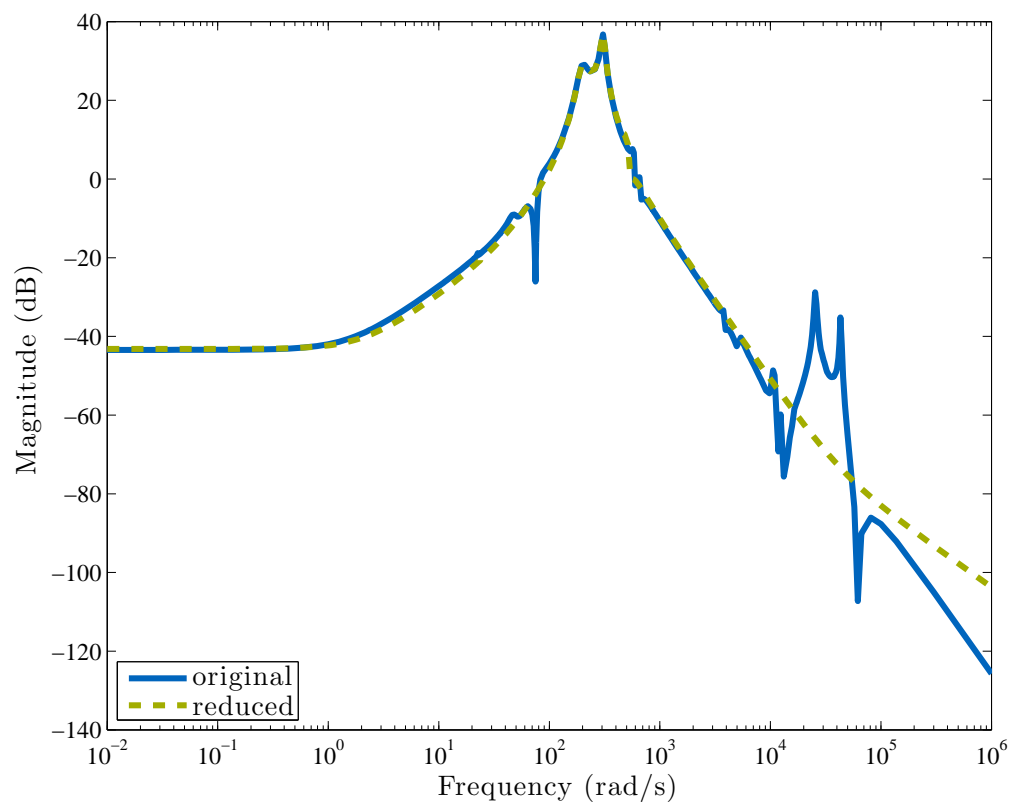


Modeling and Reduction of Complex Systems

Part B: Introduction to Model Order Reduction

Maria Cruz Varona, Alessandro Castagnotto, Thomas Wolf, Rudy Eid,
Mikhail Pak

Last change: July 30, 2019



*Das bewußte Reduzieren, das Weglassen, das Vereinfachen hat eine tiefe ethische Grundlage:
Nie kann etwas zuwider sein, was einfach ist.* Egon Eiermann

*The deliberate reduction, the omission, simplifying has a deep ethical foundation:
Never can anything/anyone be against to what is simple.* Egon Eiermann

Contents

1	Introduction	7
1.1	What is Model Order Reduction?	7
1.2	Which models are reduced?	7
1.3	Why Model Order Reduction?	9
1.4	Where do large models come from?	9
1.5	Goals of Model Order Reduction	10
1.6	Model Order Reduction Software	11
2	Mathematical Fundamentals	13
2.1	Vector space	13
2.2	Linear dependence and rank	14
2.3	Basis	15
2.4	Range and nullspace	16
2.5	Singular Value Decomposition (SVD)	17
2.5.1	Properties of the SVD	18
2.5.2	Matrix approximation	18
2.5.3	(Further) Applications of the SVD	19
2.6	Norms	19

3	Projective Model Order Reduction	23
3.1	The projector	23
3.2	Generating projectors	24
3.3	Projective MOR	27
3.4	Summary	30
4	Modal Reduction	31
4.1	Modal transformation	31
4.2	Dominance measure according to Litz	32
4.3	Modal truncation	33
4.4	Reduction of unstable systems	35
4.5	Residualization	35
4.6	Properties of modal truncation	36
4.7	Modal truncation for second-order systems	38
5	Balanced Truncation	41
5.1	Controllability	42
5.1.1	Controllability Gramian	43
5.1.2	Energy consideration	43
5.2	Observability	44
5.2.1	Observability Gramian	44
5.2.2	Energy consideration	45
5.3	Balancing	45
5.4	Truncated Balanced Realization	47
5.5	Properties of Balanced Truncation	50

6	Krylov Subspace Methods	53
6.1	Moment Matching	53
6.1.1	Moments	54
6.1.2	Krylov subspaces	55
6.1.3	Implicit Moment Matching	55
6.1.4	Markov parameters	56
6.1.5	Several expansion points	57
6.1.6	Summary	57
6.2	Computation of the matrices \mathbf{V} and \mathbf{W}	58
6.2.1	Gram-Schmidt method	58
6.2.2	Arnoldi algorithm	59
6.3	\mathcal{H}_2 -optimal reduction	62
6.4	Properties of Krylov subspace methods	63
6.5	Comparison between TBR and Krylov	65
7	Further Topics	67
7.1	Systems in implicit state-space representation	67
7.2	Balanced Truncation for $\mathbf{E} \neq \mathbf{I}$	69
	References	73

Chapter 1

Introduction

1.1 What is Model Order Reduction?

By *Model Reduction* or *Model Order Reduction* (MOR), the number of describing equations of a dynamic system shall be significantly reduced, *without* considerably changing the characteristic behavior. This is an *approximation*, i. e. a certain error occurs.

1.2 Which models are reduced?

There are different classes of mathematical models to describe dynamic systems (for which different methods of model reduction are needed). Here, a few classes are listed.

First-order linear time-invariant (LTI) systems: In general, the modeling of such systems results in an *implicit* state-space representation of the form

$$\begin{aligned}\mathbf{E}\dot{\mathbf{x}}(t) &= \tilde{\mathbf{A}}\mathbf{x}(t) + \tilde{\mathbf{B}}\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t),\end{aligned}\tag{1.1}$$

where $\mathbf{E}, \tilde{\mathbf{A}} \in \mathbb{R}^{n \times n}$, $\tilde{\mathbf{B}} \in \mathbb{R}^{n \times m}$, $\mathbf{C} \in \mathbb{R}^{p \times n}$ and $\mathbf{D} \in \mathbb{R}^{p \times m}$. Hereby, $\mathbf{x}(t) \in \mathbb{R}^n$ denotes the state vector, $\mathbf{u}(t) \in \mathbb{R}^m$ the inputs and $\mathbf{y}(t) \in \mathbb{R}^p$ the outputs of the model. Provided that the *descriptor matrix* \mathbf{E} is regular¹, the state differential equation can be brought into the well-known *explicit* form by multiplication with \mathbf{E}^{-1} :

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t).\end{aligned}\tag{1.2}$$

¹Models with singular \mathbf{E} matrix are generally called *descriptor systems* or *Differential-Algebraic Equations* (DAE). These include additional algebraic equations/constraints, making analysis and reduction a little more difficult. The bibliography at the end of the script contains some references regarding this topic.

This representation is theoretically equivalent to (1.1) (in particular: neither the transfer behavior nor the solution $\mathbf{x}(t)$ are changed by it), so that we will only examine the explicit representation in the following. At this point, however, it is emphasized that the implicit representation (1.1) has considerable advantages for the numerics².

The “characteristic behavior” is the input-output behavior, described in frequency-domain by the transfer function matrix $\mathbf{G}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}$ from input $\mathbf{u}(t)$ to output $\mathbf{y}(t)$. Please note that – by model order reduction – only the number of state equations/variables is reduced. The number of inputs and outputs remains the same.

Second-order (LTI) systems:

$$\begin{aligned}\mathbf{M}\ddot{\mathbf{z}}(t) + \mathbf{D}\dot{\mathbf{z}}(t) + \mathbf{K}\mathbf{z}(t) &= \mathbf{g}u(t), \\ y(t) &= \mathbf{I}^T\mathbf{z}(t),\end{aligned}\tag{1.3}$$

where \mathbf{M} , \mathbf{D} , $\mathbf{K} \in \mathbb{R}^{n \times n}$ are the mass, damping and stiffness matrices. A second-order system may be reformulated as a first-order system with $\mathbf{x} = [\mathbf{z}^T, \dot{\mathbf{z}}^T]^T$. Nevertheless, the definiteness properties of the matrices \mathbf{M} , \mathbf{D} , \mathbf{K} can be much easily preserved by a direct reduction of the second-order model (1.3) instead of using the reformulated first-order model.

Port-Hamiltonian (LTI) systems:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= (\mathbf{J} - \mathbf{R})\nabla H(\mathbf{x}(t)) + \mathbf{b}u(t), \\ y(t) &= \mathbf{b}^T\nabla H(\mathbf{x}(t)).\end{aligned}\tag{1.4}$$

The aim of MOR here is to reduce the number of state equations and preserve the port-Hamiltonian structure.

Parametric (LTI) systems: Depend on d (material, geometry) parameters $\mathbf{p} = [p_1, p_2, \dots, p_d]^T$.

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}(\mathbf{p})\mathbf{x}(t) + \mathbf{B}(\mathbf{p})\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{C}(\mathbf{p})\mathbf{x}(t) + \mathbf{D}(\mathbf{p})\mathbf{u}(t).\end{aligned}\tag{1.5}$$

The main goal of parametric model order reduction (pMOR) is to preserve the parameter dependency in the reduced model, thus allowing for a variation of the parameters without the need to repeat the reduction step for every new parameter set.

Nonlinear (time-invariant) systems:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), u(t)), \\ y(t) &= \mathbf{h}(\mathbf{x}(t), u(t)).\end{aligned}\tag{1.6}$$

Here, the main tasks are (1) to reduce the number of state equations via *dimensional reduction* and (2) to speed-up the evaluation of the nonlinear terms via so-called *hyper reduction* methods.

In this lecture, we restrict ourselves to LTI systems in explicit representation with only one input and one output ($m = p = 1$), i. e. single-input, single-output (SISO). The presented methods, however, can be generalized to the multiple-input, multiple-output (MIMO) case.

²A short explanation can be found in Sec. 7.1.

1.3 Why Model Order Reduction?

In principle: If the system order/dimension is too high!

Simulation: If the system order is too high, a simulation becomes time-consuming (takes e.g. several days) or even impossible (due to limited working memory).

Control: The control of high-order plants can have the following disadvantages:

- **Complexity:** The control design becomes the more complicated/more untransparent, the higher the order of the plant is (exception: output-feedback).
- **Computation time:** The calculation of the control input could take longer than the sampling time allows; this may lead from reduced control quality to even instabilities.
- **Hardware:** The higher the order of a controller, the more powerful hardware is needed.

Optimization: For each optimization step and each parameter update, the output to be optimized must be recalculated, i.e. the dynamic system must be simulated again. Complex optimization problems typically require several thousand iterations to converge. This can lead, in the case of high-order models, to several-day simulations in order to find an optimal solution for the problem.

1.4 Where do large models come from?

- Spatial discretization of partial differential equations (PDEs), using e.g. the finite element (FEM), finite difference (FDM) or finite volume (FVM) method.

Example: The heat conduction equation for the temperature $T(x, t)$ along a one-dimensional rod is:

$$\frac{\partial T(x, t)}{\partial t} = \frac{\partial^2 T(x, t)}{\partial^2 x} + u(x, t). \quad (1.7)$$

A numerical approximation can be obtained by spatial discretization, i.e. by evaluating the PDE (1.7) at specific points (x_1, x_2, \dots, x_n) . When the state vector $\mathbf{x} = [T(x_1, t), T(x_2, t), \dots, T(x_n, t)]^T$ is introduced, a LTI system of the type (1.2) is obtained.

- The modeling of technical systems consisting of a large number of individual components. The most prominent example are integrated circuits with a high degree of integration (see also *Very Large Scale Integration*) with hundreds of thousands to millions of transistors.

1.5 Goals of Model Order Reduction

Given: Full Order Model (FOM)

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{b}u(t), & \mathbf{x}(t=0) &= \mathbf{x}_0 \\ y(t) &= \mathbf{c}^T\mathbf{x}(t),\end{aligned}\tag{1.8}$$

with $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{b} \in \mathbb{R}^n$, $\mathbf{c}^T \in \mathbb{R}^{1 \times n}$ and $n \approx 10^2 \dots 10^6$.

Wanted: Reduced Order Model (ROM)

$$\begin{aligned}\dot{\mathbf{x}}_r(t) &= \mathbf{A}_r\mathbf{x}_r(t) + \mathbf{b}_r u(t), & \mathbf{x}_r(t=0) &= \mathbf{x}_{r,0} \\ y_r(t) &= \mathbf{c}_r^T\mathbf{x}_r(t),\end{aligned}\tag{1.9}$$

of order $q \ll n$. A model order reduction procedure should provide the reduced quantities \mathbf{A}_r , \mathbf{b}_r and \mathbf{c}_r^T under consideration of the following objectives:

1. **Good approximation:** The reduction technique should yield a ROM, which captures the most dominant dynamics and well approximates the state vector or input-output behavior of the FOM either in (a specific interesting) time- or frequency-domain. The approximation quality can, for instance, be measured point-wise in time by $\|\mathbf{y}(t) - \mathbf{y}_r(t)\|_{(\cdot)}$ or $\|\mathbf{x}(t) - \mathbf{V}\mathbf{x}_r(t)\|_{(\cdot)}$, or point-wise in frequency by $\|\mathbf{G}(i\omega) - \mathbf{G}_r(i\omega)\|_{(\cdot)}$ using suitable matrix and vector norms $(\cdot) = \{1, 2, \infty, \text{Fro}, \dots\}$. Another possibility is to use norm-wise error measures as $\|\mathbf{y} - \mathbf{y}_r\|_{(\star)}$ or $\|\mathbf{x} - \mathbf{V}\mathbf{x}_r\|_{(\star)}$ with $(\star) = \{\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_\infty, \dots\}$ in time-domain, or $\|\mathbf{G} - \mathbf{G}_r\|_{(\star)}$ with $(\star) = \{\mathcal{H}_2, \mathcal{H}_\infty, \text{Hankel} \dots\}$ in frequency-domain.
2. **Preservation of system properties:** Basic features of the original model (e. g. stability, passivity, second-order / Port-Hamiltonian structure, etc.) should be preserved in the ROM. This requirement is achieved by applying special or adapted reduction methods tailored to address these demands.
3. **Numerical efficiency:** Model reduction pays off, if the benefit of having multiple, cheap online evaluations (required e.g. for design analysis, optimization and control) outweighs the upfront offline cost needed for the computation of the reduced model. Thus, the reduction method should be as numerically efficient (fast) and stable as possible. Expensive offline, and specially online computations should be avoided. To be avoided: matrix inversion, solution of Lyapunov / Sylvester / Riccati equations, eigenvalue problems, singular value problems, ...

Remark: Since we are interested in the approximation of the transfer behavior, we will assume in the following that the initial values $\mathbf{x}(t=0)$ and $\mathbf{x}_r(t=0)$ are in the origin.

Remark: If the system (1.8) has a feedthrough d , this remains unchanged by the reduction, since the dimensions of d depend only on the number of inputs and outputs, which are not influenced by the reduction. Only the number n of state variables shall be reduced! (However, there are methods with $d_r \neq d$, in order to use additional degrees of freedom in the reduction.)




1.6 Model Order Reduction Software

High-dimensional models arising in real-life problems are usually too complex to be reduced by hand. Therefore, model order reduction is typically performed on a computer. Over time, several software packages in different languages (MATLAB, Python, C) have been developed. An overview can be found at

https://morwiki.mpi-magdeburg.mpg.de/morwiki/index.php/Comparison_of_Software.

The numerical investigations within the lectures and exercises are carried out in MATLAB and are based on the MATLAB built-in functions of the Control System Toolbox, as well as on the Chair's own *sss* and *sssMOR* toolboxes. The *sss* and *sssMOR* toolboxes are available for free download at the Chair's homepage <http://www.rt.mw.tum.de/?morlab> or in GitHub <https://github.com/MORLab>.

In the script, you will find sometimes blue boxes containing MATLAB functions that are linked to the presented contents. These functions can be used to supplement the theoretical discussion from lecture and script with numerical investigations in MATLAB using practical *benchmark* systems. Links to various benchmark collections can be found at <http://www.eu-mor.net/support/benchmark-collections/>.

	MATLAB function(s):	<i>ss</i> , <i>dss</i>
	SSS function(s):	<i>sss</i>
	SSSMOR function(s):	<i>ssRed</i>

Chapter 2

Mathematical Fundamentals

In this chapter some necessary mathematical fundamentals – specially of linear algebra – will be revised or newly introduced.

2.1 Vector space

Definition 1. A *vector space* \mathcal{V} is a set of vectors, in which the addition of the vectors and the multiplication of a vector by a scalar from the field \mathbb{K} (for us: $\mathbb{K} = \mathbb{R}$ or $\mathbb{K} = \mathbb{C}$) are defined and the results of these operations are again elements of \mathcal{V} . Moreover, the following holds for $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathcal{V}$ and $\alpha, \beta \in \mathbb{K}$:

$$\begin{aligned} \text{Commutative law :} & \quad \mathbf{x} + \mathbf{y} = \mathbf{y} + \mathbf{x} \\ \text{Associative law :} & \quad (\mathbf{x} + \mathbf{y}) + \mathbf{z} = \mathbf{x} + (\mathbf{y} + \mathbf{z}) \\ \text{Existence of zero element :} & \quad \exists \mathbf{0} \in \mathcal{V} : \quad \mathbf{x} + \mathbf{0} = \mathbf{x} \\ \text{Existence of the inverse element :} & \quad \exists (-\mathbf{x}) \in \mathcal{V} : \quad \mathbf{x} + (-\mathbf{x}) = \mathbf{0} \\ \text{Distributive law 1 :} & \quad (\alpha + \beta)\mathbf{x} = \alpha\mathbf{x} + \beta\mathbf{x} \\ \text{Distributive law 2 :} & \quad \alpha(\mathbf{x} + \mathbf{y}) = \alpha\mathbf{x} + \alpha\mathbf{y} \\ & \quad (\alpha\beta)\mathbf{x} = \alpha(\beta\mathbf{x}) \\ \text{Existence of the identity element :} & \quad \exists 1 \in \mathbb{K} : \quad 1\mathbf{x} = \mathbf{x} \end{aligned}$$

Example: Vectors in \mathbb{R}^n .

Definition 2. A subset \mathcal{W} of a vector space \mathcal{V} , i. e. $\mathcal{W} \subset \mathcal{V}$, is called *vector subspace* (short: *subspace*), if the elements of \mathcal{W} with the same rules of the vector space \mathcal{V} (addition, scalar multiplication) again form a vector space, i. e.

$$\forall \mathbf{x}, \mathbf{y} \in \mathcal{W} \text{ and } \alpha \in \mathbb{K} : \quad \mathbf{x} + \mathbf{y} \in \mathcal{W} \text{ and } \alpha\mathbf{x} \in \mathcal{W}. \quad (2.1)$$

Example: Plane in \mathbb{R}^3 , e. g. the x_1 - x_2 -plane:

$$\left\{ \mathbf{x} \in \mathbb{R}^3 : \mathbf{x} = \alpha \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \alpha, \beta \in \mathbb{R} \right\} \subset \mathbb{R}^3 \quad (2.2)$$

2.2 Linear dependence and rank

Definition 3. The vectors $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n \in \mathcal{V}$ are called **linearly dependent**, if there are numbers $c_1, c_2, \dots, c_n \in \mathbb{R}$, which are not all equal to zero, such that:

$$c_1 \mathbf{v}_1 + c_2 \mathbf{v}_2 + \dots + c_n \mathbf{v}_n = \mathbf{0}. \quad (2.3)$$

Otherwise, they are called **linearly independent**.

Linear dependence means therefore that a vector \mathbf{v}_i can be represented by a linear combination of all other remaining vectors. The number of linearly independent column vectors of a matrix corresponds to the rank.

Definition 4. The rank of a matrix \mathbf{V} is defined as the number of linearly independent column vectors of \mathbf{V} .

It follows that a matrix $\mathbf{V} \in \mathbb{R}^{m \times n}$ has the full rank $\min(m, n)$, if all column vectors are linearly independent. In the literature you will also find the terms “row rank” and “column rank”. Since both are identical, these three terms can be used as synonyms.

Corollary 1. Let $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n \in \mathbb{R}^n$ and $\mathbf{V} := [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n]$. Then, the following holds:

$$\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n \text{ linearly independent} \quad \Leftrightarrow \quad \text{rank}(\mathbf{V}) = n \quad \Leftrightarrow \quad \det(\mathbf{V}) \neq 0. \quad (2.4)$$

Remark: While the equivalence between linear independence and column rank applies to any matrix, the condition with the determinant only applies to square matrices.

Example: Consider the following matrix:

$$\mathbf{V} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \\ 1 & 4 & 5 \\ 2 & 5 & 7 \\ 3 & 6 & 9 \end{bmatrix} \quad (2.5)$$

The columns of \mathbf{V} are linearly dependent, since \mathbf{v}_3 is the sum of the other two columns:

$$1\mathbf{v}_1 + 1\mathbf{v}_2 - 1\mathbf{v}_3 = \mathbf{0} \quad \Rightarrow \quad \mathbf{v}_3 = \mathbf{v}_1 + \mathbf{v}_2 \quad (2.6)$$

On the other hand, \mathbf{v}_1 and \mathbf{v}_2 are linearly independent, and hence $\text{rank}(\mathbf{V}) = 2$.



MATLAB function(s): rank, det

2.3 Basis

Definition 5. A basis \mathcal{B} of a vector space \mathcal{V} is a subset $\mathcal{B} \subset \mathcal{V}$, with the two properties:

1. Each element of \mathcal{V} can be represented as a linear combination of vectors from \mathcal{B} .
2. The basis vectors are linearly independent.

Example: The columns of all 3×3 matrices with rank 3 form bases for the \mathbb{R}^3 .

Definition 6. The dimension $\dim(\mathcal{V})$ of a vector space \mathcal{V} is defined as the number of basis vectors in \mathcal{B} .

Definition 7. A basis $\mathcal{B} = \{\mathbf{b}_1, \dots, \mathbf{b}_n\}$ is called orthogonal, if all vectors are pairwise orthogonal to each other, i. e.

$$\mathbf{b}_i^T \mathbf{b}_j = 0, \quad \text{for } i \neq j. \quad (2.7)$$

Are the basis vectors also normalized, i. e. $\|\mathbf{b}_i\|_2 = 1$, then we speak of an orthonormal basis.

Example: Canonical basis for \mathbb{R}^3 (orthonormal basis):

$$\mathbf{b}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{b}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{b}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \quad (2.8)$$

Matrices whose column vectors form an orthonormal basis are of particular importance and are introduced below.

Definition 8. A square real matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is called orthogonal, if its column vectors are orthonormal (i.e. not only orthogonal) to each other. In this case, its transpose is equal to the inverse:

$$\mathbf{A}^T \mathbf{A} = \mathbf{A} \mathbf{A}^T = \mathbf{I}_n \quad \Leftrightarrow \quad \mathbf{A}^T = \mathbf{A}^{-1} \quad (2.9)$$

The analogue for complex matrices $\mathbf{A} \in \mathbb{C}^{n \times n}$ is called unitary matrix.

By definition, an orthogonal matrix must be square. In model order reduction, however, rectangular matrices $\mathbf{V} \in \mathbb{R}^{n \times q}$ with $n \gg q$ play an important role. Their column vectors form an orthonormal basis for a q -dimensional subspace of \mathbb{R}^n . Since there is no clear designation for such matrices in the mathematical literature, we will allow ourselves a so-called “abuse

of notation”¹ and occasionally also assign the attribute *orthogonal* to rectangular matrices $\mathbf{V} \in \mathbb{R}^{n \times q}$, if they fulfill the property:

$$\mathbf{V}^T \mathbf{V} = \mathbf{I}_q. \quad (2.10)$$

 MATLAB function(s): `orth`

2.4 Range and nullspace

Definition 9. *The range or image of a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ is the set of all vectors $\mathbf{y} \in \mathbb{R}^m$, which can be represented as the product $\mathbf{y} = \mathbf{A}\mathbf{x}$ for arbitrary $\mathbf{x} \in \mathbb{R}^n$.*

The product $\mathbf{A}\mathbf{x}$ can be represented as a weighted sum of the columns \mathbf{A}_i in the form

$$\mathbf{y} = \mathbf{A}\mathbf{x} = \mathbf{A}_1 x_1 + \mathbf{A}_2 x_2 + \cdots + \mathbf{A}_n x_n,$$

which shows, that the image of the matrix \mathbf{A} is precisely the vector space spanned by its columns.

Definition 10. *The nullspace or kernel of a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ is the set of all vectors $\mathbf{x} \in \mathbb{R}^n$, for which the following holds: $\mathbf{A}\mathbf{x} = \mathbf{0}$.*

Example:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 \\ 3 & 3 & 6 \\ 0 & 2 & 2 \\ 1 & 1 & 2 \end{bmatrix} \quad (2.11)$$

This matrix has $\text{rank}(\mathbf{A}) = 2$, i. e. the *range/image* has dimension 2:

$$\text{range}(\mathbf{A}) = \text{im}(\mathbf{A}) = \text{span} \left(\left(\left[\begin{array}{c} 1 \\ 3 \\ 0 \\ 1 \end{array} \right], \left[\begin{array}{c} 0 \\ 3 \\ 2 \\ 1 \end{array} \right] \right) \right) \quad (2.12)$$

and the *nullspace* has dimension 1:

$$\ker(\mathbf{A}) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \mathbf{x} = \alpha \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}, \alpha \in \mathbb{R} \right\}. \quad (2.13)$$

A basis for the image and nullspace can be calculated, for example, by a singular value decomposition.

¹“In mathematics, abuse of notation occurs when an author uses a mathematical notation in a way that is not formally correct but that seems likely to simplify the exposition or suggest the correct intuition (while being unlikely to introduce errors or cause confusion)”. (source: Wikipedia)

 MATLAB function(s): `null`

2.5 Singular Value Decomposition (SVD)

The Singular Value Decomposition (SVD) is related to the eigenvalue decomposition, but it is also defined for rectangular matrices $\mathbf{A} \in \mathbb{R}^{m \times n}$:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \quad (2.14)$$

where the matrices $\mathbf{U} \in \mathbb{R}^{m \times m}$ and $\mathbf{V} \in \mathbb{R}^{n \times n}$ are orthogonal ($\mathbf{U}^T\mathbf{U} = \mathbf{I}_m$ and $\mathbf{V}^T\mathbf{V} = \mathbf{I}_n$). The matrix $\mathbf{\Sigma} \in \mathbb{R}^{m \times n}$ has only entries on the diagonal, and is dependent on the dimensions of \mathbf{A} :

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1 & & & & \\ & \sigma_2 & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & \sigma_{\min(m,n)} \end{bmatrix}. \quad (2.15)$$

The entries $\sigma_1 \geq \sigma_2 \dots \geq \sigma_{\min(m,n)}$ are non-negative (i. e. positive or zero), they are sorted in descending order and called singular values of the matrix \mathbf{A} . The following figure illustrates the decomposition for different dimensions:

$$\begin{array}{l} \mathbf{A} \quad = \quad \mathbf{U} \quad \mathbf{\Sigma} \quad \mathbf{V}^T \\ [m \times n] \quad [m \times m] \quad [m \times n] \quad [n \times n] \\ \\ m > n : \quad \boxed{\mathbf{A}} \quad = \quad \boxed{\mathbf{U}} \quad \boxed{\mathbf{\Sigma}} \quad \boxed{\mathbf{V}^T} \\ \\ m = n : \quad \boxed{\mathbf{A}} \quad = \quad \boxed{\mathbf{U}} \quad \boxed{\mathbf{\Sigma}} \quad \boxed{\mathbf{V}^T} \\ \\ m < n : \quad \boxed{\mathbf{A}} \quad = \quad \boxed{\mathbf{U}} \quad \boxed{\mathbf{\Sigma}} \quad \boxed{\mathbf{V}^T} \end{array}$$

2.5.1 Properties of the SVD

Let $\text{rank}(\mathbf{A}) = p \leq \min(m, n)$, $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, and write $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_m]$, $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_n]$. Then the following holds:

1. $\sigma_i = 0$ for $i = p + 1, \dots, \min(m, n)$.
2. The singular values σ_i are the roots of the first $\min(m, n)$ eigenvalues of $\mathbf{A}\mathbf{A}^T$ or $\mathbf{A}^T\mathbf{A}$:

$$\sigma_i = \sqrt{\lambda_i(\mathbf{A}^T\mathbf{A})} = \sqrt{\lambda_i(\mathbf{A}\mathbf{A}^T)}. \quad (2.16)$$

Therefore, all singular values are real and non-negative: $\sigma_i \geq 0$.

3. The eigenvectors of $\mathbf{A}\mathbf{A}^T$ (or $\mathbf{A}^T\mathbf{A}$) are the singular vectors \mathbf{u}_i (or \mathbf{v}_i).
4. The right-singular vectors $\mathbf{v}_{p+1}, \dots, \mathbf{v}_n$ form a basis for the nullspace of \mathbf{A} .
5. The left-singular vectors $\mathbf{u}_1, \dots, \mathbf{u}_p$ form a basis for the image of \mathbf{A} .
6. From the definition of the SVD it follows that a matrix can be represented as a dyadic sum:

$$\mathbf{A} = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T + \sigma_2 \mathbf{u}_2 \mathbf{v}_2^T + \dots + \sigma_p \mathbf{u}_p \mathbf{v}_p^T. \quad (2.17)$$

Special case: For positive (semi-)definite matrices (i.e. symmetric matrices with positive (or rather non-negative) eigenvalues) it holds: SVD = EVD (eigenvalue decomposition)

$$\mathbf{A} = \mathbf{A}^T \geq \mathbf{0} \quad \Rightarrow \quad \mathbf{A} \stackrel{\text{SVD}}{=} \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \stackrel{\text{EVD}}{=} \mathbf{T}\mathbf{\Lambda}\mathbf{T}^{-1} \quad \Rightarrow \quad \mathbf{U} = \mathbf{T}, \quad \mathbf{\Sigma} = \mathbf{\Lambda}, \quad \mathbf{V}^T = \mathbf{T}^{-1}$$

since the eigenvectors form an *orthogonal* basis and all eigenvalues are non-negative.

2.5.2 Matrix approximation

The SVD can be used – among others – to approximate matrices. Hereby, a matrix \mathbf{X} with rank $k \leq p$ is sought, such that the following difference is minimized:

$$\min_{\text{rank}(\mathbf{X})=k} \|\mathbf{A} - \mathbf{X}\|_2 \quad (2.18)$$

The 2-norm of a matrix is defined as an induced norm as follows:

$$\|\mathbf{A}\|_2 = \max_{\|\mathbf{x}\|_2=1} \|\mathbf{A}\mathbf{x}\|_2 = \sigma_{\max}(\mathbf{A}) = \sqrt{\lambda_{\max}(\mathbf{A}^T\mathbf{A})} \quad (2.19)$$

One can show that the solution of the minimization problem is exactly the sum of the first k dyadic products of the SVD,

$$\mathbf{X} = \sum_{i=1}^k \sigma_i \mathbf{u}_i \mathbf{v}_i^T = \mathbf{U}\hat{\mathbf{\Sigma}}\mathbf{V}^T, \quad (2.20)$$

where $\hat{\mathbf{\Sigma}} = \text{diag}(\sigma_1, \dots, \sigma_k, 0, \dots, 0)$. Moreover, for the minimum holds:

$$\min_{\text{rank}(\mathbf{X})=k} \|\mathbf{A} - \mathbf{X}\|_2 = \sigma_{k+1}(\mathbf{A}). \quad (2.21)$$

2.5.3 (Further) Applications of the SVD

For the sake of completeness, some other important applications of the SVD are listed below:

- **Range, nullspace and rank:** As mentioned before, the SVD provides a basis for the range and nullspace of a matrix. Moreover, the rank of a matrix equals the number of non-zero singular values in Σ .
- **Low-rank matrix approximation / Deflation:** As discussed in Sec. 2.5.2, the SVD can be used to obtain a low-rank approximation of a matrix. This application is strongly related to the often required *deflation* of a basis, in order to eliminate redundant column vectors and obtain a full rank, orthogonal matrix.
- **Nearest orthogonal matrix:** Related to the application before, the SVD can be used to determine the orthogonal matrix \mathbf{O} closest to a square matrix \mathbf{A} . Moreover, it is also employed for the solution of the so-called *orthogonal Procrustes problem* $\mathbf{O} = \arg \min_{\mathbf{X}} \|\mathbf{A}\mathbf{X} - \mathbf{B}\|_{\text{Fro}}$ s.t. $\mathbf{X}^T \mathbf{X} = \mathbf{I}$.
- **Pseudoinverse:** The SVD can be used for computing the pseudoinverse of a matrix. The pseudoinverse is one way to solve linear least-squares minimization problems such as $\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2$, arising e.g. in underdetermined linear systems of equations.
- **MOR / Reduced order modeling:** *Proper Orthogonal Decomposition* (POD) is a (nonlinear) reduction method, which is based on the SVD of a so-called *snapshot matrix*. The idea is to collect samples of the FOM state trajectory in $\mathbf{X} = [\mathbf{x}(t_1), \dots, \mathbf{x}(t_{n_s})]$ and to build the reduction basis with the q leading left-singular vectors of \mathbf{X} .
- **Others:** The SVD is widely applied in different fields such as signal processing, image processing, statistics, pattern recognition, machine learning and big data.



MATLAB function(s): `svd, svds, gsvd (orth, rank, null, pinv)`

2.6 Norms

For certain analysis purposes, it is useful to assess the “intensity” of a time signal or a dynamic system in the form of a norm. I. e. a signal in the time-domain or a function in the frequency-domain will be assigned a scalar non-negative value by the norm. Time signals are usually evaluated by the \mathcal{L}_p -norms, where p is usually 1, 2 or ∞ . The same applies to the \mathcal{H}_p -norms to assess/evaluate dynamic systems. We will use the following norms:

Definition 11. *The \mathcal{L}_2 -norm for time signals is defined as:*

$$\|\mathbf{x}\|_{\mathcal{L}_2} = \sqrt{\int_0^{\infty} |\mathbf{x}(t)|^2 dt}. \quad (2.22)$$

Definition 12. The \mathcal{H}_2 -norm of LTI systems is defined as:

$$\|G\|_{\mathcal{H}_2} = \sqrt{\frac{1}{2\pi} \int_{-\infty}^{\infty} |G(i\omega)|^2 d\omega} = \sqrt{\int_0^{\infty} |g(t)|^2 dt}, \quad (\text{SISO}) \quad (2.23a)$$

$$\|\mathbf{G}\|_{\mathcal{H}_2} = \sqrt{\frac{1}{2\pi} \int_{-\infty}^{\infty} \text{tr}(\mathbf{G}(i\omega) \mathbf{G}(-i\omega)^T) d\omega} = \sqrt{\int_0^{\infty} \text{tr}(\mathbf{g}(t) \mathbf{g}(t)^T) dt}. \quad (\text{MIMO}) \quad (2.23b)$$

Corollary 2. The \mathcal{H}_2 -norm of an asymptotically stable LTI system can be equivalently represented as (here only SISO, but analog for MIMO):

$$\begin{aligned} \|G\|_{\mathcal{H}_2}^2 &= \int_0^{\infty} |g(t)|^2 dt = \int_0^{\infty} \underbrace{\mathbf{c}^T e^{\mathbf{A}t} \mathbf{b}}_{g(t)} \underbrace{\mathbf{b}^T e^{\mathbf{A}^T t} \mathbf{c}}_{g(t)^T} dt = \int_0^{\infty} \underbrace{\mathbf{b}^T e^{\mathbf{A}^T t} \mathbf{c}}_{g(t)^T} \underbrace{\mathbf{c}^T e^{\mathbf{A}t} \mathbf{b}}_{g(t)} dt \\ &= \mathbf{c}^T \underbrace{\int_0^{\infty} e^{\mathbf{A}t} \mathbf{b} \mathbf{b}^T e^{\mathbf{A}^T t} dt}_{\mathbf{W}_c} \mathbf{c} = \mathbf{b}^T \underbrace{\int_0^{\infty} e^{\mathbf{A}^T t} \mathbf{c} \mathbf{c}^T e^{\mathbf{A}t} dt}_{\mathbf{W}_o} \mathbf{b} \\ &= \mathbf{c}^T \mathbf{W}_c \mathbf{c} = \mathbf{b}^T \mathbf{W}_o \mathbf{b}, \end{aligned} \quad (2.24)$$

where the matrices \mathbf{W}_c and \mathbf{W}_o are the so-called controllability and observability Gramians of the system (cf. Ch. 5).

Definition 13. The \mathcal{H}_∞ -norm of LTI systems is defined as:

$$\|G\|_{\mathcal{H}_\infty} = \sup_{\|u\|_{\mathcal{L}_2} \neq 0} \frac{\|y\|_{\mathcal{L}_2}}{\|u\|_{\mathcal{L}_2}} = \sup_{\omega \in \mathbb{R}} |G(i\omega)|, \quad (\text{SISO}) \quad (2.25a)$$

$$\|\mathbf{G}\|_{\mathcal{H}_\infty} = \sup_{\omega \in \mathbb{R}} \max_{i=1, \dots, n} \sigma_i(\mathbf{G}(i\omega)) = \sup_{\omega \in \mathbb{R}} \sigma_{\max}(\mathbf{G}(i\omega)). \quad (\text{MIMO}) \quad (2.25b)$$


Some further explanations for SISO LTI systems (similar for MIMO)

The \mathcal{H}_2 -norm is equivalent to the \mathcal{L}_2 -norm of the impulse response $g(t)$, i.e. $\|G\|_{\mathcal{H}_2} = \|g\|_{\mathcal{L}_2}$. The \mathcal{H}_∞ -norm describes the highest value in the amplitude response of the system. This is equivalent to the maximum steady-state (DC) gain of the system for harmonic excitation.



For the evaluation of the approximation quality of LTI reduced order models, the error is often quantified in the frequency-domain using the \mathcal{H}_2 - or \mathcal{H}_∞ -norms, i.e. via $\|G - G_r\|_{\mathcal{H}_2}$ or $\|G - G_r\|_{\mathcal{H}_\infty}$. Hereby, $G_r(s)$ represents the transfer function of the reduced model. It can be shown that the error measures in the frequency-domain also represent an upper bound for the approximation error in the time-domain, according to the relationships

$$\begin{aligned} \|y - y_r\|_{\mathcal{L}_\infty} &\leq \|G - G_r\|_{\mathcal{H}_2} \|u\|_{\mathcal{L}_2}, \\ \|y - y_r\|_{\mathcal{L}_2} &\leq \|G - G_r\|_{\mathcal{H}_\infty} \|u\|_{\mathcal{L}_2}. \end{aligned} \quad (2.26)$$

Matrix or vector norms

 MATLAB function(s): `norm(X)`, `norm(X, 'fro')`, `norm(v,p)`

System norms

 MATLAB function(s): `norm(sys)`, `norm(sys,inf)`
 SSS function(s): `norm(sys)`, `norm(sys,inf)`

Chapter 3

Projective Model Order Reduction

The MOR methods that we want to consider here are based on projections. An illustrative example for a projection is the shadow cast: the shadow is a two-dimensional image of a three-dimensional object. This chapter introduces projections in general, regardless of the MOR method used.

3.1 The projector

Mathematically speaking, a projection is a linear mapping of a vector space \mathcal{V} into itself, and is represented by a multiplication with the projector $\mathbf{P} \in \mathbb{R}^{n \times n}$: $\mathbf{x}_{\text{projected}} = \mathbf{P}\mathbf{x}$. The image/range of a projection is either a subspace of \mathcal{V} or \mathcal{V} itself.

Definition 14. *A projector is defined by:*

$$\mathbf{P} = \mathbf{P}^2. \quad (3.1)$$

This simple definition is indeed sufficient, and such matrices are also called *idempotent*. A clear explanation is that a repeated projection does not change anything, i. e. if one takes a projected vector $\mathbf{x}_{\text{projected}} = \mathbf{P}\mathbf{x}$ as a new output vector and projects it again, then the same vector must result: $\mathbf{P}\mathbf{x}_{\text{projected}} = \mathbf{x}_{\text{projected}}$.

Theorem 1. *The eigenvalues of a projector \mathbf{P} are either 0 or 1.*

Proof. Let $\mathbf{P} = \mathbf{V}\mathbf{D}\mathbf{V}^{-1}$ be the eigendecomposition of the projector. From the definition follows

$$\begin{aligned} \mathbf{P}^2 = \mathbf{V}\mathbf{D}\mathbf{V}^{-1}\mathbf{V}\mathbf{D}\mathbf{V}^{-1} = \mathbf{V}\mathbf{D}^2\mathbf{V}^{-1} &\stackrel{\mathbf{P}=\mathbf{P}^2}{\implies} \mathbf{D} = \mathbf{D}^2 \quad (\text{diagonal matrix}) \\ &\implies \lambda_i(\mathbf{P}) = \lambda_i(\mathbf{D}) \in \{0, 1\}. \end{aligned}$$

□

Remark: Conversely, the theorem generally does not apply, as the matrix $\begin{bmatrix} 1 & 3 \\ 0 & 1 \end{bmatrix}$ shows.

The eigenspaces of \mathbf{P} can be computed as follows:

$$\begin{aligned} \text{eigenvectors corresponding to the eigenvalues } \lambda_i = 0 &: \ker(\mathbf{P}) \\ \text{eigenvectors corresponding to the eigenvalues } \lambda_i = 1 &: \text{range}(\mathbf{P}) \end{aligned}$$

$\text{range}(\mathbf{P})$ denotes the subspace onto it is projected, whereas $\ker(\mathbf{P})$ defines the direction of the projection. The whole vector space (in our case mostly \mathbb{R}^n) is the sum of these two vector subspaces:

$$\mathcal{V} = \ker(\mathbf{P}) \oplus \text{range}(\mathbf{P}). \quad (3.2)$$

Consequently, each vector $\mathbf{x} \in \mathcal{V}$ can be written as the direct sum of both components:

$$\left. \begin{aligned} \mathbf{v} &:= \mathbf{P}\mathbf{x} \\ \mathbf{u} &:= \mathbf{x} - \mathbf{P}\mathbf{x} = (\mathbf{I} - \mathbf{P})\mathbf{x} \end{aligned} \right\} \Rightarrow \mathbf{x} = \mathbf{u} + \mathbf{v}, \quad (3.3)$$

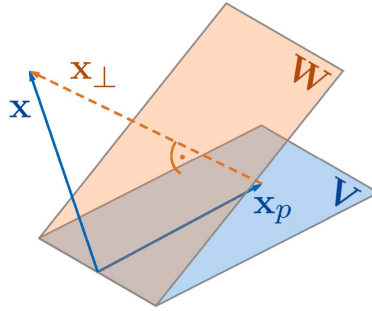
where $\mathbf{u} \in \ker(\mathbf{P})$ and $\mathbf{v} \in \text{range}(\mathbf{P})$ are lying in the kernel or in the range of \mathbf{P} .

Proof.

$$\begin{aligned} \mathbf{P}\mathbf{v} = \mathbf{P}^2\mathbf{x} = \mathbf{P}\mathbf{x} = \mathbf{v} &\Rightarrow \mathbf{v} \in \text{range}(\mathbf{P}) \\ \mathbf{P}\mathbf{u} = \mathbf{P}\mathbf{x} - \mathbf{P}^2\mathbf{x} = \mathbf{0} &\Rightarrow \mathbf{u} \in \ker(\mathbf{P}) \quad \square \end{aligned}$$

3.2 Generating projectors

How can projectors \mathbf{P} be specifically be generated, in order to achieve a desired projection? This shall be derived exemplarily using a projection in \mathbb{R}^3 :



The vector $\mathbf{x} \in \mathbb{R}^3$ shall be projected onto the plane spanned by the columns of the matrix $\mathbf{V} \in \mathbb{R}^{3 \times 2}$. The projection should be carried out along the direction \mathbf{x}_\perp . For this purpose, the matrix $\mathbf{W} \in \mathbb{R}^{3 \times 2}$ is defined, whose columns are orthogonal to \mathbf{x}_\perp . If the projected vector \mathbf{x}_p shall lie in \mathbf{V} , then \mathbf{x}_p must be represented as a linear combination of the columns of \mathbf{V} , thus:

$$\mathbf{x}_p = \mathbf{V} \mathbf{r}, \quad (3.4)$$

where $\mathbf{r} \in \mathbb{R}^2$ is still unknown. We also know that the vector chain must be closed:

$$\mathbf{x}_\perp = \mathbf{x} - \mathbf{x}_p. \quad (3.5)$$

The columns of \mathbf{W} are orthogonal to \mathbf{x}_\perp , so the following applies:

$$\mathbf{W}^T \mathbf{x}_\perp \stackrel{!}{=} \mathbf{0}. \quad (3.6)$$

Inserting equations (3.4) and (3.5) in (3.6) yields:

$$\mathbf{W}^T \mathbf{x}_\perp = \mathbf{W}^T \mathbf{x} - \mathbf{W}^T \mathbf{x}_p = \mathbf{W}^T \mathbf{x} - \mathbf{W}^T \mathbf{V} \mathbf{r} = \mathbf{0} \quad (3.7)$$

$$\Leftrightarrow \mathbf{W}^T \mathbf{V} \mathbf{r} = \mathbf{W}^T \mathbf{x} \quad (3.8)$$

$$\Leftrightarrow \mathbf{r} = (\mathbf{W}^T \mathbf{V})^{-1} \mathbf{W}^T \mathbf{x} \quad (3.9)$$

Using the equation (3.4) the projector can be specified:

$$\mathbf{x}_p = \mathbf{V} \mathbf{r} = \mathbf{V} \underbrace{(\mathbf{W}^T \mathbf{V})^{-1} \mathbf{W}^T \mathbf{x}}_{\mathbf{P}} \quad (3.10)$$

This derivation also applies to any higher dimensions, which is why any projection matrices \mathbf{V} , $\mathbf{W} \in \mathbb{R}^{n \times q}$ lead to a projector $\mathbf{P} \in \mathbb{R}^{n \times n}$:

$$\mathbf{P} = \mathbf{V} (\mathbf{W}^T \mathbf{V})^{-1} \mathbf{W}^T \quad (3.11)$$

$$\square = \square (\square)^{-1} \square$$

Hereby we would like to assume – without loss of generality – that the projection matrices \mathbf{V} and \mathbf{W} have full column rank, i.e. are basis matrices for the respective subspaces $\text{range}(\mathbf{V})$ and $\text{range}(\mathbf{W})$. Similarly, the assumption of invertibility of $\mathbf{W}^T \mathbf{V}$, which is required in equation (3.11), makes sense: If the matrix $\mathbf{W}^T \mathbf{V}$ is singular, then the underlying projection is not meaningful, because the projection direction and the projection plane partly run parallel to each other, as the following theorem shows.

Theorem 2. *The relation $\det(\mathbf{W}^T \mathbf{V}) = 0$ is valid, if and only if a vector $\tilde{\mathbf{v}} \in \text{range}(\mathbf{V})$ exists, such that $\mathbf{W}^T \tilde{\mathbf{v}} = \mathbf{0}$ holds.*

Proof.

$$\det(\mathbf{W}^T \mathbf{V}) = 0 \iff \exists \mathbf{x} \in \mathbb{R}^q, \mathbf{x} \neq \mathbf{0} : (\mathbf{W}^T \mathbf{V}) \mathbf{x} = \mathbf{0}$$

$$\iff \exists \mathbf{x} \in \mathbb{R}^q, \mathbf{x} \neq \mathbf{0} : \mathbf{W}^T \begin{bmatrix} \mathbf{v}_1 & \dots & \mathbf{v}_q \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_q \end{bmatrix}$$

$$\iff \exists \tilde{\mathbf{v}} \in \text{range}(\mathbf{V}), \tilde{\mathbf{v}} \neq \mathbf{0} : \mathbf{W}^T \tilde{\mathbf{v}} = \mathbf{0} \quad \square$$

Theorem 3. $\mathbf{P} = \mathbf{V} (\mathbf{W}^T \mathbf{V})^{-1} \mathbf{W}^T$ is a projector.

Proof. The proof is done by checking the definition:

$$\mathbf{P}^2 = \mathbf{V} (\mathbf{W}^T \mathbf{V})^{-1} \underbrace{\mathbf{W}^T \mathbf{V} (\mathbf{W}^T \mathbf{V})^{-1}}_{=\mathbf{I}} \mathbf{W}^T = \mathbf{P}. \quad \square$$

For our purposes, it makes sense to consider projectors in the formulation (3.11), since $\text{range}(\mathbf{V})$ is the subspace onto it is projected, and the orthogonal complement of $\text{range}(\mathbf{W})$ determines the direction of the projection. For this, we need the following definition.

Definition 15. Let \mathcal{W} be a subspace of \mathbb{R}^n . Then, the orthogonal complement \mathcal{W}^\perp of \mathcal{W} is defined as:

$$\mathcal{W}^\perp := \left\{ \mathbf{x} \in \mathbb{R}^n : \mathbf{x}^T \mathbf{y} = 0 \text{ for all } \mathbf{y} \in \mathcal{W} \right\}$$

The directions of a projection are the orthogonal complement of $\text{range}(\mathbf{W})$, i. e. $[\text{range}(\mathbf{W})]^\perp$, since $\mathbf{W}^T \mathbf{x}$ is formed on the right side of the product $\mathbf{P}\mathbf{x}$. Through the product $\mathbf{W}^T \mathbf{x}$, all directions that are orthogonal to $\text{range}(\mathbf{W})$ are “deleted” during the projection. In our example above, that is exactly \mathbf{x}_\perp , because $\mathbf{W}^T \mathbf{x}_\perp = \mathbf{0}$. From similar considerations and from equation (3.4) also follows:

$$\text{range}(\mathbf{P}) = \text{range}(\mathbf{V}), \quad \ker(\mathbf{P}) = [\text{range}(\mathbf{W})]^\perp. \quad (3.12)$$

Definition 16. A projection is called orthogonal, if – in addition to $\mathbf{P} = \mathbf{P}^2$ – it further holds: $\mathbf{P} = \mathbf{P}^T$. Otherwise, it is called a skew projection.

Proof. The eigenvectors of a symmetric matrix $\mathbf{P} = \mathbf{P}^T$ form an orthogonal basis. It follows that $\text{range}(\mathbf{P}) \perp \ker(\mathbf{P})$. However, since $\ker(\mathbf{P})$ defines the direction of the projection, it is thus projected orthogonally onto $\text{range}(\mathbf{P})$. \square

For a projector (3.11), the special choice $\mathbf{W} := \mathbf{V}$ leads to an orthogonal projection: $\mathbf{P} = \mathbf{V}(\mathbf{V}^T \mathbf{V})^{-1} \mathbf{V}^T = \mathbf{P}^T$. This can be verified with the figure at the beginning of the chapter. What has been said so far about projections is summarized in the following table:

	General projection	Generated projection	Illustrative example
Projection	Mapping with: \mathbf{P}	Mapping with: $\mathbf{V}(\mathbf{W}^T \mathbf{V})^{-1} \mathbf{W}^T$	Shadow cast
Projection onto	$\text{range}(\mathbf{P})$	$\text{range}(\mathbf{V})$	Ground
Direction of projection	$\ker(\mathbf{P})$	orthogonal to \mathbf{W} or $[\text{range}(\mathbf{W})]^\perp$	Axis of the sun
Orthogonal projection	$\mathbf{P} = \mathbf{P}^T$	$\mathbf{V}(\mathbf{V}^T \mathbf{V})^{-1} \mathbf{V}^T$	Sun in zenith
Skew projection	$\mathbf{P} \neq \mathbf{P}^T$	$\mathbf{V}(\mathbf{W}^T \mathbf{V})^{-1} \mathbf{W}^T$, with: $\text{span}(\mathbf{V}) \neq \text{span}(\mathbf{W})$	e. g. evening sun

As it is shown in exercise 3.2, projectors are *independent* of the choice of the basis matrices \mathbf{V}, \mathbf{W} for the subspaces \mathcal{V}, \mathcal{W} . Thus, for any regular matrices $\mathbf{T}_v, \mathbf{T}_w \in \mathbb{R}^{q \times q}$ with $\tilde{\mathbf{V}} = \mathbf{V}\mathbf{T}_v$ and $\tilde{\mathbf{W}} = \mathbf{W}\mathbf{T}_w$ it holds

$$\mathbf{P} = \mathbf{V}(\mathbf{W}^T \mathbf{V})^{-1} \mathbf{W}^T = \tilde{\mathbf{V}}(\tilde{\mathbf{W}}^T \tilde{\mathbf{V}})^{-1} \tilde{\mathbf{W}}^T. \quad (3.13)$$

Hence it follows that in an orthogonal projection – without loss of generality – $\mathbf{W} = \mathbf{V}$ may be assumed, because the choice of any other basis leads to the same projector.

There are still two important special cases to mention:

1. Are the two projection matrices biorthogonal, i. e. $\mathbf{W}^T \mathbf{V} = \mathbf{I}_q$, then the (generally skew or oblique) projector follows:

$$\mathbf{P} = \mathbf{V} \mathbf{W}^T. \quad (3.14)$$

Attention: “Orthogonal projection” and “biorthogonal bases” denote two different things!

2. Is a “truncation-operation” desired (i. e. keeping only the first q coordinate directions of a vector), then the projection matrices

$$\mathbf{V} = \mathbf{W} = \begin{bmatrix} \mathbf{I}_q \\ \mathbf{0} \end{bmatrix} \text{ are used.} \quad \Rightarrow \quad \mathbf{P} = \begin{bmatrix} \mathbf{I}_q & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

3.3 Projective MOR

In model reduction, the state vector $\mathbf{x}(t) \in \mathbb{R}^n$ shall be approximated by a vector $\mathbf{x}_r(t) \in \mathbb{R}^q$ of lower dimension $q \ll n$. For this, one chooses a projection matrix $\mathbf{V} \in \mathbb{R}^{n \times q}$ that spans a q -dimensional subspace and whose columns represent the basis vectors for the reduced state $\mathbf{x}_r(t)$. If the subspace contains/includes the main dynamics of the original system, then the following approximation is valid:

$$\mathbf{x}(t) \approx \mathbf{V} \mathbf{x}_r(t). \quad (3.15)$$

For a system in minimal realization, this equation can never be exactly fulfilled, there is an error $\mathbf{e}(t)$:

$$\mathbf{x}(t) = \mathbf{V} \mathbf{x}_r(t) + \mathbf{e}(t). \quad (3.16)$$

Substituting this in the state equation of the original system results in:

$$\mathbf{V} \dot{\mathbf{x}}_r(t) = \mathbf{A} \mathbf{V} \mathbf{x}_r(t) + \mathbf{b} u(t) + \boldsymbol{\epsilon}(t), \quad (3.17)$$

where the residual $\boldsymbol{\epsilon}(t) = \mathbf{A} \mathbf{e}(t) - \dot{\mathbf{e}}(t)$ contains the error terms caused by the approximation.

The state equation is overdetermined: q variables in \mathbf{x}_r and n equations. As known from linear algebra, an overdetermined system of equations in the general form $\mathbf{M} \boldsymbol{\xi} = \boldsymbol{\gamma}$ has a unique solution, if the right-hand side $\boldsymbol{\gamma}$ lies in the vector space $\text{range}(\mathbf{M})$, i. e. if $\boldsymbol{\gamma}$ can be completely represented by a linear combination of the columns of \mathbf{M} . For equation (3.17) this condition is generally not fulfilled. In order to still obtain a unique solution for $\mathbf{x}_r(t)$, this condition is *enforced* by projecting the state differential equation onto the q -dimensional

subspace $\text{range}(\mathbf{V})^1$:

$$\mathbf{P}\mathbf{V}\dot{\mathbf{x}}_r(t) = \mathbf{P}\mathbf{A}\mathbf{V}\mathbf{x}_r(t) + \mathbf{P}\mathbf{b}u(t) + \mathbf{P}\boldsymbol{\epsilon}(t), \quad (3.18)$$

$$\begin{aligned} \Rightarrow \mathbf{V}(\mathbf{W}^T\mathbf{V})^{-1}\mathbf{W}^T\mathbf{V}\dot{\mathbf{x}}_r(t) &= \mathbf{V}(\mathbf{W}^T\mathbf{V})^{-1}\mathbf{W}^T\mathbf{A}\mathbf{V}\mathbf{x}_r(t) + \\ &+ \mathbf{V}(\mathbf{W}^T\mathbf{V})^{-1}\mathbf{W}^T\mathbf{b}u(t) + \\ &+ \mathbf{V}(\mathbf{W}^T\mathbf{V})^{-1}\mathbf{W}^T\boldsymbol{\epsilon}(t). \end{aligned} \quad (3.19)$$

The projected system (3.19) can thus be solved uniquely in reduced coordinates for any residual $\boldsymbol{\epsilon}(t)$, which is generally not known. In order to solve the problem, the solution of the projected differential equation is chosen such that the resulting $\boldsymbol{\epsilon}(t)$ disappears during the projection: $\mathbf{V}(\mathbf{W}^T\mathbf{V})^{-1}\mathbf{W}^T\boldsymbol{\epsilon}(t) = \mathbf{0}$. One interpretation is that the residual $\boldsymbol{\epsilon}(t)$ lies in the $\ker(\mathbf{P})$, i. e. $\mathbf{W}^T\boldsymbol{\epsilon}(t) = \mathbf{0}$, or $\boldsymbol{\epsilon}(t) \perp \mathbf{W}$.

Definition 17. *The condition $\boldsymbol{\epsilon}(t) \perp \mathbf{W}$ is called Petrov-Galerkin condition. If an orthogonal projection is applied – i. e. $\mathbf{W} = \mathbf{V}$ and thus $\boldsymbol{\epsilon}(t) \perp \mathbf{V}$ – then it is called Galerkin condition.*

In each summand of (3.19), the matrix \mathbf{V} is on the left and a q -dimensional vector $\mathbf{x}_r(t)$ is on the right. Since it is assumed that the columns of \mathbf{V} are linearly independent, the equation (3.19) is only satisfied, if it is fulfilled by the q -dimensional vectors to the right of \mathbf{V} ; this means that one can simply omit the matrix \mathbf{V} in this case. What remains is the reduced model, i. e. q equations with q unknowns, which can be solved uniquely for $\mathbf{x}_r(t)$:

$$\begin{aligned} \dot{\mathbf{x}}_r(t) &= \overbrace{(\mathbf{W}^T\mathbf{V})^{-1}\mathbf{W}^T\mathbf{A}\mathbf{V}}^{\mathbf{A}_r}\mathbf{x}_r(t) + \overbrace{(\mathbf{W}^T\mathbf{V})^{-1}\mathbf{W}^T\mathbf{b}}^{\mathbf{b}_r}u(t), \\ y_r(t) &= \underbrace{\mathbf{c}^T\mathbf{V}}_{\mathbf{c}_r^T}\mathbf{x}_r(t). \end{aligned} \quad (3.20)$$

Are the bases \mathbf{V} and \mathbf{W} biorthogonal, i. e. $\mathbf{W}^T\mathbf{V} = \mathbf{I}_q$, then the reduced system is given by:

$$\begin{aligned} \dot{\mathbf{x}}_r(t) &= \mathbf{W}^T\mathbf{A}\mathbf{V}\mathbf{x}_r(t) + \mathbf{W}^T\mathbf{b}u(t), \\ y_r(t) &= \mathbf{c}^T\mathbf{V}\mathbf{x}_r(t). \end{aligned} \quad (3.21)$$

Remark: The Petrov-Galerkin condition is needed to remove the (unknown) error terms that inevitably result from the approximation of the state vector. The resulting reduced model would take exactly the form (3.20), if the ansatz $\mathbf{x}(t) \approx \mathbf{V}\mathbf{x}_r(t)$ would be used in the derivation, i. e. without considering the error terms. However, the error plays an important role in the interpretation of (3.20): Solving the projected differential equation in reduced


¹Note at this point the similarity to the “least-squares” approach for solving overdetermined systems of equations. Since there is generally no unique solution, the solution that causes the smallest Euclidean error norm is calculated. This is achieved by an orthogonal projection of the equation onto $\text{range}(\mathbf{V})$ (cf. pseudoinverse).

coordinates results in an error in the state differential equation (namely the residual $\epsilon(t)$), which disappears through the projection. By solving the reduced model (3.20) the Petrov-Galerkin condition is always fulfilled!

Proof.

$$\begin{aligned}
& \dot{\mathbf{x}}(t) - \mathbf{A}\mathbf{x}(t) - \mathbf{b}u(t) = \mathbf{0}, \\
\Rightarrow & \quad \mathbf{P}(\dot{\mathbf{x}}(t) - \mathbf{A}\mathbf{x}(t) - \mathbf{b}u(t)) = \mathbf{0}, \\
\Rightarrow & \quad \mathbf{P}(\mathbf{V}\dot{\mathbf{x}}_r(t) - \mathbf{A}\mathbf{V}\mathbf{x}_r(t) - \mathbf{b}u(t)) - \mathbf{P}\epsilon(t) = \mathbf{0}, \\
\Rightarrow & \quad \mathbf{P}\epsilon(t) = \mathbf{0},
\end{aligned} \tag{3.22}$$

where the last step follows from the solution of (3.20).

 sssMOR function(s): projectiveMOR

Orthogonal and biorthogonal bases

As we have seen before, it is essential that (i) the projection matrices $\mathbf{V}, \mathbf{W} \in \mathbb{R}^{n \times q}$ have full column rank (i.e. they do not contain redundant basis vectors) and that (ii) the matrix $\mathbf{W}^T \mathbf{V}$ is invertible.

For better numerical robustness, it is highly recommended (but not mandatory!) that the projection matrices $\mathbf{V}, \mathbf{W} \in \mathbb{R}^{n \times q}$ are orthogonal (i.e. $\mathbf{V}^T \mathbf{V} = \mathbf{I}_q$, $\mathbf{W}^T \mathbf{W} = \mathbf{I}_q$) or even biorthogonal to each other (i.e. $\mathbf{W}^T \mathbf{V} = \mathbf{I}_q$). In the following, we list different cases:

1. Two-sided reduction ($\mathbf{V} \neq \mathbf{W}$) with orthogonal bases, i.e. $\mathbf{V}^T \mathbf{V} = \mathbf{I}_q$, $\mathbf{W}^T \mathbf{W} = \mathbf{I}_q$:

$$\mathbf{P} = \mathbf{V}(\mathbf{W}^T \mathbf{V})^{-1} \mathbf{W}^T \tag{3.23}$$

2. Two-sided reduction ($\mathbf{V} \neq \mathbf{W}$) with biorthogonal bases, i.e. $\mathbf{W}^T \mathbf{V} = \mathbf{I}_q$:

$$\mathbf{P} = \mathbf{V} \mathbf{W}^T \tag{3.24}$$

3. One-sided reduction ($\mathbf{W} = \mathbf{V}$ or $\mathbf{V} = \mathbf{W}$, i.e. orthogonal projection!) with non-orthogonal bases:

$$\begin{aligned}
\mathbf{P} &= \mathbf{V}(\mathbf{V}^T \mathbf{V})^{-1} \mathbf{V}^T \\
\mathbf{P} &= \mathbf{W}(\mathbf{W}^T \mathbf{W})^{-1} \mathbf{W}^T
\end{aligned} \tag{3.25}$$

4. One-sided reduction ($\mathbf{W} = \mathbf{V}$ or $\mathbf{V} = \mathbf{W}$, i.e. orthogonal projection!) with orthogonal bases, i.e. $\mathbf{V}^T \mathbf{V} = \mathbf{I}_q$, $\mathbf{W}^T \mathbf{W} = \mathbf{I}_q$:

$$\begin{aligned}
\mathbf{P} &= \mathbf{V} \mathbf{V}^T \\
\mathbf{P} &= \mathbf{W} \mathbf{W}^T
\end{aligned} \tag{3.26}$$

3.4 Summary

In this chapter, we have learned the basic procedure for model reduction of linear systems, where the reduced model emerges from a projection of the original model. Therefore, the main goal of a model reduction method in this projective setting is to find appropriate bases \mathbf{V} and \mathbf{W} . In the following, three common methods for the calculation of suitable projection matrices are presented.

Chapter 4

Modal Reduction

One of the first approaches for model order reduction of linear time-invariant systems in state-space representation was the *modal reduction* (aka. *modal truncation*). This reduction method originated in the 1960s and was originally developed for second-order models arising in structural dynamics. The main idea is based on the consideration of the transfer behavior after state transformation to *modal coordinates*. This way, the system can be represented by n *decoupled paths*, where each path is associated to an eigenvalue and a modal coordinate. Hereby it is obvious to neglect paths (i.e. modal coordinates and eigenvalues) that have no significant effect on the transfer behavior.

In the following, the procedure according to Litz is presented, as it can be found in the book [Föl13, Chapter 8.6]. This assumes that all eigenvalues of \mathbf{A} are simple, and thus the matrix \mathbf{A} is diagonalizable.

4.1 Modal transformation

By an eigenvalue decomposition of the system matrix $\mathbf{T}^{-1}\mathbf{A}\mathbf{T} = \mathbf{\Lambda}$ and a change of basis $\mathbf{x} = \mathbf{T}\mathbf{z}$ (*modal transformation*), the state-space representation in modal coordinates results

$$\begin{aligned}\dot{\mathbf{z}}(t) &= \mathbf{\Lambda}\mathbf{z}(t) + \hat{\mathbf{b}}u(t), \\ y(t) &= \hat{\mathbf{c}}^T\mathbf{z}(t),\end{aligned}\tag{4.1}$$

with $\mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_n)$, $\hat{\mathbf{b}} = \mathbf{T}^{-1}\mathbf{b}$, $\hat{\mathbf{c}}^T = \mathbf{c}^T\mathbf{T}$. Here, \mathbf{T} is the eigenvector matrix of \mathbf{A} , so that for each column \mathbf{t}_k it holds: $\mathbf{A}\mathbf{t}_k = \lambda_k\mathbf{t}_k$. The above equation (4.1) thus represents a system of n decoupled differential equations in the modal coordinates:

$$\dot{z}_k(t) = \lambda_k z_k(t) + \hat{b}_k u(t), \quad k = 1, \dots, n.\tag{4.2}$$

Applying the Laplace transform to (4.1) yields

$$\begin{aligned} z_1(s) &= \frac{1}{s - \lambda_1} \hat{b}_1 u(s), \\ &\vdots \\ z_n(s) &= \frac{1}{s - \lambda_n} \hat{b}_n u(s), \\ y(s) &= \sum_{k=1}^n \hat{c}_k z_k(s), \end{aligned} \tag{4.3}$$

from which the block diagram 4.1 with n decoupled paths can be easily constructed.

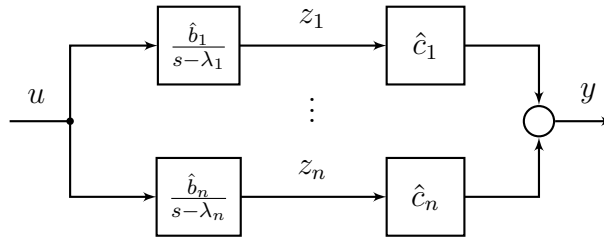


Figure 4.1: Block diagram of the system in modal coordinates.

In addition, the transfer function $G(s)$ takes the simple form of

$$G(s) = \hat{\mathbf{c}}^T (s\mathbf{I} - \mathbf{\Lambda})^{-1} \hat{\mathbf{b}} = \sum_{k=1}^n \frac{\hat{c}_k \hat{b}_k}{s - \lambda_k}. \tag{4.4}$$

To each of the n eigenvalues λ_k and each modal coordinate z_k belongs exactly one path with the share $\frac{\hat{c}_k \hat{b}_k}{s - \lambda_k}$ to the transfer behavior.

 MATLAB function(s): `eig`

4.2 Dominance measure according to Litz

In order to decide which modal coordinates may be neglected during reduction, each eigenvalue λ_k is assigned a positive real number D_k , which represents a measure for its significance in the transfer behavior (4.4).

As one can see from the summands in (4.4), a mere consideration of the eigenvalue λ_k or its absolute value will not suffice to assess its dominance. Instead, the coefficients \hat{c}_k and \hat{b}_k (which, according to the Gilbert criterion, also allow statements on the observability and controllability of λ_k) must be considered.

For this reason, Litz proposed the following dominance measure in the 1970s

$$D_k = \left| \frac{\hat{c}_k \hat{b}_k}{\lambda_k} \right|. \quad (4.5)$$

This can be interpreted as the contribution of the k -th path to the steady-state response of $y(t)$, according to the Laplace transform's final value theorem.

Note at this point that the Litz dominance measure has a direct extension for multiple-input, multiple-output systems ($m, p > 1$) based on the influence of an eigenvalue on individual transmission paths. A more detailed explanation and other measures of dominance can be found in the book [Föl13].

4.3 Modal truncation

The state-space representation in modal coordinates (4.1) can be rearranged using the dominance measure, such that the eigenvalues are sorted from high to low dominance:

$$\begin{aligned} \frac{d}{dt} \begin{bmatrix} z_1 \\ \vdots \\ z_q \\ z_{q+1} \\ \vdots \\ z_n \end{bmatrix} &= \begin{bmatrix} \lambda_1 & & & & & \\ & \ddots & & & & \\ & & \lambda_q & & & \\ \hline & & & \lambda_{q+1} & & \\ & \mathbf{0} & & & \ddots & \\ & & & & & \lambda_n \end{bmatrix} \begin{bmatrix} z_1 \\ \vdots \\ z_q \\ z_{q+1} \\ \vdots \\ z_n \end{bmatrix} + \begin{bmatrix} \hat{b}_1 \\ \vdots \\ \hat{b}_q \\ \hat{b}_{q+1} \\ \vdots \\ \hat{b}_n \end{bmatrix} u \\ y &= \begin{bmatrix} \hat{c}_1 & \dots & \hat{c}_q & \hat{c}_{q+1} & \dots & \hat{c}_n \end{bmatrix} \begin{bmatrix} z_1 \\ \vdots \\ z_q \\ z_{q+1} \\ \vdots \\ z_n \end{bmatrix}. \end{aligned} \quad (4.6)$$

Based from the sorted representation (4.6), the reduced model results by truncating the non-dominant subsystem to

$$\begin{aligned} \dot{\mathbf{z}}_1(t) &= \mathbf{\Lambda}_1 \mathbf{z}_1(t) + \hat{\mathbf{b}}_1 u(t), \\ y(t) &= \hat{\mathbf{c}}_1^T \mathbf{z}_1(t), \end{aligned} \quad (4.7)$$

according to the partitioning $\mathbf{\Lambda}_1 = \text{diag}(\lambda_1, \dots, \lambda_q)$.

The choice of a suitable reduced order q differs from case to case and is usually based either on the relative decrease of the dominance measure or on the absolute value of D_k , which should fall below a certain tolerance.

Modal truncation as projection

Note that modal reduction is also a projective model reduction procedure, because the reduced model (4.7) can also be obtained directly from the original state-space representation by projection (3.20), i.e. via $\mathbf{A}_r = (\mathbf{W}^T \mathbf{V})^{-1} \mathbf{W}^T \mathbf{A} \mathbf{V}$, $\mathbf{b}_r = (\mathbf{W}^T \mathbf{V})^{-1} \mathbf{W}^T \mathbf{b}$ and $\mathbf{c}_r^T = \mathbf{c}^T \mathbf{V}$.

General case The projection matrices \mathbf{V} and \mathbf{W}^T result from the selection of the right and left eigenvectors corresponding to the q most dominant eigenvalues:

$$\mathbf{V} = \mathbf{T}_q = [\mathbf{t}_1 \quad \cdots \quad \mathbf{t}_q] \quad \text{and} \quad \mathbf{W}^T = \mathbf{L}_q^T = \begin{bmatrix} \mathbf{l}_1^T \\ \vdots \\ \mathbf{l}_q^T \end{bmatrix}, \quad (4.8)$$

where the right eigenvectors \mathbf{t}_k and left eigenvectors \mathbf{l}_k^T , $k = 1, \dots, n$ fulfill

$$\begin{aligned} \mathbf{A} \mathbf{t}_k = \lambda_k \mathbf{t}_k &\Leftrightarrow \mathbf{A} \mathbf{T} = \mathbf{T} \mathbf{\Lambda}, \\ \mathbf{l}_k^T \mathbf{A} = \lambda_k \mathbf{l}_k^T &\Leftrightarrow \mathbf{L}^T \mathbf{A} = \mathbf{\Lambda} \mathbf{L}^T. \end{aligned} \quad (4.9)$$

If, for example, $D_3 > D_1 > D_2 \gg D_k$ holds for all other indices $k = 4, \dots, n$, then the following projection matrices result

$$\mathbf{V} = [\mathbf{t}_3 \quad \mathbf{t}_1 \quad \mathbf{t}_2] \quad \text{and} \quad \mathbf{W}^T = \begin{bmatrix} \mathbf{l}_3^T \\ \mathbf{l}_1^T \\ \mathbf{l}_2^T \end{bmatrix}. \quad (4.10)$$

Please note that in this case the projection matrices are not biorthogonal to each other (i.e. $\mathbf{W}^T \mathbf{V} \neq \mathbf{I}_q$), unless they are explicitly biorthogonalized subsequently.

Further note that for a symmetric matrix \mathbf{A} *only one* eigenvalue problem has to be solved, since for $\mathbf{A} = \mathbf{A}^T$ it follows $\mathbf{t}_k = \mathbf{l}_k$ (cf. Equation (4.9)), and hence $\mathbf{V} = \mathbf{W}$.

Special case In this case, the projection matrices \mathbf{V} and \mathbf{W}^T are given by the eigenvector matrix \mathbf{T}_q and its inverse \mathbf{T}_q^{-1} :

$$\mathbf{V} = \mathbf{T}_q = [\mathbf{t}_1 \quad \cdots \quad \mathbf{t}_q] \quad \text{and} \quad \mathbf{W}^T = \mathbf{T}_q^{-1} = \begin{bmatrix} \mathbf{l}_1^T \\ \vdots \\ \mathbf{l}_q^T \end{bmatrix}. \quad (4.11)$$

In this case, the following applies

$$\mathbf{l}_i^T \mathbf{t}_j = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}, \quad (4.12)$$

thus $\mathbf{W}^T \mathbf{V} = \mathbf{I}_q$ holds, meaning that \mathbf{V} and \mathbf{W} are biorthogonal. This leads to

$$\mathbf{l}_k^T \mathbf{A} = \lambda_k \mathbf{l}_k^T \Leftrightarrow \mathbf{T}^{-1} \mathbf{A} = \mathbf{\Lambda} \mathbf{T}^{-1} \Rightarrow \mathbf{L}^T = \mathbf{T}^{-1}. \quad (4.13)$$

Thus, the special choice $\mathbf{W}^T = \mathbf{T}_q^{-1}$ yields biorthogonal projection matrices.

Regardless of the case: modal reduction can be interpreted as a projection onto the subspace of the most dominant (right) eigenvectors, orthogonal to the subspace of the left eigenvectors.

 sssMOR function(s): modalMor

4.4 Reduction of unstable systems

The dominance analysis from section 4.2 only makes sense for asymptotically stable eigenvalues, because eigenvalues to the right or on the imaginary axis are always to be regarded as dominant and have to be kept in the reduced model.

Indeed, this applies to all model reduction procedures: If a model is not asymptotically stable, a splitting similar to a modal reduction should first be performed, in which the *asymptotically stable* (“s”) subsystem is separated from the *unstable/antistable* (“a”) subsystem

$$\begin{aligned} \frac{d}{dt} \begin{bmatrix} \mathbf{z}_s \\ \mathbf{z}_a \end{bmatrix} &= \begin{bmatrix} \mathbf{\Lambda}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{\Lambda}_a \end{bmatrix} \begin{bmatrix} \mathbf{z}_s \\ \mathbf{z}_a \end{bmatrix} + \begin{bmatrix} \hat{\mathbf{b}}_s \\ \hat{\mathbf{b}}_a \end{bmatrix} u \\ y &= \begin{bmatrix} \hat{\mathbf{c}}_s^T & \hat{\mathbf{c}}_a^T \end{bmatrix} \begin{bmatrix} \mathbf{z}_s \\ \mathbf{z}_a \end{bmatrix}. \end{aligned} \quad (4.14)$$

Subsequently, a reduction of the stable subsystem can be performed, for example with modal reduction, balanced truncation or Krylov subspace methods. Finally, the unstable subsystem should be reintegrated into the reduced model.

$$\begin{aligned} \frac{d}{dt} \begin{bmatrix} \mathbf{z}_{s,r} \\ \mathbf{z}_a \end{bmatrix} &= \begin{bmatrix} \mathbf{A}_{s,r} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Lambda}_a \end{bmatrix} \begin{bmatrix} \mathbf{z}_{s,r} \\ \mathbf{z}_a \end{bmatrix} + \begin{bmatrix} \mathbf{b}_{s,r} \\ \hat{\mathbf{b}}_a \end{bmatrix} u \\ y &= \begin{bmatrix} \mathbf{c}_{s,r}^T & \hat{\mathbf{c}}_a^T \end{bmatrix} \begin{bmatrix} \mathbf{z}_{s,r} \\ \mathbf{z}_a \end{bmatrix}. \end{aligned} \quad (4.15)$$

 MATLAB function(s): stabsep

4.5 Residualization

A disadvantage of modal reduction is that generally no steady-state accuracy can be guaranteed. This means that the amplitude response at frequency $s = 0$, which corresponds to the stationary final value of the step response $y(t \rightarrow \infty)$, of the original and reduced model can deviate from each other.

However, since steady-state accuracy plays an important role in many applications, it can be guaranteed by the *residualization* method. This strategy is summarized below and can be read in more detail in [Ant05, p.285]. Since the problem of steady-state accuracy can

generally occur in approximations in which part of the state variables are truncated, it is presented here in general form.

The starting point is any partitioned state-space representation of the form

$$\begin{aligned} \frac{d}{dt} \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix} &= \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix} u, \\ y &= \begin{bmatrix} \mathbf{c}_1^T & \mathbf{c}_2^T \end{bmatrix} \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix}, \end{aligned} \quad (4.16)$$

in which the relevant dynamics are represented by the state directions \mathbf{z}_1 and the remaining dynamics are to be neglected. The usual approach at this point would be a *truncation*, which leads to the reduced model

$$\begin{aligned} \dot{\mathbf{z}}_1 &= \mathbf{A}_{11} \mathbf{z}_1 + \mathbf{b}_1 u, \\ y_r &= \mathbf{c}_1^T \mathbf{z}_1, \end{aligned} \quad (4.17)$$

which generally has no steady-state accuracy.

By neglecting the dynamics of the state directions in \mathbf{z}_2 , i.e. by assuming $\dot{\mathbf{z}}_2 = \mathbf{0}$, the state vector \mathbf{z}_2 can be represented as a function of \mathbf{z}_1 .

$$\mathbf{z}_2 \approx -\mathbf{A}_{22}^{-1} (\mathbf{A}_{21} \mathbf{z}_1 + \mathbf{b}_2 u), \quad (4.18)$$

where \mathbf{A}_{22} must be invertible. Inserting \mathbf{z}_2 in (4.16) finally yields the reduced model by *residualization*

$$\begin{aligned} \dot{\mathbf{z}}_1 &= \underbrace{\left(\mathbf{A}_{11} - \mathbf{A}_{12} \mathbf{A}_{22}^{-1} \mathbf{A}_{21} \right)}_{\mathbf{A}_r} \mathbf{z}_1 + \underbrace{\left(\mathbf{b}_1 - \mathbf{A}_{12} \mathbf{A}_{22}^{-1} \mathbf{b}_2 \right)}_{\mathbf{b}_r} u, \\ y_r &= \underbrace{\left(\mathbf{c}_1^T - \mathbf{c}_2^T \mathbf{A}_{22}^{-1} \mathbf{A}_{21} \right)}_{\mathbf{c}_r^T} \mathbf{z}_1 + \underbrace{\left(-\mathbf{c}_2^T \mathbf{A}_{22}^{-1} \mathbf{b}_2 \right)}_{d_r} u, \end{aligned} \quad (4.19)$$

where the following holds

$$G(0) = \mathbf{c}^T (-\mathbf{A})^{-1} \mathbf{b} = \mathbf{c}_r^T (-\mathbf{A}_r)^{-1} \mathbf{b}_r + d_r = G_r(0). \quad (4.20)$$

The proof follows directly from the inversion of block matrices and can be read in [FN82]. Note that by residualization one achieves steady-state accuracy, but generally at the cost of accuracy at high frequencies:




$$G(s \rightarrow \infty) = \lim_{s \rightarrow \infty} \mathbf{c}^T (s\mathbf{I} - \mathbf{A})^{-1} \mathbf{b} = 0 \neq d_r = G_r(s \rightarrow \infty) \quad (4.21)$$

4.6 Properties of modal truncation

- + The reduced model is available in modal coordinates.
- + From the stability of the original system it also follows the stability of the reduced system, since the eigenvalues of the reduced system are a subset of the eigenvalues of the original system.

- + Eigenvalues and modal coordinates are exactly preserved. This is especially advantageous, if certain eigenvalues and their eigenmodes are of special importance and should be exactly preserved in the reduced model (e.g. unstable eigenvalues).
- + The reduced state vector has a physical meaning, namely, it describes the direction along certain modal coordinates.
 - The choice of the reduced coordinates is restricted to the set of eigenvectors of the original model. This can sometimes be a limitation, especially if other state directions play a greater role in the transfer behavior (see exercise).
 - The Litz dominance measure considers only the contribution amount of the individual eigenvalues to the transfer behavior. However, there are systems in which compensation effects occur that cannot be captured by this measure. In these cases, the reduction result may become unsatisfactory.
 - The main disadvantage of the method is the high memory requirement $\mathcal{O}(n^2)$. Even if the system matrix \mathbf{A} is sparse, the matrix of the eigenvectors \mathbf{T} is generally dense. Moreover, a high computational effort $\mathcal{O}(n^3)$ is required for the solution of the eigenvalue problem (Schur decomposition, cf. [GV96]), wherefore the method cannot be readily applied to very high-dimensional models without further “tweaks”.

Implementation for large-scale models. Modal reduction as presented in this chapter is only suitable for medium-sized models. However, there are some extensions and adjustments for high-dimensional, sparse models: In general, the entire eigendecomposition $\mathbf{T}^{-1}\mathbf{A}\mathbf{T} = \mathbf{\Lambda}$ of the FOM is not computed, since the matrix $\mathbf{T} \in \mathbb{C}^{n \times n}$ is usually dense! Instead, *power methods* or *Krylov methods* can be used to *iteratively* calculate eigenspaces for certain eigenvalues, for example for those with the largest or smallest magnitude. The interested reader may have a look at the MATLAB functions `eig`, `eigs` as well as at the books [Dem97] and [Saa03].

	MATLAB function(s):	<code>eigs</code>
	SSS function(s):	<code>eigs</code>
	SSSMOR function(s):	<code>modalMor</code>

Selection criteria for (dominant) eigenmodes. As mentioned before, for large-scale systems, only a few eigenvectors are generally computed using power iteration methods or the Arnoldi/Lanczos iteration (Krylov methods). In the first-order case, only the eigenvectors associated to the eigenvalues fulfilling a certain criterion (e.g. smallest magnitude (`'sm'`), largest real part (`'lr'`), etc.) or closest to a given complex shift σ (`eigs(A, 6, SIGMA)`) are calculated. In the second-order case, the eigenmodes associated to the smallest eigenfrequencies (`'sm'` or `SIGMA=0`) or to a specific, relevant frequency range are often computed.

4.7 Modal truncation for second-order systems

As mentioned at the beginning of this chapter, modal truncation was originally developed for second-order models arising in the context of structural dynamics. Therefore, we briefly explain this reduction concept for second-order systems in the following.

Starting point is a linear, second-order model of the form

$$\begin{aligned} \mathbf{M}\ddot{\mathbf{z}}(t) + \mathbf{D}\dot{\mathbf{z}}(t) + \mathbf{K}\mathbf{z}(t) &= \mathbf{g}u(t), \\ y(t) &= \mathbf{1}^T \mathbf{z}(t), \end{aligned} \quad (4.22)$$

with symmetric, positive definite mass and stiffness matrices \mathbf{M} , \mathbf{K} , and (mostly) symmetric damping matrix \mathbf{D} . Depending on the damping, two different cases can be distinguished.

Proportional or zero damping

For *proportional* (aka. *Rayleigh*) damping, where $\mathbf{D} = \alpha\mathbf{M} + \beta\mathbf{K}$ with $\alpha, \beta \geq 0$ is commonly used, or for *zero* damping $\mathbf{D} = \mathbf{0}$, the second-order system (4.22) can be diagonalized and thus brought into modal coordinates by using the so-called *quadratic eigenvalue problem*.

We exemplarily derive the quadratic eigenvalue problem for the undamped case, where $\mathbf{D} = \mathbf{0}$ and the homogeneous system is considered ($u(t) = 0$). The ansatz for the displacements and accelerations is then given by

$$\mathbf{z}(t) = \sum_{k=1}^n \boldsymbol{\phi}_k e^{\lambda_k t}, \quad \ddot{\mathbf{z}}(t) = \sum_{k=1}^n \boldsymbol{\phi}_k \lambda_k^2 e^{\lambda_k t}, \quad (4.23)$$

where $\lambda_k = \delta_k \pm i\omega_k$. If zero damping is considered, then $\delta_k = 0$, yielding $\lambda_k = \pm i\omega_k$ and $\lambda_k^2 = -\omega_k^2$. Inserting the above ansatz into the equations of motion (4.22) delivers

$$\begin{aligned} -\omega_k^2 \mathbf{M} \sum_{k=1}^n \boldsymbol{\phi}_k e^{\pm i\omega_k t} + \mathbf{K} \sum_{k=1}^n \boldsymbol{\phi}_k e^{\pm i\omega_k t} &= \mathbf{0}, \\ \iff (\mathbf{K} - \omega_k^2 \mathbf{M}) \boldsymbol{\phi}_k e^{\pm i\omega_k t} &= \mathbf{0}. \end{aligned} \quad (4.24)$$

This equation holds for all $e^{\pm i\omega_k t}$, yielding the quadratic eigenvalue problem

$$(\mathbf{K} - \omega_k^2 \mathbf{M}) \boldsymbol{\phi}_k = \mathbf{0} \iff \mathbf{K}\boldsymbol{\Phi} - \mathbf{M}\boldsymbol{\Phi}\boldsymbol{\Omega}^2 = \mathbf{0}. \quad (4.25)$$

Hereby, the eigenfrequencies and eigenmodes $\{\omega_k, \boldsymbol{\phi}_k\}_{k=1}^n$ are encoded in the matrices $\boldsymbol{\Omega} = \text{diag}(\omega_1, \dots, \omega_n) \in \mathbb{R}^{n \times n}$ and $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, \dots, \boldsymbol{\phi}_n] \in \mathbb{R}^{n \times n}$. Typically, the eigenmodes are normalized using the inner product weighted by \mathbf{M} such that

$$\begin{aligned} \boldsymbol{\phi}_i^T \mathbf{M} \boldsymbol{\phi}_j &= \delta_{ij}, & \boldsymbol{\phi}_i^T \mathbf{K} \boldsymbol{\phi}_j &= \omega_i^2 \delta_{ij}, \\ \iff \boldsymbol{\Phi}^T \mathbf{M} \boldsymbol{\Phi} &= \mathbf{I}, & \boldsymbol{\Phi}^T \mathbf{K} \boldsymbol{\Phi} &= \boldsymbol{\Omega}^2. \end{aligned} \quad (4.26)$$

To sum up: in this case, the eigenvalues λ_k are either purely imaginary ($\mathbf{D} = \mathbf{0}$) or complex conjugated (proportional damping). The eigenfrequencies ω_k and the eigenmodes $\boldsymbol{\phi}_k$ are *real*.

General damping

For general damping $\mathbf{D} \neq \mathbf{0}$, the quadratic eigenvalue problem cannot be used anymore. Instead, the second-order model (4.22) has firstly to be reformulated as a first-order model of dimension $2n$ in explicit

$$\begin{aligned} \begin{bmatrix} \dot{\mathbf{z}} \\ \ddot{\mathbf{z}} \end{bmatrix} &= \overbrace{\begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{D} \end{bmatrix}}^{\mathbf{A}} \begin{bmatrix} \mathbf{z} \\ \dot{\mathbf{z}} \end{bmatrix} + \overbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1}\mathbf{g} \end{bmatrix}}^{\mathbf{b}} u, \\ y &= \underbrace{\begin{bmatrix} \mathbf{1}^T & \mathbf{0}^T \end{bmatrix}}_{\mathbf{c}^T} \begin{bmatrix} \mathbf{z} \\ \dot{\mathbf{z}} \end{bmatrix}, \end{aligned} \quad (4.27)$$

or implicit form (\mathbf{F} is a degree of freedom; $\mathbf{F} = \alpha\mathbf{I}$, $\alpha = 10^3, \dots, 10^6$, $\mathbf{F} = -\mathbf{K}$, $\mathbf{F} = \mathbf{M}$, etc.)

$$\begin{aligned} \overbrace{\begin{bmatrix} \mathbf{F} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \end{bmatrix}}^{\mathbf{E}} \begin{bmatrix} \dot{\mathbf{z}} \\ \ddot{\mathbf{z}} \end{bmatrix} &= \overbrace{\begin{bmatrix} \mathbf{0} & \mathbf{F} \\ -\mathbf{K} & -\mathbf{D} \end{bmatrix}}^{\mathbf{A}} \begin{bmatrix} \mathbf{z} \\ \dot{\mathbf{z}} \end{bmatrix} + \overbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{g} \end{bmatrix}}^{\mathbf{b}} u, \\ y &= \underbrace{\begin{bmatrix} \mathbf{1}^T & \mathbf{0}^T \end{bmatrix}}_{\mathbf{c}^T} \begin{bmatrix} \mathbf{z} \\ \dot{\mathbf{z}} \end{bmatrix}. \end{aligned} \quad (4.28)$$

After the reformulation, the first-order eigenvalue problems

$$\begin{aligned} (\mathbf{A} - \lambda_k \mathbf{I})\mathbf{t}_k = \mathbf{0} &\Leftrightarrow \mathbf{A}\mathbf{T} = \mathbf{T}\mathbf{\Lambda}, & \text{or} & \quad (\mathbf{A} - \lambda_k \mathbf{E})\mathbf{t}_k = \mathbf{0} \Leftrightarrow \mathbf{A}\mathbf{T} = \mathbf{E}\mathbf{T}\mathbf{\Lambda}, \\ \mathbf{1}_k^T(\mathbf{A} - \lambda_k \mathbf{I}) = \mathbf{0}^T &\Leftrightarrow \mathbf{L}^T \mathbf{A} = \mathbf{\Lambda} \mathbf{L}^T, & \text{or} & \quad \mathbf{1}_k^T(\mathbf{A} - \lambda_k \mathbf{E}) = \mathbf{0}^T \Leftrightarrow \mathbf{L}^T \mathbf{A} = \mathbf{\Lambda} \mathbf{L}^T \mathbf{E}, \end{aligned} \quad (4.29)$$

have to be solved, in order to compute the complex eigenvalues λ_k , and the *complex* right and left eigenvectors \mathbf{t}_k and $\mathbf{1}_k^T$.

Note again that for a symmetric matrix \mathbf{A} (in the implicit case: for symmetric \mathbf{A} and \mathbf{E}), *only one* eigenvalue problem has to be solved.

To sum up: in this general case, the eigenvalues λ_k are *complex-valued* and the eigenvectors \mathbf{t}_k and $\mathbf{1}_k^T$ are also *complex-valued*. This leads to complex projection matrices $\mathbf{V}, \mathbf{W} \in \mathbb{C}^{n \times q}$, which generally yield complex-valued reduced quantities $\mathbf{A}_r, \mathbf{b}_r$ and \mathbf{c}_r^T . This is usually not desired, especially if the original matrices are real-valued. Nevertheless, if the eigenvalues come in complex conjugate pairs (as it is mostly the case), then a splitting of the eigenvectors \mathbf{t}_k and $\mathbf{1}_k^T$ in real and imaginary part is possible, in order to obtain real projection matrices $\mathbf{V}, \mathbf{W} \in \mathbb{R}^{n \times q}$ spanning the same subspaces as their complex counterparts.

Chapter 5

Balanced Truncation

In this chapter, we learn a widely used method for model reduction, which originated in the 1980s. It is the so-called *Balanced Truncation* (BT) or *Truncated Balanced Realization* (TBR): first, the system is brought – by a state transformation – into a balanced representation, where the “unimportant state variables” can be easily identified; then, the reduced model is obtained by truncation.

The basic idea of TBR can be formulated as follows: find a state representation, where every single state variable x_i is just as strong or weak controllable as it is observable (= balancing). Neglect then the state variables with the slightest contribution to the input-output behavior (= truncation).

The terms controllability and observability are repeated in the following comparison.

Controllability

- Relation between input $u(t)$ and state $\mathbf{x}(t)$
- Matrices \mathbf{A} and \mathbf{b}
- Complete controllability: the system can be transferred from the initial state $\mathbf{x}(t_0) = \mathbf{0}$ to any arbitrary final state $\mathbf{x}(t) = \mathbf{x}_e$ in finite time by a suitable control input $u(t)$.

Observability

- Relation between state $\mathbf{x}(t)$ and output $y(t)$
- Matrices \mathbf{A} and \mathbf{c}^T
- Complete observability: let $u(t)$ be known; then the initial state $\mathbf{x}(t_0) = \mathbf{x}_0$ can be uniquely determined over a finite time period solely from the measurement of $y(t)$.

5.1 Controllability

Which states are well controllable? In order to assess controllability, we need a suitable measure; in the following, this should be the energy¹: the states $\mathbf{x}(t_e) = \mathbf{x}_e$, which can be reached with little energy from the origin $\mathbf{x}(t = 0) = \mathbf{0}$, are well / easily controllable. In order to identify these states, we assume an asymptotically stable system which dwells at the equilibrium $\mathbf{x} = \mathbf{0}$. To carry the system to the final state \mathbf{x}_e , control input energy is needed. The goal is therefore to determine the minimum energy required in order to bring the system to the final state \mathbf{x}_e .

This task can be formulated as an optimization problem as known from optimal control (cf. e.g. “Moderne Methoden der Regelungstechnik 2”):

$$\begin{aligned} \text{Cost functional:} \quad & J = \int_0^{t_e} u^2(t) dt \\ \text{with the constraint:} \quad & \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{b}u(t) \\ \text{and the boundary conditions:} \quad & \mathbf{x}(t = 0) = \mathbf{0} \quad \text{and} \quad \mathbf{x}(t = t_e) = \mathbf{x}_e, \end{aligned}$$

where t_e is free and is therefore part of the optimization. To avoid this, we could choose $t_e \rightarrow \infty$; but then the integral would no longer exist, since control input energy would be permanently needed to keep the system in \mathbf{x}_e . Instead, we have to apply a trick: we consider the problem in negative time direction: $\tau = t_e - t$; $\frac{d\tau}{dt} = -1 \Leftrightarrow dt = -d\tau$. This way, the sign on the right side of the ODE changes and the cost functional becomes:

$$J = \int_{t=0}^{t=t_e} u^2(t) dt = \int_{\tau=t_e}^{\tau=0} -u^2(\tau) d\tau = \int_0^{t_e} u^2(\tau) d\tau. \quad (5.1)$$

For times $\tau > t_e$, the system remains in the equilibrium without control input energy, wherefore we can now also set $t_e \rightarrow \infty$ to eliminate the optimization variable t_e . This yields the optimization problem in the well-known form:

$$\begin{aligned} \text{Cost functional:} \quad & J = \int_0^{\infty} u^2(\tau) d\tau \\ \text{with the constraint:} \quad & \dot{\mathbf{x}}(\tau) = -\mathbf{A}\mathbf{x}(\tau) - \mathbf{b}u(\tau) \\ \text{and the boundary conditions:} \quad & \mathbf{x}(\tau = 0) = \mathbf{x}_e \quad \text{and} \quad \mathbf{x}(\tau \rightarrow \infty) = \mathbf{0}. \end{aligned}$$

The solution provides the optimal control law (see e.g. “Moderne Methoden 2”):

$$u_{\text{opt}}(\tau) = -\frac{1}{2} \left(-\mathbf{b}^T \right) \mathbf{P}\mathbf{x}(\tau) = \frac{1}{2} \mathbf{b}^T \mathbf{P}\mathbf{x}(\tau), \quad (5.2)$$

where \mathbf{P} is the solution of a Riccati equation. However, there is no weighting $\mathbf{x}^T \mathbf{Q} \mathbf{x}$ of the states in the cost functional. Thus, the corresponding term is omitted, and the Riccati

¹Note that the quantity designated here as energy is actually to be understood as *generalized energy* and must e.g. not possess the unit Joule. It should be a measure for the required control input energy, cf. e.g. equation (5.1).

equation becomes the Lyapunov equation:

$$(-\mathbf{A}^T)\mathbf{P} + \mathbf{P}(-\mathbf{A}) - \frac{1}{2}\mathbf{P}(-\mathbf{b})(-\mathbf{b}^T)\mathbf{P} = \mathbf{0}, \quad (5.3)$$

$$\Leftrightarrow -\mathbf{A}^T\mathbf{P} - \mathbf{P}\mathbf{A} - \frac{1}{2}\mathbf{P}\mathbf{b}\mathbf{b}^T\mathbf{P} = \mathbf{0}, \quad (5.4)$$

$$\Leftrightarrow 2\mathbf{P}^{-1}\mathbf{A}^T + 2\mathbf{A}\mathbf{P}^{-1} + \mathbf{b}\mathbf{b}^T = \mathbf{0}. \quad (5.5)$$

5.1.1 Controllability Gramian

Definition 18. We define $\mathbf{W}_c := 2\mathbf{P}^{-1}$ as the controllability Gramian. It is the solution of the Lyapunov equation:

$$\mathbf{A}\mathbf{W}_c + \mathbf{W}_c\mathbf{A}^T + \mathbf{b}\mathbf{b}^T = \mathbf{0}. \quad (5.6)$$

Furthermore, it can be shown (see Exercise), that:

$$\mathbf{W}_c = \int_0^\infty e^{\mathbf{A}t}\mathbf{b}\mathbf{b}^T e^{\mathbf{A}^T t} dt. \quad (5.7)$$



MATLAB function(s): `gram(sys, 'c')`, `lyap`, `lyapchol`

5.1.2 Energy consideration

Theorem 4. The controllability Gramian specifies the minimum value of the cost functional $J^*(\mathbf{x}_e)$ depending on the final state \mathbf{x}_e :

$$J^*(\mathbf{x}_e) = \mathbf{x}_e^T \mathbf{W}_c^{-1} \mathbf{x}_e. \quad (5.8)$$

Proof. If the optimal control law $u(\tau) = \frac{1}{2}\mathbf{b}^T \mathbf{P}\mathbf{x}(\tau) = \mathbf{b}^T \mathbf{W}_c^{-1} \mathbf{x}(\tau)$ is inserted into the state equation, then this yields the dynamics:

$$\dot{\mathbf{x}}(\tau) = (-\mathbf{A} - \mathbf{b}\mathbf{b}^T \mathbf{W}_c^{-1}) \mathbf{x}(\tau). \quad (5.9)$$

Furthermore, the following term is also needed:

$$\frac{d}{dt} (\mathbf{x}^T(\tau) \mathbf{W}_c^{-1} \mathbf{x}(\tau)) = \dot{\mathbf{x}}^T(\tau) \mathbf{W}_c^{-1} \mathbf{x}(\tau) + \mathbf{x}^T(\tau) \mathbf{W}_c^{-1} \dot{\mathbf{x}}(\tau) \quad (5.10)$$

$$\stackrel{(5.9)}{=} \mathbf{x}^T(\tau) (-\mathbf{A}^T \mathbf{W}_c^{-1} - \mathbf{W}_c^{-1} \mathbf{A} - 2\mathbf{W}_c^{-1} \mathbf{b}\mathbf{b}^T \mathbf{W}_c^{-1}) \mathbf{x}(\tau) \quad (5.11)$$

$$= -\mathbf{x}^T(\tau) \mathbf{W}_c^{-1} (\mathbf{W}_c \mathbf{A}^T + \mathbf{A} \mathbf{W}_c + 2\mathbf{b}\mathbf{b}^T) \mathbf{W}_c^{-1} \mathbf{x}(\tau) \quad (5.12)$$

$$\stackrel{(5.6)}{=} -\mathbf{x}^T(\tau) \mathbf{W}_c^{-1} \mathbf{b}\mathbf{b}^T \mathbf{W}_c^{-1} \mathbf{x}(\tau) \quad (5.13)$$

The term $\mathbf{x}^T(\tau)\mathbf{W}_c^{-1}\mathbf{x}(\tau)$ is thus antiderivative of (5.13), and it follows:

$$J^*(\mathbf{x}_e) = \int_0^\infty u_{\text{opt}}^2(\tau) d\tau = \int_0^\infty u_{\text{opt}}^T(\tau)u_{\text{opt}}(\tau) d\tau \quad (5.14)$$

$$\stackrel{(5.2)}{=} \int_0^\infty \mathbf{x}^T(\tau)\mathbf{W}_c^{-1}\mathbf{b}\mathbf{b}^T\mathbf{W}_c^{-1}\mathbf{x}(\tau)d\tau \stackrel{(5.13)}{=} \left[-\mathbf{x}^T(\tau)\mathbf{W}_c^{-1}\mathbf{x}(\tau)\right]_0^\infty \quad (5.15)$$

$$= -\mathbf{x}^T(\tau \rightarrow \infty)\mathbf{W}_c^{-1}\mathbf{x}(\tau \rightarrow \infty) + \mathbf{x}^T(0)\mathbf{W}_c^{-1}\mathbf{x}(0) = \mathbf{0} + \mathbf{x}_e^T\mathbf{W}_c^{-1}\mathbf{x}_e \quad (5.16)$$

□

The product $\mathbf{x}_e^T\mathbf{W}_c^{-1}\mathbf{x}_e$ describes the minimum energy required to reach the final state \mathbf{x}_e . This result allows the following interpretation: After singular value decomposition of $\mathbf{W}_c = \mathbf{U}\Sigma\mathbf{U}^T$ with $\sigma_1 \geq \sigma_2 \dots \geq \sigma_n$ (or eigendecomposition, since $\mathbf{W}_c = \mathbf{W}_c^T > \mathbf{0}$), the following holds:

- The first singular vectors \mathbf{u}_i point in the directions that are “easily” controllable: σ_1 is maximal, and with $\mathbf{W}_c^{-1} = \mathbf{U}\Sigma^{-1}\mathbf{U}^T$ follows that $\mathbf{u}_1^T\mathbf{W}_c^{-1}\mathbf{u}_1 = \frac{1}{\sigma_1}$ is minimal.
- The corresponding singular values σ_i describe how much energy is needed to reach these directions in the state-space.

The interested reader is referred to [ZDG+96, Ch. 3.2] and [Ant05, Ch. 4.2.1] for more information.

5.2 Observability

Which states are well observable? The observability should also be assessed by means of the energy: the states $\mathbf{x}(t=0) = \mathbf{x}_0$, which provide for $u(t) \equiv 0$ the most energy at the output in the steady-state process, are well observable. In order to identify these states, we define the observability Gramian.

5.2.1 Observability Gramian

Definition 19. We define the solution $\mathbf{W}_o = \mathbf{W}_o^T > \mathbf{0}$ of the Lyapunov equation:

$$\mathbf{A}^T\mathbf{W}_o + \mathbf{W}_o\mathbf{A} + \mathbf{c}\mathbf{c}^T = \mathbf{0}, \quad (5.17)$$

as observability Gramian. Analogously it holds:

$$\mathbf{W}_o = \int_0^\infty e^{\mathbf{A}^T t}\mathbf{c}\mathbf{c}^T e^{\mathbf{A}t} dt. \quad (5.18)$$



MATLAB function(s): `gram(sys, 'o')`, `lyap`, `lyapchol`

5.2.2 Energy consideration

The observability Gramian describes how well or badly observable the state variables are. This relationship stems from the following considerations. The unactuated system is:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) \\ y(t) &= \mathbf{c}^T \mathbf{x}(t).\end{aligned}\tag{5.19}$$

The solution $y(t) = \mathbf{c}^T \mathbf{x}(t)$ with the initial value $\mathbf{x}(t=0) = \mathbf{x}_0$ is given by:

$$y(t) = \mathbf{c}^T e^{\mathbf{A}t} \mathbf{x}_0.\tag{5.20}$$

Thus, the initial state \mathbf{x}_0 provides the following energy at the output (measured in the \mathcal{L}_2 -norm):

$$\|y\|_{\mathcal{L}_2}^2 = \int_0^\infty y^T(t) y(t) dt\tag{5.21}$$

$$\stackrel{(5.20)}{=} \mathbf{x}_0^T \int_0^\infty (e^{\mathbf{A}^T t} \mathbf{c} \mathbf{c}^T e^{\mathbf{A}t}) dt \mathbf{x}_0\tag{5.22}$$

$$\stackrel{(5.18)}{=} \mathbf{x}_0^T \mathbf{W}_o \mathbf{x}_0.\tag{5.23}$$

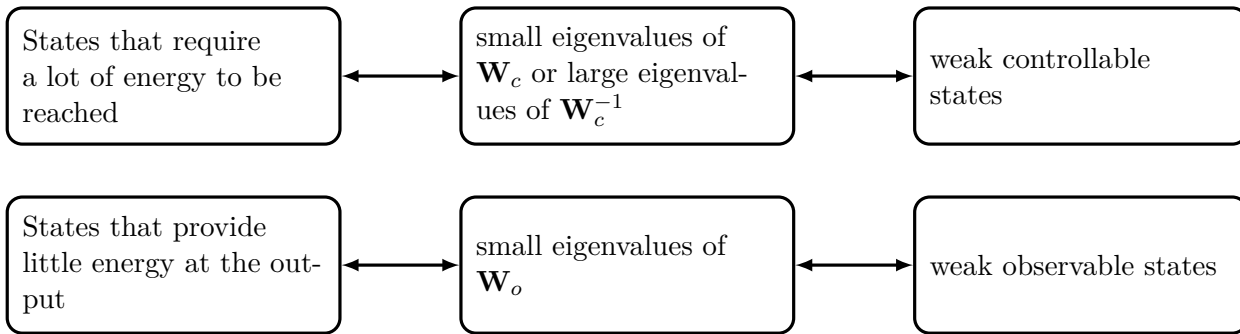
The product $\mathbf{x}_0^T \mathbf{W}_o \mathbf{x}_0$ describes the energy that the state \mathbf{x}_0 provides by observing/measuring the output. After the singular value decomposition of \mathbf{W}_o , the following analog interpretation follows:

- The first singular vectors point in the directions that generate most energy at the output, i. e. the directions that are “strongly” observable.
- The corresponding singular values (= eigenvalues) describe, how large this energy is.

The interested reader is referred to [ZDG+96, Ch. 3.2] and [Ant05, Ch. 4.2.2] for more information.

5.3 Balancing

The previous considerations about the energy flow in the system can be summarized in the following chart:



As follows from the observations of Exercise 4, only those state variables that are both poorly controllable and poorly observable may be neglected. Thus, the information of \mathbf{W}_c and \mathbf{W}_o must be considered at the same time! A direction, that e. g. is well controllable, does not necessarily have to be a well observable direction.

In order to find directions that are as well controllable as observable, we have to introduce a state transformation of the form $\mathbf{z} = \mathbf{T}\mathbf{x}$, since this way linear combinations of state variables \mathbf{x} are introduced as new state variables \mathbf{z} . This first leads to the question: how do the Gramians change with such a state transformation?

Theorem 5. Let $\mathbf{W}_c, \mathbf{W}_o$ be the Gramian matrices in the coordinates \mathbf{x} . The transformed Gramians $\widehat{\mathbf{W}}_c, \widehat{\mathbf{W}}_o$, in the coordinates $\mathbf{z} = \mathbf{T}\mathbf{x}$ are:

$$\widehat{\mathbf{W}}_c = \mathbf{T}\mathbf{W}_c\mathbf{T}^T, \quad (5.24)$$

$$\widehat{\mathbf{W}}_o = \mathbf{T}^{-T}\mathbf{W}_o\mathbf{T}^{-1}. \quad (5.25)$$

(Proof see Exercise.)

This means that the eigenvalues of \mathbf{W}_c and \mathbf{W}_o change with the representation of the system! However, one can see that the eigenvalues of the product $\mathbf{W}_c\mathbf{W}_o$ are invariant under state transformations

$$\widehat{\mathbf{W}}_c\widehat{\mathbf{W}}_o = \mathbf{T}(\mathbf{W}_c\mathbf{W}_o)\mathbf{T}^{-1}. \quad (5.26)$$

The eigenvalues of $\mathbf{W}_c\mathbf{W}_o$ are thus independent of the representation of the system. They are invariant system variables to describe the transfer behavior.

Definition 20. The Hankel singular values σ_i (HSV) of a system are defined as

$$\sigma_i = \sqrt{\lambda_i(\mathbf{W}_c\mathbf{W}_o)}, \quad (5.27)$$

and represent a measure for the energy transfer from the inputs to the outputs. They are independent from the state representation.

Reminder: We are looking for state directions which are *both* badly controllable *as well as* badly observable! The solution is provided by the *balanced representation*, which can be defined using the HSVs.

Definition 21. A system is called **balanced**, if: $\mathbf{W}_c = \mathbf{W}_o = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n)$ with $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n$. That means: both Gramian matrices are equal, are in diagonal form and have the HSVs as entries.

In a balanced system, all state variables are therefore just as well controllable as observable. One possible calculation of the state transformation \mathbf{T} , which transforms the system to balanced representation, is given in the following theorem.

Theorem 6. The state transformation \mathbf{T} to balanced representation can be calculated as follows:

1. Cholesky decomposition² of $\mathbf{W}_c = \mathbf{S}\mathbf{S}^T$ and $\mathbf{W}_o = \mathbf{R}\mathbf{R}^T$
2. SVD of $\mathbf{R}^T\mathbf{S} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$
3. $\mathbf{T} := \mathbf{\Sigma}^{-1/2}\mathbf{U}^T\mathbf{R}^T$ and $\mathbf{T}^{-1} := \mathbf{S}\mathbf{V}\mathbf{\Sigma}^{-1/2}$

Note: Due to this transformation, the state variables lose their physical meaning!

Proof.

$$\begin{aligned}
 \widehat{\mathbf{W}}_c &= \mathbf{T}\mathbf{W}_c\mathbf{T}^T \quad \text{with } \mathbf{W}_c = \mathbf{S}\mathbf{S}^T \\
 &= \mathbf{\Sigma}^{-\frac{1}{2}}\mathbf{U}^T\mathbf{R}^T (\mathbf{S}\mathbf{S}^T) \mathbf{R}\mathbf{U}\mathbf{\Sigma}^{-\frac{1}{2}} \\
 &= \mathbf{\Sigma}^{-\frac{1}{2}} \underbrace{\mathbf{U}^T\mathbf{U}}_{=\mathbf{I}} \mathbf{\Sigma} \underbrace{\mathbf{V}^T\mathbf{V}}_{=\mathbf{I}} \mathbf{\Sigma} \underbrace{\mathbf{U}^T\mathbf{U}}_{=\mathbf{I}} \mathbf{\Sigma}^{-\frac{1}{2}} \\
 &= \mathbf{\Sigma}^{-\frac{1}{2}} \mathbf{\Sigma}^2 \mathbf{\Sigma}^{-\frac{1}{2}} = \mathbf{\Sigma}.
 \end{aligned}$$

The proof of $\widehat{\mathbf{W}}_o = \mathbf{\Sigma}$ is analogous. □

In the balanced representation, the state variables are ordered according to their importance for the transfer behavior, i.e. according to the HSVs. Thus, we have found a measure to evaluate which state variables we can keep in the reduced system and which we can neglect.



MATLAB function(s): `hsvd`, `hsvplot`, `balreal`

5.4 Truncated Balanced Realization

Once the balanced representation of the system has been found (i.e. after state transformation with \mathbf{T}), the reduction can finally be carried out. To this end, the balanced system is

²The *Cholesky* decomposition is only defined for symmetric positive definite matrices. The factors \mathbf{S} , \mathbf{R} are lower triangular, cf. [GV96].

Algorithm 1 Order reduction by balanced truncation

1. Solve the Lyapunov equations:

$$\begin{aligned}\mathbf{A}\mathbf{W}_c + \mathbf{W}_c\mathbf{A}^T + \mathbf{b}\mathbf{b}^T &= \mathbf{0}, \\ \mathbf{A}^T\mathbf{W}_o + \mathbf{W}_o\mathbf{A} + \mathbf{c}\mathbf{c}^T &= \mathbf{0}.\end{aligned}$$

2. Cholesky decompositions: $\mathbf{W}_c = \mathbf{S}\mathbf{S}^T$, $\mathbf{W}_o = \mathbf{R}\mathbf{R}^T$, $\mathbf{S}, \mathbf{R} \in \mathbb{R}^{n \times n}$.
3. SVD: $\mathbf{R}^T\mathbf{S} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \sum_{i=1}^n \sigma_i \mathbf{u}_i \mathbf{v}_i^T$
4. Choose the first q columns of \mathbf{U} and \mathbf{V} : $\mathbf{U}_q = [\mathbf{u}_1, \dots, \mathbf{u}_q]$, $\mathbf{V}_q = [\mathbf{v}_1, \dots, \mathbf{v}_q]$, and the first Hankel singular values: $\mathbf{\Sigma}_q = \text{diag}(\sigma_1, \dots, \sigma_q)$. Calculate the projection matrices \mathbf{V}_{BT} and \mathbf{W}_{BT} :

$$\begin{aligned}\mathbf{V}_{\text{BT}} &:= \mathbf{S}\mathbf{V}_q\mathbf{\Sigma}_q^{-1/2} \in \mathbb{R}^{n \times q}, \\ \mathbf{W}_{\text{BT}}^T &:= \mathbf{\Sigma}_q^{-1/2}\mathbf{U}_q^T\mathbf{R}^T \in \mathbb{R}^{q \times n}.\end{aligned}$$

5. Since \mathbf{V}_{BT} and \mathbf{W}_{BT} are biorthogonal, the reduced system of order $q \ll n$ is:

$$\mathbf{A}_r = \mathbf{W}_{\text{BT}}^T \mathbf{A} \mathbf{V}_{\text{BT}}, \quad \mathbf{b}_r = \mathbf{W}_{\text{BT}}^T \mathbf{b}, \quad \mathbf{c}_r = \mathbf{c}^T \mathbf{V}_{\text{BT}}.$$

5.5 Properties of Balanced Truncation

- + The reduced model is also balanced and a minimal realization.
- + From the stability of the original system, the stability of the reduced system follows: the reduced Gramians are diagonal and have the largest HSVs as entries \Rightarrow positive definite Lyapunov solutions \Rightarrow stability proof by Lyapunov.
- + It can be shown that:

$$\|G(s) - G_r(s)\|_\infty \leq 2 \sum_{i=q+1}^n \sigma_i. \quad (5.31)$$

The error in the \mathcal{H}_∞ -norm is thus limited by the double sum of the truncated Hankel singular values. This means that a priori, i. e. before computing the reduced system, the approximation error can be controlled by the choice of the reduced order q . This is a considerable advantage of the method, since the reliable estimation of the approximation error is one of the most difficult challenges in model reduction.

- The main disadvantage of TBR is the high memory requirement $\mathcal{O}(n^2)$. Even if the system matrix \mathbf{A} is sparse, the Gramians and their Cholesky factors are generally dense! In addition, a high computational effort $\mathcal{O}(n^3)$ is required for the solution of the Lyapunov equations and the SVD of $\mathbf{R}^T \mathbf{S}$, wherefore the method cannot be readily applied to very high-dimensional models without further “tweaks”.

Example: When increasing the order of the original system from $n = 500$ to $n = 1000$ (factor 2), balanced truncation requires about $4\times$ more memory and $8\times$ more time. On today’s standard computers, TBR is feasible up to a maximum order of $\approx 5000 - 10000$, since the lower triangular, dense Cholesky factors and the dense Gramians may already need more than 9 GB of memory!

Implementation for large-scale models. Balanced Truncation as it is presented in Algorithm 1 or implemented in MATLAB (`balred`, `balancmr`) is only suitable for medium-sized models. However, there are some extensions and adjustments for high-dimensional, sparse models, like e.g. the *Low-Rank Square-Root Balanced Truncation* (LRSR-BT). The main difference lies in the approximate solution $\mathbf{W}_c \approx \mathbf{Z}_c \mathbf{Z}_c^T$ and $\mathbf{W}_o \approx \mathbf{Z}_o \mathbf{Z}_o^T$ of the two Lyapunov equations using low-rank factors of the dimension $\mathbf{Z}_c \in \mathbb{R}^{n \times q_c}$, $\mathbf{Z}_o \in \mathbb{R}^{n \times q_o}$ with $q_c, q_o \ll n$. The most important methods in this regard are the *Low Rank Alternating Direction Implicit* (LR-ADI) (see [Pen00]) and the *Rational Krylov Subspace Method* (RKSM) (see [DS11]). Both approaches are very closely related to the Krylov subspace methods presented in the next chapter. For the sake of completeness, it should be noted that the advantages of Balanced Truncation described above are generally lost due to the approximate solution of the Lyapunov equations.

Our `lyapchol` command uses the third-party toolbox M-M.E.S.S.³ (LR-ADI) or our own implemented `crksm` function to approximately solve large sparse Lyapunov equations.

³Available under <https://www.mpi-magdeburg.mpg.de/projects/mess>

Approximate solution of Lyapunov equations via `mess_lradi` or `crksm`

```
sss function(s):  lyapchol (mess_lradi, crksm)
```

This `sss` command plays a crucial role for many functions. For instance, `lyapchol` is needed for the computation of the \mathcal{H}_2 -norm of a LTI system via `norm(sys, 2)` (cf. (2.24)). Moreover, it is also employed within `tbr`.

Truncated Balanced Realization (TBR)

```
sssMOR function(s):  tbr
```


Chapter 6

Krylov Subspace Methods

In this chapter we learn another important approach for model order reduction: the *Krylov subspace methods*. The basic idea is the *local* approximation of the transfer behavior. Just as an arbitrary function $f(x)$ can be approximated by a Taylor series around a point x_0 , we also want to approximate the transfer function $G(s)$ around a frequency s_0 . This gives us indeed a powerful tool, which is however not easy to use! Professor Athanasios C. Antoulas, an expert in the field, expresses this fact like this:

The advantage of Krylov subspace methods is that there are many degrees of freedom.

The disadvantage of Krylov subspace methods is that there are many degrees of freedom.

The problem is that it is not always perfectly clear, how to appropriately choose the degrees of freedom of the method. Therefore, the Krylov subspace methods are still subject of intensive research. The method is also called *Moment Matching*.

6.1 Moment Matching

The transfer function $G(s)$ of a system is a rational function of the Laplace variable s and can be expanded into an infinite Taylor series around a (complex) frequency s_0 with the (still to be determined) coefficients m_i :

$$G(s) = -m_0 - m_1(s - s_0) - m_2(s - s_0)^2 - \dots = -\sum_{i=0}^{\infty} m_i (s - s_0)^i \quad (6.1)$$

The minus before m_i is a sign convention (definition issue). The goal of moment matching is to find a reduced model, such that the first q coefficients m_i , $i = 0, 1, \dots, q - 1$ of the Taylor series of the original and reduced model match. It is thus a local approximation around the frequency s_0 .

6.1.1 Moments

Definition 22. The moments m_i of a transfer function (or of a dynamic system) around an expansion point (aka. shift) s_0 are defined as the negative coefficients of the Taylor series representation around s_0 :

$$G(s) = - \sum_{i=0}^{\infty} m_i (s - s_0)^i, \quad (6.2)$$

or in other words:

$$m_i = - \frac{1}{i!} \left. \frac{d^i G(s)}{ds^i} \right|_{s=s_0} \quad i = 0, 1, \dots \quad (6.3)$$

For a dynamic system in state-space representation the following holds:

$$m_i = \mathbf{c}^T (\mathbf{A} - s_0 \mathbf{I})^{-(i+1)} \mathbf{b}. \quad (6.4)$$

Proof. The transfer function is

$$\begin{aligned} G(s) &= \mathbf{c}^T (s\mathbf{I} - \mathbf{A})^{-1} \mathbf{b} = \mathbf{c}^T (s\mathbf{I} - \mathbf{A} + s_0\mathbf{I} - s_0\mathbf{I})^{-1} \mathbf{b} = \mathbf{c}^T \underbrace{[(s - s_0)\mathbf{I} - (\mathbf{A} - s_0\mathbf{I})]^{-1}}_{\hat{\mathbf{A}}} \mathbf{b} \\ &= \mathbf{c}^T \left[\hat{\mathbf{A}} (\hat{s}\hat{\mathbf{A}}^{-1} - \mathbf{I}) \right]^{-1} \mathbf{b} = -\mathbf{c}^T (\mathbf{I} - \hat{s}\hat{\mathbf{A}}^{-1})^{-1} \hat{\mathbf{A}}^{-1} \mathbf{b}. \end{aligned}$$

We use the so-called Neumann series: $(\mathbf{I} - \mathbf{T})^{-1} = \sum_{i=0}^{\infty} \mathbf{T}^i$. With $\mathbf{T} := \hat{s}\hat{\mathbf{A}}^{-1}$ follows:

$$G(s) = -\mathbf{c}^T \sum_{i=0}^{\infty} (\hat{s}\hat{\mathbf{A}}^{-1})^i \hat{\mathbf{A}}^{-1} \mathbf{b} = -\sum_{i=0}^{\infty} \mathbf{c}^T (\mathbf{A} - s_0\mathbf{I})^{-(i+1)} \mathbf{b} (s - s_0)^i.$$

The comparison with (6.2) delivers the result: $m_i = \mathbf{c}^T (\mathbf{A} - s_0\mathbf{I})^{-(i+1)} \mathbf{b}$. □


The goal of *moment matching* is to match the first q moments of the original and reduced system:

$$\begin{aligned} G(s) &= -m_0 - m_1(s - s_0) - \dots - m_{q-1}(s - s_0)^{q-1} - m_q(s - s_0)^q - \dots \\ G_r(s) &= -m_{r,0} - m_{r,1}(s - s_0) - \dots - m_{r,q-1}(s - s_0)^{q-1} - m_{r,q}(s - s_0)^q - \dots \\ \text{with } m_i &= m_{r,i}, \quad i = 0, 1, \dots, q-1 \\ m_j &\neq m_{r,j}, \quad j = q, q+1, \dots \end{aligned}$$

Since we look at systems in state-space representation, we are seeking for a reduced system, such that the following applies:

$$\mathbf{c}^T (\mathbf{A} - s_0\mathbf{I})^{-(i+1)} \mathbf{b} \stackrel{!}{=} \mathbf{c}_r^T (\mathbf{A}_r - s_0\mathbf{I})^{-(i+1)} \mathbf{b}_r, \quad i = 0, 1, \dots, (q-1) \quad (6.5)$$

It will turn out that we can achieve this, if we use so-called *Krylov subspaces* for the projection matrices \mathbf{V} and \mathbf{W} .

 sssMOR function(s): moments

6.1.2 Krylov subspaces

Definition 23. The **Krylov subspace** \mathcal{K}_q of a matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$ and a vector $\mathbf{v} \in \mathbb{R}^n$ is generally defined as:

$$\mathcal{K}_q(\mathbf{M}, \mathbf{v}) = \text{span} \left\{ \mathbf{v}, \mathbf{M}\mathbf{v}, \mathbf{M}^2\mathbf{v}, \dots, \mathbf{M}^{q-1}\mathbf{v} \right\}. \quad (6.6)$$

A Krylov space denotes a q -dimensional subspace $\mathcal{K}_q \subseteq \mathbb{R}^n$, for which (infinitely) many bases $\mathbf{V} \in \mathbb{R}^{n \times q}$ exist. Krylov subspaces were introduced to solve large eigenvalue problems and linear systems of equations iteratively. They are also often used to approximately solve linear matrix equations, such as e.g. Lyapunov equations (cf. Ch. 5). For model reduction we need two particular Krylov subspaces, which are defined below.

Definition 24. $\mathcal{K}_q((\mathbf{A} - s_0\mathbf{I})^{-1}, (\mathbf{A} - s_0\mathbf{I})^{-1}\mathbf{b})$ is called **input Krylov subspace** and $\mathcal{K}_q((\mathbf{A} - s_0\mathbf{I})^{-T}, (\mathbf{A} - s_0\mathbf{I})^{-T}\mathbf{c})$ is called **output Krylov subspace** around the expansion point s_0 .

6.1.3 Implicit Moment Matching

Using the previous definitions, we can already formulate the (theoretical) solution to the moment matching problem. The practical implementation follows in Section 6.2.

Theorem 7 (Moment Matching 1). *Form the columns of \mathbf{V} a q -dimensional basis of the input Krylov subspace around s_0 , and is $\mathbf{W} \in \mathbb{R}^{n \times q}$ arbitrary such that $\det(\mathbf{A}_r - s_0\mathbf{I}) \neq 0$, then the first q moments of the original and the reduced system around s_0 match.*

Proof. The first moment of the reduced system is:

$$m_{r,0} = \mathbf{c}_r^T (\mathbf{A}_r - s_0\mathbf{I})^{-1} \mathbf{b}_r \quad (6.7)$$

$$= \mathbf{c}^T \mathbf{V} \left((\mathbf{W}^T \mathbf{V})^{-1} \mathbf{W}^T \mathbf{A} \mathbf{V} - s_0 \mathbf{I} \right)^{-1} (\mathbf{W}^T \mathbf{V})^{-1} \mathbf{W}^T \mathbf{b} \quad (6.8)$$

$$= \mathbf{c}^T \mathbf{V} \left((\mathbf{W}^T \mathbf{V})^{-1} (\mathbf{W}^T \mathbf{A} \mathbf{V} - s_0 \mathbf{W}^T \mathbf{V}) \right)^{-1} (\mathbf{W}^T \mathbf{V})^{-1} \mathbf{W}^T \mathbf{b} \quad (6.9)$$

$$= \mathbf{c}^T \mathbf{V} (\mathbf{W}^T \mathbf{A} \mathbf{V} - s_0 \mathbf{W}^T \mathbf{V})^{-1} \mathbf{W}^T \mathbf{b} \quad (6.10)$$

$$= \mathbf{c}^T \mathbf{V} (\mathbf{W}^T \mathbf{A} \mathbf{V} - s_0 \mathbf{W}^T \mathbf{V})^{-1} \mathbf{W}^T \underbrace{(\mathbf{A} - s_0\mathbf{I})(\mathbf{A} - s_0\mathbf{I})^{-1}}_{=\mathbf{I}} \mathbf{b} \quad (6.11)$$

Since $(\mathbf{A} - s_0\mathbf{I})^{-1}\mathbf{b}$ is the first direction of the Krylov subspace $\mathcal{K}_q((\mathbf{A} - s_0\mathbf{I})^{-1}, (\mathbf{A} - s_0\mathbf{I})^{-1}\mathbf{b})$ and \mathbf{V} represents a basis for this Krylov subspace, the following applies:

$$\exists \mathbf{r}_0 \in \mathbb{R}^q : (\mathbf{A} - s_0\mathbf{I})^{-1}\mathbf{b} = \mathbf{V}\mathbf{r}_0. \quad (6.12)$$

It follows:

$$m_{r,0} = \mathbf{c}^T\mathbf{V}(\mathbf{W}^T\mathbf{A}\mathbf{V} - s_0\mathbf{W}^T\mathbf{V})^{-1}\mathbf{W}^T(\mathbf{A} - s_0\mathbf{I})\mathbf{V}\mathbf{r}_0 \quad (6.13)$$

$$= \mathbf{c}^T\mathbf{V}(\mathbf{W}^T\mathbf{A}\mathbf{V} - s_0\mathbf{W}^T\mathbf{V})^{-1}(\mathbf{W}^T\mathbf{A}\mathbf{V} - s_0\mathbf{W}^T\mathbf{V})\mathbf{r}_0 \quad (6.14)$$

$$= \mathbf{c}^T\mathbf{V}\mathbf{r}_0 \stackrel{(6.12)}{=} \mathbf{c}^T(\mathbf{A} - s_0\mathbf{I})^{-1}\mathbf{b} = m_0. \quad (6.15)$$

The proof for the remaining $q-1$ moments can be carried out analogously using mathematical induction. \square

Theorem 8 (Moment Matching 2). *Form the columns of \mathbf{W} a q -dimensional basis of the output Krylov subspace around s_0 , and is $\mathbf{V} \in \mathbb{R}^{n \times q}$ arbitrary such that $\det(\mathbf{A}_r - s_0\mathbf{I}) \neq 0$, then the first q moments of the original and the reduced system around s_0 match.*

Proof. The proof follows from the duality, i. e. in the previous proof, one must substitute \mathbf{A} with \mathbf{A}^T , \mathbf{b} with \mathbf{c} (\mathbf{B} with \mathbf{C}^T in the MIMO case) and \mathbf{V} with \mathbf{W} , and the condition

$$\exists \mathbf{l}_0 \in \mathbb{R}^q : (\mathbf{A} - s_0\mathbf{I})^{-T}\mathbf{c} = \mathbf{W}\mathbf{l}_0 \quad (6.16)$$

must be used. \square

Corollary 3 (Moment Matching 3). *Form the columns of \mathbf{V} a basis of the input Krylov subspace **and** the columns of \mathbf{W} the basis of the output Krylov subspace, then the first $2q$ moments of the original and the reduced system around s_0 match.*

6.1.4 Markov parameters¹

If one wants to approximate the transient behavior, i.e. for $t \rightarrow 0$ and $s \rightarrow \infty$, then the so-called Markov parameters must match. They represent the counterpart of the moments for $s_0 \rightarrow \infty$.

Definition 25. *The Markov parameters M_i of a system are defined as:*

$$M_i = \left. \frac{d^i g(t)}{dt^i} \right|_{t=0} = \mathbf{c}^T \mathbf{A}^i \mathbf{b} \quad i = 0, 1, \dots \quad (6.17)$$

and it can be shown that:

$$G(s) = \sum_{i=0}^{\infty} M_i \frac{1}{s^{i+1}} = M_0 \frac{1}{s} + M_1 \frac{1}{s^2} + M_2 \frac{1}{s^3} + \dots \quad (6.18)$$

¹The section on Markov parameters is given only for the sake of completeness and without proofs.

In order to match Markov parameters in the reduced system, one needs the following theorem.

Theorem 9 (Moment Matching 4). *Form the columns of \mathbf{V} a basis of the Krylov subspace $\mathcal{K}_q(\mathbf{A}, \mathbf{b})$ and the columns of \mathbf{W} a basis of the Krylov subspace $\mathcal{K}_q(\mathbf{A}^T, \mathbf{c})$, then the first $2q$ Markov parameters of the original and of the reduced system match.*

6.1.5 Several expansion points

It is possible to guarantee moment matching around more than one expansion point (including $s_0 \rightarrow \infty$). For this purpose (aka. *multipoint moment matching*), the projection matrices \mathbf{V} and \mathbf{W} are formed such that they span the union of appropriate Krylov subspaces.

Example: $\mathbf{V} = [\mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3]$, whereby:

$$\begin{aligned} \text{range}(\mathbf{V}_1) &= \mathcal{K}_{q_1} \left((\mathbf{A} - s_1 \mathbf{I})^{-1}, (\mathbf{A} - s_1 \mathbf{I})^{-1} \mathbf{b} \right), \\ \text{range}(\mathbf{V}_2) &= \mathcal{K}_{q_2} \left((\mathbf{A} - s_2 \mathbf{I})^{-1}, (\mathbf{A} - s_2 \mathbf{I})^{-1} \mathbf{b} \right), \\ \text{range}(\mathbf{V}_3) &= \mathcal{K}_{q_3}(\mathbf{A}, \mathbf{b}). \end{aligned}$$

Then, q_1 moments around s_1 match, q_2 moments around s_2 match, and additionally q_3 Markov parameters match.

6.1.6 Summary

Generally, $2q$ is the maximum number of matching moments, since a transfer function of order q has only $2q$ degrees of freedom (in partial fractional decomposition: q residues in the numerator plus q poles in the denominator). According to the number of moments that match, a distinction is made between *one-sided* and *two-sided* methods:

One-sided reduction		Two-sided reduction
\mathbf{V} input Krylov space and $\mathbf{W} := \mathbf{V}$	\mathbf{W} output Krylov space and $\mathbf{V} := \mathbf{W}$	\mathbf{V} input Krylov space \mathbf{W} output Krylov space
q moments match		$2q$ moments match
State transformation in the original system changes the reduced dynamics!		$2q$ degrees of freedom fixed \Rightarrow reduced dynamics are unique!

Note: The choice of a one-sided or a two-sided reduction depends on the goals and priorities of the reduction. For instance, one can use the remaining degrees of freedom in \mathbf{W} (or in \mathbf{V}) to guarantee stability of the ROM or to preserve other system properties. On the other hand, one can use these degrees of freedom to match more moments to obtain a better accuracy.



SSSMOR function(s): `rk`, `crksm`

6.2 Computation of the matrices \mathbf{V} and \mathbf{W}

To guarantee moment matching, a matrix \mathbf{V} or \mathbf{W} must be calculated, whose columns span the desired Krylov subspace. The explicit calculation of the Krylov directions $(\mathbf{A} - s_0\mathbf{I})^{-i}\mathbf{b}$ has to be avoided, as this would lead to numerical problems.

Consider the example with $s_0 = 0$, i. e. $\mathbf{V} = [\mathbf{A}^{-1}\mathbf{b}, \dots, \mathbf{A}^{-q}\mathbf{b}]$. The individual Krylov directions could theoretically be determined as follows:

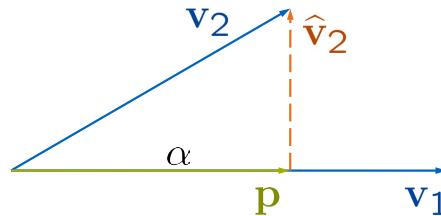
$$\mathbf{v}_1 = \mathbf{A}^{-1}\mathbf{b} \quad (6.19)$$

$$\mathbf{v}_i = \mathbf{A}^{-1}\mathbf{v}_{i-1}, \quad \text{for } i = 2, \dots, q \quad (6.20)$$

The problem lies in the successive multiplication by one and the same matrix. By a multiplication, the shares of an eigenvector are stretched by its own eigenvalue. This means that shares of eigenvectors corresponding to small eigenvalues are getting shorter and shorter, compared to shares of eigenvectors corresponding to large eigenvalues. The more the magnitude of the eigenvalues of \mathbf{A} differ, the faster this difference is potentiated. The directions \mathbf{v}_i become linearly dependent/parallel, so that after a few iterations there are no more q independent columns in \mathbf{V} ! Orthogonal matrices, i. e. $\mathbf{V}^T\mathbf{V} = \mathbf{I}$, are numerically best conditioned with the minimum condition number of 1. The aim of this section is therefore to find an orthonormal basis \mathbf{V} of a desired Krylov subspace. To this end, we first need the so-called Gram-Schmidt method.

6.2.1 Gram-Schmidt method

For example, two vectors \mathbf{v}_1 and \mathbf{v}_2 are given, which span a two-dimensional subspace. Now an orthogonal basis $\{\mathbf{v}_1, \hat{\mathbf{v}}_2\}$ of the same subspace is sought. The following figure illustrates the relationship between the vectors,



wherein $\alpha \in \mathbb{R}$ represents a suitable scaling of \mathbf{v}_1 :

$$\mathbf{p} = \alpha\mathbf{v}_1 \quad (6.21)$$

$$\hat{\mathbf{v}}_2 = \mathbf{v}_2 - \mathbf{p} \stackrel{(6.21)}{=} \mathbf{v}_2 - \alpha\mathbf{v}_1 \quad (6.22)$$

The aim is $\widehat{\mathbf{v}}_2 \perp \mathbf{v}_1$, i. e. $\widehat{\mathbf{v}}_2^T \mathbf{v}_1 = 0$:

$$(\mathbf{v}_2 - \alpha \mathbf{v}_1)^T \mathbf{v}_1 = 0 \quad (6.23)$$

$$\Leftrightarrow \alpha = \frac{\mathbf{v}_2^T \mathbf{v}_1}{\mathbf{v}_1^T \mathbf{v}_1} \quad (6.24)$$

$$\stackrel{(6.22)}{\implies} \widehat{\mathbf{v}}_2 = \mathbf{v}_2 - \frac{\mathbf{v}_2^T \mathbf{v}_1}{\mathbf{v}_1^T \mathbf{v}_1} \mathbf{v}_1 \quad (6.25)$$

The vectors $\{\mathbf{v}_1, \widehat{\mathbf{v}}_2\}$ span the same space as $\{\mathbf{v}_1, \mathbf{v}_2\}$, but they are orthogonal to each other. In order to obtain an *orthonormal* basis, we choose:

$$\mathbf{v}_1^* := \frac{\mathbf{v}_1}{\|\mathbf{v}_1\|}, \quad \mathbf{v}_2^* := \frac{\widehat{\mathbf{v}}_2}{\|\widehat{\mathbf{v}}_2\|}. \quad (6.26)$$

The resulting projection matrix $\mathbf{V} = [\mathbf{v}_1^*, \mathbf{v}_2^*]$ is then orthogonal, i. e. $\mathbf{V}^T \mathbf{V} = \mathbf{I}$. The whole procedure is summarized in the following algorithm for arbitrary subspaces.

Algorithm 2 Gram-Schmidt method

Input: $\mathbf{v}_1, \dots, \mathbf{v}_q$

Output: Orthonormal basis $\mathbf{v}_1^*, \dots, \mathbf{v}_q^*$

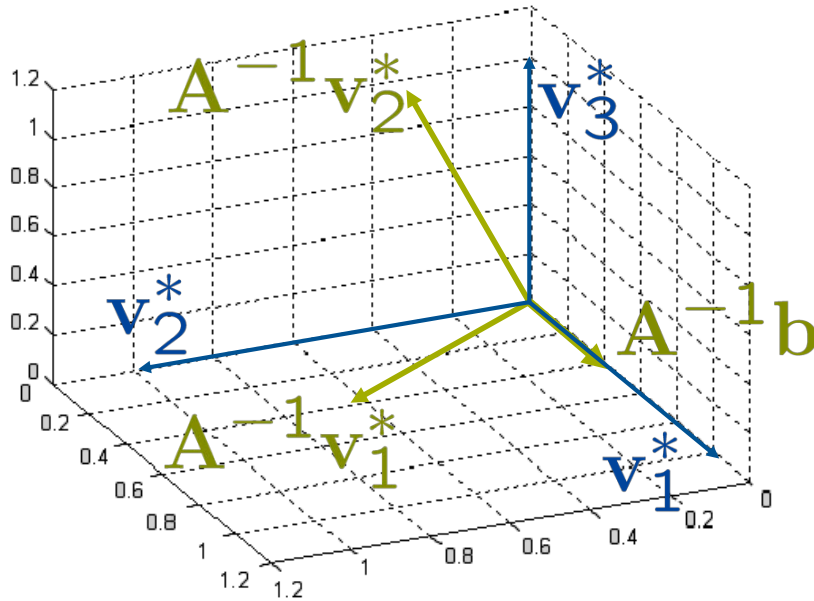
- 1: $\mathbf{v}_1^* = \frac{\mathbf{v}_1}{\|\mathbf{v}_1\|}$ // Normalization of the first vector
 - 2: **for** $i = 2$ to q **do**
 - 3: **for** $j = 1$ to $(i - 1)$ **do**
 - 4: $\mathbf{v}_i \leftarrow \mathbf{v}_i - \frac{\mathbf{v}_i^T \mathbf{v}_j}{\mathbf{v}_j^T \mathbf{v}_j} \mathbf{v}_j$ // Orthogonalization to all previous vectors
 - 5: **end for**
 - 6: $\mathbf{v}_i^* = \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|}$ // Normalization of the resulting vector
 - 7: **end for**
-



SSSMOR function(s): `gramSchmidt (in arnoldi)`

6.2.2 Arnoldi algorithm

The Gram-Schmidt method calculates an orthonormal basis for a given set of vectors. To improve the numerical conditioning of the iteration (6.20), the Gram-Schmidt process must be applied in each iteration. In the example (6.20), it means that – instead of $\mathbf{v}_3 := \mathbf{A}^{-1} \mathbf{v}_2$ – the orthogonal direction \mathbf{v}_2^* must be used: $\mathbf{v}_3 := \mathbf{A}^{-1} \mathbf{v}_2^*$. After that, \mathbf{v}_3 must be orthonormalized to \mathbf{v}_1^* and \mathbf{v}_2^* using the Gram-Schmidt method, in order to obtain \mathbf{v}_3^* . Then, the next direction must be determined: $\mathbf{v}_4 = \mathbf{A}^{-1} \mathbf{v}_3^*$; the Gram-Schmidt procedure is run again; and so on ... The following figure shows a graphic illustration of how the directions could look like.



The resulting orthonormal basis $\mathbf{V} := [\mathbf{v}_1^*, \mathbf{v}_2^*, \dots]$ spans the desired Krylov subspace (proof: see exercise). The combination of the Gram-Schmidt method with the iterative calculation of the next Krylov directions is summarized in the Arnoldi algorithm. But before the algorithm is given, we have to deal with another problem from a numerical point of view: to calculate a Krylov direction $(\mathbf{A} - s_0 \mathbf{I})^{-1} \mathbf{v}_i$, the inverse of the matrix $\mathbf{A} - s_0 \mathbf{I}$ is (theoretically) required. This, however, is numerically expensive ($\mathcal{O}(n^3)$), usually dense (thus memory intensive, $\mathcal{O}(n^2)$) and badly conditioned. For all these reasons, the inverse should absolutely be avoided! In order to avoid the explicit inversion, the problem can be equivalently reformulated into a linear system of equations (LSE):

$$\mathbf{v}_i = (\mathbf{A} - s_0 \mathbf{I})^{-1} \mathbf{v}_{i-1} \iff (\mathbf{A} - s_0 \mathbf{I}) \mathbf{v}_i = \mathbf{v}_{i-1} \quad (6.27)$$

This system of equations can be efficiently solved using e.g. *Gauss-Jordan elimination* (`mldivide`, “\”-operator in MATLAB). Since such a system of equations needs to be solved several times (only with different right-hand sides \mathbf{v}_i) for a Krylov basis, it makes sense to additionally perform a so-called *LU decomposition*. Hereby, an arbitrary matrix is decomposed into the product of an upper and lower triangular matrix (command `lu` in MATLAB):

$$\mathbf{A} = \mathbf{L}\mathbf{U} = \underbrace{\begin{pmatrix} \diagdown & & \\ & \diagdown & \\ & & \diagdown \end{pmatrix}}_{\mathbf{L}} \underbrace{\begin{pmatrix} \diagdown & & \\ & \diagdown & \\ & & \diagdown \end{pmatrix}}_{\mathbf{U}}$$



Systems of equations with triangular matrices can be easily solved by iterative insertion. Due to the triangular form of \mathbf{L} and \mathbf{U} the system of equations (6.27) is effectively solved in two steps, $(\mathbf{A} - s_0 \mathbf{I})^{-1} = (\mathbf{L}\mathbf{U})^{-1} = \mathbf{U}^{-1} \mathbf{L}^{-1}$:

1. Solve $\mathbf{L}\mathbf{y} = \mathbf{v}_{i-1}$, starting with the first entry of \mathbf{y} .
2. Solve $\mathbf{U}\mathbf{v}_i = \mathbf{y}$, starting with the last entry of \mathbf{v}_i .

These two steps can be briefly summarized in MATLAB notation with the “\”-operator:

$$\mathbf{LU}\mathbf{v}_i = \mathbf{v}_{i-1} \iff \mathbf{v}_i = \mathbf{U} \setminus (\mathbf{L} \setminus \mathbf{v}_{i-1})$$

The LU decomposition makes sense, because it only has to be performed once at the beginning. With the stored matrices \mathbf{L} and \mathbf{U} , the q systems of equations for q Krylov directions can be solved immediately! The numerically most expensive part of the reduction with Krylov subspaces is therefore the LU decomposition.

 MATLAB function(s): `lu, mldivide`
 SSS function(s): `solveLse`

Similar to `sss/lyapchol`, the command `solveLse` to solve LSEs $\mathbf{Ax} = \mathbf{b}$ plays a crucial role for many functions. For instance, it is needed for the computation of the frequency response of a LTI system via `sss/freqresp`. Moreover, it is also needed within `arnoldi`.



Algorithm 3 shows the *Arnoldi* algorithm for the efficient computation of an orthonormal basis of a Krylov subspace. It includes the LU decomposition, the Gram-Schmidt method and the iterative calculation of the Krylov directions.

Algorithm 3 Arnoldi algorithm (multimoment)

Input: Matrices \mathbf{A} , \mathbf{b} , expansion point s_0 , order q

Output: orthonormal basis of the Krylov subspace $\mathbf{v}_1^*, \dots, \mathbf{v}_q^*$

- 1: $\mathbf{L}, \mathbf{U} \leftarrow \text{lu}(\mathbf{A} - s_0\mathbf{I})$ // LU decomposition
 - 2: $\mathbf{v}_1 = \mathbf{U} \setminus (\mathbf{L} \setminus \mathbf{b})$ // first Krylov direction
 - 3: $\mathbf{v}_1^* = \frac{\mathbf{v}_1}{\|\mathbf{v}_1\|}$ // Normalization of the first vector
 - 4: **for** $i = 2$ to q **do**
 - 5: $\mathbf{v}_i = \mathbf{U} \setminus (\mathbf{L} \setminus \mathbf{v}_{i-1}^*)$ // Computation of the next Krylov direction
 - 6: $\mathbf{v}_i^* = \text{gramSchmidt}(\mathbf{v}_1^*, \dots, \mathbf{v}_{i-1}^*, \mathbf{v}_i)$ // Orthonormalization to previous directions
 - 7: **end for**
-

 SSS function(s): `solveLse`
 SSSMOR function(s): `arnoldi, gramSchmidt (rk)`

Please note that Algorithm 3 can be generalized to match Markov parameters, match moments at several expansion points, etc. (cf. implementation in the `sssMOR` toolbox). Further note that for every new expansion point s_0 , a new LU decomposition of the matrix $\mathbf{A} - s_0\mathbf{I}$ has to be performed! Hence: the *multipoint moment matching* strategy – where moments around different shifts are matched – is in general computationally more expensive than the *multimoment matching* strategy – where high-order moments are matched at a single shift. The combination of both strategies is mostly the best choice.

6.3 \mathcal{H}_2 -optimal reduction

We are now able to approximate a dynamic system locally in a numerically efficient way. But if the best possible approximation over the entire frequency range is desired, then the question arises how to choose the expansion points. One possibility is trying to find a ROM of desired fixed order q , which minimizes the \mathcal{H}_2 -error:

$$G_r(s) = \arg \min_{\deg(\tilde{G}_r)=q} \|G - \tilde{G}_r\|_{\mathcal{H}_2}. \quad (6.28)$$

Since this optimization problem is non-convex, the goal at first is to find a *local* solution. Considering the cost functional $J = \|G - \tilde{G}_r\|_{\mathcal{H}_2}$ and the analysis of its vanishing gradient, Meier and Luenberger derived in [ML67] first-order necessary optimality conditions for an \mathcal{H}_2 -optimal SISO reduced model:

$$G(-\bar{\lambda}_{r,i}) = G_r(-\bar{\lambda}_{r,i}), \quad (6.29a)$$

$$G'(-\bar{\lambda}_{r,i}) = G'_r(-\bar{\lambda}_{r,i}), \quad (6.29b)$$

for $i = 1, \dots, q$. These interpolatory conditions can be interpreted as follows: for an \mathcal{H}_2 -optimal ROM, i.e. a reduced model with minimal error in the \mathcal{H}_2 -norm, the first two moments at the mirror images of the reduced eigenvalues, i.e. at $s_0 \leftarrow -\bar{\lambda}_{r,i}$, must match. If one knew the reduced eigenvalues in advance, then one could choose the expansion points accordingly and perform a standard Krylov reduction. Since this is not the case, an iterative procedure is proposed with the *Iterative Rational Krylov Algorithm* (IRKA), first published in 2006 [GAB08]. Without illuminating the backgrounds in more detail, the algorithm is only presented here:

Algorithm 4 IRKA

Input: System $\mathbf{A}, \mathbf{b}, \mathbf{c}^T$, initial values $s_{0,i}$ (e. g. $s_{0,i} = 0$, $i = 1, \dots, q$), tolerance ε

Output: local \mathcal{H}_2 -optimal reduced system


- 1: **repeat:**
 - 2: $\mathbf{V} \leftarrow$ input Krylov subspace around the expansion points $s_{0,i}$
 - 3: $\mathbf{W} \leftarrow$ output Krylov subspace around the expansion points $s_{0,i}$
 - 4: $\mathbf{A}_r = (\mathbf{W}^T \mathbf{V})^{-1} \mathbf{W}^T \mathbf{A} \mathbf{V}$, $\mathbf{b}_r = (\mathbf{W}^T \mathbf{V})^{-1} \mathbf{W}^T \mathbf{b}$, $\mathbf{c}_r^T = \mathbf{c}^T \mathbf{V}$ // Project system
 - 5: $s_{0,i} \leftarrow -\bar{\lambda}_i(\mathbf{A}_r)$ // New expansion points from mirroring
 - 6: **until:** $|s_{0,i} + \bar{\lambda}_i(\mathbf{A}_r)| < \varepsilon$, $\forall i = 1, \dots, q$
-

One strength of IRKA is its simple implementation: One only has to calculate the reduced eigenvalues, mirror them along the imaginary axis, and then use them as new expansion points for the next iteration! Another advantage is that the shifts are *adaptively* chosen by the algorithm. The user only has to define the reduced order q , some initial shifts and a tolerance. The disadvantage is that new Krylov subspaces have to be calculated in each iteration, leading to an increased numerical effort compared to a standard Krylov reduction.

If IRKA converges, i.e. if the shifts no longer significantly change, then it converges against a local \mathcal{H}_2 optimum. Until today, there is no general proof for convergence. Practice, however, shows that IRKA actually converges in most cases.

Even after convergence of IRKA, it must be ensured that the reduced model is stable. After all, like most Krylov methods, the stability preservation is not guaranteed! In particular, it should be noted that the \mathcal{H}_2 -norm is not defined for unstable systems. Therefore, the above considerations are no longer valid and have to be adapted for unstable original models.

All these properties make IRKA – despite the lack of proofs – one of the most important methods of model reduction!

 SSSMOR function(s): `irka, cirka`



6.4 Properties of Krylov subspace methods

- + Numerically efficient, since no “expensive” calculations are necessary: apart from the LU-decompositions, only matrix-vector multiplications are involved.
- + The moments themselves are never explicitly calculated (which is anyway numerically ill-conditioned); one speaks of *implicit* moment matching.
- The stability of the reduced system is not guaranteed, i.e. even if the original system is stable, the reduced system does not necessarily have to be stable.
- There are no general, efficiently computable error bounds for the approximation quality; error bounds have been derived so far only for special cases.
- The result of a one-sided reduction ($\mathbf{W} = \mathbf{V}$) is not unique and depends on the original state-space representation, i. e. a state transformation $\mathbf{z} = \mathbf{T}\mathbf{x}$ in the original system changes the reduced system!
- The problem with the choice and number/multiplicity of expansion points in order to achieve a good approximation is not completely solved yet. As said before:


The advantage of Krylov subspace methods is that there are many degrees of freedom.

The disadvantage of Krylov subspace methods is that there are many degrees of freedom.


Implementation for very large-scale models. Krylov reduction as presented so far is indeed applicable to large-scale models. A sparse LU-decomposition $[\mathbf{L}, \mathbf{U}, \mathbf{P}, \mathbf{Q}, \mathbf{D}] = \text{lu}(\mathbf{A})$ together with a sparse *direct* solver (e.g. “\” in MATLAB) allow for orders $n \approx 10^4 - 10^5$. If very large-scale models ($n \geq 10^6$) shall be reduced, then *iterative* solvers such as e.g. the *generalized minimal residual method* (GMRES) or the preconditioned conjugate gradient (PCG) can be used instead.

 MATLAB function(s): `ilu, gmres, pcg, bicg, bicgstab, cgs, etc.`
 SSS function(s): `iterSolve (in solveLse)`

Selection of shifts. The appropriate choice of expansion points is in fact the biggest issue in Krylov subspace methods. Lamentably, there is no universal rule, since the selection of suitable shifts is usually problem-dependent and should therefore be customized for the system at hand. Nevertheless, a few guidelines can be given: if lower frequencies are of interest, then moments at $s_0 = 0$ should be matched. If higher frequencies are relevant, then $s_0 = \infty$ should be chosen. Apart from that, one could choose the mirror images of some eigenvalues of the FOM as expansion points, i.e. `s0 = -eigs(sys, q) .'` or spread the shifts over the interesting frequency range, e.g. `s0=logspace(log10(wmin), log10(wmax), q)`. Some other more sophisticated procedures, such as the heuristic Penzl method, the Wachspress approach or exploiting the Ritz values (i.e. the eigenvalues of a reduced system), can be used to adaptively get new shifts (e.g. within `crksm`).

 `sssMOR` function(s): `initializeShifts, getShifts, rkOp`

Stability-preserving schemes. Krylov subspace methods are generally not stability preserving. However, there exist some algorithms which specifically select the degrees of freedom to preserve stability. One possibility is given by the Iterative SVD-Rational Krylov Algorithm (ISRK). The algorithm combines an SVD-based method like balanced truncation with rational Krylov. The approach requires computing the observability Gramian \mathbf{W}_o (only one Lyapunov equation needs thus to be solved) and an input Krylov subspace \mathbf{V} . Then, the method selects $\mathbf{W} := \mathbf{W}_o \mathbf{V}$. This choice preserves the stability in the reduced system. Another related possibility is to restrict the search to so-called *pseudo-optimal* shifts. An \mathcal{H}_2 -pseudo-optimal ROM is a model that only fulfills the interpolatory condition (6.29a), and which has eigenvalues as mirror images of the shifts. Such a ROM can be obtained by the *Pseudo-Optimal Rational Krylov* (PORK) algorithm, which is exploited within the *Stability Preserving Adaptive Rational Krylov* (SPARK) algorithm. For more information regarding these procedures the reader is referred to [WPL13; Pan+13; Pan14].

 `sssMOR` function(s): `isrk, spark (porkV, porkW)`

6.5 Comparison between TBR and Krylov

The properties of balanced truncation and Krylov subspace methods are compared again in the following table:

	TBR	Krylov
Advantages	<ul style="list-style-type: none"> + automatable + stability is preserved + a priori error bound 	<ul style="list-style-type: none"> + efficient numerical computation + suitable for large systems ($n \approx 10^6$) + many degrees of freedom
Disadvantages	<ul style="list-style-type: none"> - computationally expensive - high memory consumption - suitable only for small to medium-sized systems, $n \approx 5000$ 	<ul style="list-style-type: none"> - stability is generally not preserved - no error bounds - many degrees of freedom: e. g. expansion point(s), multiplicity, ...

Chapter 7

Further Topics

The following topics serve as supplement and outlook for the contents described so far, and will also be covered during the lecture depending on time and interest. Therefore, only the contents from this chapter which were also covered in lecture and/or exercise will be relevant for the exam.

7.1 Systems in implicit state-space representation

The theory presented can be extended to implicit state-space representations of the form:

$$\begin{aligned}\mathbf{E}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{b}u(t), \\ y(t) &= \mathbf{c}^T \mathbf{x}(t) + du(t),\end{aligned}\tag{7.1}$$

with $\det(\mathbf{E}) \neq 0$. The naive procedure is to multiply the state equation by \mathbf{E}^{-1} and then proceed as before. However, this approach is mostly not advisable due to the inverse! Firstly, the calculation of the inverse is numerically ill-conditioned. Secondly, the sparsity of the matrices is usually lost through the inversion and subsequent multiplication. This is shown in Figure 7.1 for the benchmark model `heat-cont`.

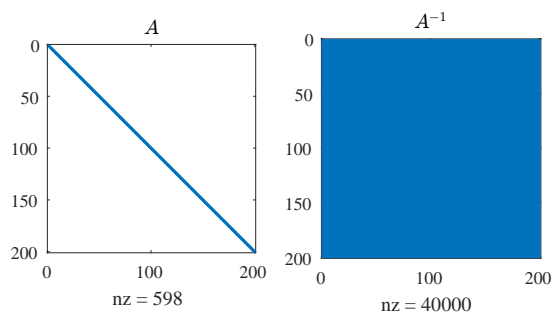


Figure 7.1: Sparsity pattern for the matrix \mathbf{A} and its inverse.

For these reasons, model reduction methods are usually extended to deal with implicit state-space representations. In fact, eigenvalue problems, Lyapunov equations and Krylov subspaces can be generalized to the case with \mathbf{E} matrix.

Generalized eigenvalue problem. The generalized eigenvalue problems are given by

$$(\mathbf{A} - \lambda_k \mathbf{E}) \mathbf{t}_k = \mathbf{0} \Leftrightarrow \mathbf{A} \mathbf{T} = \mathbf{E} \mathbf{T} \mathbf{\Lambda}, \quad (7.2a)$$

$$\mathbf{l}_k^T (\mathbf{A} - \lambda_k \mathbf{E}) = \mathbf{0}^T \Leftrightarrow \mathbf{L}^T \mathbf{A} = \mathbf{\Lambda} \mathbf{L}^T \mathbf{E}, \quad (7.2b)$$

to compute the eigenvalues λ_k and the right and left eigenvectors $\mathbf{t}_k, \mathbf{l}_k^T$ of the pencil (\mathbf{A}, \mathbf{E}) .

Generalized Lyapunov equations. The generalized Lyapunov equations for computing the controllability and observability Gramians are

$$\mathbf{A} \mathbf{W}_c \mathbf{E}^T + \mathbf{E} \mathbf{W}_c \mathbf{A}^T + \mathbf{b} \mathbf{b}^T = \mathbf{0}, \quad (7.3a)$$

$$\mathbf{A}^T \mathbf{W}_o \mathbf{E} + \mathbf{E}^T \mathbf{W}_o \mathbf{A} + \mathbf{c} \mathbf{c}^T = \mathbf{0}. \quad (7.3b)$$

Note that the impulse response of the system (7.1) is $g(t) = \mathbf{c}^T e^{\mathbf{E}^{-1} \mathbf{A} t} \mathbf{E}^{-1} \mathbf{b} \sigma(t) + d \delta(t)$.

Generalized input/output Krylov subspaces. The generalized input and output Krylov subspaces are

$$\mathcal{K}_q \left((\mathbf{A} - s_0 \mathbf{E})^{-1} \mathbf{E}, (\mathbf{A} - s_0 \mathbf{E})^{-1} \mathbf{b} \right), \quad (7.4a)$$

$$\mathcal{K}_q \left((\mathbf{A}^T - s_0 \mathbf{E}^T)^{-1} \mathbf{E}^T, (\mathbf{A}^T - s_0 \mathbf{E}^T)^{-1} \mathbf{c} \right). \quad (7.4b)$$

Note that the transfer function of the system (7.1) is $G(s) = \mathbf{c}^T (s \mathbf{E} - \mathbf{A})^{-1} \mathbf{b} + d$.

The projection matrices \mathbf{V} and \mathbf{W} can be calculated using these generalized equations, thereby avoiding the inverse \mathbf{E}^{-1} . The reduced model in implicit form is then:

$$\begin{aligned} \overbrace{\mathbf{W}^T \mathbf{E} \mathbf{V}}^{\mathbf{E}_r} \dot{\mathbf{x}}_r(t) &= \overbrace{\mathbf{W}^T \mathbf{A} \mathbf{V}}^{\mathbf{A}_r} \mathbf{x}_r(t) + \overbrace{\mathbf{W}^T \mathbf{b}}^{\mathbf{b}_r} u(t), \\ y_r(t) &= \underbrace{\mathbf{c}^T \mathbf{V}}_{\mathbf{c}_r^T} \mathbf{x}_r(t) + du(t), \end{aligned} \quad (7.5)$$

where the projector is now given by $\mathbf{P} = \mathbf{E} \mathbf{V} (\mathbf{W}^T \mathbf{E} \mathbf{V})^{-1} \mathbf{W}^T$. Hence, the system is projected onto the subspace $\mathcal{U} = \text{range}(\mathbf{E} \mathbf{V})$, orthogonal to the subspace $\mathcal{W} = \text{range}(\mathbf{W})$.

This is only mentioned here for the sake of completeness; nevertheless, the derivations of the generalized equations go beyond the scope of the lecture. The important message is: from a computational point of view, the inversion of \mathbf{E} becomes prohibited in the large-scale setting and should therefore be circumvented by using adapted/generalized algorithms! The extension of Krylov subspace methods to the case $\mathbf{E} \neq \mathbf{I}$ is extensively explained, e.g. in [Gri97; BG14]. The generalization of Balanced Truncation for $\mathbf{E} \neq \mathbf{I}$ is not consistent, and sometimes even wrong, in different references. Therefore, its generalization is briefly summarized with our nomenclature in the following section.

7.2 Balanced Truncation for $\mathbf{E} \neq \mathbf{I}$

We now consider a (MIMO) state-space model in implicit form

$$\begin{aligned}\mathbf{E}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t),\end{aligned}\tag{7.6}$$

which – due to $\det \mathbf{E} \neq 0$ – is equivalent to the explicit state-space representation

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \overbrace{\mathbf{E}^{-1}\mathbf{A}}^{\mathbf{A}_e}\mathbf{x}(t) + \overbrace{\mathbf{E}^{-1}\mathbf{B}}^{\mathbf{B}_e}\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t).\end{aligned}\tag{7.7}$$

Gramian matrices. As already known, the explicit representation (7.7) can be used to compute the controllability (\mathbf{W}_c) and observability (\mathbf{W}_o) Gramians as solutions to the following Lyapunov equations

$$\mathbf{A}_e\mathbf{W}_c + \mathbf{W}_c\mathbf{A}_e^T + \mathbf{B}_e\mathbf{B}_e^T = \mathbf{0},\tag{7.8a}$$

$$\mathbf{A}_e^T\mathbf{W}_o + \mathbf{W}_o\mathbf{A}_e + \mathbf{C}^T\mathbf{C} = \mathbf{0}.\tag{7.8b}$$

Inserting the relations from (7.7) delivers the following controllability Lyapunov equation

$$\begin{aligned}\mathbf{A}_e\mathbf{W}_c + \mathbf{W}_c\mathbf{A}_e^T + \mathbf{B}_e\mathbf{B}_e^T &= \mathbf{0} \\ \Rightarrow \mathbf{E}^{-1}\mathbf{A}\mathbf{W}_c + \mathbf{W}_c\mathbf{A}^T\mathbf{E}^{-T} + \mathbf{E}^{-1}\mathbf{B}\mathbf{B}^T\mathbf{E}^{-T} &= \mathbf{0} \\ \Rightarrow \mathbf{A}\mathbf{W}_c\mathbf{E}^T + \mathbf{E}\mathbf{W}_c\mathbf{A}^T + \mathbf{B}\mathbf{B}^T &= \mathbf{0}.\end{aligned}\tag{7.9}$$

Equations of the form (7.9) are also called *generalized* Lyapunov equations, for which there are also adapted algorithms. The computation of the observability Gramian with the implicit state-space representation (7.6) requires an additional intermediate step:

$$\begin{aligned}\mathbf{A}_e^T\mathbf{W}_o + \mathbf{W}_o\mathbf{A}_e + \mathbf{C}^T\mathbf{C} &= \mathbf{0} \\ \Rightarrow \mathbf{A}^T\mathbf{E}^{-T}\mathbf{W}_o + \mathbf{W}_o\mathbf{E}^{-1}\mathbf{A} + \mathbf{C}^T\mathbf{C} &= \mathbf{0} \\ \Rightarrow \mathbf{A}^T\widetilde{\mathbf{W}}_o\mathbf{E} + \mathbf{E}^T\widetilde{\mathbf{W}}_o\mathbf{A} + \mathbf{C}^T\mathbf{C} &= \mathbf{0}.\end{aligned}\tag{7.10}$$

where the auxiliary variable $\widetilde{\mathbf{W}}_o$ was introduced as the solution of (7.10) according to the relationship

$$\mathbf{W}_o = \mathbf{E}^T\widetilde{\mathbf{W}}_o\mathbf{E}.\tag{7.11}$$

As already discussed, the “square root” variant of Balanced Truncation is based on the calculation of the *Cholesky* factors of the Gramian matrices. Since Cholesky factors for the solutions of the generalized Lyapunov equations (7.9) and (7.10) can be computed directly, it makes sense to specify the Cholesky decomposition for \mathbf{W}_c and $\widetilde{\mathbf{W}}_o$:

$$\mathbf{W}_c = \mathbf{S}\mathbf{S}^T,\tag{7.12a}$$

$$\widetilde{\mathbf{W}}_o = \widetilde{\mathbf{R}}\widetilde{\mathbf{R}}^T.\tag{7.12b}$$

Balanced representation. By a suitable state transformation

$$\mathbf{x}_b = \mathbf{T}_b \mathbf{x} \quad (7.13)$$

(7.6) can be transformed into the balanced representation

$$\begin{aligned} \overbrace{\mathbf{T}_b \mathbf{E} \mathbf{T}_b^{-1}}^{\mathbf{E}_b} \dot{\mathbf{x}}_b(t) &= \overbrace{\mathbf{T}_b \mathbf{A} \mathbf{T}_b^{-1}}^{\mathbf{A}_b} \mathbf{x}_b(t) + \overbrace{\mathbf{T}_b \mathbf{B}}^{\mathbf{B}_b} \mathbf{u}(t), \\ \mathbf{y}(t) &= \underbrace{\mathbf{C} \mathbf{T}_b^{-1}}_{\mathbf{C}_b} \mathbf{x}_b(t). \end{aligned} \quad (7.14)$$

Analogous to the explicit case, the transformed Gramians in balanced representation are

$$\mathbf{W}_{c,b} = \mathbf{T}_b \mathbf{W}_c \mathbf{T}_b^T, \quad (7.15a)$$

$$\mathbf{W}_{o,b} = \mathbf{T}_b^{-T} \mathbf{W}_o \mathbf{T}_b^{-1} = \mathbf{T}_b^{-T} \mathbf{E}^T \widetilde{\mathbf{W}}_o \mathbf{E} \mathbf{T}_b^{-1}. \quad (7.15b)$$

The transformation matrix \mathbf{T}_b can be derived from the claim $\mathbf{W}_{c,b} = \mathbf{W}_{o,b} = \mathbf{\Sigma}$ with $\mathbf{\Sigma} = \text{diag}(\sigma_1, \dots, \sigma_n)$ and $\sigma_i = \sqrt{\lambda_i(\mathbf{W}_c \mathbf{W}_o)}$.

Theorem 10. *The state transformation \mathbf{T}_b into balanced representation can be calculated as follows:*

1. *Solution of the generalized Lyapunov equations (7.9) and (7.10) to find the Cholesky factors $\mathbf{W}_c = \mathbf{S} \mathbf{S}^T$ and $\widetilde{\mathbf{W}}_o = \widetilde{\mathbf{R}} \widetilde{\mathbf{R}}^T$.*
2. *SVD of $\widetilde{\mathbf{R}}^T \mathbf{E} \mathbf{S} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T$.*

Then, \mathbf{T}_b and \mathbf{T}_b^{-1} result in

$$\mathbf{T}_b := \mathbf{\Sigma}^{-1/2} \mathbf{U}^T \widetilde{\mathbf{R}}^T \mathbf{E}, \quad (7.16a)$$

$$\mathbf{T}_b^{-1} := \mathbf{S} \mathbf{V} \mathbf{\Sigma}^{-1/2}. \quad (7.16b)$$

Proof.

$$\begin{aligned} \mathbf{W}_{c,b} &= \mathbf{T}_b \mathbf{W}_c \mathbf{T}_b^T \quad \text{with} \quad \mathbf{W}_c = \mathbf{S} \mathbf{S}^T \\ &= \mathbf{\Sigma}^{-1/2} \mathbf{U}^T \widetilde{\mathbf{R}}^T \mathbf{E} (\mathbf{S} \mathbf{S}^T) \mathbf{E}^T \widetilde{\mathbf{R}} \mathbf{U} \mathbf{\Sigma}^{-1/2} \\ &= \mathbf{\Sigma}^{-1/2} \underbrace{\mathbf{U}^T \mathbf{U}}_{=\mathbf{I}} \mathbf{\Sigma} \underbrace{\mathbf{V}^T \mathbf{V}}_{=\mathbf{I}} \mathbf{\Sigma} \underbrace{\mathbf{U}^T \mathbf{U}}_{=\mathbf{I}} \mathbf{\Sigma}^{-1/2} \\ &= \mathbf{\Sigma}^{-1/2} \mathbf{\Sigma}^2 \mathbf{\Sigma}^{-1/2} = \mathbf{\Sigma}. \end{aligned}$$

The proof for $\mathbf{W}_{o,b} = \mathbf{\Sigma}$ is given analogously using (7.15b) and (7.12b). \square

Remark 1. One can see that the equations (7.16) boil down for $\mathbf{E} = \mathbf{I}$ to those given in Theorem 6.

Remark 2. Note that in the literature sometimes the expression

$$\mathbf{T}_b := \Sigma^{-1/2} \mathbf{U}^T \mathbf{R} \quad (7.17)$$

is found. Hereby, it should be noticed that \mathbf{R} must result from the Cholesky decomposition $\mathbf{W}_o = \mathbf{R}\mathbf{R}^T$ and – due to (7.11) – $\mathbf{R} = \mathbf{E}^T \widetilde{\mathbf{R}}$ holds. As long as this relationship is respected, equations (7.16a) and (7.17) coincide.

Similarly, slight changes in the expressions may occur if the Cholesky factors are defined as $\mathbf{W}_c = \mathbf{S}^T \mathbf{S}$ and $\mathbf{W}_o = \widetilde{\mathbf{R}}^T \widetilde{\mathbf{R}}$, or the SVD of $\mathbf{S}^T \mathbf{E}^T \widetilde{\mathbf{R}}$ is performed.

Balanced Truncation. Different from the explicit case $\mathbf{E} = \mathbf{I}$, the projection matrices in case $\mathbf{E} \neq \mathbf{I}$ do *not* result from the truncation in the balanced representation. To obtain a stable, balanced representation for the reduced model, the projection matrices must be defined as follows

$$\mathbf{V}_{\text{BT}} := \mathbf{S} \mathbf{V}_q \Sigma_q^{-1/2}, \quad (7.18a)$$

$$\mathbf{W}_{\text{BT}}^T := \Sigma_q^{-1/2} \mathbf{U}_q^T \widetilde{\mathbf{R}}^T. \quad (7.18b)$$

In particular, (7.18b), as opposed to (7.16a), does not contain an \mathbf{E} matrix. It follows

$$\mathbf{E}_r = \mathbf{W}_{\text{BT}}^T \mathbf{E} \mathbf{V}_{\text{BT}} = \Sigma_q^{-1/2} \mathbf{U}_q^T \left(\widetilde{\mathbf{R}}^T \mathbf{E} \mathbf{S} \right) \mathbf{V}_q \Sigma_q^{-1/2} = \mathbf{I}_q, \quad (7.19)$$

i.e. \mathbf{V}_{BT} and \mathbf{W}_{BT} are biorthogonal w.r.t. \mathbf{E} .

For the reduced controllability Gramian it follows

$$\begin{aligned} & \mathbf{A} \mathbf{W}_c \mathbf{E}^T + \mathbf{E} \mathbf{W}_c \mathbf{A}^T + \mathbf{B} \mathbf{B}^T &= \mathbf{0}, \\ \implies & \mathbf{W}_{\text{BT}}^T \left(\mathbf{A} \mathbf{W}_c \mathbf{E}^T + \mathbf{E} \mathbf{W}_c \mathbf{A}^T + \mathbf{B} \mathbf{B}^T \right) \mathbf{W}_{\text{BT}} &= \mathbf{0}, \\ \iff & \mathbf{W}_{\text{BT}}^T \mathbf{A} \left(\mathbf{S} \mathbf{S}^T \right) \mathbf{E}^T \widetilde{\mathbf{R}} \mathbf{U}_q \Sigma_q^{-1/2} + \Sigma_q^{-1/2} \mathbf{U}_q^T \widetilde{\mathbf{R}}^T \mathbf{E} \left(\mathbf{S} \mathbf{S}^T \right) \mathbf{A}^T \mathbf{W}_{\text{BT}} + \mathbf{B}_r \mathbf{B}_r^T &= \mathbf{0}, \\ \iff & \mathbf{W}_{\text{BT}}^T \mathbf{A} \left(\mathbf{S} \mathbf{V}_q \Sigma_q^{-1/2} \right) \Sigma_q + \Sigma_q \left(\Sigma_q^{-1/2} \mathbf{V}_q^T \mathbf{S}^T \right) \mathbf{A}^T \mathbf{W}_{\text{BT}} + \mathbf{B}_r \mathbf{B}_r^T &= \mathbf{0}, \\ \iff & \mathbf{A}_r \Sigma_q + \Sigma_q \mathbf{A}_r^T + \mathbf{B}_r \mathbf{B}_r^T &= \mathbf{0}. \end{aligned} \quad (7.20)$$

Similarly, the following applies to the observability Gramian

$$\begin{aligned} & \mathbf{A}^T \widetilde{\mathbf{W}}_o \mathbf{E} + \mathbf{E}^T \widetilde{\mathbf{W}}_o \mathbf{A} + \mathbf{C}^T \mathbf{C} &= \mathbf{0}, \\ \implies & \mathbf{V}_{\text{BT}}^T \left(\mathbf{A}^T \widetilde{\mathbf{W}}_o \mathbf{E} + \mathbf{E}^T \widetilde{\mathbf{W}}_o \mathbf{A} + \mathbf{C}^T \mathbf{C} \right) \mathbf{V}_{\text{BT}} &= \mathbf{0}, \\ \iff & \mathbf{V}_{\text{BT}}^T \mathbf{A}^T \left(\widetilde{\mathbf{R}} \widetilde{\mathbf{R}}^T \right) \mathbf{E} \mathbf{S} \mathbf{V}_q \Sigma_q^{-1/2} + \Sigma_q^{-1/2} \mathbf{V}_q^T \mathbf{S}^T \mathbf{E}^T \left(\widetilde{\mathbf{R}} \widetilde{\mathbf{R}}^T \right) \mathbf{A} \mathbf{V}_{\text{BT}} + \mathbf{C}_r^T \mathbf{C}_r &= \mathbf{0}, \\ \iff & \mathbf{V}_{\text{BT}}^T \mathbf{A}^T \left(\widetilde{\mathbf{R}} \mathbf{U}_q \Sigma_q^{-1/2} \right) \Sigma_q + \Sigma_q \left(\Sigma_q^{-1/2} \mathbf{U}_q^T \widetilde{\mathbf{R}}^T \right) \mathbf{A} \mathbf{V}_{\text{BT}} + \mathbf{C}_r^T \mathbf{C}_r &= \mathbf{0}, \\ \iff & \mathbf{A}_r^T \Sigma_q + \Sigma_q \mathbf{A}_r + \mathbf{C}_r^T \mathbf{C}_r &= \mathbf{0}. \end{aligned} \quad (7.21)$$

Thus, it is proven that (7.18) balances the reduced order model and that it only contains the state directions that belong to the largest q Hankel singular values.

References

Mathematical Fundamentals

- [ND77] B. Noble and J. W. Daniel. *Applied linear algebra*. 2nd ed. Prentice-Hall New Jersey, 1977.
- [GV96] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, 1996.
- [Dem97] J. W. Demmel. *Applied numerical linear algebra*. SIAM, 1997.
- [HJ12] R. A. Horn and C. R. Johnson. *Matrix analysis*. 2nd ed. Cambridge, 2012.

Numerical linear algebra / Linear matrix equations

- [BS72] R. Bartels and G. Stewart. “Solution of the matrix equation $AX+XB= C$ ”. In: *Comm. ACM* 15.9 (1972), pp. 820–826.
- [Ham82] S. Hammarling. “Numerical solution of the stable, non-negative definite Lyapunov equation”. In: *IMA Journal of Numerical Analysis* 2.3 (1982), pp. 303–323. DOI: 10.1093/imanum/2.3.303.
- [Pen00] T. Penzl. “A cyclic low-rank Smith method for large sparse Lyapunov equations”. In: *SIAM Journal on Scientific Computing* 21.4 (2000), pp. 1401–1418.
- [Sim16] V. Simoncini. “Computational methods for linear matrix equations”. In: *SIAM Review* 58.3 (2016), pp. 377–441.

Fundamentals in Systems and Control Theory

- [Kai80] T. Kailath. *Linear systems*. Vol. 156. Prentice-Hall Englewood Cliffs, NJ, 1980.
- [Rug96] W. J. Rugh. *Linear system theory*. Vol. 2. Prentice Hall Upper Saddle River, NJ, 1996.
- [ZDG+96] K. Zhou, J. C. Doyle, K. Glover, et al. *Robust and optimal control*. Vol. 40. Prentice Hall New Jersey, 1996.
- [Föll13] O. Föllinger. *Regelungstechnik*. 11th ed. VDE Verlag, 2013.

Fundamentals in MOR

- [Ant05] A. C. Antoulas. *Approximation of Large-Scale Dynamical Systems*. SIAM, 2005. URL: <https://doi.org/10.1137/1.9780898718713>.
- [Ben+17] P. Benner et al. *Model reduction and approximation: theory and algorithms*. Vol. 15. SIAM, 2017. URL: <https://doi.org/10.1137/1.9781611974829>.

LTI Systems

- [ML67] L. Meier and D. Luenberger. “Approximation of linear constant systems”. In: *IEEE Transactions on Automatic Control* 12.5 (1967), pp. 585–588.
- [Gri97] E. J. Grimme. “Krylov Projection Methods for Model Reduction”. PhD thesis. Dep. of Electrical Eng., Uni. Illinois at Urbana Champaign, 1997.
- [GVV04] K. A. Gallivan, A. Vandendorpe, and P. Van Dooren. “Model reduction of MIMO systems via tangential interpolation”. In: *SIAM Journal on Matrix Analysis and Applications* 26.2 (2004), pp. 328–349.
- [GAB08] S. Gugercin, A. C. Antoulas, and C. Beattie. “ \mathcal{H}_2 model reduction for large-scale linear dynamical systems”. In: *SIAM Journal on Matrix Analysis and Applications* 30.2 (2008), pp. 609–638.
- [DS11] V. Druskin and V. Simoncini. “Adaptive rational Krylov subspaces for large-scale dynamical systems”. In: *Systems & Control Letters* 60.8 (2011), pp. 546–560.
- [Pan+13] H. K. F. Panzer et al. “A greedy rational Krylov method for \mathcal{H}_2 -pseudooptimal model order reduction with preservation of stability”. In: *American Control Conference*. 2013, pp. 5532–5537.
- [WPL13] T. Wolf, H. K. Panzer, and B. Lohmann. “ \mathcal{H}_2 pseudo-optimality in model order reduction by Krylov subspace methods”. In: *2013 European Control Conference (ECC)*. IEEE. 2013, pp. 3427–3432.
- [BG14] C. A. Beattie and S. Gugercin. “Model reduction by rational interpolation”. In: *Model reduction and approximation: theory and applications*, P. Benner, A. Cohen, M. Ohlberger, and K. Willcox, eds., SIAM (2014), pp. 297–334.
- [Pan14] H. K. Panzer. “Model order reduction by Krylov subspace methods with global error bounds and automatic choice of parameters”. PhD thesis. Technische Universität München, 2014. URL: <http://mediatum.ub.tum.de?id=1207822>.
- [CPL16] A. Castagnotto, H. K. Panzer, and B. Lohmann. “Fast \mathcal{H}_2 -optimal model order reduction exploiting the local nature of Krylov-subspace methods”. In: *2016 European Control Conference (ECC)*. IEEE. 2016, pp. 1958–1969.
- [Cas+17] A. Castagnotto et al. “sss & sssMOR: Analysis and Reduction of Large-Scale Dynamic Systems in MATLAB”. In: *at-Automatisierungstechnik* 65.2 (2017), pp. 134–150. DOI: 10.1515/auto-2016-0137.

Descriptor Systems (DAEs)

- [KM06] P. Kunkel and V. Mehrmann. *Differential-algebraic equations*. EMS textbooks in mathematics. European Math. Soc., Zürich, 2006. DOI: 10.4171/017.
- [GSW13] S. Gugercin, T. Stykel, and S. Wyatt. “Model reduction of descriptor systems by interpolatory projection methods”. In: *SIAM J. Sci. Comput.* 35.5 (2013), B1010–B1033. DOI: 10.1137/130906635.
- [BS17] P. Benner and T. Stykel. “Model Order Reduction for Differential-Algebraic Equations: A Survey”. In: *Surveys in Differential-Algebraic Equations IV*. Ed. by A. Ilchmann and T. Reis. Springer, 2017, pp. 107–160. URL: https://doi.org/10.1007/978-3-319-46618-7_3.

Second-Order Systems

- [Ray45] J. W. S. B. Rayleigh. *The theory of sound*. Vol. 2. Dover, 1945.
- [CB68] R. Craig and M. Bampton. “Coupling of substructures for dynamic analyses”. In: *AIAA Journal* 6.7 (1968), pp. 1313–1319.
- [BS05] Z. Bai and Y. Su. “Dimension reduction of large-scale second-order dynamical systems via a second-order Arnoldi method”. In: *SIAM Journal on Scientific Computing* 26.5 (2005), pp. 1692–1709.
- [SL06] B. Salimbahrami and B. Lohmann. “Order reduction of large scale second-order systems using Krylov subspace methods”. In: *Linear Algebra and its Applications* 415.2-3 (2006), pp. 385–405.
- [RS08] T. Reis and T. Stykel. “Balanced truncation model reduction of second-order systems”. In: *Mathematical and Computer Modelling of Dynamical Systems* 14.5 (2008), pp. 391–406.
- [GR14] M. Géradin and D. J. Rixen. *Mechanical vibrations: theory and application to structural dynamics*. John Wiley & Sons, 2014.

Port-Hamiltonian Systems

- [PV10] R. V. Polyuga and A. Van der Schaft. “Structure preserving model reduction of port-Hamiltonian systems by moment matching at infinity”. In: *Automatica* 46.4 (2010), pp. 665–672.
- [Wol+10] T. Wolf et al. “Passivity and structure preserving order reduction of linear port-Hamiltonian systems using Krylov subspaces”. In: *European Journal of Control* 16.4 (2010), pp. 401–406.
- [Gug+12] S. Gugercin et al. “Structure-preserving tangential interpolation for model reduction of port-Hamiltonian systems”. In: *Automatica* 48.9 (2012), pp. 1963–1974.

Parametric MOR

- [AF08] D. Amsallem and C. Farhat. “Interpolation method for adapting reduced-order models and application to aeroelasticity”. In: *AIAA Journal* 46.7 (2008), pp. 1803–1813.
- [Pan+10] H. K. F. Panzer et al. “Parametric model order reduction by matrix interpolation”. In: *at-Automatisierungstechnik* 58.8 (Aug. 2010), pp. 475–484. DOI: 10.1524/auto.2010.0863.
- [BGW15] P. Benner, S. Gugercin, and K. Willcox. “A survey of projection-based model reduction methods for parametric dynamical systems”. In: *SIAM Review* 57.4 (2015), pp. 483–531. URL: <https://doi.org/10.1137/130932715>.

MOR for Nonlinear Systems

- [Lia+02] Y. Liang et al. “Proper orthogonal decomposition and its applications—Part I: Theory”. In: *Journal of Sound and vibration* 252.3 (2002), pp. 527–544.
- [CS10] S. Chaturantabut and D. C. Sorensen. “Nonlinear model reduction via discrete empirical interpolation”. In: *SIAM Journal on Scientific Computing* 32.5 (2010), pp. 2737–2764.
- [Far+14] C. Farhat et al. “Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency”. In: *International Journal for Numerical Methods in Engineering* 98.9 (2014), pp. 625–662.
- [Goy18] P. K. Goyal. “System-Theoretic Model Order Reduction for Bilinear and Quadratic-Bilinear Control Systems”. PhD thesis. Otto-von-Guericke-Universität Magdeburg, 2018.
- [Rut18] J. Rutzmoser. “Model Order Reduction for Nonlinear Structural Dynamics: Simulation-free Approaches”. PhD thesis. Technische Universität München, 2018. URL: <http://mediatum.ub.tum.de?id=1381943>.
- [CSL19] M. Cruz Varona, N. Schneucker, and B. Lohmann. “Nonlinear Moment Matching for the Simulation-Free Reduction of Structural Systems”. In: *arXiv preprint* (2019). URL: <https://arxiv.org/abs/1903.12303>.
- [Cru+] M. Cruz Varona et al. “Simulation-Free Reduction of Dynamic Systems by Approximated Nonlinear Moment Matching”. In preparation (preliminary version: <https://arxiv.org/abs/1901.10750>).

Others

- [FN82] K. Fernando and H. Nicholson. “Singular perturbational model reduction in the frequency domain”. In: *IEEE Transactions on Automatic Control* 27.4 (1982), pp. 969–970.