$$A^\intercal W_O + W_O A = -C^\intercal C.$$

Proof. See proof of Lemma 3.9

[23] Also known as: constructability.

3.2.3. Cross Gramian

The cross gramian encodes controllability and observability into a single matrix. It was introduced for SISO systems in [65][24], and extended to (symmetric) MIMO systems in [145, 68, 69]. This system "gramian" requires that the number of inputs is equal to the number of outputs, $M = O$; a system with such property is called **square**.

Definition 3.8 (Cross Gramian)
For a square, asymptotically stable LTI system the ***cross gramian***[25]*, W_X is given by the composition of the controllability operator with the observability operator:*

$$W_X := \mathscr{C} \circ \mathscr{O} = \int_0^\infty \mathrm{e}^{At} BC \, \mathrm{e}^{At} \, \mathrm{d}t. \tag{3.5}$$

Iff the cross gramian has full rank, the associated (symmetric) system is controllable and observable and thus minimal [64, 69]. The cross gramian W_X can be computed as the solution to a matrix equation.

Lemma 3.9 (Cross Gramian)
The cross gramian satisfies the Sylvester equation:

$$AW_X + W_X A = -BC.$$

Proof.
As a preliminary remark it should be noted that A is negative definite and thus, first, A is invertible, and second $\lim_{t\to\infty} \mathrm{e}^{At} = 0$. Then, from integration by parts follows:

$$\begin{aligned}
&\int_0^\infty \mathrm{e}^{At} BC \, \mathrm{e}^{At} \, \mathrm{d}t = A^{-1} \mathrm{e}^{At} BC \, \mathrm{e}^{At} \Big|_0^\infty - A^{-1} \int_0^\infty \mathrm{e}^{At} BC \, \mathrm{e}^{At} A \mathrm{d}t \\
\Rightarrow\ & A \int_0^\infty \mathrm{e}^{At} BC \, \mathrm{e}^{At} \, \mathrm{d}t + \int_0^\infty \mathrm{e}^{At} BC \, \mathrm{e}^{At} \, \mathrm{d}t A = \mathrm{e}^{At} BC \, \mathrm{e}^{At} \Big|_0^\infty \\
\Rightarrow\ & AW_X + W_X A = -BC.
\end{aligned}$$

□

The core property of the cross gramian [64, 145, 67, 1, 210] is given by its relation to the controllability and observability gramian, which only holds for symmetric systems (2.9).

Lemma 3.10 (Cross Gramian)
For a symmetric linear system the squared cross gramian equals the product of controllability and observability gramian,

$$W_X^2 = W_C W_O.$$

Proof (First Variant).
A symmetric system implies a symmetric Hankel operator: $H = H^* \Rightarrow H = \mathscr{O}\mathscr{C} = (\mathscr{O}\mathscr{C})^*$, and thus:

$$W_X^2 = \mathscr{C}\mathscr{O}\mathscr{C}\mathscr{O} = \mathscr{C}(\mathscr{O}\mathscr{C})^*\mathscr{O} = \mathscr{C}\mathscr{C}^*\mathscr{O}^*\mathscr{O} = W_C W_O.$$

□

[24] Even though [64] is published earlier, it references [65] already, see also [63].
[25] Other symbols used for the cross gramian are W_{CO} and X_{CG}.

Proof (Second Variant).

$$\begin{aligned}
W_C W_O &= \int_0^\infty \int_0^\infty \mathrm{e}^{At}\, BB^\intercal\, \mathrm{e}^{A^\intercal t}\, \mathrm{e}^{A^\intercal \tau}\, C^\intercal C\, \mathrm{e}^{A\tau}\, \mathrm{d}t\, \mathrm{d}\tau \\
&= \int_0^\infty \int_0^\infty \mathrm{e}^{At}\, BCJ\, \mathrm{e}^{A^\intercal t}\, \mathrm{e}^{A^\intercal \tau}\, J^{-1}BC\, \mathrm{e}^{A\tau}\, \mathrm{d}t\, \mathrm{d}\tau \\
&= \int_0^\infty \int_0^\infty \mathrm{e}^{At}\, BCJ\, \mathrm{e}^{A^\intercal (t+\tau)}\, J^{-1}BC\, \mathrm{e}^{A\tau}\, \mathrm{d}t\, \mathrm{d}\tau \\
&= \int_0^\infty \int_0^\infty \mathrm{e}^{At}\, BCJ\, \mathrm{e}^{J^{-1}AJ(t+\tau)}\, J^{-1}BC\, \mathrm{e}^{A\tau}\, \mathrm{d}t\, \mathrm{d}\tau \\
&= \int_0^\infty \int_0^\infty \mathrm{e}^{At}\, BC\, \mathrm{e}^{JJ^{-1}AJJ^{-1}(t+\tau)}\, BC\, \mathrm{e}^{A\tau}\, \mathrm{d}t\, \mathrm{d}\tau \\
&= \int_0^\infty \int_0^\infty \mathrm{e}^{At}\, BC\, \mathrm{e}^{At}\, \mathrm{e}^{A\tau}\, BC\, \mathrm{e}^{A\tau}\, \mathrm{d}t\, \mathrm{d}\tau = W_X^2.
\end{aligned}$$

□

As a consequence of the previous Lemma 3.10 the following relation between the system's gramians holds [65]:

Corollary 3.11
The absolute value of the cross gramian's eigenvalues are equal to the square root of the eigenvalues of the product of controllability and observability gramians,

$$W_X^2 = W_C W_O \Rightarrow |\lambda_i(W_X)| = \sqrt{\lambda_i(W_C W_O)}, \quad i = 1 \dots N. \tag{3.6}$$

Furthermore, a relation[26] between the trace of the cross gramian and the system gain,

$$\operatorname{tr}(W_X) = -\frac{1}{2} \operatorname{tr}(CA^{-1}B),$$

was shown first for SISO systems in [65] and generalized for MIMO systems in [110], while a more general result is presented in [85]. The cross gramian also provides additional information on the underlying system [115], such as the system's transfer function's Cauchy-index through the cross gramian's signature [66], the system's singularity index [164] using the virtual system $\Sigma(W_X, B, C)$ and the cross gramian minimum information loss (CGMIL) index [78].
For the special case of state-space symmetric systems (2.10) another useful relation [155] connects the controllability, observability and cross gramian.

Note 3.12 (State-Space Symmetric System Gramians)
For state-space symmetric systems the controllability gramian, observability gramian and cross gramian are equal:

$$A = A^\intercal \wedge B = C^\intercal \Rightarrow W_C = W_O = W_X. \tag{3.7}$$

This is a direct consequence of the definition of the system gramians.

[26] This result is used in [214, 215] for a parametrized system as a sensitivity measure $S(\theta) := \operatorname{tr}(W_X(\theta))$ in the context of sensitivity analysis, in decentralized control as an interaction measure [197] and in optimal sensor placement [160].

Controllability-Based Cross Gramian

Controllability and observability are dual operators [6, Ch. 4.2.3]. The controllability gramian of a linear system is equal to the observability gramian of the adjoint system, and vice versa, the observability gramian of the system is equal to the controllability gramian of the adjoint system. For linear control systems, the cross gramian can thus be computed using only the controllability gramian as introduced in [69] and revisited in [195, 18]. Augmenting a linear system's vector field with its negative adjoint yields:

$$\dot{\breve{x}}(t) = \begin{pmatrix} A & 0 \\ 0 & A^\intercal \end{pmatrix} \breve{x}(t) + \begin{pmatrix} B \\ C^\intercal \end{pmatrix} \breve{u}(t).$$

Then, the controllability gramian of such a system has the following block structure:

$$\breve{W}_C = \begin{pmatrix} W_C & W_X \\ W_X^\intercal & W_O \end{pmatrix},$$

in which the cross gramian W_X constitutes the upper right block.

Non-Symmetric Cross Gramian

The core property of the cross gramian (3.6) is only available for symmetric systems. Yet, the cross gramian can be computed for any square system and [14] shows that even for non-symmetric systems, heuristically the cross gramian can yield usable results. For an arbitrary square system the following weaker relation between the cross gramian and the HSVs is shown in [210].

Lemma 3.13 (Non-Symmetric Cross Gramian Singular Value Properties)
For a non-symmetric square system the following inequalities for the singular values of the cross gramian $\sigma_i(W_X)$ and the Hankel singular values $\sigma_i(H)$ hold:

$$\sum_{i=1}^{k} \sigma_i(H) \geq \sum_{i=1}^{k} \sigma_i(W_X) \wedge \sum_{i=1}^{k} \sigma_{N-i+1}(W_X) \geq \sum_{i=1}^{k} \sigma_{N-i+1}(H).$$

There are three methods for computing a cross gramian for non-symmetric systems. First, the class of systems for which the cross gramian's core property holds can be expanded to orthogonally symmetric systems [1]. Second, by embedding a non-symmetric system into a symmetric system of which its cross gramian is then an approximation [209, 210]. Third, an averaged cross gramian over all SISO subsystems is proposed in [117].
A generalization to symmetric systems (see: Definition 2.6) are orthogonally symmetric systems[27].

Definition 3.14 (Orthogonally Symmetric Systems)
A linear system is called ***orthogonally symmetric*** *if:*

$$\exists! P = P^\intercal, \quad \exists U \begin{cases} \in \mathbb{R}^{M \times O} & O \leq M \\ \in \mathbb{R}^{O \times M} & M \leq O \end{cases}, \quad U^\intercal U = \mathbb{1},$$

[27] The use of an (orthogonal) output projection in [120, Ch. 5.3] is a related method.

such that:

$$\begin{aligned} AP &= PA^\intercal, \\ B &= PCU^\intercal, \\ C &= PBU. \end{aligned}$$

In case of an orthogonally symmetric system a cross gramian can be defined that satisfies the core property Lemma 3.10.

Definition & Theorem 3.15 (Orthogonally Symmetric Cross Gramian)
For an orthogonally symmetric, asymptotically stable, linear system a cross gramian is given by:

$$W_X := \int_0^\infty \mathrm{e}^{At} BUC\, \mathrm{e}^{At}\, \mathrm{d}t.$$

which fulfills:

$$W_X^2 = W_C W_O.$$

Proof.
The proof is similar to the proof of Lemma 3.10 and can be found in [1]. □

Furthermore, as shown in [1], the cross gramian of an orthogonally symmetric system can be computed only by P and either a controllability or observability gramian as:

$$W_X = W_C P^{-1} = P W_O.$$

The embedding approach from [209, 210] requires a symmetrizer matrix $J \in \mathbb{R}^{N\times N}$ to the system matrix $A \in \mathbb{R}^{N\times N}$, which is a symmetric matrix $J = J^\intercal$ with the property,

$$AJ = JA^\intercal,$$

and is the solution[28] to the Sylvester equation $AJ - JA^\intercal = 0$ [46]. Given a symmetrizer matrix J to the system matrix A of a linear control system, a system can be embedded into a symmetric system as follows:

$$\begin{aligned} \bar{A} &:= A, \\ \bar{B} &:= \begin{pmatrix} JC^\intercal & B \end{pmatrix}, \\ \bar{C} &:= \begin{pmatrix} C \\ B^\intercal J^{-1} \end{pmatrix}. \end{aligned}$$

Then, the cross gramian of the enclosing symmetric system $\bar{\Sigma}(\bar{A}, \bar{B}, \bar{C})$ approximates the cross gramian of the original system.

[28] A symmetrizer matrix can also be approximated by eigenanalysis as proposed in [53].

A recent approach [117] is related to **decentralized control**, which aims to partition a MIMO system into a set of SISO systems by selecting the input-output combinations that exhibit the highest coherence. Splitting the input matrix B by columns and the output matrix C by rows,

$$B = \begin{pmatrix} b_1 & \dots & b_M \end{pmatrix},\ b_i \in \mathbb{R}^{N\times 1}, \quad C = \begin{pmatrix} c_1 \\ \vdots \\ c_O \end{pmatrix},\ c_j \in \mathbb{R}^{1\times N},$$

yields a set of $M \times O$ many SISO sub-systems $\Sigma(A, b_i, c_j)$. The system gramians for these SISO systems are given by:

$$\begin{aligned} W_{C,i} &= \int_0^\infty \mathrm{e}^{At}\, b_i b_i^\mathsf{T}\, \mathrm{e}^{A^\mathsf{T} t}\, \mathrm{d}t, \\ W_{O,j} &= \int_0^\infty \mathrm{e}^{A^\mathsf{T} t}\, c_j^\mathsf{T} c_j\, \mathrm{e}^{At}\, \mathrm{d}t, \\ W_{X,ij} &= \int_0^\infty \mathrm{e}^{At}\, b_i c_j\, \mathrm{e}^{At}\, \mathrm{d}t. \end{aligned}$$

In decentralized control, interaction measures[29] based on Schatten-norms of the Hankel operators related to these subsystems [136, 165, 166, 167, 196] are used to quantify the coherence between an input and an output. All interaction measures of a system then constitute a participation matrix or pairing matrix, the maximum values of which, selected either row- or column-wise, determine the input-output pairings of the decentralized system.
Following [165, 3], a relation between the MIMO system gramians and the SISO system gramians is demonstrated exemplary for the cross gramian.

Lemma 3.16 (Cross Gramian Superposition)
The cross gramian W_X of a square MIMO system is equal to a sum of $M = O$ many SISO sub-system cross gramians with $i = j$:

$$W_X = \sum_{i=1}^{M} W_{X,ii}.$$

Proof.

$$\begin{aligned} W_X &= \int_0^\infty \mathrm{e}^{At}(BC)\,\mathrm{e}^{At}\,\mathrm{d}t \\ &= \int_0^\infty \mathrm{e}^{At}\Big(\sum_{i=1}^{M} b_i c_i\Big)\,\mathrm{e}^{At}\,\mathrm{d}t \\ &= \sum_{i=1}^{M} \int_0^\infty \mathrm{e}^{At}\, b_i c_i\, \mathrm{e}^{At}\,\mathrm{d}t. \end{aligned}$$

□

[29] The interaction measures are closely related to parameter identifying measures in Section 3.4.

Similarly, a relation can be derived for the controllability and observability gramian:

$$W_C = \sum_{i=1}^{M} W_{C,i}, \qquad W_O = \sum_{j=1}^{O} W_{O,j}.$$

Hence, the product of controllability and observability gramian, which is essential to Lemma 3.10, can be written in terms of the subsystem cross gramians:

$$W_C W_O = \sum_{i=1}^{M}\sum_{j=1}^{O} W_{C,i} W_{O,j} = \sum_{i=1}^{M}\sum_{j=1}^{O} W_{X,ij} W_{X,ij}.$$

This relation motivates a definition for a non-symmetric cross gramian, proposed in [117], based upon the SISO partitioning of the underlying system.

Definition 3.17 (Non-Symmetric Cross Gramian)
The non-symmetric cross gramian W_Z is defined as the sum of ***all*** *subsystem cross gramians $W_{X,ij}$:*

$$W_Z := \sum_{i=1}^{M}\sum_{j=1}^{O} W_{X,ij}.$$

As noted in [117] this cross gramian does not preserve the regular cross gramian's core property of Lemma 3.10, but is shown to have a property related to linear superposition justifying its use.

Lemma 3.18 (Average Cross Gramian)
The non-symmetric cross gramian W_Z is equal to the cross gramian of the "averaged" SISO system $\Sigma(A, \sum_{i=1}^{M} b_i, \sum_{j=1}^{O} c_j)$.

Proof.

$$\begin{aligned} W_Z &= \sum_{i=1}^{M}\sum_{j=1}^{O} \int_0^\infty \mathrm{e}^{At}\, b_i c_j\, \mathrm{e}^{At}\, \mathrm{d}t \\ &= \int_0^\infty \mathrm{e}^{At} \Big(\sum_{i=1}^{M}\sum_{j=1}^{O} b_i c_j\Big)\, \mathrm{e}^{At}\, \mathrm{d}t \\ &= \int_0^\infty \mathrm{e}^{At} \Big(\sum_{i=1}^{M} b_i\Big)\Big(\sum_{j=1}^{O} c_j\Big)\, \mathrm{e}^{At}\, \mathrm{d}t. \end{aligned}$$

□

Thus, this non-symmetric cross gramian can be seen as an average or mean cross gramian over all SISO subsystem's cross gramians.
The non-symmetric cross gramian can also be computed for nonlinear systems if a cross gramian for a nonlinear SISO system is available[30]. But it should be noted that for nonlinear systems the property from Lemma 3.18 cannot be expected to hold globally.

[30] For example using the empirical cross gramian, see Section 3.3.3.

3.2.4. Balanced Truncation

Various methods are available for the construction of projections yielding reduced input-output mappings from the preceding system gramians [231]. Following, the renowned balancing approach is summarized. An initial result of this chapter was the factorization of the Hankel operator H into the observability and controllability operators:

$$H = \mathscr{O}\mathscr{C}.$$

This leads, first, to a relation between the Hankel operator and the cross gramian:

$$\operatorname{tr}(H) = \operatorname{tr}(\mathscr{O}\mathscr{C}) = \operatorname{tr}(\mathscr{C}\mathscr{O}) = \operatorname{tr}(W_X);$$

as the controllability and observability operators are Hilbert-Schmidt operators from which this trace property follows. And second, a further relation between the HSVs and the system gramians emerges.

Lemma 3.19
The singular values of the Hankel operator coincide with the square root of the eigenvalues of the product of controllability and observability gramian:

$$\sigma_i(H) = \sqrt{\lambda_i(W_C W_O)}.$$

Proof.

$$\begin{aligned}
\sigma_i(H) = \sqrt{\lambda_i(H^*H)} &= \sqrt{\lambda_i((\mathscr{O}\mathscr{C})^*(\mathscr{O}\mathscr{C}))} \\
&= \sqrt{\lambda_i(\mathscr{C}^*\mathscr{O}^*\mathscr{O}\mathscr{C})} \\
&= \sqrt{\lambda_i(\mathscr{C}^*W_O\mathscr{C})}
\end{aligned}$$

Using [119, Proposition 1] yields:

$$\begin{aligned}
&= \sqrt{\lambda_i(W_O\mathscr{C}\mathscr{C}^*)} \\
&= \sqrt{\lambda_i(W_O W_C)} \\
&\quad\vdots \\
&= \sqrt{\lambda_i(W_C W_O)}.
\end{aligned}$$

□

And by (3.6), also the absolute value of the eigenvalues of the cross gramian (in case of a symmetric system) are equal to the HSVs:

$$\sigma_i(H) = |\lambda_i(W_X)|,$$

which means, that the HSV are computable through the system gramians, either by the controllability and observability gramians or by the cross gramian. To translate this property into a method for reduced order modelling, the underlying system has to undergo a (linear coordinate) transformation that orients the systems along the directions belonging to the singular vectors associated to the HSVs.

Balanced Realization

To obtain the reduced order model, the least controllable **and** least observable states are to be neglected[31]. This requires a balancing (ordering) of the states simultaneously in terms of controllability and observability. A balancing transformation T is obtained from the simultaneous diagonalization of the controllability and observability gramian [6, Ch. 7] such that:

$$TW_C T^\intercal = T^{-\intercal} W_O T^{-1} = \begin{pmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_N \end{pmatrix}.$$

Typically, the so called square-root-algorithm is used to compute a balancing transformation [146, 19], which does not require the full order controllability and observability gramians, but their respective Cholesky factors L_C and L_O:

$$\begin{aligned} W_C &\overset{\text{Cholesky}}{=} L_C L_C^\intercal, \\ W_O &\overset{\text{Cholesky}}{=} L_O L_O^\intercal, \\ L_C L_O^\intercal &\overset{\text{SVD}}{=} UDV. \end{aligned}$$

The Cholesky factors of the empirical gramians, which are introduced in the following Section 3.3, are not computable without assembling the full gramian. Related to the commonly used square-root-algorithm is the balancing algorithm from [19, 81]:

$$\begin{aligned} W_C &\overset{\text{SVD}}{=} U_C D_C U_C^\intercal, \\ W_O &\overset{\text{SVD}}{=} U_O D_O U_O^\intercal, \\ U_C D_C^{\frac{1}{2}} U_C^\intercal U_O D_O^{\frac{1}{2}} U_O^\intercal &= W_C^{\frac{1}{2}} W_O^{\frac{1}{2}}, \\ W_C^{\frac{1}{2}} W_O^{\frac{1}{2}} &\overset{\text{SVD}}{=} UDV. \end{aligned}$$

This algorithm can be computationally more efficient if the reduced order is known and the full system gramians are computed directly instead of their Cholesky factors [146, 222], since then truncated SVDs can be computed for the system gramians W_C and W_O as in **balanced proper orthogonal decomposition** (bPOD) [228, 184].
Now, U and V are the right and left balancing transformations and the diagonal matrix D comprises the Hankel singular values of which the right subspace spanned by U can be considered the controllability subspace and the left subspace spanned by V the observability subspace. In the following it is essential that the left and right singular vectors in U and V are sorted by their associated HSVs in descending order. The bi-orthogonal Petrov-Galerkin projection $\{U, V\}$ applied to a linear system gives a system in balanced form, with the same input-output behavior, but with balanced states sorted simultaneously by their controllability and observability.

[31] In contrast to POD, which in comparison considers only controllability [229].

Truncated Balanced Realization

Given a balancing transformation $\{U, V\}$, the reduced order model can be obtained by truncating the $(N-n)$ least significant states in terms of input-output coherence. A truncating projection S:

$$S := \begin{pmatrix} \mathbb{1}_n \\ 0_n \end{pmatrix} \in \mathbb{R}^{N \times n},$$

can then be applied to the balanced model components or directly to the balancing projections:

$$\begin{aligned} U_1 &:= U \circ S, \\ V_1 &:= S^\intercal \circ V. \end{aligned}$$

Hence, the method called **balanced truncation** (BT) consists of the composition of a balancing transformation with a truncating projection. Truncating $(N-n)$ columns and rows from U and V respectively, yields a (balanced) reducing truncated projection,

$$\begin{aligned} U = \begin{pmatrix} U_1 & U_2 \end{pmatrix} &\to U_1 \in \mathbb{R}^{N \times n}, \\ V = \begin{pmatrix} V_1 \\ V_2 \end{pmatrix} &\to V_1 \in \mathbb{R}^{n \times N}, \end{aligned} \tag{3.8}$$

from which a stability-preserving [175, 198] reduced order model follows as in (2.12).

Approximate Balancing and Direct Truncation

For the cross gramian, a balancing transformation can be determined from its eigendecomposition:

$$W_X = T \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_N \end{pmatrix} T^{-1}.$$

Assuming the eigenvalues $\lambda_i(W_X)$ are sorted in descending order by their absolute value [4], the associated eigenvectors yield left and right eigenspace projections T and T^{-1} [72, Ch. 7.4]:

$$\begin{aligned} U &= T^{-1}, \\ V &= T, \end{aligned}$$

and can be used as a two-sided balancing projection in case of a (orthogonally) symmetric system. The left (or right) eigenspace projection can then be truncated as for the truncated balanced realization and used as a reducing projection. This method is equivalent to the poor man's truncated balanced realization (PMTBR) described in [176].

The projection obtained from an eigendecomposition is generally not orthogonal. Alternatively, an approximate balancing transformation can be computed from a singular value decomposition of the cross gramian W_X:

$$W_X \overset{\text{SVD}}{=} UDV.$$

In [180, M3] it is proposed to truncate columns from U and rows from V and use it as two-sided projection. Yet, a (bi-orthogonal) Petrov-Galerkin projection $\{U_1, V_1\}$ is generally not stability preserving.

By a truncation of the left singular vectors[32] forming the columns of U, and using it as an orthogonal projection, a reduced order model is assembled from the truncated projection $\{U_1, U_1^\intercal\}$. This method is called **direct truncation**[33] (DT) [110] and yields an orthogonal Galerkin projection. A Galerkin projection is generally not stability preserving, but in this case, due to the use of adjoint information in W_X, the projection obtained from direct truncation preserves the stability of the ROM by the same argument as in [120, Ch. 5.4]. The direct truncation of the cross gramian is closely related to the balanced POD procedure from [184]. This is illustrated by viewing the SVD of W_X as an SVD of W_O with respect to the inner product $\mathscr{C}^*\mathscr{C}$:

$$\begin{aligned}
W_X = \mathscr{C}\mathscr{O} \rightarrow \mathscr{O}^*\mathscr{C}^*\mathscr{C}\mathscr{O} &= \mathscr{O}^*(\mathscr{C}^*\mathscr{C})\mathscr{O} \\
\rightarrow V(y(t)) &:= y(t)^\intercal(\mathscr{C}^*\mathscr{C})y(t) \\
\Rightarrow \dot{V}(y(t)) &= \dot{y}(t)^\intercal(\mathscr{C}^*\mathscr{C})y(t) + y(t)^\intercal(\mathscr{C}^*\mathscr{C})\dot{y}(t) \\
&= x^\intercal(A^\intercal(\mathscr{O}(t)^\intercal\mathscr{C}^*\mathscr{C}\mathscr{O}(t)) + (\mathscr{O}(t)^\intercal\mathscr{C}^*\mathscr{C}\mathscr{O}(t))A)x \overset{A<0}{<} 0,
\end{aligned}$$

with $\mathscr{O}(t) = C\,\mathrm{e}^{At}$ being the finite time observability operator. This inner product is energy-preserving, if the system is stable, since V is a Lyapunov function [105, Sec. 8.5]. The stability property of the balanced POD, resting upon an energy-preserving inner product interpretation [185, Sec. 2.3], thus transfers to the cross-gramian-based direct truncation. This illustrates the viewpoint in [184, Sec. 3.4] of balanced truncation, balanced POD (and the cross gramian) being variants of the POD using an energy-preserving inner products [185].

Overall, the advantage of the cross gramian method compared to balanced truncation is that conceptually only one gramian needs to be computed from which a single truncated SVD obtains the reducing projection, and in case of direct truncation the ROM is not only stability preserving but under a weak condition also passivity preserving[34]; yet, due to the use of a one-sided projection the resulting ROM may be less accurate compared to balanced truncation. Additionally, the cross gramian provides more information on the underlying system as outlined in Section 3.2.3.

[32] Alternatively, also the right singular vectors could be used.

[33] Generally, the singular values of the cross gramian are not equal to the HSV.

[34] A system is called passive if it does not generate energy, which means mathematically: $\int_{-\infty}^{t} u(\tau)^\intercal y(\tau)\,\mathrm{d}\tau \geq 0$, $\forall\, t \in \mathbb{R}$ (see [6, Ch. 5.9.1]). Balanced truncation is generally not passivity preserving, but there exist variants of balanced truncation which guarantee passivity [177]. The cross-gramian-based direct truncation is passivity preserving if the underlying system matrix A is negative semi-definite since the Galerkin projection is a congruence transformation that preserves definiteness.

$\mathcal{H}_\infty$-Error Bound

An acclaimed property of balanced truncation is the existence of an a-priori error bound of the model reduction $\mathcal{H}_\infty$-error for the reduced order system, which was developed concurrently, but separately, in [86] and [61].

Theorem 3.20 ($\mathcal{H}_\infty$-Error Bound)
For distinct HSVs $\sigma_1(H) < \ldots < \sigma_N(H)$ the error in the $\mathcal{H}_\infty$-norm is bounded by:

$$\begin{aligned} \|G(s) - G_r(s)\|_{\mathcal{H}_\infty} &\leq 2 \sum_{k=n+1}^{N} \sigma_k(H) \\ &= 2 \sum_{k=n+1}^{N} \sqrt{\lambda_k(W_C W_O)} \\ &= 2 \sum_{k=n+1}^{N} |\lambda_k(W_X)|. \end{aligned} \tag{3.9}$$

Hence, the model reduction error in the $\mathcal{H}_\infty$-norm is enclosed by the following error bound solely based on the system's (truncated) HSVs [8]:

$$\sigma_{n+1}(H) \leq \|G(s) - G_r(s)\|_{\mathcal{H}_\infty} \leq 2 \sum_{k=n+1}^{N} \sigma_k(H).$$

With (2.18) this $\mathcal{H}_\infty$-error bound[35] extends to the L_2-norm in the time-domain. A more conservative, yet more practical error bound is given by:

$$2 \sum_{k=n+1}^{N} \sigma_k(H) \leq 2 \sum_{k=n+1}^{N} \sigma_{n+1}(H) = 2(N-n)\sigma_{n+1}(H), \tag{3.10}$$

which does not require all HSVs to be known.

$\mathcal{H}_2$-Error Indicator

Also in the $\mathcal{H}_2$-norm, an error bound for the model reduction error $\|G - G_r\|_{\mathcal{H}_2}$ can be derived, but it requires the full order system gramians [163]. Since this may be a complex task for large systems, a simpler error indicator from [210] is presented here.
The $\mathcal{H}_2$-norm can be expressed utilizing the controllability or observability gramians:

$$\|G\|_{\mathcal{H}_2} = \sqrt{\operatorname{tr}(C W_C C^\intercal)} = \sqrt{\operatorname{tr}(B^\intercal W_O B)}; \tag{3.11}$$

and for symmetric systems also by the cross gramian:

$$\|G\|_{\mathcal{H}_2} = \sqrt{\operatorname{tr}(C W_X B)}. \tag{3.12}$$

This relation is directly derived from the definition of the gramians and used in [210] to obtain the following (a-priori) $\mathcal{H}_2$ model reduction error indicator.

[35] In [173], the $\mathcal{H}_\infty$-error bound is explored from a trace-norm point of view.

Note 3.21 ($\mathscr{H}_2$-Error Indicator)
For a balanced system $\widetilde{\Sigma}(\widetilde{A},\widetilde{B},\widetilde{C})$ and a diagonal[36] system gramian W with a partitioning of the following form:

$$\begin{pmatrix}\dot{\tilde{x}}_1(t)\\ \dot{\tilde{x}}_2(t)\end{pmatrix} = \begin{pmatrix}\widetilde{A}_{11} & \widetilde{A}_{12}\\ \widetilde{A}_{21} & \widetilde{A}_{22}\end{pmatrix}\begin{pmatrix}\tilde{x}_1(t)\\ \tilde{x}_2(t)\end{pmatrix} + \begin{pmatrix}\widetilde{B}_1\\ \widetilde{B}_2\end{pmatrix}u(t),$$
$$y(t) = \begin{pmatrix}\widetilde{C}_1 & \widetilde{C}_2\end{pmatrix}\begin{pmatrix}\tilde{x}_1(t)\\ \tilde{x}_2(t)\end{pmatrix},$$
$$W = \begin{pmatrix}W_{11} & 0\\ 0 & W_{22}\end{pmatrix},$$

an error indicator is provided by:

$$\|G(s) - G_r(s)\|_{\mathscr{H}_2} \approx \sqrt{\operatorname{tr}(\widetilde{C}_2 W_{22}\widetilde{B}_2)}. \tag{3.13}$$

For the $\mathscr{H}_2$-error indicator, a more conservative error indicator is given by:

$$\sqrt{\operatorname{tr}(\widetilde{C}_2 W_{22}\widetilde{B}_2)} \leq \sqrt{\operatorname{tr}(\widetilde{C}_2 \sigma_{n+1}\widetilde{B}_2)}, \tag{3.14}$$

which only requires the first $(n+1)$ HSVs to be known.

Balanced Gains

The relation in (3.11) also suggests that truncation of (balanced) states based on the HSVs might result in larger errors in the $\mathscr{H}_2$-norm. In [131] the concept of **balanced gains** is introduced with regard to this issue. A related, but more accessible, procedure is presented in [49], based on (3.11) and (3.12); given a balanced input matrix $\widetilde{B} = \begin{pmatrix}\tilde{b}_1 & \dots & \tilde{b}_N\end{pmatrix}^\mathsf{T}$ and a balanced output matrix $\widetilde{C} = \begin{pmatrix}\tilde{c}_1 & \dots & \tilde{c}_N\end{pmatrix}$, it is proposed to partition and truncate the states not based upon the HSVs $\sigma_i(H)$, but on the quantities d_i, which relate to the HSVs by:

$$d_i := \tilde{b}_i\tilde{b}_i^\mathsf{T}\sigma_i = \tilde{c}_i^\mathsf{T}\tilde{c}_i\sigma_i = |\tilde{b}_i\tilde{c}_i|\sigma_i.$$

L_1-Error Bound

Additionally, an L_1-error bound for the impulse response, and hence for the L_1 time-domain error, is derived in [144] and extended in [171, Ch. 1]:

$$\|g - g_r\|_{L_1} \leq 4(N+n)\sum_{i=n+1}^{N}\sigma_i, \tag{3.15}$$

which in turn is bounded, given the first $n+1$ HSVs, by

$$4(N+n)\sum_{i=n+1}^{N}\sigma_i \leq 4(N+n)(N-n-1)\sigma_{n+1} = 4(N^2-N-n^2-n)\sigma_{n+1}. \tag{3.16}$$

[36] In balanced form all system gramians are diagonal and equal and are composed of the HSVs.

Second-Order Systems

A special case of linear systems are linear second-order systems, which in practice often result from mechanical applications and are of the form:

$$\begin{aligned} M\ddot{q}(t) + G\dot{q}(t) + Kq(t) &= B_V u(t), \\ y(t) &= C_P q(t) + C_V \dot{q}(t), \end{aligned}$$

with a (non-singular) mass matrix $M \in \mathbb{R}^{\frac{N}{2}\times\frac{N}{2}}$, damping matrix $G \in \mathbb{R}^{\frac{N}{2}\times\frac{N}{2}}$, stiffness matrix $K \in \mathbb{R}^{\frac{N}{2}\times\frac{N}{2}}$, input matrix $B_V \in \mathbb{R}^{\frac{N}{2}\times M}$ and output matrices $C_P, C_V \in \mathbb{R}^{O\times\frac{N}{2}}$. Referring to the physical interpretation of second-order systems in Hamiltonian mechanics, P denotes "position" and V denotes "velocity". Such systems can be reduced to a first-order system (2.6), where the components A, B, C of the (first-order) linear control system have the following structure:

$$\begin{aligned} A &= \begin{pmatrix} 0_{\frac{N}{2}} & \mathbb{1}_{\frac{N}{2}} \\ -M^{-1}K & -M^{-1}G \end{pmatrix}, \\ B &= \begin{pmatrix} 0_{\frac{N}{2}} \\ M^{-1}B_V \end{pmatrix}, \\ C &= \begin{pmatrix} C_P & C_V \end{pmatrix}. \end{aligned} \tag{3.17}$$

In this form, the previously described ansatz for gramian-based model reduction can be applied. Yet, due to the original second-order nature of the system, structure preserving (and also more efficient) methods, called second-order balanced truncation (SOBT) [36], are available[37] [183, 233, 21].
The associated system gramians to a second-order system interpreted as first-order system have the following block structure:

$$\begin{aligned} W_C &= \begin{pmatrix} W_{C,P} & W_{C,12} \\ W_{C,12}^\intercal & W_{C,V} \end{pmatrix}, \\ W_O &= \begin{pmatrix} W_{O,P} & W_{O,12} \\ W_{O,12}^\intercal & W_{O,V} \end{pmatrix}, \\ W_X &= \begin{pmatrix} W_{X,P} & W_{X,12} \\ W_{X,21} & W_{X,V} \end{pmatrix}. \end{aligned}$$

A variety of methods for structure preserving balanced truncation are presented in [183, 220]. The common approach in those methods is the use of separate balancing projections for the position and velocity components,

$$\begin{aligned} V &= \begin{pmatrix} V_P & V_V \end{pmatrix}, \\ U &= \begin{pmatrix} U_P \\ U_V \end{pmatrix}, \end{aligned}$$

which are applied to their associated system components.

[37] See also [187, Ch. 7.2].

By the truncated projection matrices $U_{P,1}, U_{V,1} \in \mathbb{R}^{\frac{N}{2} \times \frac{n}{2}}$ and $V_{P,1}, V_{V,1} \in \mathbb{R}^{\frac{n}{2} \times \frac{N}{2}}$ a ROM is obtained which is of the same block structure as the full-order model (3.17),

$$
\begin{aligned}
A_r &= \begin{pmatrix} 0_{\frac{n}{2}} & \mathbb{1}_{\frac{n}{2}} \\ -V_{V,1} M^{-1} K U_{P,1} & -V_{V,1} M^{-1} G U_{V,1} \end{pmatrix}, \\
B_r &= \begin{pmatrix} 0_{\frac{n}{2}} \\ V_{V,1} M^{-1} B_V \end{pmatrix}, \\
C_r &= \begin{pmatrix} C_P U_{P,1} & C_V U_{V,1} \end{pmatrix}.
\end{aligned}
$$

Generally, as opposed to first-order balanced truncation and direct truncation these methods are not guaranteed to yield stable ROMs as shown in [183].
Following, three selected methods are listed from [183, 220] for balanced truncation (BT),

- balanced truncation of the position gramians:

$$\{V_P = V_V, U_P = U_V\} \rightarrow BT(W_{C,P}, W_{O,P}),$$

- balanced truncation of the velocity gramians:

$$\{V_P = V_V, U_P = U_V\} \rightarrow BT(W_{C,V}, W_{O,V}),$$

- balanced truncation of the position and velocity gramians:

$$\begin{cases} \{V_P, U_P\} \rightarrow BT(W_{C,P}, W_{O,P}) \\ \{V_V, U_V\} \rightarrow BT(W_{C,V}, W_{O,V}) \end{cases}.$$

as well as the cross-gramian-based methods [233] for the direct truncation (DT),

- direct truncation of the position cross gramian:

$$\{V_P = V_V = U_P^\intercal = U_V^\intercal\} \rightarrow DT(W_{X,P}),$$

- direct truncation of the velocity cross gramian:

$$\{V_P = V_V = U_P^\intercal = U_V^\intercal\} \rightarrow DT(W_{X,V}),$$

- direct truncation of the position and velocity cross gramian:

$$\begin{cases} \{V_P = U_P^\intercal\} \rightarrow DT(W_{X,P}) \\ \{V_V = U_V^\intercal\} \rightarrow DT(W_{X,V}) \end{cases}.$$

The above listed methods do not require the mixed position-velocity gramians, which consequently will not have to be computed or stored, and preserve the second-order structure of the system.

3.3. Empirical Gramians

Already in [159] a gramian-based approach for nonlinear model reduction is presented. Therein, after a linearization around a steady-state, a balancing transformation is determined from the linearized model, which is then truncated and applied as in (2.11).
A systematic ansatz to nonlinear balancing is pursued in [189, 190], which is based on the controllability energy function L_C and observability energy function L_O:

$$L_C := \min_{u \in \mathrm{L}_2(0,\infty)} \frac{1}{2} \int_{-\infty}^{0} \|u(t)\|^2 \,\mathrm{d}t = \frac{1}{2} x_0^\intercal W_C^{-1} x_0,$$
$$L_O := \frac{1}{2} \int_{0}^{\infty} \|y(t)\|^2 \,\mathrm{d}t = \frac{1}{2} x_0^\intercal W_O x_0.$$

From a computational point of view this method is rather involved for nonlinear systems, requiring the solution of an optimal control problem on a Hamilton-Jacobi PDE and a nonlinear Lyapunov equation.
Initially, in [168], the controllability and observability gramian were not computed by solving Lyapunov equations, but from (output) trajectories; hence, nonlinear systems are generally admissible for the construction of the system gramians. This approach is the basis for the empirical gramians[38] introduced in [142, 143]. An empirical gramian matrix is not computed as solution of a matrix equation, but as (auto-)correlation matrix W_i of trajectories $x_i(t)$,

$$W_i = \int_0^\infty (x_i(t) - \bar{x}_i)(x_i(t) - \bar{x}_i)^\intercal \,\mathrm{d}t, \tag{3.18}$$

with the temporal mean $\bar{x}_i$,

$$\bar{x}_i := \lim_{\mathscr{T} \to \infty} \frac{1}{\mathscr{T}} \int_0^{\mathscr{T}} x_i(t) \,\mathrm{d}t.$$

These correlation matrices are then averaged over a certain operating region Ω of the underlying system:

$$W = \sum_{i=1}^{|\Omega|} W_i.$$

This operating region Ω is defined by perturbations of a nominal input $u(t)$ and a nominal initial state x_0.

Definition 3.22 (Perturbation Sets)
*The **input perturbations** for an impulsive input $u(t) = \delta(t)$ are defined as:*

$$E_u = \{e_i \in \mathbb{R}^M; \|e_i\| = 1; \langle e_i, e_{j \neq i} \rangle = 0; i, j = 1, \ldots, M\},$$
$$R_u = \{S_i \in \mathbb{R}^{M \times M}; S_i^\intercal S_i = \mathbb{1}_M; i = 1, \ldots, s\},$$
$$Q_u = \{c_i \in \mathbb{R}; c_i > 0; i = 1, \ldots, q\}.$$

[38] A related approach is already described in [174] and enhanced in [158].

*The **initial state perturbations** for an initial state x_0 are defined as:*

$$E_x = \{f_i \in \mathbb{R}^N; \|f_i\| = 1; \langle f_i, f_{j\neq i}\rangle = 0; i, j = 1, \dots, N\},$$
$$R_x = \{T_i \in \mathbb{R}^{N\times N}; T_i^\intercal T_i = \mathbb{1}_N; i = 1, \dots, t\},$$
$$Q_x = \{d_i \in \mathbb{R}; d_i > 0; i = 1, \dots, r\}.$$

The perturbations can be understood as (standard) directions, rotations[39] and scales for the input and initial state, by which the whole operating region $\Omega = (E_u \times R_u \times Q_u) \times (E_x \times R_x \times Q_x)$ can be covered [44]. While the choice of the scaling sets Q_u, Q_x depends on the application, as a generic choice for the rotation sets R_u, R_x, the identity matrix and negative identity matrix $\{\mathbb{1}, -\mathbb{1}\}$ are suggested in [142, 143]. Alternatively, in [83, 82, 58] a (partial) factorial design is proposed, which enables combinations of component perturbations, yet this results in large rotation matrices.
For each point in the (Cartesian) product of the required perturbation sets, an individual empirical gramian is computed. The (arithmetic) average over all these sub-gramians then yields the final empirical gramian.
Since the computation of empirical gramians requires only trajectories, this approach extends, beyond linear systems, also to nonlinear systems; yet for linear systems the empirical gramians reduce to the (linear) system gramians (see: Lemma 3.24, Lemma 3.28, Lemma 3.33). Also, due to the initial state perturbations, empirical balanced truncation addresses the issue of inhomogeneous initial conditions raised in [104].
The empirical gramian method requires the asymptotic stability of the underlying system over the whole operating region. Accuracy of the reduced order model obtained from empirical gramians depends heavily on the coverage of the operating region encoded in the perturbation sets. Given a good coverage, the empirical gramians provide more precise information than classic algebraic methods on the controllability and observability [200] and thus the input-output behavior; and for nonlinear model reduction the empirical gramians can outperform linearization [51].
As a generalization to the empirical gramian matrices, the **empirical covariance matrices** were introduced in [100, 102]. The empirical covariance matrices allow, apart from centering the trajectories around the steady-state instead of the temporal average, for a larger class of input functions. While the empirical gramians require impulse input $u(t) = \delta(t)$, the empirical covariance matrices admit a series of step functions $u(t) = v(t)$ as input:

$$v(t) := \sum_k v_k \chi_{[t_k, t_{k+1})}(t). \tag{3.19}$$

A discrete variant, the **discrete empirical covariance matrices** can then be computed numerically [98, 99], for ***any discretized*** input function.
Beyond model reduction, the empirical gramians have a variety of applications in system identification, for example nonlinearity quantification [98]; and while in this work only time-invariant systems are of interest, in [44] the use of empirical gramians for (linear) time-varying systems is developed, and applied for example in [111].

[39] Rotation matrices are orthogonal by definition.

3.3.1. Empirical Controllability Gramian

Originally, in [168], the controllability gramian W_C is computed using the impulse responses $x^j(t) = \mathrm{e}^{At}Be_j$, with e_j being the j-th column of the $M \times M$-dimensional identity matrix, which are concatenated to $X(t) = \begin{pmatrix} x^1(t) & \dots & x^M(t) \end{pmatrix}$:

$$W_C = \int_0^\infty X(t)X^\mathsf{T}(t)\,\mathrm{d}t.$$

The empirical controllability gramian introduced in [142, 143] is a generalization to this approach, in which the trajectories are centered and the controllability gramians for each centered trajectory are averaged over the Cartesian product of all perturbations.

Definition 3.23 (Empirical Controllability Gramian)
*For sets E_u, R_u, Q_u, the **empirical controllability gramian** $\widehat{W}_C \in \mathbb{R}^{N\times N}$ is given by:*

$$\widehat{W}_C = \frac{1}{|Q_u||R_u|}\sum_{h=1}^{|Q_u|}\sum_{i=1}^{|R_u|}\sum_{j=1}^{M}\frac{1}{c_h^2}\int_0^\infty \Psi^{hij}(t)\,\mathrm{d}t,$$
$$\Psi^{hij}(t) = (x^{hij}(t) - \bar{x}^{hij})(x^{hij}(t) - \bar{x}^{hij})^\mathsf{T} \in \mathbb{R}^{N\times N},$$

with $x^{hij}(t)$ being the trajectory for the input configuration $u^{hij}(t) = c_h S_i e_j \delta(t)$ and $\bar{x}^{hij}$ the associated temporal mean state.

A justification for this definition is provided in [142, 143] by showing that the empirical controllability gramian reduces to the controllability gramian from Definition 3.3 for linear control systems.

Lemma 3.24 (Empirical Controllability Gramian)
For asymptotically stable linear systems the empirical controllability gramian is equal to the (linear) controllability gramian.

Proof.

$$\begin{aligned}
\Psi^{hij}(t) &= (\mathrm{e}^{At}Bc_hS_ie_j)(\mathrm{e}^{At}Bc_hS_ie_j)^\mathsf{T}\\
&= c_h^2(\mathrm{e}^{At}BS_ie_j)(e_j^\mathsf{T}S_i^\mathsf{T}B^\mathsf{T}\mathrm{e}^{A^\mathsf{T}t})\\
&= c_h^2\,\mathrm{e}^{At}BB^\mathsf{T}\mathrm{e}^{A^\mathsf{T}t}\\
\Rightarrow \widehat{W}_C &= \frac{1}{|Q_u||R_u|}\sum_{h=1}^{|Q_u|}\sum_{i=1}^{|R_u|}\sum_{j=1}^{M}\frac{1}{c_h^2}\int_0^\infty c_h^2\,\mathrm{e}^{At}BB^\mathsf{T}\mathrm{e}^{A^\mathsf{T}t}\,\mathrm{d}t\\
&= \int_0^\infty \mathrm{e}^{At}BB^\mathsf{T}\mathrm{e}^{A^\mathsf{T}t}\,\mathrm{d}t = W_C.
\end{aligned}$$

□

Hence, the empirical controllability gramian $\widehat{W}_C$ is subsequently denoted by W_C.

In [99, 102] the empirical controllability gramian is generalized to an empirical controllability covariance matrix, by expanding the class of admissible input function from impulse input to a series of step functions (3.19) and centering the state trajectories around a steady-state instead of the temporal mean.

Definition 3.25 (Empirical Controllability Covariance Matrix)
*For sets E_u, R_u, Q_u, input $u(t)$ and input during the steady-state $\bar{x}$, $\bar{u}$, the **empirical controllability covariance matrix** $\widetilde{W}_C \in \mathbb{R}^{N\times N}$ is given by:*

$$\widetilde{W}_C = \frac{1}{|Q_u||R_u|}\sum_{h=1}^{|Q_u|}\sum_{i=1}^{|R_u|}\sum_{j=1}^{M}\frac{1}{c_h^2}\int_0^\infty \Psi^{hij}(t)\,\mathrm{d}t,$$
$$\Psi^{hij}(t) = (x^{hij}(t)-\bar{x}^{hij})(x^{hij}(t)-\bar{x}^{hij})^\intercal \in \mathbb{R}^{N\times N},$$

with $x^{hij}(t)$ being the trajectory for the input configuration $u^{hij}(t) = c_h S_i e_j \odot u(t) + \bar{u}$.

Since an empirical controllability gramian or covariance matrix is computed from discrete trajectories, a discrete version of the previous definition follows as suggested in [98, 99].

Note 3.26 (Discrete Empirical Controllability Gramian / Covariance Matrix)
*For sets E_u, R_u, Q_u, discrete time interval $[0, \mathscr{T}]$ with time step Δt, discrete input u_t and the temporal mean (steady-) state $\bar{x}$, $\bar{u} = 0$ (input during the steady-state $\bar{x}$, $\bar{u}$), the **discrete empirical controllability gramian (discrete empirical controllability covariance matrix)** $w_C \in \mathbb{R}^{N\times N}$ is given by:*

$$w_C = \frac{1}{|Q_u||R_u|}\sum_{h=1}^{|Q_u|}\sum_{i=1}^{|R_u|}\sum_{j=1}^{M}\frac{\Delta t}{c_h^2}\sum_{t=0}^{\mathscr{T}/\Delta t}\Psi_t^{hij},$$
$$\Psi_t^{hij} = (x_t^{hij}-\bar{x}^{hij})(x_t^{hij}-\bar{x}^{hij})^\intercal \in \mathbb{R}^{N\times N},$$

with x_t^{hij} being the discrete trajectory for the input configuration $u_t^{hij} = c_h S_i e_j \odot u_t + \bar{u}$.

The empirical controllability gramian and empirical controllability covariance matrix are closely related to PCA and POD [142, 70] of time series. For a given set of impulse responses $x^j(t)$, the associated auto-correlation matrix (for an asymptotically stable, linear system) is the empirical controllability covariance matrix:

$$\widetilde{W}_C = \mathbb{E}[XX^\intercal], \tag{3.20}$$

from which an eigendecomposition yields the POD; the associated covariance matrix for the $x^j(t)$ is the empirical controllability gramian:

$$W_C = \mathbb{E}[(X-\mathbb{E}[X])(X-\mathbb{E}[X])^\intercal], \tag{3.21}$$

from which an eigendecomposition yields the PCA.

3.3.2. Empirical Observability Gramian

In [168], the observability gramian W_O is computed using the responses $y^j(t) = C\,\mathrm{e}^{At} f_j$, with f_j being the j-th column of the $N \times N$-dimensional identity matrix, which are concatenated to $Y(t) = \begin{pmatrix} y^1(t) & \dots & y^N(t) \end{pmatrix}$:

$$W_O = \int_0^\infty Y^\intercal(t) Y(t)\,\mathrm{d}t.$$

The empirical observability gramian introduced in [142, 143] is a generalization to this approach, in which the output trajectories are centered and the observability gramians for each centered trajectory are averaged over the Cartesian product of all perturbations.

Definition 3.27 (Empirical Observability Gramian)
*For sets E_x, R_x, Q_x, the **empirical observability gramian** $\widehat{W_O} \in \mathbb{R}^{N\times N}$ is given by:*

$$\widehat{W_O} = \frac{1}{|Q_x||R_x|} \sum_{k=1}^{|Q_x|} \sum_{l=1}^{|R_x|} \frac{1}{d_k^2} \int_0^\infty T_l \Psi^{kl}(t) T_l^\intercal\,\mathrm{d}t, \tag{3.22}$$
$$\Psi^{kl}_{ab}(t) = (y^{kla}(t) - \bar{y}^{kla})^\intercal (y^{klb}(t) - \bar{y}^{klb}) \in \mathbb{R},$$

with $y^{kla}(t)$ being the output trajectory for the initial state configuration $x_0^{kla} = d_k T_l f_a$ and associated temporal mean output $\bar{y}^{kla}$.

A justification for this definition is provided in [142, 143] by showing that the empirical observability gramian reduces to the observability gramian from Definition 3.6 for linear control systems.

Lemma 3.28 (Empirical Observability Gramian)
For asymptotically stable linear systems the empirical observability gramian is equal to the (linear) observability gramian.

Proof.

$$\begin{aligned}
\Psi^{kl}_{ab} &= (C\,\mathrm{e}^{At} d_k T_l f_a)^\intercal (C\,\mathrm{e}^{At} d_k T_l f_b) \\
&= d_k^2 (f_a^\intercal T_l^\intercal\,\mathrm{e}^{A^\intercal t} C^\intercal)(C\,\mathrm{e}^{At} T_l f_a) \\
\Rightarrow \Psi^{kl} &= d_k^2 T_l^\intercal\,\mathrm{e}^{A^\intercal t} C^\intercal C\,\mathrm{e}^{At} T_l \\
\Rightarrow \widehat{W_O} &= \frac{1}{|Q_x||R_x|} \sum_{k=1}^{|Q_x|} \sum_{l=1}^{|R_x|} \frac{1}{d_k^2} \int_0^\infty T_l d_k^2 T_l^\intercal\,\mathrm{e}^{A^\intercal t} C^\intercal C\,\mathrm{e}^{At} T_l T_l^\intercal\,\mathrm{d}t \\
&= \int_0^\infty \mathrm{e}^{A^\intercal t} C^\intercal C\,\mathrm{e}^{At}\,\mathrm{d}t = W_O.
\end{aligned}$$

□

Hence, the empirical observability gramian $\widehat{W_O}$ is subsequently denoted by W_O.

[40] $\Psi^{kl}(t)$ is a matrix valued function with components $\Psi^{kl}_{ab}(t)$ for $a, b = 1 \dots N$.

In [99, 102, 202] the empirical observability gramian is generalized to an empirical observability covariance matrix, by centering the output trajectories around a steady-state output instead of the output temporal mean.

Definition 3.29 (Empirical Observability Covariance Matrix)
For sets E_x, R_x, Q_x and output $\bar{y}$ during the steady-state $\bar{x}$, the ***empirical observability covariance matrix*** $\widetilde{W_O} \in \mathbb{R}^{N\times N}$ *is given by:*

$$\widetilde{W_O} = \frac{1}{|Q_x||R_x|}\sum_{k=1}^{|Q_x|}\sum_{l=1}^{|R_x|}\frac{1}{d_k^2}\int_0^\infty T_l\Psi^{kl}(t)T_l^\intercal \,\mathrm{d}t,$$
$$\Psi_{ab}^{kl}(t) = (y^{kla}(t)-\bar{y}^{kla})^\intercal(y^{klb}(t)-\bar{y}^{klb}) \in \mathbb{R},$$

with $y^{kla}(t)$ being the output trajectory for the initial state configuration $x_0^{kla} = d_k T_l f_a + \bar{x}$.

Since an empirical observability gramian or covariance matrix is computed from discrete output trajectories, a discrete version of the previous definition follows as suggested in [98, 99].

Definition 3.30 (Discrete Empirical Observability Gramian / Covariance Matrix)
For sets E_x, R_x, Q_x, discrete time interval $[0, \mathscr{T}]$ with time step Δt, and output $\bar{y}$ during steady state $\bar{x}$, the ***discrete empirical observability gramian*** *(****discrete empirical observability covariance matrix****)* $w_O \in \mathbb{R}^{N\times N}$ *is given by:*

$$w_O = \frac{1}{|Q_x||R_x|}\sum_{k=1}^{|Q_x|}\sum_{l=1}^{|R_x|}\frac{\Delta t}{d_k^2}\sum_{t=0}^{\mathscr{T}/\Delta t} T_l\Psi_t^{kl}T_l^\intercal, \tag{3.23}$$
$$\Psi_{t,ab}^{kl} = (y_t^{kla}-\bar{y})^\intercal(y_t^{klb}-\bar{y}^{klb}) \in \mathbb{R}, \tag{3.24}$$

with y_t^{kla} being the discrete output trajectory for the initial state configuration $x_0^{kla} = d_k T_l f_a + \bar{x}$.

While the empirical controllability covariance matrix differs in the types of admissible (perturbed) inputs and the centering of the trajectories from the empirical controllability gramian, the empirical observability covariance matrix only differs in the centering of the output trajectories from the empirical observability gramian.
For linear control systems it is computationally more efficient[41] to compute the equivalent empirical controllability gramian of the adjoint system,

$$W_O = W_C^*,$$

since for the choice of perturbation sets Definition 3.22, in case of an empirical observability gramian (or empirical observability covariance matrix) multiples of $N = \dim(x_0)$ trajectories have to be obtained, yet for an empirical controllability gramian (or empirical controllability covariance matrix) only multiples of $M = \dim(u(t))$ trajectories are required.

[41] For the choice $R_u = \{-\mathbb{1}_M, \mathbb{1}_M\}$ and $R_x = \{-\mathbb{1}_N, \mathbb{1}_N\}$.

3.3.3. Empirical Cross Gramian

Similar to the empirical controllability and observability gramian, for a square system, an empirical cross gramian can be computed using the impulse responses $x^j(t) = \mathrm{e}^{At}Be_j$ and adjoint impulse responses $z^j(t) = \mathrm{e}^{A^\intercal t}C^\intercal e_j$, with e_j being the j-th column of the $M \times M$-dimensional identity matrix. These trajectories are concatenated to $X(t) = \begin{pmatrix} x^1(t) & \dots & x^M(t) \end{pmatrix}$ and $Z(t) = \begin{pmatrix} z^1(t) & \dots & z^M(t) \end{pmatrix}$ respectively:

$$W_X = \int_0^\infty X(t)Z^\intercal(t)\,\mathrm{d}t.$$

The empirical cross gramian generalizes this approach and is derived in two steps. First, for linear systems, which is equivalent to computing the controllability-based cross gramian [18] from Subsection 3.2.3. Second, for nonlinear systems; this variant does not require adjoint (dual) snapshots [110].
In the linear system setting, an empirical cross gramian is given by centering the trajectories and adjoint trajectories and average their product for the Cartesian product of the perturbation spaces. Due to the utilized duality of controllability and observability the cross gramian for linear systems is related to the empirical controllability gramian.

Definition 3.31 (Empirical Linear Cross Gramian)
*For sets E_u, R_u, Q_u the **empirical linear cross gramian** $\widehat{W_Y} \in \mathbb{R}^{N\times N}$ is given by:*

$$\widehat{W_Y} = \frac{1}{|Q_u||R_u|}\sum_{h=1}^{|Q_u|}\sum_{i=1}^{|R_u|}\sum_{j=1}^{M}\frac{1}{c_h^2}\int_0^\infty \Psi^{hij}(t)\,\mathrm{d}t,$$
$$\Psi^{hij}(t) = (x^{hij}(t) - \bar{x}^{hij})(z^{hij}(t) - \bar{z}^{hij})^\intercal \in \mathbb{R}^{N\times N}.$$

With x^{hij} being the trajectory for the input configuration $u^{hij}(t) = c_h S_i e_j \delta(t)$, and z^{hij} being the adjoint trajectory for the input configuration $v^{hij}(t) = c_h S_i e_j \delta(t)$, and associated temporal means $\bar{x}^{hij}$, $\bar{z}^{hij}$.

The linear empirical cross gramian is closely related to the bPOD approach from [184], first introduced in [228], which, as indicated in [10], is essentially computed from a (linear) "cross gramian",

$$W_{\mathrm{bPOD}} = Z^\intercal X \in \mathbb{R}^{O\times M}.$$

This product has the dimension of adjoint snapshots times primal snapshots; in the empirical gramian setting the dimension is equal to number of outputs times inputs. The reducing projections $\{U, V\}$ are then computed by an SVD of W_{bPOD} of which the left and right singular vectors are partitioned and truncated as in (3.8):

$$W_{\mathrm{bPOD}} \overset{SVD}{=} UDV,$$
$$U = XV_1D_1^{\frac{1}{2}}, \quad V = ZU_1D_1^{\frac{1}{2}}.$$

While bPOD also utilizes primal and dual snapshots, a full cross gramian is not explicitly computed yielding a smaller memory footprint.

The linear empirical cross gramian and also bPOD do not (directly) extend to nonlinear systems as no adjoint system is readily available in this case[42]. A generalization of the empirical cross gramian was introduced in [214] for SISO systems and extended in [110] to MIMO systems. This cross gramian is computable for square nonlinear systems, as it requires only state and output trajectories by combining the computational formula of the empirical controllability gramian and empirical observability gramian.

Definition 3.32 (Empirical Cross Gramian)
*For sets E_u,E_x,R_u,R_x,Q_u,Q_x, the **empirical cross gramian** $\widehat{W_X} \in \mathbb{R}^{N\times N}$ is given by:*

$$\begin{aligned}
\widehat{W_X} &= \frac{1}{|Q_u||R_u|M|Q_x||R_x|}\sum_{h=1}^{|Q_u|}\sum_{i=1}^{|R_u|}\sum_{j=1}^{M}\sum_{k=1}^{|Q_x|}\sum_{l=1}^{|R_x|}\frac{1}{c_h d_k}\int_0^\infty T_l \Psi^{hijkl}(t) T_l^\mathsf{T}\,\mathrm{d}t,\\
\Psi_{ab}^{hijkl}(t) &= f_b^\mathsf{T} T_l^\mathsf{T}\Delta x^{hij}(t) e_j^\mathsf{T} S_i^\mathsf{T}\Delta y^{kla}(t) \in \mathbb{R},\\
\Delta x^{hij}(t) &= (x^{hij}(t) - \bar{x}^{hij}),\\
\Delta y^{kla}(t) &= (y^{kla}(t) - \bar{y}^{kla}).
\end{aligned} \tag{3.25}$$

[43]

With x^{hij} and y^{kla} being the trajectory and output trajectory for the input $u^{hij}(t) = c_h S_i e_j \delta(t)$ and initial state $x_0^{kla} = d_k T_l f_a$ respectively as well as associated temporal mean state $\bar{x}^{hij}$ and temporal mean output $\bar{y}^{kla}$.

For linear systems, this empirical cross gramian reduces to the classic cross gramian from Definition 3.8 as shown in [214, 110].

Lemma 3.33 (Empirical Cross Gramian)
For asymptotically stable linear systems the empirical cross gramian equals the (linear) cross gramian.

Proof.

$$\begin{aligned}
\Psi_{ab}^{hijkl}(t) &= f_b^\mathsf{T} T_l^\mathsf{T}(\mathrm{e}^{At} B c_h S_i e_j) e_j^\mathsf{T} S_i^\mathsf{T}(C\,\mathrm{e}^{At}\, d_k T_l f_a)\\
&= c_h d_k f_b^\mathsf{T} T_l^\mathsf{T}\,\mathrm{e}^{At} BC\,\mathrm{e}^{At}\, T_l f_a\\
\Rightarrow \Psi^{hijkl}(t) &= c_h d_k T_l^\mathsf{T}\,\mathrm{e}^{At} BC\,\mathrm{e}^{At}\, T_l,\\
\Rightarrow \widehat{W_X} &= \frac{1}{|Q_u||R_u|M|Q_x||R_x|}\sum_{h=1}^{|Q_u|}\sum_{i=1}^{|R_u|}\sum_{j=1}^{M}\sum_{k=1}^{|Q_x|}\sum_{l=1}^{|R_x|}\frac{1}{c_h d_k}\int_0^\infty T_l c_h d_k T_l^\mathsf{T}\,\mathrm{e}^{At} BC\,\mathrm{e}^{At}\, T_l T_l^\mathsf{T}\,\mathrm{d}t\\
&= \int_0^\infty \mathrm{e}^{At} BC\,\mathrm{e}^{At}\,\mathrm{d}t = W_X.
\end{aligned}$$

□

Hence, the empirical cross gramian $\widehat{W_X}$ is subsequently denoted by W_X.
In [110] the empirical cross gramian is generalized to an empirical cross covariance matrix, also by centering the output trajectories around a steady-state (output) instead of the temporal (output) mean.

[42] For work on nonlinear adjoint operators see [192]

[43] $\Psi^{hijkl}(t)$ is a matrix valued function with components $\Psi_{ab}^{hijkl}(t)$ for $a, b = 1 \ldots N$.

Definition 3.34 (Empirical Cross Covariance Matrix)
*For sets E_u, E_x, R_u, R_x, Q_u, Q_x, input $\bar{u}$ during steady-state $\bar{x}$ with output $\bar{y}$, the **empirical cross covariance matrix** $\widetilde{W}_X \in \mathbb{R}^{N\times N}$ is given by:*

$$\widetilde{W}_X = \frac{1}{|Q_u||R_u|M|Q_x||R_x|}\sum_{h=1}^{|Q_u|}\sum_{i=1}^{|R_u|}\sum_{j=1}^{M}\sum_{k=1}^{|Q_x|}\sum_{l=1}^{|R_x|}\frac{1}{c_h d_k}\int_0^\infty T_l \Psi^{hijkl}(t) T_l^\intercal \,\mathrm{d}t,$$

$$\Psi_{ab}^{hijkl}(t) = f_b^\intercal T_k^\intercal \Delta x^{hij}(t) e_i^\intercal S_h^\intercal \Delta y^{kla}(t) \in \mathbb{R},$$
$$\Delta x^{hij}(t) = (x^{hij}(t) - \bar{x}^{hij}),$$
$$\Delta y^{kla}(t) = (y^{kla}(t) - \bar{y}^{kla}).$$

With x^{hij} and y^{kla} being trajectory and output trajectory for the input $u^{hij}(t) = c_h S_i e_j \odot u(t) + \bar{u}$ and initial state $x_0^{kla} = d_k T_l f_a + \bar{x}$ respectively.

Since an empirical cross gramian or covariance matrix is computed from discrete (output) trajectories, a discrete version of the previous definition follows as in [110].

Definition 3.35 (Discrete Empirical Cross Gramian / Covariance Matrix)
*For sets E_u, E_x, R_u, R_x, Q_u, Q_x, discrete time interval $[0, \mathscr{T}]$ with time step Δt, input $\bar{u}$ during steady state $\bar{x}$ with output $\bar{y}$, the **discrete empirical cross gramian** (**discrete empirical cross covariance matrix**) $w_X \in \mathbb{R}^{N\times N}$ is given by:*

$$w_X = \frac{1}{|Q_u||R_u|M|Q_x||R_x|}\sum_{h=1}^{|Q_u|}\sum_{i=1}^{|R_u|}\sum_{j=1}^{M}\sum_{k=1}^{|Q_x|}\sum_{l=1}^{|R_x|}\frac{\Delta t}{c_h d_k}\sum_{t=0}^{\mathscr{T}/\Delta t} T_l \Psi_t^{hijkl} T_l^\intercal,$$

$$\Psi_{t,ab}^{hijkl} = f_b^\intercal T_k^\intercal \Delta x_t^{hij} e_i^\intercal S_h^\intercal \Delta y_t^{kla} \in \mathbb{R},$$
$$\Delta x_t^{hij} = (x_t^{hij} - \bar{x}^{hij}),$$
$$\Delta y_t^{kla} = (y_t^{kla} - \bar{y}^{kla}).$$

With x^{hij} and y^{kla} are the discrete trajectory and discrete output trajectory for the input $u^{hij}(t) = c_h S_i e_j \odot u(t) + \bar{u}$ and initial state $x_0^{kla} = d_k T_l f_a + \bar{x}$ respectively.

Even though the empirical cross gramian is a computable cross gramian for square nonlinear systems, the cross gramian of a linear system exhibits its core property (3.6) only for (orthogonally) symmetric systems; similarly for nonlinear systems, useful results can only be expected for a certain class of systems. An extension of the concept of symmetry for nonlinear systems is given by gradient systems [124, 123, 127, 191]. A generalized definition for symmetric systems is introduced in [126], which states that a system is symmetric, if the image of the observability operator is equal to the image of the pseudo-inverse of the controllability operator:

$$\operatorname{im}\mathscr{O} = \operatorname{im}\mathscr{C}^+.$$

In this context it should be noted that in [123, 124, 125, 126, 191, 127] an alternative nonlinear cross gramian, called cross operator or cross map [79], is developed.
Lastly, the empirical cross gramian can be assembled especially efficient using generalized transpositions as summarized in [116] and in the non-symmetric case from Section 3.2.3, the observability simulations with perturbed initial state have to be computed only once.

3.3.4. Parametrized Systems

For parametrized systems a reduced order model should (approximately) preserve the parameter dependency [23]. In [218], an approach for robust reduction of (linearly) parametrized systems of the form:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + F\theta, \\ y(t) &= Cx(t), \end{aligned} \tag{3.26}$$

with an additional source matrix $F \in \mathbb{R}^{N\times P}$ and parameter $\theta \in \mathbb{R}^P$ is presented. By treating the parameters θ as constant inputs $u_\theta(t) = \theta$, the system can be rewritten as:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + \begin{pmatrix} B & F \end{pmatrix} \begin{pmatrix} u(t) \\ u_\theta(t) \end{pmatrix}, \\ y(t) &= Cx(t). \end{aligned}$$

The associated empirical controllability covariance matrix is computed as an (arithmetic) average over all empirical controllability covariance matrices for each location in the input perturbation space $\Omega_u = E_u \times R_u \times Q_u$. With a discretized parameter-space,

$$\Theta_h = \{\theta_1 \dots \theta_n\},$$

expressed as perturbations around a nominal parameter $\bar{\theta}$, the input perturbation space $\Omega_{u\cup\Theta} := \Omega_u \cup \Theta_h$ is then enlarged. Alternatively, one can extend the perturbation spaces by an additional factor to the perturbation space $\Omega_\Theta := \Omega \times \Theta_h$ which then yields:

$$W_C = \frac{1}{|\Theta_h|} \sum_{i=1}^{|\Theta_h|} W_C(\theta_i).$$

A similar argument can be made for the empirical observability gramian as controllability of the adjoint system,

$$W_O = \frac{1}{|\Theta_h|} \sum_{i=1}^{|\Theta_h|} W_O(\theta_i),$$

and for the empirical cross gramian [113] that also requires input perturbations:

$$W_X = \frac{1}{|\Theta_h|} \sum_{i=1}^{|\Theta_h|} W_X(\theta_i).$$

For the discretization of high-dimensional parameter-spaces, techniques such as sparse grids (see for example: [15]) should be considered, which also holds for the construction of the other factors of the perturbation sets.
It should also be noted that for general (nonlinear) parametrizations, the parameters commonly do not act as input sources.

3.4. Parameter Reduction

Reducing the parameter-space dimension requires to identify the dominant parameter subspace, which can be accomplished by parameter identification [92]. Essentially, this approach employs the parameter reduction based on sensitivity analysis from [217]. Given a covariance matrix $\omega \in \mathbb{R}^{P\times P}$ on the parameters $\theta \in \mathbb{R}^P$, a (linear) identification of the parameters is obtained from a PCA. An SVD of ω yields the variances (singular values) $\sigma_1 \geq \cdots \geq \sigma_P$ and principal components (singular vectors) $\pi_1, \ldots, \pi_n$ to the symmetric positive semi-definite covariance matrix,

$$\omega \overset{SVD}{=} \Pi \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_P \end{pmatrix} \Pi^\intercal,$$
$$\Pi = \begin{pmatrix} \pi_1 & \ldots & \pi_P \end{pmatrix}.$$

From such a decomposition each parameter component is identifiable by its contribution to the associated singular vector [83].
The overall identifiability of the parameters, for example in the context of optimal actuator and sensor placement [201, 204, 216], can be determined by several measures[44] based on the singular values of the covariance matrix presented in [224]; for instance the trace (norm[45]) of ω,

$$\|\omega\|_* = \mathrm{tr}(\omega). \tag{3.27}$$

By sorting the parameters by their identifiability, the parameter-space dimension can be reduced by truncating the least identifiable (linear combinations of) parameters. A parameter projection (2.13) of such a parameter covariance is also obtained by the singular value decomposition of ω. Treating the principal components Π as a one-sided projection, that is partitioned and truncated similar to the state-space direct truncation (3.8),

$$\Pi = \begin{pmatrix} \Pi_1 & \Pi_2 \end{pmatrix} \to \Pi_1 \in \mathbb{R}^{P\times p},$$

then acts as a Galerkin projection. Due to the relation of the SVD to the L_2-norm this procedure excludes the (linear combinations of) parameters that carry the least energy, which is in line with the state reduction also based on the energy transfer of the states.
To obtain a covariance matrix on the parameters, the gramian-based state reduction (identification and truncation) is applied. A variant of each of the previously introduced (empirical) system gramians can also be utilized for parameter identification, which is based on the relation of the empirical gramians to the covariance (3.21). Following, a controllability-based, an observability-based and a cross-gramian-based approach to parameter identification and parameter reduction is presented.
To this end the parameters are treated as inputs or states; in case of $R_u = \{-\mathbb{1}_M, \mathbb{1}_M\}$ and $R_x = \{-\mathbb{1}_N, \mathbb{1}_N\}$ and general parametrizations one has to ensure energy is introduced into the system for example by some additional steady-state input [134, 2.2].

[44] Further measures are among others: determinant $\det(\omega)$, logarithmic determinant $\log\det(\omega)$, smallest eigenvalue $\lambda_{\min}(\omega)$, spectral radius $|\lambda_{\max}(\omega)|$ or trace of the inverse $\mathrm{tr}(\omega^{-1})$.
[45] See Appendix A.3.

3.4.1. Controllability-Based Parameter Reduction

For a linear system with linear parameter dependency (3.26), the vector field of this system can be additively decomposed,

$$\dot{x}(t) = Ax(t) + Bu(t) + F\theta = Ax(t) + Bu(t) + \sum_{i=1}^{P} F_i \theta_i, \quad F = \begin{pmatrix} F_1, & \ldots, & F_P \end{pmatrix},$$

with $F_i \in \mathbb{R}^{N\times 1}$, and as shown in [217], due to the superposition in linear systems, the controllability gramian can be decomposed similarly:

$$\begin{aligned} W_C &= W_{C,0} + \sum_{i=1}^{P} W_{C,i}, \\ W_{C,0} &= \int_0^\infty \mathrm{e}^{At} BB^\mathsf{T} \mathrm{e}^{A^\mathsf{T} t} \,\mathrm{d}t, \\ W_{C,i>0} &= \int_0^\infty \mathrm{e}^{At} F_i F_i^\mathsf{T} \mathrm{e}^{A^\mathsf{T} t} \,\mathrm{d}t. \end{aligned}$$

Similar to [199, 203, 214, 215], in [109], the parameter sensitivity is measured by the trace of the sub-controllability gramians $W_{C,i}$ as in (3.27). The trace of the controllability gramian can be interpreted as the average controllability [216] and is used for example in optimal actuator placement. This sensitivity information is compiled into a matrix.

Definition 3.36 (Sensitivity Gramian)
*The **sensitivity gramian** W_S is defined as the diagonal matrix composed of the traces of the parameter sub-controllability gramians:*

$$W_S := \begin{pmatrix} \operatorname{tr}(W_{C,1}) & & 0 \\ & \ddots & \\ 0 & & \operatorname{tr}(W_{C,P}) \end{pmatrix}.$$

For models with many homogeneous parameter components it may be helpful to center the sensitivity gramian's diagonal around its root mean square (RMS),

$$\overline{W}_{S,ii} = W_{S,ii} - \sqrt{\operatorname{tr}(W_S W_S^\mathsf{T})}. \tag{3.28}$$

Also an empirical sensitivity gramian can be computed for nonlinear systems with vector fields of the form:

$$f(x,u,\theta) = f_0(x,u) + \sum_{i=1}^{P} f_i(x,\theta_i).$$

Due to the nonlinearity, the superposition does not hold and the controllability gramian is only approximated,

$$W_C \approx W_{C,0} + \sum_{i=1}^{P} W_{C,i}.$$

In case the parameter θ does not act as an input source, an additional input perturbation has to be applied.

3.4.2. Observability-Based Parameter Reduction

In [83, 82], the observability of states is expanded to observability of parameters for purposes of parameter identification; in the related context of optimal sensor placement a similar approach is used in [201, 203]. Given a parametrized system (2.5), the parameters θ can be treated as constant states; this yields an augmented system of the form:

$$\begin{aligned} \dot{\breve{x}}(t) &= \begin{pmatrix} \dot{x}(t) \\ \dot{\theta}(t) \end{pmatrix} = \begin{pmatrix} f(x(t), u(t), \theta) \\ 0 \end{pmatrix}, \\ y(t) &= g(x(t), u(t), \theta), \\ \breve{x}_0 &= \begin{pmatrix} x_0 \\ \theta_0 \end{pmatrix}, \end{aligned} \tag{3.29}$$

with the initial state augmented by the (nominal) parameter. Due to the trivial vector field of the parameter states, the initial parameter states are the steady (parameter) state and the stability of the system is not affected [82, Ch. 4.3.3, Remark 4]. For such an augmented system, an augmented observability gramian can be computed, which is composed of the following block matrices:

$$\breve{W}_O = \left(\begin{array}{c|c} W_O & W_M \\ \hline W_M^\intercal & W_P \end{array}\right) \in \mathbb{R}^{(N+P)\times(N+P)}.$$

The upper left block $W_O \in \mathbb{R}^{N\times N}$ corresponds to the non-augmented system's observability gramian, the lower right block $W_P \in \mathbb{R}^{P\times P}$ encodes the observability of the parameters and the blocks $W_M \in \mathbb{R}^{N\times P}$, $W_M^\intercal \in \mathbb{R}^{P\times N}$ are mixed blocks. The parameter identifiability can then be extracted from this augmented observability gramian [83].

Definition 3.37 (Identifiability Gramian)
The Schur-complement of block W_O of an augmented observability gramian $\breve{W}_O$ is called ***identifiability gramian*** *W_I:*

$$W_I := W_P - W_M^\intercal {W_O}^{-1} W_M. \tag{3.30}$$

The identifiability gramian may also be approximated by the lower right block of the augmented observability gramian:

$$W_I \approx W_P.$$

Notably, the empirical observability gramian of the augmented system $\breve{W}_O$ is closely related to the Fisher information matrix $\mathscr{F}$, if the perturbations are restricted to the parameter-states, as shown in [201],

$$W_P = \int_0^\infty \left(\frac{\partial y(t)}{\partial \theta}\right)^\intercal \left(\frac{\partial y(t)}{\partial \theta}\right) \mathrm{d}t = \mathscr{F}.$$

3.4.3. Cross-Gramian-Based Parameter Reduction

The parameter-state augmented system in (3.29) is still square for square systems and conserves symmetry (in the linear case) since the matrices A, B, C are augmented with zeros. Hence, a cross gramian can be computed for such systems, too [110].

Definition 3.38 (Joint Gramian)
The cross gramian of an augmented system is called ***joint gramian*** W_J*:*

$$W_J := \left(\begin{array}{c|c} W_X & W_M \\ \hline W_m & W_P \end{array}\right) \in \mathbb{R}^{(N+P)\times(N+P)}.$$

The upper left block of the joint gramian is the cross gramian of the non-augmented system. As opposed to the observability gramian, the cross gramian is generally not symmetric,

$$W_X \neq W_X^\intercal \Rightarrow W_J \neq W_J^\intercal \Rightarrow W_M \neq W_m^\intercal.$$

Since the augmented "parameter-states" are directly and indirectly independent from the input $u(t)$ and thus uncontrollable, the rows of the joint gramians relating to these states are zero,

$$\|W_m\| = \|W_P\| = 0.$$

Hence, using the Schur-complement as for the identifiability gramian from Definition 3.37 to extract information on the parameters has a trivial result:

$$W_P - W_m W_X^{-1} W_M = 0 - 0\, W_X^{-1} W_M = 0.$$

Yet, the information on the parameter identifiability is encoded in the mixed matrix W_M. To extract this identifiability information one can use the symmetric part of the joint gramian,

$$\overline{W}_J := \frac{1}{2}(W_J + W_J^\intercal).$$

As for the identifiability gramian, the cross-gramian-based identifiability information is then retrieved by the Schur-complement.

Definition 3.39 (Cross-Identifiability Gramian)
The Schur-complement of the upper left block $\overline{W}_X = \frac{1}{2}(W_X + W_X^\intercal)$ *of the symmetric part of the joint gramian* $\overline{W}_J$ *is called* ***cross-identifiability gramian*** $W_{\bar{I}}$*:*

$$W_{\bar{I}} := -\frac{1}{2} W_M^\intercal \overline{W}_X^{-1} W_M.$$

For efficient computation of an approximation to the inverse symmetric part of the cross gramian $\overline{W}_X^{-1}$, see the approach in [230, 116], which is also summarized in Appendix A.4. Furthermore, the empirical joint gramian can also be computed for non-symmetric systems, since the non-symmetric extension to the cross gramian in Section 3.2.3 also applies to the joint gramian.

3.4.4. Nonlinear Error Indicators

For general nonlinear systems (2.5), the $\mathscr{H}_\infty$-error bounds (3.9), (3.10) and $\mathscr{H}_2$-error indicators (3.13), (3.14) cannot be used directly, since their justification utilizes the linear structure of the underlying control system. The $\mathscr{H}_2$-error indicators require the linear system components B, C for an a-priori computation (3.11), (3.12). Yet, for a nonlinear **control-affine** system of the form,

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

approximate components $\widetilde{B}$ and $\widetilde{C}$ can be obtained by linearization [105, Ch. 2]. Together with the empirical gramians, an L_∞-error indicator can be approximated as in (3.13) by the balanced and truncated parts of the linearized input $\widetilde{B}_2$, output matrices $\widetilde{C}_2$ and empirical system gramian W_{22}:

$$\|y - y_r\|_{\mathrm{L}_\infty} \approx \sqrt{\operatorname{tr}(\widetilde{C}_2 W_{22} \widetilde{B}_2)}.$$

Since the L_2-error bound is computed only from the HSVs, likewise an error indicator can be computed by the approximate HSVs obtained from empirical gramians as in (3.9),

$$\|y - y_r\|_{\mathrm{L}_2} \lesssim 2 \sum_{i=n+1}^{N} \sqrt{\lambda_i(W_C W_O)} = 2 \sum_{i=n+1}^{N} |\lambda_i(W_X)|.$$

Similarly, the same approach can be used for an L_1-error indicator from (3.15) for the impulse response,

$$\|y - y_r\|_{\mathrm{L}_1} \lesssim 4(N+n) \sum_{i=n+1}^{N} \sqrt{\lambda_i(W_C W_O)} = 4(N+n) \sum_{i=n+1}^{N} |\lambda_i(W_X)|.$$

Another error indicator, used for example in [45], is the information content or relative energy fraction of the total transferred energy represented by the trace norm of the Hankel operator H and the Hankel operator of the reduced order model H_r,

$$\varepsilon_{e,x} = \frac{\|H_r\|_*}{\|H\|_*} = \frac{\sum_{i=1}^{n} \sigma_i(H_r)}{\sum_{i=1}^{N} \sigma_i(H)}.$$

The quality of the ROM is then indicated by the disparity from one. By using the empirical gramians to obtain the Hankel singular values, the energy transfer mismatch $\varepsilon_{e,x}$ can be computed also for nonlinear systems. Similarly, for the parameter gramians $\omega \in \{W_S, W_I, W_{\tilde{I}}\}$ and the reduced parameter gramians ω_r this error indicator can also be utilized,

$$\varepsilon_{e,\theta} = \frac{\|\omega_r\|_*}{\|\omega\|_*}.$$

As for the error bounds of linear systems, also for nonlinear systems and parameter error indicators, more conservative but more efficient error bounds can be computed by approximating singular values of the tail $\sigma_i < \sigma_r$ by $\sigma_i \approx \sigma_r$ for $r < i \leq N$.

3.4.5. State and Parameter Empirical Gramian Interrelation

The presented empirical gramians are all interrelated (see Figure 3.2). From the controllability operator $\mathscr{C}$ the controllability gramian W_C is derived, and from the observability operator $\mathscr{O}$ the observability gramian W_O. The linear cross gramian W_Y fuses both, controllability and observability operators, and the cross gramian W_X uses the concept of the linear cross gramian, but utilizes the formulation of the controllability and observability gramians. Derived from the cross gramian, the non-symmetric cross gramian W_Z enables input-output energy-based state reduction for non-square or non-symmetric systems. Based on the parameter dependent fraction of average controllability, the sensitivity gramian W_S allows a sensitivity analysis on parametric systems. The identifiability gramian W_I extends observability from states to parameters using the observability gramian. Related to the identifiability gramian is the cross-gramian-based joint gramian W_J which provides the cross-identifiability gramian $W_{\hat{I}}$ for parameter identification.

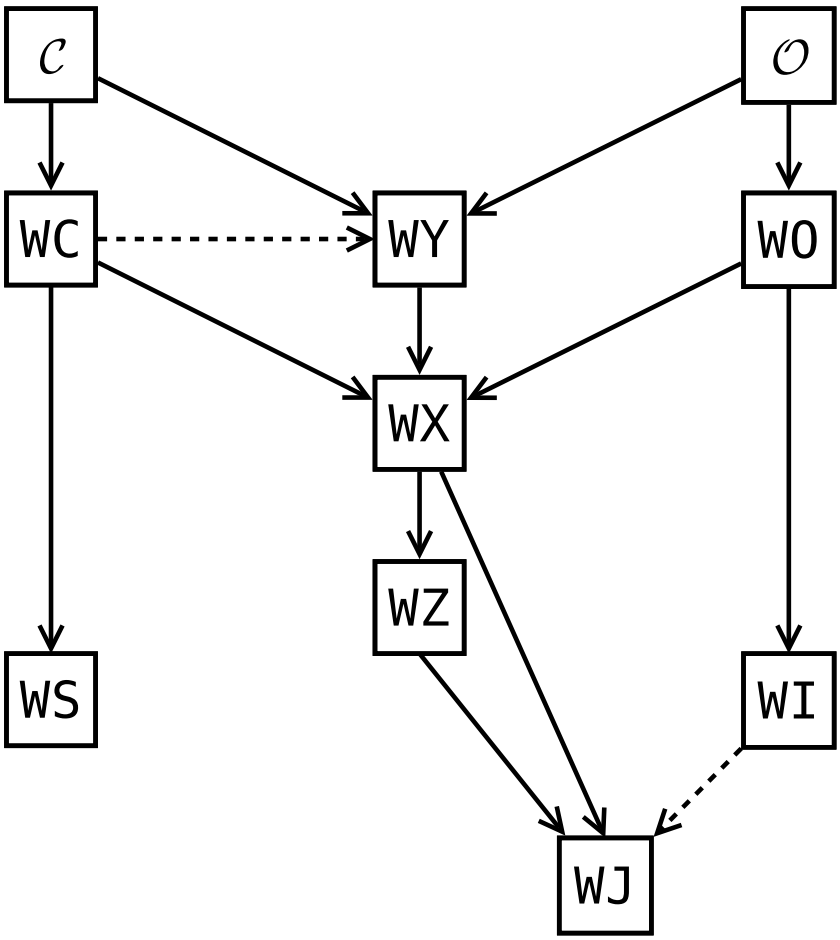


Figure 3.2.: *Overview of the relationships between the (empirical) system gramians.*

3.5. Combined Reduction

Each of the three previously presented parameter-based (empirical) gramians induces an approach for combined state and parameter reduction [110]. While other than the following combinations of state and parameter reduction methods are surmisable, this selection exhibits the advantage that along with the parameter-space gramian, a state-space gramian is computed. These three methods allow a combined reduction as described in (2.14).

3.5.1. Controllability-Based Combined Reduction

The (empirical) sensitivity gramian provides a parameter reducing projection by an SVD; and an (approximate) controllability gramian is computed in the process. Together with an observability gramian, using balanced truncation, a (two-sided) state reducing projection is obtained. This controllability-based combined reduction is summarized in Algorithm 1,

Algorithm 1: Controllability-Based Combined Reduction

Compute W_S
Extract W_C
Compute W_O
$\{U, V\} = BT(W_C, W_O)$
$\{\Pi, \Pi^\intercal\} = DT(W_S)$

and utilizes the input-to-output coherence to identify the dominant state-subspace and the input-to-state coherence to identify the dominant parameter-subspace.

3.5.2. Observability-Based Combined Reduction

The (empirical) identifiability gramian provides a parameter reducing projection by an SVD; and an observability gramian is computed in the process. Together with a controllability gramian, using balanced truncation a (two-sided) state reducing projection is obtained. This observability-based combined reduction is summarized in Algorithm 2,

Algorithm 2: Observability-Based Combined Reduction

Compute W_C
Compute $\check{W}_O$
Extract W_O
Extract W_I
$\{U, V\} = BT(W_C, W_O)$
$\{\Pi, \Pi^\intercal\} = DT(W_I)$

and utilizes the input-to-output coherence to identify the dominant state-subspace and the state-to-output coherence to identify the dominant parameter-subspace.

3.5.3. Cross-Gramian-Based Combined Reduction

The (empirical) joint gramian provides both: a parameter reducing projection through an SVD of the cross-identifiability gramian, and a (one-sided) state reducing projection through an SVD of the cross gramian. This cross-gramian-based combined reduction is summarized in Algorithm 3,

Algorithm 3: Cross-Gramian-Based Combined Reduction

Compute W_J
Extract W_X
Extract $W_{\hat{I}}$
$\{U, U^\intercal\} = DT(W_X)$
$\{\Pi, \Pi^\intercal\} = DT(W_{\hat{I}})$

and also utilizes the input-to-output coherence to identify the dominant state-subspace and the state-to-output coherence to identify the dominant parameter-subspace.

3.5.4. The Case for the Joint Gramian

The (empirical) joint gramian contains information on the states in terms of controllability and observability as well as on the (observability-based) identifiability of the parameters. Using the cross-gramian-based joint gramian has multiple advantages. First the resulting ROM is stable if the underlying model is (locally) asymptotically stable. Second, the cross-identifiability gramian can be (computationally) a memory economical alternative to the identifiability gramian. The joint gramian W_J is a rectangular matrix of dimension $N \times (N+P)$ and the augmented observability gramian $\check{W}_O$ a symmetric matrix of dimension $(N+P)\times(N+P)$, and needs to store $\frac{1}{2}(N+P)(N+P+1)$ elements. Hence, for systems with $N < P+1$, the joint gramian requires less memory. Third, the cross-gramian-based combined state and parameter reduction requires only two truncated SVDs (TSVD), as opposed to one TSVD and a balancing procedure which warrants at least the complexity of a TSVD.
It should be noted that the empirical gramians (and thus the empirical joint gramian) are explicitly targeted at nonlinear systems. For linear systems, specific algorithms can be employed to directly compute low-rank representations of the system gramians. For example the alternating direction implicit (ADI) iteration [187, 188] for low-rank approximations of the Cholesky factors of the controllability gramian and observability gramian, and the implicit restarted Arnoldi method [209, 210] or matrix sign function [14] for a low-rank cross gramian approximation. Yet, for nonlinear systems the empirical gramians can capture more accurately the dynamic behavior over a designated operating region than a linearization [200, 51] or a linear model [101].
Lastly, the cross-gramian-based joint gramian can principally handle high-dimensional parameter-spaces $P \gg 1$ for either pMOR or (combined state and) parameter reduction which is a rare trait, especially in combination with applicability to nonlinear systems.

4. Optimization-Based Combined Reduction

Contents

Optimization-based combined state and parameter reduction is an iterative model reduction approach. While the parameter-space reduction is a greedy method based upon [152, 151], the state-space reduction is related to POD.
Central to this technique is the selection of the dominant parameter-subspace, which determines the sampling points for the associated reduced state base. Alongside an extension to the basic optimization-based combined reduction targeted at inverse problems, a variant for nonlinear systems is considered.

4.1. Parameter Reduction

The core idea of the optimization-based parameter-space reduction is the iterative assembly of a parameter projection $\Pi \in \mathbb{R}^{p \times p}$ by incorporating parameter base vectors that dominantly affect the input-output behavior. In the I-th iteration, the next projection Π_{I+1} is obtained from orthogonalizing a designated parameter base vector θ_{I+1}, selected from the parameter-space, to the current projection Π_I:

$$\Pi_{I+1} = \{\Pi_I \cup (\theta_{I+1} - \Pi_I \Pi_I^\intercal \theta_{I+1})\}. \tag{4.1}$$

The parameter projection is initialized by some nominal parameter θ_0, for example the centroid or a parameter selected from additional (prior) knowledge:

$$\Pi_0 = \theta_0.$$

To obtain information on the identifiability of parameter components, the nominal parameter θ_0 is perturbed and the sampled input-output behavior is analyzed. Yet, sampling in a high-dimensional parameter-space on a uniform parameter grid, is computationally infeasible due to the curse of dimensionality. Non-uniform grids, such as sparse grids [15, 39], can be used to select parameter sampling points more efficiently. An alternative approach to predefined grids for selection of θ_{I+1} is an adaptive sampling strategy based on the current coverage, which is described over the next sections.

4.1.1. Model Constrained Optimization

Since the greedy sampling strategy will be formulated as an optimization problem, first, model constrained optimization is briefly introduced; for further information on this topic see [22] and [29].
Model constraint optimization encompasses optimization problems of which the associated cost function requires evaluations of an underlying (dynamic) model. A starting point is the linear state-space system with linear parametrization,

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + F\theta, \\ y(t) &= Cx(t). \end{aligned} \tag{4.2}$$

Given a dataset y_d to which the model's parameters θ are to be tuned, the associated optimization problem is formulated in the (energy-minimizing) least-squares sense as:

$$\begin{aligned} \theta_d &= \underset{\theta \in \mathbb{R}^P}{\arg\min} \| y(\theta) - y_d \|_{\mathrm{L}_2}^2, \\ &\text{subject to:} \\ &\dot{x}(t) = Ax(t) + Bu(t) + F\theta, \\ &y(t) = Cx(t), \\ &x(0) = x_0, \end{aligned}$$

and represents the optimization constrained by the model (4.2). Without loss of generality, over the course of this section $u(t) = 0$ and $x_0 = 0$ is assumed. Then, this formulation can be simplified, due to the linear nature of the model, using the convolution operator (3.1):

$$\begin{aligned} \theta_d &= \underset{\theta \in \mathbb{R}^P}{\arg\min} \| S(\theta) - y_d \|_{\mathrm{L}_2}^2, \\ &\text{subject to:} \\ &S(\theta) = \int_0^\infty C\, \mathrm{e}^{A\tau} F\theta \, \mathrm{d}\tau. \end{aligned} \tag{4.3}$$

Since the convolution operator (3.1) is bounded [6, Ch. 5.2], S admits a singular value decomposition,

$$S(\theta) \overset{\text{SVD}}{=} \sum_{i=1}^{\infty} \sigma_i \langle \theta, v_i \rangle u_i, \tag{4.4}$$

with which, by the pseudo-inverse operator, the least-squares solution can be obtained:

$$\theta_d = S^+(y_d) = \sum_{i=1}^{\infty} \frac{1}{\sigma_i} \langle y_d, u_i \rangle v_i.$$

An approximate solution is then given by truncating the sum in the pseudo-inverse operator, assuming the singular values are sorted in descending order. This direct solution is only exemplarily demonstrated for the linearly parametrized linear control system. Yet, other parametrizations of the linear control system model also require the boundedness of the mapping from parameters to outputs given inputs and initial state.

4.1.2. Best Approximating Subspace

While the model constraint optimization aims to minimize the norm of a residual, a best approximating subspace can be obtained by maximizing the norm of a residual. For the linear state-space system with linear parametrization (3.26), an associated parameter-space reduced system is then given by:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) + F\,\Pi_I\,\theta_r,\\ y_r(t) &= Cx(t),\end{aligned}$$

with a Galerkin parameter projection $\Pi_I \in \mathbb{R}^{P\times I}$ and $\theta_r \in \mathbb{R}^I$. Opposed to the (least-squares) model constraint optimization (4.3), which seeks a parameter that minimizes the energy of the error-output-trajectory between model and data, a best approximating subspace (in the energy-norm sense) aims to maximize the error between the original model and the reduced model,

$$\begin{aligned}\theta_{I+1} &= \operatorname*{arg\,max}_{\theta\in\mathbb{R}^P} \|y(\theta) - y_r(\Pi_I^\intercal\theta)\|_{\mathrm{L}_2}^2\\ &= \operatorname*{arg\,max}_{\theta\in\mathbb{R}^P} \|y(\theta) - y(\Pi_I\Pi_I^\intercal\theta)\|_{\mathrm{L}_2}^2.\end{aligned}$$

Subsequently, in each iteration, a resulting parameter base vector θ_{I+1} is incorporated into the parameter projection (4.1), which represents the currently worst approximated subspace. As for the model constraint optimization, this maximization problem can be rewritten in the fully linear setting to:

$$\begin{aligned}\theta_{I+1} &= \operatorname*{arg\,max}_{\theta\in\mathbb{R}^P} \|S(\theta) - S(\Pi_I\Pi_I^\intercal\theta)\|_{\mathrm{L}_2}^2\\ &= \operatorname*{arg\,max}_{\theta\in\mathbb{R}^P} \|S(\theta - \Pi_I\Pi_I^\intercal\theta)\|_{\mathrm{L}_2}^2.\end{aligned}$$

By the same argument as in (4.4) an SVD exists and the first left singular vector u_1 associated to the largest singular value σ_1 poses an approximate solution to the (one-dimensional) energy maximization problem:

$$u_1 \approx \operatorname*{arg\,max}_{\theta\in\Theta} \|S(\theta - \Pi_I\Pi_I^\intercal\theta)\|_{\mathrm{L}_2}^2.$$

This cost functional attains its maximum for $\|\theta\| \to \infty$. Hence, a compact subspace $\Theta \subset \mathbb{R}^P$ consisting only of elements with bounded length is chosen,

$$\Theta := \{\theta \in \mathbb{R}^P : \|\theta\|_2 \leq 1\};$$

from which the (normalized) energy-maximal parameter location is selected as next parameter base vector.
Like for the model constraint optimization, the best approximated subspace method requires the boundedness of the solution operator representing the mapping from parameters to outputs.
Summarizing, in each iteration, the left singular vector associated to the next largest singular value is incorporated into the parameter base Π_I. This adaptive scheme leads to the greedy sampling approach.

4.1.3. Greedy Sampling

Instead of sampling over a (high-dimensional) grid of the discretized parameter-space Θ_h, an adaptive sampling strategy can be employed, with a heuristic to select the next sampling location in the parameter-space. In the special case of **greedy sampling**, a location in parameter-space is sought that maximizes the error between output trajectories associated to the full and reduced order parameter. This greedy sampling method in the context of model reduction was introduced in [225], extended in [31] and adapted in [151]. A related greedy technique is already proposed in [33] to obtain an optimal basis from a given set of vectors. In general, a greedy approach seeks a locally optimal result, and in terms of a least-squares optimization problem this variant of the greedy algorithm can be written as:

$$\begin{aligned} \theta_{I+1} &:= \arg\max_{\theta\in\Theta} \|y(\theta) - y(\Pi_I \Pi_I^\intercal \theta)\|_{\mathrm{L}_2}^2, \\ &\text{subject to:} \\ &\dot{x}(t) = A(\theta)x(t) + B(\theta)u(t) + F(\theta), \\ &y(t) = C(\theta)x(t), \\ &x(0) = x_0, \\ &u(t) \in \mathrm{L}_2, \\ &\Pi_I^\intercal \Pi_I = \mathbb{1}, \end{aligned} \tag{4.5}$$

for a suitable parameter-space $\Theta \subset \mathbb{R}^P$. The resulting parameter θ_{I+1} then extends the reduced parameter base Π_I by orthogonalization as in (4.1). Locally, the resulting base Π_{I+1} minimizes the maximal L_2 output error between FOM and ROM (2.16d). More generally, the optimization problem (4.5) can be written using an objective function J:

$$\begin{aligned} J(\theta; \Pi_I) &:= \|y(\theta) - y(\Pi_I \Pi_I^\intercal \theta)\|_{\mathrm{L}_2}^2, \\ \theta_{I+1} &= \arg\max_{\theta\in\Theta} J(\theta; \Pi_I). \end{aligned} \tag{4.6}$$

Practically, in each iteration the equivalent minimization problem can be solved:

$$\theta_{I+1} = \arg\min_{\theta\in\Theta} -J(\theta; \Pi_I). \tag{4.7}$$

The solution to this ODE-constrained (or PDE-constrained) optimization problem usually requires numerous simulations of the constraining system. It may be advisable to use a weak greedy algorithm which replaces the error between the FOM and ROM by an error estimator $r : \mathbb{R}^P \to \mathbb{R}$,

$$\tilde{J}(\theta; \Pi_I) := \|r(\theta; \Pi_I)\|_{\mathrm{L}_2}^2.$$

As a starting parameter to the optimization algorithm, it is suggested in [31, Sec. 3.3] to use a random initial guess which lies in the null space of the last iteration's parameter projection Π_{I-1}.

Greedy Sampling Properties

The greedy sampling expresses the parameter-space reduction as an optimization problem. An optimization problem is called **well-posed** if the following three Hadamard requirements are met:

1. existence of a solution,
2. uniqueness of the solution,
3. stability of the solution[46].

If any of the above criteria is not fulfilled, the problem is considered **ill-posed**. The optimization problem, by which the greedy sampling is performed, is generally not well-posed which is illustrated next.
In [31, Sec. 4.1] the existence of a (global) maximizer is shown which is restated here:

Theorem 4.1 (Existence of Maximizer)
If $J(\theta) : \Theta \subset \mathbb{R}^P \to \mathbb{R}$ is continuous and Θ is compact in $\mathbb{R}^P$ then there exists a maximum $\bar{z}$ such that $J(\bar{z}) > J(z) \;\; \forall z \in \Theta$, which holds for each greedy sampling iteration.

Proof.
This is a direct consequence of the (Weierstrass) extreme value theorem. □

Hence, in each iteration a local maximizer z_I can be found, but it is most likely **not unique**. This means that the underlying optimization of the greedy sampling approach is considered to be ill-posed. The next section presents a counter measure to ensure uniqueness.
Secondly, it is shown in [31, Sec. 4.2] that the iterative adaptive refinement of the reduced order parameter-space is improving with each iteration, and in case of linear state-space systems, maximizers are increasing over subsequent iterations compared to previously obtained solutions:

$$J(\theta_I) < J(\theta_{I+1}) < \dots.$$

Thus, independent from the parametrization of the underlying model the greedy sampling technique can be utilized to assemble a parameter-space base. Yet, a rate of convergence can generally not be determined.
For low-dimensional parameter-spaces, an analysis of the convergence of the greedy sampling strategy, from the reduced basis point of view, of parametrized PDEs is given in [25], based on the Kolmogorov width[47] of the greedily selected low-dimensional subspace compared to the best approximating subspace of the low-order dimension. Using the Kolmogorov width, a decay rate for the convergence of the (weak) greedy algorithm is derived in [25].

[46] Stability refers in this context to a continuous dependence of the solution on the parameters.
[47] The optimality of the reduced bases obtained from POD and balanced truncation is analyzed in [50]; these bases are shown to be optimal in the Hilbert-Schmidt-norm.

Tikhonov Regularization

The previous argument characterized the solution of the optimization problem abstractly using the SVD. Practically, a constrained (ill-posed) optimization problem has to be solved, which may pose computational difficulties due to the constraint on the parameter norm and the ill-posedness. Alternatively, instead of the previous hard constraint, a soft constraint on the parameter length can be imposed using a penalty term, which leads to regularization. The following definition is adapted from [60].

Definition 4.2 (Regularization Operator)
A family of continuous operators $\mathscr{R}_\beta$ is called regularization operator, if for all $\theta \in \Theta$ there exists a map $\beta = \beta(\delta, \theta_\beta)$ such that:

$$\limsup_{\delta \to 0} \{\|\mathscr{R}_{\beta(\delta,\theta_\delta)}(\theta_\delta) - S(\theta)\| : \|\theta_\delta - \theta\| \leq \delta\} = 0,$$
$$\limsup_{\delta \to 0} \{\beta(\delta, \theta_\delta) : \|\theta_\delta - \theta\| \leq \delta\} = 0.$$

The quadratic constrained optimization problem:

$$\theta_{I+1} = \arg\max_{\theta \in \Theta} \|S(\theta)\|^2_{\mathrm{L}_2},$$

can be formulated as a regularized unconstrained optimization problem by the use of Lagrange multipliers as derived in [59]:

$$\theta_{I+1} = \arg\max_{\theta \in \mathbb{R}^p} \|S(\theta)\|^2_{\mathrm{L}_2} - \mathscr{R}_{\beta_2},$$

with the regularization operator $\mathscr{R}_{\beta_2}$ and multiplier $\beta_2 > 0$. Generally, the regularization term counters the maximization by decreasing the to-be-maximized cost functional for larger parameter norms and thus prefers solutions with smaller norms.
If the optimization problem is in standard form, so that the constraint is given by $\|\theta\|_2 \in \mathbb{R}^{>0}$, then the regularization operator is given by:

$$\mathscr{R}_{\beta_2} = \beta_2 \|\theta\|^2_2.$$

This regularization method is also called L_2**-regularization**.
In case the constraint has the form $\|\Gamma\theta\|_2 \in \mathbb{R}^{>0}$ for a symmetric positive-definite matrix Γ, the regularization operator has the form:

$$\mathscr{R}_{\beta_2,\Gamma} = \beta_2 \|\Gamma\theta\|^2_2, \tag{4.8}$$

and is called **Tikhonov regularization**, which makes the L_2-regularization operator a special case of the Tikhonov regularization operator with $\Gamma = \mathbb{1}$.
Overall, the regularization has a dual purpose: ensuring the uniqueness of a greedy sample and enforcing the constraint on the parameters.

4.2. State Reduction

Similar to the gramian-based methods in Chapter 3, the optimization-based state-space reduction also aims to minimize the energy in the error between trajectories of the full and the reduced order model. To this end a least-squares optimization can be employed to assemble a state reducing Galerkin projection U using a residual $r(U)$,

$$\hat{U} = \underset{U \in \mathbb{R}^{N \times n}}{\arg\min} \|r(U)\|_{\mathrm{L}_2}^2.$$

The residual $r(U)$ is given by the error between the full and reduced model (or an error estimator), which can be based in this control system setting on one of the following mappings:

- from inputs to states:

$$\begin{aligned} \xi : \mathrm{L}_2^M(\mathbb{R}) &\to \mathrm{L}_2^N(\mathbb{R}), \\ u(t) &\mapsto x(t), \end{aligned}$$

  which corresponds to the convolution operator (2.7) with an output matrix $C = \mathbb{1}$.

- from states to outputs:

$$\begin{aligned} \eta : \mathbb{R}^N &\to \mathrm{L}_2^O(\mathbb{R}), \\ x_0 &\mapsto y(t), \end{aligned}$$

  which corresponds to the convolution operator (2.7) with an input matrix $B = 0$.

- from inputs to outputs:

$$\begin{aligned} S : \mathrm{L}_2^M(\mathbb{R}) &\to \mathrm{L}_2^O(\mathbb{R}), \\ u(t) &\mapsto y(t), \end{aligned}$$

  which corresponds to the convolution operator (2.7).

These mappings bear not by accident resemblance to the controllability, observability and Hankel operator as illustrated in Table 4.1. Furthermore, these mappings relate to the linear system gramians as described in [6, Ch. 4.3].

Input-to-Output	$S : \mathrm{L}_2^M(\mathbb{R}) \to \mathrm{L}_2^O(\mathbb{R})$	$H : \mathrm{L}_2^M(\mathbb{R}^-) \to \mathrm{L}_2^O(\mathbb{R}^+)$	Hankel
Input-to-State	$\xi : \mathrm{L}_2^M(\mathbb{R}) \to \mathrm{L}_2^N(\mathbb{R})$	$\mathscr{C} : \mathrm{L}_2^M(\mathbb{R}^-) \to \mathbb{R}^N$	Controllability
State-to-Output	$\eta : \mathbb{R}^N \to \mathrm{L}_2^O(\mathbb{R})$	$\mathscr{O} : \mathbb{R}^N \to \mathrm{L}_2^O(\mathbb{R}^+)$	Observability

Table 4.1.: *Comparison of state mappings and operators.*

As the convolution operator is bounded for stable systems (Section 3.1), the previous mappings admit a POD. Opposed to the gramian-based method a finite spectrum is not required. In the following, the optimization problems yielding the reducing state-space projection for the input-to-state, state-to-output and input-to-output mappings, are described.

4.2.1. Input-to-State Mapping

Finding a reduced order mapping from an input u to a state x amounts to determining a low-dimensional surrogate for the dynamical system component of the linear control system,

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \\ x(0) &= 0. \end{aligned} \tag{4.9}$$

The operator mapping an input function to a state-space trajectory is a variant of the convolution operator (3.1).

Definition 4.3 (Input-to-State Map)
An ***input-to-state map*** ξ *maps a function* $u : [0, \infty) \to \mathrm{L}_2$ *to a state-space trajectory* $x : [0, \infty) \to \mathrm{L}_2$ *based on an asymptotically stable linear dynamical system* (4.9) *and is given by:*

$$\xi(u)(t) := \int_0^t \mathrm{e}^{A(\tau)} Bu(\tau)\,\mathrm{d}\tau.$$

The input-to-state map is related to the controllability operator $\mathscr{C}$ (3.3) through composition with the time-flip operator $\mathscr{C} = \xi \circ F$. A low-order approximation to the system,

$$\begin{aligned} \dot{x}_r(t) &= A_r x_r(t) + B_r u(t), \\ x_r(0) &= 0, \end{aligned}$$

is determined by a state reducing projection $U_C \in \mathbb{R}^{N\times n}$ as in (2.12), which can be obtained as the maximum input-to-state energy conserving Galerkin projection and is described next as the solution to a least-squares optimization problem:

$$U_C = \operatorname*{arg\,min}_{U\in\mathbb{R}^{N\times n}, U^\intercal U = \mathbb{1}} \|x - Ux_r\|_{\mathrm{L}_2}^2,$$

subject to:

$$\begin{aligned} x(t) &= \int_0^t \mathrm{e}^{A\tau} Bu(\tau)\,\mathrm{d}\tau, \\ x_r(t) &= \int_0^t \mathrm{e}^{U^\intercal AU\tau} U^\intercal Bu(\tau)\,\mathrm{d}\tau. \end{aligned}$$

Due to the relation of the SVD (POD) and the linear least squares method through the Schmidt-Eckhardt-Young-Mirsky theorem [6, Ch. 3], the solution is obtained as the first n left eigenvectors of the controllability gramian,

$$\xi \overset{SVD}{=} UDV \to U = \begin{pmatrix} U_1 & U_2 \end{pmatrix} \to U_C := U_1,$$

with a reduction error (for discrete trajectories):

$$\min_{U\in\mathbb{R}^{N\times n}, U^\intercal U = \mathbb{1}} \|x - Ux_r\|_{\mathrm{L}_2}^2 = \sigma_{n+1}(\xi).$$

The computation of the SVD of the discrete image of the input-to-state operator together with a selection of left singular vectors makes this approach equivalent to the POD method of snapshots from [205].

4.2.2. State-to-Output Mapping

To find a reduced order mapping from an initial state x_0 to an output y amounts to determining a low-dimensional surrogate for the input-free linear control system,

$$\begin{aligned} \dot{x}(t) &= Ax(t), \\ y(t) &= Cx(t), \\ x(0) &= x_0. \end{aligned} \tag{4.10}$$

The operator mapping an initial state to an output trajectory is also a variant of the convolution operator (3.1).

Definition 4.4 (State-to-Output Map)
*A **state-to-output map** η maps an initial state x_0 to an output trajectory $y : [0, \infty) \to \mathrm{L}_2$ based on an input-free asymptotically stable linear control system* (4.10) *and is given by:*

$$\eta(x_0)(t) := C\,\mathrm{e}^{At}\,x_0.$$

The state-to-output operator is equal to the observability operator $\mathscr{O} = \eta$ from (3.4).
A low-order approximation to the input-free system,

$$\begin{aligned} \dot{x}_r(t) &= A_r x_r(t), \\ y_r(t) &= C_r x_r(t), \\ x_r(0) &= x_{r,0}, \end{aligned}$$

is determined by a state reducing projection $U_O \in \mathbb{R}^{N\times n}$ as in (2.12), which can be obtained as the maximum energy conserving state-to-output Galerkin projection and is described next as the solution to a least-squares optimization problem:

$$U_O = \operatorname*{arg\,min}_{U\in\mathbb{R}^{N\times n}, U^\intercal U = \mathbb{1}} \|y - y_r\|^2_{\mathrm{L}_2},$$

subject to:

$$\begin{aligned} y(t) &= C\,\mathrm{e}^{At}\,x_0, \\ y_r(t) &= CU\,\mathrm{e}^{U^\intercal AUt}\,U^\intercal x_0. \end{aligned}$$

The cost function of the minimization problem can be reformulated as a weighted least-squares problem of the input-to-state map:

$$\|y - y_r\|^2_{\mathrm{L}_2} = \|Cx - C_r x_r\|^2_{\mathrm{L}_2} = \|Cx - CUx_r\|^2_{\mathrm{L}_2} = \|x - Ux_r\|^2_{C^\intercal C}. \tag{4.11}$$

A method related to the state-to-output map is the Hessian-based approach from [12, 13], which in turn is a variant of the observability gramian as shown in [154]. Alternatively, by the use of the adjoint system (see Definition 2.4), the projection can be determined as the input-to-state map from Definition 4.3.

4.2.3. Input-to-Output Mapping

Lastly, a reduced order mapping from the input u to output y leads to an approximate low-dimensional representation of the linear control system,

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t), \\ x(0) &= x_0. \end{aligned} \tag{4.12}$$

The operator mapping an input function to an output trajectory is given by the convolution operator (3.1). Using the input-to-output mapping method resembles the goal-oriented approach from [229, 30]. For sake of completeness a definition of this mapping is also given.

Definition 4.5 (Input-to-Output Mapping)
*A **input-to-output map** S maps an input function $u : [0, \infty) \to \mathrm{L}_2$ to an output trajectory $y : [0, \infty) \to \mathrm{L}_2$ based on an asymptotically stable linear control system* (4.12) *and is given by:*

$$S(u)(t) := C\,\mathrm{e}^{At}\,x_0 + \int_0^t C\,\mathrm{e}^{A\tau}\,Bu(\tau)\,\mathrm{d}\tau.$$

A low-order approximation is then given by:

$$\begin{aligned} \dot{x}_r(t) &= A_r x_r(t) + B_r u(t), \\ y_r(t) &= C_r x_r(t), \\ x_r(0) &= x_{r,0}, \end{aligned}$$

obtained by a projection $U_H \in \mathbb{R}^{N\times n}$ as in (2.12).
The maximum energy conserving input-to-output Galerkin projection $U_H \in \mathbb{R}^{N\times n}$ can be described as the solution to a least-squares optimization problem:

$$U_H = \operatorname*{arg\,min}_{U\in\mathbb{R}^{N\times n}, U^\intercal U = \mathbb{1}} \|y - y_r\|_{\mathrm{L}_2}^2,$$

subject to:

$$\begin{aligned} y(t) &= C\,\mathrm{e}^{At}\,x_0 + C\int_0^t \mathrm{e}^{A\tau}\,Bu(\tau)\,\mathrm{d}\tau, \\ y_r(t) &= CU\Big(\mathrm{e}^{U^\intercal AU t}\,x_0 + \int_0^t \mathrm{e}^{U^\intercal AU\tau}\,U^\intercal Bu(\tau)\,\mathrm{d}\tau\Big), \end{aligned}$$

which can also be viewed as a weighted least-squares problem (4.11).
As opposed to the gramian-based approach, which utilizes a finite-rank transformed variant of the convolution operator (3.2), the state reduction using the input-to-output mapping aims to extract the dominant base components (modes) from the infinite dimensional convolution operator.

4.3. Combined Reduction

The optimization-based combined state and parameter reduction joins the parameter reduction from Section 4.1 with the state reduction from Section 4.2. This combined reduction method is of iterative nature, which successively enlarges the state- and parameter-space bases and hence improves the state and parameter projections.
In each iteration, the next parameter θ_{I+1}, representing the currently worst approximated parameter subspace, is obtained from the Tikhonov-regularized greedy optimization (with $\Gamma = \mathbb{1}$) and orthogonalized into the parameter projection Π. Then, a trajectory for this parameter $x(\theta_{I+1})$ is computed from which the dominant mode $\bar{x}_{I+1}$ is extracted based on either input-to-state (Section 4.2.1), state-to-output (Section 4.2.2) or input-to-output (Section 4.2.3) mappings. Including this mode into the state-space projection U by orthogonalization completes an iteration:

$$\theta_{I+1} = \arg\max_{\theta \in \mathbb{R}^P} \| y(\theta) - y_r(\Pi_I \Pi_I^\intercal \theta) \|_{\mathrm{L}_2}^2 - \beta_2 \|\theta\|_2^2,$$

subject to:

$$\begin{aligned}
&\dot{x}(t) = f(x(t), u(t), \theta),\\
&y(t) = g(x(t), u(t), \theta),\\
&\dot{x}_r(t) = U^\intercal f(U x_r(t), u(t), \theta),\\
&y_r(t) = g(U x_r(t), u(t), \theta),\\
&x(0) = x_0\\
&x_r(0) = U^\intercal x_0\\
\Pi_{I+1} &= \{\Pi_I \cup (\Pi_I^\perp \cap \theta_{I+1})\},\\
U_{I+1} &= \{U_I \cup (U_I^\perp \cap \mathrm{pod}_1(x(\theta_{I+1})))\},
\end{aligned} \tag{4.13}$$

with linear[48] functionals f and g. Thus, the selected parameter θ_{I+1} determines the location for the trajectory of states or outputs from which the next state-space base selection is made. In the subsequent iteration, the next parameter is determined with regard to the state and parameter reduced model from the previous iteration. Overall, this approach to combined reduction is an L_∞ method in terms of the parameter-space reduction due to the greedy sampling minimizing the maximum error, while the state-space is reduced by an L_2 method. In this method, two optimization problems need to be solved in each iteration. First, the greedy sampling over the parameter-space, which can be implemented by a regularized unconstrained optimization (minimization), see Section 4.1.3. Second, the dominant modes in the state-space need to be found, for which a least-squares method can be used. From a numerically point of view, the simplest method to compute the state projection is the input-to-state mapping, since a TSVD of discrete trajectory yields the leading POD mode as principal left singular vector (Section 4.2.1),

$$\bar{x}_{I+1} = \mathrm{pod}_1(x(\theta_{I+1})).$$

[48] Linearity with respect to the state.

4.3.1. Basic Algorithm

Following, the basic optimization-based combined state and parameter reduction approach is summarized from an algorithmic point of view in Algorithm 4. Proposed in [113], this is essentially a time-dependent variant of the algorithm from [31, 151].
First, an initial parameter is set to some nominal (prior) parameter $\theta_0 = \bar{\theta}$, which is also the first base vector $\Pi_0 = \theta_0$ of the parameter base Π. The state base U is initialized by a selection from a snapshot $U_0 = \overline{x(\theta_0)}$ associated to initial parameter choice θ_0.
Second, for a preset target reduced parameter-space dimension the greedy sampling is employed to determine the next parameter base vector θ_I, which is then incorporated into the parameter base Π_{I-1} by orthogonalization. The state base U_{I-1} is augmented by a selection from a snapshot $x(\theta_I)$, also by orthogonalization. Since the orthogonalization (orth) is crucial to the ROM construction it is discussed separately in Section 4.3.3.

Algorithm 4: Basic Optimization-Based Combined Reduction

$\theta_0 \longleftarrow \bar{\theta}$
$\Pi_0 \longleftarrow \theta_0$
$U_0 \longleftarrow \mathrm{pod}_1(x(\theta_0))$
for $I = 1 : R$ **do**
  $\theta_I \longleftarrow \arg\min -J(\theta; U_{I-1}, \Pi_{I-1}) + \mathscr{R}_2(\theta; \Pi_{I-1})$
  $\Pi_I \longleftarrow \mathrm{orth}(\Pi_{I-1}, \theta_I)$
  $\bar{x}_I \longleftarrow \mathrm{pod}_1(x(\theta_I))$
  $U_I \longleftarrow \mathrm{orth}(U_{I-1}, \bar{x}_I)$

Alternatively, to a preselected reduced parameter-space dimension R, the iteration can be continued until a certain output error threshold ϵ in the output residual is met, for example in the L_2-norm output error:

$$\| y(\theta_I) - y_r(\Pi_I \Pi_I^{\mathsf{T}} \theta_I) \|_{\mathrm{L}_2} \leq \epsilon.$$

Hence, Algorithm 4 can be rewritten using an indetermined `while`-loop instead of a `for`-loop with predetermined iteration count as shown in Algorithm 5.

Algorithm 5: Basic Optimization-Based Combined Reduction II

$\theta_0 \longleftarrow \bar{\theta}$
$\Pi_0 \longleftarrow \theta_0$
$U_0 \longleftarrow \mathrm{pod}_1(x(\theta_0))$
while $\| y(\theta_{I-1}) - y_r(\Pi_{I-1} \Pi_{I-1}^{\mathsf{T}} \theta_{I-1}) \|_{\mathrm{L}_2} > \epsilon$ **do**
  $\theta_I \longleftarrow \arg\min -J(\theta; U_{I-1}, \Pi_{I-1}) + \mathscr{R}_2(\theta; \Pi_{I-1})$
  $\Pi_I \longleftarrow \mathrm{orth}(\Pi_{I-1}, \theta_I)$
  $\bar{x}_I \longleftarrow \mathrm{pod}_1(x(\theta_I))$
  $U_I \longleftarrow \mathrm{orth}(U_{I-1}, \bar{x}_I)$

4.3.2. Regularization Coefficients

A practical problem of the previous Algorithms 4 and 5 is the selection of the weighting coefficient β_2 in the regularization operator. In the scope of this work these weights are selected manually for the specific problem by heuristic testing of predetermined regularization weights, for example:

$$\beta_2 \in \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}.$$

An automation of this approach is the L-curve method [103].

4.3.3. Base Expansion

Both, the state and parameter reduced order bases, require an orthogonalization of the respective new base components in each iteration of the algorithm. In Algorithms 4 and 5 the orthogonalization is symbolized by orth.
Various algorithms are available for this task [88, Ch. 5], for example: Householder reflections, Givens rotations or the Gram-Schmidt process. Since an existing orthogonal basis Q is extended with new base vectors v, the Gram-Schmidt process,

$$\begin{aligned} v^{\perp} &= v - Q_I(Q_I^{\mathsf{T}} v), \\ Q_{I+1} &= \begin{pmatrix} Q_I & v^{\perp} \end{pmatrix}, \end{aligned}$$

is a suitable choice. In exact arithmetic, the Gram-Schmidt process would yield an orthogonal base. In finite precision arithmetic the resulting base is most likely not (completely) orthogonal to precision, due to rounding errors, and numerically unstable.
To counteract this roundoff error, an extension to the Gram-Schmidt process is proposed in [28]; in this extension the orthogonalization is repeated until the 2-norm of the orthogonalized new base vector surpasses a threshold ϵ. Algorithm 6 lists this re-iterated Gram-Schmidt process from [28], which is used for the optimization-based combined reduction.

Algorithm 6: Re-Iterated Gram-Schmidt

$Q^{\mathsf{T}}Q = \mathbb{1}$
$b \leftarrow 0$
while $b < \epsilon$ **do**
  $v \leftarrow v - Q(Q^{\mathsf{T}}v)$
  $b \leftarrow \|v\|_2$
  $v \leftarrow b^{-1}v$
$Q \leftarrow \begin{pmatrix} Q & v \end{pmatrix}$

Since the orthogonality of the state and parameter projections is essential for the quality of the ROM, the plain Gram-Schmidt algorithm is not sufficient and requires a more accurate orthogonalization method such as the re-iterated Gram-Schmidt process.

4.3.4. System Stability

If the parametrization of the underlying control system for the model constrained optimization problem involves the dynamical system as in (4.2), then, in addition to the optimality condition [152, Ch. 3.3.2], the stability of the system has to be considered.
If, in case of a linear system, the system matrix is parametrized by a map k from the parameter-space Θ to an $N \times N$ matrix,

$$\begin{aligned} k : \Theta &\to \mathbb{R}^{N\times N}, \\ \theta &\mapsto A(\theta), \end{aligned}$$

then a constraint on the eigenvalues of $A(\theta)$ may be necessary to ensure stability of the solution. This may be achieved by a constrained optimization which explicitly excludes subspaces of the parameter-space that would produce unstable systems.
Practically, it is often sufficient to initialize the optimization algorithm inside a stable region of the parameter-space.

4.3.5. Inference for Prediction

The inference for prediction (IFP) approach, introduced in [150], is a hybrid method of the gramian-based balanced truncation and the optimization-based reduction. It is based on two attributes of a pair consisting of a model and data. While balanced truncation evaluates controllability (see Section 3.2.1) and observability (see Section 3.2.2) of the model, the IFP method assesses the model's observability, the **prediction observability**, and the data observability, the **experiment observability**. The IFP algorithm aims to enable predictions for parameters based on experimental data without solving a (possibly high-dimensional) inverse problem.
The experiment observability, or short experimentability, is defined as the energy transferred from parameters θ to measured outputs y_d:

$$L(\theta) := \|y_d\|_2 = \|\mathcal{O}_e\theta\|_2 = \theta^\intercal \mathcal{O}_e^\intercal \mathcal{O}_e \theta = \theta^\intercal W_{O,e}\theta,$$

by the experiment observability gramian $W_{O,e}$.
Similarly, the prediction observability, or predictability, is introduced as the transfer of energy from parameters to simulated outputs y:

$$L(\theta) := \|y\|_2 = \|\mathcal{O}_p\theta\|_2 = \theta^\intercal \mathcal{O}_p^\intercal \mathcal{O}_p \theta = \theta^\intercal W_{O,p}\theta,$$

by the prediction observability gramian $W_{O,p}$.
The experiment observability operator $\mathcal{O}_e$ and prediction observability operator $\mathcal{O}_p$ can be understood as solution operators for a given parameter to an underlying control-system-type model. These two observability gramians can be balanced and truncated, like the controllability and observability gramian for balanced truncation in Section 3.2.4 and yield a reduced basis for the parameter-space.

4.4. Data-Driven Regularization

In this section an extension to the previous algorithm is presented, which is designed for inverse problems. While the basic algorithm from Section 4.3.1 aims to reduce the model over the whole admissible parameter-space; with this extension the resulting ROM approximates the original model over a subspace of the parameter-space which is defined by an additional soft constraint on the greedy sampling.
The original optimization-based combined (state and parameter) reduction introduced in [151] is aimed at (Bayesian) inverse problems. For inverse problems, naturally, some observed output data y_d exists, to be matched to the model's output during the inversion; for example in the least-squares sense:

$$\theta_d = \underset{\theta \in \mathbb{R}^P}{\arg\min} \, \| y_d - y(\theta) \|_{\mathrm{L}_2}^2 .$$

For model reduction prior to an inversion, the available output data can be utilized during the reduction process to refine the ROM for a subsequent inversion.
The model reduction methods in this work approximate the FOM's output y by the ROM's output y_r, hence a optimization can be performed using the ROM:

$$\theta_d \approx \underset{\theta \in \mathbb{R}^P}{\arg\min} \, \| y_d - y_r(\theta) \|_{\mathrm{L}_2}^2 .$$

This motivates a regularization term based on the data-misfit related to [114]: the distance of the FOM's output to the measured output data,

$$\mathscr{R}_{\beta_d}(\theta) := \beta_d \| y_d - y(\theta) \|_{\mathrm{L}_2}^2 , \tag{4.14}$$

and has the purpose of confining the ROMs to subspaces which are relevant to the measured data. The objective function of the greedy algorithm (4.6) augmented by the data-driven regularization (4.14) then reads:

$$\begin{aligned} \theta_I &= \underset{\theta \in \mathbb{R}^P}{\arg\max} \, J_d(\theta; U_I, \Pi_I), \\ J_d(\theta; U_I, \Pi_I) &= J(\theta; U_I, \Pi_I) - \mathscr{R}_{\beta_2}(\theta) - \mathscr{R}_{\beta_d}(\theta) \\ &= \| y(\theta) - y_r(\Pi_I \Pi_I^{\mathsf{T}} \theta) \|_{\mathrm{L}_2}^2 - \beta_2 \| \theta \|_2^2 - \beta_d \| y_d - y(\theta) \|_{\mathrm{L}_2}^2 , \end{aligned}$$

with an associated weight coefficient $\beta_d \in [0, 1]$. By this data-driven regularization, the base vectors for the parameter projection are selected in a manner to enhance the parameter optimization during the inversion of the data by the ODE-constrained optimization using the ROM. The additional computational complexity introduced by the data-driven regularization term is negligible, as the output trajectory $y(\theta)$ can be reused from the residual J.
Including the data-misfit minimization as a regularization into the reduction process has two consequences. First, part of the inversion is already performed during the model reduction which can speed up the actual inversion. Second, the resulting ROM is specific to the measured data y_d and most likely not re-usable for another dataset. Even though the ROM is only valid for a certain dataset, this extension is applicable in settings which perform many inversions on a single dataset, i.e. hypothesis testing.

4.4.1. Enhanced Algorithm

Next, the previously introduced data-driven regularization from Section 4.4 is incorporated into the basic algorithm as well as some minor modifications. The following Algorithm 7 lists the enhanced variant of the previous optimization-based combined reduction method.

Algorithm 7: Enhanced Optimization-Based Combined Reduction

$\theta_0 \longleftarrow \bar{\theta}$
$\Pi_0 \longleftarrow \theta_0$
$U_0 \longleftarrow \text{pod}_1(x(\theta_0))$
for $I = 1 : R$ **do**
  $\theta_I \longleftarrow \arg\min -J(\theta; U_{I-1}, \Pi_{I-1}) + \beta_2 \mathscr{R}_2(\theta; \Pi_{I-1}) + \beta_d \mathscr{R}_d(\theta)$
  $\Pi_I \longleftarrow \text{orth}(\Pi_{I-1}, \theta_I)$
  $x^{\perp}(\theta_I) \longleftarrow x(\theta_I) - U_I U_I^{\mathsf{T}} x(\theta_I)$
  $\bar{x}_I \longleftarrow \text{pod}_1(x^{\perp}(\theta_I))$
  $U_I \longleftarrow \text{orth}(U_{I-1}, \bar{x}_I)$

Additional to the inclusion of the data-driven regularization term into the cost functional of the optimization performing the greedy sampling, instead of expanding the basis with the principal POD mode from a state-space simulation for the state-space projection, the state basis can be augmented with the principal POD mode of the state-space projection error, as proposed for the POD-Greedy method in [55].
This modular variant of the optimization-based combined reduction algorithm only abstractly describes a central component of the reduction process, which is the maximization method. The energy maximization to determine greedily the best parameter subspace is effectively computed by a reformulation as a minimization problem with negative cost functional (4.7), which, as described in Section 4.1.3, can be solved by an unconstrained optimization method due to the (Tikhonov) regularization, or in case of further conditions on the base vectors, like system stability by constraint optimization.
Newton (or Quasi-Newton) methods are established methods for optimization procedures. The major computational challenge is the computation of the first and second order derivative information. An analytical computation of the derivatives is often impractical, hence an approximation, for example by a finite difference scheme for the Jacobian with a finite difference h, is chosen:

$$\begin{aligned} \theta_h^i &:= \theta + \delta_i^p h, \\ \frac{\partial y}{\partial \theta_i} &= \frac{y(\theta) - y(\theta_h^i)}{h}, \\ \Rightarrow J &\approx \begin{pmatrix} \frac{\partial y}{\partial \theta_1} & \dots & \frac{\partial y}{\partial \theta_p} \end{pmatrix}. \end{aligned}$$

Alternatively, automatic differentiation [93] is a tool to accelerate the computation of the Jacobian or (approximate) Hessian if analytical derivatives are not available.

4.5. Nonlinear Systems

The previously presented optimization-based approach to combined state and parameter reduction can be extended directly to nonlinear systems, yet without a theoretical justification, but with a similar argument as for the empirical gramians: Given a method to obtain discrete (output) trajectories for nonlinear IVPs, the linear system gramians can be generalized to empirical gramians for nonlinear systems (near a steady-state); hence, given methods for the nonlinear optimization-based parameter-space and state-space reduction, the optimization-based combined reduction can also be generalized to nonlinear systems near a steady-state. Broadening the scope of the optimization-based approach to nonlinear systems relies heavily on state-space simulations, hence the (asymptotic) stability of the considered systems is paramount to the computation of a ROM.

4.5.1. Parameter Reduction

Since the abstract Algorithm 7 is not relying on the linear structure of the model, it principally applies also to nonlinear models, but the practical computation of the ROM requires a maximization (minimization) method for nonlinear optimization problems:

$$\theta_{I+1} = \arg\max_{\theta\in\Theta} \|y(\theta) - y(\Pi_I \theta_r)\|_{\mathrm{L}_2}^2 - \beta_2 \|\theta\|_2^2,$$

subject to:

$$\begin{aligned}
&\dot{x}(t) = f(x(t), u(t), \theta),\\
&y(t) = g(x(t), u(t), \theta),\\
&x(0) = x_0,\\
&\theta_r = \Pi_I^{\mathsf{T}} \theta,\\
&\Pi_I^{\mathsf{T}} \Pi_I = \mathbb{1}.
\end{aligned}$$

Except for the (control system) constraint, this optimization problem is formulated similar to the linear greedy sampling formulation (4.5), which now comprises a general (and thus possibly nonlinear) control system and usually a nonlinear mapping from the parameter-space to the control system model. The greedy sampling used to assemble the parameter reducing projection, formulated as a minimization problem (4.7), then reads

$$\theta_{I+1} = \arg\min_{\theta\in\Theta} -\|y(\theta) - y(\Pi_I \theta_r)\|_{\mathrm{L}_2}^2 + \beta_2 \|\theta\|_2^2,$$

and as such can be solved with a (regularized) nonlinear least-squares method, like variants of the Newton's method [179]. For nonlinear systems or systems with nonlinear parametrization, the regularization might require a weighting matrix Γ, as in (4.8), to ensure the minimization excludes parameters which destabilize the system or produce undefined results.

4.5.2. State Reduction

Like the optimization-based parameter-space reduction, also the optimization-based state-space reduction principally extends directly to nonlinear systems, but some practical matters of the computation have to be considered.
As noted in Section 4.2, computing the state reducing projection from the input-to-state map is equivalent to obtaining a projection based on the dominant POD modes. These POD modes can be determined from discrete trajectories $x_h(t_h)$ using the SVD, and hence (only) requires the discrete numerical solutions to the IVP for the underlying nonlinear control system.
In the nonlinear setting, an adjoint system is not directly available as in the linear setting, thus an additional nonlinear model-constrained least-squares optimization problem needs to be solved (iteratively) to obtain the state reducing projection from the state-to-output mapping. Similarly for the input-to-output mapping an optimization problem has to be solved. Thus, for nonlinear systems the input-to-state mapping is the most convenient, as the state-space reducing projection can be obtained by a TSVD and is a tested method for nonlinear state-space reduction [141].

4.5.3. Combined Reduction

With nonlinear optimization-based parameter-space and state-space reduction approaches a nonlinear combined reduction method can be formulated analogously to the linear setting. Since no theoretical results are provided in this context, meaningful results can only be expected near a steady-state. The core optimization problem for the greedy sampling then has the following form in the nonlinear regime:

$$\theta_{I+1} = \arg\max_{\theta\in\Theta} \| y(\theta) - y_r(\Pi_I \Pi_I^\intercal \theta_r) \|_{\mathrm{L}_2}^2 - \beta_2 \|\theta\|_2^2,$$

subject to:

$$\begin{aligned}
&\dot{x}(t) = f(x(t), u(t), \theta),\\
&y(t) = g(x(t), u(t), \theta),\\
&x(0) = x_0,\\
&\dot{x}_r(t) = U_I^\intercal f(U_I x_r(t), u(t), \theta),\\
&y_r(t) = g(U_I x_r(t), u(t), \theta),\\
&x_r(0) = U_I^\intercal x_0,\\
&U_I^\intercal U_I = \mathbb{1}_I,\\
&\Pi_I^\intercal \Pi_I = \mathbb{1}_I\,.
\end{aligned}$$

The other components of Algorithm 4 (and Algorithm 7 for that matter) remain the same. Since the data-driven regularization from Section 4.4 is agnostic to the model, it similarly directly extends to nonlinear systems.

5. Software Implementation

Contents

This chapter describes the software developed to implement the methods from the previous chapters. After an outline of the software design decisions, a generic benchmark for testing purposes is described. Then, the implementation of the empirical-gramian-based method from Chapter 3 and the optimization-based approach from Chapter 4 are documented and tested.

5.1. Design Principles

Both implementations are developed using the Matlab programming language and are compatible with OCTAVE [42, 56, 57] and MATLAB® [122], with the former guaranteeing a full open-source software stack while the latter allows a wide circulation. The Matlab language was chosen due to four properties: the availability of an open-source interpreter, its mathematical expressiveness, widespread use and long-term compatibility. The source code is engineered with regard to the Matlab language best practices [130] and performance guidelines [5]. On the matter of performance it should be noted that, due to extensive use of vectorization, most computational intensive operations are transferred to a basic linear algebra subprogram (BLAS) backend[49].
A general principle during the development of the source code representing the algorithms is the separation of method and experiment, to encourage apart from the replicability of the results from this work also reproducibility for similar, and reusability of these implementations in other settings. Beyond the targeted field of application in combined (state and parameter) reduction, or pMOR and nMOR in general, this enables the use in further control and systems applications. To this end both implementations provide interfaces with a minimal set of mandatory arguments, an additional set of flexible optional arguments and a configurable set of options allowing a more finely tuned selection of algorithms. Since either method relies on computed trajectories, a default solver, described in Section 5.3.3, is provided as well as an interface to connect custom solvers.

[49] See Appendix B.2 for information on the used BLAS implementation.

5.2. Inverse Lyapunov Procedure

Following, a process is described that utilizes the empirical gramians from Chapter 3 to generate random systems, which are used in the course of this chapter to test the software implementations. This method to generate random systems is the **inverse Lyapunov procedure** (ILP), which is introduced in [206] and also listed in [41, ILP].
The ILP can be viewed as stepping through the balanced truncation algorithm backwards. First, random eigenvalues for the controllability gramian and the observability gramian are generated from an exponential of a uniform distribution on the interval $[0, \frac{1}{2}]$. Random eigenvectors to W_C and W_O are generated from an SVD of a random normally distributed matrix. From these random system gramians a balancing transformation is computed. Secondly, a random normally distributed input matrix B and output matrix C are created. The output matrix C is then scaled such that $(BB^\intercal)_{ii} = (C^\intercal C)_{ii}$ to satisfy the requirements in [206, Thm. 1]. Thirdly, the controllability (or observability) gramian A of the virtual system $\Sigma(W_C, B, C)$ is computed. Lastly, the system $\Sigma(A, B, C)$ is stabilized by subtracting a scaled unit matrix from A and unbalanced through the use of the inverse balancing transformation.

Algorithm 8: Inverse Lyapunov Procedure

$\lambda(W_C) \longleftarrow \exp(\mathcal{U}^N_{[0,\frac{1}{2}]})$
$\lambda(W_O) \longleftarrow \exp(\mathcal{U}^N_{[0,\frac{1}{2}]})$
$\{T_C, D, T_O\} \longleftarrow \mathrm{SVD}(\mathcal{N}^{N\times N}_{0,1})$
$W_C \longleftarrow T_C \,\mathrm{diag}(\lambda(W_C)) T_C^\intercal$
$W_O \longleftarrow T_O \,\mathrm{diag}(\lambda(W_O)) T_O^\intercal$
$\{U, D, V\} \longleftarrow \mathrm{SVD}(W_C W_O)$
$B \longleftarrow \mathcal{N}^{N\times J}_{0,1}$
$C \longleftarrow \mathcal{N}^{O\times N}_{0,1}$
$C_{ij} \longleftarrow C_{ij} \sqrt{(\sum_i B_{ij} B_{ji})/(\sum_j C_{ji} C_{ij})}$
$A \longleftarrow V(\int_0^\infty \mathrm{e}^{W_C t} BB^\intercal \mathrm{e}^{W_C^\intercal t} \,\mathrm{d}t - \frac{1}{N} \mathbb{1}_N)U$
$B \longleftarrow VB$
$C \longleftarrow CU$

Following, the implementations of the gramian-based and optimization-based methods are tested by a linear state-space symmetric MIMO system, generated by the ILP with an additional linear parameter dependency (3.26) including a source matrix $F \in \mathbb{R}^{N\times P}$:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + F\theta, \\ y(t) &= Cx(t), \end{aligned} \tag{5.1}$$

together with zero initial state $x_i(0) = 0$ and an impulse input $u_i(t) = \delta(t)$. The parameters act as a source term and are drawn from a uniform distribution $\theta_i \leftarrow \mathcal{U}_{[0,1]}$. The state dimension is selected as $N = P = 100$, the input and output dimension as $M = O = 10$.

5.3. emgr - Empirical Gramian Framework

The empirical gramian framework[50] (`emgr`) [107] encapsulates functionality to compute the (discrete) empirical gramians presented in Chapter 3, requires only basic linear algebra operations[51] and has no dependencies on other toolboxes. The implementation is contained in a single file and spans less than 500 lines of code (LoC). Released under an open-source license (Appendix B.3) and available[52] from `http://gramian.de`, `emgr` is engineered to provide a uniform interface for various types of empirical gramians [109].
A simplification of the general definitions of the perturbation sets in Definition 3.22 is made in terms of the sets of rotation matrices, which are set to $R_u = \{-\mathbb{1}_J, \mathbb{1}_J\}$, $R_x = \{-\mathbb{1}_N, \mathbb{1}_N\}$. This choice, suggested in [142], entails many numerical simplifications during the assembly of the empirical gramians, but restricts perturbations to a single component of the inputs or initial state for each trajectory.
Following, the interface of the `emgr` toolbox[53], the core features and the available configurable options are listed and described. Then, the functionality is verified and validated by a benchmark problem with the property from Note 3.12.

5.3.1. Interface

The signature of the `emgr` function encompasses five mandatory arguments (single letter) and seven optional arguments (double letter), and reads:

```
W = emgr(f,g,s,t,w,pr,nf,ut,us,xs,um,xm);
```

Hence, the minimal usage:

```
W = emgr(f,g,s,t,w);
```

requires the following five arguments:

- `f` *function handle* to a function with signature `xdot ← f(x,u,p)`, the system's vector field depending on the current state `x`, the current input `u` and the (current) parameter `p`. For example, in case of a linear system: `f = @(x,u,p) A*x + B*u;`
- `g` *function handle* to a function with signature `y ← g(x,u,p)`, the system's output functional depending on the current state `x`, the current input `u` and the (current) parameter `p`. For example in case of a linear system: `g = @(x,u,p) C*x;`
- `s` *three component vector* `s = [J,N,O]` holding number of inputs J, states N and outputs O;
- `t` *two component vector* `t = [h,T]` holding time step h and stop time $\mathscr{T}$;
- `w` *character* selecting the gramian type, for details see Section 5.3.2.

[50] See the listing in Appendix B.3.
[51] Such as vector and matrix operations contained in the BLAS library.
[52] The empirical gramian framework is also listed in the **Oberwolfach References on Mathematical Software**: no. 345 `http://orms.mfo.de/project?id=345`
[53] For `emgr` in version 3.9.

Furthermore, seven optional arguments are provided:

`pr` parameters provided to the system's vector field and output functional during the numerical simulations; the parameter gramians (W_S, W_I, $W_{\tilde{I}}$) require a two-column matrix with minimum and maximum parameter values;

vector holding the system's parameters of dimension $P \times 1$,

matrix holding multiple parameter column vectors, for pMOR, described in Section 3.3.4;

`nf` *twelve component vector* encoding the option flag settings, for details see Section 5.3.3;

`ut` input time series, default value: 1;

scalar setting uniform peak of impulse input to all inputs, normalized by the time-step `h` to ensure the discrete input fulfills the identity $\int_0^{\mathscr{T}} u_i(t)\,\mathrm{d}t = \texttt{ut(1)}$,

vector setting per-input component peaks to impulse input, normalized by the time-step `h` to ensure the discrete input fulfills the identity $\int_0^{\mathscr{T}} u_i(t)\,\mathrm{d}t = \texttt{ut(i)}$ and is of dimension $J \times 1$,

matrix holding the discrete input time series of dimension $J \times \frac{\mathscr{T}}{h}$,

handle to a function `u ← uf(t)` which is internally discretized, for example a Gaussian pulse at time `tp` and full width at half maximum `h`:

```
u = @(t) exp(-(0.5/h)*(t-tp).^2)./sqrt(2.0*pi*h);
```

∞ generates a linear chirp signal using the havercosine [226]:

```
u = @(t) 0.5*cos(pi*(t+10*t.*t))+0.5;
```

`us` steady-state input, default value: 0;

scalar setting steady-state input for all input components to provided value;

vector holding input during nominal or steady-state (`xs`) of dimension $J \times 1$,

`xs` steady-state and nominal initial state, default value: 0;

scalar setting all steady-state components to provided value;

vector holding steady-state of dimension $N \times 1$,

`um` input scales corresponding to Q_u, default value: 1;
sub-division of scale sequence configurable by option flags (see Section 5.3.3),

scalar of uniform maximum scales for all input components,

vector holding per-input scale maxima of dimension $J \times 1$,

matrix holding multiple columns of per-input scales (used as is);

`xm` state scales corresponding to Q_x, default value: 1;
sub-division of scale sequence configurable by option flags (see Section 5.3.3),

scalar uniform maximum scale for all state components,

vector holding per-state scale maxima of dimension $N \times 1$,

matrix holding multiple columns of per-state scales (used as is).

5.3.2. Features

The empirical gramian framework features seven types of empirical gramians and empirical covariance matrices[54]. For the (parametric) state-space system gramians a single matrix is returned. For the parameter-space gramians a cell-array[55] of two matrices is returned, the first element corresponds to the associated state-space gramian, the second element embodies the actual parameter-space gramian. Following is a list of the computable empirical gramians and their return values:

`'c'` Empirical Controllability Gramian W_C, see: Section 3.3.1;
`emgr` returns:

$N \times N$ empirical controllability gramian matrix

`'o'` Empirical Observability Gramian W_O, see: Section 3.3.2;
`emgr` returns:

$N \times N$ empirical observability gramian matrix

`'x'` Empirical Cross Gramian W_X, see: Section 3.3.3;
`emgr` returns:

$N \times N$ empirical cross gramian matrix

`'y'` Empirical Linear Cross Gramian W_Y, see: Section 3.3.3;
`emgr` returns:

$N \times N$ empirical linear cross gramian matrix

`'s'` Empirical Sensitivity Gramian W_S, see: Section 3.4.1;
`emgr` returns:

$N \times N$ approximate empirical controllability gramian matrix

$P \times P$ empirical sensitivity gramian (diagonal) matrix

`'i'` Empirical Identifiability Gramian W_I, see: Section 3.4.2;
`emgr` returns:

$N \times N$ empirical observability gramian matrix

$P \times P$ empirical identifiability gramian matrix

`'j'` Empirical Joint Gramian W_J, see: Section 3.4.3;
`emgr` returns:

$N \times N$ empirical cross gramian matrix

$P \times P$ empirical cross-identifiability gramian matrix

The empirical variant of the non-symmetric cross gramian W_Z from Section 3.2.3 is computable by the empirical cross gramian (`'x'`) together with activating the non-symmetric option (`nf(7) = 1`), and can also be used with the empirical joint gramian (`'j'`).

[54] The input function `ut` and the type of centering `nf(1)` controls whether an empirical gramian matrix or an empirical covariance matrix is selected. For `ut=1` and `nf(1)=3` an empirical gramian is computed.

[55] A cell-array is a generic container type array in the Matlab programming language.

5.3.3. Option Flags

The vector `nf` encodes twelve options, which are configured by an integer value at the respective position; as a default value zero is assumed for unset options. It is noted, if an option is only available for a certain subset of empirical gramians listed in Section 5.3.2.

`nf(1)` The method of centering simulated (output) trajectories:

- `= 0` zero, no centering;
- `= 1` initial state (output), for bPOD;
- `= 2` steady-state (output), for empirical covariance matrices;
- `= 3` arithmetic average of states (outputs) over time, for empirical gramians;
- `= 4` median of the states (outputs) over time;
- `= 5` midrange of states (outputs) over time;
- `= 6` root-mean-square of states (outputs) over time.

`nf(2)` Input scale sub-division of (scalar- and vector-valued) input scales `um`:

- `= 0` linear sub-division into four scales: `um` ← `um*[0.25,0.50,0.75,1.0]`;
- `= 1` logarithmic sub-division into four scales: `um` ← `um*[0.001,0.01,0.1,1.0]`;
- `= 2` geometric sub-division into four scales: `um` ← `um*[0.125,0.25,0.5,1.0]`;
- `= 3` no sub-division, just a single scale: `um` ← `um`;
- `= 4` sparse sub-division into four scales: `um` ← `um*[0.38,0.71,0.92,1.0]`.

`nf(3)` State scale sub-division of (scalar- and vector-valued) state scales `xm`:

- `= 0` linear sub-division into four scales: `xm` ← `xm*[0.25,0.50,0.75,1.0]`;
- `= 1` logarithmic sub-division into four scales: `xm` ← `xm*[0.001,0.01,0.1,1.0]`;
- `= 2` geometric sub-division into four scales: `xm` ← `xm*[0.125,0.25,0.5,1.0]`;
- `= 3` no sub-division, just a single scale: `xm` ← `xm`;
- `= 4` sparse sub-division into four scales: `xm` ← `xm*[0.38,0.71,0.92,1.0]`.

`nf(4)` Input transformation configuration emulating rotations:

- `= 0` unit and negative unit transformations $\{-\mathbb{1}, \mathbb{1}\}$: `um` ← `[um, -um]`;
- `= 1` unit and reciprocal transformations: `um` ← `[um, 1./um]`;
- `= 2` dyadic transformation `um` ⊗ `um`: `um` ← `um` ∗ `um`$^\mathsf{T}$;
- `= 3` single unit transformation $\mathbb{1}$: `um` ← `um`.

`nf(5)` State transformation configuration emulating rotations:

- `= 0` unit and negative unit transformations $\{-\mathbb{1}, \mathbb{1}\}$: `xm` ← `[xm, -xm]`;
- `= 1` unit and reciprocal transformations: `xm` ← `[xm, 1./xm]`;
- `= 2` dyadic transformation `xm` ⊗ `xm`: `xm` ← `xm` ∗ `xm`$^\mathsf{T}$;
- `= 3` single unit transformation $\mathbb{1}$: `xm` ← `xm`.

`nf(6)` System preconditioning:

- `= 0` regular run;
- `= 1` double run, equilibrate based on empirical gramian diagonal [58];
- `= 2` scaled run, equilibrate based on steady-state (input), see [219].

`nf(7)` cross gramian for non-square or non-symmetric systems, only W_X, W_J:

- `= 0` regular cross gramian;
- `= 1` non-symmetric cross gramian variant, see Definition 3.17.

`nf(8)` Robust parameters, only W_C, W_Y:

- `= 0` plain parameters;
- `= 1` treat parameters as inputs, see Section 3.3.4.

`nf(9)` Parameter action, only W_I, W_J:

- `= 0` parameter is a source;
- `= 1` parameter is not a source (applies `ut` for parameter perturbed simulations).

`nf(10)` Center parameter scales:

- `= 0` no centering;
- `= 1` centering around arithmetic mean;
- `= 2` centering around logarithmic mean.

`nf(11)` Exclusive options for parameter gramians, only W_S, W_I, W_J:

- `= 0` disabled;
- `= 1` if W_S: computes the root-mean-square centered sensitivity gramian, see (3.28); by default the sensitivity gramian comprises the traces of the parameter controllability gramians.
- `= 1` if W_I: computes the approximate Schur-complement of the augmented observability gramian as identifiability gramian using the approximate inverse (Appendix A.4), see (3.30); by default the lower right block of the augmented observability gramian is computed as approximate identifiability gramian;
- `= 1` if W_J: computes the detailed Schur-complement of the joint gramian as cross-identifiability gramian using the pseudo-inverse; by default the approximate Schur-complement is computed using the approximate inverse (Appendix A.4).

`nf(12)` Empirical gramian symmetry:

- `= 0` assume symmetry;
- `= 1` enforce symmetry by post-processing.

Default and Custom ODE Solver

A default solver, the 2nd order Ralston's Runge-Kutta method, is available for `emgr`. Runge-Kutta methods are iterative numerical integrators for nonlinear IVPs and a special case of general linear methods for integration. The provided default solver[56] for the software implementations is an explicit single-step second-order Runge-Kutta method from [181]. This Ralston Runge-Kutta-2 (RK2) method has a minimal local truncation error of all second order Runge-Kutta methods, and additionally it is the only (pseudo-)energy preserving Runge-Kutta of this order [35].

Algorithm 9: 2^{nd} Order Ralston's Runge-Kutta Method

$$k_1 = hf(t_i, y_i)$$
$$k_2 = hf(t_i + \tfrac{2}{3}h, y_i + \tfrac{2}{3}k_1)$$
$$y_{i+1} = y_i + \tfrac{1}{4}k_1 + \tfrac{3}{4}k_2$$

The Courant-Friedrichs-Lewy (CFL) coefficient [89] c for the above method computes as:

$$c = \frac{3}{2}\min\Big(\frac{1}{4}, \frac{3}{4}\Big) = \frac{3}{8},$$

hence for a space-time discretization ratio $\frac{\Delta t}{\Delta x} < \frac{3}{8}$ the Ralston RK2 method is strong stability preserving (SSP) and conforms to hyperbolic conservation laws, but it is not an optimal two stage second-order SSP method.
Generally, a custom solver, adapted to the specific problem, is preferable. A custom ODE solver can be passed by creating a global variable with the name `ODE` which is set to the function handle of the desired procedure. This integrator is expected to have the signature:

```
y ← ODE(f,g,t,x,u,p);
```

and expects the following arguments:

- `f` *function handle* to a function with signature `xdot ← f(x,u,p)`, the system's vector field;
- `g` *function handle* to a function with signature `y ← g(x,u,p)`, the system's output functional; in case $g = 1$, a state trajectory is expected to be returned;
- `t` *two component vector* `t = [h,T]` holding the time step h and stopping time $\mathscr{T}$;
- `x` *vector* initial condition for the state-space of dimension $N \times 1$;
- `u` *matrix* discrete input time series matrix of dimension $J \times \frac{\mathscr{T}}{h}$;
- `p` *vector* column vector holding parameters of dimension $P \times 1$;

and returns a discrete state trajectory of dimension $N \times \frac{\mathscr{T}}{h}$ or output trajectory of dimension $O \times \frac{\mathscr{T}}{h}$.

[56] See `code/rk2.m` in the supplementary source code archive (Appendix B.1).

5.3.4. Verification and Validation

In this section the computation of the empirical gramians using the empirical gramian framework (`emgr`) is verified and validated[57]. For these tests the system (5.1) is utilized. Due to the origin of the empirical gramian method, apart from the combined reduction, the general applicability for state-space and parameter-space reduction is tested.

Verification

The correctness of the computation of the empirical gramians is verified, first by testing the pairwise difference of the state-space empirical gramians for the state-space symmetric test system, and second, by comparing the empirical gramian to an algebraic system gramian computed by solving a matrix equation. For these computations the source matrix F of the test system is set to zero to disable the influence of the parameters.
Since all state-space system gramians are equal for a state-space symmetric system (3.7), the state-space empirical gramians, which are: the empirical controllability gramian W_C, empirical observability gramian W_O, empirical cross gramian W_X and empirical linear cross gramian W_Y, are compared with one another in the Frobenius norm[58].
For a state-space symmetric system, W_C is computation-wise identical to the W_Y; thus, the Frobenius norm of the difference is also numerically zero:

$$\|W_C - W_Y\|_F = 0,$$

which means the error is of the order of precision of the floating point number representation. The same holds for an observability gramian computed as the empirical controllability gramian of the adjoint system W_C^*:

$$\|W_C - \underbrace{W_C^*}_{=W_O}\|_F = 0.$$

Comparing W_C with W_O and W_X as well as the W_O with W_X, with W_O and W_X computed following Definition 3.30 and Definition 3.35 respectively, an error correlated with the times step-size h remains:

$$\|W_C - W_O\|_F \xrightarrow{h\to 0} 0,$$
$$\|W_C - W_X\|_F \xrightarrow{h\to 0} 0,$$
$$\|W_O - W_X\|_F \xrightarrow{h\to 0} 0.$$

This error is due to the differing assembly of the empirical gramians; while W_C utilizes state trajectories, W_O uses output trajectories and W_X employs both. For more accurate simulations using either smaller time-steps, or higher order integrators this error can be reduced. Figure 5.1a illustrates the error decay for decreasing time-step size.

[57] See `code/ch5/vernval_emgr.m` in the supplementary source code archive (Appendix B.1).
[58] The empirical non-symmetric cross gramian is verified in [117].

Next, a system gramian W is computed "algebraically" by solving a Sylvester equation,

$$AW + WA = BC \Leftrightarrow AW + WA^\intercal = BB^\intercal \Leftrightarrow A^\intercal W + WA = C^\intercal C,$$

which corresponds to either, the controllability, observability and cross gramian since the test system is state-space symmetric. Now, this algebraic system gramian W is compared to the empirical gramians W_C, W_O and W_X in the Frobenius norm:

$$\begin{aligned} \|W_C - W\|_F &\overset{\mathscr{T}\to\infty}{\longrightarrow} 0, \\ \|W_O - W\|_F &\overset{\mathscr{T}\to\infty}{\longrightarrow} 0, \\ \|W_X - W\|_F &\overset{\mathscr{T}\to\infty}{\longrightarrow} 0. \end{aligned}$$

The utilized (output) trajectories used to assemble the empirical gramians need to attain their steady-states in all components to conform to the algebraic gramians. Since the test-system model's dynamical system contains slow modes, the error between empirical system gramians and (algebraic) system gramians reduces for longer integration times[59]. This relation is demonstrated in Figure 5.1b for varying integration time-spans used in the computation of the empirical gramians.

(a) *Relative error between numerically different but analytically equal empirical gramians for varying time-step width in the Frobenius norm.*

(b) *Relative error between different empirical gramians but analytically equal algebraic gramians for varying integration length in the Frobenius norm.*

Figure 5.1.: *Numerical verification of the empirical gramian computation.*

[59] Additionally, the accuracy of the integration is contributing to the error.

Validation

The validation of the empirical gramian computation is accomplished by the use of the state-space symmetric test system from (5.1). This system is tested in terms of state-space reduction, parameter-space reduction and combined state- and parameter-space reduction.

State-Space Reduction

First, empirical-gramian-based state-space reduction is tested for variants of balanced truncation and direct truncation in Figure 5.2. In this setting, the source matrix is set to zero $F = 0$, to exploit the equality of all gramians. The tested methods are evaluated in the (time-domain) ℓ_1-, ℓ_2- and ℓ_∞-norm of the output error (2.15) and in the associated error bounds and error indicators for varying reduced state-space dimensions.
Tested are:

a) balanced truncation using the empirical controllability gramian W_C and empirical observability gramian W_O, see Figure 5.2a;

b) balanced truncation using the empirical controllability gramian W_C and adjoint empirical controllability gramian W_C^*, see Figure 5.2b;

c) direct truncation using the empirical linear cross gramian W_Y,
see Figure 5.2c;

d) direct truncation using the empirical cross gramian W_X,
see Figure 5.2d;

e) direct truncation based on balanced gains using the empirical cross gramian W_X,
see Figure 5.2e;

f) direct truncation using the empirical non-symmetric cross gramian W_Z,
see Figure 5.2f.

Figure 5.2a, Figure 5.2b, Figure 5.2c and Figure 5.2d show that for a state-space symmetric system, the linear empirical balanced truncation (W_C, W_C^*) and nonlinear empirical balanced truncation (W_C, W_O) as well as the linear empirical direct truncation (W_Y) and nonlinear empirical direct truncation (W_X) produce nearly equivalent ROMs. Similarly, the ROM obtained by balanced gains (see Figure 5.2e) using empirical direct truncation is also almost equal to the empirical balanced truncation and direct truncation ROMs. The direct truncation by the non-symmetric cross gramian W_Z, as reported in [117], generates a ROM (see Figure 5.2f) for which the model reduction error decays faster, but does not reach the same accuracy as balanced and direct truncation.
The global $\mathscr{H}_\infty$-error bound (3.9), which in turn is a bound for the L_2-error (2.18), is tight for this system with a self-adjoint Hankel operator [173]. Also, the $\mathscr{H}_2$-error indicator (3.13), follows the error in the L_∞-norm (2.17) closely. Lastly, the L_1-error bounds all other errors which is predicted by the Minkowski inequality [84, Ch. 2.1.5].

(a) *BT of* W_C, W_O.

(b) *BT of* W_C, W_C^*.

(c) *DT of* W_Y.

(d) *DT of* W_X.

(e) *DT of* W_X *(balanced gains)*.

(f) *DT of* W_Z.

Figure 5.2.: *Relative* ℓ_1*-,* ℓ_2*-,* ℓ_∞*-norm output error,* L_1*-,* L_2*-norm error bound and* L_∞*-norm error indicator for varying reduced state dimensions using six gramian-based state-space reduction methods.*

Parametric State-Space Reduction

As a second test, the state-space reduction for the same test problem is conducted, yet now including the linear parametrization. In this setting the parameter $\theta \in \mathbb{R}^P$, $P = N$, is constrained to $\theta_i \in [0, 1]$ and the source matrix is set to the unit matrix $F = \mathbb{1}_N$. For the evaluations the parameters are selected from a uniformly random distribution $\mathscr{U}_{[0,1]}^P$. As described in Section 3.3.4 the parametric gramian-based reduction can be performed by averaging the associated system gramians for different parameter samples. The errors are evaluated in the joint state- and parameter-space norms (2.16). For these tests, the $\ell_1 \otimes \ell_2$-norm, $\ell_2 \otimes \ell_2$- and the $\ell_\infty \otimes \ell_2$-norm are employed. The results are depicted in Figure 5.3 for the parametric tests based on four parameter samples during the construction of the ROM and 100 uniformly random parameter samples for the ROM evaluation. Tested are the same gramian-based state reduction methods as for the non-parametric state reduction:

a) parametric balanced truncation using the empirical controllability gramian W_C and empirical observability gramian W_O, see Figure 5.3a;

b) parametric balanced truncation using the empirical controllability gramian W_C and adjoint empirical controllability gramian W_C^*, see Figure 5.3b;

c) parametric direct truncation using the empirical linear cross gramian W_Y, see Figure 5.3c;

d) parametric direct truncation using the empirical cross gramian W_X, see Figure 5.3d;

e) parametric direct truncation based on balanced gains using the empirical cross gramian W_X, see Figure 5.3e;

f) parametric direct truncation using the empirical non-symmetric cross gramian W_Z, see Figure 5.3f.

Overall, the shape of the error curves of the parametric state-space reduction tests are similar to the errors of the non-parametric state-space reduction tests, but at a lesser accuracy. For the parametric balanced truncation tests, illustrated in Figure 5.3a and Figure 5.3b, the parametric direct truncation tests in Figure 5.3c and Figure 5.3d, as well as the parametric balanced gains direct truncation in Figure 5.3e, the relative output errors also drop sharply, yet level out already at about 10^{-10} instead of 10^{-15} for the non-parametric state-space reduction. This loss of accuracy is due to the averaging of the respective system gramians for the chosen parameter samples.
The direct truncation based on the non-symmetric cross gramian Figure 5.3f performs worst in this setting. This is due to the repeated averaging; first, over the input and output matrices (see Lemma 3.18) and second, over the parameter samples. In this basic variant the non-symmetric cross gramian is not a viable method for pMOR.

(a) *Parametric BT of* W_C, W_O.

(b) *Parametric BT of* W_C, W_C^*.

(c) *Parametric DT of* W_Y.

(d) *Parametric DT of* W_X.

(e) *Parametric DT of* W_X *(balanced gains).*

(f) *Parametric DT of* W_Z.

Figure 5.3.: *Relative* $\ell_1 \otimes \ell_2$*-,* $\ell_2 \otimes \ell_2$*-,* $\ell_\infty \otimes \ell_2$*-norm output error for varying reduced state dimensions using six gramian-based parametric state-space reduction methods.*

Parameter-Space Reduction

Thirdly, empirical-gramian-based parameter-space reduction is tested for variants of direct truncation in Figure 5.4. These tests have the same parameter setup as the parametric state-space reduction. The following empirical parameter gramians are tested for varying reduced parameter-space dimension over 100 uniformly random samples from the parameter-space in the $\ell_1 \otimes \ell_2$-norm, $\ell_2 \otimes \ell_2$- and the $\ell_\infty \otimes \ell_2$-norm:

a) direct truncation of the empirical sensitivity gramian W_S,
see Figure 5.4a;

b) direct truncation of the empirical identifiability gramian W_I,
see Figure 5.4b;

c) direct truncation of the empirical cross-identifiability gramian $\widetilde{W}_{\tilde{I}}$,
see Figure 5.4c;

d) direct truncation of the centered empirical sensitivity gramian $\overline{W}_S$,
see Figure 5.4d;

e) direct truncation of the approximate empirical identifiability gramian $\widetilde{W}_I$,
see Figure 5.4e;

f) direct truncation of the approximate empirical cross-identifiability gramian $W_{\tilde{I}}$,
see Figure 5.4f.

In comparison, the controllability-based parameter reduction using the empirical sensitivity gramian performs worst (Figure 5.4a). This is due to parameter projection being based only on sorting the parameters by their average controllability and not composing linear combinations of parameters as the observability-based methods do. And since the parameter-space is rather homogeneous the parameters have all about the same sensitivities. The centering of the sensitivity gramian in Figure 5.4d improves the ROMs slightly, but in relation to the observability-based parameter reduction remains least useful. The observability-based identifiability gramian presents a better accuracy in the ROMs than the controllability-based approach, and the (approximate) identifiability gramian yields the overall best results reaching an error of about 10^{-12}; see Figure 5.4b and Figure 5.4e. Figure 5.4c shows, that the parameter reduction error related to the cross-gramian-based cross-identifiability gramian, which is also observability-based, has a comparable initial decay in errors as the (approximate) identifiability gramian, but levels out at about 10^{-8}. The approximate cross-identifiability gramian, in Figure 5.4f, performs with a lesser accuracy yielding ROMS with an error of about 10^{-5} after the initial descent. Notably, for the empirical identifiability and cross-identifiability gramians, the error curves decay steeply, similar to the gramian-based state-space reduction. This indicates that the principal parameter subspace is relatively low-dimensional.
Compared to the parametric state-space reduction, the parameter-space reduction reaches similar error levels, but not of the order of the machine precision for double-precision floating-point arithmetic $\mathsf{eps} \approx 10^{-15}$ which is used for all numerical computations in this work.

(a) *DT of* W_S. (b) *DT of* W_I. (c) *DT of* $W_{\bar{I}}$.

(d) *DT of* $\overline{W}_S$. (e) *DT of* $\widetilde{W}_I$. (f) *DT of* $\widetilde{W}_{\bar{I}}$.

Figure 5.4.: *Relative* $\ell_1 \otimes \ell_2$-, $\ell_2 \otimes \ell_2$-, $\ell_\infty \otimes \ell_2$-*norm output error for varying reduced parameter dimensions using six gramian-based parameter-space reduction methods.*

Combined State and Parameter Reduction

Lastly, the combined (state and parameter) reduction, which fuses the three previously presented parameter reduction methods with the (empirical) gramian-based state reduction, as described in Section 3.5, is tested in Figure 5.5. For these tests the previously employed linear system including the linear parametrization (3.26) is utilized. Compared are the combined reduction methods for varying state- and parameter-space dimensions in the $\ell_1 \otimes \ell_2$-norm, $\ell_2 \otimes \ell_2$- and the $\ell_\infty \otimes \ell_2$-norm over uniformly random 100 samples from the parameter-space:

a) balanced truncation of the approximate empirical controllability gramian and empirical observability gramian $\{\widetilde{W}_C, W_O\}$ and direct truncation of the centered empirical sensitivity gramian $\overline{W}_S$ (controllability-based), see Figure 5.5a;

b) balanced truncation of the empirical controllability gramian and empirical observability gramian $\{W_C, W_O\}$ and direct truncation of the approximate empirical identifiability gramian $\widetilde{W}_I$ (observability-based), see Figure 5.5b;

c) direct truncation of the empirical cross gramian W_X and direct truncation of the empirical cross-identifiability gramian $\widetilde{W}_{\tilde{I}}$ (cross-gramian-based), see Figure 5.5c;

d) controllability-based combined reduction as in a), but restricted to the same reduced state- and parameter-space dimension $n = p$, see Figure 5.5d;

e) observability-based combined reduction as in b), but restricted to the same reduced state- and parameter-space dimension $n = p$, see Figure 5.5e;

f) cross-gramian-based combined reduction as in c), but restricted to the same reduced state- and parameter-space dimension $n = p$, see Figure 5.5f.

The pairing of balanced truncation (using the approximate controllability gramian which is assembled during the computation of the sensitivity gramian) for state-space reduction with direct truncation of the sensitivity gramian for parameter-space reduction in Figure 5.5a performs worst, which is due to insufficient parameter reduction capabilities of the sensitivity gramian. Both observability-based combined reduction methods, the combination of balanced truncation (using the observability gramian which is assembled during the computation of the identifiability gramian) for state-space reduction and direct truncation of the identifiability gramian for parameter-space reduction illustrated in Figure 5.5b, and the cross-gramian-based direct truncation (using the cross gramian assembled during the computation of the cross-identifiability gramian) for state-space reduction and the direct truncation of the approximate cross-identifiability gramian for parameter-space reduction (Figure 5.5c), reach error plateaus similar to the parameter reduction errors with an initial steep decline.
The cross-section of the error surfaces for same reduced state-space and parameter-space dimension (Figure 5.5d, Figure 5.5e, Figure 5.5f) confirm the previous observations on the combined reduction methods.

(a) *BT of* $\widetilde{W}_C$, W_O *and DT* $\overline{W}_S$.

(b) *BT of* W_C, W_O *and DT of* $\widetilde{W}_I$.

(c) *DT of* W_X *and DT of* $W_{\bar{I}}$.

(d) *BT of* $\widetilde{W}_C$, W_O *and DT of the* $\overline{W}_S$ *for* $n = p$.

(e) *BT of* W_C, W_O *and DT of the* $\widetilde{W}_I$ *for* $n = p$.

(f) *DT of* W_X *and DT of* $W_{\bar{I}}$ *for* $n = p$.

Figure 5.5.: *Relative* $\ell_2 \otimes \ell_2$*-norm of the output error for varying reduced state and parameter dimensions using three gramian-based combined reduction methods and relative* $\ell_1 \otimes \ell_2$*-,* $\ell_2 \otimes \ell_2$*-,* $\ell_\infty \otimes \ell_2$*-norm output error for equally varying reduced state and parameter dimension.*

5.4. optmor - Optimization-Based Model Order Reduction

The optimization-based model order reduction framework[60] (`optmor`) [108] provides an implementation of the algorithms in Chapter 4 and is designed to allow a modular combination of the presented variants. It should be noted, that `optmor` is explicitly designed for the purpose of combined state- and parameter-space reduction. `optmor` is also a single file implementation of less than 300 LoC with no dependencies on other toolboxes and is released under an open-source license (Appendix B.4).
The use of `optmor` is similar to `emgr`, yet as opposed to `emgr` this implementation directly returns the low-rank state- and parameter-space truncated projections. The core component of `optmor` is the inner optimization algorithm of the greedy sampling, for which default choices as well as an interface for custom optimizers are available.
Following, the interface of the `optmor` toolbox[61], the core features and the available configurable options are listed and described. Then, the functionality is verified and validated using the benchmark problem from (5.1).

5.4.1. Interface

The signature of the `optmor` function encompasses six mandatory arguments (single letter) and five optional arguments (double letter), and reads:

```
XP = optmor(f,g,s,t,r,q,nf,ut,x0,co,yd);
```

Hence, the minimal usage:

```
XP = optmor(f,g,s,t,r,q);
```

requires six mandatory arguments:

- `f` *function handle* to a function with signature `xdot ← f(x,u,p)`, the system's vector field depending on the current state `x`, the current input `u` and the (current) parameter `p`. For example, in case of a linear system: `f = @(x,u,p) A*x + B*u;`
- `g` *function handle* to a function with signature `y ← g(x,u,p)`, the system's output functional depending on the current state `x`, the current input `u` and the (current) parameter `p`. For example in case of a linear system: `g = @(x,u,p) C*x;`
- `s` *three component vector* `s = [J,N,O]` holding number of inputs J, states N and outputs O;
- `t` *two component vector* `t = [h,T]` holding time-step h and stop time $\mathcal{T}$;
- `r` *positive scalar* either an integer setting a target reduced order to the parameter-space or a floating-point number (< 1) setting the overall target output error;
- `q` *vector* holding the prior or nominal parameters of dimension $P \times 1$, initializing the greedy sampling.

[60] See the listing in Appendix B.4.
[61] For `optmor` in version 2.5.

Furthermore, five optional arguments are provided:

`nf` *six component vector* encoding the option settings, for details see Section 5.4.3.

`ut` input time series, default value: 1;

scalar setting uniform peak of impulse input to all inputs, normalized by the time-step `h` to ensure the discrete input fulfills the identity $\int_0^{\mathscr{T}} u_i(t)\,\mathrm{d}t = \texttt{ut(1)}$,

vector setting per-input component peaks to impulse input, normalized by the time-step `h` to ensure the discrete input fulfills the identity $\int_0^{\mathscr{T}} u_i(t)\,\mathrm{d}t = \texttt{ut(i)}$ and is of dimension $J \times 1$,

matrix holding the discrete input time series of dimension $J \times \frac{\mathscr{T}}{h}$,

handle to a function `u` $\leftarrow$ `uf(t)` which is internally discretized,
for example a Gaussian pulse at time `tp` and full width at half maximum `h`:
`u = @(t) exp(-(0.5/h)*(t-tp).^2)./sqrt(2.0*pi*h)`,

∞ generates a linear chirp signal using the havercosine [226]:
`u = @(t) 0.5*cos(pi*(t+10*t.*t))+0.5`

`x0` nominal initial state, default value: 0;

scalar sets all initial state components to provided value;

vector holding initial state of dimension $N \times 1$,

`co` prior covariance information, default value: 0;

scalar uniform prior variance on all parameter components,

vector holding per parameter component prior variance of dimension $P \times 1$,

matrix holding variance-covariance matrix of dimension $P \times P$;

`yd` *matrix* of dimension $O \times \frac{\mathscr{T}}{h}$ holding the discrete target time-series; required argument for data-driven regularization from Section 4.4 activated by `nf(4)>0`.

5.4.2. Features

The optimization-based model order reduction toolbox features the concurrent computation of

- a state-space Galerkin projection and
- a parameter-space Galerkin projection,

which are returned as a cell-array of two projection matrices in `XP`. Contained in the first component `XP{1}` is the state-space projection matrix, while the second component `XP{2}` holds the parameter-space projection matrix. The different variants of the greedy sampling over the parameter-space and the selection of state-space modes can be configured by the option vector `nf`.

5.4.3. Option Flags

The configurable options are encoded in a six component vector passed by the argument `nf`, wherein each component represents a separate option.

`nf(1)` Type of optimization algorithm utilized for the greedy sampling, default value: 0;

- `= 0` unconstrained optimization[62] (Quasi-Newton) ,
- `= 1` derivative-free optimization[63] (Nelder-Mead),
- `=-1` custom optimizer.

`nf(2)` Non-negative floating-point weighting coefficient β_1 for L_1 regularization operator[64] in the greedy cost functional, default value: 0.

`nf(3)` Non-negative floating-point weighting coefficient β_2 for L_2 regularization operator in the greedy cost functional see (Section 4.1.3), default value: $\frac{1}{10}$.

`nf(4)` Non-negative floating-point weighting coefficient β_d for data-driven regularization operator in the greedy cost functional, see Section 4.4, default value: 0.

`nf(5)` Non-negative integer setting the maximum number of iterations in the greedy sampling optimization algorithm per iteration, default value: 4.

`nf(6)` Selects the greedy sampling initial parameter;

- `= 0` last iterations maximizer,
- `= 1` last iterations maximizer with added multivariate standard normal distribution sample.

In case a custom optimizer is to be used via `nf(1)=-1`, a function handle to (a wrapper function around) the optimization procedure is passed through a global variable named `FMIN` and expects the following signature:

```
p ← FMIN(J,p0).
```

with the arguments:

`J` *function handle* to the cost function with signature `c ← J(p)`, which given a parameter `p` returns an associated scalar cost `c`;

`p0` *vector* column vector holding the initial nominal parameter of dimension P;

and returns the next error maximizing parameter `p`. Standard minimizing algorithms are applicable here, since the cost value is internally negated (4.7). This minimal interface can also be used indirectly to supply additional information to the optimizer by nested wrapper functions.
A custom solver for the computation of the (output) trajectories, required by the state-space reduction and the optimization algorithms, can be set in the same manner as in Section 5.3.3 by passing an integrator function handle by a global variable named `ODE`.

[62] Uses `fminunc`, see also: `octave.sourceforge.net/octave/function/fminunc.html` .
[63] Uses `fminsearch`, see also: `octave.sourceforge.net/octave/function/fminsearch.html` .
[64] The L_1 regularization is not evaluated in this work.

5.4.4. Verification and Validation

In this section the computation of the combined state and parameter reduction using the optimization-based model order reduction (`optmor`) is verified and validated[65]. Since `optmor` is exclusively targeted at system with a high-dimensional parameter-space only the parametric setting is considered.

Verification

To verify the computations of the optimization-based approach, multiple ROMs are computed using the default options and validated for the (same) test system (5.1) with varying nominal (initial) parameters. The resulting ROMs are compared in terms of parametric state-space reduction, parameter-space reduction and combined reduction in the joint $\ell_2 \otimes \ell_\infty$-norm over 100 uniform random parameter samples $\mathscr{U}_{[0,1]}^{P}$ in Figure 5.6.
The parametric state-space reduction in Figure 5.6a, the parameter-space reduction in Figure 5.6b and the combined state- and parameter-space reduction with evenly reduced state and parameter dimensions in Figure 5.6c, show that the resulting errors are contained in a narrow range in the initial steep decline. Even though this does not substantiate a unique ROM, it indicates very similar ROMs for varying nominal parameters.

(a) *Output error for the parametric state-space reduction.*

(b) *Output error for the parameter-space reduction.*

(c) *Output error for the even combined state and parameter reduction.*

Figure 5.6.: *Output errors in the $\ell_2 \otimes \ell_\infty$-norm for optimization-based reduced order models computed with varying nominal parameters.*

[65] See `code/ch5/vernval_optmor.m` in the supplementary source code archive (Appendix B.1).

Validation

The validation for the optimization-based model order reduction is performed using the parametric state-space symmetric system (5.1) generated by the ILP. Following, four tests are conducted: the ROM quality is compared for different Tikhonov regularization coefficients, different maximum iterations of the inner optimization procedure, varying prescribed error thresholds and different data-driven regularization coefficients.

Tikhonov Regularization

First, the influence of the regularization coefficient β_2 for the Tikhonov regularization operator $\mathscr{R}_{\beta_2}$ (see Section 4.1.3) in the optimization algorithm of the greedy sampling is tested. The regularization coefficient balances the soft constraint of the optimization problem against the maximization. Even though this coefficient is specific to the test problem, it can provide a starting point for other systems.
The following values of coefficient β_2 are tested, for varying reduced state- and parameter-space dimensions over 100 uniform random samples from the parameter-space, using the relative output error in the joint $\ell_2 \otimes \ell_\infty$-norm in Figure 5.7:

a) optimization-based combined reduction without Tikhonov regularization, meaning $\beta_2 = 10^{-16} \approx 0$, see Figure 5.7a;

b) optimization-based combined reduction using a Tikhonov regularization coefficient $\beta_2 = 10^{-4}$, see Figure 5.7b;

c) optimization-based combined reduction using a Tikhonov regularization coefficient $\beta_2 = 10^{-3}$, see Figure 5.7c;

d) optimization-based combined reduction using a Tikhonov regularization coefficient $\beta_2 = 10^{-2}$, see Figure 5.7d;

e) optimization-based combined reduction using a Tikhonov regularization coefficient $\beta_2 = 10^{-1}$, see Figure 5.7e;

f) optimization-based combined reduction using a Tikhonov regularization coefficient $\beta_2 = 1$, see Figure 5.7f.

Overall, the error is dominated by the parameter-space reduction. Figure 5.7a shows, that the optimization-based combined reduction without regularization of the greedy sampling ($\beta_2 = 0$) produces the worst results; the error surface flattens already at a relative error of about 10^{-5} and continues to decay slowly. For the regularization coefficients $\beta_2 \in \{10^{-4}, 10^{-3}, 10^{-2}\}$ the error falls steeply to about 10^{-6} for either, the varying reduced state- and parameter-space dimension as illustrated in Figure 5.7b, Figure 5.7c, Figure 5.7d. Given equal influence of the Tikhonov regularization and the maximization term $\beta_2 = 1$, with results depicted in Figure 5.7f; the error also falls to about 10^{-7} as for $\beta_2 < 10^{-1}$, but continues to decay slightly steeper. The best results are obtained for $\beta_2 = 10^{-1}$, for which the error steeply declines to an error plateau at about 10^{-8} concurrently for reduced state- and parameter-spaces, as illustrated in Figure 5.7e.

(a) *Optimization-based combined reduction with* $\beta_2 = 10^{-16} \approx 0$.

(b) *Optimization-based combined reduction with* $\beta_2 = 10^{-4}$.

(c) *Optimization-based combined reduction with* $\beta_2 = 10^{-3}$.

(d) *Optimization-based combined reduction with* $\beta_2 = 10^{-2}$.

(e) *Optimization-based combined reduction with* $\beta_2 = 10^{-1}$.

(f) *Optimization-based combined reduction with* $\beta_2 = 1$.

Figure 5.7.: *Relative output error in the* $\ell_2 \otimes \ell_\infty$*-norm for varying reduced state and parameter dimensions, using different Tikhonov regularization coefficients for the optimization-based combined reduction.*

Inner Greedy Sampling Iterations

As it is not necessary and usually not feasible to require a certain error tolerance from the optimization algorithm to be reached in each iteration of the greedy sampling, a fixed number of iterations is prescribed to the optimizer. In this second test, the optimization-based combined reduction is tested for different numbers of iterations in the optimization method realizing the maximization in each cycle of the greedy sampling. This test is specifically conducted for the unconstrained optimization algorithm; for alternative choices such as the derivative-free optimization or a custom optimizer these results are not valid. An increasing number of iterations n_I is tested, for varying reduced state- and parameter-space dimensions over 100 uniform random samples from the parameter-space, using the relative output error in the joint $\ell_2 \otimes \ell_\infty$-norm in Figure 5.8:

a) optimization-based combined reduction using $n_I = 2$ iterations in the optimizer, see Figure 5.8a;

b) optimization-based combined reduction using $n_I = 3$ iterations in the optimizer, see Figure 5.8b;

c) optimization-based combined reduction using $n_I = 4$ iterations in the optimizer, see Figure 5.8c;

d) optimization-based combined reduction using $n_I = 5$ iterations in the optimizer, see Figure 5.8d;

e) optimization-based combined reduction using $n_I = 6$ iterations in the optimizer, see Figure 5.8e;

f) optimization-based combined reduction using $n_I = 10$ iterations in the optimizer, see Figure 5.8f.

Figure 5.8a shows that two iterations of the inner optimization are not sufficient to reach a comparable accuracy. The error declines steeply to 10^{-5} and continues onward to decline gradually, dominated by the parameter-space error. For three and four iterations, as depicted in Figure 5.8b and Figure 5.8c, the error to the FOM falls steeply to about 10^{-7} for the state- and parameter-space error. While for three iterations the error slowly decreases after the initial drop, using four iterations the error flattens out already at a lower level. In case of five (see Figure 5.8d) or more iterations (Figure 5.8e, Figure 5.8f) the error decreases to about 10^{-8}.

The number of iterations predominantly defines the overall computational complexity of the optimization-based approach. Unlike the choice of regularization coefficients, the number of iterations does not only affect the accuracy of the ROM, but also the duration of its construction. This test system does not benefit from more than four iterations, yet models with more complex vector fields, output functionals or parametrizations may improve with additional iterations. Hence, a heuristic selection of a suitable number of iterations balancing accuracy and complexity may be necessary.

(a) *Optimization-based combined reduction with* $n_I = 2$.

(b) *Optimization-based combined reduction with* $n_I = 3$.

(c) *Optimization-based combined reduction with* $n_I = 4$.

(d) *Optimization-based combined reduction with* $n_I = 5$.

(e) *Optimization-based combined reduction with* $n_I = 6$.

(f) *Optimization-based combined reduction with* $n_I = 10$.

Figure 5.8.: *Relative output error in the* $\ell_2 \otimes \ell_\infty$*-norm for varying reduced state and parameter dimensions, using different numbers of optimizer iterations for the optimization-based combined reduction.*

Threshold Determined Reduced Order

For the first two tests a parameter-space reduced order is prescribed and determines the number of iterations of the optimization-based combined reduction algorithm. In this third test a threshold on the output error of the full to the state and parameter reduced model is selected, which is tested with each iteration's approximate maximizer from the greedy sampling,

$$\varepsilon_{\max} = \|y(\theta_I) - y_r(\Pi_I \Pi_I^{\mathsf{T}} \theta_I)\|_{\ell_2}.$$

Following, for different error thresholds $\varepsilon_{\max}$, the $\ell_2 \otimes \ell_\infty$ error for varying reduced state- and parameter-space dimensions is depicted in Figure 5.9 over 100 samples from the parameter-space:

a) optimization-based combined reduction with a threshold $\varepsilon_{\max} = 10^{-2}$, see Figure 5.9a;

b) optimization-based combined reduction with a threshold $\varepsilon_{\max} = 10^{-4}$, see Figure 5.9b;

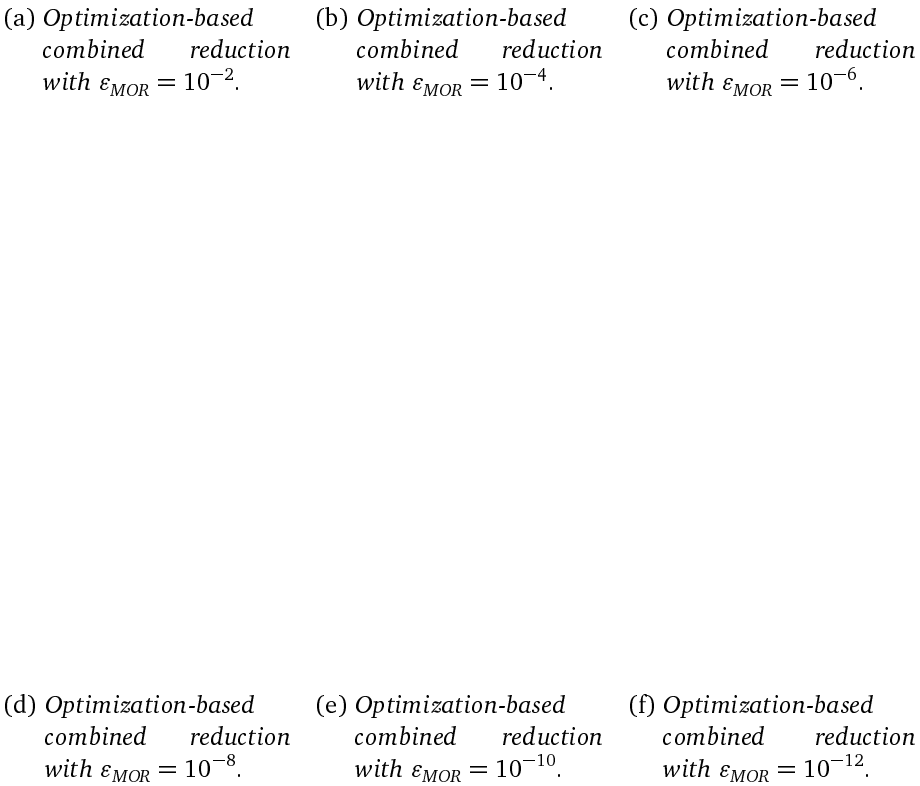
c) optimization-based combined reduction with a threshold $\varepsilon_{\max} = 10^{-6}$, see Figure 5.9c;

d) optimization-based combined reduction with a threshold $\varepsilon_{\max} = 10^{-8}$, see Figure 5.9d;

e) optimization-based combined reduction with a threshold $\varepsilon_{\max} = 10^{-10}$, see Figure 5.9e;

f) optimization-based combined reduction with a threshold $\varepsilon_{\max} = 10^{-12}$, see Figure 5.9f.

First of all it should be noted, that the error threshold $\varepsilon_{\max}$ is not a guaranteed bound on the resulting error of the ROM. This is due to the fixed number of iterations in the optimization implementing the greedy sampling, since a maximizer is most likely not found after the prescribed small number of iterations. Hence, $\varepsilon_{\max}$ has to be understood as a virtual error indicator.
For the threshold $\varepsilon_{\max} = 10^{-2}$ an error of about 10^{-2} is reached, while for $\varepsilon_{\max} = 10^{-4}$ an error of about 10^{-5} is obtained. With a thresholds $\varepsilon_{\max} = 10^{-6}$ and $\varepsilon_{\max} = 10^{-8}$ only an error level of about 10^{-5} is reached. Using a threshold $\varepsilon_{\max} = 10^{-10}$ provides a ROM with an error of about 10^{-6} and lastly for $\varepsilon_{\max} = 10^{-12}$ an error of 10^{-7} is achieved. These results illustrate the previous comment on $\varepsilon_{\max}$; even though lower values of $\varepsilon_{\max}$ result in lower errors, it is does not bound the resulting error. An improved significance of this virtual error indicator $\varepsilon_{\max}$ can be obtained for better approximations of the maximizer θ_I. The use of the threshold-based reduced order selection requires careful selection of the error indicating threshold.



(a) *Optimization-based combined reduction with* $\varepsilon_{MOR} = 10^{-2}$.

(b) *Optimization-based combined reduction with* $\varepsilon_{MOR} = 10^{-4}$.

(c) *Optimization-based combined reduction with* $\varepsilon_{MOR} = 10^{-6}$.

(d) *Optimization-based combined reduction with* $\varepsilon_{MOR} = 10^{-8}$.

(e) *Optimization-based combined reduction with* $\varepsilon_{MOR} = 10^{-10}$.

(f) *Optimization-based combined reduction with* $\varepsilon_{MOR} = 10^{-12}$.

Figure 5.9.: *Relative output error in the* $\ell_2 \otimes \ell_\infty$*-norm for varying reduced state and parameter dimensions, using different error thresholds dynamically selecting the reduced order for the optimization-based combined reduction.*

Data-Driven Regularization

Lastly, the influence of the data-driven regularization from Section 4.4 tested. For this test, a single uniformly distributed random sampled parameter θ_d is selected, and the reduced order models obtained with and without data-driven regularization are tested over 100 multivariate normally distributed random samples $\mathcal{N}_{\theta_d, \frac{1}{10}}$. The errors are compared in the $\ell_2 \otimes \ell_\infty$-norm for varying data-driven regularization coefficients β_d. The Tikhonov regularization coefficient is kept at its default value $\beta_2 = \frac{1}{10}$, and an associated output trajectory $y_d(\theta_d)$ is passed to the reduction algorithm in case of $\beta_d \neq 0$. Figure 5.10 shows the evaluation for varying reduced state- and parameter-space dimensions for the following data-driven regularization weights β_d:

a) optimization-based combined reduction without data-driven regularization, meaning $\beta_d = 0$, see Figure 5.10a;

b) optimization-based combined reduction using a data-driven regularization coefficient $\beta_d = 10^{-12}$, see Figure 5.10b;

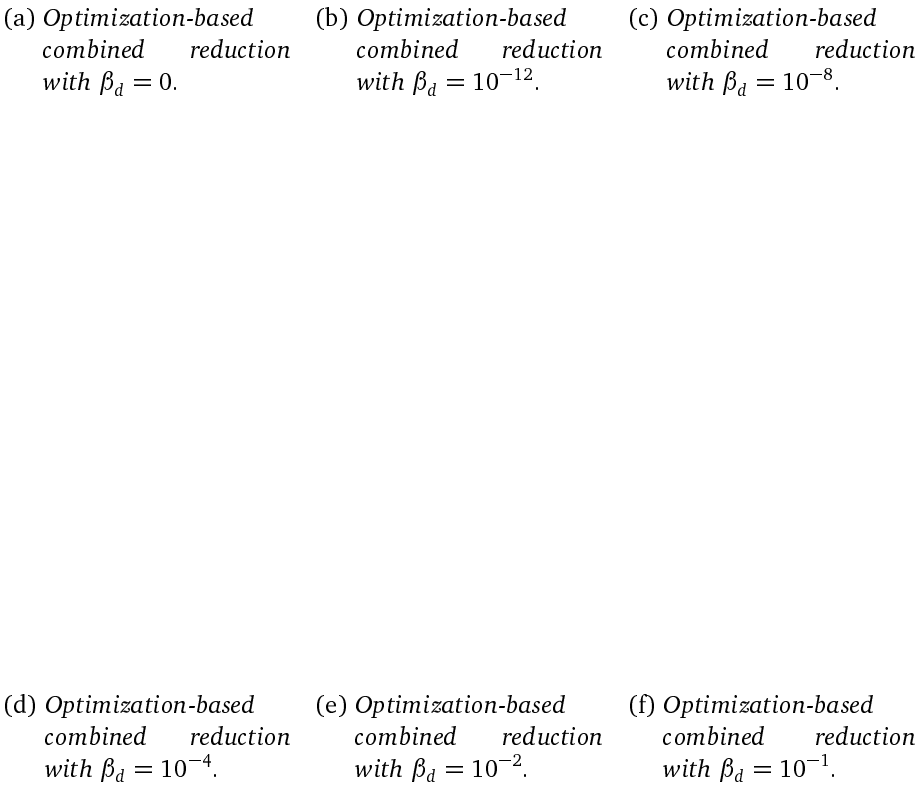
c) optimization-based combined reduction using a data-driven regularization coefficient $\beta_d = 10^{-8}$, see Figure 5.10c;

d) optimization-based combined reduction using a data-driven regularization coefficient $\beta_d = 10^{-4}$, see Figure 5.10d;

e) optimization-based combined reduction using a data-driven regularization coefficient $\beta_d = 10^{-2}$, see Figure 5.10e;

f) optimization-based combined reduction using a data-driven regularization coefficient $\beta_d = 10^{-1}$, see Figure 5.10f.

Without using the data-driven regularization, the resulting errors for a fixed parameter in Figure 5.10a correspond to the previous results, with a steep decline in error for both, the reduced state and parameter dimension, to a plateau near 10^{-7}.
In the tests using relatively larger regularization coefficients $\beta_d = \{10^{-4}, 10^{-2}, 10^{-1}\}$, the obtained ROMs exhibit lesser accuracy in the combined reduction error than the ROMs without data-driven regularization, which is illustrated in Figure 5.10d, Figure 5.10e and Figure 5.10f. Yet, the ROMs from the test with smaller regularization coefficients $\beta_d = \{10^{-12}, 10^{-8}\}$ are more slightly accurate than the ROMs without data-driven regularization, as shown in Figure 5.10b and Figure 5.10c. The disparity in scale of regularization coefficients β_d and β_2 is to be expected since the associated operators evaluate different expressions. While $\mathscr{R}_2$ computes a weighted norm of a parameter vector, $\mathscr{R}_d$ provides the norm of a residual of trajectories. Thus, for a suitably selected regularization coefficient β_d, this regularization can accelerate the assembly of the ROMs and improve their accuracy.



(a) *Optimization-based combined reduction with* $\beta_d = 0$.

(b) *Optimization-based combined reduction with* $\beta_d = 10^{-12}$.

(c) *Optimization-based combined reduction with* $\beta_d = 10^{-8}$.

(d) *Optimization-based combined reduction with* $\beta_d = 10^{-4}$.

(e) *Optimization-based combined reduction with* $\beta_d = 10^{-2}$.

(f) *Optimization-based combined reduction with* $\beta_d = 10^{-1}$.

Figure 5.10.: *Relative output error in the* $\ell_2 \otimes \ell_\infty$*-norm for varying reduced state and parameter dimensions, using different data-driven regularization coefficients for the optimization-based combined reduction.*

5.5. Nonlinear Benchmark

In this section an additional test[66] is conducted, which compares the implementation for the empirical-gramian-based and optimization-based methods for a nonlinear benchmark system to establish the evaluation criteria for the numerical experiments in Chapter 7.

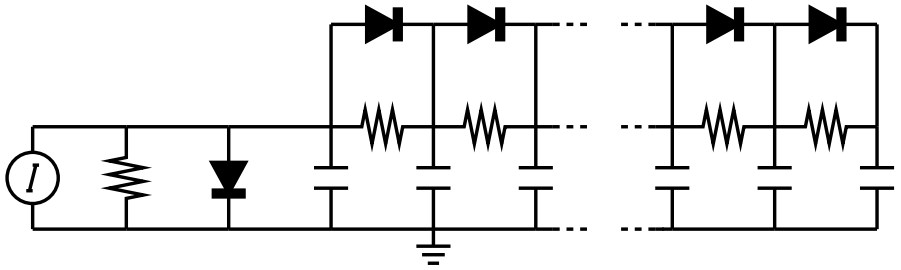


Figure 5.11.: *Circuit diagram for the nonlinear RC ladder model.*

5.5.1. Nonlinear RC Ladder

The considered benchmark[67] is a model of a circuit with linear resistor and capacitor (RC) elements as well as nonlinear diode components. This benchmark for nonlinear model reduction was established in [40] and is used among others in [44, 45, 43, 110, 113].
The model depicted in Figure 5.11 embodies a cascade of nonlinear resistors and capacitors, the latter are set to a unit value. A nonlinear resistor is composed of a linear resistor and a diode linked in parallel. This so-called **nonlinear RC ladder** is a parametric SISO system with N stages:

$$\dot{x}(t) = \begin{pmatrix} -g(x_1(t)) - g(x_1(t)-x_2(t)) \\ g(x_1(t)-x_2(t)) - g(x_2(t)-x_3(t)) \\ \vdots \\ g(x_{k-1}(t)-x_k(t)) - g(x_k(t)-x_{x+1}(t)) \\ \vdots \\ g(x_{N-1}(t)-x_N(t)) \end{pmatrix} + \begin{pmatrix} -2\theta_1 + \theta_2 \\ \theta_1 - 2\theta_2 + \theta_3 \\ \vdots \\ \theta_{k-1} - 2\theta_k + \theta_{k+1} \\ \vdots \\ \theta_{N-1} - \theta_N \end{pmatrix} \odot x(t) + \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{pmatrix} u(t),$$

$$y(t) = x_1(t),$$

with a nonlinear function $g : \mathbb{R} \to \mathbb{R}$,

$$g(x_i(t)) = s\, \mathrm{e}^{v x_i(t)} - 1,$$

and scalar constants $s \in \mathbb{R}$ and $v \in \mathbb{R}$ that characterize the nonlinear resistors (diodes) modelled by to the simple non-ideal diode model [182, Ch. 5.5]. The linear resistor values are individually parametrized, thus each stage gives rise to a parameter component and forms a $P = \dim(\theta) = \dim(x(t)) = N$ dimensional parameter-space.

[66] See `code/ch5/compare_rc.m` in the supplementary source code archive (Appendix B.1).
[67] The nonlinear RC ladder is a benchmark listed in the MORwiki [41, Nonlinear RC Ladder].

5.5.2. System Dimensions

For the subsequent tests a network with $N = 100$ nodes, and constants $s = 1$, $v = 40$ as well as a scalar-valued step input signal,

$$u(t) = \begin{cases} 1, & t \leq 0.3 \\ 0, & t > 0.3 \end{cases},$$

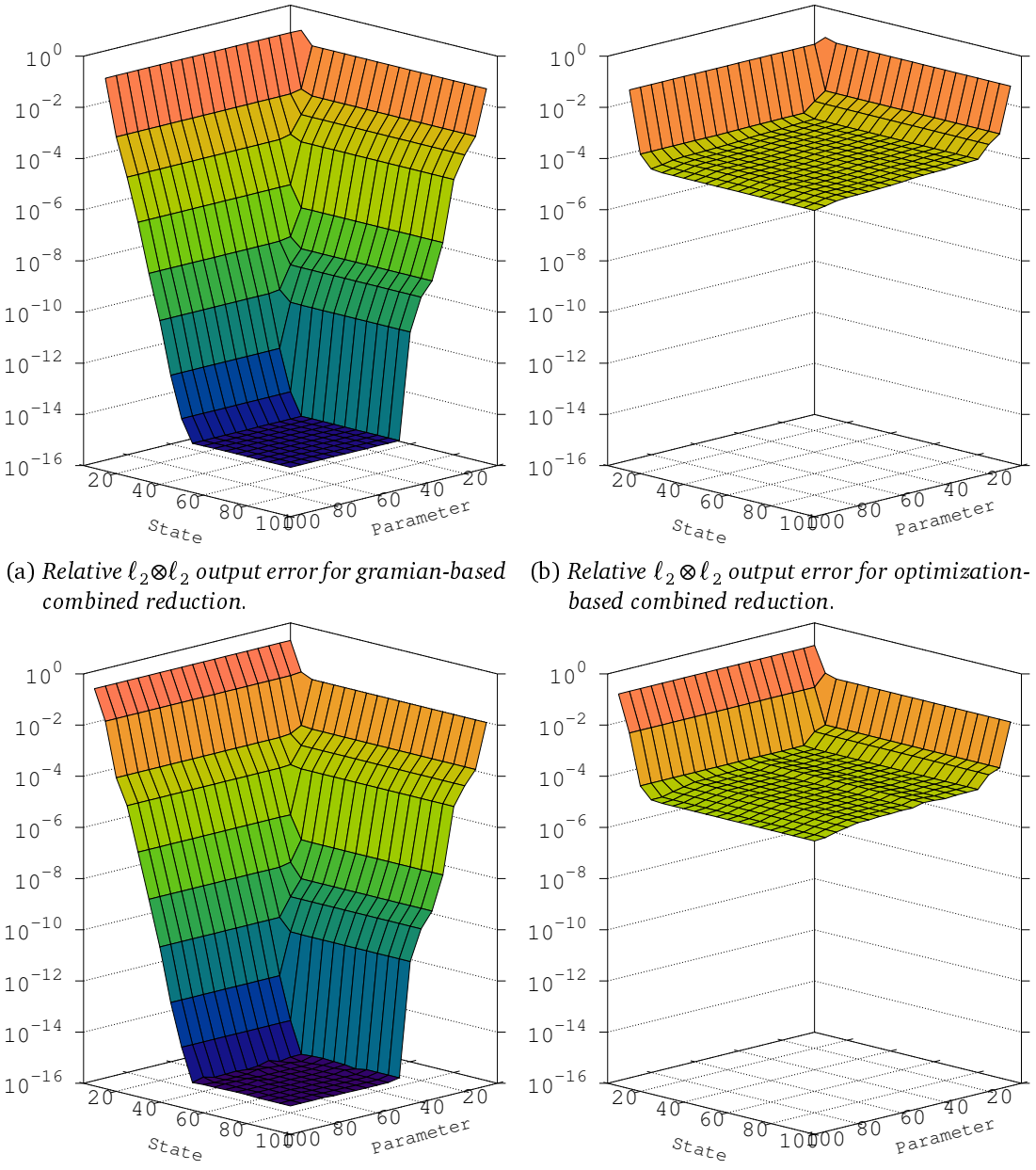
is selected and the parameter $\theta \in \mathbb{R}^{100}$ is drawn from a uniform random distribution $\theta_i \in \mathcal{U}_{[0,2]}$.

5.5.3. Combined Reduction

The combined state and parameter reduction is performed in case of the empirical-gramian-based approach using the empirical joint covariance matrix, yielding a empirical cross covariance matrix for the state-space reduction and a empirical cross-identifiability covariance matrix for the parameter-space reduction. For the greedy-optimization-based combined reduction approach a Tikhonov regularization coefficient $\beta_2 = \frac{1}{10}$ is used.

5.5.4. Combined Reduction Error

In Figure 5.12 the resulting ROMs are evaluated by comparing the relative output error in the joint $\ell_2 \otimes \ell_2$- and $\ell_2 \otimes \ell_\infty$-norms for varying reduced state- and parameter-space dimensions. The model reduction error of the gramian-based and optimization-based ROMs for the $\ell_2 \otimes \ell_2$-norm is depicted in Figure 5.12a and Figure 5.12c, for the $\ell_2 \otimes \ell_\infty$-norm in Figure 5.12b and Figure 5.12d.
Initially, the combined reduction error of the ROMs from both approaches and in both norms declines steeply, but for the optimization-based approach the error flattens out at about 10^{-5}, while the error continues to decrease for the gramian-based approach to about 10^{-15}. Notably, the error surfaces for the $\ell_2 \otimes \ell_2$- and $\ell_2 \otimes \ell_\infty$-norms exhibit a similar shape. In comparison the combined reduction error is dominated by the contribution of the parameter-space reduction for the optimization-based approach. For the gramian-based the state- and parameter-space reduction contribute equally to the combined reduction error.
Additionally, the combined reduction error is assessed for a cross-section of the previous error surfaces for the ROMs with a equally reduced state- and parameter-space dimension in Figure 5.13. This plot shows that a ROM obtained by the empirical joint gramian with a reduced state- and parameter-space dimension of $n = p = 50$ is sufficient to reach an error that near machine precision. For the optimization-based approach a ROM with state- and parameter-space dimension $n = p = 20$ yields the lowest error and can be assembled by 20 iterations of the algorithm. This initial decline in error over the ROMs up to order 20 runs equally for both methods in either norm as shown in Figure 5.13a and Figure 5.13b.



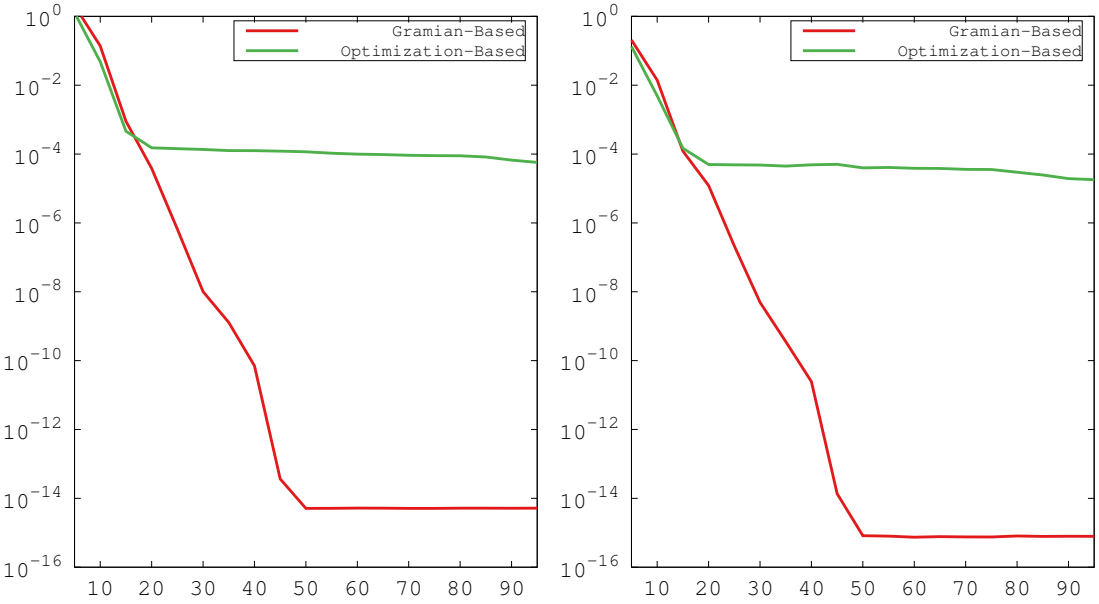
(a) *Relative $\ell_2 \otimes \ell_2$ output error for gramian-based combined reduction.*

(b) *Relative $\ell_2 \otimes \ell_2$ output error for optimization-based combined reduction.*

(c) *Relative $\ell_2 \otimes \ell_\infty$ output error for gramian-based combined reduction.*

(d) *Relative $\ell_2 \otimes \ell_\infty$ output error for optimization-based combined reduction.*

Figure 5.12.: *Numerical results for the combined state and parameter reduction of the nonlinear RC ladder.*



(a) *Relative $\ell_2 \otimes \ell_2$ output error of gramian- and optimization-based combined reduction.*

(b) *Relative $\ell_2 \otimes \ell_\infty$ output error of gramian- and optimization-based combined reduction.*

Figure 5.13.: *Relative output errors of the reduced order nonlinear RC ladder model for equally reduced state-parameter-space dimensions $n = p$.*

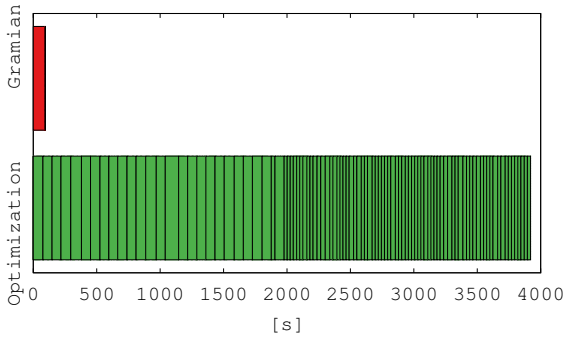


Figure 5.14.: *Comparison of timings for the nonlinear RC ladder model with highlighted per iteration durations in seconds.*

5.5.5. Combined Reduction Performance

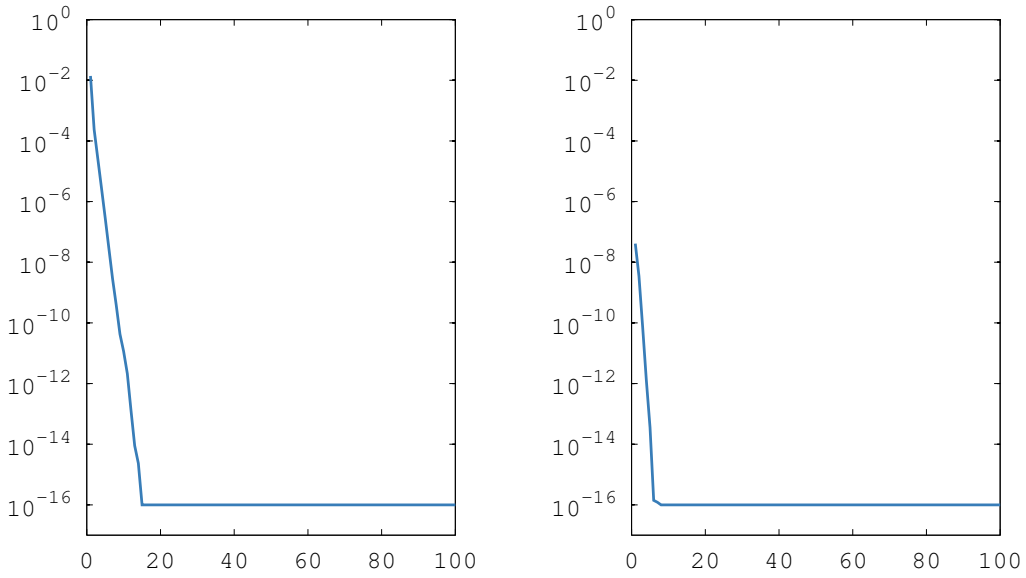
In Figure 5.14 the offline times of the gramian-based ROM and the optimization-based ROM are compared. The indicated duration of each iteration highlights the considerably slower performance of the optimization-based approach.

5.5.6. ROM Analysis

Following, the combined reduction error behavior of the gramian-based and optimization-based ROMs is analyzed. This enables an interpretation of the results as well as insight into the selection of the reduced basis components. For both methods the construction of the respective state- and parameter-space reducing projections is examined, and the ROM and FOM outputs are compared for specific parameters.

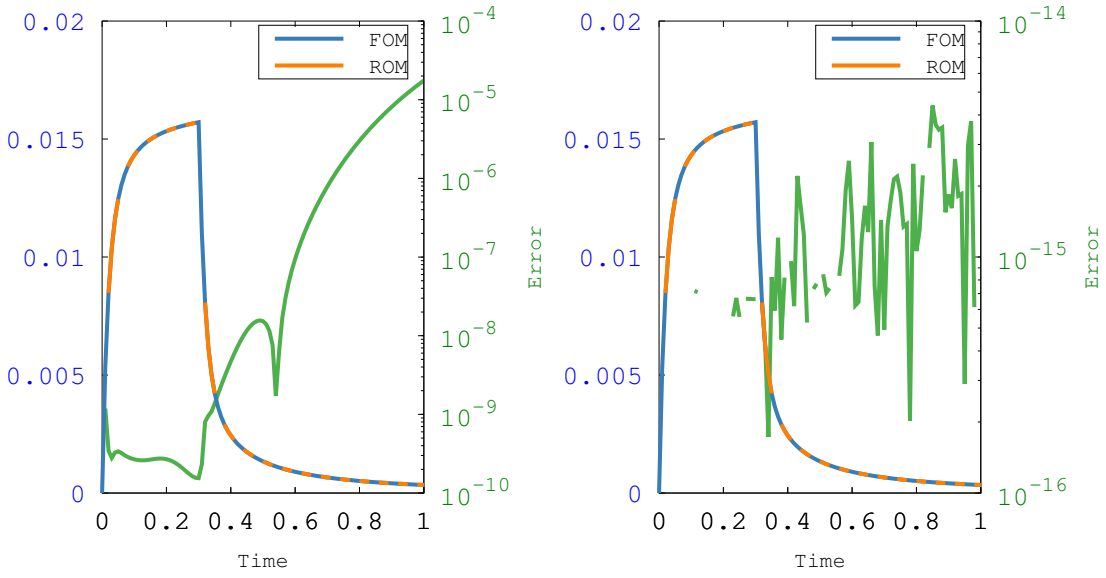
Gramian-Based ROMs

A byproduct of the cross-gramian-based state- and parameter-space reducing projections of the empirical cross gramian and empirical cross-identifiability gramian, are the associated singular values. Figure 5.15 shows the singular values of the utilized empirical gramians. Since the absolute values of the eigenvalues of the cross gramian are equivalent to the HSVs (Section 3.11) and thus quantify the input-output coherence for a linear symmetric system, the singular values of the empirical cross gramian W_X can at least act as an indicator (Section 3.4.4) for this nonlinear symmetric system. The steep decline of the singular values of the empirical cross gramian in Figure 5.15a suggests that all dominant dynamics are contained in the first twenty modes. Similarly, the singular values of the empirical cross-identifiability $W_{\tilde{I}}$ indicate the state-to-output influence of the parameters. An even steeper decline in the singular values of the empirical cross-identifiability gramian in Figure 5.15b implies that less than ten reduced parameters are sufficient to represent the parameter-space. Furthermore, the magnitude of the largest singular value of $W_{\tilde{I}}$ of the order 10^{-7} shows the lesser influence of the parameters in relation to the states for this variant of the benchmark. For a fixed parameter $\hat{\theta}$ sampled from $\mathscr{U}_{[0,2]}$, the ROM's output is compared to the FOM's output at two selected reduced orders in Figure 5.16 alongside with the error between this specific ROM and the FOM. In Figure 5.16a, the reduced order $n = p = 20$ is chosen, at which the error in the $\ell_2 \otimes \ell_2$-norm is computed to be 10^{-3}. Simulating trajectories for the fixed parameter $\hat{\theta}$, the ROMs output matches the FOM's output already closely, and the relative error, in the ℓ_2-norm, between the ROM and the FOM is 10^{-8}. This agrees with the $\ell_2 \otimes \ell_2$-norm, which is only approximated by sparsely sampling the parameter-space due to its high dimension. In case of the reduced order $n = p = 50$, for which the output is shown in Figure 5.16b, the error in the $\ell_2 \otimes \ell_2$-norm is at 10^{-15}. For the fixed parameter $\hat{\theta}$, the relative ℓ_2 error reaches 10^{-17}, which also agrees with the $\ell_2 \otimes \ell_2$-norm. Hence, the error of both ROMs conforms to the predicted combined reduction error.
In this example, the singular values of the utilized empirical gramians do not directly map to a tight error indicator, but practically, the magnitude of the singular values can serve as an indicator for the quality of the ROM. This is underscored by the simulation for $n = p = 20$, in which the ROM output matches the FOM output, at least superficially, very closely. Additionally, the singular values could be used to estimate the rank of the reducing projections, for example by an iterative computation of the SVD, with termination criteria based on the singular values, as in [38].



(a) *Empirical cross gramian.* (b) *Empirical cross-identifiability gramian.*

Figure 5.15.: *Singular values of the empirical gramians for nonlinear the RC ladder.*

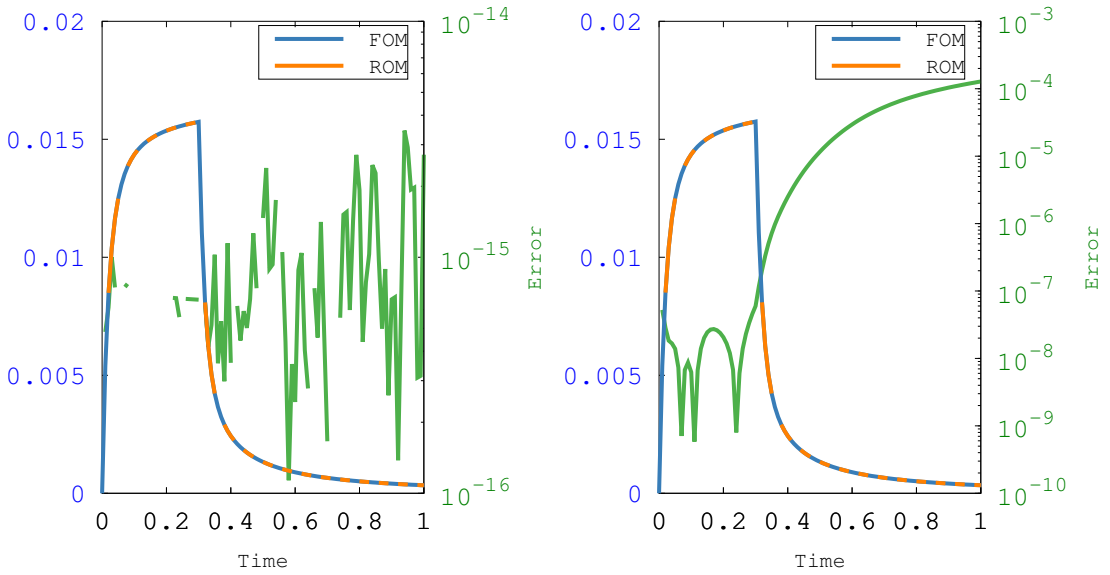


(a) *FOM and gramian-based ROM output with* $n = p = 20$. (b) *FOM and gramian-based ROM output with* $n = p = 50$.

Figure 5.16.: *Comparison of FOM and gramian-based combined reduction ROM output for a fixed randomly sampled parameter.*

Optimization-Based ROMs

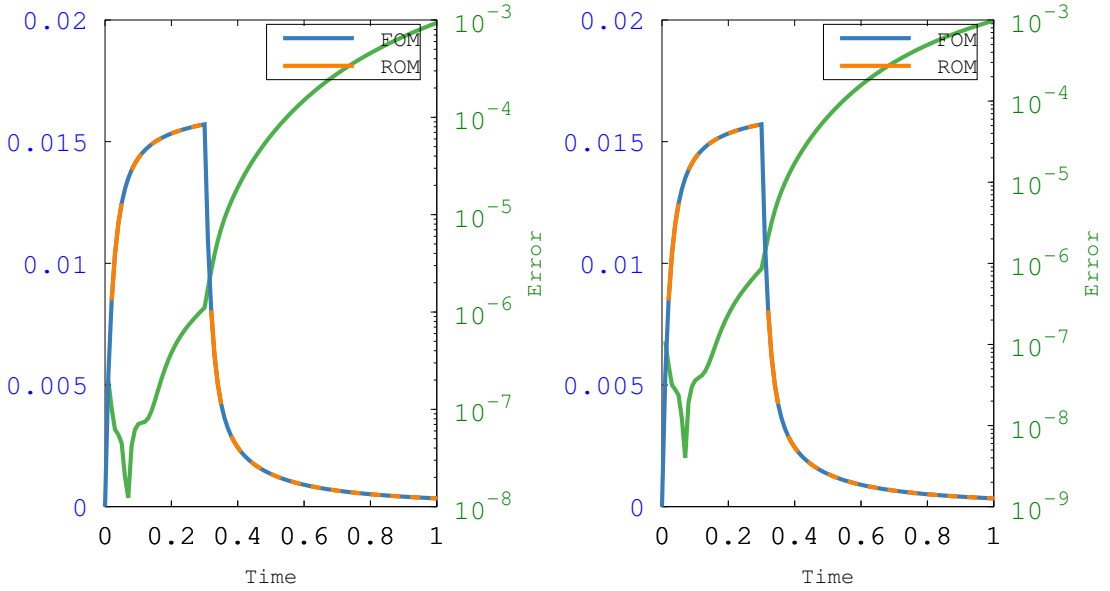
Opposed to the gramian-based combined reduction method, the ROMs constructed by the optimization-based approach do not reach a model reduction error near machine precision in the considered $\ell_2 \otimes \ell_\infty$- and $\ell_2 \otimes \ell_\infty$-norms. To examine the related reduced order accuracies, the outputs of different ROMs are compared to the output of the FOM in Figure 5.17. First, in Figure 5.17a, the FOM and ROM output as well as the relative error between FOM and ROM for a parameter θ_1 contained in the parameter-space basis, is shown. This error is of the order 10^{-15}, which confirms that a "trained" parameter is accurately approximated by the combined state and parameter reduced order model. In Figure 5.17b, the FOM and ROM output and the relative error between FOM and ROM for the parameter θ_1, but additively perturbed by a sample drawn from $\mathcal{N}^p_{0,\frac{1}{10}}$ is illustrated. The error, which is of order 10^{-7}, is much higher compared to error for the unperturbed parameter θ_1 even though the perturbation is relatively small.
Next, the optimization-based ROMs of reduced orders $n = p = 20$ and $n = p = 50$ are evaluated, using a fixed parameter $\hat{\theta}$ sampled from $\mathcal{U}_{[0,2]}$, in Figure 5.18. Figure 5.18a and Figure 5.18b depict the FOM and ROM output for $n = p = 20$ and $n = p = 50$ respectively as well as their associated relative error between the FOM and ROM outputs. The comparatively small improvement in the output error from $1.7 \cdot 10^{-6}$ for order $n = p = 20$ to $1.6 \cdot 10^{-6}$ for order $n = p = 50$ agrees with the previously sampled $\ell_2 \otimes \ell_\infty$-norm errors of $1.5 \cdot 10^{-4}$ and $1.2 \cdot 10^{-4}$. Yet, the ROM output matches the FOM output closely even for $n = p = 20$.
Since the maxima determined by the inner greedy sampling of the optimization-based approach are local and not unique, it can be assumed that due to the overall fit of the ROM output, no significant further maximization is achievable by the adaptive greedy strategy in its default setup. Improvements in accuracy of the utilized inner optimization method may yield lower errors, yet would come at the cost of longer assembly durations of the state and parameter projections. In comparison to the gramian-based combined reduction, the optimization-based combined reduction requires 41 times longer for this configuration, hence a prolonged duration does not seem viable.
For this experiment, a numerical approximation of the (Hessian) derivative information required by the inner optimization is used, which consumes the dominant fraction of the computational time. If analytical derivatives with respect to the parametrization are available, the computational complexity, and thus the assembly time, can be greatly reduced. Alternatively, an empirical gramian could also reduce the computational time of the optimization-based approach. In [12, 149], the relation between the observability gramian and Hessian for source inversion is explained. Such a setting would also allow to use an empirical observability gramian, and in case of a parameter inversion the empirical identifiability gramian may be utilized as a Hessian approximation. Yet, since in the scope of this work the gramian-based and optimization-based combined reduction methods are compared this relation is not exploited.



(a) *FOM and optimization-based ROM output with* $n = p = 50$.

(b) *FOM and optimization-based ROM output with* $n = p = 50$.

Figure 5.17.: *Comparison of FOM and optimization-based combined reduction ROM output for a trained and perturbed trained parameter.*



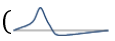
(a) *FOM and optimization-based ROM output with* $n = p = 20$.

(b) *FOM and optimization-based ROM output with* $n = p = 50$.

Figure 5.18.: *Comparison of FOM and optimization-based combined reduction ROM output for a fixed randomly sampled parameter.*

6. An Application in Neuroscience

Contents

An important effort in systems neuroscience is the investigation of individual and universal networks between different brain regions. Such mesoscale brain connectivity models [211] encode the propagation of information in causal **neuronal activity**.
Since neuronal activity cannot be measured directly, indirect measurements conveying information on this hidden process have to be recorded. In this work, models for two classes of functional neuroimaging techniques[68] are considered. First, measurements of the blood-oxygen level dependent response (BOLD), embodying neuronal oxygen intake through the blood, that indicates an increased neuronal activity. These hemodynamic measurements are conducted with functional magneto resonance imaging (fMRI) or functional near infrared spectroscopy (fNIRS) based on the magnetic susceptibility of the blood. Second, measurements of electromagnetic fields originating in the neuronal ion exchange during the information propagation by so called **action potentials**, which are spike impulses () emitted by neurons under a preprogrammed stimulus pattern. Regional synchronous neuronal activity (action potentials) can be recorded by Electroencephalography (EEG) measuring voltage changes or by Magnetoencephalography (MEG) measuring magnetic fields induced by these ionic currents.
Modeling the neuronal activity by a dynamic network with parametrized connectivity and the indirect measurements as function of the neuronal activity, yields an inverse problem constrained by a control system model. Inference on the connectivity parameters of the network model using a Bayesian statistics approach leads to the **dynamic causal modelling** framework introduced in [77], which is also summarized in [106] for both considered models.
To reconstruct the connectivity between (many) network nodes, the associated inverse problem may become computationally prohibitively expensive. Hence, commonly the measured data undergoes a dimension reduction, for example by a PCA (Section 2.5.1). Alternatively, model reduction techniques, such as combined reduction, can be utilized to accelerate the inversion[69].

[68] For an overview of neuroimaging techniques see [129]
[69] Previous efforts to introduce model reduction into neuroscience can be found, for example, in [135].

6.1. Neuronal Networks

The network representing the connectivity can be modelled by a graph-theoretic ansatz. For a linear control system (2.6), the system matrix A can be interpreted as the **adjacency matrix** [84, Ch. 8.1.2] representing a directed graph of the associated network by encoding the connection from the j-th region to the i-th region by the component A_{ij}.
To model the activation of the neuronal network, a function representing the firing rate of action potentials is introduced.

Definition 6.1 (Sigmoid Function)
*A **sigmoid function** $\varsigma : \mathbb{R} \to \mathbb{R}$ is an "S"-shaped function which is differentiable, has horizontal asymptotes for $x \to \pm\infty$ and a derivative of the form: $\varsigma'(x) = \varsigma(x)(1 - \varsigma(x)) > 0$.*

An example of a sigmoid function is a scaled and shifted hyperbolic tangent, visualized in Figure 6.1, parametrized by $\kappa \in (0, 1]$ controlling the incline of the slope:

$$\varsigma_\kappa(x) = \frac{1}{2}\tanh\left(\frac{\kappa}{2}x\right) + \frac{1}{2} \quad \left(= \frac{1}{1 + \mathrm{e}^{-\kappa x}}\right).$$

Figure 6.1.: *Plot of a sigmoid function $\varsigma_\kappa(x)$ for varying κ.*

6.1.1. Hyperbolic Network Model

The hyperbolic network model [178] is a nonlinear extension to the linear control system (2.6) utilizing a component-wise hyperbolic tangent - a sigmoid-like activation function.

Definition 6.2 (Hyperbolic Network Model)
*For a linear time-invariant control system $\Sigma(A, B, C)$ and a diagonal gain matrix K, $K_{i,i} \in (0, 1]$, the associated **hyperbolic network model** is given by:*

$$\begin{aligned} \dot{x}(t) &= A\tanh(Kx(t)) + Bu(t), \\ y(t) &= Cx(t). \end{aligned} \tag{6.1}$$

The exponential stability of (6.1) follows from the exponential stability of the system matrix A as a consequence of diagonal stability [94, Ch. 9].

6.2. Dynamic Causal Modelling

Dynamic causal modelling (DCM) [77] is a framework for inferring connectivity between brain regions from functional neuroimaging data. The construction of a dynamic causal model is based on modelling the transformation of neuronal activity to observable measurements of a single brain region as a SISO system. Several of these single regions are then coupled by a set of connectivity rules to a MIMO system describing the networked neuronal activity. Hence, each model consists of two components [213]: a dynamic submodel describing the change in neuronal activity over time, and a forward submodel transforming the neuronal activity to the measured output (Figure 6.2).

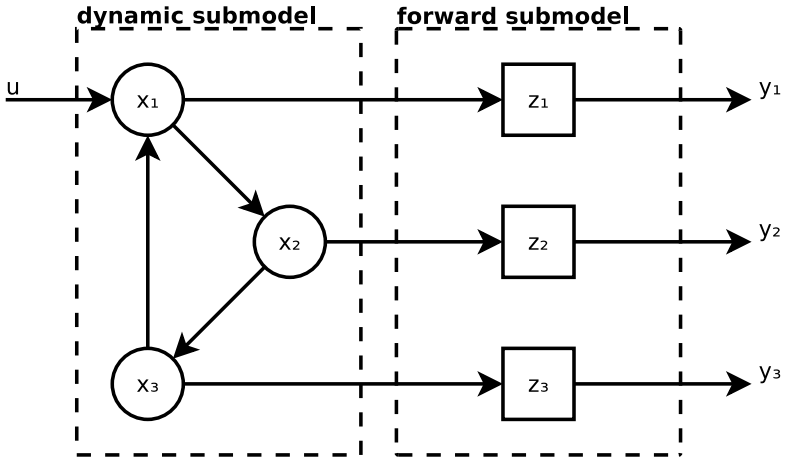


Figure 6.2.: *Schematic illustration of dynamic and forward submodels for a three node (region) network with input u, network nodes x_i, per-region states z_i, and measurable outputs y_i.*

These dynamic causal models do not regard location of the considered brain regions, thus they are considered spatially zero-dimensional and the dynamics are represented by ODEs with respect to time.
Foremost, the parametrizations represent the connectivity between the analyzed brain regions, but also include parameters allowing variability in the behavior among the individual network nodes. The latter group of "physiological" parameters is kept constant in the scope of this work, and thus is excluded from the parameter identification and reduction procedures. The former set of connectivity parameters in the considered models enclose two subsets: effective connectivity parameters and lateral connectivity parameters. Effective connectivity characterizes the permanent coupling between regions while the lateral connectivity describes input or activity dependent interconnection. This work considers only the parameters related to effective connectivity.
Models for two classes of neuroimaging techniques are investigated, which due to their means of recording have different dynamic and forward submodels. The first class includes fMRI and fNIRS methods, the second class encompasses EEG and MEG techniques. Both classes of models are originally nonlinear and a linearization is considered.

6.3. fMRI & fNIRS Dynamic Causal Model

Based on the observation that more active regions of the brain consume more oxygen, the fMRI and fNIRS neuroimaging techniques measure deoxygenated hemoglobin in relation to the blood volume of designated recorded volume elements (voxels) yielding the BOLD signal. A single voxel or patch of voxels[70] can describe a brain region, for which the hemodynamic forward submodel converts the local neuronal activity to the associated BOLD output. The networked neuronal activity is modeled by the multi-region dynamic submodel that comprises the connectivity between the various regions. Together, the joint model consists of a MIMO system (dynamic submodel) where each node has a SISO system (forward submodel) attached to generate the measurable output signal [77].

6.3.1. Dynamic Submodel

Under the assumption that the change in neuronal activity over time follows some nonlinear dynamics, the general dynamical system (2.5) is approximated in [77] using the Taylor series. Following, this linear approximation is derived around the zero steady-state with zero input:

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t), \theta) \\ &\approx f(0,0,\theta) + \frac{\partial f}{\partial x} x(t) + \frac{\partial f}{\partial u} u(t) \\ \Rightarrow \dot{\tilde{x}}(t) &= Ax(t) + Bu(t). \end{aligned} \tag{6.2}$$

The linear approximation (6.2) corresponds to the vector field of the linear control system (2.6) and represents the **effective connectivity** between regions by the system matrix A and the action of the external input by the input matrix B. Higher order components of the Taylor series would introduce **lateral connectivity**, for example the bilinear approximation in [77] would enable the influence of external input on the connectivity. For a model of k regions, the parametrization[71] of the model is given by the components of the matrix $A \in \mathbb{R}^{k\times k}$; this means a vectorization (see Appendix A.2) of A yields the parameter vector θ which is of dimension k^2:

$$\begin{aligned} \theta &= \mathrm{vec}(A) \in \mathbb{R}^{k^2} \\ \Rightarrow A(\theta) &= \mathrm{vec}^{-1}(\theta) \\ \rightarrow \dot{x}(t) &= A(\theta)x(t) + Bu(t). \end{aligned} \tag{6.3}$$

The off-diagonal parameter components of $A(\theta)$ model the connectivity between two regions (nodes) and the diagonal components represent the self-regulatory decay of local (per-region) activity ensuring the (asymptotic) stability of the dynamics, which is an obvious assumption for a regular operating neuronal network.
The local neuronal activity x_i of the fMRI & fNIRS linear dynamic submodel acts as an input to the subsequent forward model transforming these hidden states to a measurable BOLD response.

[70] This includes a linear combination of voxels obtained for example through a PCA.
[71] In [77] also the components of $B \in \mathbb{R}^{k\times M}$ are parametrized, which is also omitted in this setting.

6.3.2. Forward Submodel

The hemodynamic forward submodel transforms the non-physiological neuronal activity to the measured BOLD response. A description of the model is given in [76] and [77] in conjunction with the dynamic submodel and is briefly summarized here.
Based on the balloon model, which correlates blood flow with the BOLD response, the extended balloon model [76, Sec. 2] relates a neuronal activity signal to the BOLD signal. A change in the vasodilatory[72] signal $\dot{s}_i(t)$ is induced by the neuronal activity in the i-th associated network node $x_i(t)$ from the dynamic submodel (6.2), which acts as input to this SISO system. The parameter κ characterizes the decay of the signal s_i, while γ determines the self-regulating feedback from the inflow f_i. Change in inflow $\dot{f}_i(t)$ linearly relates to the signal $s_i(t)$ and the normalized change in blood volume $\dot{v}_i(t)$ is given by the difference between inflow $f_i(t)$ and outflow $v_i(t)^{\frac{1}{\alpha}}$, which is regulated by the parameter α, the so-called Grubb's exponent. Similarly, the normalized change in deoxyhemoglobin content $\dot{q}_i(t)$ is given by the difference of the normalized inflow-related oxygen extraction fraction $1-(1-\rho)^{\frac{1}{f_i(t)}}$ (ρ represents the *resting oxygen extraction fraction*) and the released outflow-dependent fraction $v_i(t)^{\frac{1}{\alpha}}\frac{q_i(t)}{v_i(t)}$. The change in volume and content is normalized by the *mean hemodynamic transit time* τ of the oxygen from the blood vessel to the neuron. A combination[73] of volume $v_i(t)$ and content $q_i(t)$ weighted by $k_1, k_2 \in \mathbb{R}$ and normalized by the *resting blood volume fraction* $V_0 \in \mathbb{R}$ generates the output $y_i(t)$ that corresponds to the BOLD signal. In condensed form, the SISO forward system (see also Figure 6.3) is represented in per-region hemodynamic state $z_i(t)$ by:

$$\dot{z}_i(t) := \begin{pmatrix} \dot{s}_i(t) \\ \dot{f}_i(t) \\ \dot{v}_i(t) \\ \dot{q}_i(t) \end{pmatrix} = \underbrace{\begin{pmatrix} x_i(t) - \kappa s_i(t) - \gamma(f_i(t)-1) \\ s_i(t) \\ \frac{1}{\tau}(f_i(t) - v_i(t)^{\frac{1}{\alpha}}) \\ \frac{1}{\tau}(\frac{1}{\rho}f_i(t)(1-(1-\rho)^{\frac{1}{f_i(t)}}) - v_i(t)^{\frac{1}{\alpha}}\frac{q_i(t)}{v_i(t)}) \end{pmatrix}}_{=:h_i},$$

$$y_i(t) = \underbrace{V_0(k_1(1-q_i(t)) + k_2(1-v_i(t)))}_{=:g_i}.$$

The equilibrium and considered initial state is given by the zero input response:

$$z_{i,0} := \begin{pmatrix} s_{i,0} \\ f_{i,0} \\ v_{i,0} \\ q_{i,0} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}.$$

The stability of the hemodynamic forward submodel can be examined locally through the spectrum of the Jacobian [121, 133] resulting from linearization of h_i around $z_{i,0}$, which is conducted next.

[72] The term vasodilatory means widening of blood vessels.
[73] A third component in [77] embodying the ratio of volume and content is omitted here as in [133].

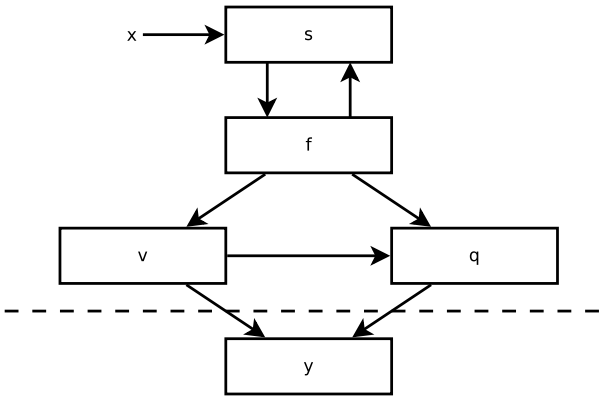


Figure 6.3.: *Schematic illustration of the hemodynamic forward submodel.*

Further following [133], a linearization at the equilibrium steady-state using:

$$A_h = \left.\frac{\partial h_i}{\partial z_i}\right|_{z_{i,0}}, \quad B_h = \left.\frac{\partial h_i}{\partial x_i}\right|_{z_{i,0}}, \quad C = \left.\frac{\partial g_i}{\partial z_i}\right|_{z_{i,0}},$$

yields a linear forward submodel:

$$\dot{z}_i(t) = \underbrace{\begin{pmatrix} -\kappa & -\gamma & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & \frac{1}{\tau} & -\frac{1}{\tau\alpha} & 0 \\ 0 & \frac{(1-(1-\rho)(1-\ln(1-\rho)))}{\tau\rho} & -\frac{1-\alpha}{\tau\alpha} & -\frac{1}{\tau} \end{pmatrix}}_{=:A_h} z_i(t) + \underbrace{\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}}_{=:B_h} x_i(t),$$

$$y_i(t) = \underbrace{V_0 \begin{pmatrix} 0 & 0 & -k_1 & k_2 \end{pmatrix}}_{=:C_h} z_i(t).$$

For this linearized hemodynamic model, the equilibrium and initial state becomes: $z_{i,0} = \begin{pmatrix} 0 & 0 & 0 & 0 \end{pmatrix}^\intercal$. Using an eigenvalue analysis of the system matrix A_h, which corresponds to the Jacobian of the nonlinear system at the steady-state, the stability of the linearized model can be assessed, as in [133], by the negativity of the real parts of the eigenvalues $\lambda_{1\ldots4}(A_h)$:

$$\lambda_1(A_h) = -\frac{1}{\tau} \overset{\tau>0}{<} 0,$$

$$\lambda_2(A_h) = -\frac{1}{\tau\alpha} \overset{\tau\alpha>0}{<} 0,$$

$$\lambda_{3,4}(A_h) = -\frac{1}{2\kappa} \pm \sqrt{\frac{1}{4\kappa^2} - \frac{1}{\gamma}} \overset{\gamma>0}{<} 0.$$

This confirms the global asymptotic stability for the linearized and the local asymptotic stability for the nonlinear hemodynamic forward submodel near the zero steady-state.

6.3.3. Joint Model

The joint model in state-space system form for the fMRI & fNIRS dynamic causal model has a $5k$-dimensional state-space and is given in the nonlinear setting by:

$$\begin{pmatrix} \dot{x}(t) \\ \dot{z}_1(t) \\ \vdots \\ \dot{z}_k(t) \end{pmatrix} = \begin{pmatrix} A(\theta)x(t) \\ h_1(z_1(t), x_1(t)) \\ \vdots \\ h_k(z_k(t), x_k(t)) \end{pmatrix} + \begin{pmatrix} B \\ 0 \\ \vdots \\ 0 \end{pmatrix} u(t),$$
$$y(t) = \begin{pmatrix} g_1(z_1(t)) & \dots & g_k(z_k(t)) \end{pmatrix}^\intercal.$$

For a more efficient numerical evaluation, the (forward submodel) system components are re-ordered by grouping the individual regions' components s_i, f_i, v_i and q_i together. This leads to the following representation of the nonlinear joint model:

$$\begin{pmatrix} \dot{x}(t) \\ \dot{s}(t) \\ \dot{f}(t) \\ \dot{v}(t) \\ \dot{q}(t) \end{pmatrix} = \begin{pmatrix} A(\theta)x(t) \\ x(t) - \kappa s(t) - \gamma(f(t) - \vec{1}_k) \\ s(t) \\ \frac{1}{\tau}(f(t) - v(t)^{\frac{1}{\alpha}}) \\ \frac{1}{\tau}(\frac{1}{\rho} f(t)(\vec{1}_k - ((1-\rho)\,\vec{1}_k)^{\frac{1}{f(t)}}) - v(t)^{\frac{1}{\alpha}-1} \odot q(t)) \end{pmatrix} + \begin{pmatrix} B \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} u(t), \tag{6.4}$$
$$y(t) = V_0(k_1(\vec{1}_k - q(t)) + k_2(\vec{1}_k - v(t))),$$

with $\vec{1}$ representing a vector of ones, the element-wise Hadamard product $\odot$, and all exponentiations understood as element-wise operations[74].
In the linear(ized) case, also sorted by components, the joint forward submodel has the form:

$$\begin{pmatrix} \dot{x}(t) \\ \dot{s}(t) \\ \dot{f}(t) \\ \dot{v}(t) \\ \dot{q}(t) \end{pmatrix} = \begin{pmatrix} A(\theta) & 0 & 0 & 0 & 0 \\ \mathbb{1}_k & -\kappa\,\mathbb{1}_k & -\gamma\,\mathbb{1}_k & 0 & 0 \\ 0 & \mathbb{1}_k & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\tau}\,\mathbb{1}_k & -\frac{1}{\tau\alpha}\,\mathbb{1}_k & 0 \\ 0 & 0 & \frac{(1-(1-\rho)(1-\ln(1-\rho)))}{\tau\rho}\,\mathbb{1}_k & -\frac{1-\alpha}{\tau\alpha}\,\mathbb{1}_k & -\frac{1}{\tau}\,\mathbb{1}_k \end{pmatrix} \begin{pmatrix} x(t) \\ s(t) \\ f(t) \\ v(t) \\ q(t) \end{pmatrix} + \begin{pmatrix} B \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} u(t),$$

$$y(t) = \begin{pmatrix} 0 & 0 & 0 & -V_0 k_1\,\mathbb{1} & V_0 k_2\,\mathbb{1} \end{pmatrix} \begin{pmatrix} x(t) \\ s(t) \\ f(t) \\ v(t) \\ q(t) \end{pmatrix}. \tag{6.5}$$

Lastly, the dynamic submodel and forward submodel evolve on similar scales if the dynamic submodel's system matrix $A(\theta)$ is normalized [77], hence a joint integration of the submodels is feasible.

[74] Similarly, $((1-\rho)\,\vec{1}_k)^{\frac{1}{f(t)}}$ is given by the element-wise reciprocal entries of $f(t)$ as exponents applied element-wise to its base.

6.3.4. Parametrization

The parameter-space for both, the nonlinear and the linearized model, is given by a k^2-dimensional space of (effective) connectivity strength coefficients, which map to the components of the system matrix $A(\theta)$ (6.3). Notably, the dimension of the parameter-space exceeds the state-space dimension for $k > 5$. Since the neuronal activity and thus the dynamic submodel is expected to evolve in a stable regime, restrictions have to be imposed on the parameters θ such that the system matrix $A(\theta)$ induces a stable system (2.4). In [77] the elements of $A(\theta)$ are normalized by a factor $\sigma \in \mathbb{R}$, such that the diagonal elements, which are assumed homogeneous, become $a_{ii} = -1$:

$$A = \sigma \begin{pmatrix} -1 & a_{12} & \dots & a_{1k} \\ a_{21} & -1 & & \vdots \\ \vdots & & \ddots & a_{(k-1)k} \\ a_{k1} & \dots & a_{k(k-1)} & -1 \end{pmatrix},$$

and the off-diagonal components are selected relative to the diagonal. Here, the system matrix diagonal components are shifted by addition of a scaled unit matrix to facilitate stability justified by Gershgorin's circle theorem (see for example: [88, Thm. 7.2.1]):

$$A = -k\,\mathbb{1}_k + \begin{pmatrix} a_{11} & \dots & a_{1k} \\ \vdots & \ddots & \vdots \\ a_{k1} & \dots & a_{kk} \end{pmatrix}, \quad -k + a_{ii} \overset{!}{>} \max\Big(\sum_{\substack{i=1 \\ i\neq j}}^{k} |a_{ij}|, \sum_{\substack{j=1 \\ i\neq j}}^{k} |a_{ij}|\Big).$$

The remaining parameters, such as the components of the input matrix B, the dynamic hemodynamic parameters $\{\kappa, \gamma, \alpha, \tau, \rho\}$ and output hemodynamic parameters $\{V_0, k_1, k_2\}$ are fixed to constant values, since the focus of this work rests on the connectivity parameters. While the components of the input matrix are preset depending on the experiments, the constant hemodynamic parameters are listed in Table 6.4, which are taken from [77, Tab. 1, Eq. (4)].

Signal Decay Rate	κ	$[s^{-1}]$	0.065
Flow-Dependent Elimination Rate	γ	$[s^{-1}]$	0.41
Hemodynamic Transit Time	τ	[s]	0.98
Resting Oxygen Extraction Fraction	ρ		0.34
Grubb's Exponent	α		0.32
Resting Blood Volume Fraction	V_0		0.02
Volume Weight	k_1		2.38
Concentration Weight	k_2		0.48

Table 6.4.: *fMRI & fNIRS model fixed hemodynamic parameters.*

6.4. EEG & MEG Dynamic Causal Model

The EEG and MEG functional neuroimaging techniques measure electric potentials or magnetic fields induced by the former originating from neuronal ionic currents. Each sensor records the average change in voltage over time associated to a local region of the neural mass. Thus, these **neural mass model**s represent the mean state of neuronal activity [48]. Contrary to the fMRI & fNIRS model, the EEG & MEG dynamic submodel is physiologically motivated and the forward model accounts only for a sensory offset.

6.4.1. Dynamic Submodel

The dynamic submodel is a hierarchical model. Each individual region consists of three hardwired sub-regions that make up a SISO system. Several such regions are then connected by a set of connectivity rules yielding the MIMO system [47, 48].
For a causal, exponentially decaying synaptic impulse response,

$$g(t) = \begin{cases} H\frac{t}{\tau}\mathrm{e}^{-\frac{t}{\tau}}, & t \geq 0 \\ 0, & t < 0 \end{cases}, \tag{6.6}$$

with a maximum amplitude[75] H and a lumped parameter rate τ. The convolution of the (pre-synaptic) input $u(t)$ with the impulse response kernel $g(t)$ results in the average post-synaptic membrane potential $v(t)$:

$$\begin{aligned} & v(t) = (g * u)(t) \\ \Rightarrow & \ddot{v}(t) = \frac{H}{\tau}u(t) - \frac{2}{\tau}\dot{v}(t) - \frac{1}{\tau^2}v(t) \\ \Rightarrow & \begin{cases} \dot{x}(t) = v(t) \\ \dot{v}(t) = \frac{H}{\tau}u(t) - \frac{2}{\tau}x(t) - \frac{1}{\tau^2}v(t) \end{cases} \end{aligned} \tag{6.7}$$

This so-called Jansen model is a second-order ODE reduced to a system of first-order ODEs. Following [169], a brain region (in layer 4 of the mammalian cortex) is modelled by three different layers of neuronal sub-populations, which in turn are composed of altogether five sub-regions (6.7). The first, supragranular layer, which contains an inhibitory (interneuron) sub-population encloses an excitatory and an inhibitory sub-region. Second, an excitatory (spiny-cell) sub-population in the granular layer that also receives the external input consists of a single excitatory sub-region. Lastly, the third, infragranular (pyramidal cell) layer, is an excitatory (pyramidal-cell) sub-population modelled by an excitatory and an inhibitory sub-region.
These three layers are interconnected by intrinsic (intra-region) connections illustrated in Figure 6.5. The supra-granular inhibitory layer has a two-way connection to the infra-granular excitatory layer γ_1, γ_5 as well as a self connection γ_2. The granular excitatory layer also has a two way connection to the infra-granular excitatory layer γ_3, γ_4.

[75] For the impulse response (6.6) H is rather an initial quantity, yet for the final model it will represent the maximum amplitude.

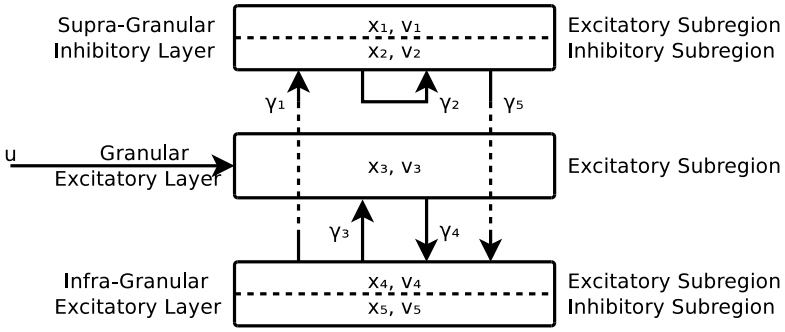


Figure 6.5.: *Single region neural mass model schematic.*

Input from one subregion to another[76] is then transformed from a membrane potential to a firing rate of action potentials using a sigmoid-like function $\bar{\varsigma}_\kappa(v(t))$ related to Definition 6.1. Thus, the dynamical system is nonlinear due to the coupling of the sub-regions by the nonlinear sigmoid-like function.
The joint single-region model with all five subregions, divided into three layers, has a ten-dimensional state-space:

$$
\begin{aligned}
\dot{v}_1(t) &= x_1(t),\\
\dot{x}_1(t) &= \frac{H_e}{\tau_e}\gamma_1\bar{\varsigma}_\kappa(v_4(t)-v_5(t)) - \frac{2}{\tau_e}x_1(t) - \frac{1}{\tau_e^2}v_1(t),\\
\dot{v}_2(t) &= x_2(t),\\
\dot{x}_2(t) &= \frac{H_i}{\tau_i}\gamma_2\bar{\varsigma}_\kappa(v_1(t)-v_2(t)) - \frac{2}{\tau_i}x_2(t) - \frac{1}{\tau_i^2}v_2(t),\\
\hline
\dot{v}_3(t) &= x_3(t),\\
\dot{x}_3(t) &= \frac{H_e}{\tau_e}\gamma_3\bar{\varsigma}_\kappa(v_4(t)-v_5(t)) - \frac{2}{\tau_e}x_3(t) - \frac{1}{\tau_e^2}v_3(t) + \frac{H_e}{\tau_e}u(t),\\
\hline
\dot{v}_4(t) &= x_4(t),\\
\dot{x}_4(t) &= \frac{H_e}{\tau_e}\gamma_4\bar{\varsigma}_\kappa(v_3(t)) - \frac{2}{\tau_e}x_4(t) - \frac{1}{\tau_e^2}v_4(t),\\
\dot{v}_5(t) &= x_5(t),\\
\dot{x}_5(t) &= \frac{H_i}{\tau_i}\gamma_5\bar{\varsigma}_\kappa(v_1(t)-v_2(t)) - \frac{2}{\tau_i}x_5(t) - \frac{1}{\tau_i^2}v_5(t),
\end{aligned}
\tag{6.8}
$$

with a zero initial state $v_{i,0} = x_{i,0} = 0$, of which a solution $v(t)$ represents the average membrane potentials over time. The index e and i of the physiological parameters H, τ denote their affiliation to an excitatory or inhibitory sub-region, respectively.
The stability of this system can be assessed near a steady-state by a linearization of the sigmoid-like function, see (6.10), and an eigenvalue analysis of the resulting system matrix of the linearized system as in [169].

[76] Or to itself.

Sorting by first-order and second-order terms and separating (6.8) into a linear and a non-linear part then yields the following representation of the system:

$$\begin{pmatrix}\dot{v}(t)\\ \dot{x}(t)\end{pmatrix} = \begin{pmatrix}0 & \mathbb{1}_5\\ -T^2 & -2T\end{pmatrix}\begin{pmatrix}v(t)\\ x(t)\end{pmatrix} + \begin{pmatrix}0\\ A_v\end{pmatrix}\bar{\varsigma}_\kappa(A_\varsigma v(t)) + \frac{H_e}{\tau_e}\delta_{8,1}^{10\times 1}u(t), \tag{6.9}$$

$$T = \begin{pmatrix}\frac{1}{\tau_e} & 0 & 0 & 0 & 0\\ 0 & \frac{1}{\tau_i} & 0 & 0 & 0\\ 0 & 0 & \frac{1}{\tau_e} & 0 & 0\\ 0 & 0 & 0 & \frac{1}{\tau_e} & 0\\ 0 & 0 & 0 & 0 & \frac{1}{\tau_i}\end{pmatrix},\ A_v = \begin{pmatrix}\frac{H_e}{\tau_e}\gamma_1 & 0 & 0 & 0 & 0\\ 0 & \frac{H_i}{\tau_i}\gamma_2 & 0 & 0 & 0\\ 0 & 0 & \frac{H_e}{\tau_e}\gamma_3 & 0 & 0\\ 0 & 0 & 0 & \frac{H_e}{\tau_e}\gamma_4 & 0\\ 0 & 0 & 0 & 0 & \frac{H_i}{\tau_i}\gamma_5\end{pmatrix},\ A_\varsigma = \begin{pmatrix}0 & 0 & 0 & 1 & -1\\ 1 & -1 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & -1\\ 0 & 0 & 1 & 0 & 0\\ 1 & -1 & 0 & 0 & 0\end{pmatrix},$$

modelling a single region in which the sigmoid-like function is assumed to act element-wise on its vector-valued argument. The multi-region model has essentially the same structure as (6.9), and without coupling each scalar of the single region model becomes a block containing a diagonal matrix. Hence, a model for k regions has a $10k$ dimensional state-space and $v(t), x(t) \in \mathbb{R}^{5k}$. For the coupling of multiple regions, the connectivity scheme from [47] is employed, which is illustrated in Figure 6.6.

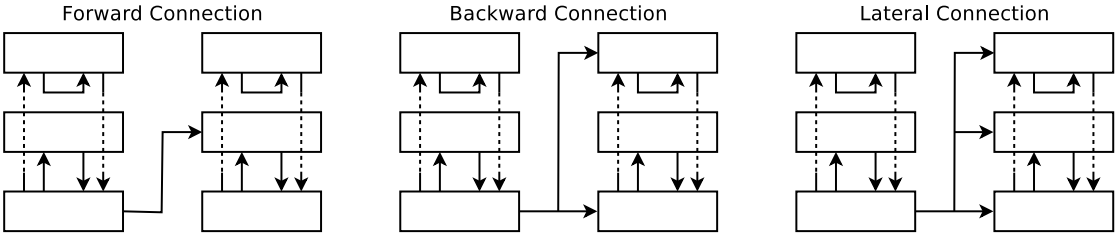


Figure 6.6.: *Forward, backward and lateral connectivity rules for the neural mass model.*

This coupling scheme allows three types of connections, which all originate in the source region's infra-granular (excitatory) sub-region. A **forward** connection terminates in the target region's granular layer, a **backward** connection couples to supragranular and infragranular layers of the target region, and the **lateral** connection links to all target region's subregions. The networked multi-region model is then given by:

$$\begin{pmatrix}\dot{v}(t)\\ \dot{x}(t)\end{pmatrix} = \begin{pmatrix}0 & \mathbb{1}_{5k}\\ -T^2\otimes\mathbb{1}_k & -2T\otimes\mathbb{1}_k\end{pmatrix}\begin{pmatrix}v(t)\\ x(t)\end{pmatrix} + \begin{pmatrix}0\\ A_v\end{pmatrix}\bar{\varsigma}_\kappa(A_\varsigma v(t)) + \frac{H_e}{\tau_e}(\delta_{8,1}^{10\times 1}\otimes B)u(t),$$

$$A_v = \begin{pmatrix}\frac{H_e}{\tau_e}(A_B + A_L + \gamma_1\mathbb{1}_k) & 0 & 0 & 0 & 0\\ 0 & \frac{H_i}{\tau_i}\gamma_2\mathbb{1}_k & 0 & 0 & 0\\ 0 & 0 & \frac{H_e}{\tau_e}(A_F + A_L + \gamma_3\mathbb{1}_k) & 0 & 0\\ 0 & 0 & 0 & \frac{H_e}{\tau_e}\gamma_4\mathbb{1}_k & \frac{H_e}{\tau_e}(A_B + A_L)\\ 0 & 0 & 0 & 0 & \frac{H_i}{\tau_i}\gamma_5\mathbb{1}_k\end{pmatrix},$$

with $A_F(\theta_F) \in \mathbb{R}^{k\times k}$ encoding forward, $A_B(\theta_B) \in \mathbb{R}^{k\times k}$ backward, $A_L(\theta_L) \in \mathbb{R}^{k\times k}$ lateral connections, which, similar to the fMRI & fNIRS dynamic submodel (6.3), are parametrized by an inverse vectorization $A_* = \mathrm{vec}^{-1}(\theta_*)$, and $B \in \mathbb{R}^{k\times M}$ modelling input dispersion.

The sigmoid-like activation function selected in [47] reads in the hyperbolic tangent representation:

$$\bar{\varsigma}_\kappa(x) = \frac{1}{1+\mathrm{e}^{-\kappa x}} - \frac{1}{2} = \varsigma_\kappa(x) - \frac{1}{2} = \frac{1}{2}\tanh\Big(\frac{\kappa}{2}x\Big).$$

Following [169], a linearization of the previous dynamic submodel is given by a linear approximation of the sigmoid-like function around the steady-state ($x = v = 0$):

$$\bar{\varsigma}_\kappa(x) \approx \bar{\varsigma}_\kappa(0) + \frac{\mathrm{d}\bar{\varsigma}_\kappa(0)}{\mathrm{d}x}x = 0 + (\frac{\kappa}{4}(1-\tanh^2(\kappa x))\big|_{x=0})x = \frac{\kappa}{4}x. \tag{6.10}$$

The slope of the sigmoid-like activation function defines the sigmoid gain [169] characterizing the linear approximation. Then, the non-trivial part of the linearized dynamical submodel has the form:

$$\begin{pmatrix}\dot{v}(t)\\ \dot{x}(t)\end{pmatrix} = \begin{pmatrix}0 & \mathbb{1}_{5k}\\ -T^2\otimes\mathbb{1}_k & -2T\otimes\mathbb{1}_k\end{pmatrix}\begin{pmatrix}v(t)\\ x(t)\end{pmatrix} + \begin{pmatrix}0\\ A_v A_\varsigma\end{pmatrix}\frac{\kappa}{4}v(t) + \frac{H_e}{\tau_e}(\delta_{8,1}^{10\times 1}\otimes B)u(t).$$

This dynamic submodel is a second-order control system in first-order form and considered around the zero steady-state in the nonlinear and linearized case.

6.4.2. Forward Submodel

The forward submodel connects the neural mass dynamic submodel with the EEG & MEG observations and is of linear nature. In case of a single region, the measurable output is generated by the difference in membrane potentials inside the granular excitatory sub-region:

$$y(t) = v_4(t) - v_5(t).$$

This is the same signal that is forwarded to connected regions in a multi-region model. The forward submodel for a multi-region model is a linear combination of the output membrane potentials represented by the contribution matrix $L \in \mathbb{R}^{k\times k}$:

$$\begin{aligned} y(t) &= L(v_4(t) - v_5(t))\\ &= \begin{pmatrix}0 & 0 & 0 & L & -L\end{pmatrix}v(t)\\ &= \begin{pmatrix}0 & 0 & 0 & L & -L & 0 & 0 & 0 & 0 & 0\end{pmatrix}\begin{pmatrix}v(t)\\ x(t)\end{pmatrix}, \end{aligned}$$

which accounts for the contributions of the individual regions to a sensor's readings and the mean (dipole) orientation[77] of the considered region's neuronal populations.
Usually, this linear transformation also describes the spatial distribution of the sensors. In special cases, such as intracranial EEG (iEEG) recordings, the contribution matrix L is diagonal and thus only weights the considered regions signals individually; such a model will be assumed over the course of this work.

[77] A synchronously active neuronal region acts as a dipole of which its mean orientation is reflected by the sign of the associated entry in the contribution matrix.

6.4.3. Joint Model

The joint models in state-space form for the EEG & MEG dynamic causal model is given in the nonlinear case by:

$$\begin{pmatrix}\dot{v}(t)\\ \dot{x}(t)\end{pmatrix} = \begin{pmatrix}0 & \mathbb{1}_{5k}\\ -T^2\otimes\mathbb{1}_k & -2T\otimes\mathbb{1}_k\end{pmatrix}\begin{pmatrix}v(t)\\ x(t)\end{pmatrix} + \begin{pmatrix}0\\ A_v\end{pmatrix}\bar{\varsigma}_\kappa(A_\varsigma v(t)) + \frac{H_e}{\tau_e}(\delta_{8,1}^{10\times 1}\otimes B)u(t),$$

$$y(t) = \begin{pmatrix}0 & 0 & 0 & L & -L & 0 & 0 & 0 & 0 & 0\end{pmatrix}\begin{pmatrix}v(t)\\ x(t)\end{pmatrix}. \tag{6.11}$$

and in the linear(ized) case by:

$$\begin{pmatrix}\dot{v}(t)\\ \dot{x}(t)\end{pmatrix} = \begin{pmatrix}0 & \mathbb{1}_{5k}\\ -T^2\otimes\mathbb{1}_k & -2T\otimes\mathbb{1}_k\end{pmatrix}\begin{pmatrix}v(t)\\ x(t)\end{pmatrix} + \begin{pmatrix}0\\ A_vA_\varsigma\end{pmatrix}\frac{\kappa}{4}v(t) + \frac{H_e}{\tau_e}(\delta_{8,1}^{10\times 1}\otimes B)u(t),$$

$$y(t) = \begin{pmatrix}0 & 0 & 0 & L & -L & 0 & 0 & 0 & 0 & 0\end{pmatrix}\begin{pmatrix}v(t)\\ x(t)\end{pmatrix}. \tag{6.12}$$

Both models consists of second-order systems, reduced to first order, with associated state-spaces of dimension $10k$.

6.4.4. Parametrization

Only forward connections are considered, thus $A_B = A_L = 0$ and $A_F = A_F(\theta) = \mathrm{vec}^{-1}(\theta)$ with parameter-space dimension of k^2. For the parametrization of the connectivity a shift, like for the fMRI & fNIRS model parametrization, is generally not required to ensure stability as the intrinsic connections dampen the local activity. Similar to the fMRI & fNIRS model, the physiological parameters $\{H_e, H_i, \tau_e, \tau_i, \gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_5, \kappa\}$ are fixed at their respective prior mean value, while the matrix L is assumed to be a unit matrix $L = \mathbb{1}_k$. These constant parameters of the models are listed in Table 6.7, and are taken from [169, Tab. 1].

Excitatory Maximum Post-Synaptic Potential	H_e	[mV]	4
Inhibitory Maximum Post-Synaptic Potential	H_i	[mV]	32
Excitatory Average Lumped Rate Constant	τ_e	[ms]	4
Inhibitory Average Lumped Rate Constant	τ_i	[ms]	16
Intrinsic Connection Strength 1	γ_1		64
Intrinsic Connection Strength 2	γ_2		16
Intrinsic Connection Strength 3	γ_3		128
Intrinsic Connection Strength 4	γ_4		128
Intrinsic Connection Strength 5	γ_5		64
Sigmoid Slope	κ	[mV^{-1}]	0.56

Table 6.7.: *EEG & MEG model fixed physiological parameters.*

6.5. Bayesian Inference

To estimate the connectivity parameters from the previous presented models, a Bayesian approach is employed; its outline follows the concise description in [71].
Bayesian inference is based upon Bayes' rule,

$$P(\theta|y_d) = \frac{P(y_d|\theta)P(\theta)}{P(y_d)},$$

which states, that a **posterior distribution** $P(\theta|y_d)$ on the parameter θ for the data y_d is given by the **likelihood distribution** $P(y_d|\theta)$ of the data for the parameter multiplied by a **prior distribution** $P(\theta)$ representing beforehand beliefs on the parameter, divided by the evidence $P(y_d)$. Since the evidence is only a normalizing factor, the posterior distribution is proportional to the product of likelihood and prior distribution:

$$P(\theta|y_d) \propto P(y_d|\theta)P(\theta).$$

If the data is assumed to be the model's output that contains some additive noise,

$$y_d = y(\theta) + \epsilon,$$

the likelihood distribution can be expressed in terms of the noise:

$$\begin{aligned} \Rightarrow \epsilon &= y_d - y(\theta) \\ \Rightarrow P(\epsilon) &= P(y_d - y(\theta)) \\ \Rightarrow P(\theta|y_d) &\propto P(y_d - y(\theta))P(\theta). \end{aligned}$$

In case of Gaussian noise $\epsilon = \mathcal{N}_{0,v}$ with zero mean, the likelihood is also specified by a Gaussian distribution with mean $\mu_{y|\theta} \in \mathbb{R}^P$ and covariance $\sigma_{y|\theta} \in \mathbb{R}^{P\times P}$:

$$P(y_d|\theta) \propto \exp\Big(-\frac{1}{2}\|y(\theta) - y_d\|^2_{\sigma^{-1}_{y|\theta}}\Big).$$

If also a Gaussian prior with mean $\mu_\theta \in \mathbb{R}^P$ and covariance $\sigma_\theta \in \mathbb{R}^{P\times P}$ is provided,

$$P(\theta) \propto \exp\Big(-\frac{1}{2}\|\theta - \mu_\theta\|^2_{\sigma^{-1}_\theta}\Big),$$

then the posterior is specified by:

$$P(\theta|y_d) \propto \exp\Big(-\frac{1}{2}\|y(\theta) - y_d\|^2_{\sigma^{-1}_{y|\theta}} - \frac{1}{2}\|\theta - \mu_\theta\|^2_{\sigma^{-1}_\theta}\Big),$$

which is most-likely not Gaussian in case of a nonlinear parameter mapping $\theta \mapsto y(\theta)$.
A Bayesian variant of the maximum-likelihood (ML) estimator is the maximum-a-posteriori (MAP) estimator, which maximizes the product of likelihood and prior distribution. This is equal to minimizing the sum of negative log-likelihood and negative log-prior:

$$\begin{aligned} \theta_{\text{MAP}} &= \arg\max_{\theta\in\mathbb{R}^P} \exp\left(-\frac{1}{2}\|f(\theta) - y_d\|^2_{\sigma^{-1}_{y|\theta}} - \frac{1}{2}\|\theta - \mu_\theta\|^2_{\sigma^{-1}_\theta}\right) \\ &= \arg\min_{\theta\in\mathbb{R}^P} \left(\frac{1}{2}\|f(\theta) - y_d\|^2_{\sigma^{-1}_{y|\theta}} + \frac{1}{2}\|\theta - \mu_\theta\|^2_{\sigma^{-1}_\theta}\right). \end{aligned} \tag{6.13}$$

This can be seen as a weighted least-squares optimization with Tikhonov regularization.

7. Numerical Results

Contents

In this chapter the gramian-based and optimization-based model reduction approaches are evaluated for several models with respect to the presented model reduction techniques. For the gramian-based approach, the empirical (non-symmetric) cross gramian from Definition 3.17 and Section 3.3.3 with the implementation described in Section 5.3 is employed, and for the optimization-based approach, the enhanced variant from Section 4.4.1 with the implementation described in Section 5.4 is utilized.
Three sets of numerical experiments are considered. The respective underlying models for these experiments are:

1. the hyperbolic network model from Section 6.1;
2. the fMRI & fNIRS dynamic causal model from Section 6.3;
3. the EEG & MEG dynamic causal model from Section 6.4.

Each of these models is investigated in its original nonlinear form as well as in a linearized variant in terms of combined state- and parameter-space reduction. Additionally, a combined reduction is performed for the fMRI & fNIRS model in a Bayesian inverse problem context. This parameter inference corresponds to the neuroscientific application of statistical connectivity reconstruction from measured outputs y_d:

$$\theta_d = \arg\min_{\theta} \|y(\theta) - y_d\|_{\mathrm{L}_2}^2.$$

The considered models cover various classes of systems. While the hyperbolic network model is a mildly nonlinear system, the fMRI & fNIRS model comprises various nonlinearities and the EEG & MEG model is of second-order. A commonality among all models is the high-dimensional, yet homogeneous[78] parameter-space, which motivates the parameter-space reduction. The state-spaces for these experiments are relatively small, as the focus lies on the parameter-space and combined reduction.

[78] Since for the dynamic causal models all parameters embody connectivity between network nodes, they are of the same scale and mapping.

7.1. Experimental Setup

All presented numerical experiments are structured in a common scheme: after the computation of the gramian-based ROM using `emgr` (see Section 5.3) or optimization-based ROM by `optmor` (see Section 5.4), the ROMs are compared to the ***nonlinear*** FOM in the joint norms. Additionally, performance measurements during the computation and the evaluation of the ROMs are conducted.
The source code for the following experiments is openly available from the locations referenced in the code availability section Appendix B.1. A specification of the employed computational environment is given in Appendix B.2.

7.1.1. Error Measures

For the presented MOR methods, a joint state- and parameter-space time-domain norm of the output error trajectory between the original and reduced parametrized systems, as described in Section 2.4, is of interest. Both considered methods use a L_2-norm based method to compute the state-space ROM; the parameter-space ROM is obtained from a L_2-norm method for the gramian-based approach and from an L_∞-norm method for the optimization-based approach. Hence, the output error of the discrete trajectories for the state and parameter reduced systems is evaluated in the joint $\ell_2 \otimes \ell_2$-norm (2.16b) and the joint $\ell_2 \otimes \ell_\infty$-norm (2.16d). For the parameter-space component of the error assessment, a set of random parameter samples is generated with statistics specific to the respective experiment, since a systematic sampling of the considered parameter-spaces is infeasible due to their high dimension.

7.1.2. Offline and Online Phases

The experiments consist of two phases. During the first phase, which is called **offline phase**, the global ROM is assembled. In the second phase, subsequently called **online phase**, the ROM is utilized, which in this setting means their evaluation[79] to obtain model reduction errors in the aforementioned joint norms. For the inverse problem experiments, the statistical parameter inference, outlined in Section 6.5, using the ROM is considered the online phase.

7.1.3. Performance Measurement

The computational efficiency of the presented methods is evaluated in terms of compute requirements, which is indirectly measured through time consumption. Therefore, to analyze the performance of the gramian-based and the optimization-based approaches, the durations of the offline and online phases are recorded separately as elapsed wall-time. Since the optimization-based method assembles the reducing projections incrementally the duration of each iteration is highlighted. The memory consumption is not relevant due to the small state-space dimensions for the following experiments.

[79] The FOM evaluation is treated as an online phase without a prior offline phase.

7.2. Hyperbolic Network Model

A first set of experiments utilizes the hyperbolic network model from Section 6.1.1. In this setting a parametric ROM obtained by combined state and parameter reduction is evaluated for the nonlinear model and a linearized variant.

7.2.1. Parametrization

In this setting the (diagonal) gain matrix $K \in \mathbb{R}^{N \times N}$ is parametrized by mapping each diagonal entry to an individual parameter component θ_i:

$$K(\theta) := \begin{cases} K_{ii} = \theta_i \\ K_{ij} = 0, \quad i \neq j \end{cases}. \tag{7.1}$$

Hence, the parameter-space dimension is equal to the state-space dimension $P = N$ and the parameters are constrained to $\theta \in (0, 1]^P$. Such a parametrization is used, for example, in neural networks since the learning rate is related to the gain of the activation function [221].

7.2.2. Nonlinear Model

For this experiment, the parametrized variant of the original hyperbolic network model (6.1),

$$\begin{aligned} \dot{x}(t) &= A \tanh(K(\theta)x(t)) + Bu(t), \\ y(t) &= Cx(t), \end{aligned} \tag{7.2}$$

is used. The hyperbolic network model is mildly nonlinear and has a similar structure as the linear control system, yet the nonlinear parametrization occurs inside the nonlinearity.

7.2.3. Linearized Model

The linear variant of (6.1) uses a linearization around the steady-state $x_i(0) = 0$ and, as in (6.10), the sigmoid-like function $\xi(x(t))$ is linearly approximated by:

$$\tanh(K_{i*}x(t)) = \tanh(\theta_i x_i(t)) \approx \theta_i x_i(t).$$

This results in the following linear state-space system:

$$\begin{aligned} \dot{x}(t) &= AK(\theta)x(t) + Bu(t), \\ y(t) &= Cx(t), \end{aligned} \tag{7.3}$$

with the same parametrization of the gain matrix as the nonlinear model (7.1). In this nonlinear parametrization of the linearized hyperbolic network model, the gain matrix $K(\theta)$ becomes a slope matrix.

7.2.4. System Dimensions

For this experiment, the dimensions of the system components are selected as follows: The input and output dimensions are set to $J = O = 1$ yielding a SISO system, and the state and parameter dimensions are set to $N = P = 100$. A system matrix $A \in \mathbb{R}^{100\times100}$ is generated as a sparsely populated matrix of 10% density with elements drawn from a uniform random distribution $\mathcal{U}_{[0,10]}$. To ensure a stable system the diagonal entries are set to $A_{ii} = -10$ ([228]). The components of the parameter θ constituting the (diagonal of the) gain matrix $K \in \mathbb{R}^{100\times100}$ are constraint to the interval $\theta_i \in [\frac{1}{10}, 1]$. A sparse input matrix $B \in \mathbb{R}^{100\times1}$ also of density 10% with elements drawn from a uniform random distribution $\mathcal{U}_{[0,10]}$ is utilized. To excite the network, a delta impulse is selected as an external input source $u(t) = \delta(t)$. The components of the output matrix $C \in \mathbb{R}^{1\times100}$ are sampled from a uniform random distribution $\mathcal{U}_{[0,1]}$. An RK2 time-discretization of 200 uniform time-steps $\Delta t = \frac{1}{100}$ on the interval $[0, 2]$ is utilized to generate trajectories.

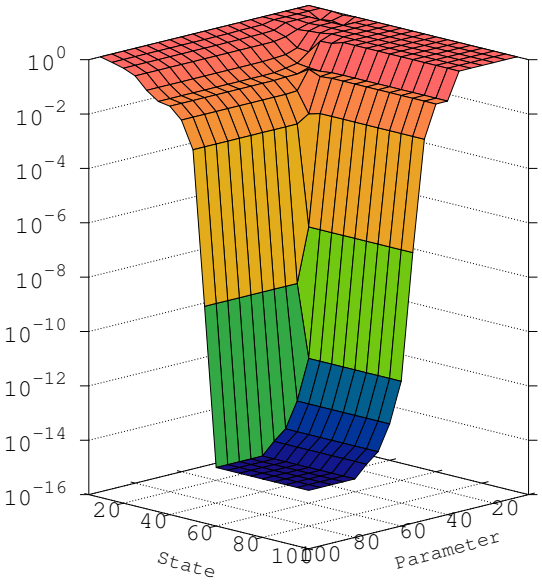
7.2.5. Combined Reduction

To both, the nonlinear and the linearized hyperbolic network models, the gramian-based and optimization-based combined reduction methods are applied and the resulting ROMs are compared in the $\ell_2 \otimes \ell_2$- and $\ell_2 \otimes \ell_\infty$-norms to the nonlinear FOM over 100 uniformly distributed random samples from the parameter-space $[\frac{1}{10}, 1]^{100}$. The gramian-based combined reduction is performed by the empirical joint gramian from Definition 3.38, which yields an (empirical) cross gramian (Definition 3.32) for the state-space reduction and an (empirical) cross-identifiability gramian (Definition 3.39) for the parameter-space reduction. Since (7.2) and (7.3) resemble SISO systems for the selected input/output components, the regular cross gramian can be applied. The optimization-based combined reduction (4.13) is accomplished using the enhanced algorithm variant from Section 4.4.1, and for the inner optimization the Tikhonov regularization coefficient is set to $\beta_2 = \frac{1}{10}$.

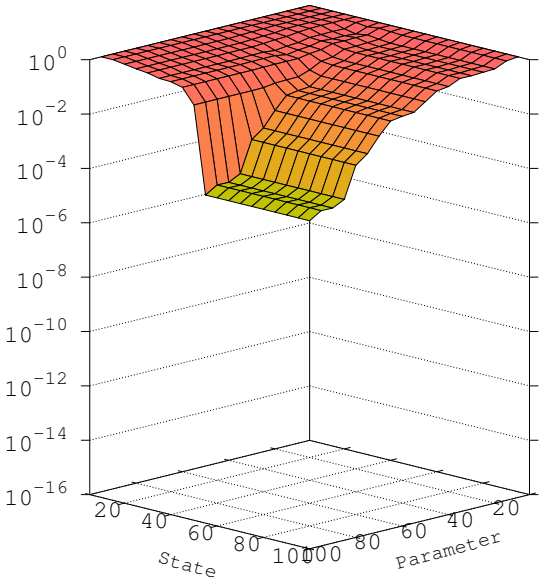
7.2.6. Combined Reduction Error

The numerical results[80] in Figure 7.1 and Figure 7.2 show the relative $\ell_2 \otimes \ell_2$-norm and $\ell_2 \otimes \ell_\infty$-norm output error surfaces for varying state and parameter dimensions resulting from the gramian-based and optimization-based combined reduction for the nonlinear and linearized hyperbolic network model respectively. For the nonlinear hyperbolic network model, Figure 7.1a and Figure 7.1c depict the combined reduction error of the ROMs from the gramian-based approach in the $\ell_2 \otimes \ell_2$- and $\ell_2 \otimes \ell_\infty$-norm; Figure 7.1b and Figure 7.1d illustrate the combined reduction error of the ROMs from the optimization-based approach in the $\ell_2 \otimes \ell_2$- and $\ell_2 \otimes \ell_\infty$-norm. For the linearized hyperbolic network model, Figure 7.2a and Figure 7.2c visualize the combined reduction error of the ROMs from the gramian-based approach in the $\ell_2 \otimes \ell_2$- and $\ell_2 \otimes \ell_\infty$-norm; Figure 7.2b and Figure 7.2d show the combined reduction error of the ROMs from the optimization-based approach in the $\ell_2 \otimes \ell_2$- and $\ell_2 \otimes \ell_\infty$-norm.

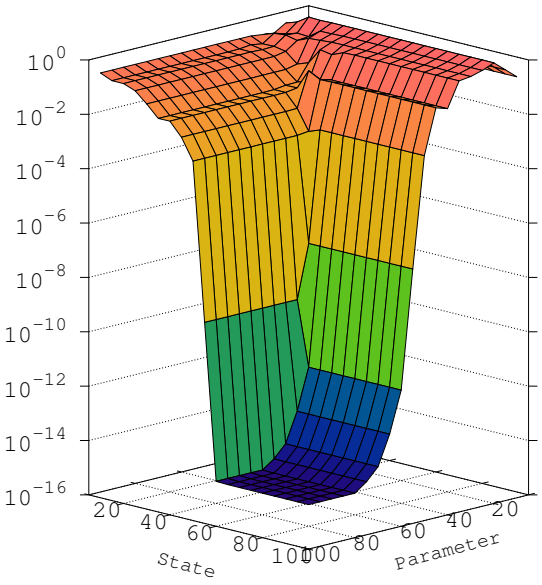
[80] See `code/ch7/hnm.m` in the supplementary source code archive referenced in Appendix B.1.



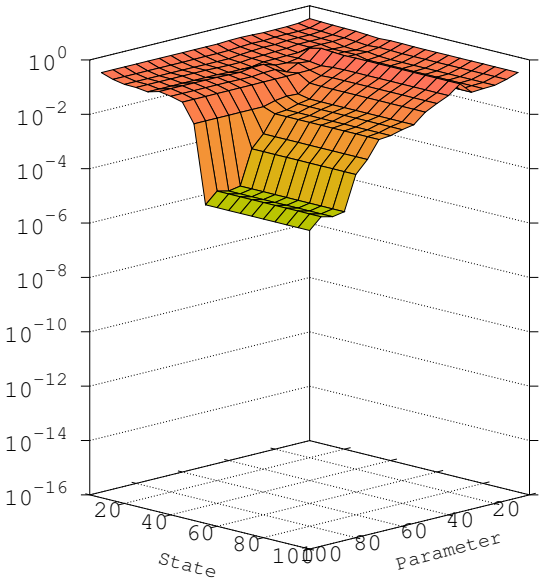
(a) *Relative $\ell_2 \otimes \ell_2$ output error for gramian-based combined reduction.*



(b) *Relative $\ell_2 \otimes \ell_2$ output error for optimization-based combined reduction.*

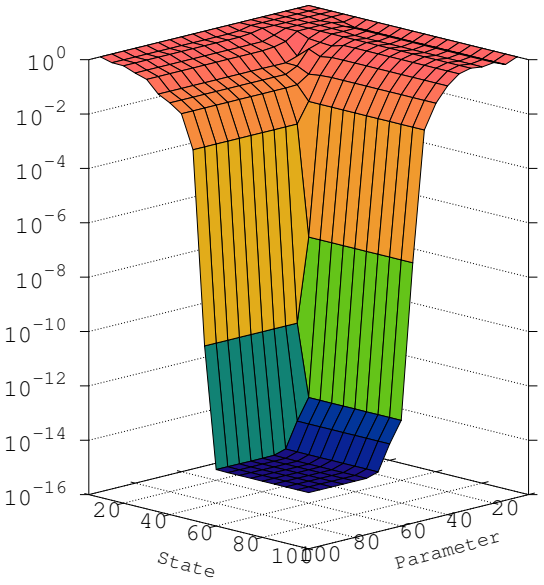


(c) *Relative $\ell_2 \otimes \ell_\infty$ output error for gramian-based combined reduction.*

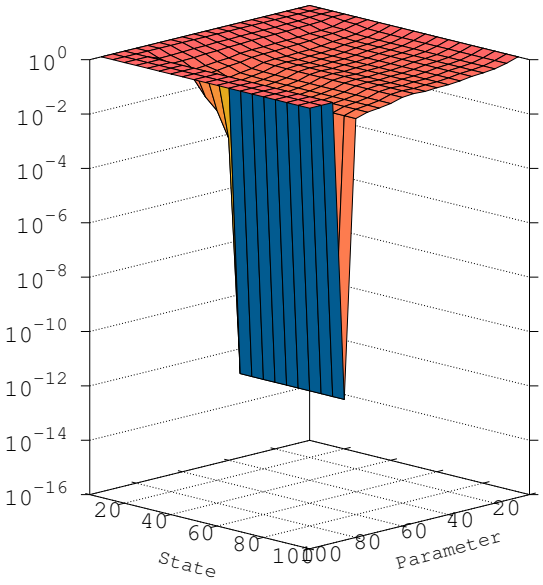


(d) *Relative $\ell_2 \otimes \ell_\infty$ output error for optimization-based combined reduction.*

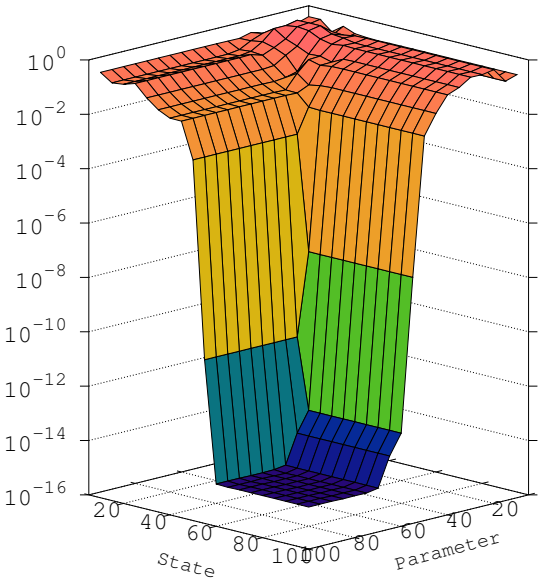
Figure 7.1.: *Numerical results for the combined state and parameter reduction of the* ***nonlinear*** *hyperbolic network model.*



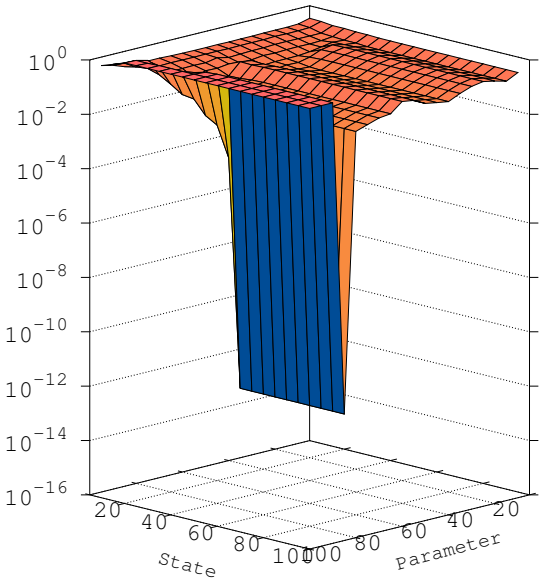
(a) *Relative $\ell_2 \otimes \ell_2$ output error for gramian-based combined reduction.*



(b) *Relative $\ell_2 \otimes \ell_2$ output error for optimization-based combined reduction.*

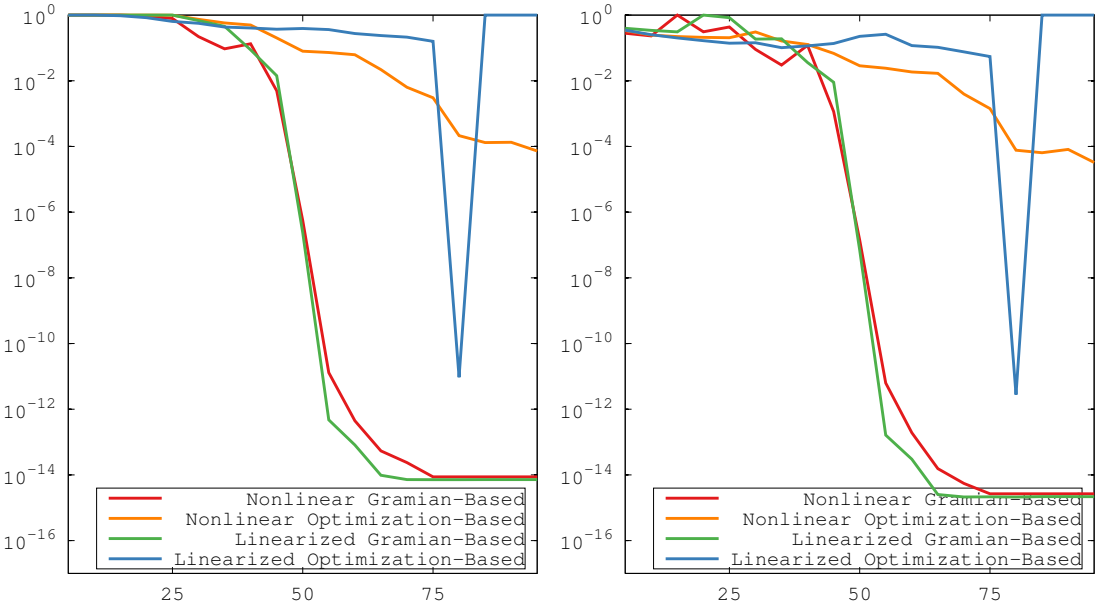


(c) *Relative $\ell_2 \otimes \ell_\infty$ output error for gramian-based combined reduction.*



(d) *Relative $\ell_2 \otimes \ell_\infty$ output error for optimization-based combined reduction.*

Figure 7.2.: *Numerical results for the combined state and parameter reduction of the* ***linearized*** *hyperbolic network model.*



(a) *Relative $\ell_2 \otimes \ell_2$ output error of gramian- and optimization-based combined reduction for the nonlinear and linearized model.*

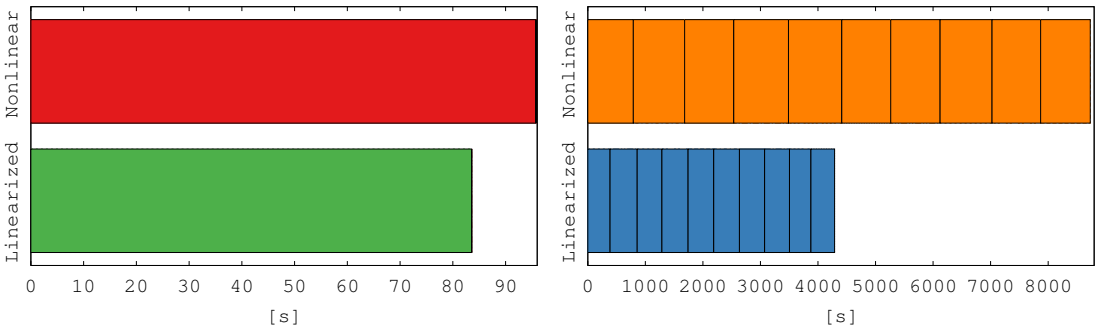
(b) *Relative $\ell_2 \otimes \ell_\infty$ output error of gramian- and optimization-based combined reduction for the nonlinear and linearized model.*

Figure 7.3.: *Relative output errors of the reduced order hyperbolic network models for equally reduced state-parameter-space dimensions $n = p$.*

For both combined reduction methods the joint error is dominated by the parameter-space reduction error. While the gramian-based method generates ROMs of similar accuracy for the nonlinear and linearized model, which eventually approach machine precision in both norms, the optimization-based method constructs less accurate ROMs for the nonlinear model and fails to produce usable ROMs for the linearized model. Notably, the errors are not decaying for ROMs with a small number of base components, but require about half of the FOM's order in state- and parameter-space dimension.

A comparison of the combined reduction error of both methods[81] for concurrently reduced state- and parameter-space dimensions $n = p$ is diagrammed in Figure 7.3. Figure 7.3a and Figure 7.3b show the error in the $\ell_2 \otimes \ell_2$- and $\ell_2 \otimes \ell_\infty$-norm respectively. In case of the gramian-based approach, the ROMs derived from the nonlinear and linearized models reach an accuracy of 10^{-6} for $n = p \geq 50$ and of 10^{-12} for $n = p \geq 60$ with a similar slope for both considered norms, with the linearized variant exhibiting slightly better accuracy. The optimization-based approach yields ROMs for the nonlinear model with an error of 10^{-2} for $n = p \geq 70$ and of 10^{-4} for $n = p \geq 76$, the ROMs from the linearized model do not surpass an error of 10^{-3}. Comparing the respective ROMs, the error of the gramian-based ROMs declines steeply, while the error of the (nonlinear) optimization-based ROMs decays slowly. The sudden drop in the error of the optimization-based ROMs appears only for $n = p = 64$.

[81] See also the additional figures in Appendix C.1.



(a) *Offline times for gramian-based combined reduction in seconds.*

(b) *Offline times for optimization-based combined reduction with highlighted per* 10 *iteration durations in seconds.*

Figure 7.4.: *Comparison of offline timings (note the different scales) for the combined reduction of the hyperbolic network models.*

7.2.7. Combined Reduction Performance

In Figure 7.4 the offline times for the nonlinear and the linearized model of the gramian-based ROM (Figure 7.4a) and the optimization-based ROM (Figure 7.4b) are compared. The gramian-based approach needs 95 s with the nonlinear model and about 84 s (−13%) with the linearized model. For $P = 100$ iterations, the optimization-based approach needs about 8700 s with the nonlinear model and about 4300 s (−51%) with the linearized model. On average an iteration of the optimization-based method needs for the nonlinear model 87 s and 43 s (−51%) for the linear model; thus the offline time of the gramian-based method takes less than the duration of two iterations of the optimization-based approach.
This comparison is only an indicator of performance, since for the gramian-based ROM a full SVD of the empirical cross gramian and empirical cross identifiability gramian is computed. To lower offline times in large-scale settings [18] a truncated SVD, based on the singular values, is used to obtain reducing projections from the empirical gramians, for example by an adaptive Lanczos procedure [24, 38]. For the optimization-based reduction, the error indicator (Section 5.4.4) can be used to avoid iterating many times.

7.2.8. Assessment

In a direct comparison in terms of accuracy and performance, the gramian-based approach, utilizing the empirical joint gramian, produces better results, as the model reduction error of the ROMs decays to lower levels and this method consumes less than 2% of the overall offline time of the optimization-based approach. This difference in performance results from the number of required simulated trajectories, which relates linearly to the parameter-space dimension for the gramian-based approach and superlinearly for the optimization-based approach (see also: (8.1), (8.2)).

7.3. fMRI & fNIRS Dynamic Causal Model

The second set of numerical experiments encompasses a combined state and parameter reduction of the nonlinear and linearized fMRI & fNIRS dynamic causal models presented in Section 6.3.

7.3.1. Parametrization

The parametrized connectivity matrix $A(\theta)$ is set up as for a hypothesis test experiment. A hypothesis on the connectivity is selected as a sparse and stabilized matrix $A_1 \in \mathbb{R}^{k\times k}$, $A_{1,ii} = -1$. The parameter $\theta \in \mathbb{R}^{k^2}$ is then scaling the elements of the hypothesis:

$$A(\theta) = A_1 \odot \mathrm{vec}^{-1}(\theta),$$

by the Hadamard product with the inverse vectorization of the parameter.

7.3.2. Nonlinear Model

The fMRI & fNIRS model is a coupled system of a linear component ($\dot{x}$) with nonlinear parametrization, a second-order component ($\dot{s}$, $\dot{f}$) and nonlinear components ($\dot{v}$, $\dot{q}$). For observed k regions of the brain, the joint dynamic and forward submodel,

$$\begin{pmatrix} \dot{x}(t) \\ \dot{s}(t) \\ \dot{f}(t) \\ \dot{v}(t) \\ \dot{q}(t) \end{pmatrix} = \begin{pmatrix} A(\theta)x(t) \\ x(t) - \kappa s(t) - \gamma(f(t) - \vec{1}_k) \\ s(t) \\ \frac{1}{\tau}(f(t) - v(t)^{\frac{1}{\alpha}}) \\ \frac{1}{\tau}(\frac{1}{\rho} f(t)(\vec{1}_k - ((1-\rho)\,\vec{1}_k)^{\frac{1}{f(t)}}) - v(t)^{\frac{1}{\alpha}-1} \odot q(t)) \end{pmatrix} + \begin{pmatrix} B \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} u(t),$$
$$\tilde{y}(t) = V_0(k_1(\vec{1}_k - q(t)) + k_2(\vec{1}_k - v(t))),$$

has total state-space dimension of $N = 5k$. For a single input and due to the number of $O = k$ outputs, this is a single-input-multiple-output (SIMO) system.

7.3.3. Linearized Model

By a linearization of the forward submodel's components of the vector field and the output functional, the linear joint dynamic and forward model is given by (6.5):

$$\begin{pmatrix} \dot{x}(t) \\ \dot{s}(t) \\ \dot{f}(t) \\ \dot{v}(t) \\ \dot{q}(t) \end{pmatrix} = \begin{pmatrix} A(\theta) & 0 & 0 & 0 & 0 \\ \mathbb{1}_k & -\kappa\,\mathbb{1}_k & -\gamma\,\mathbb{1}_k & 0 & 0 \\ 0 & \mathbb{1}_k & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\tau}\,\mathbb{1}_k & -\frac{1}{\tau\alpha}\,\mathbb{1}_k & 0 \\ 0 & 0 & \frac{(1-(1-\rho)(1-\ln(1-\rho)))}{\tau\rho}\,\mathbb{1}_k & -\frac{1-\alpha}{\tau\alpha}\,\mathbb{1}_k & -\frac{1}{\tau}\,\mathbb{1}_k \end{pmatrix} \begin{pmatrix} x(t) \\ s(t) \\ f(t) \\ v(t) \\ q(t) \end{pmatrix} + \begin{pmatrix} B \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} u(t),$$
$$\tilde{y}(t) = V_0(k_1 q(t) - k_2 v(t)),$$

with the same total state-space dimension and parametrization as the nonlinear model.

7.3.4. System Dimensions

For these experiments the number of brain regions is set to $k = 16$, thus the state-space dimension is $N = 5k = 80$ and the parameter-space dimension is $P = k^2 = 256$. Since the parameters can affect the stability of the system, the hypothesis matrix A_1 is selected as sparse and uniform random distributed with a density of about 6% and elements drawn from $\mathcal{U}_{[0,1]}$. The parameter components θ_i associated to off-diagonal elements of $\operatorname{vec}^{-1}(\theta)$ are drawn from a log-normal random distribution $\ln \mathcal{N}_{0,\frac{1}{4}}$, and parameters corresponding to diagonal elements from $\ln \mathcal{N}_{0,\frac{1}{40}}$. A single source of input $J = 1$ is considered resulting in $B \in \mathbb{R}^{80\times 1}$ driven by an impulse input $u(t) = \delta(t)$; the input matrix B is generated as a sparse matrix with a single nonzero entry which is drawn from a uniform random distribution $\mathcal{U}_{[0,1]}$. For all methods an RK2 time-discretization of 200 uniform time-steps $\Delta t = \frac{1}{10}$ on the interval $[0, 20]$ is utilized to generate trajectories. The fixed parameters $\{\kappa, \gamma, \alpha, \tau, \rho, V_0, k_1, k_2\}$ are taken from Table 6.4.

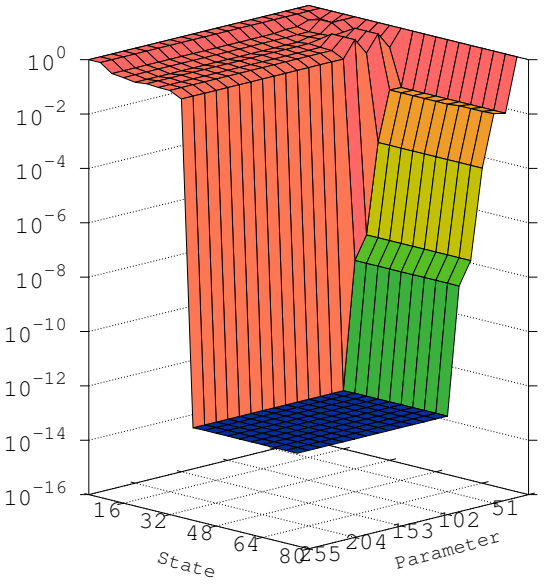
7.3.5. Combined Reduction

To both, the nonlinear and the linearized fMRI & fNIRS dynamic causal models, the gramian-based and optimization-based combined reduction methods are applied and the resulting ROMs are compared in the $\ell_2 \otimes \ell_2$- and $\ell_2 \otimes \ell_\infty$-norms to the nonlinear FOM over 100 sparse log-normally distributed random samples. The gramian-based combined reduction is performed by the empirical joint gramian from Definition 3.38 in the non-symmetric variant due to the SIMO nature of the system, which yields an (empirical) non-symmetric cross gramian (Definition 3.17) for the state-space reduction and an (empirical) cross-identifiability gramian (Definition 3.39) for the parameter-space reduction. The optimization-based combined reduction (4.13) is accomplished using the enhanced algorithm variant from Section 4.4.1, and for the inner optimization of the greedy sampling, the Tikhonov regularization coefficient is set to $\beta_2 = \frac{1}{10}$.

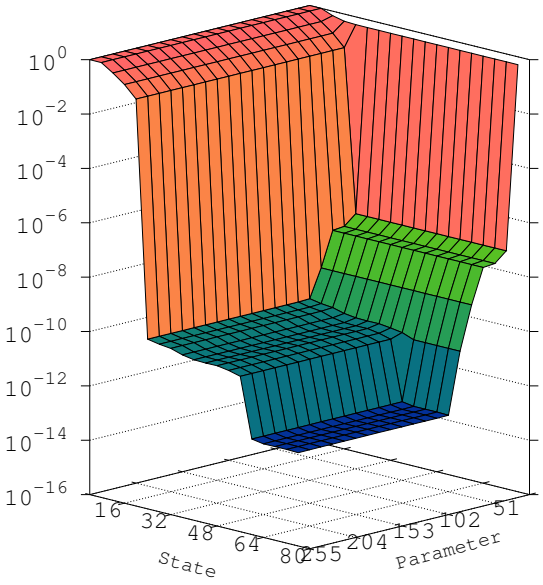
7.3.6. Combined Reduction Error

The numerical results[82] in Figure 7.5 and Figure 7.6 show the relative $\ell_2 \otimes \ell_2$-norm and $\ell_2 \otimes \ell_\infty$-norm output error surfaces for varying state and parameter dimensions resulting from the gramian-based and optimization-based combined reduction for the nonlinear and linearized fMRI & fNIRS dynamic causal model respectively. For the nonlinear fMRI & fNIRS dynamic causal model, Figure 7.5a and Figure 7.5c depict the combined reduction error of the ROMs from the gramian-based approach in the $\ell_2 \otimes \ell_2$- and $\ell_2 \otimes \ell_\infty$-norm; Figure 7.5b and Figure 7.5d illustrate the combined reduction error of the ROMs from the optimization-based approach in the $\ell_2 \otimes \ell_2$- and $\ell_2 \otimes \ell_\infty$-norm. For the linearized fMRI & fNIRS dynamic causal model, Figure 7.6a and Figure 7.6c visualize the combined reduction error of the ROMs from the gramian-based approach in the $\ell_2 \otimes \ell_2$- and $\ell_2 \otimes \ell_\infty$-norm; Figure 7.6b and Figure 7.6d show the combined reduction error of the ROMs from the optimization-based approach in the $\ell_2 \otimes \ell_2$- and $\ell_2 \otimes \ell_\infty$-norm.

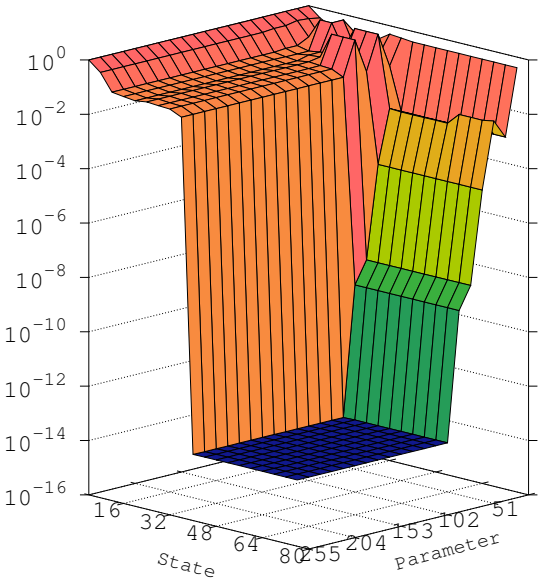
[82] See `code/ch7/mri.m` in the supplementary source code archive referenced in Appendix B.1.



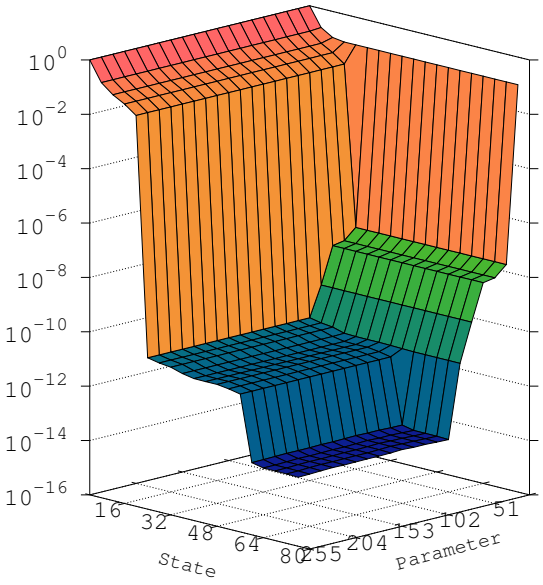
(a) *Relative $\ell_2 \otimes \ell_2$ output error for gramian-based combined reduction.*



(b) *Relative $\ell_2 \otimes \ell_2$ output error for optimization-based combined reduction.*

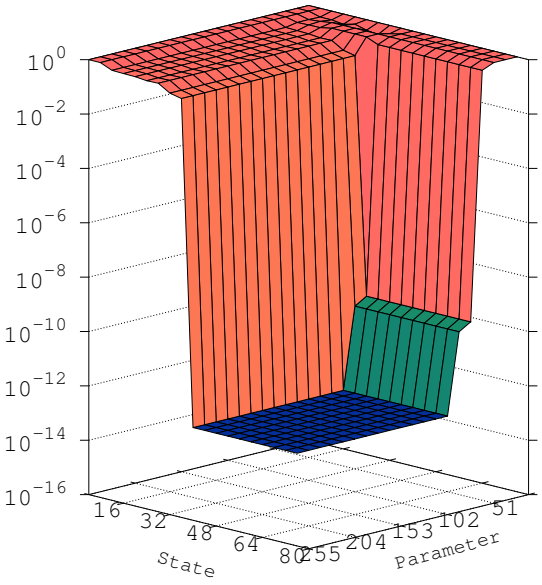


(c) *Relative $\ell_2 \otimes \ell_\infty$ output error for gramian-based combined reduction.*

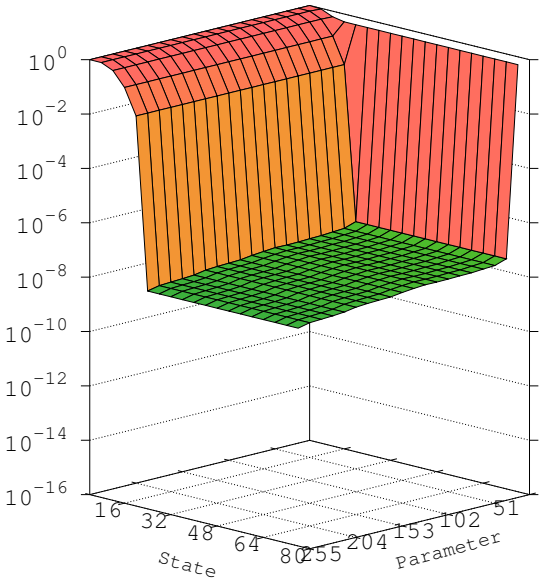


(d) *Relative $\ell_2 \otimes \ell_\infty$ output error for optimization-based combined reduction.*

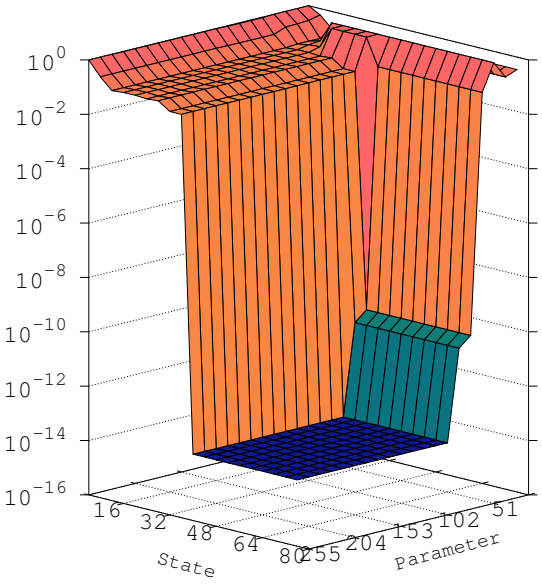
Figure 7.5.: *Numerical results for the combined state and parameter reduction of the* ***nonlinear*** *fMRI & fNIRS dynamic causal model.*



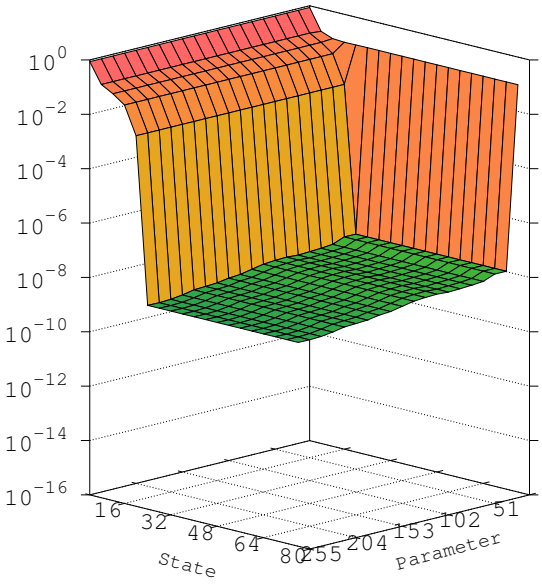
(a) *Relative $\ell_2 \otimes \ell_2$-norm error for gramian-based combined reduction.*



(b) *Relative $\ell_2 \otimes \ell_2$-norm error for optimization-based combined reduction.*

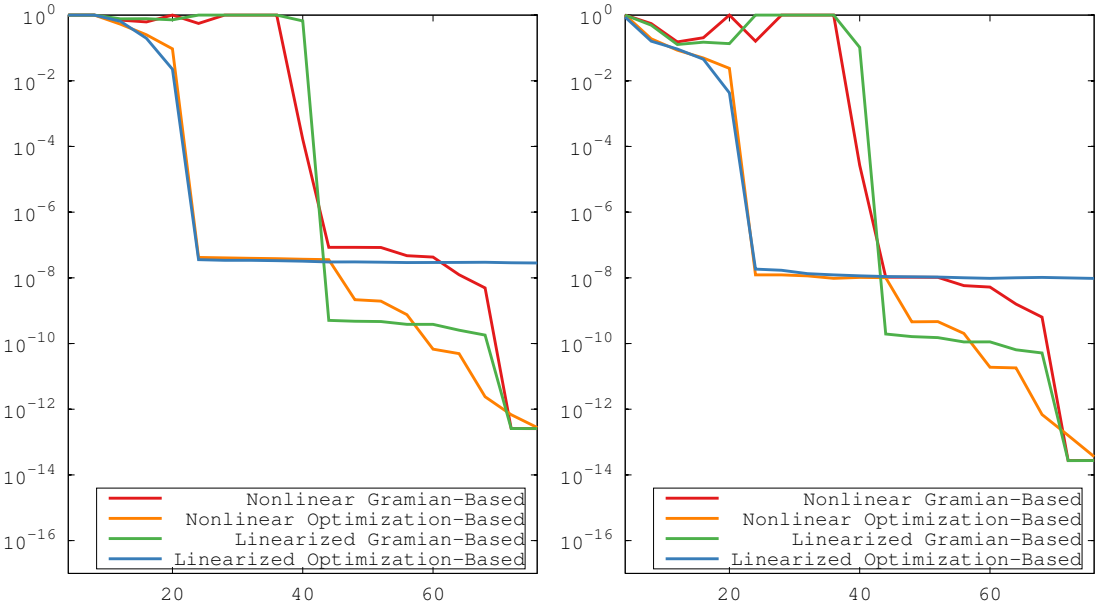


(c) *Relative $\ell_2 \otimes \ell_\infty$-norm error for gramian-based combined reduction.*



(d) *Relative $\ell_2 \otimes \ell_\infty$-norm error for optimization-based combined reduction.*

Figure 7.6.: *Numerical results for the combined state and parameter reduction of the* ***linearized*** *fMRI & fNIRS dynamic causal model.*



(a) *Relative $\ell_2 \otimes \ell_2$ output error of gramian- and optimization-based combined reduction for the nonlinear and linearized model.*

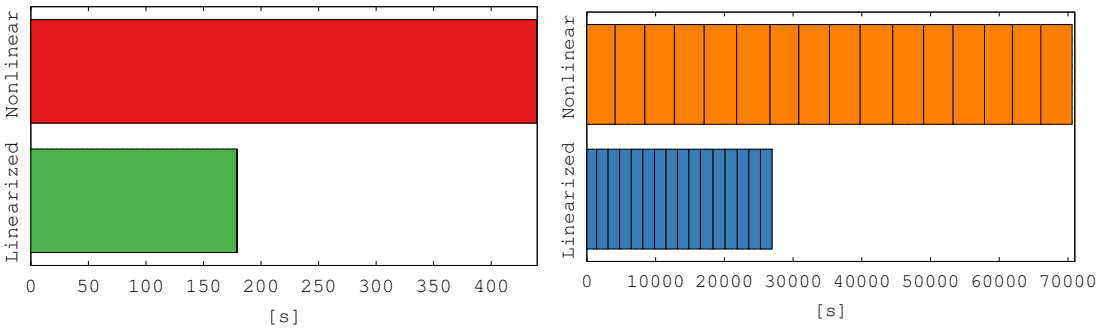
(b) *Relative $\ell_2 \otimes \ell_\infty$ output error of gramian- and optimization-based combined reduction for the nonlinear and linearized model.*

Figure 7.7.: *Relative output errors of the reduced order fMRI & fNIRS dynamic causal models for equally reduced state-parameter-space dimensions $n = p \leq 80$.*

For both approaches the joint error is relatively dominated by the state-space reduction error and the ROMs from the respective linearized variants show a slightly preferably slope in error decay. In case of the gramian-based combined reduction, the ROMs derived from the nonlinear and linearized models are of similar accuracy; the ROMs from the optimization-based combined reduction show a similar behavior for lower order ROMs, yet the combined reduction error from the nonlinear variant reaches a lower level than from the linearized variant.

A comparison of the combined model reduction error of both considered methods[83] for the first $n = p \leq N = 80$ concurrently reduced state- and parameter-space dimensions is diagrammed in Figure 7.7. Figure 7.7a and Figure 7.7b show the error in the $\ell_2 \otimes \ell_2$- and $\ell_2 \otimes \ell_\infty$-norm respectively. These visualizations show that the optimization-based approaches produce more accurate models than the gramian-based approaches for $n = p \leq 44$. For $n = p \geq 44$, the gramian-based method reaches an error of order 10^{-8} using the nonlinear model, but utilizing the linearized variant an error of order 10^{-10} is obtained. The error of the ROMs from the linearized optimization-based method flattens out at 10^{-8} for $n = p \geq 24$, while error of the ROMs from the nonlinear optimization-based method decays up to order 10^{-12} for $n = p \geq 72$. Compared to the gramian-based approach the optimization-based approach needs about 20 fewer reduced state and parameter base components to reach an error of about 10^{-8}.

[83] See also the additional figures in Appendix C.2.



(a) *Offline times for gramian-based combined reduction in seconds.*

(b) *Offline times for optimization-based combined reduction with highlighted per* 16 *iteration durations in seconds.*

Figure 7.8.: *Comparison of offline timings (note the different scales) for the combined reduction of the fMRI & fNIRS dynamic causal model.*

7.3.7. Combined Reduction Performance

In Figure 7.8 the offline times for the nonlinear and the linearized model of the gramian-based ROM (Figure 7.8a) and the optimization-based ROM (Figure 7.8b) are compared. The gramian-based approach needs about 440 s with the nonlinear model and about 180 s (−40%) with the linearized model. For $P = 100$ iterations, the optimization-based approach needs about 71000 s with the nonlinear model and 27000 s (−62%) with the linearized model. As this model has a more complicated structure and contains several nonlinearities compared to the hyperbolic network model from Section 7.2 this acceleration is not surprising. On average an iteration of the optimization-based method needs for the nonlinear model 276 s and 105 s (−62%) for the linear model; thus the offline time of the gramian-based method takes less than the duration of two iterations of the optimization-based approach. The parameter-space dimension is the driving factor for the performance of the experiments in this chapter. Compared to the experiment in Section 7.2, the parameter-space dimension is 2.56 times larger. And while the gramian-based approach has a 4.6 (2.1) times longer offline phase, the optimization-based approach takes 8.1 (6.3) times longer.

7.3.8. Assessment

In this experiment the optimization-based combined reduction produces more accurate ROMs, yet the gramian-based method requires only 0.7% of the overall offline time of the optimization-based method. Also, a model reduction error near machine precision is not reached. Due to the sparse and sequential structure, the fMRI & fNIRS dynamic causal model has only little redundant information, which does not encourage a modularized reduction of the SISO components as in [118]. This is underscored by the slowly decaying HSVs of an individual SISO hemodynamic forward system (see Appendix A.5).

7.4. EEG & MEG Dynamic Causal Model

The third set of numerical experiments investigates a combined state and parameter reduction of the nonlinear and linearized EEG & MEG dynamic causal model from Section 6.4.

7.4.1. Parametrization

A similar parametrization as for the fMRI & fNIRS experiments in Section 7.3.1 is selected for the forward connectivity matrix $A_F(\theta)$ of the EEG & MEG model. For the parametrized $A_F(\theta)$ a sparse matrix $A_1 \in \mathbb{R}^{k\times k}$ with a zero diagonal $A_{1,ii} = 0$ is set up. The parameter $\theta \in \mathbb{R}^{k^2}$ is then scaling the elements of the hypothesis:

$$A_F(\theta) = A_1 \odot \operatorname{vec}^{-1}(\theta),$$

by the Hadamard product with the inverse vectorization of the parameter.

7.4.2. Nonlinear Model

For k considered regions (sensors) of the brain, the joint dynamic and forward submodel of the nonlinear EEG & MEG dynamic causal model, described in Section 6.4,

$$\begin{aligned}\begin{pmatrix}\dot v(t)\\ \dot x(t)\end{pmatrix} &= \begin{pmatrix}0 & \mathbb{1}_{5k}\\ -T^2\otimes\mathbb{1}_k & -2T\otimes\mathbb{1}_k\end{pmatrix}\begin{pmatrix}v(t)\\ x(t)\end{pmatrix} + \begin{pmatrix}0\\ A_v(\theta)\end{pmatrix}\bar\varsigma(A_\varsigma v(t)) + \frac{H_e}{\tau_e}\delta_{8,1}^{10\times 1}\otimes Bu(t),\\ y(t) &= \begin{pmatrix}0 & 0 & 0 & L & -L\end{pmatrix} v(t),\end{aligned} \tag{7.4}$$

has total state-space dimension of $N = 10k$. This is a nonlinear second-order system, with the nonlinearity contained in the sigmoid-like function $\bar\varsigma$. The parametrization encompasses an element-wise parameter mapping as in (6.3) of the forward connectivity matrix $A_F(\theta)$, which is part of a block in $A_v(\theta)$,

$$A_v(\theta) = \begin{pmatrix}\frac{H_e}{\tau_e}\gamma_1\mathbb{1}_k & 0 & 0 & 0 & 0\\ 0 & \frac{H_i}{\tau_i}\gamma_2\mathbb{1}_k & 0 & 0 & 0\\ 0 & 0 & \frac{H_e}{\tau_e}(A_F(\theta)+\gamma_3\mathbb{1}_k) & 0 & 0\\ 0 & 0 & 0 & \frac{H_e}{\tau_e}\gamma_4\mathbb{1}_k & 0\\ 0 & 0 & 0 & 0 & \frac{H_i}{\tau_i}\gamma_5\mathbb{1}_k\end{pmatrix}.$$

7.4.3. Linearized Model

By a linearization of the dynamic submodel's nonlinear component of the vector field as detailed in (6.11), the linear joint dynamic and forward model is given by (6.12):

$$\begin{aligned}\begin{pmatrix}\dot v(t)\\ \dot x(t)\end{pmatrix} &= \begin{pmatrix}0 & \mathbb{1}_{5k}\\ -T^2\otimes\mathbb{1}_k & -2T\otimes\mathbb{1}_k\end{pmatrix}\begin{pmatrix}v(t)\\ x(t)\end{pmatrix} + \begin{pmatrix}0\\ A_v(\theta)A_\varsigma\end{pmatrix}\frac{\kappa}{4}v(t) + \frac{H_e}{\tau_e}\delta_{8,1}^{10\times 1}\otimes Bu(t),\\ y(t) &= \begin{pmatrix}0 & 0 & 0 & L & -L\end{pmatrix} v(t),\end{aligned} \tag{7.5}$$

with the same total state-space dimension and parametrization as the nonlinear model.

7.4.4. System Dimensions

For this experiment the number of considered brain regions is also set to $k = 16$, thus the state-space dimension is $N = 10k = 160$ and the forward connectivity matrix of the dynamical submodel is then of dimension $A_F(\theta) \in \mathbb{R}^{16\times 16}$, while the parameter-space dimension is $P = k^2 = 256$. The elements of the hypothesis matrix A_1 are drawn from a sparse and uniform random distribution $\mathcal{U}_{[0,32]}$ with a density of about 6%, and the parameter components θ_i, associated to off-diagonal elements of $\text{vec}^{-1}(\theta)$, are drawn from a log-normal random distribution $\ln \mathcal{N}_{0,\frac{1}{4}}$, while parameters corresponding to diagonal elements are drawn from $\ln \mathcal{N}_{0,\frac{1}{40}}$. An impulse input $u(t) = \delta(t)$ excites the system through a sparse input matrix $B \in \mathbb{R}^{16\times 1}$ with a single non-zero element drawn from a uniform random distribution $\mathcal{U}_{[0,1]}$. For all trajectories an RK2 time-discretization of 200 uniform time-steps $\Delta t = \frac{1}{100}$ on the interval $[0, 2]$ is utilized. The fixed parameters $\{H_e, H_i, \tau_e, \tau_i, \gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_5, \kappa\}$ are taken from Table 6.7.

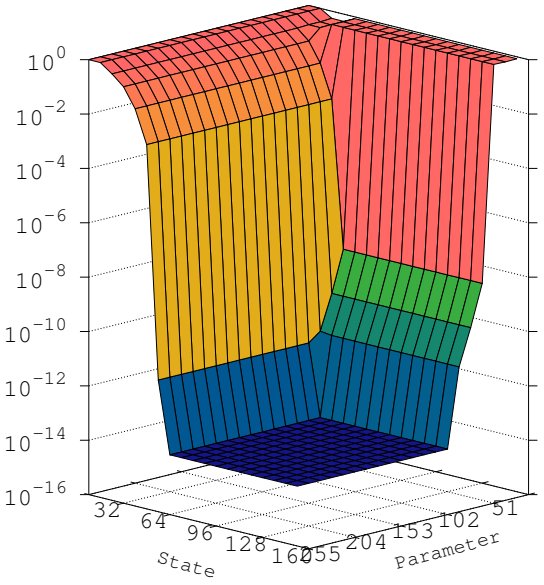
7.4.5. Combined Reduction

To both, the nonlinear and the linearized EEG & MEG dynamic causal models, the gramian-based and optimization-based combined reduction methods are applied and the resulting ROMs are compared in the $\ell_2 \otimes \ell_2$- and $\ell_2 \otimes \ell_\infty$-norms to the nonlinear FOM over 100 sparse log-normally distributed random samples. For the SIMO systems (7.4) and (7.5) the non-symmetric joint gramian is applied using the structure-preserving balancing variant from Section 3.2.4, which yields a (empirical) non-symmetric cross gramian (Definition 3.17) for the state-space reduction and an (empirical) cross-identifiability gramian (Definition 3.39) for the parameter-space reduction. The optimization-based combined reduction (4.13) is accomplished using the enhanced algorithm variant. For the inner optimization of the greedy sampling, the Tikhonov regularization coefficient is set to $\beta_2 = \frac{1}{10}$. Either method uses solely "position" information for the state reduction.

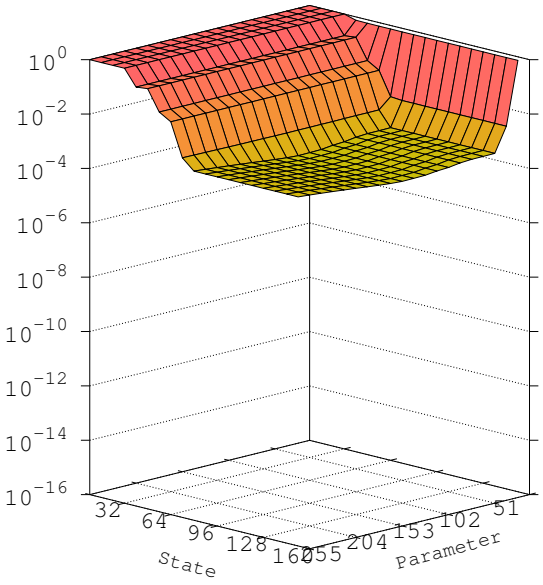
7.4.6. Combined Reduction Error

The numerical results[84] in Figure 7.9 and Figure 7.10 show the relative $\ell_2 \otimes \ell_2$-norm and $\ell_2 \otimes \ell_\infty$-norm output error surfaces for varying state and parameter dimensions resulting from the gramian-based and optimization-based combined reduction for the nonlinear and linearized EEG & MEG dynamic causal model respectively. For the nonlinear EEG & MEG dynamic causal model, Figure 7.5a and Figure 7.9c depict the combined reduction error of the ROMs from the gramian-based approach in the $\ell_2 \otimes \ell_2$- and $\ell_2 \otimes \ell_\infty$-norm; Figure 7.9b and Figure 7.9d illustrate the combined reduction error of the ROMs from the optimization-based approach in the $\ell_2 \otimes \ell_2$- and $\ell_2 \otimes \ell_\infty$-norm. For the linearized EEG & MEG dynamic causal model, Figure 7.10a and Figure 7.10c visualize the combined reduction error of the ROMs from the gramian-based approach in the $\ell_2 \otimes \ell_2$- and $\ell_2 \otimes \ell_\infty$-norm; Figure 7.10b and Figure 7.10d show the combined reduction error of the ROMs from the optimization-based approach in the $\ell_2 \otimes \ell_2$- and $\ell_2 \otimes \ell_\infty$-norm.

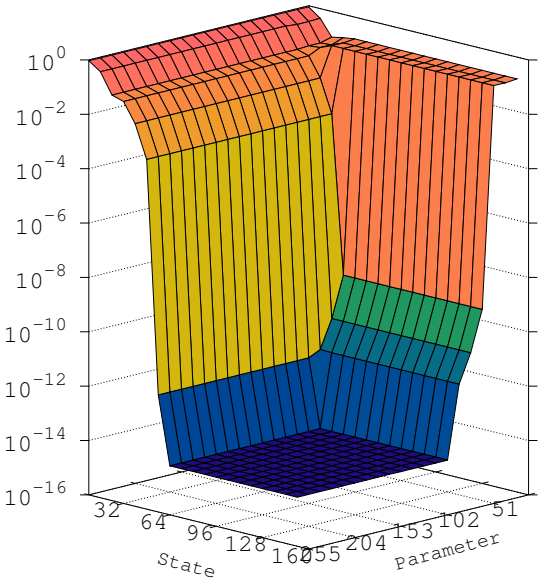
[84] See `code/ch7/eeg.m` in the supplementary source code archive referenced in Appendix B.1.



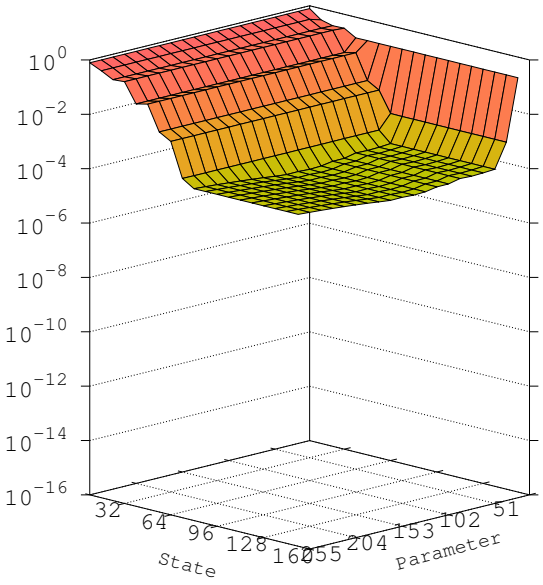
(a) *Relative $\ell_2 \otimes \ell_2$ output error for gramian-based combined reduction.*



(b) *Relative $\ell_2 \otimes \ell_2$ output error for optimization-based combined reduction.*

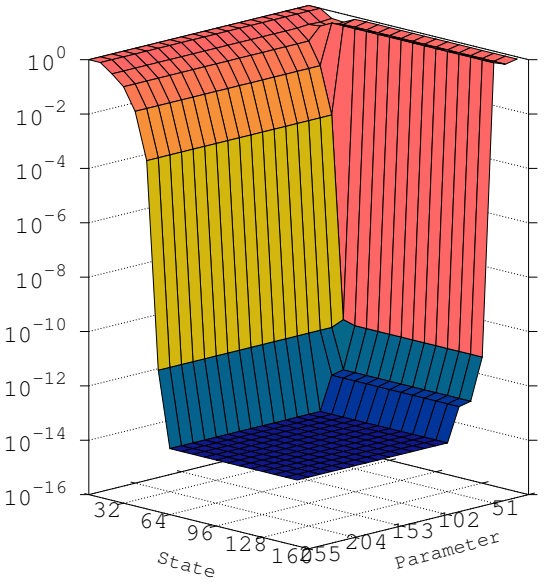


(c) *Relative $\ell_2 \otimes \ell_\infty$ output error for gramian-based combined reduction.*

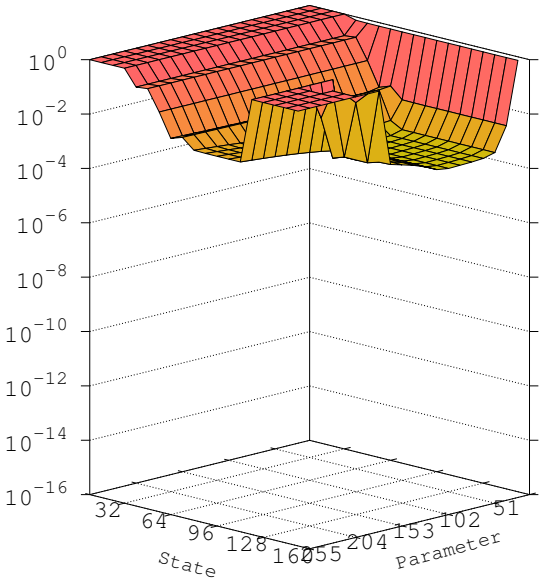


(d) *Relative $\ell_2 \otimes \ell_\infty$ output error for optimization-based combined reduction.*

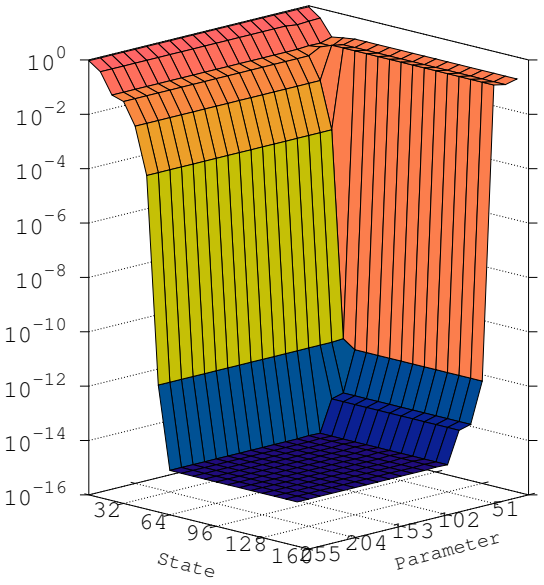
Figure 7.9.: *Numerical results for the combined state and parameter reduction of the* ***nonlinear*** *EEG & MEG dynamic causal model.*



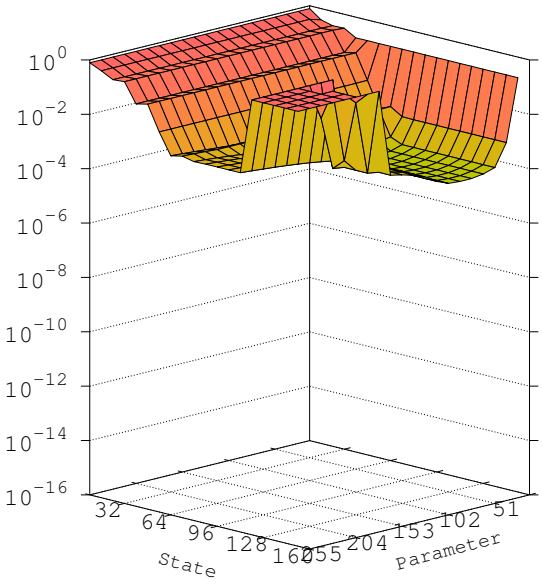
(a) *Relative $\ell_2 \otimes \ell_2$ output error for gramian-based combined reduction.*



(b) *Relative $\ell_2 \otimes \ell_2$ output error for optimization-based combined reduction.*

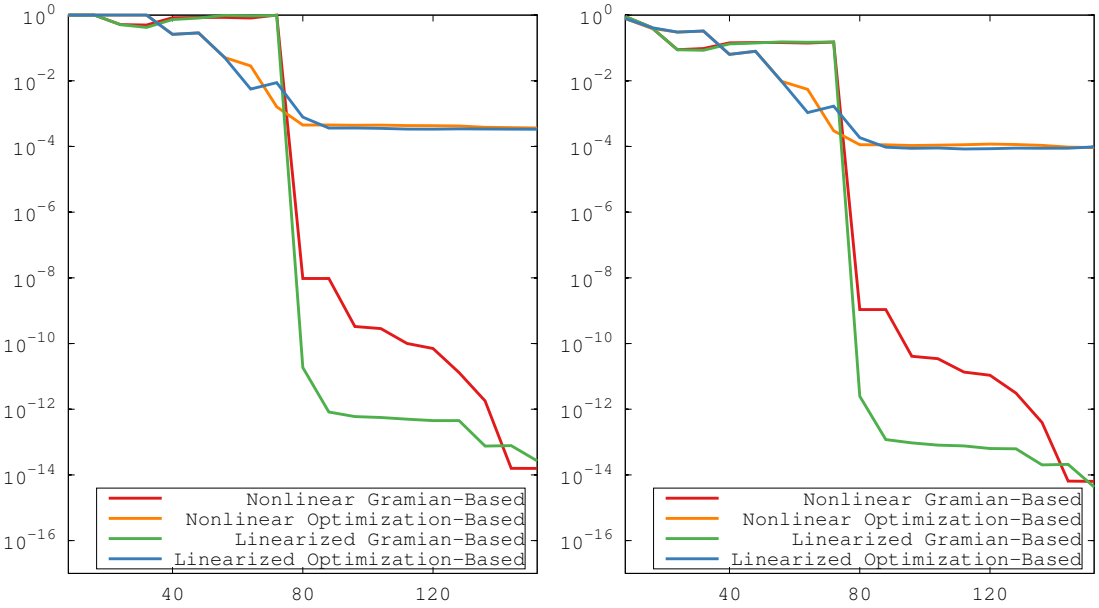


(c) *Relative $\ell_2 \otimes \ell_\infty$ output error for gramian-based combined reduction.*



(d) *Relative $\ell_2 \otimes \ell_\infty$ output error for optimization-based combined reduction.*

Figure 7.10.: *Numerical results for the combined state and parameter reduction of the* ***linearized*** *EEG & MEG dynamic causal model.*



(a) *Relative $\ell_2 \otimes \ell_2$ output error of gramian- and optimization-based combined reduction for the nonlinear and linearized model.*

(b) *Relative $\ell_2 \otimes \ell_\infty$ output error of gramian- and optimization-based combined reduction for the nonlinear and linearized model.*

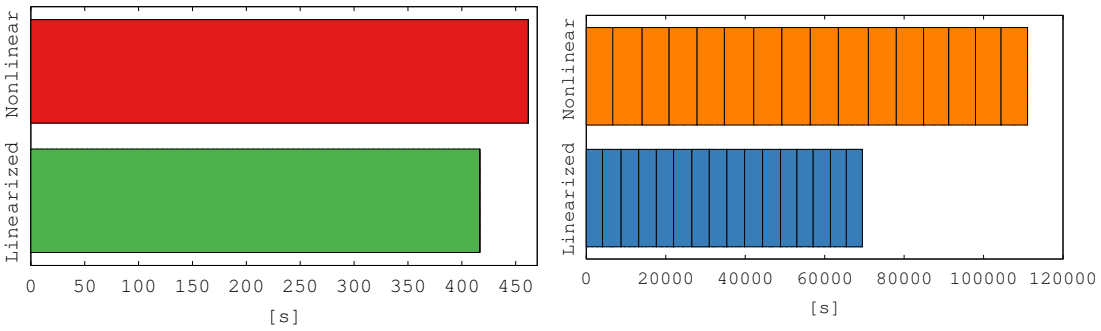
Figure 7.11.: *Relative output errors of the reduced order EEG & MEG dynamic causal models for equally reduced state-parameter-space dimensions $n = 2p \leq 160$.*

It should be noted, that due to the structure-preserving state-space reduction, each additional state base vector increases the ROM's state order by two.

For both methods, superficially, the error behavior of the nonlinear and linearized variants is similar. Yet, while the error of the optimization-based ROMs decays slowly, dominated by the parameter-space error, the gramian-based ROMs exhibit a steep decline in error. And as for the previous experiments, the respective $\ell_2 \otimes \ell_2$- and $\ell_2 \otimes \ell_\infty$-norms produce comparable error surfaces, except for the optimization-based approach using the linearized model which yields for $n \geq 120$ and $p \geq 144$ unstable ROMs.

A comparison of the combined reduction error of both methods[85] for concurrently reduced state- and parameter-space dimensions is diagrammed in Figure 7.11. Figure 7.11a and Figure 7.11b show the error in the $\ell_2 \otimes \ell_2$- and $\ell_2 \otimes \ell_\infty$-norm respectively. In case of the gramian-based approach, the ROMs derived from the nonlinear model reaches an accuracy of 10^{-8} for $n \geq 80$, $p \geq 40$ and of 10^{-10} for $n \geq 128$, $p \geq 64$, the linearized variant also achieves an accuracy of 10^{-10} for $n \geq 80$, $p \geq 40$ yet of 10^{-12} for $n \geq 88$, $p \geq 44$. The optimization-based approach in its nonlinear and linearized variant exhibits an accuracy of 10^{-3} for $n \geq 80$, $p \geq 40$, and the ROMs have very similar error decay. Comparing the respective ROMs obtained from the gramian-based and the optimization-based method, the ROMs of gramian-based approach are at more than four orders of magnitude lower in joint model reduction error for $n \geq 80$, $p \geq 40$.

[85] See also the additional figures in Appendix C.3.



(a) *Offline times for gramian-based combined reduction in seconds.*

(b) *Offline times for optimization-based combined reduction with highlighted per* 16 *iteration durations in seconds.*

Figure 7.12.: *Comparison of offline timings (note the different scales) for the combined reduction of the EEG & MEG dynamic causal model.*

7.4.7. Combined Reduction Performance

In Figure 7.12 the offline times for the nonlinear and the linearized model of the gramian-based ROM (Figure 7.12a) and the optimization-based ROM (Figure 7.12b) are compared. The gramian-based approach needs about 460 s using the nonlinear model and about 416 s (−10%) utilizing the linearized model. For $P = 256$ iterations, the optimization-based approach needs about 111000 s for the nonlinear model and 69500 s (−37%) for the linearized model. On average an iteration of the optimization-based method requires in case of the nonlinear model 433 s and 272 s (−37%) for the linearized model; thus the offline time of the gramian-based method takes less than the duration of two iterations of the optimization-based approach.
The linearization of the model has less impact on the offline time requirements in this experiment. This is due first, to the second-order structure of the model, which also effectively doubles the state-space dimension compared to the experiment in Section 7.3. Second, due to the use of a nonlinearity of a single type which additionally can be evaluated in vectorized form as in Section 7.2.

7.4.8. Assessment

The gramian-based method, more specifically the empirical joint gramian outperforms the optimization-based strategy in ROM accuracy and assembly time. In terms of the quality, this is in part due to the difference in the construction of the ROMs: the gramian-based methods select (or construct) only the position or velocity components necessary, which does not affect the empirical gramian computation prior to the direct truncation, yet for the optimization-based method inside each inner iteration a (non-structure-preserving) state-reduction is performed. The offline times are of the similar orders as for the previous fMRI & fNIRS experiment for the respective methods.

7.5. Combined Reduction for Inverse Problems

A last set of experiments explores combined state and parameter reduced models for the fMRI & fNIRS dynamic causal model in the context of inverse problems.

7.5.1. Reduced Inverse Problem

Independent from the specific method of neuroimaging, a general data model is assumed by a control-system-type model,

$$\dot{x}(t) = f(x(t), u(t), \theta),$$
$$y(t) = g(x(t), u(t), \theta),$$

wherein measured output y is assumed to be distorted by white noise. This is reflected in the data model by including an additive Gaussian noise component $\varepsilon = \mathcal{N}_{0,v}$ with zero mean,

$$y_d = y(\theta_d) + \varepsilon.$$

Given measurements y_d, which are postulated to obey this data model, an inverse problem is formed by the identification of a parameter θ_d, that minimizes the error compared to the observed data:

$$\theta_d \approx \arg\min_{\theta \in \Theta} \|y_d - y(\theta)\|.$$

Practically, the most likely parameter can be determined by a regularized least-squares optimization problem:

$$\theta_d \approx \arg\min_{\theta \in \Theta} \|y_d - y(\theta)\|_{\ell_2}^2 + \beta_2 \|\theta\|_2^2.$$

Instead of the FOM, a ROM obtained by combined reduction,

$$\dot{x}_r(t) = f_r(x_r(t), u(t), \theta_r),$$
$$y_r(t) = g_r(x_r(t), u(t), \theta_r),$$

can be used as underlying model. An adaption of the data model,

$$y_d \approx y_r(\theta_{d,r}) + \varepsilon,$$

leads to the reduced order inversion by an optimization problem utilizing the reduced state- and parameter-space,

$$\theta_{d,r} \approx \arg\min_{\theta_r \in \Theta_r} \|y_d - y_r(\theta_r)\|_{\ell_2}^2 + \beta_2 \|\theta_r\|_2^2. \tag{7.6}$$

Opposed to directly selecting an underlying model with less states or a smaller parameter-space, the ROM approximates the behavior of the FOM and thus the combined reduction has to be understood as a part of the inversion and not as part of the modelling.

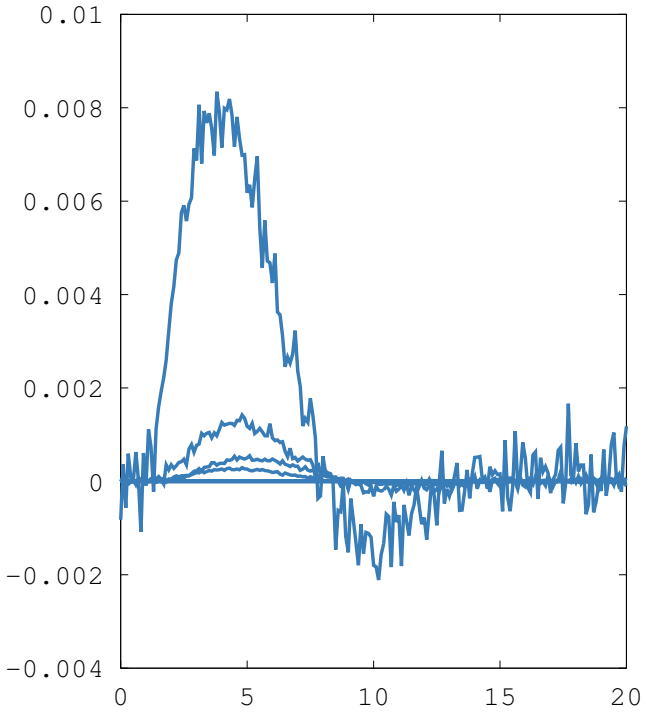


Figure 7.13.: *fMRI & fNIRS DCM synthetic data signal.*

7.5.2. Synthetic Signal Generation

The setup of the inverse problem experiment is analogous to the forward problem combined reduction experiments in Section 7.3. First, for a random but sparse connectivity pattern, a parameter θ_d is sampled from $\mathcal{N}_{1,1}$ for the elements mapping to the off-diagonal components and from $\mathcal{N}_{1,\frac{1}{100}}$ for the elements mapping the diagonal components. Then, a trajectory is simulated using either nonlinear dynamic causal model for this parameter to which white Gaussian noise for a signal-to-noise ratio (SNR) of 10 dB is added. These synthetic fMRI / fNIRS (Figure 7.13) signals act as data for the subsequent inverse problems.

7.5.3. Parameter Inference

For the fMRI & fNIRS dynamic causal model, the connectivity parameters are inferred[86] from the FOM and from the gramian-based and optimization-based ROMs in a Bayesian setting by computing[87] the MAP (6.13). The parameter inference is performed by an unconstrained[88] optimization algorithm with Tikhonov regularization with $\beta_2 = 10^{-4}$. As a start value for the optimization the expectation of the prior parameter is used.

[86] Originally in [77], an expectation-maximization algorithm is used for the parameter inference.
[87] See `code/ch7/bayinv.m` in the supplementary source code archive referenced in Appendix B.1.
[88] Here, "unconstrained" refers to hard constraints.

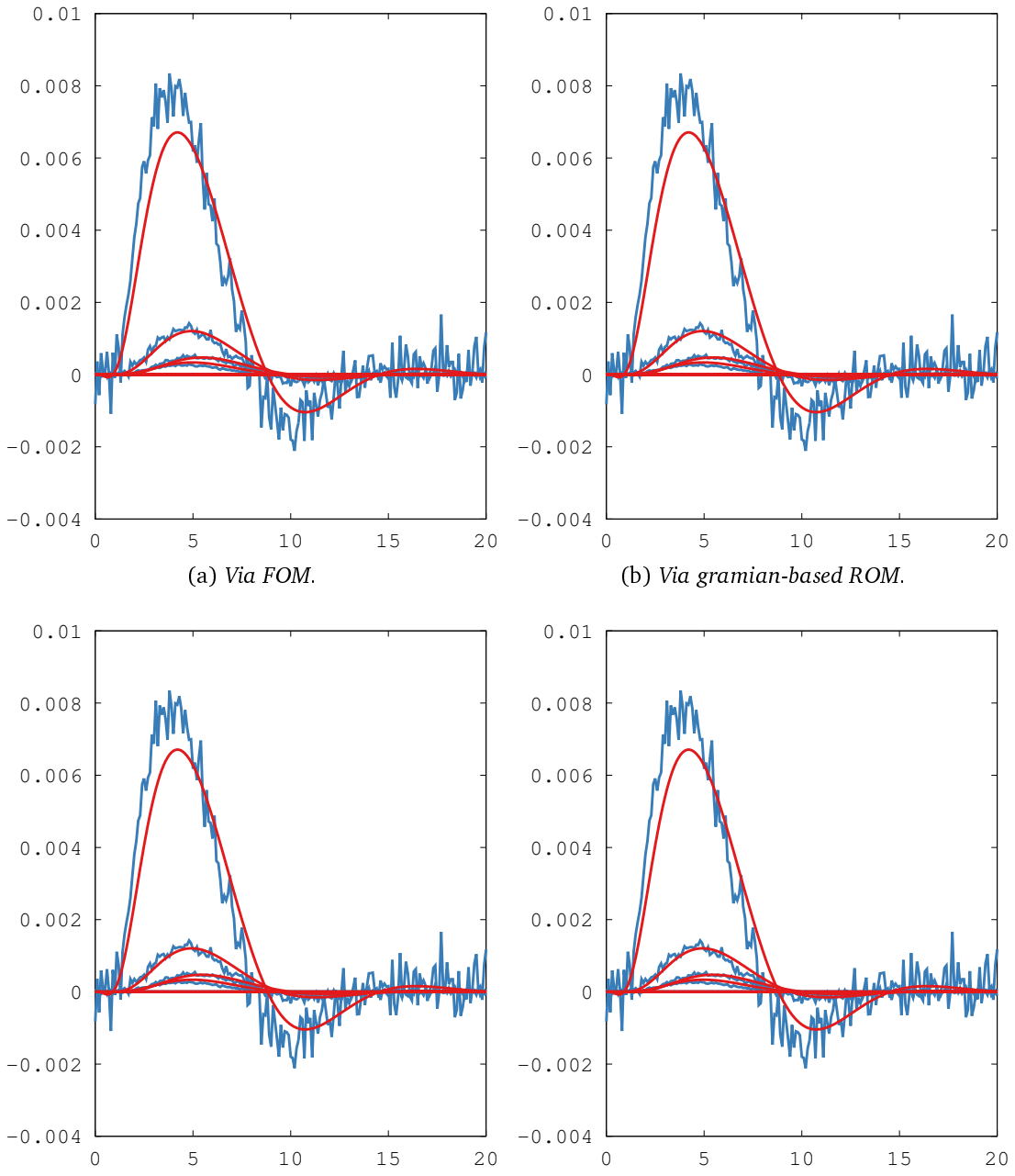
7.5.4. fMRI & fNIRS Dynamic Causal Model

For the fMRI & fNIRS DCM a parameter inference is performed using the FOM, a gramian-based ROM following Section 7.3, an optimization-based ROM also following Section 7.3 and an optimization-based ROM additionally utilizing the data-driven (DD) regularization enhancement from Section 4.4. The ROMs are assembled using the nonlinear fMRI & fNIRS DCM for the gramian-based and optimization-based approach, since for either combined reduction method those yield the best results. Based on the results from Section 7.3, the reduced order for the (nonlinear) gramian-based method is set for state- and parameter-space to $n = p = 40$, and for the (nonlinear) optimization-based method to $n = p = 24$. For the data-driven regularized optimization-based variant, the associated regularization coefficient is set to $\beta_d = 10^{-4}$.
The results of the parameter inference for the fMRI & fNIRS DCM using the gramian-based and optimization-based ROMs are summarized in Table 7.14, which shows that the statistical parameter estimation for the FOM and ROMs is of similar quality. For the inference of the FOM and the ROMs, the output error $\|y_d - y(\theta)\|_{\ell_2}$ using the inferred parameter θ is of the order 10^{-2}. Even though the output error is in an acceptable range, the relative parameter error $\|\theta_d - \theta\|_2$ is at 25% and thus does not retrieve the parameter θ_d, the data was generated with. Yet, this examples was not chosen to demonstrate its absolute accuracy but the performance improvement considering the relative accuracy of the ROMs compared to the FOM. In terms of online timings the gramian-based ROM needs 34% less time for the parameter inference compared to the FOM, the optimization-based ROM and the data-driven regularized optimization-based ROM merely 16% of the FOM online time. For an online phase the gramian-based approach consumes 60% total time (offline + online time) in relation to the FOM. The data-driven regularized variant shortens the offline phase by 19% for the optimization-based method.

fMRI & fNIRS DCM	FOM	Gramian-Based ROM	Optim.-Based ROM	Optim.-Based ROM (DD)
Parameter Dim.	256	40	24	24
Output Error	0.01	0.01	0.01	0.01
Parameter Error	0.25	0.25	0.25	0.25
Offline Time	-	381 s	11957 s	9774 s
Online Time	1447 s	488 s	233 s	226 s
Single Total Time	1447 s	869 s	12190 s	10000 s
Multi Breakeven	-	1	10	8

Table 7.14.: *Inference performance for the fMRI & fNIRS DCM FOM and ROM.*

Furthermore, the plots in Figure 7.15 show the synthetic data and the FOM outputs for the inferred parameters. In Figure 7.15a the output associated to the parameter obtained by inference using the FOM is depicted, while Figure 7.15b shows the output for the parameter inference based on the gramian-based ROM. Figure 7.15c and Figure 7.15d present the output for the parameters inferred by utilization of the basic and the data-driven regularized optimization-based ROM.



(a) *Via FOM.*

(b) *Via gramian-based ROM.*

(c) *Via optimization-based ROM.*

(d) *Via optimization-based ROM (DD).*

Figure 7.15.: *Comparison of synthetic output model data with fMRI & fNIRS dynamic causal ROM outputs using the inferred parameter.*

8. Conclusion

Contents

Following, a comparison and evaluation of the combined reduction methods, with respect to the previous numerical results, is conducted as well as a layout of prospective advancements for the presented techniques.

8.1. Summary

This work investigated the combined state- and parameter-space reduction for nonlinear systems. Two classes of methods for combined reduction have been explored. First, a gramian-based approach (Chapter 3), with special focus on the empirical (non-symmetric) cross gramian and the related empirical joint gramian inducing the empirical cross-identifiability gramian. Second, an optimization-based approach (Chapter 4), connecting a greedy parameter reduction with POD-related state reduction. While the gramian-based methods originate in (linear) system theory, the optimization-based methods root in inverse problem solution. An implementation of the empirical-gramian-based method and of the optimization-based method was described and tested in Chapter 5. And an application from the field of systems neuroscience, aimed at connectivity inference for neuronal networks based on dynamic causal models, for fMRI & fNIRS and EEG & MEG neuroimaging techniques, was detailed in Chapter 6 and evaluated in Chapter 7 alongside with the related generic hyperbolic network model.
Overall, both methods are capable of producing reduced order models of sufficient quality. The experiments in Chapter 7 indicate that it depends on the model which method provides better results, but the (empirical-cross-)gramian-based method seems to obtain more accurate results in general. Furthermore, the optimization-based method takes significantly longer (using the default inner optimizers) to compute the ROMs. For the considered data-driven combined reduction methods it was also observed, that the use of the original nonlinear model for the generation of snapshots, instead of a linearization, can improve the ROM.

8.2. Abstract Comparison

Both methods for the combined state and parameter reduction of nonlinear systems rely on simulated trajectories from which the information, encoding the system's dynamics, is extracted. Exemplary for the gramian-based approach, the cross-gramian-based joint gramian (Section 3.4.3) constructs the state-space reducing projection from the input-output relations, while the parameter-space reducing projection is obtained from state-to-output effects. The enhanced optimization-based approach (Section 4.4.1) uses input-to-state information for the state-space reducing projection and input-output relations for the parameter-space reducing projection (see Table 8.1).

	Cross-Gramian-Based	Optimization-Based
State Reduction	Input-Output	Input-State
Parameter Reduction	State-Output	Input-Output

Table 8.1.: *Comparison of combined reduction methods by subspace constructions.*

Both methods could be extended to assemble the reducing projection based on input-output behavior for state and parameter reduction. For the cross-gramian-based approach using the joint gramian to acquire the parameter reducing projection, this would entail to include as many inputs as parameters which was initially proposed for the joint gramian in [109]. Yet, for high-dimensional parameter-spaces a joint gramian of dimension $N+P \times N+P$ would have to be stored instead of $N+P \times N$ and also additional trajectories of order P would be required, since each parameter would have to be perturbated separately by an input. In case of the optimization-based approach the state reducing projection would have to be augmented using the goal-oriented approach from [229], which means a second optimization problem would have to be solved in each iteration for a general nonlinear model. Hence, fully input-output derived ROMs would result in higher computational loads for both approaches.
A central difference lies in the conceptual assembly of the ROM. For the gramian-based method one or more empirical gramians of the order of the full order system are computed from which the reducing projections are extracted. In case of the empirical joint gramian this means a memory bound is given by the gramian matrix dimension of $N \times N + P$. This is not the case for the optimization-based method, since it directly assembles a reducing projection of desired order increasingly, but this approach is bound by the compute capabilities if no derivative information is supplied to the inner optimizer. In terms of the validity of the ROM, for the empirical-gramian-based method is valid for the specified operating region defined (manually) through the perturbation sets, while for the optimization-based approach the model is valid for the adaptively scanned state- and parameter-space.
Lastly, the two methods are differing in their precedence of states and parameters during the combined reduction. In the optimization-based combined reduction, the reduced parameter-space determines the state-space sampling points; for the empirical joint gramian the parameters are treated as (constant) states.

From an implementation point of view, the empirical gramian method requires mere basic linear algebra operations; the optimization-based approach requires the eponymous optimization algorithm for the greedy search that is principal to the overall performance. Yet, the solver to obtain the required trajectories is crucial to both methods. While the accuracy of empirical gramian derived ROMs can be improved with more detailed knowledge on the targeted operating region of the model, the optimization-based method can be accelerated by derivative information on the associated residuals.
Due to the simplification in the empirical gramian framework of perturbing a single component of input, initial state or parameter component, an assessment of the presented gramian-based and optimization-based combined reduction methods can also be seen as a comparison between sparse and greedy sampling.
Following, the computational complexity is summarized in terms of number of needed FOM trajectories. The computation of the trajectories consumes the dominant fraction of overall offline time in the tested settings. And while the remaining operations are (parallelizable) matrix operations[89], for nonlinear models inherently sequential methods like Runge-Kutta integrators are usually utilized. For the empirical joint gramian, assuming a single perturbation element for inputs, states and parameters, as many trajectories as the sum of inputs, states and parameters have to be computed, which amounts to

$$O_{W_J} = J + N + P \tag{8.1}$$

trajectories. In a single iteration of the optimization-based method, assuming a Gauss-Newton inner optimization with a first-order finite-difference approximation, $P+1$ trajectories are needed for the approximate Jacobian plus an additional trajectory for the state-space projection. Thus for p iterations, altogether

$$O_{\text{opt}} = (p-1) \times (P+1) + 1 \tag{8.2}$$

trajectories are required, since the initial iteration requires only a single trajectory as it uses the provided nominal or prior parameter as parameter base component and no greedy sampling is performed.

A Note on Frequency-Domain Validity

The work at hand considers only time-domain and not frequency-domain behavior of the ROMs. This is especially relevant for the empirical gramian approach due to its origin in system theory and availability of an $\mathscr{H}_\infty$-norm error indicator. But, since for parametric systems the state-space validity is ensured essentially by averaging gramians for different sampling points in the parameter-space, the ROMs from empirical gramians loose accuracy rapidly in frequency-space error measures. Such behavior can be observed exemplarily for the empirical linear cross gramian in [18]. Due to the sole consideration of state-space accuracy of the presented combined reduction methods, a worse performance in frequency-domain error measures is predicted. If frequency-space accuracy is targeted, for example the Loewner framework [128], based on frequency- instead of state-space samples, is better suited for frequency response preserving ROMs.

[89] Such as matrix multiplication and matrix decompositions.

8.3. Numerical Comparison

In Chapter 7 the conducted experiments are evaluated separately for each model to assess the quality for the tested combined reduction methods. Following, all experiments are jointly compared in terms of numerical efficiency. For the purpose at hand, efficiency is considered offline time cost in relation to the joint error the associated ROM exhibits. To illustrate this efficiency concept, Figure 8.2 shows the offline time consumption correlated with the respective achieved error for all experiments from Chapter 7, associated with the gramian-based or optimization-based approach.

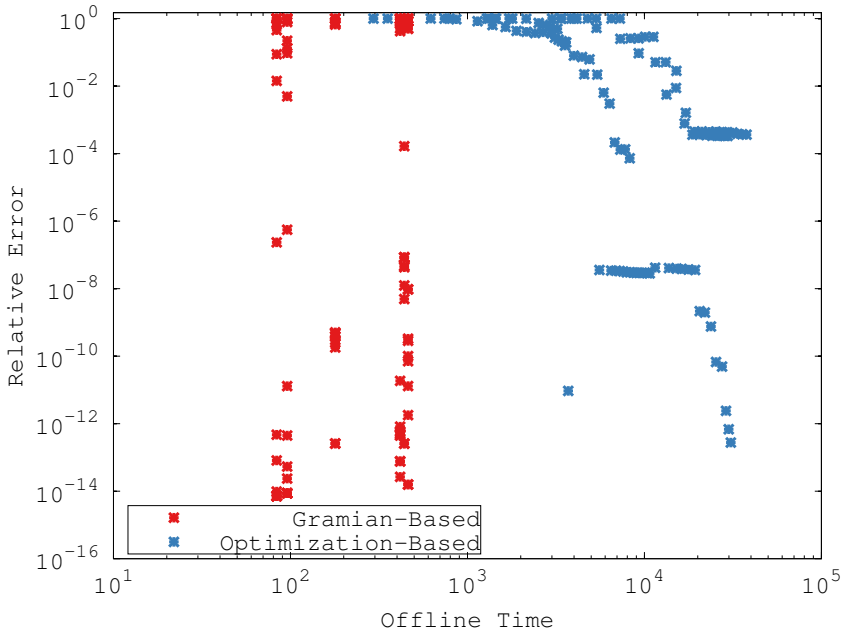


Figure 8.2.: *Efficiency comparison by gramian-based and optimization-based time consumption versus achieved joint error in the $\ell_2 \otimes \ell_2$-norm for $n = p < N$.*

The efficiency plot in Figure 8.2 shows the constant time requirement of the gramian-based experiments for all accuracies. This is due to the full order SVDs which are applied to the empirical cross gramian and the empirical cross-identifiability gramian. It should also be noted, that for the experiments involving SIMO systems, the empirical non-symmetric joint gramian is used. To models with a large state-space dimension a TSVD can be applied which reduces the offline time depending on the reduced order selected manually or based on the singular values. For the optimization-based experiments, in which the inner optimizations approximate the derivative information numerically, longer offline times are required to reach the same error levels as the gramian-based experiments. It is also shown, that little complexity is added by reaching lower errors once a certain accuracy is reached. Additionally, for each optimization-based experiment suitable regularization coefficients have to be determined. This is often requires multiple, at least partial, ROM computations.

8.4. Outlook

In a brief outline, the prospective enhancements to the presented methods and models are described. On the basis of the neuorscientific application in Chapter 6 a generalization to (complex) networks or towards network discovery [194] is conceivable. For the featured methods of combined reduction various further developments from an algorithmic as well as from a technical perspective are surmisable.

8.4.1. Further Development

In further investigations of the model reduction techniques devised in this work two complementary paths are pursuable. First, algorithmic enhancements can improve accuracy of the ROM; second, the use of parallel computation can increase the performance.

Methodical Enhancements

For nonlinear systems the presented methods pose the disadvantage that the high-dimensional vector field and output functional have to be evaluated after lifting the reduced quantities. This lifting bottleneck can be overcome by a hyper-reduction using the empirical interpolation method (EIM) [11] or the discrete empirical interpolation method (DEIM) [37]. An example for combining DEIM with gramian-based model reduction can be found in [162] and a combination of optimization-based model reduction with EIM in [80].
Specifically for the empirical-gramian-based model reduction, an extension using the concepts of reproducing kernel Hilbert spaces (RKHS) [27] related to the KPCA is conceivable. Furthermore, the cross-gramian-based methods could be improved by stability-preserving two-sided projections, for which a starting point could be [95].
In case of the optimization-based approach, the use of L_1-norm regularization [34] instead or additionally to the Tikhonov regularization could improve the accuracy. Also an automated selection of the regularization weights, for example by the L-curve procedure [103] may improve the interaction of the various terms in the optimization function.

Explicit Parallelization

The presented implementations are extensively vectorized to promote implicit parallelization by the utilized BLAS backends. Explicit parallelization is also applicable, which allows the computation of multiple trajectories in parallel[90] by a sequential general linear method on a multi-core CPU (Central Processing Unit).
All empirical gramians could also profit from heterogeneous compute systems. The actual gramian matrix can be assembled by a massively parallel matrix multiplication[91] on a GPGPU (General Purpose Graphics Processing Unit). In case the GPU (Graphics Processing Unit) is integrated, additionally zero-copy functionality may be exploited to prevent copies of the discrete trajectory data between the CPU and GPU memory spaces.

[90] The corresponding `for`-loops are tagged with the keyword `parfor` in a comment.
[91] The relevant products are tagged with the keyword `offload` in a comment.

A distributed memory parallelization of the empirical cross gramian, and thus also for the empirical joint gramian and non-symmetric cross gramian, is especially promising. The empirical cross gramian and its variants can be assembled column-wise, which is illustrated by a simplified empirical cross gramian w_X:

$$w_X = \sum_{j=1}^{J} \int_0^\infty \Psi^j(t)\,\mathrm{d}t \in \mathbb{R}^{N\times N}, \quad \Psi^j_{ab}(t) := \langle x^j_a(t), y^b_j(t)\rangle,$$
$$\Rightarrow w_{X,*i} = \sum_{j=1}^{J} \int_0^\infty \psi^{ji}(t)\,\mathrm{d}t \in \mathbb{R}^{N\times 1}, \quad \psi^{ji}_a(t) := \langle x^j_a(t), y^i_j(t)\rangle,$$

and together with the use of a parallel technique for large (truncated) SVDs from [207, 208], a low communication parallel variant may be computed.
A parallelization of the optimization-based combined reduction algorithm would have to focus on the inner optimization method utilized for the greedy sampling. For example the computation of the Jacobian can be accelerated by parallelly generate parameter perturbed trajectories.

8.4.2. Future Applications

A future application for the presented methods are **complex networks**, which establish a generic framework for large networked systems enabling applications beyond connectivity in the brain, for example network cosmology [139, 26] or digital social networks. Complex networks are based on random graphs (see for example [54]) and exhibit, opposed to Erdős-Rényi networks, topological features like small world behavior due to a scale-free degree distribution. With the recent adoption of system-theoretic methods for network analysis, like network controllability (structural controllability) [156] and network observability (structural observability) [157] the system identification and model reduction methods in this work present feasible tools for the study of complex networks and associated metrics [216]. And beyond linear systems, also nonlinear networks are analyzed utilizing controllability and observability properties [227].
Such an analysis requires further investigation of parametrized time-varying systems [44],

$$\dot{x}(t) = A(\theta(t))x(t) + Bu(t),$$
$$y(t) = Cx(t),$$

with the special case of time-varying parameters [111]. An associated nonlinear extension could be given by the hyperbolic network model from Section 6.1.1:

$$\dot{x}(t) = A(\theta(t))\tanh(K(\theta(t))x(t)) + Bu(t),$$
$$y(t) = Cx(t).$$

The cross-gramian-based empirical (non-symmetric) joint gramian could lead to insights into small surrogate networks (model reduction) or driving input-output network components (system identification). The optimization-based approach may be used for the discovery of likely network configurations from given output data (uncertainty quantification).

8.5. Concluding Remarks

The two demonstrated MOR methods allow a combined state and parameter reduction for input-output systems, including parametric nonlinear systems. Both, the empirical-gramian-based and the greedy-optimization-based approach, rely mainly on simulated trajectories for the assembly of the reducing projections. Hence, if a numerical solution is computable, either method is applicable.
While both approaches yield workable reduced order models, the empirical (cross) gramian method produces more accurate reduced models in significantly less computational time than the optimization-based greedy sampling, which has been demonstrated numerically for different types of linear and nonlinear systems.
In the investigation of networks in terms of connectivity reconstruction for neuronal networks, both methods have been demonstrated to provide suitable reduced order models. Thus, for mathematical engineering or acceleration of research-software the presented model reduction methods can be of vital use.
Lastly, in future research on nonlinear model order reduction and combined reduction, the optimization-based, but especially the empirical-gramian-based techniques enable further exploration of reduced order modelling based on the coherence between system inputs and outputs.

Appendix

Contents

A. Mathematical Tools

A.1. Young's Inequality

A variant of Young's inequality holds for convolutions [232]. Given functions $f \in \mathrm{L}_p$ and $g \in \mathrm{L}_q$ then the following holds:

$$\|f * g\|_s \leq \|f\|_p \|g\|_q,$$

for a Lebesgue norm $\|\cdot\|_s$ with an s satisfying:

$$\frac{1}{p} + \frac{1}{q} = 1 + \frac{1}{s}.$$

A.2. Vectorization Operator

The vectorization operator vec maps a matrix $A \in \mathbb{R}^{N \times N}$ to a vector $a \in \mathbb{R}^{N^2}$ by stacking the columns of $A = \begin{pmatrix} a_1 & \dots & a_N \end{pmatrix}$,

$$\mathrm{vec}(A) := \begin{pmatrix} a_1 \\ \vdots \\ a_N \end{pmatrix}.$$

This operator is isomorphic and its inverse is defined by its composition with the vectorization operator:

$$\mathrm{vec}^{-1}(\mathrm{vec}(A)) = A.$$

A.3. Trace Norm

The trace norm of a matrix is defined as the sum of all singular values,

$$\|A\|_* := \sum_{i=1}^{N} \sigma_i(A).$$

The trace norm is an unitarily invariant norm and also known as the nuclear norm, Schatten-1 norm and Ky-Fan-N norm.

A.4. Approximate Matrix Inversion

The inverse to a regular matrix A can be computed using the Neumann series.

Lemma 9.1 (Matrix Inverse by Neumann Series)
For a regular matrix $A \in \mathbb{R}^{N\times N}$ its inverse is given by:

$$A^{-1} = \sum_{k=0}^{\infty} (\mathbb{1} - A)^k.$$

A truncated Neumann series can be used as an approximation to the matrix inverse. Following [230, Sec. B], utilizing an additive matrix decomposition, a coarse, yet computationally very cheap approximate matrix inverse can be computed as follows.

Corollary 9.2 (Approximate Inverse)
For a regular matrix A and an additive decomposition $A = D + E$ into a diagonal matrix D, $D_{ii} = A_{ii}$ and $E = A - D$, an approximation to A^{-1} is given by:

$$A^{-1} \approx D^{-1} - D^{-1}ED^{-1}.$$

The computation of this approximate inverse matrix has a complexity of $\mathcal{O}(N^2)$.

A.5. Hankel Singular Values of a Single Region Hemodynamic Forward Model

For the hemodynamic forward model from Section 6.3.2 of a single region :

$$\dot{z}(t) := \begin{pmatrix} \dot{s}(t) \\ \dot{f}(t) \\ \dot{v}(t) \\ \dot{q}(t) \end{pmatrix} = \begin{pmatrix} u(t) - \kappa s(t) - \gamma(f(t) - 1) \\ s(t) \\ \frac{1}{\tau}(f(t) - v_i(t)^{\frac{1}{\alpha}}) \\ \frac{1}{\tau}(\frac{1}{\rho} f(t)(1 - (1-\rho)^{\frac{1}{f(t)}}) - v(t)^{\frac{1}{\alpha}} \frac{q(t)}{v(t)}) \end{pmatrix},$$
$$y(t) = V_0(k_1(1 - q(t)) + k_2(1 - v(t))),$$

with $u(t) = \delta(t)$, the associated (approximate) Hankel singular values of this system, obtained by an empirical cross gramian, compute[92] as:

$$\sigma_{\text{hemodynamic}} = \{1.2 \cdot 10^{-1}, 1.8 \cdot 10^{-2}, 2.2 \cdot 10^{-3}, 2.1 \cdot 10^{-4}\}.$$

[92] See `code/app/mri_siso.m` in the supplementary source code archive referenced in Section B.1.

B. Software Annex

B.1. Code Availability

The source code of the implementations used to compute the presented results can be obtained from:

doi:10.5281/zenodo.48122

and is authored by CHRISTIAN HIMPE.

B.2. Computational Environment

Hardware

The following computer system was used to compute the numerical results:

CPU

Model AMD A10-7800[93] (x86-64)

Topology 1 Socket - 2 Modules - 4 Cores (4 Threads)

Clock 3.5 Ghz (Dynamic Frequency Scaling Disabled)

L1 Cache 2 × 96 KB (Instruction), 4 × 16 KB (Data)

L2 Cache 2 × 2 MB (Last Level Cache)

SIMD AVX, FMA3, FMA4 (4 × 64-bit Double Precision per module)

RAM

Type Dual Rank DDR3 (Dual Channel Configuration)

Amount 32 GB (4 × 8GB)

Rate 2133 MT/s (max. 17000 MB/s)

Software

The following software stack was used to compute the numerical results:

Matlab Runtime GNU Octave 4.0.0 (Compiled[94] with gcc 5.2.1)

Linked tcmalloc_minimal[95] (Thread-Caching Memory Allocation)

Linked FlexiBLAS 1.3 (BLAS wrapper with runtime exchangeable backends [137, 138])

BLAS & LAPACK AMD Core Math Library (ACML) 6.1.0.31

C Library GNU libc 2.21

Operating System Lubuntu 15.10 (64-bit Linux Kernel 4.2)

[93] This APU consists of a CPU and a GPU unit; the latter is not utilized for the computations.

[94] Main compilation flags: `-Os -mtune=native -march=native -m64 -mfpmath=sse -malign-double`

[95] `tcmalloc` is part of the `gperftools` (2.4) by GOOGLE

B.3. emgr - Empirical Gramian Framework

Code Metadata

name (short name)	Empirical Gramian Framework (emgr)
version (release date)	3.9 (2016-02-25)
topic (type)	Model reduction (Toolbox)
author (ORCID)	Christian Himpe (0000-0003-2194-6754)
license (type)	2-Clause BSD (open-source)
repository (type)	`github.com/gramian/emgr` (git)
language	Matlab
dependencies (system)	Octave | Matlab (Linux, Windows)
website	`gramian.de`
keywords	empirical gramians, MOR

License

Source Code

```
function W = emgr(f,g,s,t,w,pr,nf,ut,us,xs,um,xm)
% emgr - Empirical Gramian Framework ( Version: 3.9 )
% Copyright (c) 2013-2016 Christian Himpe ( gramian.de )
% released under BSD 2-Clause License ( opensource.org/licenses/BSD-2-Clause )
%
% SYNTAX:
%    W = emgr(f,g,s,t,w,[pr],[nf],[ut],[us],[xs],[um],[xm]);
%
% SUMMARY:
%    emgr - EMpirical GRamian framemwork,
%    computation of empirical gramians for model reduction,
%    system identification and uncertainty quantification.
%    Enables gramian-based nonlinear model order reduction.
%    Compatible with OCTAVE and MATLAB.
%
% ARGUMENTS:
%   (func handle)  f - system function handle; signature: xdot = f(x,u,p)
%   (func handle)  g - output function handle; signature:    y = g(x,u,p)
%        (vector)  s - system dimensions [inputs,states,outputs]
%        (vector)  t - time discretization [step,stop]
%          (char)  w - gramian type:
%              * 'c' : empirical controllability gramian (WC)
%              * 'o' : empirical observability gramian (WO)
%              * 'x' : empirical cross gramian (WX)
%              * 'y' : empirical linear cross gramian (WY)
%              * 's' : empirical sensitivity gramian (WS)
%              * 'i' : empirical identifiability gramian (WI)
%              * 'j' : empirical joint gramian (WJ)
% (matrix,vector,scalar) [pr = 0] - parameters, each column is one set
%        (vector,scalar) [nf = 0] - options, 12 components:
%              + zero(0),init(1),steady(2),mean(3),median(4),midr(5),rms(6) center
%              + linear(0), log(1), geom(2), single(3), sparse(4) input scales
%              + linear(0), log(1), geom(2), single(3), sparse(4) state scales
%              + unit(0), reciproce(1), dyadic(2), single(3) input rotations
%              + unit(0), reciproce(1), dyadic(2), single(3) state rotations
%              + single(0), double(1), scaled(2) run
%              + regular(0), non-symmetric(1) cross gramian; only: WX, WJ
%              + plain(0), robust(1) parameters; only: WC, WY
%              + active(0), passive(1) parameter; only: WI, WJ
%              + none(0), linear(1), logarithmic(2) parameter centering
%              + default(0), exclusive options:
%                    * use rms-centering(1); only: WS
%                    * use Schur-complement(1); only: WI
%                    * use detailed Schur-complement(1); only: WJ
%              + assume(0), enforce(1) gramian symmetry
%  (matrix,vector,scalar) [ut = 1] - input; default: delta impulse
%         (vector,scalar) [us = 0] - steady-state input
%         (vector,scalar) [xs = 0] - steady-state and initial state x0
%  (matrix,vector,scalar) [um = 1] - input scales
%  (matrix,vector,scalar) [xm = 1] - initial-state scales
%
% RETURNS:
%             (matrix)  W - Gramian Matrix (only: WC, WO, WX, WY)
%               (cell)  W - {State-,Parameter-} Gramian (only: WS, WI, WJ)
%
% CITATION:
%    C. Himpe (2016). emgr - Empirical Gramian Framework (Version 3.9)
%    [Software]. Available from http://gramian.de . doi:10.5281/zenodo.46523 .
%
% SEE ALSO:
%    gram
%
% KEYWORDS:
%    model reduction, empirical gramian, cross gramian, mor
%
% Further information: <http://gramian.de>
%*

    % Custom ODE Solver
    global ODE;
    if(isa(ODE,'function_handle')==0), ODE = @rk2; end;
```

```
    % Version Info
    if( (nargin==1) && strcmp(f,'version') ), W = 3.9; return; end;

    % Default Arguments
    if( (nargin<6)  || isempty(pr) ), pr = 0.0; end;
    if( (nargin<7)  || isempty(nf) ), nf = 0;   end;
    if( (nargin<8)  || isempty(ut) ), ut = 1.0; end;
    if( (nargin<9)  || isempty(us) ), us = 0.0; end;
    if( (nargin<10) || isempty(xs) ), xs = 0.0; end;
    if( (nargin<11) || isempty(um) ), um = 1.0; end;
    if( (nargin<12) || isempty(xm) ), xm = 1.0; end;

    % System Dimensions
    J = s(1);               % number of inputs
    N = s(2);               % number of states
    O = s(3);               % number of outputs
    M = 0;                  % internal variable used by WS, WI, WJ
    if(numel(s)==4)
        M = s(4);
    end;

    h = t(1);               % width of time step
    T = floor(t(2)/h) + 1;  % number of time steps plus initial value

    w = lower(w);           % ensure lower case gramian type

    P = size(pr,1);         % number of parameters
    Q = size(pr,2);         % number of parameter sets

    % Linear Chirp Input
    if( isnumeric(ut) && numel(ut)==1 && ut==Inf )
        ut = @(t) 0.5*cos(pi*(t+10*t.*t))+0.5;
    end;

    % Discretize Procedural Input
    if(isa(ut,'function_handle'))
        uf = ut;
        ut = zeros(J,T);
        for l=1:T
            ut(:,l) = uf(l*h);
        end;
    end;

    % Lazy Arguments
    if( isnumeric(g) && g==1 ), g = @(x,u,p) x; O = N; end;

    if(numel(nf)<12), nf(12)    = 0;  end;
    if(numel(ut)==1), ut(1:J,1) = (1.0/h)*ut; end;
    if(numel(us)==1), us(1:J,1) = us; end;
    if(numel(xs)==1), xs(1:N,1) = xs; end;
    if(numel(um)==1), um(1:J,1) = um; end;
    if(numel(xm)==1), xm(1:N+(w=='y')*(J-N),1) = xm; end;

    if(size(ut,2)==1), ut(:,2:T) = 0.0; end;
    if(size(us,2)==1), us = repmat(us,[1,T]); end;
    if(size(um,2)==1), um = scales(um,nf(2),nf(4)); end;
    if(size(xm,2)==1), xm = scales(xm,nf(3),nf(5)); end;

%% PARAMETRIC SETUP

    if( (nf(8) && w~='o') || w=='s' || w=='i' || w=='j' )

        if(Q==1), error('ERROR! emgr: min and max parameter required!'); end;

        pmin = min(pr,[],2);
        pmax = max(pr,[],2);

        if( nf(8) || w=='s' ) % assemble (controllability) parameter scales
            pn = size(um,2);
        else                  % assemble (observability) parameter scales
            pn = size(xm,2);
        end;

        pl = (1.0:floor(pn/2))./floor(pn/2);
```

```
        pu = (1.0:ceil(pn/2))./ceil(pn/2);

        switch(nf(10)) % parameter centering

            case 1, % linear
                pr = mean(pr,2);
                pm = [(pmin - pr)*pl , (pmax - pr)*pu];

            case 2, % logarithmic
                lpmin = log(pmin);
                lpmax = log(pmax);
                lpavg = 0.5*(lpmax - lpmin);
                pr = pmin.*exp(lpavg);
                pm = [bsxfun(@times,exp((lpavg - lpmin)*pl),pmin), ...
                      bsxfun(@times,exp((lpmax - lpavg)*pu),pr) ];
                pm = bsxfun(@minus,pm,pr);

            otherwise, % none
                pr = pmin;
                pm = (pmax - pmin)*((1:pn)./pn);
        end;

        Q = 1;
    end;

%% STATE-SPACE SETUP

    if( w=='c' || w=='o' || w=='x' || w=='y' )

        C = size(um,2); % number of input scales
        D = size(xm,2); % number of state scales

        switch(nf(1)) % residual types

            case 1, % initial state
                res = @(d) d(:,1);

            case 2, % steady state
                res = @(d) d(:,end);

            case 3, % mean state
                res = @(d) mean(d,2);

            case 4, % median state
                res = @(d) median(d,2);

            case 5, % midrange
                res = @(d) 0.5*(min(d,2)+max(d,2));

            case 6, % rms
                res = @(d) sqrt(sum(d.*d,2));

            otherwise, % zero state
                res = @(d) zeros(size(d,1),1);
        end;

        switch(nf(6)) % scaled runs

            case 1, % preconditioned run
                nf(6) = 0;
                WT = emgr(f,g,s,t,w,pr,nf,ut,us,xs,um,xm);
                TX = sqrt(diag(WT));
                TX = TX(1:(N-(M>0 && w~='c')*P));
                tx = 1.0./TX;
                F = f; f = @(x,u,p) TX.*F(tx.*x,u,p);
                G = g; g = @(x,u,p)     G(tx.*x,u,p);

            case 2, % steady state (input) scaled run
                TU = us(:,1);
                TX = xs;
                TX = TX(1:(N-(M>0 && w~='c')*P));
                TU(TU==0) = 1.0; tu = 1.0./TU;
                TX(TX==0) = 1.0; tx = 1.0./TX;
                F = f; f = @(x,u,p) TX.*F(tx.*x,tu.*u,p);
```

```
            G = g; g = @(x,u,p)     G(tx.*x,tu.*u,p);
        end;

        if(nf(8)) % robust parameter
            J = J + P;
            ut = [ut;ones(P,T)];
            us = [us;repmat(pr,[1,T])];
            um = [um;pm];
            if(w=='y'), xm = [xm;pm]; end;
            F = f; f = @(x,u,p) F(x,u(1:J-P),u(J-P+1:J));
            G = g; g = @(x,u,p) G(x,u(1:J-P),u(J-P+1:J));
        end;

        m = N - P*(M>0 && w=='x'); % non-zero rows if joint gramian
        W = zeros(m,N);            % preallocate gramian
    end;

%% GRAMIAN COMPUTATION

    switch(w) % empirical gramian types

        case 'c', % controllability gramian
            for q=1:Q
                pp = pr(:,q);
                for c=1:C
                    for j=1:J % parfor
                        if(um(j,c)==0), continue; end;
                        if(M>0)
                            up = pr + sparse(M,1,um(j,c),P,1);
                            x = ODE(f,1,t,xs,us,up);
                        else
                            uu = us + bsxfun(@times,ut,um(j,c)*(1:J==j)');
                            x = ODE(f,1,t,xs,uu,pp);
                        end;
                        x = bsxfun(@minus,x,res(x));
                        x = x * (1.0./um(j,c));
                        W = W + (x*x'); % offload
                    end;
                end;
            end;
            W = W * (h/(C*Q));

        case 'o', % observability gramian
            o = zeros(O*T,N);
            for q=1:Q
                pp = pr(:,q);
                for d=1:D
                    for n=1:N % parfor
                        if(xm(n,d)==0), continue; end;
                        xx = xs + xm(n,d)*(1:N==n)';
                        if(M>0 && n>M && nf(9))
                            y = ODE(f,g,t,xx(1:M),us+ut,xx(M+1:end));
                        elseif(M>0)
                            y = ODE(f,g,t,xx(1:M),us,xx(M+1:end));
                        else
                            y = ODE(f,g,t,xx,us,pp);
                        end;
                        y = bsxfun(@minus,y,res(y));
                        y = y * (1.0/xm(n,d));
                        o(:,n) = y(:);
                    end;
                    W = W + (o'*o); % offload
                end;
            end;
            W = W * (h/(D*Q));

        case 'x', % cross gramian
            if(J~=O && nf(7)==0), error('ERROR! emgr: non-square system!'); end;
            o = zeros(O,T,N);
            for q=1:Q
                pp = pr(:,q);
                for d=1:D
                    for n=1:N % parfor
                        if(xm(n,d)==0), continue; end;
```

```
                xx = xs + xm(n,d)*(1:N==n)';
                if(M>0 && n>M && nf(9))
                    y = ODE(f,g,t,xx(1:M),us+ut,xx(M+1:end));
                elseif(M>0)
                    y = ODE(f,g,t,xx(1:M),us,xx(M+1:end));
                else
                    y = ODE(f,g,t,xx,us,pp);
                end;
                y = bsxfun(@minus,y,res(y));
                y = y * (1.0/xm(n,d));
                o(:,:,n) = y;
            end;
            o = permute(o,[2,3,1]); % generalized transposition
            if(nf(7))
                o(:,:,1) = sum(o,3);
            end;
            for c=1:C
                for j=1:J % parfor
                    if(um(j,c)==0), continue; end;
                    uu = us + bsxfun(@times,ut,um(j,c)*(1:J==j)');
                    if(M>0)
                        x = ODE(f,1,t,xs(1:M),uu,xs(M+1:end));
                    else
                        x = ODE(f,1,t,xs,uu,pp);
                    end;
                    x = bsxfun(@minus,x,res(x));
                    x = x * (1.0./um(j,c));
                    if(nf(7)) % non-symmetric cross gramian
                        W = W + (x*o(:,:,1)); % offload
                    else      % regular cross gramian
                        W = W + (x*o(:,:,j)); % offload
                    end;
                end;
            end;
            o = reshape(o,O,T,N); % reset
        end;
    end;
    W = W * (h/(C*D*Q));

case 'y', % linear cross gramian
    if(J~=O && nf(8)==0), error('ERROR! emgr: non-square system!'); end;
    for q=1:Q
        pp = pr(:,q);
        for c=1:C
            for j=1:J % parfor
                if(um(j,c)==0 || xm(j,c)==0), continue; end;
                uu = us + bsxfun(@times,ut,um(j,c)*(1:J==j)');
                x = ODE(f,1,t,xs,uu,pp);
                x = bsxfun(@minus,x,res(x));
                x = x * (1.0./um(j,c));
                uu = us + bsxfun(@times,ut,xm(j,c)*(1:J==j)');
                z = ODE(g,1,t,xs,uu,pp);
                z = bsxfun(@minus,z,res(z));
                z = z * (1.0./xm(j,c));
                W = W + (x*z'); % offload
            end;
        end;
    end;
    W = W * (h/(C*Q));

case 's', % sensitivity gramian
    W = cell(1,2);
    ps = sparse(P,1);
    nf(8) = 0;
    W{1} = emgr(f,g,[J,N,O],t,'c',ps,nf,ut,us,xs,um,xm);
    W{2} = zeros(P,1);
    for p=1:P
        V = emgr(f,g,[1,N,O,p],t,'c',pr,nf,ut,us,xs,pm(p,:),xm);
        W{1} = W{1} + V;        % approximate controllability gramian
        W{2}(p) = trace(V);
    end;
    if(nf(11))
        W{2} = W{2} - mean(W{2});
```

```
            end;
            W{2} = spdiags(W{2},0,P,P); % sensitivity gramian

        case 'i', % identifiability gramian
            W = cell(1,2);
            ps = sparse(P,1);
            V = emgr(f,g,[J,N+P,O,N],t,'o',ps,nf,ut,us,[xs;pr],um,[xm;pm]);
            W{1} = V(1:N,1:N);         % observability gramian
            W{2} = V(N+1:N+P,N+1:N+P); % identifiability gramian
            if(nf(11))
                W{2} = W{2} - V(N+1:N+P,1:N)*ainv(W{1})*V(1:N,N+1:N+P);
            end;

        case 'j', % joint gramian
            W = cell(1,2);
            ps = sparse(P,1);
            V = emgr(f,g,[J,N+P,O,N],t,'x',ps,nf,ut,us,[xs;pr],um,[xm;pm]);
            W{1} = V(1:N,1:N); % cross gramian
            %W{2} = zeros(P,P); % cross-identifiability gramian
            if(nf(11))
                W{2} = -0.5*V(1:N,N+1:N+P)'*pinv(W{1}+W{1}')*V(1:N,N+1:N+P);
            else
                W{2} = -0.5*V(1:N,N+1:N+P)'*ainv(W{1}+W{1}')*V(1:N,N+1:N+P);
            end;

        otherwise,
            error('ERROR! emgr: unknown gramian type!');
    end;

    if(nf(12) && (w=='c' || w=='o' || w=='y' || w=='x') ) % enforce symmetry
        W(1:m,1:m) = 0.5*(W(1:m,1:m) + W(1:m,1:m)');
    end;
end

%% ======== SCALES SELECTOR ========
function s = scales(s,d,e)

    switch(d)

        case 0, % linear
            s = s*[0.25,0.50,0.75,1.0];

        case 1, % logarithmic
            s = s*[0.001,0.01,0.1,1.0];

        case 2, % geometric
            s = s*[0.125,0.25,0.5,1.0];

        case 4, % sparse
            s = s*[0.38,0.71,0.92,1.0];

        otherwise, % single
            %s = s;
    end;

    switch(e)

        case 1, % reciproce
            s = [1.0./s,s];

        case 2, % dyadic
            s = s*s';

        case 3, % single
            %s = s;

        otherwise, % unit
            s = [-s,s];
    end;
end

%% ======== FAST APPROXIMATE INVERSION ========
function x = ainv(m)
```

```
    d = diag(m);
    d(d~=0) = 1.0./d(d~=0);
    n = numel(d);
    x = bsxfun(@times,m,-d);
    x = bsxfun(@times,x,d');
    x(1:n+1:end) = d;
end

%% ======== DEFAULT ODE INTEGRATOR ========
function x = rk2(f,g,t,z,u,p)

    if(isnumeric(g) && g==1), g = @(x,u,p) x; end;

    h = t(1);
    L = floor(t(2)/h) + 1;

    x(:,1) = g(z,u(:,end),p);
    x(end,L) = 0; % preallocate trajectory

    for l=2:L % 2nd order Ralston's Runge-Kutta Method
        k1 = h*f(z,u(:,l-1),p);
        k2 = h*f(z + 0.666666666666667*k1,u(:,l-1),p);
        z = z + 0.25*k1 + 0.75*k2;
        x(:,l) = g(z,u(:,l-1),p);
    end;
end
```

B.4. optmor - Optimization-Based Model Order Reduction

Code Metadata

name (short name)	Optimization-Based Model Order Reduction (optmor)
version (release date)	2.5 (2016-02-28)
topic (type)	Model reduction (Toolbox)
author (ORCID)	Christian Himpe (0000-0003-2194-6754)
license (type)	2-Clause BSD (open-source)
repository (type)	`github.com/gramian/optmor` (git)
language	Matlab
dependencies (system)	Octave | Matlab (Linux, Windows)
website	`gramian.github.io/optmor`
keywords	greedy sampling, MOR

License

Source Code

```
function XP = optmor(f,g,s,t,r,q,nf,ut,x0,co,yd)
% optmor (Version 2.5)
% by Christian Himpe, 2013-2016 ( wwwmath.uni-muenster.de/u/himpe )
% released under BSD 2-Clause License ( opensource.org/licenses/BSD-2-Clause )
%
% SYNTAX:
%    W = optmor(f,g,s,t,r,q,[nf],[ut],[x0],[co],[yd]);
%
% SUMMARY:
%    optmor - optimization-based model order reduction,
%    for the computation of combined state and parameter
%    reduced order models of state-space input-output systems.
%    Compatible with OCTAVE and MATLAB.
%
% ARGUMENTS:
%   (func handle)  f - system function handle; signature: xdot = f(x,u,p)
%   (func handle)  g - output function handle; signature:    y = g(x,u,p)
%        (vector)  s - system dimensions [inputs,states,outputs]
%        (vector)  t - time discretization [step,stop]
%        (scalar)  r - reduced order(>1) or error threshold(<1)
%        (vector)  q - nominal parameter
%        (vector,scalar) [nf = 0] - options, 6 components:
%            + Optimization Algorithm: fminunc(0), fminsearch(1), custom(-1)
%            + Lasso Regularization Weight: default(0)
%            + Tikhonov Regularization Weight: default(0.1)
%            + Data-Driven Regularization Weight: default(0)
%            + Number of Maximum Optimizer Iterations: default(4)
%            + Initial Parameter: last(0), random(1)
%        (matrix,vector,scalar,handle)  ut - input; default: delta impulse
%        (vector,scalar)  x0 - initial state; default: zeros
%        (matrix,vector,scalar)  [co = 1] - covariance matrix: unit
%        (matrix)  yd - experimental data: empty
%
% RETURNS:
%             (cell)  XP - {State-,Parameter-} Projection
%
% CITATION:
%    C. Himpe (2016). optmor - Optimization-Based Model Order Reduction
%    (Version 2.5) [Software]. Available from http://github.com/gramian/optmor .
%    doi: 10.5281/zenodo.46683 .
%
% KEYWORDS:
%    model reduction, combined reduction, greedy sampling
%*

    % Custom Solver
    global ODE;
    if(isa(ODE,'function handle')==0), ODE = @rk2; end;

    % Version Info
    if( nargin==1 && strcmp(f,'version') ), XP = 2.5; return; end;

    %
    if(~exist('OCTAVE_VERSION','builtin')), vec = @(m) m(:); end;

    % Default Arguments
    if( nargin<7  || isempty(nf) ), nf = 0; end; % Assume default options
    if( nargin<8  || isempty(ut) ), ut = 1.0; end; % Assume impulse input
    if( nargin<9  || isempty(x0) ), x0 = 0; end; % Assume zero initial state
    if( nargin<10 || isempty(co) ), co = 1.0; end; % Assume covariance
    if( nargin<11 || isempty(yd) ), yd = 0; end; % Assume no experimental data

    % System Constants
    J = s(1);               % Number of inputs
    N = s(2);               % Number of states
    O = s(3);               % Number of outputs
    h = t(1);               % Time step width
    T = floor(t(2)/h) + 1;  % Number of time steps
    Q = numel(q);           % Number of parameters

    % Linear Chirp Input
    if( isnumeric(ut) && numel(ut)==1 && ut==Inf )
```

```
        ut = @(t) 0.5*cos(pi*(t+10*t.*t))+0.5;
    end;

    % Discretize Procedural Input
    if(isa(ut,'function_handle'))
        uf = ut;
        ut = zeros(J,T);
        for l=1:T
            ut(:,l) = uf(l*h);
        end;
    end;

    % Lazy arguments
    if(numel(nf)<6),  nf(6) = 0; end;
    if(numel(ut)==1), ut(1:J,1) = ut./h; end;
    if(numel(x0)==1), x0(1:N,1) = x0; end;
    if(numel(co)==1), co(1:Q,1) = co; end;

    if(size(ut,2)==1), ut(:,2:T) = 0.0; end;
    if(size(co,1)==1 || size(co,2)==1), co = spdiags(co,0,Q,Q); end;

%% SETUP

    % Set Abort Critera
    r = abs(r);
    if(r >= 1.0)
        n = r;
    else
        n = Q;
    end;

    % Set Default Tikhonov Regularization Coefficient
    if(nf(3)==0 && nf(2)==0)
        nf(3) = 0.1;
    end;

    % Check for Data if Data-Driven Regularization
    if(nf(4) && ( (numel(yd)==1 && yd==0) || size(yd,1)~=O || size(yd,2)~=T) )
        error('ERROR! optmor: yd data dimension mismatch!');
    end;

    % Set Default Maximum Optimizer Iterations
    if(nf(5)==0)
        nf(5) = 4;
    end;

    % Set up Regularization Operators
    if(nf(2)~=0 && nf(3)==0 && nf(4)==0)
        R = @(p,y) nf(2)*norm(p,1);
    elseif(nf(2)==0 && nf(3)~=0 && nf(4)==0)
        R = @(p,y) nf(3)*norm(p,2)^2;
    elseif(nf(2)~=0 && nf(3)~=0 && nf(4)==0)
        R = @(p,y) nf(2)*norm(p,1) + nf(3)*norm(p,2)^2;
    elseif(nf(2)==0 && nf(3)==0 && nf(4)~=0)
        R = @(p,y) nf(4)*t(1)*norm(vec(y-yd),2)^2;
    elseif(nf(2)~=0 && nf(3)==0 && nf(4)~=0)
        R = @(p,y) nf(2)*norm(p,1) + nf(4)*t(1)*norm(vec(y-yd),2)^2;
    elseif(nf(2)==0 && nf(3)~=0 && nf(4)~=0)
        R = @(p,y) nf(3)*norm(p,2)^2 + nf(4)*t(1)*norm(vec(y-yd),2)^2;
    elseif(nf(2)~=0 && nf(3)~=0 && nf(4)~=0)
        R = @(p,y) nf(2)*norm(p,1) + nf(3)*norm(p,2)^2 ...
                                   + nf(4)*t(1)*norm(vec(y-yd),2)^2;
    end;

%% INIT LOOP

    fprintf('optmor progress:\n');

    % Set Initial Parameter Projection
    p = q;
    P = q./norm(q,2);

    % Compute Trajectory for Initial Parameter
    z = ODE(f,1,t,x0,ut,p);
```

```
    [X,dtemp,vtemp] = svds(z,1);

    % Set Optimizer Options
    flags = optimset('Display','off','MaxIter',nf(5));

    % Per Interation Timing
    global TIMINGS;
    TIMINGS = zeros(Q,1);

    fprintf('|');

%% MAIN LOOP

    for I=2:n

        k = tic;

        % Current reduced order model
        fr = @(x,u,p) X'*f(X*x,u,p);
        gr = @(x,u,p) g(X*x,u,p);
        x0r = X'*x0;

        % Termination Test
        if(r < 1.0 && t(1)*norm(vec(ODE(f,g,t,x0,ut,p) ...
                                -ODE(fr,gr,t,x0r,ut,P*(P'*p))),2) < r)
            break;
        end;

        % Set up cost function
        jf = @(p,y) -t(1)*norm(vec(y-ODE(fr,gr,t,x0r,ut,P*(P'*p))),2)^2 +R(p,y);
        Jf = @(p) jf(p,ODE(f,g,t,x0,ut,p));

        % Initial Parameter
        if(nf(6))
            p = q + co * randn(Q,1);
        end;

        % Greedy Sampling Algorithm
        switch(nf(1))

            case 0, % Unconstrained (Quasi-Newton)
                p = fminunc(Jf,p,flags);

            case 1, % Derivative-Free (Nelder-Mead)
                p = fminsearch(Jf,p,flags);

            case -1, % Custom Optimizer
                global FMIN;
                p = FMIN(Jf,p);
        end;

        % Extract and Incorporate New State Base
        z = ODE(f,1,t,x0,ut,p);
        z = z - X * (z' * X)';
        [x,dtemp,vtemp] = svds(z,1);
        X = gramschmidt(X,x);

        % Incorporate New Parameter Base
        P = gramschmidt(P,p);

        TIMINGS(I) = toc(k);

        if(mod(I,100)), fprintf('|'); else, fprintf('#'); end;
    end;

    fprintf('\n');

    XP = {X,P};
end

%% ======== RE-ITERATED GRAM-SCHMIDT ========
function Q = gramschmidt(Q,v)

    m = size(v,2);
```

```
    for I=1:m
        if(size(Q,2)>=size(Q,1))
            return;
        end;

        w = v(:,I);
        b = 0;

        for J=1:10

            w = w - Q * (w' * Q)';
            b = norm(w,2);
            w = w./b;
            if(b>0.1)
                break;
            end;
        end;

        Q = [Q,w];
    end;
end

%% ======== DEFAULT ODE INTEGRATOR ========
function x = rk2(f,g,t,z,u,p)

    if(isnumeric(g) && g==1), g = @(x,u,p) x; end;

    h = t(1);
    L = floor(t(2)/h) + 1;

    x(:,1) = g(z,u(:,end),p);
    x(end,L) = 0; % preallocate trajectory

    for l=2:L % 2nd order Ralston's Runge-Kutta Method
        k1 = h*f(z,u(:,l-1),p);
        k2 = h*f(z + 0.666666666666667*k1,u(:,l-1),p);
        z = z + 0.25*k1 + 0.75*k2;
        x(:,l) = g(z,u(:,l-1),p);
    end;
end
```

C. Additional Figures

Supplementary to the plots for the experiments in Chapter 7, this section provides additional comparisons of outputs between FOMs and ROMs for fixed reduced orders and specific parameters.

C.1. Hyperbolic Network Model

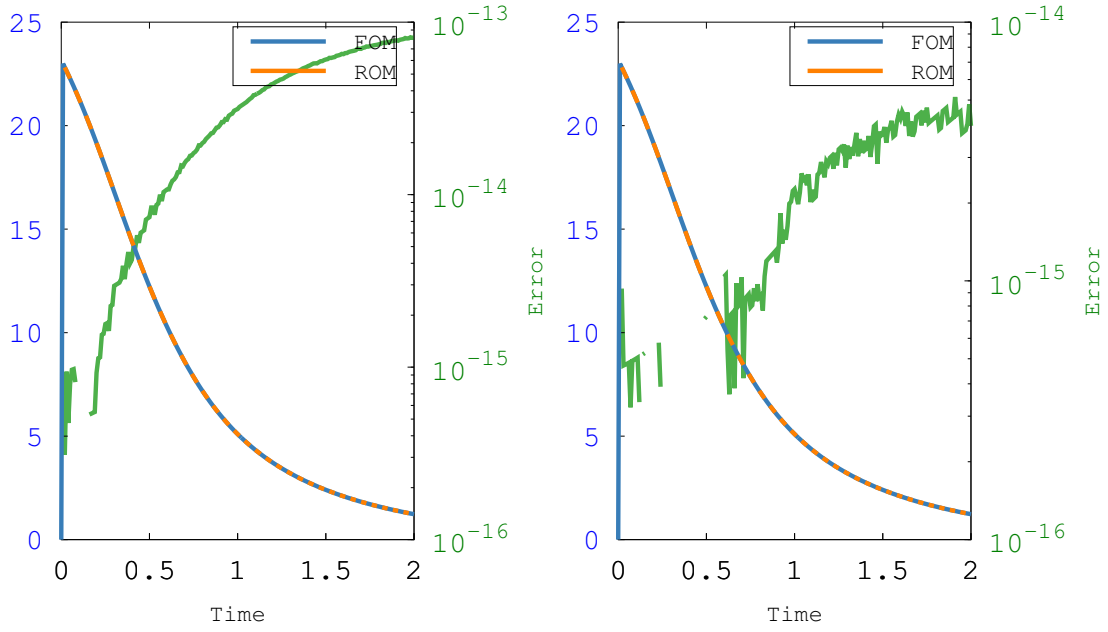
In Figure C.1 the outputs of the ROMs obtained from combined reduction methods are compared to the output of the full order hyperbolic network model for a parameter θ sampled from the uniform random distribution $\mathcal{U}^P_{[\frac{1}{10},1]}$. The comparison of outputs of the nonlinear FOM with the outputs from the gramian-based ROMs using the nonlinear and linearized hyperbolic network model, of reduced order $n = p = 60$, is shown in Figure C.1a and Figure C.1b, and with the outputs from the optimization-based ROMs using the nonlinear and linearized hyperbolic network model, of reduced order $n = p = 60$, is shown in Figure C.1c and Figure C.1d.

C.2. fMRI & fNIRS Dynamic Causal Model

In Figure C.2 the outputs of the ROMs obtained from combined reduction methods are compared to the output of the full order fMRI & fNIRS dynamic causal model for a parameter θ sampled from the log-normal random distribution $\ln \mathcal{N}_{1,\frac{1}{4}}$ for the components related to the scaling of off-diagonal components and $\ln \mathcal{N}_{1,\frac{1}{40}}$ for the components related to the scaling of diagonal components. The comparison of outputs of the nonlinear FOM with the outputs from the gramian-based ROMs using the nonlinear and linearized fMRI & fNIRS model, of reduced order $n = p = 44$, is shown in Figure C.2a and Figure C.2b, and with the outputs from the optimization-based ROMs using the nonlinear and linearized fMRI & fNIRS model, of reduced order $n = p = 24$, is shown in Figure C.2c and Figure C.2d.

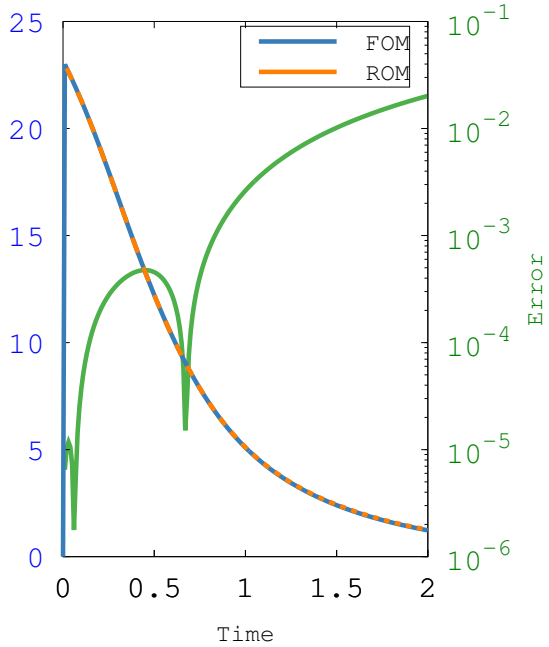
C.3. EEG & MEG Dynamic Causal Model

In Figure C.3 the outputs of the ROMs obtained from combined reduction methods are compared to the output of the full order EEG & MEG dynamic causal model for a parameter θ sampled from the log-normal random distribution $\ln \mathcal{N}_{1,\frac{1}{4}}$ for the components related to the scaling of off-diagonal components and $\ln \mathcal{N}_{1,\frac{1}{40}}$ for the components related to the scaling of diagonal components. The comparison of outputs of the nonlinear FOM with the outputs from the gramian-based ROMs using the nonlinear and linearized EEG & MEG model, of reduced order $n = 80$, $p = 40$, is shown in Figure C.3a and Figure C.3b, and with the outputs from the optimization-based ROMs using the nonlinear and linearized EEG & MEG model, of reduced order $n = 80$, $p = 40$, is shown in Figure C.3c and Figure C.3d.

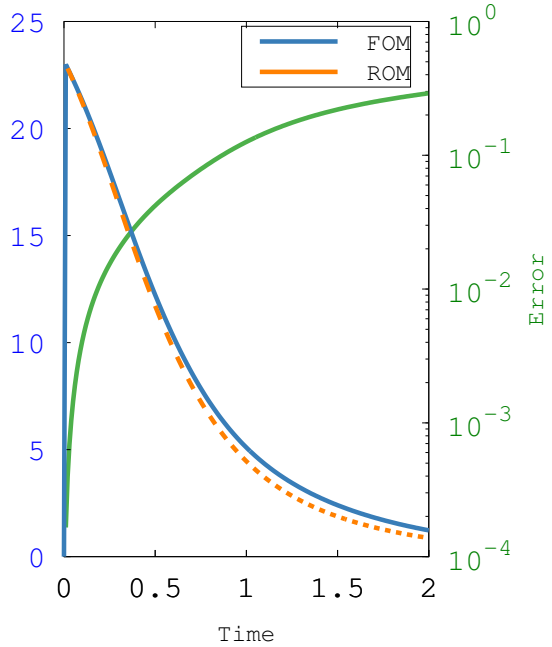


(a) *Comparison of FOM and nonlinear gramian-based ROM for* $n = p = 60$.

(b) *Comparison of FOM and linearized gramian-based ROM for* $n = p = 60$.

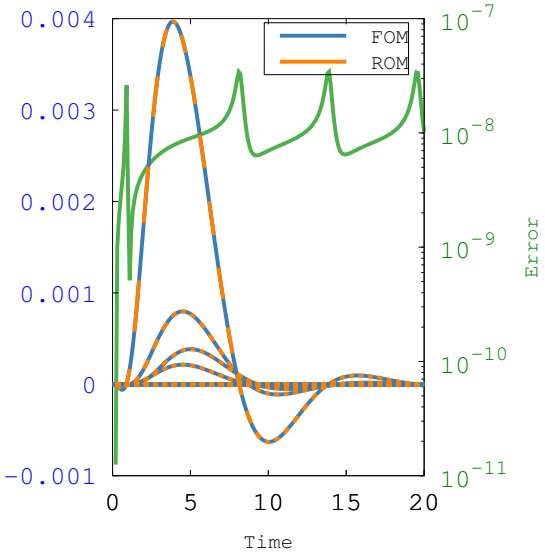


(c) *Comparison of FOM and nonlinear optimization-based ROM for* $n = p = 60$.

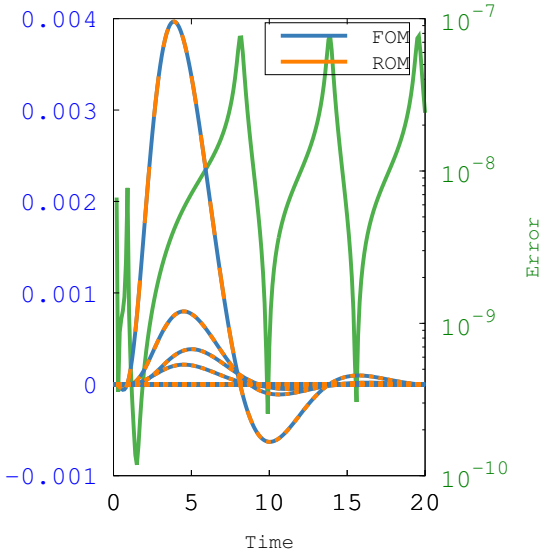


(d) *Comparison of FOM and linearized optimization-based ROM for* $n = p = 60$.

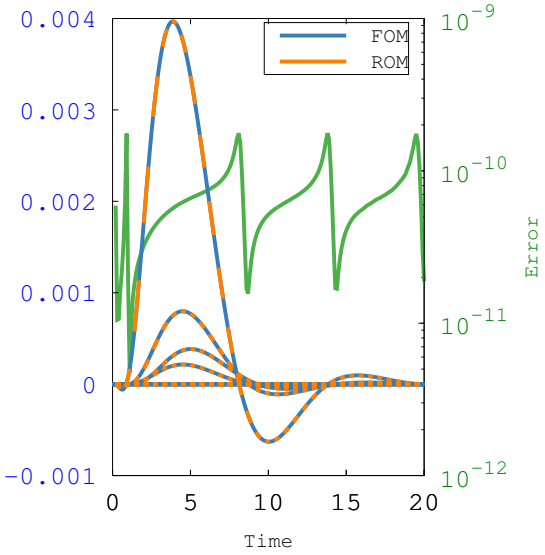
Figure C.1.: *Comparison of FOM and ROM outputs for the hyperbolic network model.*



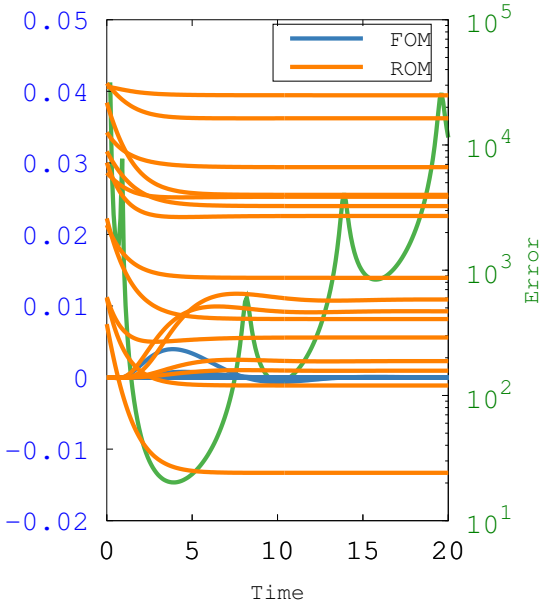
(a) *Comparison of FOM and nonlinear gramian-based ROM for* $n = p = 44$.



(b) *Comparison of FOM and linearized gramian-based ROM for* $n = p = 44$.

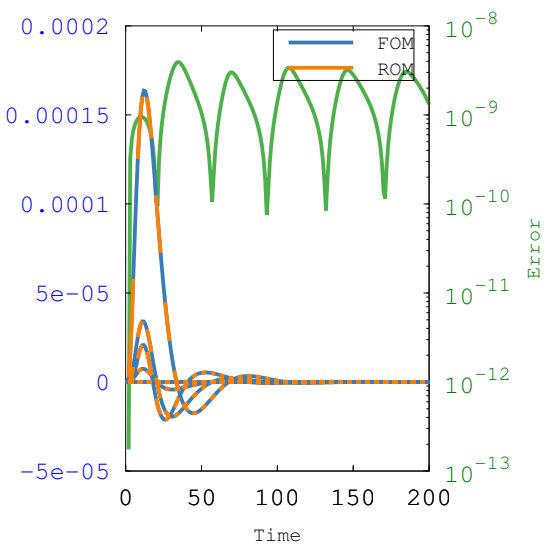


(c) *Comparison of FOM and nonlinear optimization-based ROM for* $n = p = 24$.

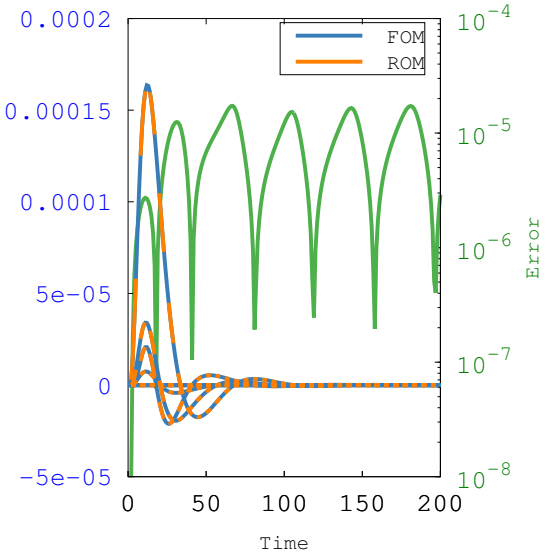


(d) *Comparison of FOM and linearized optimization-based ROM for* $n = p = 24$.

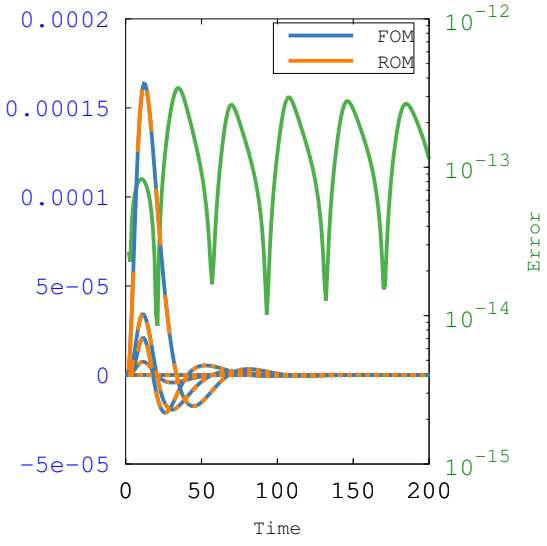
Figure C.2.: *Comparison of FOM and ROM outputs for the fMRI & fNIRS model.*



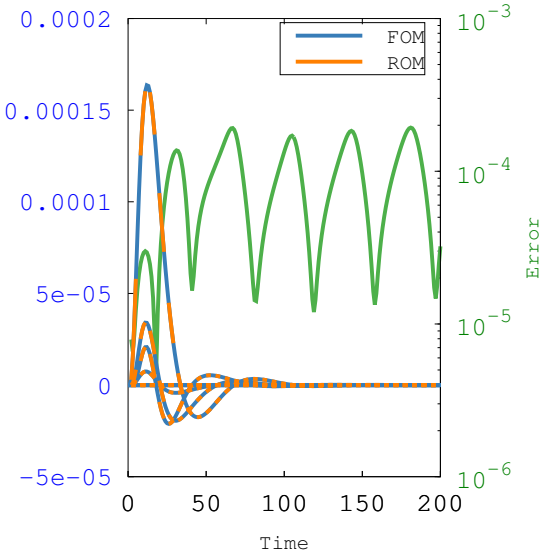
(a) *Comparison of FOM and nonlinear gramian-based ROM for* $n = 80$, $p = 40$.



(b) *Comparison of FOM and linearized gramian-based ROM for* $n = 80$, $p = 40$.



(c) *Comparison of FOM and nonlinear optimization-based ROM for* $n = 80$, $p = 40$.



(d) *Comparison of FOM and linearized optimization-based ROM for* $n = 80, p = 40$.

Figure C.3.: *Comparison of FOM and ROM outputs for the EEG & MEG model.*

D. List of Acronyms

ADI – Alternating Direction Implicit

APU – Accelerated Processing Unit

BIBO – Bounded-Input-Bounded-Output

BLAS – Basic Linear Algebra Subprogram

BOLD – Blood Oxygen Level Dependent

bPOD – balanced Proper Orthogonal Decomposition

BT – Balanced Truncation

CFL – Courant-Friedrichs-Lewy

CGMIL – Cross Gramian Minimum Information Loss

CPU – Central Processing Unit

DCM – Dynamic Causal Modelling

DD – Data-Driven

DEIM – Discrete Empirical Interpolation Method

DT – Direct Truncation

EEG – Electroencephalography

EIM – Empirical Interpolation Method

fMRI – functional Magneto Resonance Imaging

fNIRS – functional Near Infrared Spectroscopy

FOM – Full Order Model

GPGPU – General Purpose Graphics Processing Unit

GPU – Graphics Processing Unit

HSV – Hankel Singular Values

iEEG – Intracranial EEG

IFP – Inference for Prediction

ILP – Inverse Lyapunov Procedure

IVP – Initial Value Problem

KPCA – Kernel Principal Component Analysis

LAPACK – Linear Algebra Package

LoC – Lines of Code

LTI – Linear Time-Invariant

MAP – Maximum A-Posteriori

MEG – Magnetoencephalography

MIMO – Multiple-Input-Multiple-Output

ML – Maximum Likelihood

MOR – Model Order Reduction

nMOR – nonlinear Model Order Reduction

ODE – Ordinary Differential Equation

PCA – Principal Component Analysis

PDE – Partial Differential Equation

pMOR – parametrized Model Order Reduction

PMTBR – Poor Man's Truncated Balanced Realization

POD – Proper Orthogonal Decomposition

QoI – Quantity of Interest

RAM – Random Access Memory

RBM – Reduced Basis Method

RC – Resistor Capacitor

RK – Runge Kutta

RK2 – Second-Order Runge-Kutta

RKHS – Reproducing Kernel Hilbert Spaces

RMS – Root Mean Square

ROM – Reduced Order Model

SIMD – Single-Instruction-Multiple-Data

SIMO – Single-Input-Multiple-Output

SISO – Single-Input-Single-Output

SNR – Signal-to-Noise Ratio

SOBT – Second-Order Balanced Truncation

SSP – Strong Stability Preserving

SVD – Singular Value Decomposition

TSVD – Truncated Singular Value Decomposition

E. Related Contributions

Bookchapter U. Baur, P. Benner, B. Haasdonk, C. Himpe, I. Martini and M. Ohlberger, "Comparison of Methods for Parametric Model Order Reduction of Instationary Problems", to appear in *P. Benner, A. Cohen, M. Ohlberger and K. Willcox: "Model Reduction and Approximation: Theory and Algorithms"*, 2017. See also [18].

Article C. Himpe and M. Ohlberger, "A note on the cross Gramian for non-symmetric systems", in *System Science and Control Engineering*, 2016. See also [117].

Talk C. Himpe and M. Ohlberger, "Empirical Gramians 4 MOR", at *MOR 4 MEMS*, 2015 in Karlsruhe (Germany).

Talk C. Himpe and M. Ohlberger, "Combined State and Parameter Reduction (for Input-Output Systems)", at *Model Reduction in Industry*, 2015 in Luxembourg (Luxembourg).

Poster C. Himpe and M. Ohlberger, "The Versatile Cross Gramian", at *Model Reduction for Parametrized Systems (MoRePaS) 3*, 2015 in Trieste (Italy). See also [115].

Poster C. Himpe and M. Ohlberger, "emgr - Empirical Gramian Framework", at *OctConf*, 2015 in Darmstadt (Germany).

Minisymposium U. Baur and C. Himpe, "Parametric Model Order Reduction: Challenges and Solutions", at *International Conference on Scientific Computation And Differential Equations (SciCADE)*, 2015 in Potsdam (Germany).

Poster C. Himpe and M. Ohlberger, "Optimization-Based Combined Reduction for Nonlinear Systems", at *Modelling Revisited and Model Reduction (MORE) Workshop*, 2015 in Pilzň (Czech Republic).

Article C. Himpe and M. Ohlberger, "Data-Driven Combined State and Parameter Reduction for Inverse Problems", in *Advances in Computational Mathematics (Special Issue on Model Order Reduction of Parameterized Systems)*, 2015. See also [114].

Talk C. Himpe and M. Ohlberger, "Zero-Copy Parallelized Empirical Gramians", at *Power Aware Computing Workshop (PACO)*, 2015 in Magdeburg (Germany).

Talk C. Himpe and M. Ohlberger, "Accelerating the Computation of Empirical Gramians and Related Methods", at *5th International Workshop for Model Reduction in Reactive Flows (IWMRRF)*, 2015 in Lübbenau (Germany). See also [116].

Talk C. Himpe and M. Ohlberger, "Model Reduction for Parametrized Systems", at *Twente-Münster Minisymposium*, 2015 in Enschede (Netherlands).

Talk C. Himpe and M. Ohlberger, "Combined State and Parameter Reduction for the Inversion of Functional Neuroimaging Data", at *Conference on Computational Science and Engineering (CSE)*, 2015 in Salt Lake City (USA).

Poster C. Himpe and M. Ohlberger "emgr - Empirical Gramian Framework", at *Conference on Computational Science and Engineering (CSE)*, 2015 in Salt Lake City (USA).

Proceedings C. Himpe and M. Ohlberger, "The Empirical Cross Gramian for Parametrized Nonlinear Systems", in *Mathematical Modelling*, 2015. See also [113].

Talk C. Himpe and M. Ohlberger, "The Empirical Cross Gramian for Parametrized Nonlinear Systems", at *8th International Conference on Mathematical Modelling (MathMod)*, 2015 in Vienna (Austria).

Article C. Himpe and M. Ohlberger, "Cross-Gramian-Based Combined State and Parameter Reduction for Large-Scale Control Systems", in *Mathematical Problems in Engineering*, 2014. See also [110].

Talk U. Baur, P. Benner, B. Haasdonk, C. Himpe, I. Maier and M. Ohlberger, "Comparison of Methods for Parametric Model Order Reduction of Instationary Problem (II)", held at *European Model Reduction Network (EU-MORNET) Kick-Off Meeting*, 2014 in Eindhoven (Netherlands).

Talk C. Himpe and M. Ohlberger, "emgr (Software Short Communications)", at *Reduced Basis Summer School*, 2014 in Münster (Germany).

Talk C. Himpe and M. Ohlberger, "Yet Another Talk About Empirical Gramians", at *Reduced Basis Summer School*, 2014 in Münster (Germany).

Talk C. Himpe and M. Ohlberger, "Efficient Cross-Gramian-Based State and Parameter Reduction", at *WCCM XI / ECCM V / ECFD VI*, 2014 in Barcelona (Spain).

Proceedings C. Himpe and M. Ohlberger, "Model Reduction For Complex Hyperbolic Networks", in *Proceedings of the European Control Conference*, 2014. See also [111].

Talk C. Himpe and M. Ohlberger, "Model Reduction for Complex Hyperbolic Networks", at European Control Conference (ECC), 2014 in Strasbourg (France).

Talk C. Himpe and M. Ohlberger, "emgr - Empirical Gramian Framework", at *4th European Seminar on Computing (ESCO)*, 2014 in Pilzň (Czech Republic).

Proceedings C. Himpe and M. Ohlberger, "Combined State and Parameter Reduction", in *Proceedings in Applied Mathematics and Mechanics (PAMM)*, 2014. See also [112].

Talk C. Himpe and M. Ohlberger, "Combined State and Parameter Reduction", at *Jahrestagung der Gesellschaft für Angewandte Mathematik und Mechanik (GAMM)*, 2014 in Erlangen (Germany).

Article C. Himpe and M. Ohlberger, “A Unified Software Framework for Empirical Gramians”, in *Journal of Mathematics*, 2013. See also [109].

Talk C. Himpe and M. Ohlberger, “Combined Reduction for EEG Inverse Models”, at *2nd Model Reduction of Complex Dynamical Systems (ModRed)*, 2013 in Magdeburg (Germany).

Talk C. Himpe and M. Ohlberger, “Model Reduction for Inverse Network Models”, at *Mathemtical Technology of Networks*, 2013 in Bielefeld (Germany).

Talk C. Himpe and M. Ohlberger, “Combined Reduction for Neural Networks”, at *University of Konstanz - Oberseminar Numerik*, 2013 in Konstanz (Germany)

Poster C. Himpe and M. Ohlberger, “Reduced Bayesian Inversion”, at *Aachen Conference on Computational Engineering Science (AC.CES)*, 2013 in Aachen (Germany).

Talk C. Himpe and M .Ohlberger “Combined State and Parameter Reduction of Hierarchical Systems”, at *European Conference on Numerical Mathematics and Advanced Applications (ENuMath)*, 2013 in Lausanne (Switzerland).

Talk C. Himpe and M. Ohlberger, “emgr - Empirical Gramian Framework”, at *Reduced Basis Summer School*, 2013 in Kopp (Germany).

Poster C. Himpe and M. Ohlberger, “Combined Reduction”, at *Model Reduction and Approximation for Complex Systems (ModRedCIRM)*, 2013 in Marseille (France).

Talk C. Himpe and M. Ohlberger, “Data-Driven Parameter Reduction Methods for Bayesian Inverse Problems”, at *Model Reduction for Parametrized Systems (MoRePaS) 2*, 2012 in Ulm (Germany).

Bibliography

[1] J.A. De Abreu-Garcia and F.W. Fairman. A Note on Cross Grammians for Orthogonally Symmetric Realizations. *IEEE Transactions on Automatic Control*, 31(9):866–868, 1986. DOI 10.1109/TAC.1986.1104421.

[2] V.M. Adamjan, D.Z. Arov, and M.G. Kreĭn. Analytic Properties of Schmidt Pairs for a Hankel Operator and the Generalized Schur-Takagi Problem. *Sbornik: Mathematics*, 15(1):31–73, 1971. DOI 10.1070/SM1971v015n01ABEH001531.

[3] A. Alavian and M.C. Rotkowitz. On a Hankel-based Measure of Decentralized Controllability and Observability. In *5th IFAC Workshop on Distributed Estimation and Control in Networked Systems*, volume 48(22), pages 227–232, 2015. DOI 10.1016/j.ifacol.2015.10.335.

[4] R.W. Aldhaheri. Model order reduction via real Schur-form decomposition. *International Journal of Control*, 53(3):709–716, 1991. DOI 10.1080/00207179108953642.

[5] Y.M. Altman. *Accelerating MATLAB Performance: 1001 tips to speed up MATLAB programs*. CRC Press, 2015. ISBN 9781482211306.

[6] A.C. Antoulas. *Approximation of Large-Scale Dynamical Systems*, volume 6 of *Advances in Design and Control*. SIAM, 2005. ISBN 9780898715293. DOI 10.1137/1.9780898718713.

[7] A.C. Antoulas. An overview of approximation methods for large-scale dynamical systems. *Annual Reviews in Control*, 29(2):181–190, 2005. DOI 10.1016/j.arcontrol.2005.08.002.

[8] A.C. Antoulas, D.C. Sorensen, and S. Gugercin. A survey of model reduction methods for large-scale systems. In *Structured Matrices in Mathematics, Computer Science, and Engineering I*, volume 280 of *Contemporary Mathematics*, pages 193–219. AMS, 2001. DOI 10.1090/conm/280/04630.

[9] S. Bagheri, L. Brandt, and D.S. Henningson. Input-output analysis, model reduction and control of the flat-plate boundary layer. *Journal of Fluid Mechanics*, 620:263–298, 2009. DOI 10.1017/S0022112008004394.

[10] A. Barbagallo, V.F. De Felice, and K.K. Nagarajan. Reduced Order Modelling of a Couette Flow Using Balanced Proper Orthogonal Decomposition. In *2nd Young ERCOFTAC Workshop*, 2008. URL dokumente.unibw.de/pub/bscw.cgi/2700941.

[11] M. Barrault, Y. Maday, N.C. Nguyen, and A.T. Patera. An ‘empirical interpolation‘ method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathematique*, 339(9):667–672, 2004. DOI 10.1016/j.crma.2004.08.006.

[12] O. Bashir. Hessian-Based Model Reduction with Applications to Initial-Condition Inverse Problems. Master's thesis, Massachusetts Institute of Technology, 2007. URL hdl.handle.net/1721.1/42049.

[13] O. Bashir, K. Willcox, O. Ghattas, B. van Bloemen Waanders, and J. Hill. Hessian-based model reduction for large-scale systems with initial-condition inputs. *International Journal for Numerical Methods in Engineering*, 73(6):844–868, 2008. DOI 10.1002/nme.2100.

[14] U. Baur and P. Benner. Cross-Gramian Based Model Reduction for Data-Sparse Systems. *Electronic Transactions on Numerical Analysis*, 31:256–270, 2008. URL eudml.org/doc/130548.

[15] U. Baur and P. Benner. Parametrische Modellreduktion mit dünnen Gittern (Parametric model reduction with sparse grids). In *GMA Fachausschuß 1.30 Workshop*, pages 262–271, 2008. ISBN 9783950245110. URL www2.mpi-magdeburg.mpg.de/mpcsc/benner/pub/BaurBenner-GMA-Proceedings2008.pdf.

[16] U. Baur, C. Beattie, P. Benner, and S. Gugercin. Interpolatory Projection Methods for Parameterized Model Reduction. *SIAM Journal on Scientific Computing*, 33(5): 2489–2518, 2011. DOI 10.1137/090776925.

[17] U. Baur, P. Benner, and L. Feng. Model Order Reduction for Linear and Nonlinear Systems: A System-Theoretic Perspective. *Archives of Computational Methods in Engineering*, 21(4):331–358, 2014. DOI 10.1007/s11831-014-9111-2.

[18] U. Baur, P. Benner, B. Haasdonk, C. Himpe, I. Martini, and M. Ohlberger. Comparison of methods for parametric model order reduction of instationary problems. In P. Benner, A. Cohen, M. Ohlberger, and K. Willcox, editors, *Model Reduction and Approximation: Theory and Algorithms*. SIAM, 2017. URL www2.mpi-magdeburg.mpg.de/preprints/2015/MPIMD15-01.pdf. To appear.

[19] P. Benner and G. Quintana-Ortì E.S. Quintana-Ortì. Model Reduction of Large-Scale Dense Systems. In *Proceedings of the IEE UKACC International Conference on Control*, volume 3, pages 1–6, 2000. URL citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.29.5706.

[20] P. Benner and H. Fassbender. Model Order Reduction: Techniques and Tools. In *Encyclopedia of Systems and Control*, pages 1–10. Springer, 2014. DOI 10.1007/978-1-4471-5102-9_142-1.

[21] P. Benner and J. Saak. Efficient balancing-based MOR for large-scale second-order systems. *Mathematical and Computer Modelling of Dynamical Systems*, 17(2):123–143, 2011. DOI 10.1080/13873954.2010.540822.

[22] P. Benner, E. Sachs, and S. Volkwein. Model Order Reduction for PDE Constrained Optimization. In *Trends in PDE Constrained Optimization*, volume 165 of *International Series of Numerical Mathematics*, pages 303–326. Springer, 2014. ISBN 9783319050829. DOI 10.1007/978-3-319-05083-6_19.

[23] P. Benner, S. Gugercin, and K. Willcox. A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems. *SIAM Review*, 57(4):483–531, 2015. DOI 10.1137/130932715.

[24] D. Bickson. *Gaussian Belief Propagation: Theory and Application*. PhD thesis, The Hebrew University of Jerusalem, 2009. URL arxiv.org/abs/0811.2518.

[25] P. Binev, A. Cohen, W. Dahmen, R. DeVore, G. Petrova, and P. Wojtaszczyk. Convergence Rates for Greedy Algorithms in Reduced Basis Methods. *SIAM Journal on Mathematical Analysis*, 43(3):1457–1472, 2011. DOI 10.1137/100795772.

[26] M. Boguñá, M. Kitsak, and D. Krioukov. Cosmological networks. *New Journal of Physics*, 16(9):1–13, 2014. DOI 10.1088/1367-2630/16/9/093031.

[27] J. Bouvrie and B. Hamzi. Kernel Methods for the Approximation of Nonlinear Systems. *arXiv Preprint*, math.OC(1108.2903):1–31, 2016. URL arxiv.org/abs/1108.2903.

[28] A. Buhr, C. Engwer, M. Ohlberger, and S. Rave. A numerically stable a posteriori error estimator for reduced basis approximations of elliptic equations. In *Proceedings of the 11th World Congress on Computational Mathematics*, pages 4094–4102, 2014. URL arxiv.org/abs/1407.8005.

[29] T. Bui-Thanh. *Model-Constrained Optimization Methods for Reduction of Parameterized Large-Scale Systems*. PhD thesis, Massachusetts Institute of Technology, 2007. URL hdl.handle.net/1721.1/40305.

[30] T. Bui-Thanh, K. Willcox, O. Ghattas, and B. van Bloemen Waanders. Goal-oriented, model-constrained optimization for reduction of large-scale systems. *Journal of Computational Physics*, 224(2):880–896, 2007. DOI 10.1016/j.jcp.2006.10.026.

[31] T. Bui-Thanh, K. Willcox, and O. Ghattas. Model Reduction for Large-Scale Systems with High-Dimensional Parametric Input Space. *SIAM Journal on Scientific Computing*, 30(6):3270–3288, 2008. DOI 10.1137/070694855.

[32] T. Bui-Thanh, C. Burstedde, O. Ghattas, J. Martin, G. Stadler, and L.C. Wilcox. Extreme-scale UQ for Bayesian inverse problems governed by PDEs. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, pages 1–11, 2012. DOI 10.1109/SC.2012.56.

[33] O. Burdakov. A Greedy Algorithm for the Optimal Basis Problem. *BIT Numerical Mathematics*, 37(3):591–599, 1997. DOI 10.1007/BF02510241.

[34] M. Burger. A Note on Sparse Reconstruction Methods. *Journal of Physics: Conference Series*, 124(1):1–9, 2008. DOI 10.1088/1742-6596/124/1/012002.

[35] E. Celledoni, R.I. McLachlan, D.I. McLaren, B. Owren, G. Reinout, W. Quispel, and W.M. Wright. Energy-Preserving Runge-Kutta Methods. *ESAIM: Mathematical Modelling and Numerical Analysis*, 43:645–649, 2009. DOI 10.1051/m2an/2009020.

[36] Y. Chahlaoui, D. Lemonnier, A. Vandendorpe, and P. Van Dooren. Second-order balanced truncation. *Linear Algebra and its Applications*, 415(2–3):373–384, 2006. DOI 10.1016/j.laa.2004.03.032.

[37] S. Chaturantabut and D.C. Sorensen. Nonlinear Model Reduction via Discrete Empirical Interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, 2010. DOI 10.1137/090766498.

[38] J. Chen and Y. Saad. Lanczos Vectors versus Singular Vectors for Effective Dimension Reduction. *IEEE Transactions on Knowledge and Data Engineering*, 21(8):1091–1103, 2009. DOI 10.1109/TKDE.2008.228.

[39] P. Chen and C. Schwab. Adaptive Sparse Grid Model Order Reduction for Fast Bayesian Estimation and Inversion. In *Sparse Grids and Applications - Stuttgart 2014*, volume 109 of *Lecture Notes in Computational Science and Engineering*, pages 1–27. Springer, 2016. DOI 10.1007/978-3-319-28262-6_1.

[40] Y. Chen. Model Order Reduction for Nonlinear Systems. Master's thesis, Massachusetts Institute of Technology, 1999. URL hdl.handle.net/1721.1/9381.

[41] MORwiki Community. MORwiki - Model Order Reduction Wiki. modelreduction.org, 2016.

[42] Octave Community. GNU Octave (Version 4.0). www.gnu.org/software/octave, 2015.

[43] M. Condon. Model reduction of nonlinear systems. In *Proceedings in Applied Mathematics and Mechanics*, volume 7(1), pages 2130011–2130012, 2007. DOI 10.1002/pamm.200701084.

[44] M. Condon and R. Ivanov. Empirical Balanced Truncation of Nonlinear Systems. *Journal of Nonlinear Science*, 14(5):405–414, 2004. DOI 10.1007/s00332-004-0617-5.

[45] M. Condon and R. Ivanov. Model reduction of nonlinear systems. *COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, 23(2):547–557, 2004. DOI 10.1108/03321640410510730.

[46] K. Datta. The Matrix Equation $XA - BX = R$ and its Applications. *Linear Algebra and its Applications*, 109:91–105, 1988. DOI 10.1016/0024-3795(88)90200-5.

[47] O. David, L.M. Harrison, and K.J. Friston. Modelling event-related responses in the brain. *NeuroImage*, 25(3):756–770, 2005. DOI 10.1016/j.neuroimage.2004.12.030.

[48] O. David, S.J. Kiebel, L.M. Harrison, J. Mattout, J.M. Kilner, and K.J. Friston. Dynamic causal modeling of evoked responses in EEG and MEG. *NeuroImage*, 30(4):1255–1272, 2006. DOI 10.1016/j.neuroimage.2005.10.045.

[49] A. Davidson. Balanced Systems and Model Reduction. *Electronics Letters*, 22(10):531–532, 1986. DOI 10.1049/el:19860362.

[50] S.M. Djouadi. On the Optimality of the Proper Orthogonal Decomposition and Balanced Truncation. In *47th IEEE Conference on Decision and Control*, pages 4221–4226, 2008. DOI 10.1109/CDC.2008.4739458.

[51] I. Dones, S. Skogestad, and H.A. Preisig. Application of Balanced Truncation to Nonlinear Systems. *Industrial and Engineering Chemistry Research*, 50(17):10093–10101, 2011. DOI 10.1021/ie200706d.

[52] P.M. Van Dooren. Gramian Based Model Reduction of Large-Scale Dynamical Systems. In *Numerical Analysis 1999*, volume 420 of *Research Notes in Mathematics*, pages 231–247, 2000. URL citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.29.2500.

[53] F. Dopico and F. Uhlig. Computing matrix symmetrizers, part 2: New methods using eigendata and linear means; a comparison. *Linear Algebra and its Applications*, 504: 590–622, 2016. DOI 10.1016/j.laa.2015.06.031.

[54] S.N. Dorogovtsev and J.F.F. Mendes. *Evolution of Networks: From Biological Nets to the Internet and WWW*. Oxford University Press, 2003. ISBN 9780198515906. DOI 10.1093/acprof:oso/9780198515906.001.0001.

[55] M. Drohmann, B. Hassdonk, and M. Ohlberger. Reduced Basis Approximation for Nonlinear Parametrized Evolution Equations based on Empirical Operator Interpolation. *SIAM Journal on Scientific Computing*, 34(2):A937–A969, 2012. DOI 10.1137/10081157X.

[56] J.W. Eaton. GNU Octave and reproducible research. *Journal of Process Control*, 22(8, Ken Muske Special Issue):1433–1438, 2012. DOI 10.1016/j.jprocont.2012.04.006.

[57] J.W. Eaton, D. Bateman, S. Hauberg, and R. Wehbring. GNU Octave manual: a high-level interactive language for numerical computations. www.gnu.org/software/octave/octave.pdf, 2015.

[58] C. Eberle and C. Ament. Identifiability and online estimation of diagnostic parameters with in the glucose insulin homeostasis. *Biosystems*, 107(3):135–141, 2012. DOI 10.1016/j.biosystems.2011.11.003.

[59] L. Eldén. Algorithms for the Regularization of Ill-Conditioned Least Squares Problems. *BIT Numerical Mathematics*, 17(2):134–145, 1977. DOI 10.1007/BF01932285.

[60] H.W. Engl, M. Hanke, and A. Neubauer. *Regularization of Inverse Problems*, volume 375 of *Mathematics and its Applications*. Springer, 2000. ISBN 9780792361404.

[61] D.F. Enns. Model Reduction with Balanced Realizations: An Error Bound and a Frequency Weighted Generalization. In *IEEE Conference on Decision and Control*, volume 23, pages 127–132, 1984. DOI 10.1109/CDC.1984.272286.

[62] J. Falkenhain. Operatornormen für lineare dynamische Systeme. Technical report, Ruhr-Universität Bochum, 2011. URL homepage.ruhr-uni-bochum.de/Jan.Falkenhain/Dokumente/Operatornormen.pdf.

[63] K.V. Fernando. *Covariance and Gramian matrices in control and systems theory*. PhD thesis, University of Sheffield, 1982. URL ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.280627.

[64] K.V. Fernando and H. Nicholson. Minimality of SISO Linear Systems. In *Proceedings of the IEEE*, volume 70(10), pages 1241–1242, 1982. DOI 10.1109/PROC.1982.12460.

[65] K.V. Fernando and H. Nicholson. On the Structure of Balanced and Other Principal Representations of SISO Systems. *IEEE Transactions on Automatic Control*, 28(2): 228–231, 1983. DOI 10.1109/TAC.1983.1103195.

[66] K.V. Fernando and H. Nicholson. On the Cauchy Index of Linear Systems. *IEEE Transactions on Automatic Control*, 28(2):222–224, 1983. DOI 10.1109/TAC.1983.1103200.

[67] K.V. Fernando and H. Nicholson. On a Fundamental Property of the Cross-Gramian Matrix. *IEEE Transactions on Circuits and Systems*, 31(5):504–505, 1984. DOI 10.1109/TCS.1984.1085524.

[68] K.V. Fernando and H. Nicholson. Reachability, Observability, and Minimality of MIMO Systems. In *Proceedings of the IEEE*, volume 72(12), pages 1820–1821, 1984. DOI 10.1109/PROC.1984.13094.

[69] K.V. Fernando and H. Nicholson. On the Cross-Gramian for Symmetric MIMO Systems. *IEEE Transactions on Circuits and Systems*, 32(5):487–489, 1985. DOI 10.1109/TCS.1985.1085737.

[70] K.V.M. Fernando and H. Nicholson. Karhunen-Loéve expansion with reference to singular-value decomposition and separation of variables. *IEE Proceedings D, Control Theory and Applications*, 127(5):204–206, 1980. DOI 10.1049/ip-d.1980.0034.

[71] H.P. Flath, L.C. Wilcox, V. Akçelik, J. Hill, B. van Bloemen Waanders, and O. Ghattas. Fast Algorithms for Bayesian Uncertainty Quantification in Large-Scale Linear Inverse Problems Based on Low-Rank Partial Hessian Approximations. *SIAM Journal on Scientific Computing*, 33(1):407–432, 2011. DOI 10.1137/090780717.

[72] L. Fortuna and M. Fransca. *Optimal and Robust Control: Advanced Topics with MATLAB*. CRC Press, 2012. ISBN 9781466501911.

[73] L. Fortuna, A. Gallo, C. Guglielmino, and G. Nunnari. On the solution of a nonlinear matrix equation for MIMO symmetric realizations. *Systems and Control Letters*, 11 (1):79–82, 1988. DOI 10.1016/0167-6911(88)90115-6.

[74] B.A. Francis. *A Course in $\mathscr{H}_\infty$ Control Theory*, volume 88 of *Lecture Notes in Control and Information Sciences*. Springer, 1987. ISBN 9783540170693. DOI 10.1007/BFb0007371.

[75] M. Frangos, Y. Marzouk, K. Willcox, and B. van Bloemen Waanders. Surrogate and Reduced-Order Modeling: A Comparison of Approaches for Large-Scale Statistical Inverse Problems. In *Large-Scale Inverse Problems and Quantification of Uncertainty*, pages 123–149. John Wiley & Sons, Ltd, 2010. ISBN 9780470685853. DOI 10.1002/9780470685853.ch7.

[76] K.J. Friston. Bayesian Estimation of Dynamical Systems: An Application to fMRI. *NeuroImage*, 16(2):513–530, 2002. DOI 10.1006/nimg.2001.1044.

[77] K.J. Friston, L. Harrison, and W. Penny. Dynamic causal modelling. *NeuroImage*, 19 (4):1273–1302, 2003. DOI 10.1016/S1053-8119(03)00202-7.

[78] J.-B. Fu, H. Zhang, and Y.-X. Sun. Model reduction by minimizing information loss based on cross-Gramian matrix. *Journal of Zhejiang University (Engineering Science)*, 43(5):817–821, 2009. URL en.cnki.com.cn/Article_en/CJFDTOTAL-ZDZC200905005.htm.

[79] K. Fujimoto and J.M.A. Scherpen. On balanced truncation for symmetric nonlinear systems. In *International Symposium on Mathematical Theory of Networks and Systems*, volume 21, pages 1498–1502, 2014. URL hdl.handle.net/11370/48dc6589-2b34-473a-9e4f-1f1a47b3a892.

[80] D. Galbally, K. Fidkowski, K. Willcox, and O. Ghattas. Non-linear model reduction for uncertainty quantification in large-scale inverse problems. *International Journal for Numerical Methods in Engineering*, 81(12):1581–1608, 2010. DOI 10.1002/nme.2746.

[81] J.S. Garcia and J.C. Basilio. Computation of reduced-order models of multivariable systems by balanced truncation. *International Journal of Systems Science*, 33(10): 847–854, 2002. DOI 10.1080/00207720210000017308.

[82] D. Geffen. Parameter Identifiability of Biochemical Reaction Networks in Systems Biology. Master's thesis, Queen's University, 2008. URL hdl.handle.net/1974/1347.

[83] D. Geffen, R. Findeisen, M. Schliemann, F. Allgöwer, and M. Guay. Observability Based Parameter Identifiability for Biochemical Reaction Networks. In *Proceedings of the American Control Conference*, pages 2130–2135, 2008. DOI 10.1109/ACC.2008.4586807.

[84] J.E. Gentle. *Matrix Algebra: Theory, Computations, and Applications in Statistics*. Springer Texts in Statistics. Springer, 2007. ISBN 9780387708737. DOI 10.1007/978-0-387-70873-7. URL mason.gmu.edu/~jgentle/books/matbk.

[85] A. Gheondea and R.J. Ober. A Trace Formula For Hankel Operators. *Proceedings of the American Mathematical Society*, 127(7):2007–2012, 1999. DOI 10.1090/S0002-9939-99-04669-9.

[86] K. Glover. All optimal Hankel-norm approximations of linear multivariable systems and their L^∞-error bounds. *International Journal of Control*, 39(6):1115–1193, 1984. DOI 10.1080/00207178408933239.

[87] K. Glover and J.R. Partington. Bounds on the Achievable Accuracy in Model Reduction. In *Modelling, Robustness and Sensitivity Reduction in Control Systems*, volume 34, pages 95–118. Springer, 1987. DOI 10.1007/978-3-642-87516-8_7.

[88] G.H. Golub and C.F. Van Loan. *Matrix Computations*. John Hopkins University Press, 1996. ISBN 9780801854149. URL jhupbooks.press.jhu.edu/content/matrix-computations.

[89] S. Gottlieb, C.-W. Shu, and E. Tadmor. Strong Stability-Preserving High-Order Time Discretization Methods. *SIAM Review*, 43(1):89–112, 2001. DOI 10.1137/S003614450036757X.

[90] W.S. Gray and J.M.A. Scherpen. Hankel Operators and Gramians for Nonlinear Systems. In *Proceedings of the 37th IEEE Conference on Decision and Control*, volume 2, pages 1416–1421, 1998. DOI 10.1109/CDC.1998.758485.

[91] M. Gresser. Vergleich projektionsbasierter Methoden zur Modellreduktion von Kontrollproblemen. Bachelor's thesis, University of Ulm, 2014.

[92] M.S. Grewal and K. Glover. Identifiability of Linear and Nonlinear Dynamical Systems. *IEEE Transactions on Automatic Control*, 21(6):833–837, 1976. DOI 10.1109/TAC.1976.1101375.

[93] A. Griewank and A. Walther. Introduction to Automatic Differentiation. *Proceedings in Applied Mathematics and Mechanics*, 2(1):45–49, 2003. DOI 10.1002/pamm.200310012.

[94] L. Grüne and O. Junge. *Gewöhnliche Differentialgleichungen. Eine Einführung aus der Perspektive der dynamischen Systeme*. Bachelorkurs Mathematik. Vieweg+Teubner, 2009. ISBN 9783834803818. DOI 10.1007/978-3-8348-9261-4. URL dgl-buch.de.

[95] S. Gugercin, A.C. Antoulas, and C. Beattie. $\mathscr{H}_2$ Model Reduction for Large-Scale Linear Dynamical Systems. *SIAM Journal on Matrix Analysis and Applications*, 30(2): 609–638, 2008. DOI 10.1137/060666123.

[96] B. Haasdonk and M. Ohlberger. Reduced Basis Method for Finite Volume Approximations of Parametrized Linear Evolution Equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 42(2):277–302, 2008. DOI 10.1051/m2an:2008001.

[97] J. Hahn and T.F. Edgar. Reduction of nonlinear models using balancing of empirical gramians and Galerkin projections. In *Proceedings of the American Control Conference*, volume 4, pages 2864–2868, 2000. DOI 10.1109/ACC.2000.878734.

[98] J. Hahn and T.F. Edgar. A Gramian Based Approach to Nonlinearity Quantification and Model Classification. *Industrial and Engineering Chemistry Research*, 40(24):5724–5731, 2001. DOI 10.1021/ie010155v.

[99] J. Hahn and T.F. Edgar. Balancing Approach to Minimal Realization and Model Reduction of Stable Nonlinear Systems. *Industrial and Engineering Chemistry Research*, 41(9):2204–2212, 2002. DOI 10.1021/ie0106175.

[100] J. Hahn and T.F. Edgar. An improved method for nonlinear model reduction using balancing of empirical gramians. *Computers and Chemical Engineering*, 26(10):1379–1397, 2002. DOI 10.1016/S0098-1354(02)00120-5.

[101] J. Hahn, U. Kruger, and T.F. Edgar. Application of Model Reduction for Model Predictive Control. In *Proceedings of the IFAC World Congress*, volume 15(1), pages 632–637, 2000. DOI 10.3182/20020721-6-ES-1901.00634.

[102] J. Hahn, T.F. Edgar, and W. Marquardt. Controllability and observability covariance matrices for the analysis and order reduction of stable nonlinear systems. *Journal of Process Control*, 13(2):115–127, 2003. DOI 10.1016/S0959-1524(02)00024-0.

[103] P.C. Hansen and D.P. O'Leary. The Use of the L-curve in the Regularization of Discrete Ill-Posed Problems. *SIAM Journal on Scientific Computing*, 14(6):1487–1503, 1993. DOI 10.1137/0914086.

[104] M. Heinkenschloss, T. Reis, and A.C. Antoulas. Balanced truncation model reduction for systems with inhomogeneous initial conditions. *Automatica*, 47(3):559–564, 2011. DOI 10.1016/j.automatica.2010.12.002.

[105] J. Hespanha. *Linear Systems Theory*. Princeton University Press, 2009. ISBN 9780691140216. URL www.ece.ucsb.edu/~hespanha/linearsystems/.

[106] C. Himpe. Implementation and Analysis of Dynamic Causal Modeling for EEG/MEG Data. Diplomarbeit, University of Münster, 2011. DOI 10.6084/m9.figshare.1027354.v1. URL wwwmath.uni-muenster.de/num/publications/2011/Him11.

[107] C. Himpe. emgr - **Em**pirical **Gr**amian Framework (Version 3.9). gramian.de, 2016. DOI 10.5281/zenodo.46523.

[108] C. Himpe. optmor - **Opt**imization-based **M**odel **O**rder **R**eduction (Version: 2.5). gramian.github.io/optmor, 2016. DOI 10.5281/zenodo.46683.

[109] C. Himpe and M. Ohlberger. A Unified Software Framework for Empirical Gramians. *Journal of Mathematics*, 2013:1–6, 2013. DOI 10.1155/2013/365909.

[110] C. Himpe and M. Ohlberger. Cross-Gramian Based Combined State and Parameter Reduction for Large-Scale Control Systems. *Mathematical Problems in Engineering*, 2014:1–13, 2014. DOI 10.1155/2014/843869.

[111] C. Himpe and M. Ohlberger. Model Reduction for Complex Hyperbolic Networks. In *Proceedings of the European Control Conference*, pages 2739–2743, 2014. DOI 10.1109/ECC.2014.6862188.

[112] C. Himpe and M. Ohlberger. Combined State and Parameter Reduction. In *Proceedings in Applied Mathematics and Mechanics*, volume 14(1), pages 825–826, 2014. DOI 10.1002/pamm.201410393.

[113] C. Himpe and M. Ohlberger. The Empirical Cross Gramian for Parametrized Nonlinear Systems. In *Proceedings of the 8th Vienna International Conferenceon Mathematical Modelling - MATHMOD 2015*, volume 48(1), pages 727–728, 2015. DOI 10.1016/j.ifacol.2015.05.163.

[114] C. Himpe and M. Ohlberger. Data-driven combined state and parameter reduction for inverse problems. *Advanced in Computational Mathematics (Special Issue on Model Order Reduction of Parameterized Systems)*, 41(5):1343–1364, 2015. DOI 10.1007/s10444-015-9420-5.

[115] C. Himpe and M. Ohlberger. The Versatile Cross Gramian. In *ScienceOpen Posters*, volume MoRePas 3, 2015. DOI 10.14293/P2199-8442.1.SOP-MATH.PSAHPZ.v1.

[116] C. Himpe and M. Ohlberger. Accelerating the Computation of Empirical Gramians and Related Methods. In *5th International Workshop on Model Reduction in Reacting Flows (IWMRRF)*, 2015. DOI 10.5281/zenodo.46643.

[117] C. Himpe and M. Ohlberger. A note on the cross Gramian for non-symmetric systems. *System Science and Control Engineering*, 4(1):199–208, 2016. DOI 10.1080/21642583.2016.1215273.

[118] M. Hinze, M. Kunkel, A. Steinbrecher, and T. Stykel. Model order reduction of coupled circuit-device systems. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, 25(4):362–377, 2012. DOI 10.1002/jnm.840.

[119] M. Hladnik and M. Omladič. Spectrum of the Product of Operators. *Proceedings of the American Mathematical Society*, 102(2):300–302, 1988. DOI 10.1090/S0002-9939-1988-0920990-2.

[120] P. Holmes, J.L. Lumley, G. Berkooz, and C.W. Rowley. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge Monographs on Mechanics. Cambridge University Press, 2012. ISBN 9781107008250. DOI 10.1017/CBO9780511919701.

[121] Z. Hu, X. Zhao, H. Liu, and P. Shi. Nonlinear Analysis of the BOLD Signal. *EURASIP Journal on Advances in Signal Processing*, 2009:1–13, 2009. DOI 10.1155/2009/215409.

[122] The MathWorks Inc. MATLAB (Version 8.3, R2014a). mathworks.com/products/matlab, 2014.

[123] T.C. Ionescu and J.M.A. Scherpen. Cross Gramians for Nonlinear Systems. In *TC7 Conference on System Modeling and Optimization*, volume 23, pages 322–323, 2007. URL ifip2007.agh.edu.pl/abstracts/48.pdf.

[124] T.C. Ionescu and J.M.A. Scherpen. Nonlinear Cross Gramians and Gradient Systems. In *IEEE Conference on Decision and Control*, volume 46, pages 3745–3750, 2007. DOI 10.1109/CDC.2007.4434707.

[125] T.C. Ionescu and J.M.A. Scherpen. Nonlinear Cross Gramians. In *System Modeling and Optimization*, volume 312 of *IFIP Advances in Information and Communication Technology*, pages 293–306. Springer, 2009. DOI 10.1007/978-3-642-04802-9_16.

[126] T.C. Ionescu, K. Fujimoto, and J.M.A. Scherpen. The Cross Operator and the Singular Value Analysis for Nonlinear Symmetric Systems. In *Proceedings of the European Control Conference 2009*, pages 1565–1570, 2009. URL hdl.handle.net/11370/1bcfd3be-c006-49e0-bbbe-5c09ca055d87.

[127] T.C. Ionescu, K. Fujimoto, and J.M.A. Scherpen. Singular Value Analysis of Nonlinear Symmetric Systems. *Transactions on Automatic Control*, 56(9):2073–2086, 2011. DOI 10.1109/TAC.2011.2126630.

[128] A.C. Ionita and A.C. Antoulas. Data-Driven Parametrized Model Reduction in the Loewner Framework. *SIAM Journal on Scientific Computing*, 36(3):A984–A1007, 2014. DOI 10.1137/130914619.

[129] T. Jiang. Brainnetome: A new -ome to understand the brain and its disorders. *NeuroImage*, 80:263–272, 2013. DOI 10.1016/j.neuroimage.2013.04.002.

[130] R.K. Johnson. *The Elements of MATLAB Style*. Cambridge University Press, 2011. ISBN 9780521732581. DOI 10.1017/CBO9780511842290.

[131] P.T. Kabamba. Balanced Gains and Their Significance for L^2 Model Reduction. *IEEE Transactions on Automatic Control*, 30(7):690–693, 1985. DOI 10.1109/TAC.1985.1104017.

[132] R.E. Kalman. Mathematical Description of Linear Dynamical Systems. *Journal of the Sciety for Industrial and Applied Mathematics Series A Control*, 1(2):152–192, 1963. DOI 10.1137/0301010.

[133] E. Kamrani, A.N. Foroushani, M. Vaziripour, and M. Sawan. Detecting the Stable, Observable and Controllable States of the Human Brain Dynamics. *Open Journal of Medical Imaging*, 2(4):128–136, 2012. DOI 10.4236/ojmi.2012.24024.

[134] A. Keil and J.L. Gouzé. Model reduction of modular systems using balancing methods. Technical report, Technische Universität München, 2003. URL citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.485.2449.

[135] A.R. Kellems, D. Roos, N. Xiao, and S.J. Cox. Low-dimensional, morphologically accurate models of subthreshold membrane potential. *Journal of Computational Neuroscience*, 27:161–176, 2009. DOI 10.1007/s10827-008-0134-2.

[136] A. Khaki-Sedigh and A. Shahmansourian. Input-output pairing using balanced realisations. *Electronics Letters*, 32(21):2027–2028, 1996. DOI 10.1049/el:19961346.

[137] M. Köhler and J. Saak. FlexiBLAS - A flexible BLAS library with runtime exchangeable backends. Technical Report 284, LAPACK Working Note, 2013. URL netlib.org/lapack/lawnspdf/lawn284.pdf.

[138] M. Köhler and J. Saak. FlexiBLAS - A flexible BLAS library with runtime exchangeable backends (Version 1.3). www.mpi-magdeburg.mpg.de/projects/flexiblas, 2015.

[139] D. Krioukov, M. Kitsak, R.S. Sinkovits, D. Rideout, D. Meyer, and M. Boguñá. Network Cosmology. *Scientific Reports*, 2:1–6, 2012. DOI 10.1038/srep00793.

[140] K. Kunisch and S. Volkwein. Control of the Burgers Equation by a Reduced-Order Approach Using Proper Orthogonal Decomposition. *Journal of Optimization Theory and Applications*, 102(2):345–371, 1999. DOI 10.1023/A:1021732508059.

[141] K. Kunisch and S. Volkwein. Galerkin Proper Orthogonal Decomposition Methods for a General Equation in Fluid Dynamics. *SIAM Journal on Numerical Analysis*, 40(2):492–515, 2002. DOI 10.1137/S0036142900382612.

[142] S. Lall, J.E. Marsden, and S. Glavaški. Empirical Model Reduction of Controlled Nonlinear Systems. In *Proceedings of the 14th IFAC World Congress*, volume F, pages 473–478, 1999. URL resolver.caltech.edu/CaltechAUTHORS:20101007-154754737.

[143] S. Lall, J.E. Marsden, and S. Glavaški. A subspace approach to balanced truncation for model reduction of nonlinear control systems. *International Journal of Robust and Nonlinear Control*, 12(6):519–535, 2002. DOI 10.1002/rnc.657.

[144] J. Lam and B.D.O Anderson. L_1 impulse response error bound for balanced truncation. *Systems and Control Letters*, 18(2):129–137, 1992. DOI 10.1016/0167-6911(92)90017-M.

[145] A.J. Laub, L.M. Silverman, and M. Verma. A Note on Cross-Grammians for Symmetric Realizations. In *Proceedings of the IEEE*, volume 71(7), pages 904–905, 1983. DOI 10.1109/PROC.1983.12688.

[146] A.J. Laub, M.T. Heath, C. Paige, and R. Ward. Computation of System Balancing Transformations and Other Applications of Simultaneous Diagonalization Algorithms. *IEEE Transactions on Automatic Control*, 32(2):115–122, 1987. DOI 10.1109/TAC.1987.1104549.

[147] J.R. Leigh. *Functional Analysis and Linear Control Theory*, volume 156 of *Mathematics in Science and Engineering*. Academic Press, 1980. ISBN 9780486458137.

[148] R.J. LeVeque. *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*. SIAM, 2007. ISBN 9780898716290. DOI 10.1137/1.9780898717839.

[149] C. Lieberman and B. Van Bloemen Waanders. Hessian-Based Model Reduction Approach to Solving Large-Scale Source Inversion Problems. In *CSRI Summer Proceedings*, pages 37–48, 2007. URL csri.sandia.gov/Proceedings/CSRI2007.pdf#page=43.

[150] C. Lieberman and K. Willcox. Goal-Oriented Inference: Approach, Linear Theory, and Application to Advection Diffusion. *SIAM Journal on Scientific Computing*, 34(4): A1880–A1904, 2012. DOI 10.1137/110857763.

[151] C. Lieberman, K. Willcox, and O. Ghattas. Parameter and State Model Reduction for Large-Scale Statistical Inverse Problems. *SIAM Journal on Scientific Computing*, 32 (5):2523–2542, 2010. DOI 10.1137/090775622.

[152] C.E. Lieberman. Parameter and State Model Reduction for Bayesian Statistical Inverse Problems. Master's thesis, Massachusetts Institute of Technology, 2009. URL hdl.handle.net/1721.1/54213.

[153] C.E. Lieberman. *Goal-oriented inference : Theoretical foundations and application to carbon capture and storage*. PhD thesis, Massachusetts Institute of Technology, 2013. URL hdl.handle.net/1721.1/82472.

[154] C.E. Lieberman, K. Fidkowski, K. Willcox, and B. Van Bloemen Waanders. Hessian-based model reduction: large-scale inversion and prediction. *International Journal for Numerical Methods in Fluids*, 71(2):135–150, 2013. DOI 10.1002/fld.3650.

[155] W.Q. Liu, V. Sreeram, and K.L. Teo. Model reduction for state-space symmetric systems. *Systems and Control Letters*, 34(4):209–215, 1998. DOI 10.1016/S0167-6911(98)00024-3.

[156] Y.Y. Liu, J.J. Slotine, and A.L. Barabási. Controllability of complex networks. *Nature*, 473:167–173, 2011. DOI 10.1038/nature10011.

[157] Y.Y. Liu, J.J. Slotine, and A.L. Barabási. Observability of complex systems. *Proceedings of the National Academy of Sciences*, 110(7):2460–2465, 2013. DOI 10.1073/pnas.1215508110.

[158] H.P. Löffler and W. Marquardt. Order reduction of non-linear differential-algebraic process models. *Journal of Process Control*, 1(1):32–40, 1991. DOI 10.1016/0959-1524(91)87005-I.

[159] X. Ma and J.A. De Abreu-Garcia. On the Computation of Reduced Order Models of Nonlinear Systems using Balancing Technique. In *Proceedings of the 27th IEEE Conference on Decision and Control*, volume 2, pages 1165–1166, 1988. DOI 10.1109/CDC.1988.194502.

[160] D. La Manna, A. Fagiolini, and F. Pasqualetti. Attack and Defense Location in Line Network Systems. In *Preprint*, pages 1–6, 2016. URL www.fabiopas.it/papers/DLM-AF-FP-15.pdf.

[161] W. Marquardt. Nonlinear Model Reduction for Optimization Based Control of Transient Chemical Processes. In *AIChE Symposium Series*, pages 12–42, 2002. URL hdl.handle.net/10068/132792.

[162] S. Melchior, V. Legat, and P. Van Dooren. Gramian Based Model Reduction of Nonlinear MIMO Systems. In *Mathematical Theory of Networks and Systems*, 2012. URL www.mtns2012.conference.net.au/Extended%20Abstract/MTNS2012_0170_paper.pdf.

[163] H.B. Minh and C. Batlle. $\mathscr{H}_2$ optimal model reduction - Wilson's conditions for the cross-Gramian. Technical report, Hanoi University of Science and Technology, Universitat Politécnica de Catalunya, 2013. URL www-ma4.upc.edu/~carles/fitxers/h2_cross_gramian_v3.pdf.

[164] L.A. Mironovskii and T.N. Solov'eva. Analysis of Multiplicity of Hankel Singular Values of Control Systems. *Automation and Remote Control*, 76(2):205–218, 2015. DOI 10.1134/S0005117915020022.

[165] B. Moaveni and A. Khaki-Sedigh. Input-Output Pairing based on Cross-Gramian Matrix. In *International Joint Conference SICE-ICAS*, pages 2378–2380, 2006. DOI 10.1109/SICE.2006.314989.

[166] B. Moaveni and A. Khaki-Sedigh. A New Approach to Compute the Cross-Gramian Matrix and its Application in Input-Output Pairing of Linear Multivariable Plants. *Journal of Applied Sciences*, 8(4):608–614, 2008. DOI 10.3923/jas.2008.608.614.

[167] B. Moaveni and A. Khaki-Sedigh. Input-output pairing analysis for uncertain multivariable processes. *Journal of Process Control*, 18(6):527–532, 2008. DOI 10.1016/j.jprocont.2007.10.015.

[168] B. Moore. Principal Component Analysis in Linear Systems: Controllability, Observability, and Model Reduction. *IEEE Transactions on Automatic Control*, 26(1):17–32, 1981. DOI 10.1109/TAC.1981.1102568.

[169] R.J. Moran, S.J. Kiebel, K.E. Stephan, R.B. Reilly, J. Daunizeau, and K.J. Friston. A neural mass model of spectral responses in electrophysiology. *NeuroImage*, 37(3): 706–720, 2007. DOI 10.1016/j.neuroimage.2007.05.032.

[170] N.H. Nguyen, B.C. Khoo, and K. Willcox. Model order reduction for Bayesian approach to inverse problems. *Asia Pacific Journal on Computational Engineering*, 1(2):1–17, 2014. DOI 10.1186/2196-1166-1-2.

[171] G. Obinata and B.D.O. Anderson. *Model Reduction for Control System Design*. Communications and Control Engineering. Springer, 2001. ISBN 9781447102830. DOI 10.1007/978-1-4471-0283-0.

[172] M. Ohlberger and F. Schindler. Error Control for the Localized Reduced Basis Multiscale Method with Adaptive On-Line Enrichment. *SIAM Journal on Scientific Computing*, 37(6):A2865–A2895, 2015. DOI 10.1137/151003660.

[173] M.R. Opmeer and T. Reis. The balanced truncation error bound in Schatten norms. *Hamburger Beiträge zur Angewandten Mathematik*, 2014(1):1–6, 2014. URL edok01.tib.uni-hannover.de/edoks/e01fn14/788530836.pdf.

[174] U. Pallaske. Ein Verfahren zur Ordnungsreduktion mathematischer Prozessmodelle. *Chemie Ingenieur Technik*, 59(7):604–605, 1987. DOI 10.1002/cite.330590720.

[175] L. Pernebo and L. Silverman. Model Reduction via Balanced State Space Representations. *IEEE Transactions on Automatic Control*, 27(2):382–387, 1982. DOI 10.1109/TAC.1982.1102945.

[176] J.R. Phillips and L.M. Silveira. Poor Man's TBR: A Simple Model Reduction Scheme. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Ssystems*, 24(1): 43–55, 2005. DOI 10.1109/TCAD.2004.839472.

[177] J.R. Phillips, L. Daniel, and L.M. Silveira. Guaranteed Passive Balancing Transformations for Model Order Reduction. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 22(8):1027–1041, 2003. DOI 10.1109/TCAD.2003.814949.

[178] Y. Quan, H. Zhang, and L. Cai. Modeling and Control Based on a New Neural Network Model. In *Proceedings of the American Control Conference*, volume 3, pages 1928–1929, 2001. DOI 10.1109/ACC.2001.946022.

[179] Å. Björck. *Numerical Methods for Least Squares Problems*. SIAM, 1996. ISBN 9780898713602. DOI 10.1137/1.9781611971484.

[180] S. Rahrovani, M.K. Vakilzadeh, and T. Abrahamsson. On Gramian-Based Techniques for Minimal Realization of Large-Scale Mechanical Systems. In *Topics in Modal Analysis*, volume 7, pages 797–805, 2014. DOI 10.1007/978-1-4614-6585-0_75.

[181] A. Ralston. Runge-Kutta Methods with Minimum Error Bounds. *Mathematics of Computation*, 16(80):431–437, 1962. DOI 10.2307/2003133.

[182] T. Reis. Mathematical Modeling and Analysis of Nonlinear Time-Invariant RLC Circuits. In *Large-Scale Networks in Engineering and Life Sciences*, Modeling and Simulation in Science, Engineering and Technology, pages 125–198. Springer, 2014. DOI 10.1007/978-3-319-08437-4_2.

[183] T. Reis and T. Stykel. Balanced truncation model reduction of second-order systems. *Mathematical and Computer Modelling of Dynamical Systems: Methods, Tools and Applications in Engineering and Related Sciences*, 14(5):391–406, 2008. DOI 10.1080/13873950701844170.

[184] C.W. Rowley. Model Reduction for Fluids, Using Balanced Proper Orthogonal Decomposition. *International Journal of Bifurcation and Chaos*, 15(3):997–1013, 2005. DOI 10.1142/S0218127405012429.

[185] C.W. Rowley, T. Colonius, and R.M. Murray. Model reduction for compressible flows using POD and Galerkin projection. *Physica D: Nonlinear Phenomena*, 189(1–2):115–129, 2004. DOI 10.1016/j.physd.2003.03.001.

[186] G. Rozza, D.B.P. Huynh, and A.T. Patera. Reduced Basis Approximation and a Posteriori Error Estimation for Affinely Parametrized Elliptic Coercive Partial Differential Equations. *Archives of Computational Methods in Engineering*, 15:229–275, 2008. DOI 10.1007/s11831-008-9019-9.

[187] J. Saak. *Efficient Numerical Solution of Large Scale Algebraic Matrix Equations in PDE Control and Model Order Reduction*. PhD thesis, Technische Universität Chemnitz, 2009. URL nbn-resolving.de/urn:nbn:de:bsz:ch1-200901642.

[188] J. Saak, P. Benner, and P. Kürschner. A Goal-Oriented Dual LRCF-ADI for Balanced Truncation. In *Proceedings of the 7th Vienna International Conference on Mathematical Modelling*, volume 45(2), pages 752–757, 2012. DOI 10.3182/20120215-3-AT-3016.00133.

[189] J.M.A. Scherpen. Balancing for nonlinear systems. *Systems and Control Letters*, 21 (2):143–153, 1993. DOI 10.1016/0167-6911(93)90117-O.

[190] J.M.A. Scherpen. *Balancing for Nonlinear Systems*. PhD thesis, University of Twente, 1994. URL primo-utwente.hosted.exlibrisgroup.com/TWE2:All_resources:TWENTE_MARC11427181X.

[191] J.M.A. Scherpen and A.J. van der Schaft. Balanced model reduction of gradient systems. In *Proceedings of the 18th IFAC World Congress*, volume 44(1), pages 12745–12750, 2011. DOI 10.3182/20110828-6-IT-1002.03550.

[192] J.M.A. Scherpen, K. Fujimoto, and W.S. Gray. On Adjoints and Singular Value Functions for Nonlinear Systems. In *Conference on Information Sciences and Systems*, 2000. URL hdl.handle.net/11370/174d84d2-33a6-46db-bc01-df00d5684230.

[193] B. Schölkopf, A. Smola, and K.-R. Müller. Kernel Principal Component Analysis. In *Artificial Neural Networks ICANN'97*, volume 1327 of *Lecture Notes in Computer Science*, pages 583–588. Springer, 1997. DOI 10.1007/BFb0020217.

[194] M.L. Seghier and K.J. Friston. Network discovery with large DCMs. *NeuroImage*, 68: 181–191, 2013. DOI 10.1016/j.neuroimage.2012.12.005.

[195] H.R. Shaker. Generalized Cross-Gramian for Linear Systems. In *7th IEEE Conference on Industrial Electronics and Applications*, pages 749–751, 2012. DOI 10.1109/ICIEA.2012.6360824.

[196] H.R. Shaker and M. Komareji. Control Configuration Selection for Multivariable Nonlinear Systems. *Industrial and Engineering Chemistry Research*, 51(25):8583–8587, 2012. DOI 10.1021/ie301137k.

[197] H.R. Shaker and J. Stoustrup. An interaction measure for control configuration selection for multivariable bilinear systems. *Nonlinear Dynamics*, 72(1):165–174, 2013. DOI 10.1007/s11071-012-0700-z.

[198] S. Shokoohi, L.M. Silverman, and P. Van Dooren. Linear Time-Variable Systems: Balancing and Model Reduction. *IEEE Transactions on Automatic Control*, 28(8):810–822, 1983. DOI 10.1109/TAC.1983.1103331.

[199] A.K. Singh and J. Hahn. Optimal Sensor Location for Nonlinear Dynamic Systems via Empirical Gramians. In *Proceedings of Dynamics and Control of Process Systems*, volume 7, pages 965–970, 2004. URL www.nt.ntnu.no/users/skoge/prost/proceedings/dycops04/pdffiles/papers/17.pdf.

[200] A.K. Singh and J. Hahn. On the Use of Empirical Gramians for Controllability and Observability Analysis. In *Proceedings of the American Control Conference*, volume 2005, pages 140–141, 2005. DOI 10.1109/ACC.2005.1469922.

[201] A.K. Singh and J. Hahn. Determining Optimal Sensor Locations for State and Parameter Estimation for Stable Nonlinear Systems. *Industrial and Engineering Chemistry Research*, 44(15):5645–5659, 2005. DOI 10.1021/ie040212v.

[202] A.K. Singh and J. Hahn. State estimation for high-dimensional chemical processes. *Computers and Chemical Engineering*, 29(11–12):2326–2334, 2005. DOI 10.1016/j.compchemeng.2005.05.009.

[203] A.K. Singh and J. Hahn. Sensor Location for Stable Nonlinear Dynamic Systems: Multiple Sensor Case. *Industrial and Engineering Chemistry Research*, 45(10):3615–3623, 2006. DOI 10.1021/ie0511175.

[204] A.K. Singh and J. Hahn. Computing Sensor Locations for Nonlinear Systems Under the Influence of Disturbances. In *AIChE Annual Meeting*, 2006. URL www.nt.ntnu.no/users/skoge/prost/proceedings/aiche-2006/data/papers/P50021.pdf.

[205] L. Sirovich. Turbulence and the Dynamics of Coherent Structures Part I: Coherent Structures. *Quarterly of Applied Mathematics*, 45(3):561–571, 1987. URL www.jstor.org/stable/43637457.

[206] S.C. Smith and J. Fisher. On generating random systems: a gramian approach. In *Proceedings of the American Control Conference*, volume 3, pages 2743–2748, 2003. DOI 10.1109/ACC.2003.1243494.

[207] S.A. Solovyev and S. Tordeux. Large SVD Computations for Analysis of Inverse Problems in Geophysics. In *Proceedings of the WCCM XI - ECCM V - ECFD VI*, pages 2861–2869, 2014. URL www.wccm-eccm-ecfd2014.org/admin/files/filePaper/p2861.pdf.

[208] S.A. Solovyev and S. Tordeux. Compute SVD of a Very Large Matrix in the Context of Geological Prospection. In *6th EAGE Saint Petersburg International Conference and Exhibition*, 2014. DOI 10.3997/2214-4609.20140190.

[209] D.C. Sorensen and A.C. Antoulas. Projection methods for balanced model reduction. Technical report, Rice University, 2001. URL www.caam.rice.edu/caam/trs/2001/TR01-03.pdf.

[210] D.C. Sorensen and A.C. Antoulas. The Sylvester equation and approximate balanced reduction. *Linear Algebra and its Applications*, 351–352:671–700, 2002. DOI 10.1016/S0024-3795(02)00283-5.

[211] O. Sporns. Brain connectivity. *Scholarpedia*, 2(10):4695, 2007. DOI 10.4249/scholarpedia.4695.

[212] V. Sreeram and P. Agathoklis. On the Properties of the Gram Matrix. *IEEE transactions on Circuits and Systems I: Fundamental Theory and Applications*, 41(3):234–237, 1994. DOI 10.1109/81.273922.

[213] K.E. Stephan and K.J. Friston. Models of Effective Connectivity in Neural Systems. In *Handbook of Brain Connectivity*, Understanding Complex Systems, pages 303–327. Springer, 2007. DOI 10.1007/978-3-540-71512-2_10.

[214] S. Streif, R. Findeisen, and E. Bullinger. Relating Cross Gramians and Sensitivity Analysis in Systems Biology. *Theory of Networks and Systems*, 10.4:437–442, 2006. URL hdl.handle.net/2268/130730.

[215] S. Streif, S. Waldherr, F. Allgöwer, and R. Findeisen. Steady State Sensitivity Analysis of Biochemical Reaction Networks: A Brief Review and New Methods. In *Methods*

in Bioengineering, Systems Analysis of Biological Networks, pages 129–148. Artech House MIT Press, 2009. URL books.google.de/books?id=Haod3KR-tR8C.

[216] T.H. Summers, F.L. Cortesi, and J. Lygeros. On Submodularity and Controllability in Complex Dynamical Networks. *IEEE Transactions on Control of Network Systems*, PP (99):1–11, 2015. DOI 10.1109/TCNS.2015.2453711.

[217] C. Sun and J. Hahn. Parameter reduction for stable dynamical systems based on Hankel singular values and sensitivity analysis. *Chemical Engineering Science*, 61(16): 5393–5403, 2006. DOI 10.1016/j.ces.2006.04.027.

[218] C. Sun and J. Hahn. Model reduction in the presence of uncertainty in model parameters. *Journal of Process Control*, 16(6):645–649, 2006. DOI 10.1016/j.jprocont.2005.10.001.

[219] C. Sun and J. Hahn. Nonlinear Model Reduction Routines for MATLAB. Technical report, Rensselaer Polytechnic Institute, 2006. URL www.hahnresearchgroup.com/downloads/nonlinear-model-reduction-routines-matlab.

[220] C. Teng. Second-Order Model Reduction Based on Gramians. *Journal of Control Science and Engineering*, 2012:1–9, 2012. DOI 10.1155/2012/302498.

[221] G. Thimm, P. Moerland, and E. Fiesler. The Interchangeability of Learning Rate and Gain in Backpropagation Neural Networks. *Neural Computation*, 8(2):451–460, 1996. DOI 10.1162/neco.1996.8.2.451.

[222] M.S. Tombs and I. Postlethwaite. Truncated balanced realization of stable non-minimal state-space system. *International Journal of Control*, 46(4):1319–1330, 1987. DOI 10.1080/00207178708933971.

[223] R. Toscano. *Signal and System Norms*, pages 25–44. Advances in Industrial Control. Springer, 2013. ISBN 9781447151876. DOI 10.1007/978-1-4471-5188-3_2.

[224] F.W.J. van den Berg, H.C.J. Hoefsloot, H.F.M. Boelens, and A.K. Smilde. Selection of optimal sensor position in a tubular reactor using robust degree of observability criteria. *Chemical Engineering Science*, 55(4):827–837, 2000. DOI 10.1016/S0009-2509(99)00360-7.

[225] K. Veroy, C. Prud'homme, D.V. Rovas, and A.T. Patera. A Posteriori Error Bounds for Reduced-Basis Approximation of Parametrized Noncoercive and Nonlinear Elliptic Partial Differential Equations. In *Proceedings of the 16th AIAA computational fluid dynamics conference*, pages 1–18, 2003. DOI 10.2514/6.2003-3847.

[226] E. W. Weisstein. Havercosine. Technical report, From MathWorld–A Wolfram Web Resource, 2016. URL mathworld.wolfram.com/Havercosine.html.

[227] A.J. Whalen, S.N. Brennan, T.D. Sauer, and S.J. Schiff. Observability and Controllability of Nonlinear Networks: the Role of Symmetry. *Physical Review X*, 5(1):011005, 2015. DOI 10.1103/PhysRevX.5.011005.

[228] K. Willcox and J. Peraire. Balanced Model Reduction via the Proper Orthogonal Decomposition. *AIAA Journal*, 40(11):2323–2330, 2002. DOI 10.2514/2.1570.

[229] K. Willcox, O. Ghattas, B. van Bloemen Waanders, and B. Bader. An Optimization Framework for Goal-Oriented, Model-Based Reduction of Large-Scale Systems. In *44th IEEE Conference on Decision and Control and European Control Conference*, pages 2265–2271, 2005. DOI 10.1109/CDC.2005.1582499.

[230] M. Wu, B. Yin, A. Vosoughi, C. Studer, J.R. Cavallaro, and C. Dick. Approximate Matrix Inversion for High-Throughput Data Detection in the Large-Scale MIMO Uplink. In *IEEE International Symposium of Circuits and Systems*, pages 2155–2158, 2013. DOI 10.1109/ISCAS.2013.6572301.

[231] B. Yan, H. Wang, and S.X.-D Tan. A Survey of RLCK Reduction and Simulation Methods by Fast Truncated Balanced Realization. In *Solid-State and Integrated-Circuit Technology*, volume 9, pages 2236–2239, 2008. DOI 10.1109/ICSICT.2008.4735035.

[232] W.H. Young. On the Multiplication of Successions of Fourier Constants. *Proceedings of the Royal Society of London, Series A*, 87(596):331–339, 1912. URL jstor.org/stable/93120.

[233] M. Zhanfeng and H. Chao. Structure-preserving balanced truncation for flexible spacecraft using cross Gramian. *Journal of Beijing University of Aeronautics and Astronautics*, 34(12):1437–1440, 2008. URL en.cnki.com.cn/Article_en/CJFDTOTAL-BJHK200812017.htm.