Complexity Reduction in Control

Lars Grüne
Chair of Applied Mathematics
Mathematical Institute
University of Bayreuth
95440 Bayreuth, Germany
lars.gruene@uni-bayreuth.de
https://num.math.uni-bayreuth.de/en/team/lars-gruene/

Lecture Notes

Summer Semester 2021

Preface

These Lecture Notes were written to accompany a Master Course in Applied Mathematics that I gave in the Summer Semester 2021 at the University of Bayreuth, Germany. I would like to thank all the students of the course for their valuable feedback, which considerably helped to improve these notes. My sincere thanks go to Peter Benner and Heike Faßbender for providing me with a preliminary version of their textbook *Modellreduktion. Eine systemtheoretisch-orientierte Einführung* [1], which was extremely helpful for writing some of the chapters of this book, particularly Chapter 4.

Since the course was given in a virtual format, all lectures are recorded and I'd be happy to provide links to the respective recordings upon request.

Bayreuth, July 2021 LARS GRÜNE

Contents

Chapter 1

Introduction

In this lecture we will mainly be concerned with linear control systems, either in continuous time

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{1.1}$$

or in discrete time

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k). \end{aligned} \tag{1.2}$$

Here $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{l \times n}$, $D \in \mathbb{R}^{l \times m}$, $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $y \in \mathbb{R}^l$.

We will consider three key questions:

- If we want to describe the behaviour of a real world system, what is the best option to choose (A, B, C, D)? Particularly, we will be looking at minimal models in terms of system dimension, as these are the simplest models to use in terms of computational effort.

  Realisation theory will answer this question, which we will address in Chapter 2.

- If the exact description is still too large to be handled for a particular purpose (e.g., for optimal control), how can we obtain a smaller and thus less complex, model that behaves "almost" like the large model?

  For this purpose we will introduce the method of "balanced truncation" in Chapter 4.

- The derivation of a model for a real process is a very complex task in itself. In the last part of this lecture we will derive a method in which this is done automatically from measured data. We will look at this question in the context of Model Predictive Control (MPC).

Chapter 2

Realisation Theory

August 9, 2021

In this chapter we will investigate the question how a minimal model of the form (1.1) realising a certain behaviour can be characterised.

2.1 Basic definitions

To this end, we consider the behaviour of the control system for initial value $x_0 = 0$. Clearly, with this choice we can only obtain very particular trajectories; more precisely the trajectories

$$x(t;0,u) = \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau.$$

Nevertheless, we will see at the end of this chapter that the minimal models that result from considering these trajectories are very meaningful.

Once the initial value is fixed, we obtain a map from u to y, defined by

$$y(t) = \int_0^t Ce^{A(t-\tau)}Bu(\tau)d\tau + Du(t) = g \star u(t) + Du(t),$$

with $g(t) = Ce^{At}B$ and "$\star$" denoting the convolution $g \star h(t) = \int_0^t g(t-\tau)h(\tau)d\tau$.

Passing to the Laplace transform, we can write this relation as

$$\hat{y} = G\hat{u}$$

with $G(s) = \hat{g}(s) = C(s\mathrm{Id} - A)^{-1}B + D$.

We can now define the meaning of a minimal realisation.

Definition 2.1 Let $G(s) = \widetilde{C}(s\mathrm{Id} - \widetilde{A})^{-1}\widetilde{B} + D$ be a transfer function.

(i) A control system of the form (1.1) defined by (A,B,C,D) with appropriate dimensions is called a *realisation* of G, if $G(s) = C(s\mathrm{Id} - A)^{-1}B$.

(ii) The *dimension* n of a realisation is the dimension of the state x, i.e., if the dimension of a realisation is $n \in \mathbb{N}$ then $A \in \mathbb{R}^{n\times n}$.

(iii) A realisation (1.1) of G with dimension $n \in \mathbb{N}$ is called a *minimal realisation* of G, if any other realisation of G has a dimension $n' \geq n$. □

Example 2.2 We consider the linear inverted pendulum, given by

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ g & -k & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \tag{2.1}$$

with constants $g, k > 0$. Here x_1 represents the linearised angle ϕ of the pendulum, which increases in counterclockwise direction, where $x_1 = 0$ corresponds to the upright pendulum. x_2 is the angular velocity, x_3 the position of the cart and x_4 its velocity. The constant k is a measure for the friction in the model (the larger k the more friction) and $g \approx 9.81 m/s^2$ is the gravitational constant.

As output we consider two different options: on the one hand we consider $y = (y_1, y_2)^T$ with y_1 the position of the pendulum and y_2 the position of the cart, resulting in

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

On the other hand, we consider only $y = (y_1)$ as the position of the pendulum, leading to

$$C' = \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}.$$

In both cases, we set $D = 0$. We then have

$$(s\mathrm{Id} - A)^{-1} = \begin{pmatrix} s & -1 & 0 & 0 \\ -g & s+k & 0 & 0 \\ 0 & 0 & s & -1 \\ 0 & 0 & 0 & s \end{pmatrix}^{-1} = \begin{pmatrix} \frac{s+k}{ks+s^2-g} & \frac{1}{ks+s^2-g} & 0 & 0 \\ \frac{g}{ks+s^2-g} & \frac{s}{ks+s^2-g} & 0 & 0 \\ 0 & 0 & \frac{1}{s} & \frac{1}{s^2} \\ 0 & 0 & 0 & \frac{1}{s} \end{pmatrix},$$

and thus

$$(s\mathrm{Id} - A)^{-1} B = \begin{pmatrix} \frac{1}{ks+s^2-g} \\ \frac{s}{ks+s^2-g} \\ \frac{1}{s^2} \\ \frac{1}{s} \end{pmatrix}.$$

This implies that the transfer functions read

$$G(s) = \begin{pmatrix} \frac{1}{ks+s^2-g} \\ \frac{1}{s^2} \end{pmatrix} \quad \text{and} \quad G'(s) = \begin{pmatrix} \frac{1}{ks+s^2-g} \end{pmatrix}.$$

□

2.2 Characterisation of the minimal realisation

Theorem 2.3 Let $G(s) = \widetilde{C}(s\mathrm{Id} - \widetilde{A})^{-1}\widetilde{B} + D$ be a transfer function, which is not constant in s. Then a realisation (A, B, C, D) is minimal if and only if (A, B) is controllable and (A, C) is observable.

Before we prove the theorem, we prove the following auxiliary lemma.

Lemma 2.4 Let $A \in \mathbb{R}^{n\times n}$. If $s \in \mathbb{C}$ is so large that $|s^{-1}\lambda| < 1$ for all eigenvalues λ of A, then

$$(s\mathrm{Id} - A)^{-1} = \sum_{i=1}^{\infty} s^{-i}A^{i-1}.$$

Proof: If all eigenvalues of a matrix Z have modulus less than 1, then the inverse of $\mathrm{Id} - Z$ can be written as a Neumann series

$$(\mathrm{Id} - Z)^{-1} = \sum_{i=0}^{\infty} Z^i,$$

see, e.g., [5, Satz II.1.11]. Now let s be as in the assumption, then this applies to $Z = s^{-1}A$. Hence

$$(s\mathrm{Id} - A)^{-1} = [s(\mathrm{Id} - s^{-1}A)]^{-1} = s^{-1}(\mathrm{Id} - s^{-1}A)^{-1} = s^{-1}\sum_{i=0}^{\infty} s^{-i}A^i = \sum_{i=0}^{\infty} s^{-i-1}A^i,$$

which yields the assertion after renumbering of the index. □

Proof of Theorem 2.3: "⇒": We show this implication by contraposition, i.e., we show that when (A,B,C,D) is a realisation of G for which (A,B) is not controllable or (A,C) is not observable, then (A,B,C,D) cannot be a minimal realisation.

Consider first the case that (A,B) is not controllable. We may exclude the case $B = 0$, since in this case $G \equiv D$ is constant in s. Then Lemma 2.14 from [4] implies that there exist $r \in \{1,\dots,n-1\}$ and an invertible matrix $T \in \mathbb{R}^{n\times n}$, such that

$$T^{-1}AT = \begin{pmatrix} A_1 & A_2 \\ 0 & A_3 \end{pmatrix}, \quad T^{-1}B = \begin{pmatrix} B_1 \\ 0 \end{pmatrix}$$

with $A_1 \in \mathbb{R}^{r\times r}$, $A_2 \in \mathbb{R}^{r\times(n-r)}$, $A_3 \in \mathbb{R}^{(n-r)\times(n-r)}$, $B_1 \in \mathbb{R}^{r\times m}$ and the pair (A_1, B_1) is controllable. We also write $CT = (C_1\, C_2)$ with $C_1 \in \mathbb{R}^{l\times r}$ and $C_2 \in \mathbb{R}^{l\times(n-r)}$.

We note that since $s\mathrm{Id} - T^{-1}AT$ has block triangular structure, we obtain

$$(s\mathrm{Id} - T^{-1}AT)^{-1} = \begin{pmatrix} s\mathrm{Id} - A_1 & -A_2 \\ 0 & s\mathrm{Id} - A_3 \end{pmatrix}^{-1} = \begin{pmatrix} (s\mathrm{Id} - A_1)^{-1} & * \\ 0 & (s\mathrm{Id} - A_3)^{-1} \end{pmatrix}.$$

This implies

$$\begin{aligned}
G(s) &= C(s\mathrm{Id} - A)^{-1}B + D \\
&= CT(s\mathrm{Id} - T^{-1}AT)^{-1}TB + D \\
&= (C_1\, C_2)\left(s\mathrm{Id} - \begin{pmatrix} A_1 & A_2 \\ 0 & A_3 \end{pmatrix}\right)^{-1}\begin{pmatrix} B_1 \\ 0 \end{pmatrix} + D \\
&= (C_1\, C_2)\begin{pmatrix} (s\mathrm{Id} - A_1)^{-1} & * \\ 0 & (s\mathrm{Id} - A_3)^{-1} \end{pmatrix}\begin{pmatrix} B_1 \\ 0 \end{pmatrix} + D \\
&= C_1(s\mathrm{Id} - A_1)^{-1}B_1 + D.
\end{aligned}$$

This shows that (A_1, B_1, C_1, D) is a realisation of G with dimension $r < n$ and thus (A,B,C,D) cannot be a minimal realisation. In case (A,C) is not observable, the proof proceeds similarly, using the duality between controllability and observability.

"$\Leftarrow$": Consider now a realisation (A, B, C, D) with dimension n, for which (A, B) is controllable and (A, C) is observable. Then the matrices

$$R = (B\ AB\ A^2B\ \dots\ A^{n-1}B)$$

and

$$O = \begin{pmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{pmatrix}$$

have full rank n. Consider a second realisation (A', B', C', D') of G with dimension $n' \leq n$. In order to prove minimality of (A, B, C, D), we have to show that $n' \geq n$.

To this end, define

$$R' = (B'\ A'B'\ (A')^2B'\ \dots\ (A')^{n'-1}B')$$

and

$$O' = \begin{pmatrix} C' \\ C'A' \\ C'(A')^2 \\ \vdots \\ C'(A')^{n'-1} \end{pmatrix}.$$

Clearly, due to their dimension, the rank of R' and O' is at most n', hence the rank of the product matrix $O'R'$ satisfies $\text{rank} O'R' \leq n'$. We will now show that $OR = O'R'$, which implies $\text{rank} O'R' = \text{rank} OR = n$, and thus $n' \geq n$.

Explicit computation yields that the matrices OR and $O'R'$ consist of blocks of the form CA^jB and $C'(A')^kB'$ with $j = 0, \dots, 2n-2$ and $k = 0, \dots, 2n'-2$. Using Lemma 2.4 and the fact that both (A, B, C, D) and (A', B', C', D') realise G, for all sufficiently large s we obtain

$$G(s) = C(s\text{Id} - A)^{-1}B + D = C\sum_{i=1}^{\infty} s^{-i}A^{i-1}B + D = D + \sum_{i=1}^{\infty} CA^{i-1}Bs^{-i}$$

and

$$G(s) = C'(s\text{Id} - A')^{-1}B' + D' = D' + \sum_{i=1}^{\infty} C'(A')^{i-1}B's^{-i}.$$

Defining $H(s) := G(1/s)$, one thus computes

$$H^{(p)}(s) = \frac{d^p}{ds^p}H(s) = \sum_{i=p}^{\infty} q(i)CA^{i-1}Bs^{i-p}$$

with $q(i) \in \mathbb{N} \setminus 0$. For $p \geq 1$ this implies

$$\lim_{s\to 0} H^{(p)}(s) = q(p)CA^{p-1}B.$$

The same computation carried out for the representation of G via (A', B', C', D') yields

$$\lim_{s\to 0} H^{(p)}(s) = q(p)C'(A')^{p-1}B'.$$

This implies

$$CA^{p-1}B = C'(A')^{p-1}B'$$

for all $p \geq 1$ and thus the desired identity $OR = O'R'$. □

Example 2.5 (Continuation of Example 2.2) For the linearised inverted pendulum one computes that (A, B) is controllable since

$$R = (B\,AB\,A^2B\,A^3B) = \begin{pmatrix} 0 & 1 & -k & g+k^2 \\ 1 & -k & g+k^2 & -2gk-k^3 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

has full rank 4. The pair (A, C) is also observable, since

$$O = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

also has full rank 4. Hence, (A, B, C, D) is a minimal representation of the transfer function G.

However, the pair (A, C') is not observable, since

$$O' = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ g & -k & 0 & 0 \\ -gk & k^2 & 0 & 0 \end{pmatrix}$$

does not have full rank 4 but only rank 2. Hence, (A, B, C', D) is not a minimal representation of the transfer function G'. In this case it is easy to find the minimal representation by hand, because it suffices to omit the cart coordinates x_3 and x_4. The corresponding computations will be done in the exercises. □

2.3 Discussion

Remark 2.6 In this remark we discuss which information we lose when passing from a control system to the minimal realisation of the corresponding transfer function. Obviously, some information is lost and this is due to two facts:

- The transfer function only “sees” the output y, not the state x. Thus, it does not see the unobservable states $x \in \mathcal{N}$.

- The construction of the transfer function only considers trajectories with initial value $x_0 = 0$. Thus, it only considers the trajectories in the reachable set $\mathcal{R}$.

The first fact implies that we need to make sure that all we are truely interested in is also included in the output, either explicitly (as the positions x_1 and x_3 in the pendulum example with output matrix C) or implicitly (due to observability, as the velocities x_2 and x_4 in the pendulum example with output matrix C). If we do not do this (as in the case of the cart position x_3 and velocity x_4 in the pendulum example with output matrix C'), and design our control based on the output, then we will not be able to influence it. Thus, we need to make sure to choose C and thus y appropriately.

The second fact limits the trajectories which contribute to the definition of G (and thus of the minimal realisation) to those in the reachable set $\mathcal{R}$. However, with non-0 initial value other solutions may emerge, which are thus not modelled in the minimal realisation. Generally, we can write each solution $x(t; x_0, u)$ in the form

$$x(t; x_0, u) = x_{\mathcal{R}}(t) + x_V(t),$$

with $x_{\mathcal{R}}(t) \in \mathcal{R}$ and $x_V(t) \in V$, where V is a subspace of $\mathbb{R}^n$ with $\mathcal{R} + V = \mathbb{R}^n$. In the coordinates of Lemma 2.14 from [4], $\mathcal{R} = \langle e_1, \ldots, e_r \rangle$ and $V = \langle e_{r+1}, \ldots, e_n \rangle$. Hence, the control u only influences the $x_{\mathcal{R}}$-part of the solution, while $x_V(\cdot)$ is independent from u and from $x_{\mathcal{R}}$.

If A_3 from this lemma is Hurwitz, i.e., if all its eigenvalues have negative real part, then $x_V(\cdot)$ decays exponentially and will not play a role for the long term behavior of the system. If, however, A_3 is not Hurwitz, then $x_v(t)$ will not tend to 0 or may even grow exponentially, and these solutions are "overlooked" by the minimal realisation. However, as we know from [4], A_3 being not Hurwitz is equivalent to the system being not stabilisable. This means that while in the non-minimal realisation the non-vanishing or even growing solution components are present, there is nothing we can do about is with the control function. Hence, in terms of modelling the way we can influence the system by the control input, no information is lost in the minimal realisation. We only lose information about those parts of the solution that we cannot influence, anyway. Moreover, if the system we start from is stabilizable, then we only lose exponentially decaying parts of the solution when we pass to the minimal realization. □

Example 2.7 It appears reasonable to expect that when the input and output of a system is low dimensional, then the state dimension of the minimal realisation should also be relatively small. This example shows that this expectation is, unfortunately, wrong.

We consider a 1d heat equation

$$v_t(t, z) = v_{zz}(t, z)$$

on $\Omega = (0, 1)$. Here t denotes time and $z \in \Omega$ the spatial variable. The subscript denotes partial derivatives. We use Neumann boundary condition $v_z(t, 0) = 0$ at the left end (which models perfect isolation) and Dirichlet boundary condition $v(t, 1) = u(t)$ on the right, where $u : \mathbb{R} \to \mathbb{R}$ is the control input.

Clearly, this is not a finite dimensional control system in the sense of (1.1), because the evolution of v is governed by a partial differential equation (PDE). However, we can approximate this control system by a system of type (1.1) by means of discretisation in space. Here we use the simplest way of discretising the heat equation by finite differences. To this end, we introduce a grid $0 = z_0 < z_1 < \ldots < z_{n+1} = 1$ with $z_{i+1} - z_i = \Delta z = 1/(n+1)$ for all $i = 0, \ldots, n$, and use the second order difference quotient

$$v_{zz}(t, z_i) = \frac{\partial^2}{\partial z^2} v(t, z_i) \approx \frac{v(t, z_{i-1}) - 2v(t, z_i) + v(t, z_{i+1})}{\Delta z^2}.$$

We can then define approximations $x_i(t) \approx v(t, z_i)$ via the equations

$$\dot{x}_i(t) = \frac{x_{i-1}(t) - 2x_i(t) + x_{i+1}(t)}{\Delta z^2}$$

for $i = 1, \ldots, n$. For $i = 0$ we use the first order difference quotient and the Neumann boundary condition

$$0 = v_z(t, z_0) \approx \frac{v(t, z_1) - v(t, z_0)}{\Delta z}$$

in order to obtain the relation

$$x_0(t) = x_1(t)$$

and in $i = n+1$ we insert the Dirichlet boundary condition $x_{n+1}(t) = u(t)$. The output y is the temperature measured at the left boundary, which coincides with that in the leftmost node x_1.

This leads to the control system

$$\dot{x}(t) = Ax(t) + Bu(t), \qquad y(t) = Cx(t)$$

with

$$A = \frac{1}{\Delta z^2} \begin{pmatrix} -1 & 1 & 0 & \cdots & \cdots & \cdots & 0 \\ 1 & -2 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 1 & -2 & 1 \\ 0 & \cdots & \cdots & \cdots & 0 & 1 & -2 \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \frac{1}{\Delta z^2} \end{pmatrix}, \quad C = (1\,0 \ldots 0).$$

Clearly, since u and y are one-dimensional, the transfer function is one-dimensional, too. Yet, computing the reachability and the observability matrix yields

$$R = \begin{pmatrix} 0 & \cdots & \cdots & 0 & \ast \\ 0 & \cdots & 0 & \ast & \ast \\ \vdots & \iddots & \iddots & \iddots & \vdots \\ 0 & \ast & \ast & \cdots & \ast \\ \ast & \ast & \cdots & \cdots & \ast \end{pmatrix} \quad \text{and} \quad O = \begin{pmatrix} \ast & 0 & \ldots & \ldots & 0 \\ \ast & \ast & 0 & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \ast & \cdots & \ast & \ast & 0 \\ \ast & \cdots & \cdots & \ast & \ast \end{pmatrix},$$

where the elements on the diagonal are non zero for both matrices. This implies that they have full rank n. Hence, by Theorem 2.3 this is a minimal realisation. Thus, while the transfer function is only one-dimensional, the minimal realisation has arbitrary large dimension, depending on the number n of nodes of the discretisation. □

Hence, just by looking at the *exact* minimal realisation we cannot necessarily expect to reduce the complexity. We have to resort to *approximations* in order to be able to obtain models with smaller dimensions.

Chapter 3

Singular Value Decomposition

August 9, 2021

The singular value decomposition (short SVD) will be important for the subsequent considerations for control systems.

3.1 Existence of the SVD

Theorem 3.1 For any matrix $A \in \mathbb{R}^{m \times n}$ there exist two orthogonal matrices $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ (i.e., satisfying $U^T U = \mathrm{Id}$, $V^T V = \mathrm{Id}$) and a matrix $\Sigma \in \mathbb{R}^{m \times n}$, in which only the diagonal elements σ_i can assume values $\neq 0$, such that

$$A = U \Sigma V^T.$$

Moreover, the diagonal entries of Σ are unique (up to permutation), coincide with the square roots of the eigenvalues of $A^T A$ and are called the *singular values* of A.

Proof: Since $A^T A$ is symmetric, there is an orthogonal matrix $V \in \mathbb{R}^{n \times n}$ with

$$V^T A^T A V = \mathrm{diag}(\lambda_1, \dots, \lambda_n).$$

Here we choose V such that the moduli of the λ_j are decreasing, i.e., $|\lambda_{j+1}| \leq |\lambda_j|$. The columns of v_j of V then satisfy $A^T A v_j = \lambda_j v_j$ and thus

$$\lambda_j = \lambda_j v_j^T v_j = v_j^T A^T A v_j = \|A v_j\|_2^2 \geq 0.$$

Hence, if we define $\sigma_i := \sqrt{\lambda_i}$ we obtain real numbers.

Let now $\sigma_1, \dots, \sigma_r \neq 0$ and $\sigma_{r+1}, \dots, \sigma_n = 0$. We define $u_i = \sigma_i^{-1} A v_i \in \mathbb{R}^m$ for $i = 1, \dots, r$. Then

$$u_i^T u_i = \lambda_i^{-1} (A v_i)^T A v_i = \lambda_i^{-1} v_i^T A^T A v_i = v_i^T v_i = 1$$

and

$$u_i^T u_j = \lambda_i^{-1/2} \lambda_j^{-1/2} (A v_i)^T A v_j = \lambda_i^{-1/2} \lambda_j^{-1/2} v_i^T A^T A v_j = \lambda_i^{-1/2} \lambda_j^{1/2} v_i^T v_j = 0$$

for $i \neq j$. Consequently, the $u_1, \dots, u_r$ can be augmented to form an orthonormal basis $(u_1, \dots, u_m)$ of $\mathbb{R}^m$. Since $\sigma_{r+1}, \dots, \sigma_n = 0$, we obtain $\lambda_{r+1}, \dots, \lambda_n = 0$ and thus $A v_j = 0$ für

$j = r+1, \dots, n$. For $U = (u_1, \dots, u_n) \in \mathbb{R}^{m \times m}$ this implies $U\Sigma = (Av_1, \dots, Av_r, 0, \dots, 0) = (Av_1, \dots, Av_n) = AV$ and hence

$$U\Sigma V^T = AVV^T = A.$$

□

In the following we always order the singular values such that $\sigma_{j+1} \le \sigma_j$ holds for all $j = 1, \dots, n-1$.

The interpretation of the singular values is as follows. Let v_k be the k-th column of V, u_k the k-th column of U, and e_k the k-th unit vector. Then, due to orthogonality of V, we obtain

$$Av_k = U\Sigma \underbrace{V^T v_k}_{=e_k} = U\sigma_k e_k = \sigma_k U e_k = \sigma_k u_k.$$

Hence, $(v_k, \sigma_k u_k)$ are pairs of vectors that are mapped onto each other by A and the corresponding singular value σ_k determines the length of the image. For an arbitrary vector $x = \sum_{k=1}^n \mu_k v_k$, we thus obtain

$$Ax = \sum_{k=1}^n \mu_k \sigma_k u_k. \tag{3.1}$$

Hence, if some of the σ_k are very small compared to others, the corresponding v_k-components of x contribute much less to the image Ax. Note that the coefficients μ_j are easily computed by $\mu_j = v_j^T x$, since

$$v_j^T x = \sum_{k=1}^n v_j^T \mu_k v_k = \mu_j.$$

We note that the proof shows that the singular values of A are exactly the roots of the eigenvalues of $A^T A$, which coincide with those of AA^T. For complex matrices, the same holds true with $\bar{A}^T A$ or $A\bar{A}^T$, respectively.

3.2 Low-rank approximations

This opens a way to approximate large matrices A by matrices A_r of lower rank that yield approximately the same value when multiplied to a vector. To this end, let U_r be the matrix containing the first r columns of U, V_r the matrix containing the first r columns of V, and Σ_r the $r \times r$ diagonal matrix containing the singular values $\sigma_1, \dots, \sigma_r$. Defining

$$A_r := U_r \Sigma_r V_r^T,$$

we then obtain a matrix with $\mathrm{rank} A_r \le r$, which for $x = \sum_{k=1}^n \mu_k v_k$ yields the image

$$A_r x = \sum_{k=1}^r \mu_k \sigma_k u_k, \tag{3.2}$$

i.e., compared to (3.1) the summands $\mu_{r+1}\sigma_{r+1}v_{r+1}, \dots, \mu_n \sigma_n v_n$ are surpressed. If the singular values $\sigma_{r+1}, \dots, \sigma_n$ are small, then the difference between (3.1) and (3.2) should also be small.

The following theorem, known as Schmidt-Mirsky or Eckart-Young theorem, makes this statement precise. We recall the definition of the induced 2-norm for matrices

$$\|A\|_2 = \max_{\|x\|_2=1} \|Ax\|_2 = \max_{\lambda \text{ Eigenvalue of } A^TA} \sqrt{\lambda}.$$

Theorem 3.2 The matrices A and A_r just defined satisfy

$$\|A - A_r\|_2 = \min_{\text{rank}B \le r} \|A - B\|_2 = \sigma_{r+1}.$$

Proof: Since orthogonal matrices do not change the 2-norm, we obtain

$$\|A - B\|_2 = \|U^T(A - B)V\|_2.$$

Since $U^TA_rV = \text{diag}(\sigma_1, \dots, \sigma_r, 0, \dots, 0)$, it holds that

$$U^T(A - A_r)V = \begin{pmatrix} 0 & & & & & \\ & \ddots & & & & \\ & & 0 & & & \\ & & & \sigma_{r+1} & & \\ & & & & \ddots & \\ & & & & & \sigma_n \end{pmatrix}.$$

We thus obtain $\|A - A_r\|_2 = \sigma_{r+1}$.

In order to complete the proof it remains to show that $\|A - B\|_2 \ge \sigma_{r+1}$ for all matrices B with rank$B \le r$. Again, we can use $\|A - B\|_2 = \|U^T(A - B)V\|_2 = \|\Sigma - Z\|_2$ with $Z = U^TBV$. Note that rank$Z \le r$. Hence, it suffices to show that

$$\|\Sigma - Z\|_2 \ge \sigma_{r+1}$$

for all Z with rank$Z \le r$. This rank condition implies that the image of Z^T has dimension less or equal to r. Thus, there is a vector $x = \sum_{k=1}^{r+1} \alpha_k e_k$ with $x \perp \text{im}Z^T$ and $1 = \|x\|_2^2 = \sum_{k=1}^{r+1} \alpha_k^2$. This implies $0 = (Z^Ty)^Tx = y^TZx$ for all $y \in \mathbb{R}^n$, and thus $Zx = 0$. Hence

$$\|(\Sigma - Z)x\|_2^2 = \|\Sigma x\|_2^2 = \sum_{k=1}^{r+1} \sigma_k^2\alpha_k^2 \ge \sum_{k=1}^{r+1} \sigma_{r+1}^2\alpha_k^2 = \sigma_{r+1}^2$$

and thus

$$\max_{\|x\|_2=1} \|(\Sigma - Z)x\|_2 \ge \sigma_{r+1}.$$

□

Clearly, if $r \ll n$, then the matrix product $A_rx = U_r\Sigma_rV_rx$ can be computed much faster than Ax and the amount of memory for storing A_r is also much smaller.

3.3 Examples

Example 3.3 We illustrate the low rank approximation with a simple 2×2 example. Consider the matrix

$$A = \begin{pmatrix} 1 & 2 \\ 2 & 5 \end{pmatrix}.$$

Its SVD reads

$$U = \begin{pmatrix} -0.3827 & -0.9239 \\ -0.9239 & 0.3827 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} 5.8284 & 0 \\ 0 & 0.1716 \end{pmatrix}, \quad V = \begin{pmatrix} -0.3827 & -0.9239 \\ -0.9239 & 0.3827 \end{pmatrix}.$$

The geometric meaning of the singular values can be visualised when plotting the set

$$\{Ax \,|\, \|x\|_2 = 1\},$$

which is shown in Figure 3.1.

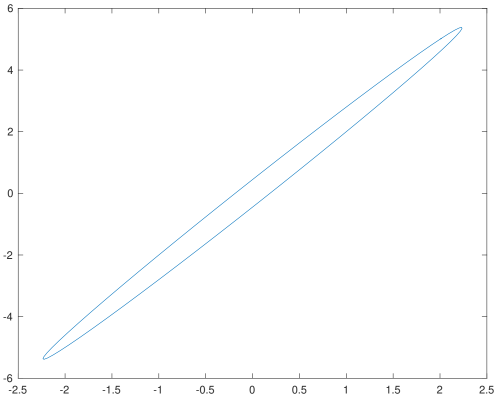

Figure 3.1: $\{Ax \,|\, \|x\|_2 = 1\}$

The figure shows an ellipse whose principal axes correspond to the columns of U. The diameters correspond to the radii: the ellipse has a width of two times 5.8284 from the lower left to the upper right corner but only a width of two times 0.1716 from the lower right to the upper left.

The low rank approximation of A with $r = 1$ evaluates to

$$A_1 = \begin{pmatrix} 0.8536 & 2.0607 \\ 2.0607 & 4.9749 \end{pmatrix}.$$

This matrix has rank 1, meaning that its image is a one-dimensional subspace of $\mathbb{R}^2$. It is exactly the subspace given by the principal axis of the ellipse from Figure 3.1 with the larger diameter. This is confirmed by Figure 3.2.

□

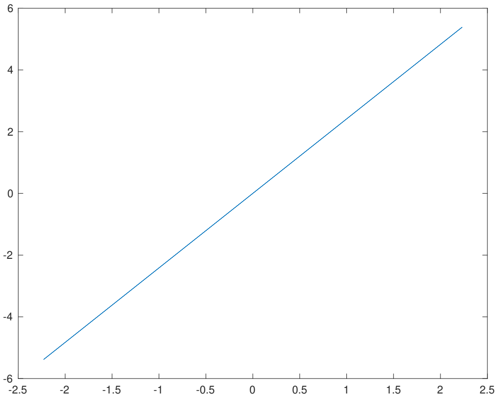

Figure 3.2: $\{A_1 x \mid \|x\|_2 = 1\}$

Example 3.4 A very descriptive application of the SVD is image compression. To this end, we represent a grey-scale image by means of a matrix A, in which each element corresponds to the grey-scale value of a pixel in the image. Figure 3.3 shows a picture of a familar place with 4320×3240 pixels. This amounts to $4320 \cdot 3240 = 13\,996\,800$ pixel values.

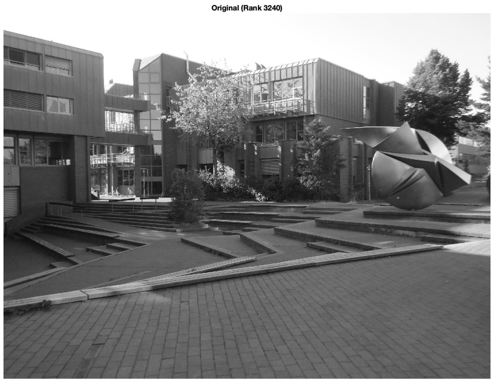

Figure 3.3: Original picture with $4320 \cdot 3240 = 13\,996\,800$ pixels

Figure 3.4 shows the figure corresponding to the low rank matrices A_{200} (left) and A_{100} (right). Storing the respective entries of U_r, V_r and Σ_r requires $200 \cdot (4320 + 3240 + 1) = 1\,512\,200$ and $200 \cdot (4320 + 3240 + 1) = 756\,100$ values, respectively.

It should be noted that in practice the SVD is not used for image compression, because competing algorithms such as jpeg compression are more efficient and provide visually better results. □

The SVD can be computed via the eigenvalues of $A^T A$, but this can be numerically unstable in case A has a large condition number. More sophisticated alrorithms compute the singular value decomposition avoiding the use of $A^T A$, see Section 1.7 in my lecture notes "Vertiefung der Numerischen Mathematik" [3].

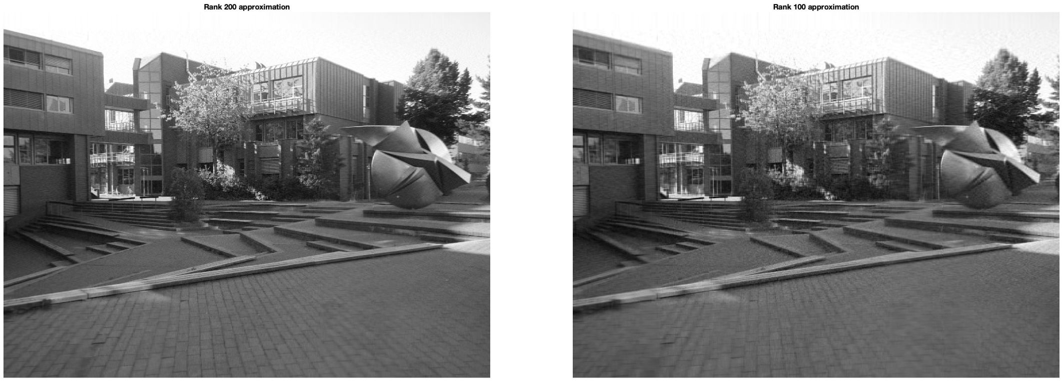

Figure 3.4: Approximated picture with low rank matrix A_r with rank $r = 200$ (left) and $r = 100$ (right), amounting to $200 \cdot (4320 + 3240 + 1) = 1\,512\,200$ and $200 \cdot (4320 + 3240 + 1) = 756\,100$ values, respectively.

Chapter 4

Balanced Truncation

August 9, 2021

The idea of balanced truncation is motivated by the fact that passing from an arbitrary realisation to the minimal realisation removes exactly those solutions that are

- *not observable*, i.e., yield the output $y \equiv 0$
- *not reachable* from $x_0 = 0$.

In order to obtain even smaller approximate models, we could thus remove those solutions of the model, which are "difficult to observe", i.e., produce only a small output and "difficult to reach", i.e., require a very large control to be reached. If both conditions are satisfied at the same time, then these solutions will only contribute very little to the map from u to y. Removing exactly these solutions is the idea of balanced truncation.

Throughout this chapter we assume that A is Hurwitz. We will comment in Section 4.7 at the end about how to proceed if this is not the case. Moreover, we assume that all realisations under consideration are minimal, i.e., that (A, B) are controllable and (A, C) is observable.

4.1 Controllability and observability Gramians

Associated to the control system (1.1) we define the following two matrices.

$$P = \int_0^\infty e^{At} B B^T e^{A^T t} dt \quad \text{and} \quad Q = \int_0^\infty e^{A^T t} C^T C e^{At} dt. \tag{4.1}$$

The matrix P is called the *controllability Gramian* and the matrix Q the *observability Gramian.* As proved in Lemma 2.11 of [4], the image of

$$W_\tau = \int_0^\tau e^{At} B B^T e^{A^T t} dt$$

is exactly the reachability subspace $\mathcal{R}$. Since A is Hurwitz, there exist $C, \sigma > 0$ such that

$$\|W_\tau - P\| = \left\| \int_\tau^\infty e^{At} B B^T e^{A^T t} dt \right\| \leq C e^{-\sigma\tau}.$$

In case (A,B) is controllable, we have that $\mathcal{R} = \mathbb{R}^n$ and thus W_τ has full rank for each $\tau > 0$. Since, moreover, the integrand is positive semidefinite, W_τ is positive definite for each τ and $\tau \mapsto x^T W_\tau x$ is positive and increasing for each $x \in \mathbb{R}^n \setminus \{0\}$. This implies that there is $c > 0$ with $x^T W_\tau x \geq c\|x\|^2$ for all $\tau \geq 1$ and consequently we obtain $x^T P x \geq c\|x\|^2$. Hence, P is positive definite and in particular it has full rank. Via duality, the same holds for Q is (A,C) is observable.

The interpretation of P follows from a fact that we observed in [4]: If we want to find a control function that steers the system from $x_0 = 0$ at time 0 to $x(t;0,u) = x_1$ at time $t > 0$, then this is accomplished by the control function

$$u(\tau) = B^T e^{A^T(t-\tau)} W_t^{-1} x_1.$$

Now consider the SVD of W_t and let $x_1 = u_k$ be the k-th column of the matrix U. Then

$$W_t^{-1} x_1 = \frac{1}{\sigma_k} v_k$$

is large if the singular value is small. This means that a large control $u(\cdot)$ is needed in order to steer the system into the direction u_k. In other words, subspaces corresponding to small singular values of W_t are difficult to reach, in the sense that with small control effort we will only move very little in this direction. Since small singular values of P correspond to small singular values of all W_t, $t \geq 0$, we can also use the singular values of P in order to determine subspaces that are difficult to reach. Similarly, subspaces that correspond to small singular values of Q will only slightly affect the output y.

Computing P and Q via the integral formulas in (4.1) is possible but not very convenient. The following theorem provides an alternative way.

Theorem 4.1 If A is Hurwitz, then the Gramians P and Q from (4.1) are the unique solutions of the Lyapunov equations

$$AP + PA^T + BB^T = 0 \quad \text{and} \quad A^TQ + QA + C^TC = 0. \tag{4.2}$$

Proof: We first show that P and Q solve the Lyapunov equations. Inserting the definition of P from (4.1) into the left equation in (4.2) yields

$$\begin{aligned} AP + PA^T + BB^T &= A\int_0^\infty e^{At}BB^Te^{A^Tt}dt + \int_0^\infty e^{At}BB^Te^{A^Tt}dtA + BB^T \\ &= \int_0^\infty \underbrace{Ae^{At}BB^Te^{A^Tt} + e^{At}BB^Te^{A^Tt}A^T}_{=\frac{d}{dt}e^{At}BB^Te^{A^Tt}} dt + BB^T \\ &= \underbrace{\lim_{t\to\infty} e^{At}BB^Te^{A^Tt}}_{=0 \text{ since } A \text{ is Hurwitz}} - \underbrace{e^{A0}BB^Te^{A^T0}}_{=\mathrm{Id}B^TB\mathrm{Id}=BB^T} + BB^T \;=\; 0. \end{aligned}$$

A similar computation shows the claim for Q.

The Lyapunov equations are systems of n^2 linear equations for the coefficients of P and Q, respectively. As we have seen in Lemma 3.13 of [4], if A is Hurwitz then a Lyapunov equation has a unique solution. There the matrix C (which plays the role of BB^T and C^TC, respectively, in (4.2)) is assumed to be positive definite, but this property is not important for proving the unique solvability. □

Readily implemented solution algorithms for Lyapunov equations are available for Python in various packages and for MATLAB in the control systems toolbox. We will discuss the ideas behind some of these algorithms in Section 4.6.

Example 4.2 We consider the Heat equation from Example 2.7. Computing P and Q and their singular value decompositions yields the singular values

$$60.5925, 16.2403, 6.1467, 1.3219, 0.1808, 0.1808, 0.0010, 0.0000, \dots$$

for P and

$$0.0315, 0.0034, 0.0005, 0.0001, 0.0000, \dots$$

for Q. These values were computed for $n = 12$ discretization points but actually do not change much if n is changed. What is immediately seen is that most of these values are very small. There thus seems to be potential for reducing the order of the model without changing the solutions very much. □

The main conclusion of this section is: good candidates for subspaces that can be neglected are those for which the singular values of both P and Q are small. This however, leads to the question how we can determine singular values *simultaneously* for P and Q. Clearly, in general they do not need to coincide, at all. However, in suitable coordinates we can achieve this property. This is the concept of balanced realisations that is described in the next section.

4.2 Balanced realisations

We recall that we assume that (A, B, C, D) is a minimal realisation, i.e., (A, B) is controllable and (A, C) is observable. Then P and Q are positive definite, hence invertible, and we obtain

$$PQ = P(QP)P^{-1},$$

implying that PQ and QP are similar matrices and thus have the same eigenvalues.

Moreover, P and Q are symmetric (because the integrands in (4.1) are symmetric), hence they are diagonalizable. In particular, there are an orthogonal matrix V and a diagonal matrix Λ with

$$P = V\Lambda V^T.$$

The diagonal matrix $\Lambda = \operatorname{diag}(\lambda_1, \dots, \lambda_n)$ contains the eigenvalues λ_j of P on the diagonal, which due to the positive definiteness of P are all positive. Hence, their square roots are real and $\Lambda^{1/2} = \operatorname{diag}(\sqrt{\lambda_1}, \dots, \sqrt{\lambda_n})$ is a real matrix. We then define

$$P^{1/2} = V\Lambda^{1/2}V^T.$$

Then $P = P^{1/2}P^{1/2}$. The eigenvalues of $P^{1/2}$ are $\sqrt{\lambda_1}, \dots, \sqrt{\lambda_n} > 0$, thus $P^{1/2}$ is positive definite, hence invertible, and symmetric since $(P^{1/2})^T = (V\Lambda^{1/2}V^T)^T = (V^T)^T(\lambda^{1/2})^T V^T = V\Lambda^{1/2}V^T = P^{1/2}$.

Since Q is also positive definite we can thus conclude that

$$P^{1/2}QP^{1/2}$$

is positive definite, because $x^T P^{1/2}QP^{1/2}x = x^T(P^{1/2})^T QP^{1/2}x = y^T Qy > 0$ if $x \neq 0$, because then $y = P^{1/2}x \neq 0$ since $P^{1/2}$ is invertible.

Denoting the inverse of $P^{1/2}$ by $P^{-1/2}$, this yields

$$PQ = P(QP)P^{-1} = P^{1/2}P^{1/2}QP^{1/2}P^{1/2}P^{-1/2}P^{-1/2} = P^{1/2}(P^{1/2}QP^{1/2})P^{-1/2}.$$

This means that PQ (and thus also QP) is similar to a positive definite matrix and thus positive definite itself.

Definition 4.3 The square roots of the (positive) eigenvalues of PQ are called the *Hankel singular values* of (1.1). We denote them by $\sigma_1, \dots, \sigma_n$ with the convention that $\sigma_{i+1} \leq \sigma_i$. □

Note that the these values are in general not singular values in the sense of Theorem 3.1. While these are the roots of the eigenvalues of A^TA, the Hankel singular values are the roots of the eigenvalues of PQ. However, the Hankel singular values are "regular" singular values of PQ if $P = Q$ holds, because then $PQ = PP = P^TP$ due to symmetry of P. We will next show that with an appropriate coordinate transformation we can always achieve $P = Q$, even with a particularly nice form.

To ensure that this procedure makes sense, we first need to show that the Hankel singular values do not change under coordinate transformations. To this end, consider a system (1.1) defined by (A, B, C, D), an invertible matrix T, and the transformed system given by

$$(\widetilde{A}, \widetilde{B}, \widetilde{C}, D) = (TAT^{-1}, TB, CT^{-1}, D).$$

Then, using Theorem 4.1 and $(T^T)^{-1} = (T^{-1})^T =: T^{-T}$ one sees that $\widetilde{P} = TPT^T$ satisfies

$$\widetilde{A}\widetilde{P} + \widetilde{P}\widetilde{A}^T + \widetilde{B}\widetilde{B}^T = TAPT^T + TPA^TT^T + TBB^TT^T = T(AP + PA^T + BB^T)T^T = 0$$

and thus again by Theorem 4.1 $\widetilde{P}$ is the controllability Gramian for the transformed system. Analogously one checks that $\widetilde{Q} = T^{-T}QT^{-1}$ is the observability Gramian for the transformed system. This leads to the following proposition.

Proposition 4.4 Assume that A is Hurwitz and T is invertible. Then the Hankel singular values of a system (1.1) defined by (A, B, C, D) and the transformed system

$$(\widetilde{A}, \widetilde{B}, \widetilde{C}, D) = (TAT^{-1}, TB, CT^{-1}, D)$$

coincide.

Proof: The Hankel singular values are uniquely determined by the eigenvalues of PQ for the non-transformed system and by the eigenvalues of $\widetilde{P}\widetilde{Q}$ for the transformed system. Since

$$\widetilde{P}\widetilde{Q} = TPT^TT^{-T}QT^{-1} = TPQT^{-1}$$

the matrices $\widetilde{P}\widetilde{Q}$ and PQ are similar and thus have the same eigenvalues. □

As already mentioned above, it would now be desirable to find a coordinate transformation such that $\widetilde{P} = \widetilde{Q}$, because then the Hankel singular values are "regular" singular values of both matrices P and Q and thus allow us to identify subspaces which are simultaneously difficult to reach and difficult to observe. The next theorem shows that this is indeed possible, even with diagonal matrices $\widetilde{P}$ and $\widetilde{Q}$.

Theorem 4.5 Assume that A is Hurwitz. Consider the coordinate transformation

$$T_b = \Sigma^{-1/2}V^TR,$$

where $P = S^TS$ and $Q = R^TR$ are Choleski factorisations of P and Q, respectively, and $SR^T = U\Sigma V^T$ is an SVD of SR^T. Then the transformed Gramians are of the form

$$\widetilde{P} = \widetilde{Q} = \Sigma = \begin{pmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{pmatrix},$$

where $\sigma_1, \dots, \sigma_n$ are the Hankel singular values. The corresponding realisation $(\widetilde{A}, \widetilde{B}, \widetilde{C}, D)$ is then called a *balanced realisation.*

Proof: Since $(U\Sigma V^T)^{-T} = (V^{-T}\Sigma^{-1}U^{-1})^T = U^{-T}\Sigma^{-1}V^T = U\Sigma^{-1}V^T$, we obtain

$$S^TU\Sigma^{-1/2}T_b = S^TU\Sigma^{-1/2}\Sigma^{-1/2}V^TR = S^T(U\Sigma V^T)^{-T}R = S^T(SR^T)^{-T}R = S^TS^{-T}R^{-1}R = \text{Id}.$$

Hence, $T_b^{-1} = S^TU\Sigma^{-1/2}$. This implies

$$\begin{aligned} T_bPQT_b^{-1} &= \Sigma^{-1/2}V^T\,RS^T\,SR^T\,RS^T\,U\Sigma^{-1/2} \\ &= \Sigma^{-1/2}V^T\,V\Sigma U^T\,U\Sigma V^TV\Sigma U^T\,U\Sigma^{-1/2} \\ &= \Sigma^{-1/2}\Sigma\Sigma\Sigma\Sigma^{-1/2} \;=\; \Sigma^2, \end{aligned}$$

which shows that Σ contains the Hankel singular values. Further we obtain

$$\begin{aligned} T_bPT_b^T &= \Sigma^{-1/2}V^T\,RS^T\,SR^T\,V\Sigma^{-1/2} \\ &= \Sigma^{-1/2}V^T\,V\Sigma U^TU\,\Sigma V^T\,V\Sigma^{-1/2} \\ &= \Sigma^{-1/2}\Sigma\Sigma\Sigma^{-1/2} \;=\; \Sigma, \end{aligned}$$

which proves the claimed form of P. Finally

$$\begin{aligned} T_b^{-T}QT_b^{-1} &= \Sigma^{-1/2}U^T\,SR^T\,RS^T\,U\Sigma^{-1/2} \\ &= \Sigma^{-1/2}U^T\,U\Sigma V^T\,V\Sigma U^T\,U\Sigma^{-1/2} \\ &= \Sigma^{-1/2}\Sigma\Sigma\Sigma^{-1/2} \;=\; \Sigma \end{aligned}$$

shows that Q has the claimed form.

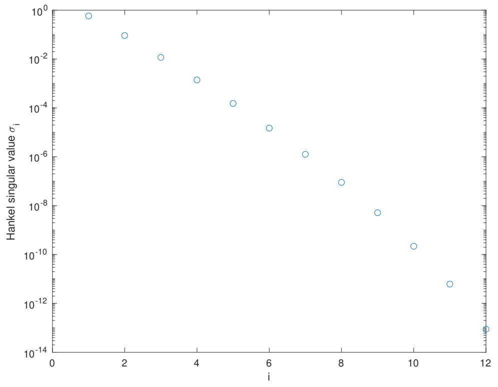

Figure 4.1: Hankel singular values for Example 2.7 with $n = 15$

Example 4.6 We consider again the Heat equation from Example 2.7 and 4.2. For $n = 15$ we obtain the Hankel singular values shown in Figure 4.1.

It is clear that only the first few of these values have a significant size. □

4.3 The algorithm

The balanced realisation leads to a transformation of (1.1) of the form

$$\begin{aligned} \dot{\tilde{x}}(t) &= \widetilde{A}\tilde{x}(t) + \widetilde{B}u(t) \\ y(t) &= \widetilde{C}\tilde{x}(t) + Du(t) \end{aligned} \tag{4.3}$$

with $\tilde{x} = T_b x$. In these new coordinates, the components $\tilde{x}_i$ of the transformed state $\tilde{x}$ and the associated subspace $\langle e_i \rangle$ correspond to the singular value σ_i of $\widetilde{P}$ and $\widetilde{Q}$. This means that if $\tilde{\sigma}$ is small, then this subspace is both difficult to reach (i.e., the $\tilde{x}_i$-component is affected only very little by the control u) and difficult to observe (i.e., the $\tilde{x}_i$-component contributes only very little to y). Thus, the $\tilde{x}_i$-component play only a very small role for the relation between u and y and when it is removed from the model, then the transfer function changes only very little (this intuitive idea will be made quantitatively precise later in this chapter).

The idea of balanced truncation thus lies in removing the subspaces corresponding to singular values that are very small. As Example (4.8) shows, this can be the vast majority of the singular values. To this end, we select r such that $\sigma_{r+1}, \ldots, \sigma_n$ are below a chosen threshold, partition (4.3) as

$$\begin{aligned} \begin{pmatrix} \dot{x}_r(t) \\ \dot{\tilde{x}}_2(t) \end{pmatrix} &= \begin{pmatrix} A_r & \widetilde{A}_{12} \\ \widetilde{A}_{21} & \widetilde{A}_{22} \end{pmatrix} \begin{pmatrix} x_r(t) \\ \tilde{x}_2(t) \end{pmatrix} + \begin{pmatrix} B_r \\ \widetilde{B}_2 \end{pmatrix} u(t) \\ y(t) &= (C_r \; \widetilde{C}_2) \begin{pmatrix} x_r(t) \\ \tilde{x}_2(t) \end{pmatrix} + Du(t) \end{aligned} \tag{4.4}$$

and define the reduced system as

$$\begin{aligned} \dot{x}_r(t) &= A_r x_r(t) + B_r u(t) \\ y_r(t) &= C_r x_r(t) + Du(t). \end{aligned} \tag{4.5}$$

This procedure can be slightly simplified, by observing that if we only need A_r, B_r and C_r and not the rest of $\widetilde{A}$, $\widetilde{B}$ and $\widetilde{C}$, then it is sufficient to compute only the part of the coordinate transformation T_b that is actually needed for computing A_r, B_r and C_r. If we split up the coordinate transformation and its inverse in the form

$$T_b = \begin{pmatrix} T_r \\ T_2 \end{pmatrix}, \qquad T_b^{-1} = \begin{pmatrix} T_r^i & T_2^i \end{pmatrix}$$

according to the partition of $\widetilde{A}$, then we obtain

$$\begin{pmatrix} A_r & \widetilde{A}_{12} \\ \widetilde{A}_{21} & \widetilde{A}_{22} \end{pmatrix} = T_b A T_b^i = \begin{pmatrix} T_r \\ T_2 \end{pmatrix} A (T_r^i \; T_2^i) = \begin{pmatrix} T_r A T_r^i & T_r A T_2^i \\ T_2 A T_r^i & T_2 A T_2^i \end{pmatrix},$$

implying that $A_r = T_r A T_r^i$. This means, we only need to compute T_r and T_r^i. Since

$$T_b = \Sigma^{-1/2} V^T R \quad \text{and} \quad T_b^{-1} = S^T U \Sigma^{-1/2},$$

when we partition the singular value decomposition of SR^T as

$$U \Sigma V^T = \begin{pmatrix} U_1 & U_2 \end{pmatrix} \begin{pmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{pmatrix} \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix},$$

then we obtain that

$$T_b = \begin{pmatrix} \Sigma_1^{-1/2} V_1^T R \\ \Sigma_2^{-1/2} V_2^T R \end{pmatrix} \quad \text{and} \quad T_b^{-1} = \begin{pmatrix} S^T U_1 \Sigma_1^{-1/2} & S^T U_2 \Sigma_2^{-1/2} \end{pmatrix}$$

and thus

$$T_r = \Sigma_1^{-1/2} V_1^T R \quad \text{and} \quad T_r^i = S^T U_1 \Sigma_1^{-1/2}.$$

All in all, this leads to the following algorithm.

Algorithm 4.7 (Balanced Truncation)

Input: $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{l \times n}$, $D \in \mathbb{R}^{l \times m}$, (A, B) controllable, (A, C) observable, A Hurwitz

(1) Solve the Lyapunov equations

$$AP + PA^T = -BB^T, \qquad A^T Q + QA = -C^T C.$$

(2) Compute the Choleski factorisation

$$P = S^T S, \qquad Q = R^T R.$$

(3) Compute the singular value decomposition

$$SR^T = U\Sigma V^T = \begin{pmatrix} U_1 & U_2 \end{pmatrix} \begin{pmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{pmatrix} \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix}$$

with $\Sigma_1 \in \mathbb{R}^{r\times r}$ for a given $r \in \mathbb{N}$, $r < n$.

(4) Compute the (reduced) coordinate transformations

$$T_r = \Sigma_1^{-1/2} V_1^T R \quad \text{and} \quad T_r^i = S^T U_1 \Sigma_1^{-1/2}.$$

(5) Compute the reduced system matrices

$$A_r = T_r A T_r^i, \quad B_r = T_r B, \quad C_r = C T_r^i \quad D_r = D.$$

Output: $A_r \in \mathbb{R}^{r\times r}$, $B_r \in \mathbb{R}^{r\times m}$, $C_r \in \mathbb{R}^{l\times r}$, $D_r \in \mathbb{R}^{l\times m}$ □

Example 4.8 We consider once more the Heat equation from Example 2.7, 4.2, and 4.8. We computed A_r for $r = 2$ and $r = 5$, solved the resulting control system with $u(t) = \cos t$ and $x_0 = 0$ and plotted the resulting $y(t)$ (black) and $y_r(t)$ (red) in Figure 4.2.

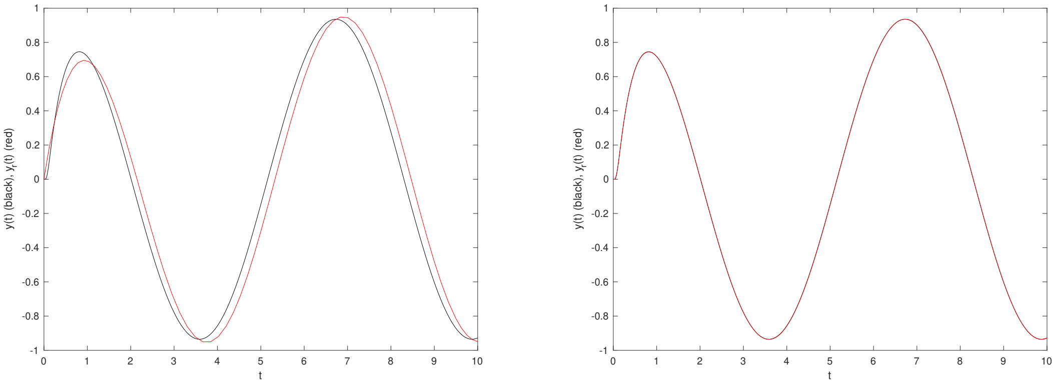

Figure 4.2: Output of full (black) and reduced model (red) with $r = 1$ (left) and $r = 3$ (right)

One sees that already with $r = 1$ the output of the (2-dimensional) reduced order system matches that of the full model rather well. For $r = 3$, there is almost no visible difference anymore between the outputs.

□

4.4 Asymptotic stability of the reduced system

We now want to analyze the properties of the reduced model. Recall that we have assumed A to be Hurwitz, implying that solutions converge to 0 if $u \equiv 0$. Clearly, if A_r is supposed to be a meaningful approximation to A, then it should have the same property. This will also be a necessary property for being able to analyze the approximation error in the next section.

Theorem 4.9 Consider a system (1.1), which is a minimal realisation with Hurwitz matrix A and Hankel singular values $\sigma_1 \geq \sigma_2 \geq \sigma_n > 0$. Then for any $r = 1, \ldots, n-1$ the reduced system (4.5) is balanced with Hurwitz matrix A_r and Hankel singular values $\sigma_1, \sigma_2, \ldots, \sigma_r$.

Proof: From Theorem 4.5 it follows that the matrices from the balanced system (4.3) satisfy

$$\widetilde{A}\Sigma + \Sigma\widetilde{A}^T + \widetilde{B}\widetilde{B}^T = 0 \tag{4.6}$$

as well as

$$\widetilde{A}^T\Sigma + \Sigma\widetilde{A} + \widetilde{C}^T\widetilde{C} = 0. \tag{4.7}$$

Inserting the decomposition from (4.4) into (4.6) and decomposing Σ accordingly into $\Sigma_1 = \operatorname{diag}(\sigma_1, \ldots, \sigma_r)$ and $\Sigma_2 = \operatorname{diag}(\sigma_{r+1}, \ldots, \sigma_n)$, we obtain

$$\begin{pmatrix} A_r & \widetilde{A}_{12} \\ \widetilde{A}_{21} & \widetilde{A}_{22} \end{pmatrix} \begin{pmatrix} \Sigma_1 & \\ & \Sigma_2 \end{pmatrix} + \begin{pmatrix} \Sigma_1 & \\ & \Sigma_2 \end{pmatrix} \begin{pmatrix} A_r^T & \widetilde{A}_{21}^T \\ \widetilde{A}_{12}^T & \widetilde{A}_{22}^T \end{pmatrix} + \begin{pmatrix} B_r B_r^T & B_r \widetilde{B}_2^T \\ \widetilde{B}_2 B_r^T & \widetilde{B}_2 \widetilde{B}_2^T \end{pmatrix} = 0,$$

implying

$$A_r\Sigma_1 + \Sigma_1 A_r^T + B_r B_r^T = 0 \tag{4.8}$$

Analogously we obtain

$$A_r^T\Sigma_1 + \Sigma_1 A_r + C_r^T C_r = 0 \tag{4.9}$$

If A_r is Hurwitz then these equations have unique solutions, which implies that Σ_1 is both the controllability and the observability Gramian for (4.5). This shows that (4.5) is balanced with Hankel singular values $\sigma_1, \sigma_2, \ldots, \sigma_r$.

It remains to be shown that A_r is Hurwitz, i.e., that it has only eigenvalues with negative real part. We start by proving that all eigenvalues have nonpositive real part. To this end, let $\lambda \in \mathbb{C}$ be an eigenvalue of A_r with left eigenvector $\bar{v} \neq 0$, i.e., $\bar{v}^T A_r = \bar{\lambda}\bar{v}^T$ or, equivalently, $A_r^T v = \lambda v$. Multiplying (4.8) with v^T and v from the left and right, respectively, yields

$$0 = \bar{v}^T(A_r\Sigma_1 + \Sigma_1 A_r^T + B_r B_r^T)v = \bar{\lambda}\bar{v}^T\Sigma_1 x + \bar{v}^T\Sigma_1\lambda v + \bar{v}^T B_r B_r^T v.$$

This implies

$$(\bar{\lambda} + \lambda)\bar{v}^T\Sigma_1 v = -\bar{v}^T B_r B_r^T v.$$

Since Σ_1 is positive definite and $B_r B_r^T$ is positive semidefinite, this can only be true when $\bar{\lambda} + \lambda \leq 0$ holds. This, in turn, is only possible if the real part of λ is less or equal 0.

In the final (and longest) step of this proof we now show that A_r cannot have purely imaginary eigenvalues. We proceed by contradiction and thus assume that A_r has purely imaginary eigenvalues. Then there exists a coordinate transformation T_1 such that

$$T_1 A_r T_1^{-1} = \begin{pmatrix} F_{11} & 0 \\ 0 & F_{22} \end{pmatrix}$$

holds, where F_{11} has only eigenvalues with negative real part and F_{22} has only purely imaginary eigenvalues. Defining $T = \begin{pmatrix} T_1 & \\ & \mathrm{Id} \end{pmatrix}$ and using the decomposition from (4.4), we obtain

$$\widehat{A} := T\widetilde{A}T^{-1} = \begin{pmatrix} F_{11} & 0 & F_{13} \\ 0 & F_{22} & F_{23} \\ F_{31} & F_{32} & \widetilde{A}_{22} \end{pmatrix}.$$

Likewise, we transform

$$\widehat{B} := T\widetilde{B} = \begin{pmatrix} G_1 \\ G_2 \\ \widetilde{B}_2 \end{pmatrix}, \qquad \widehat{C} := \widetilde{C}T^{-1} = \begin{pmatrix} H_1 & H_2 & \widetilde{C}_2 \end{pmatrix},$$

$$\widehat{P} := T\widetilde{P}T^T = \begin{pmatrix} P_{11} & P_{12} & 0 \\ P_{12}^T & P_{22} & 0 \\ 0 & 0 & \Sigma_2 \end{pmatrix}, \qquad \widehat{Q} := T^{-T}\widetilde{Q}T^{-1} = \begin{pmatrix} Q_{11} & Q_{12} & 0 \\ Q_{12}^T & Q_{22} & 0 \\ 0 & 0 & \Sigma_2 \end{pmatrix}.$$

We now prove that

$$G_2 = 0, H_2 = 0, P_{12} = 0, Q_{12} = 0, F_{23} = 0, \text{ and } F_{32} = 0. \tag{4.10}$$

This in particular implies

$$\widehat{A} = \begin{pmatrix} F_{11} & 0 & F_{13} \\ 0 & F_{22} & 0 \\ F_{31} & 0 & \widetilde{A}_{22} \end{pmatrix} \quad \text{and} \quad \widehat{B} = \begin{pmatrix} G_1 \\ 0 \\ \widetilde{B}_2 \end{pmatrix}.$$

Using this particular block structure, it follows that A_r has no purely imaginary eigenvalues: if $\lambda = \beta i$, $\beta \in \mathbb{R}$ is such an eigenvalue, then by choice of T_1 it must be an eigenvalue of F_{22}. Then $F_{22} - \lambda \mathrm{Id}$ is singular, and consequently $(\hat{A} - \lambda \mathrm{Id} \ B)$ is singular. By the Hautus criterion (Theorem 2.16 in [4]), this implies that $(\widehat{A}, \widehat{B})$ is not controllable, hence (A, B) are also not controllable, since controllability is invariant under coordinate transformations. This, however, contradicts the assumed minimality of the realisation.

We are thus left to prove (4.10). To this end, we consider the Lyapunov equations for $\widehat{P}$ and $\widehat{C}$, which in partitioned form read

$$\begin{pmatrix} F_{11} & 0 & F_{13} \\ 0 & F_{22} & F_{23} \\ F_{31} & F_{32} & \widetilde{A}_{22} \end{pmatrix}\begin{pmatrix} P_{11} & P_{12} & 0 \\ P_{12}^T & P_{22} & 0 \\ 0 & 0 & \Sigma_2 \end{pmatrix} + \begin{pmatrix} P_{11} & P_{12} & 0 \\ P_{12}^T & P_{22} & 0 \\ 0 & 0 & \Sigma_2 \end{pmatrix}\begin{pmatrix} F_{11}^T & 0 & F_{31}^T \\ 0 & F_{22}^T & F_{32}^T \\ F_{13}^T & F_{23}^T & \widetilde{A}_{22}^T \end{pmatrix} + \begin{pmatrix} G_1G_1^T & G_1G_2^T & G_1\widetilde{B}_2^T \\ G_2G_1^T & G_2G_2^T & G_2\widetilde{B}_2^T \\ \widetilde{B}_2G_1^T & \widetilde{B}_2G_2^T & \widetilde{B}_2\widetilde{B}_2^T \end{pmatrix} = 0$$

and

$$\begin{pmatrix} F_{11}^T & 0 & F_{31}^T \\ 0 & F_{22}^T & F_{32}^T \\ F_{13}^T & F_{23}^T & \widetilde{A}_{22}^T \end{pmatrix}\begin{pmatrix} Q_{11} & Q_{12} & 0 \\ Q_{12}^T & Q_{22} & 0 \\ 0 & 0 & \Sigma_2 \end{pmatrix} + \begin{pmatrix} Q_{11} & Q_{12} & 0 \\ Q_{12}^T & Q_{22} & 0 \\ 0 & 0 & \Sigma_2 \end{pmatrix}\begin{pmatrix} F_{11} & 0 & F_{13} \\ 0 & F_{22} & F_{23} \\ F_{31} & F_{32} & \widetilde{A}_{22} \end{pmatrix} + \begin{pmatrix} H_1^TH_1 & H_1^TH_2 & H_1^T\widetilde{C}_2 \\ H_2^TH_1 & H_2^TH_2 & H_2^T\widetilde{C}_2 \\ \widetilde{C}_2^TH_1 & \widetilde{C}_2^TH_2 & \widetilde{C}_2^T\widetilde{C}_2 \end{pmatrix} = 0.$$

The $(2,2)$ block of the first equation yields

$$F_{22}P_{22} + P_{22}F_{22}^T + G_2G_2^T = 0.$$

One can now prove that since F_{22} has only purely imaginary eigenvalues and P_{22} is positive definite, $F_{22}P_{22} + P_{22}F_{22}^T$ is always indefinite, i.e., neither positive nor negative definite. Since $G_2G_2^T$ is always positive semidefinite, the equation can only be satisfied if $G_2G_2^T = 0$, i.e., if $G_2 = 0$. Analogously, from the $(2,2)$ block of the equation for $\widehat{Q}$ one obtains $H_2 = 0$. This shows the first two claims in (4.10).

Using $G_2 = 0$, from the $(1,2)$ block of the equation for $\widehat{P}$ one gets

$$F_{11}P_{12} + P_{12}F_{22}^T = 0.$$

This is a so-called Sylvester equation and one can prove that it has a unique solution P_{12} if F_{11} and $-F_{22}$ have disjoint sets of eigenvalues, which is the case here. Since $P_{12} = 0$ is obviously a solution, $P_{12} = 0$ follows. Similarly, one obtains $Q_{12} = 0$ from the $(1,2)$ block of the equation for $\widehat{Q}$. This shows the third and fourth claim in (4.10).

Finally, the $(2,3)$ block of the equation for $\widehat{P}$ together with $G_2 = 0$ and $P_{12} = 0$ yields

$$F_{23}\Sigma_2 + P_{22}F_{32}^T = 0.$$

Analogously, the $(2,3)$ block of the equation for $\widehat{Q}$ yields

$$F_{32}^T\Sigma_2 + Q_{22}F_{23} = 0.$$

Multiplying the first equation with Σ_2 from the right and using the second equation we obtain

$$0 = F_{23}\Sigma_2^2 + P_{22}F_{32}^T\Sigma_2 = F_{23}\Sigma_2^2 - P_{22}Q_{22}F_{23},$$

which is a Sylvester equation for F_{23}. In order to ensure that this equation has a unique solution, we have to show that Σ_2^2 and $P_{22}Q_{22}$ have no common eigenvalues. The matrix Σ_2^2 has the eigenvalues $\sigma_{r+1}^2, \dots, \sigma_n^2$. Since $P_{12} = 0$ and $Q_{12} = 0$, we have that

$$\widehat{P} := T\widetilde{P}T^T = \begin{pmatrix} P_{11} & 0 & 0 \\ 0 & P_{22} & 0 \\ 0 & 0 & \Sigma_2 \end{pmatrix}, \qquad \widehat{Q} := T^{-T}\widetilde{Q}T^{-1} = \begin{pmatrix} Q_{11} & 0 & 0 \\ 0 & Q_{22} & 0 \\ 0 & 0 & \Sigma_2 \end{pmatrix},$$

implying

$$\widehat{P}\widehat{Q} = \begin{pmatrix} P_{11}Q_{11} & 0 & 0 \\ 0 & P_{22}Q_{22} & 0 \\ 0 & 0 & \Sigma_2^2 \end{pmatrix}.$$

Moreover, we know that

$$\widehat{P}\widehat{Q} = T\widetilde{P}\widetilde{Q}T^{-1} = \begin{pmatrix} T_1\Sigma_1^2T_1^{-1} & 0 \\ & \Sigma_2^2 \end{pmatrix}.$$

Together this implies that the eigenvalues of $P_{22}Q_{22}$ must be contained in set of eigenvalues of Σ_1^2, i.e., in $\{\sigma_1^2, \dots, \sigma_r^2\}$. Thus, Σ_2 and $P_{22}Q_{22}$ have disjoint sets of eigenvalues and thus the Sylvester equation for F_{23} has a unique solution, which must thus be 0. This also implies $F_{32} = 0$ and thus we have obtained the last two claims in (4.10). □

4.5 Approximation error

Finally we want to determine the error of this procedure, i.e., the difference between (1.1) and (4.5). To this end we want to estimate the error between the outputs $y(\cdot)$ and $y_r(\cdot)$. Since these depend on the applied input $u(\cdot)$, it makes sense to compute the error between the maps from u to y or y_r, respectively. It turns out that it is easier to do this for the L^2-norm of the input and the output. If we measure the difference between y and y_r in the L^2-norm, then we can use the inequality

$$\|y - y_r\|_{L^2} \leq \|G - G_r\|_{H^\infty}\|u\|_{L^2},$$

which is proved in [1, Discussion after Korollar 7.29]. Here $\|\cdot\|_{H^\infty}$ denotes the H-infinity norm, which for a function $G : \mathbb{C} \to \mathbb{C}^{l\times m}$ is given by

$$\|G\|_{H^\infty} := \sup_{\omega\in\mathbb{R}} \sigma_{\max}(G(i\omega)),$$

where $\sigma_{max}(Z)$ denotes the largest singular value of a matrix Z. Clearly, the fact that the H-infinity norm is determined by a singular value helps for our considerations. However, it should be mentioned that other norms might be favourable. For instance, if we liked to estimate the difference $\|y - y_r\|_{L^\infty}$ in the L^∞-norm, then the H^2-norm of G would be the more appropriate object. We will come back to this at the end of this section.

In order to obtain the desired error estimate we renumber the Hankel singular values by removing all values that occur multiple times, leading to pairwise distinct values $\sigma_1 > \sigma_2 > \ldots > \sigma_\ell > 0$, with $\ell \le n$. Denoting by n_i the multiplicity of each σ_i, we can rewrite $\widetilde{P} = \widetilde{Q} = \Sigma$ from Theorem 4.5 as

$$\widetilde{P} = \widetilde{Q} = \begin{pmatrix} \sigma_1 \mathrm{Id}_{n_1} & & & \\ & \sigma_2 \mathrm{Id}_{n_2} & & \\ & & \ddots & \\ & & & \sigma_\ell \mathrm{Id}_{n_\ell} \end{pmatrix},$$

where Id_{n_i} is the $n_i \times n_i$ identity matrix. We now consider first the case that we truncate the $\sigma_\ell \mathrm{Id}_{n_\ell}$ block from the system, i.e., that $r = n - n_\ell$. We define the "error"-transfer function

$$G_e(s) := G(s) - G_r(s) = \widetilde{C}(s\mathrm{Id} - \widetilde{A})^{-1}\widetilde{B} - C_r(s\mathrm{Id} - A_r)^{-1}B_r.$$

Then, using the notation from (4.4), G_e is the transfer function of system (1.1) with matrices

$$A_e := \begin{pmatrix} \widetilde{A} & 0 \\ 0 & A_r \end{pmatrix} = \begin{pmatrix} A_r & \widetilde{A}_{12} & 0 \\ \widetilde{A}_{21} & \widetilde{A}_{22} & 0 \\ 0 & 0 & A_r \end{pmatrix}, \quad B_e := \begin{pmatrix} B_r \\ \widetilde{B}_2 \\ B_r \end{pmatrix}, \quad \text{and} \quad C_e := (C_r \, \widetilde{C}_2 \, -C_r).$$

The state and output of this system we denote by

$$z := \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{x}_r \end{pmatrix} \quad \text{and} \quad e := y - y_r.$$

Since $\widetilde{A}$ and A_r are Hurwitz, A_e is Hurwitz, too. By this construction, we obtain

$$G_e(s) = C_e(s\mathrm{Id} - A_e)^{-1}B_e.$$

Lemma 4.10 Consider a system (1.1), which is a minimal realisation with Hurwitz matrix A and pairwise distinct Hankel singular values $\sigma_1 > \sigma_2 > \ldots > \sigma_\ell > 0$ with multiplicities $n_1, \ldots, n_\ell$. Then for $r = n - n_\ell$ the inequality

$$\|G_e\|_{H^\infty} \le 2\sigma_\ell$$

holds.

Proof: We use the state transformation

$$T = \frac{1}{2}\begin{pmatrix} -\mathrm{Id}_r & 0 & \mathrm{Id}_r \\ 0 & 2\mathrm{Id}_{n_\ell} & 0 \\ \mathrm{Id}_r & 0 & \mathrm{Id}_r \end{pmatrix} \quad \text{with inverse} \quad T^{-1} = \begin{pmatrix} -\mathrm{Id}_r & 0 & \mathrm{Id}_r \\ 0 & \mathrm{Id}_{n_\ell} & 0 \\ \mathrm{Id}_r & 0 & \mathrm{Id}_r \end{pmatrix}.$$

This yields the transformed state

$$Tz = \begin{pmatrix} x_r - \tilde{x}_1 \\ 2\tilde{x}_2 \\ x_r + \tilde{x}_1 \end{pmatrix}$$

and the transformed matrices

$$\widetilde{A}_e = TA_eT^{-1} = \begin{pmatrix} A_r & -\frac{1}{2}\widetilde{A}_{12} & 0 \\ -\frac{1}{2}\widetilde{A}_{21} & \widetilde{A}_{22} & \widetilde{A}_{21} \\ 0 & \frac{1}{2}\widetilde{A}_{12} & A_r \end{pmatrix}, \quad \widetilde{B}_e = TB_e = \begin{pmatrix} 0 \\ \widetilde{B}_2 \\ B_r \end{pmatrix}, \quad \widetilde{C}_e = C_eT^{-1} = (-2C_r\, \widetilde{C}_2\, 0).$$

Since the transfer function is invariant under state transformation, we obtain $G_e(s) = \widetilde{C}_r(s\mathrm{Id} - \widetilde{A}_e)\widetilde{B}_e$. Now we augment $\widetilde{B}_e$ and $\widetilde{C}_e$ by setting

$$\widetilde{\widetilde{B}}_e := \begin{pmatrix} 0 & \sigma_\ell\Sigma_1^{-1}C_r^T \\ \widetilde{B}_2 & -\widetilde{C}_2^T \\ B_r & 0 \end{pmatrix} = (\widetilde{B}_e\, \widetilde{\widetilde{B}}_2) \quad \text{and} \quad \widetilde{\widetilde{C}}_e = \begin{pmatrix} -2C_r & \widetilde{C}_2 & 0 \\ 0 & -\widetilde{B}_2^T & -2\sigma_\ell B_r^T\Sigma_1^{-1} \end{pmatrix} = \begin{pmatrix} \widetilde{C}_e \\ \widetilde{\widetilde{C}}_2 \end{pmatrix}$$

and define

$$\widetilde{\widetilde{D}}_e = \begin{pmatrix} 0 & 2\sigma_\ell\mathrm{Id}_{n_\ell} \\ 2\sigma_\ell\mathrm{Id}_{n_\ell} & 0 \end{pmatrix}.$$

The transfer function for $(\widetilde{A}_e, \widetilde{\widetilde{B}}_e, \widetilde{\widetilde{C}}_e, \widetilde{\widetilde{D}}_e)$ is

$$\widetilde{G}_e(s) = \widetilde{\widetilde{C}}_e(s\mathrm{Id} - \widetilde{A}_e)^{-1}\widetilde{\widetilde{B}}_e + \widetilde{\widetilde{D}}_e = \begin{pmatrix} G_e(s) & \widetilde{C}(s\mathrm{Id} - \widetilde{A}_e)^{-1}\widetilde{\widetilde{B}}_2 + 2\sigma_\ell\mathrm{Id}_{n_\ell} \\ \widetilde{\widetilde{C}}_2(s\mathrm{Id} - \widetilde{A}_e)^{-1}\widetilde{B} + 2\sigma_\ell\mathrm{Id}_{n_\ell} & \widetilde{\widetilde{C}}_2(s\mathrm{Id} - \widetilde{A}_e)^{-1}\widetilde{\widetilde{B}}_2 \end{pmatrix}.$$

From [Horn & Johnson, *Topics in Matrix Analysis*, Cambridge University Press, 1994, Theorem 3.1.2 and Corollary 3.1.3] it follows that the singular values of the submatrix $G_e(s)$ are less or equal than that of the full matrix $\widetilde{G}_e(s)$, which implies

$$\|G_e\|_{H^\infty} \le \|\widetilde{G}_e\|_{H^\infty}.$$

It thus suffices to show $\|\widetilde{G}_e\|_{H^\infty} \le 2\sigma_\ell$. To this end, we use that

$$\widetilde{\widetilde{P}}_e = \begin{pmatrix} \sigma_\ell^2\Sigma_1 & & \\ & 2\sigma_\ell\mathrm{Id}_{n_\ell} & \\ & & \Sigma_1 \end{pmatrix}$$

satisfies the equations

$$\begin{aligned} \widetilde{A}_e\widetilde{\widetilde{P}}_e + \widetilde{\widetilde{P}}_e\widetilde{A}_e^T + \widetilde{\widetilde{B}}_e\widetilde{\widetilde{B}}_e^T &= 0 && (4.11) \\ \widetilde{\widetilde{P}}_e\widetilde{\widetilde{C}}_e^T + \widetilde{\widetilde{B}}_e\widetilde{\widetilde{D}}_e^T &= 0. && (4.12) \end{aligned}$$

Here the second equation follows by direct computation and the first from (4.6) and the fact that by our choice of r it holds that $\Sigma_2 = \sigma_\ell \mathrm{Id}_{n_\ell}$. Instead of computing the singular values of $\widetilde{G}_e(i\omega)$ directly, we compute the roots of the eigenvalues of $\widetilde{G}_e(i\omega)\overline{\widetilde{G}_e(i\omega)}^T = \widetilde{G}_e(i\omega)\widetilde{G}_e(-i\omega)^T$.

To this end we consider

$$\widetilde{G}_e(s)\widetilde{G}_e(-s)^T = \left(\widetilde{\widetilde{C}}_e(s\mathrm{Id} - \widetilde{A}_e)^{-1}\widetilde{\widetilde{B}}_e + \widetilde{\widetilde{D}}_e\right)\left(\widetilde{\widetilde{B}}_e^T(-s\mathrm{Id} - \widetilde{A}_e^T)^{-1}\widetilde{\widetilde{C}}_e^T + \widetilde{\widetilde{D}}_e^T\right).$$

Since

$$\left(s\mathrm{Id} - \begin{pmatrix} \widetilde{A}_e & -\widetilde{\widetilde{B}}_e\widetilde{\widetilde{B}}_e^T \\ 0 & -\widetilde{A}_e \end{pmatrix}\right)^{-1} = \begin{pmatrix} (s\mathrm{Id} - \widetilde{A}_e)^{-1} & -(s\mathrm{Id} - \widetilde{A}_e)^{-1}\widetilde{\widetilde{B}}_e\widetilde{\widetilde{B}}_e^T(s\mathrm{Id} + \widetilde{A}_e)^{-1} \\ 0 & (s\mathrm{Id} + \widetilde{A}_e)^{-1} \end{pmatrix},$$

$\widetilde{G}_e(s)\widetilde{G}_e(-s)^T$ is the transfer function for the matrices

$$\begin{pmatrix} \widetilde{A}_e & -\widetilde{\widetilde{B}}_e\widetilde{\widetilde{B}}_e^T \\ 0 & -\widetilde{A}_e \end{pmatrix}, \quad \begin{pmatrix} \widetilde{\widetilde{B}}_e\widetilde{\widetilde{D}}_e^T \\ \widetilde{\widetilde{C}}_e^T \end{pmatrix}, \quad (\widetilde{\widetilde{C}}_e \; - \widetilde{\widetilde{D}}_e\widetilde{\widetilde{B}}_e^T), \quad (\widetilde{\widetilde{D}}_e\widetilde{\widetilde{D}}_e^T).$$

Transforming these matrices with

$$T = \begin{pmatrix} \mathrm{Id} & \widetilde{P} \\ 0 & \mathrm{Id} \end{pmatrix} \quad \text{with inverse} \quad T^{-1} = \begin{pmatrix} \mathrm{Id} & -\widetilde{P} \\ 0 & \mathrm{Id} \end{pmatrix}$$

(which does not change the transfer function) and using (4.12) yields the matrices

$$\begin{pmatrix} \widetilde{A}_e & 0 \\ 0 & -\widetilde{A}_e \end{pmatrix}, \quad \begin{pmatrix} 0 \\ \widetilde{\widetilde{C}}_e^T \end{pmatrix}, \quad (\widetilde{\widetilde{C}}_e \, 0), \quad \widetilde{\widetilde{D}}_e\widetilde{\widetilde{D}}_e^T.$$

This finally yields

$$\begin{aligned} \widetilde{G}_e(s)\widetilde{G}_e(-s)^T &= (\widetilde{\widetilde{C}}_e \, 0)\begin{pmatrix} (s\mathrm{Id} - \widetilde{A}_e)^{-1} & 0 \\ 0 & (s\mathrm{Id} + \widetilde{A}_e)^{-1} \end{pmatrix}\begin{pmatrix} 0 \\ \widetilde{\widetilde{C}}_e^T \end{pmatrix} + \widetilde{\widetilde{D}}_e\widetilde{\widetilde{D}}_e^T \\ &= \widetilde{\widetilde{D}}_e\widetilde{\widetilde{D}}_e^T = \begin{pmatrix} 4\sigma_\ell^2\mathrm{Id}_{n_\ell} & 0 \\ 0 & 4\sigma_\ell^2\mathrm{Id}_{n_\ell} \end{pmatrix}. \end{aligned}$$

Obviously, all eigenvalues of this matrix are equal to $4\sigma_\ell^2$, which implies that all singular values of $\widetilde{G}_e(s)$ are equal to $2\sigma_\ell$ for all $s \in \mathbb{C}$. This finishes the proof. □

Remark 4.11 A more involved proof (see [1, Theorem 7.3]) shows that in fact $\|G_e\|_{H^\infty} = 2\sigma_\ell$ holds. □

This leads to the main theorem on the error of balanced truncation.

Theorem 4.12 Consider a system (1.1), which is a minimal realisation with Hurwitz matrix A and pairwise distinct Hankel singular values $\sigma_1 > \sigma_2 > \sigma_\ell > 0$ with multiplicities $n_1, \dots, n_\ell$. Consider the reduced system (4.5) with $r = n_1 + \dots + n_q$ for some $q \in \{1, \dots, \ell - 1\}$. Then (4.5) has the pairwise distinct Hankel singular values $\sigma_1 > \sigma_2 > \sigma_q > 0$ with multiplicities $n_1, \dots, n_q$ and the transfer functions G of (1.1) and G_r of (4.5) satisfy

$$\|G - G_r\|_{H^\infty} \leq 2\sum_{k=q+1}^{\ell} \sigma_k.$$

Proof: The statement about the Hankel singular values of (4.5) follows directly from the construction of (4.5). Now by G_r^k we denote the transfer function of the reduced system (4.5) with $r = n_1 + \ldots + n_k$. This implies that $G = G_r^\ell$ and $G_r = G_r^q$. Moreover, denoting the corresponding matrices by $(A_r^k, B_r^k, C_r^k, D_r^k)$, the construction of the reduced system implies that $(A_r^k, B_r^k, C_r^k, D_r^k)$ is the reduced system obtained from $(A_r^{k+1}, B_r^{k+1}, C_r^{k+1}, D_r^{k+1})$ by removing the singular value σ_{k+1}, as in the setting of Lemma 4.10. Hence, this lemma applies to these systems. Using the triangle inequality and applying Lemma 4.10 to the terms in the resulting sum we can thus estimate

$$\begin{aligned} \|G - G_r\|_{H^\infty} &= \|G_r^\ell - G_r^q\|_{H^\infty} \\ &= \|G_r^\ell - G_r^{\ell-1} + G_r^{\ell-1} - \ldots + G_r^{q+1} - G_r^q\|_{H^\infty} \\ &\le \sum_{k=q+1}^{\ell} \|G_r^k - G_r^{k-1}\|_{H^\infty} \le \sum_{k=q+1}^{\ell} 2\sigma_k. \end{aligned}$$

This shows the claim. □

Remark 4.13 It is also possible to prove the lower bound $\|G - G_r\|_{H^\infty} \ge \sigma_{r+1}$, see [1, Lemma 8.14]. □

For the L^∞ error of the output one can use the inequality

$$\|y - y_r\|_{L^\infty} \le \|G - G_r\|_{H^2} \|u\|_{L^2},$$

which is proved in [1, Korollar 7.31]. The H^2-norm of a transfer function can be computed via

$$\|G\|_{H^2} = \operatorname{trace}(B^T Q B) = \operatorname{trace}(C P C^T).$$

While balanced truncation does not guarantee that the H^2-Norm of the difference of the transfer function is small, we typically obtain a small value of this norm, which is, moreover, easily verified after G_r is computed.

4.6 Numerical implementation

For most of the steps in Algorithm 4.7 there exist highly efficient standard algorithms in almost every scientific programming environment like MATLAB or Python. While the Choleski factorisation and the SVD are standard tasks in numerical linear algebra, the solution of the Lyapunov equation in Step (1) is a somewhat more specific problem. Here we briefly sketch the main ideas behind two popular algorithms for this task for the Lyapunov equation

$$AX + XA^T = F. \tag{4.13}$$

Here we only consider symmetric F, which implies that the solution X is also symmetric.

First of all, it is not too difficult to rewrite a Lyapunov equation in vector form such that standard linear solvers can be used for its solution. However, this is not a very efficient method, because it leads to a system of linear equations with n^2 unknowns (if we use that X is symmetric, we still have $n(n+1)/2 = (n^2+n)/2$ unknowns). Thus, if n is large, then this becomes a huge system of equations. Given that balanced truncation (and, for that matter, any model order reduction technique) is particularly needed when n is very large, it is of utmost importance that we can solve Lyapunov equations in very high dimensions.

The Bartels-Stewart Algorithm

A standard algorithm for this task, which is also implemented in MATLAB and Python, is the Bartels-Stewart Algorithm from 1972. This algorithm is available in the control systems toolbox of MATLAB under the name `lyap` and in Python under the name `scipy.linalg.solve_lyapunov`.

In this algorithm the matrix A in (4.13) is first transformed into real Schur form (i.e., in upper diagonal or quasi-upper diagonal form) by means of QR-transformations. Then the transformed A, denoted by $R = Q^T AQ$, is of the form

$$R = \begin{pmatrix} R_{11} & R_{12} & \cdots & R_{1\ell} \\ & R_{22} & \cdots & R_{2\ell} \\ & & \ddots & \vdots \\ & & & R_{\ell\ell} \end{pmatrix} = \begin{pmatrix} R_1 & R_2 \\ & R_{\ell\ell} \end{pmatrix},$$

where the diagonal blocks R_{jj} are one or two dimensional, depending on whether the corresponding eigenvalue is real or complex. Decomposing the transformed unknown $\widetilde{X} = Q^T XQ$ and the transformed right hand side $\widetilde{F} = Q^T FQ$ in (4.13) accordingly, and using the fact that X and F and thus $\widetilde{X}$ and $\widetilde{F}$ are symmetric, one then writes the equation $R\widetilde{X} + \widetilde{X}R^T = \widetilde{F}$ as

$$\begin{pmatrix} R_1 & R_2 \\ & R_{\ell\ell} \end{pmatrix}\begin{pmatrix} X_1 & X_2 \\ X_2^T & X_3 \end{pmatrix} + \begin{pmatrix} X_1 & X_2 \\ X_2^T & X_3 \end{pmatrix}\begin{pmatrix} R_1^T & \\ R_2^T & R_{\ell\ell}^T \end{pmatrix} = \begin{pmatrix} F_1 & F_2 \\ F_2^T & F_3 \end{pmatrix}.$$

From this equation X_3 can be computed easily, either by solving a scalar equation or by solving a 3×3 system of linear equations. Once X_3 is known, X_2 can be computed via backward substitution and the solution of small systems of linear equations (for details see [1, Section 8.4.1]). When this is done, one proceeds with the equation

$$R_1 X_1 + X_1 R_1^T = F_1,$$

which can be treated in the same way, leading to an induction in which all elements of $\widetilde{X}$ are eventually computed. After retransformation, $X = Q\widetilde{X}Q^T$ yields the result.

A variant of the Bartels-Stewart Algorithm is the Hammarling-Algorithm. This algorithm does not compute X but directly computes the Cholesky factor Y in the Cholesky factorisation $X = YY^T$. This means that this algorithm covers Steps (1) and (2) of Algorithm 4.7 in one step. Clearly, this is much more efficient. In the control systems toolbox of MATLAB, this algorithm is available under the name `lyapchol`.

According to [1, Section 8.4], both the Bartels-Stewart Algorithm and the Hammarling-Algorithm have a computational effort of order $O(n^3)$ and are applicable until around $n \approx 10\,000$.

The ADI method

For even higher dimensions, the so-called ADI algorithm is more appropriate (ADI method = Alternating Direction Implicit method). To derive this algorithm, equation (4.13) is rewritten as

$$AX = -XA^T + F. \tag{4.14}$$

A very simple idea would now be to perform the iteration

$$AX_{j+1} := -X_j A^T + F.$$

This iteration, however, does not guarantee the symmetry of X_j. This can be fixed by performing a so-called ADI iteration (ADI = **A**lternating **D**irection **I**mplicit). Before we define what this is, we first make a modification to equation (4.14), which will yield an acceleration of the convergence of the method. We add pX, p scalar, on both sides of (4.14), leading to

$$(A + p\mathrm{Id})X = -X(A^T - p\mathrm{Id}) + F.$$

Since $X = X^T$, we can rewrite this equation to

$$(A + p\mathrm{Id})X = -X^T(A^T - p\mathrm{Id}) + F.$$

Now the ADI iteration uses these two equations alternatingly (hence the "A" in ADI) in the following way: Starting from an initial guess X_0, we compute

$$\begin{aligned} (A + p_j\mathrm{Id})X_{j+1/2} &:= -X_j(A^T - p_j\mathrm{Id}) + F \\ (A + p_j\mathrm{Id})X_{j+1} &:= -X_{j+1/2}^T(A^T - p_j\mathrm{Id}) + F. \end{aligned}$$

There are systematic ways to compute factors p_j for which this iteration converges quickly.

With a little bit of computation one verifies that the matrices X_j satisfy

$$\begin{aligned} X_{j+1} = {} & 2p_j(A + p_j\mathrm{Id})^{-1}F(A^T + p_j\mathrm{Id})^{-1} \\ & + (A + p_j\mathrm{Id})^{-1}(A - p_j\mathrm{Id})X_j^T(A^T - p_j\mathrm{Id})(A^T + p_j\mathrm{Id})^{-1}. \end{aligned} \tag{4.15}$$

From this one can conclude that the matrices X_j are symmetric if X_0 and F are (although the $X_{j+1/2}$ are not necessarily symmetric).

Clearly, each iteration requires the solution of two systems of linear equations, because $X_{j+1/2}$ and X_{j+1} are implicitly defined (hence the "I" in ADI). In order to make this more efficient, the matrix A can first be transformed into real Schur form (as in the Bartels-Stewart Algorithm). However, even with this transformation the algorithm in this basic form is computationally not more efficient than Bartels-Stewart. The trick to make it efficient lies in the observation that (4.15) implies that the rank of X_j satisfies

$$\mathrm{rank}(X_{j+1}) \le \mathrm{rank}(F) + \mathrm{rank}(X_j).$$

Abbreviating $r_F := \mathrm{rank}(F)$, with the typical choice $X_0 = 0$ this implies $\mathrm{rank}(X_j) \le jr_F$. Given that the rank of F in the Lyapunov equations appearing in balanced truncation is bounded by the dimension of the input and the output, which are typically quite small, this means that X_j is a low-rank matrix as long as j does not become very large. It can be written in the form

$$X_j = Z_j Z_j^T,$$

with $Z_j \in \mathbb{R}^{n \times jr_F}$.

The ADI iteration can now be rewritten as an iteration for the factors Z_j if we assume that $F = -HH^T$, which is the case in balanced truncation. Inserting $X_j = Z_j Z_j^T$ into (4.15) yields

$$\begin{aligned} Z_{j+1}Z_{j+1}^T &= -2p_j(A+p_j\mathrm{Id})^{-1}HH^T(A^T+p_j\mathrm{Id})^{-1} \\ &\quad + (A+p_j\mathrm{Id})^{-1}(A-p_j\mathrm{Id})Z_jZ_j^T(A^T-p_j\mathrm{Id})(A^T+p_j\mathrm{Id})^{-1} \\ &= \Big(\sqrt{-2p_j}(A+p_j\mathrm{Id})^{-1}H\Big)\Big(\sqrt{-2p_j}(A+p_j\mathrm{Id})^{-1}H\Big)^T \\ &\quad + \Big((A+p_j\mathrm{Id})^{-1}(A-p_j\mathrm{Id})Z_j\Big)\Big((A+p_j\mathrm{Id})^{-1}(A-p_j\mathrm{Id})Z_j\Big)^T, \end{aligned}$$

implying that

$$Z_{j+1} = \Big(\sqrt{-2p_j}(A+p_j\mathrm{Id})^{-1}H \quad (A+p_j\mathrm{Id})^{-1}(A-p_j\mathrm{Id})Z_j\Big).$$

Note that this procedure requires p_j to be negative in order to avoid the use of complex arithmetic. With a clever reformulation, it can be achieved that in each iteration only r_F columns in Z_{j+1} need to be computed, while the rest can be copied from Z_j. Thus, while the iteration still requires the solution of linear equation systems, the number of unknowns is drastically reduced.

Moreover, Step (2) of Algorithm (4.7) becomes obsolete. Even though the product $Z_jZ_j^T$ is not a Choleski factorisation of X_j, it can be used in place of Z resulting from Step (2), because in the derivation of the algorithm only the identity $X = ZZ^T$ is important, but not necessarily that Z results from a Choleski factorisation.

This algorithm is implemented in the `pymor` package for model reduction in Python.

4.7 Approaches for non-Hurwitz A

One of the fundamental assumptions on our system (1.1) is that the matrix A is Hurwitz. Indeed, this property was crucial for the proof of Theorem 4.1, which provides the basis for computing the Gramians by solving Lyapunov equations.

However, many real-world problems and also many mathematically interesting problems—such as the inverted pendulum from Example 2.1—have a non-Hurwitz A-matrix. Here we briefly describe three approaches that can be used in this case.

Decomposition

For some models it may be possible, typically after a suitable coordinate transformation, to guarantee that the matrix can be decomposed into the form

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

with $A_{11} \in \mathbb{R}^{q\times q}$ for a small (compared to the overall dimension n) index $q \in \mathbb{N}$ and A_{22} being Hurwitz. We then define

$$x = \begin{pmatrix} x^u \\ x^s \end{pmatrix} \quad \text{with} \quad x^u = \begin{pmatrix} x_1 \\ \vdots \\ x_q \end{pmatrix} \quad \text{and} \quad x^s = \begin{pmatrix} x_{q+1} \\ \vdots \\ x_n \end{pmatrix}$$

and decompose the overall system according to this splitting of x as

$$\dot{x} = \begin{pmatrix} \dot{x}^u \\ \dot{x}^s \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} x^u \\ x^s \end{pmatrix} + \begin{pmatrix} B_1 \\ B_2 \end{pmatrix} u$$

and (setting $D = 0$ for simplicity of exposition)

$$y = Cx = (C_1\ C_2)x = C_1 x^u + C_2 x^s.$$

We can write this as two individual systems as follows

$$\begin{aligned} \dot{x}^u &= A_{11}x^u \quad + A_{12}x^s + B_1 u \\ y^u &= C_1 x^u \end{aligned}$$

$$\begin{aligned} \dot{x}^s &= A_{22}x^s \quad + A_{21}x^u + B_2 u \\ y^s &= C_2 x^s. \end{aligned}$$

Our goal is now to reduce the Hurwitz subsystem with state x^s. When doing this, we need, however, to take into account that the terms $A_{21}x^u$ and $A_{12}x^s$ are taken into account in the model reduction, i.e., that these terms remain approximately the same after the model reduction. To this end, we define

$$C^s = \begin{pmatrix} C_2 \\ A_{12} \end{pmatrix}, \quad B^s = \begin{pmatrix} B_2\ A_{21} \end{pmatrix}$$

and a new output and input

$$\hat{y}^s = C^s x^s = \begin{pmatrix} y^s \\ A_{12}x^s \end{pmatrix}, \quad \hat{u}^s = \begin{pmatrix} u \\ x^2 \end{pmatrix},$$

such that we can write

$$\begin{aligned} \dot{x}^s &= A_{22}x^s \quad + B^s \hat{u}^s \\ \hat{y}^s &= C^s x^s. \end{aligned}$$

Now we can apply balanced truncation to this system and then re-assemble the full system by combining the reduced model with the x^u-system.

Pre-stabilization

The goal of a model reduction may be the application of optimal control, for which it may be needed to reduce the dimension in order to arrive at a model for which the necessary computations can be performed (e.g., the solution of a Riccati equation). Another application could be the computation of a complicated controller (e.g., for trajectory tracking or robustification against perturbations), which may also only be possible for a sufficiently small model.

In contrast to these tasks, the computation of a merely stabilising controller may be much simpler and is often also feasible for very high-dimensional models. More concretely, assume that we have a high-dimensional system (1.1) and a stabilising controller F. For simplicity of exposition we assume that a state feedback can be implemented. In order to be able to control the system despite the fact that the controller is used, we set the control to $u = Fx + \hat{u}$. This leads to the closed loop system

$$\dot{x} = Ax + BFx + B\hat{u} = (A + BF)x + B\hat{u}.$$

The new input $\hat{u}$ can then be used to compute an optimal controller, a tracking controller etc. Since this system now has a Hurwitz matrix $A + BF$ in place of A, it can now be reduced by balanced truncation.

Shifting of the spectrum

A simple way of turning a non-Hurwitz matrix into a Hurwitz matrix is by subtracting $\lambda_0\mathrm{Id}$, where $\lambda_0 > \mathrm{Re}\lambda$ for all eigenvalues λ of A. Then the model reduction can be performed for the matrix

$$\widehat{A} = A - \lambda_0\mathrm{Id},$$

which has the shifted eigenvalues $\lambda - \lambda_0$. The resulting matrix $\widehat{A}_r$ can be re-transformed via

$$A_r = \widehat{A}_r + \lambda_0\mathrm{Id}.$$

This procedure works quite well in practice, but has the severe limitation that there is usually no guarantee that a stabilizing controller for the reduced system will also stabilize the full system.

LQG-Balanced Truncation

When we look at the derivation of the balanced truncation algorithm, we see that once P and Q are computed, the rest of the algorithm can be performed without knowing where these matrices came from. This means that we can define variants of the balanced truncation algorithm by choosing different Gramians P and Q. A variant that is suitable for non-Hurwitz A is to choose P and Q as the solutions of the algebraic Riccati equations

$$\begin{aligned} AP + PA^T - PC^TCP + BB^T &= 0 \\ A^TQ + QA - QBB^TQ + C^TC &= 0. \end{aligned}$$

The values x^TPx and x^TQx then characterise how "expensive" it is to reach a point x with a trajectory converging to 0 as $t \to -\infty$ and, respectively, how expensive it is to control x

to 0 for $t \to \infty$, when an optimal control is used (see Chapters 6 and 7 in [4] for details). Here the cost of the trajectory is measured by

$$\int_{-\infty}^{0} \|Cx(t, x_0, u)\|_2^2 + \|u(t)\|_2^2 dt$$

and

$$\int_{0}^{\infty} \|Cx(t, x_0, u)\|_2^2 + \|u(t)\|_2^2 dt,$$

respectively, i.e., it integrates the squared norm of the input and the output along the respective trajectories.

A state x for which these two values are small contributes only very little to this cost, hence the intuitive interpretation of these quantities is: when the task is to minimise this cost, then states with small cost can be neglegted. Beyond this interpretation, one can show that if P and Q are used in balanced truncation, then the resulting transfer functions of the reduced and the full model are close in the so-called gap metric. Hence, this approach is not only intuitively but also rigorously justified.

The main problem with this approach is that the solution of a Riccati equation is in general quite costly and definitely much more complicated than the solution of a Lyapunov equation. This means that it is in general difficult to use this approach for a very large system.

4.8 Other model reduction techniques

In this final section we briefly describe a selection of other model reduction techniques.

Modal truncation

Modal truncation is a (very much) simplified version of balanced truncation. It relies on the eigenvalues of A and reduced the system by projection to the eigenspaces of A that correspond to the eigenvalues with largest real parts. One can show that when the eigenvalues $\lambda_{r+1}, \dots, \lambda_n$ are truncated, then the estimate

$$\|G - G_r\|_{H^\infty} \le \|C_2\|_2 \|B_2\|_2 \max_{\lambda \in \{\lambda_{r+1}, \dots, \lambda_n\}} \frac{1}{|\mathrm{Re}(\lambda)|}$$

holds, where B_2 and C_2 are transformed versions of B and C using the transformation that brings A into diagonal form (assuming that this is possible).

Proper orthogonal decomposition

Proper orthogonal decomposition (POD) can be seen as a nonlinear generalisation of singular value decomposition-based model reduction. The problem with nonlinear control systems is, that the right hand side is not described by a matrix (or by matrices) and thus there is not a canonical object that can be used for computing singular values.

In order to generate a substitute for this matrix, one creates a large matrix Ψ containing many (simulated) solutions — the rows of Ψ — evaluated at discrete time instants — the column of Ψ. The hope is that then one can express any other solution at least approximately as a linear combination of these solutions, i.e., as the image of the linear map represented by Ψ. This is very often indeed the case, if the solutions contained in Ψ are chosen appropriately. Then, the SVD-based approximation of Ψ by a low-rank matrix as in Section 3.2 yields a low order model. Of course, this methods can be refined in many ways.

A particular challenge for this method when addressing control problems is that a priori one does not know which are the "good" control functions that need to be considered when computing the solutions contained in Ψ.

Moment matching

A class of model reduction techniques that is completely different from eigenvalue or singular value-based approaches are interpolatory techniques. As the name already suggests, these methods rely on interpolation methods, which are applied to the transfer function G. More precisely, given k nodes $s_1, \ldots, s_k \in \mathbb{C}$ and corresponding degrees $p_j \in \mathbb{N}$, the goal is to find a reduced order system (A_r, B_r, C_r, D_r) such that the resulting transfer function $\widetilde{G}$ satisfies the *moment matching* conditions

$$G_r^{(p)}(s_j) = G^{(p)}(s_j)$$

hold for all $j = 1, \ldots, k$ and $p = 0, \ldots, p_j$, where $G^{(p)}$ denotes the p-th derivative of G and $G^{(0)} = G$.

Since transfer functions are rational (matrix valued) functions, the algorithmic realisation of this idea needs techniques from rational interpolation, also known as Padé-Approximation. Efficient algorithms directly yield the matrics A_r, B_r and C_r rather than only the transfer function G_r. In contrast to balanced truncation (or modal truncation), for this method one can obtain an error bound for the H^2 norm $\|G - G_r\|_{H^2}$.

Chapter 5

Data driven model generation and its use in model predictive control

August 9, 2021

In the first part of this chapter we will explain a purely data-driven approach to generate a model for a nonlinear control system. We will show that this model can be rewritten in the form (1.2), however, this requires the solution of an in general underdetermined system of linear equations, which one would like to avoid. In the second part we will thus explain how the model in its original form can be efficiently used in a model predictive control algorithm.

5.1 Assumptions

There are many different ways to derive models from data. A very common approach is that some physical insight into the controlled process is available, which allows to build a model, but with unknown parameters. The measured data is then used in order to identify the values in the model. Here we describe an approach that works entirely without any a priori knowledge on the process and derives a model entirely based on the measured output data of a discrete time control system. The approach relies on a theoretical result by Willems, Rapisarda, Markovsky and De Moor published in [6]. We present it here using the notation from the article [2] by Berberich, Köhler, Müller, and Allgöwer.

The basic assumption in this chapter is that the unknown model of the process to be controlled is of the form (1.2). We note that every continuous-time system of the form (1.1) satisfies this assumption if we proceed as follows: we fix a sampling time $T > 0$ and the sampling instants $t_k = kT$, $k \in \mathbb{N}$. Then we consider control functions $u \in \mathcal{U}$ such that $u|_{[t_k,t_{k+1}]}$ is constant for each $k \in \mathbb{N}$. Then, any solution $x(t)$ of (1.1) satisfies $x_{dt}(k) = x(t_k)$ for the solution x_{dt} of (1.2) with

$$A_{dt} = e^{AT}, \quad B_{dt} = \int_0^T e^{A(T-t)} B dt.$$

Hence, although the approach we present is developed for discrete-time systems, it is readily applicable in continuous time if we restrict ourselves to piecewise constant control inputs.

The idea now is as follows: we run the process that we want to model over the (discrete) time interval $k = 0, \dots, N-1$ with a control sequence $u = (u(0), \dots, u(N-1))$ and measure the output $y(0), \dots, y(N-1)$. Formally:

Definition 5.1 Given a transfer function G, we say that a pair of sequences $(u, y) = (u(k), y(k))_{k=0,\dots,N-1}$ is an *input-output trajectory* of G, if there exists a minimal realisation (1.2) of G and an initial value x_0, such that the corresponding solution satisfies

$$y(k) = Cx(k, x_0, u) + Du(k) \quad \text{for} \quad k = 0, \dots, N-1.$$

□

We note that minimal realisations in discrete time are defined completely analogously as in continuous time.

From such an input-output trajectory we then want to derive a model that completely describes the process. We note that with this approach we only measure the input-output behavior of the model, hence we will not be able to determine a state x of the system that represents physical (or economic, chemical, ...) quantities. The state that we construct here will be a purely auxiliary object with the only purpose to construct a system that exactly reproduces the measures input-output behavior.

The sequences under consideration can be written in a particular matrix form according to the following definition.

Definition 5.2 Consider a sequence $z = (z(k))_{k=0,\dots,N-1}$ with $z(k) \in \mathbb{R}^p$. Then for $L \in \mathbb{N}$ with $L \leq N$ we define the *Hankel matrix*

$$H_L(z) := \begin{pmatrix} z(0) & z(1) & \cdots & z(N-L) \\ z(1) & z(2) & \cdots & z(N-L+1) \\ \vdots & \vdots & \ddots & \vdots \\ z(L-1) & z(L) & \cdots & z(N-1) \end{pmatrix} \in \mathbb{R}^{pL \times (N-L+1)}.$$

Moreover, for $0 \leq a \leq b \leq N-1$ we define

$$z_{[a,b]} := \begin{pmatrix} z(a) \\ \vdots \\ z(b) \end{pmatrix} \in \mathbb{R}^{p(b-a+1)}.$$

□

With this notation we can in particular write the Hankel matrix in the compact form

$$H_L(z) = \left(z_{[0,L-1]} \; z_{[1,L]} \cdots z_{[N-L,N-1]} \right).$$

In order to be able to derive a model from the measured values, the control input used for generating the data needs to satisfy the following assumption.

Definition 5.3 We say that a control sequence $(u(k))_{k=0,\dots,N-1}$ with $u(k) \in \mathbb{R}^m$ is *persistently exciting* of order L if

$$\operatorname{rank}(H_L(u)) = mL.$$

□

Since $H_L(u) \in \mathbb{R}^{mL \times (N-L+1)}$, this means that the matrix has maximal row rank. We note that this is only possible if $N - L + 1 \geq mL$, i.e., if $N \geq (m+1)L - 1$.

5.2 Characterisation of the solutions

The following main theorem of this first part of this chapter shows that with the data from a single input-output trajectory with persistently exciting u we can characterise all solutions of the control systems.

Theorem 5.4 Suppose the pair $(u^d, y^d) = (u^d(k), y^d(k))_{k=0,\dots,N-1}$ is an input-output trajectory of a transfer function G and $u^d = (u^d(k))_{k=0,\dots,N-1}$ is persistently exciting of order $L+n$, where n is the dimension of the minimal realisation of G. Then, any pair of sequences $(u(k), y(k))_{k=0,\dots,L-1}$ is an input-output trajectory of G if and only if there exists $\alpha \in \mathbb{R}^{N-L+1}$ such that

$$\begin{pmatrix} H_L(u^d) \\ H_L(y^d) \end{pmatrix} \alpha = \begin{pmatrix} u \\ y \end{pmatrix} \tag{5.1}$$

for

$$u = \begin{pmatrix} u(0) \\ \vdots \\ u(L-1) \end{pmatrix} \quad \text{and} \quad y = \begin{pmatrix} y(0) \\ \vdots \\ y(L-1) \end{pmatrix}.$$

Proof: We first show that for any $\alpha \in \mathbb{R}^{N-L+1}$ the expression

$$\begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} H_L(u^d) \\ H_L(y^d) \end{pmatrix} \alpha$$

defines an input-output trajectory. To see this, observe that each column $u_{[i,i+L-1]}$ of $H_L(u^d)$ together with the corresponding column $y_{[i,i+L-1]}$ of $H_L(u^d)$ forms an input-output trajectory of length L. By linearity of (1.2), for any two solutions $x(k, x_1, u_1)$, $x(k, x_2, u_2)$ and any $\alpha_1, \alpha_2 \in \mathbb{R}$ the identity

$$\alpha_1 x(k, x_1, u_1) + \alpha_2 x(k, x_2, u_2) = x(k, \alpha_1 x_1 + \alpha_2 x_2, \alpha_1 u_1 + \alpha_2 u_2).$$

This and the definition of an input-output trajectory implies that the linear combinations

$$\hat{u} = H_L(u^d)\alpha = \sum_{i=0}^{N-L} \alpha_i u_{[i,i+L-1]}, \quad \hat{y} = H_L(y^d)\alpha = \sum_{i=0}^{N-L} \alpha_i y_{[i,i+L-1]}$$

again a form an input-output trajectory. This shows the claim.

Next we show that for any input-output trajectory (u, y) there is an α such that (5.1) holds. To do this, since from the first part we already know that

$$\operatorname{im}\begin{pmatrix} H_L(u^d) \\ H_L(y^d) \end{pmatrix}$$

is contained in the space of input-output trajectories of length L, it suffices to prove hat the dimensions of these two spaces coincide.

Let $x_k^d = x(k, x_0^d, u^d)$ be the state trajectory corresponding to the input-output trajectory. Since the control system generating this trajectory is minimal, it is in particular controllable. Using this property, one can show that the matrix

$$\widetilde{H} = \begin{pmatrix} u^d(0) & u^d(1) & \cdots & u^d(N-L) \\ u^d(1) & u^d(2) & \cdots & u^d(N-L+1) \\ \vdots & \vdots & \ddots & \vdots \\ u^d(L-1) & u^d(L) & \cdots & u^d(N-1) \\ x_0^d & x_1^d & \cdots & x_{N-L}^d \end{pmatrix}$$

has full row rank, i.e., rank $mL+n$. This is proved in Corollary 2 in [6] using sophisticated algebraic techniques, which we will not reproduce here.

Now the columns of $H_L(y^d)$ are of the form

$$\begin{pmatrix} y^d(j) \\ \vdots \\ y^d(j+L-1) \end{pmatrix} = \begin{pmatrix} Cx_j^d + Du^d(j) \\ \vdots \\ Cx_{j+L-1}^d + Du^d(j+L-1) \end{pmatrix}$$

for $j = 0, \dots, N-L-1$. For the entries of these columns, the identity

$$Cx_{j+k}^d = C(A^k x_j^d + A^{k-1}Bu^d(j) + A^{k-2}Bu^d(j+1) + \ldots + A^0 Bu^d(j+k-1))$$

holds. This implies that $H_L(y^d)$ can be written as

$$H_L(y^d) = \begin{pmatrix} D & 0 & \cdots & \cdots & 0 & C \\ CB & D & 0 & \cdots & 0 & CA \\ CAB & CB & \ddots & \ddots & 0 & CA^2 \\ \vdots & & \ddots & \ddots & 0 & \vdots \\ CA^{L-2}B & CA^{L-2}B & \cdots & CB & D & CA^{L-1} \end{pmatrix} \widetilde{H},$$

which in turn implies

$$\begin{pmatrix} H_L(u^d) \\ H_L(y^d) \end{pmatrix} = \underbrace{\begin{pmatrix} \mathrm{Id} & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \mathrm{Id} & 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & \mathrm{Id} & 0 \\ D & 0 & \cdots & \cdots & 0 & C \\ CB & D & 0 & \cdots & 0 & CA \\ CAB & CB & \ddots & \ddots & 0 & CA^2 \\ \vdots & & \ddots & \ddots & 0 & \vdots \\ CA^{L-2}B & CA^{L-3}B & \cdots & CB & D & CA^{L-1} \end{pmatrix}}_{=:M} \widetilde{H}.$$

The matrix M thus satisfies

$$\mathrm{rank} M = mL + \mathrm{rank} \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{L-1} \end{pmatrix},$$

which is precisely the dimension of the space of input-output trajectories of length L. Since $\widetilde{H}$ has full row rank and thus defines a surjective linear map, the dimension of the image of $M\widetilde{H}$ also equals the dimension of the space of input-output trajectories. This finishes the proof. □

5.3 Reformulation as a standard model

Theorem 5.4 provides a characterisation of the sequences, but not yet a model of the form (1.2) that yields the measured y as output. However, using the observability of the minimal realisation (which we have not used in the proof of Theorem 5.4), we can arrive at an explicit model.

To this end, choose L minimal with the property that

$$\mathrm{rank} \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{L-2} \end{pmatrix} = n.$$

Since due to the observability we know that the rank of this matrix equals n for $L-2 = n-1$, such an L must exist. Then we decompose M in the form

$$M = \begin{pmatrix} M_1 \\ M_2 \end{pmatrix}$$

with M_2 consisting of the last l rows of M. Then, similar as above, we can conclude that

$$\operatorname{rank} M_1 = mL + \operatorname{rank}\begin{pmatrix} C \\ CA \\ \vdots \\ CA^{L-2} \end{pmatrix} = mL + n.$$

Since this number coincides with the number of columns in M and M_1, the matrix M_1 has full rank and its rank is identical with the rank of M. This means that the last m rows of M depend linearly on the first rows, implying that $M_2 = ZM_1$ for a suitable matrix Z and thus

$$\begin{pmatrix} H_L(u^d) \\ H_L(y^d) \end{pmatrix} = \begin{pmatrix} M_1\widetilde{H} \\ ZM_1\widetilde{H} \end{pmatrix}.$$

Splitting

$$y = \begin{pmatrix} y(0) \\ \vdots \\ y(L-2) \\ y(L-1) \end{pmatrix} = \begin{pmatrix} y_1 \\ y(L-1) \end{pmatrix}$$

and using (5.1), we then obtain

$$\begin{pmatrix} u \\ y_1 \\ y(L-1) \end{pmatrix} = \begin{pmatrix} H_L(u^d) \\ H_L(y^d) \end{pmatrix}\alpha = \begin{pmatrix} M_1\widetilde{H}\alpha \\ ZM_1\widetilde{H}\alpha \end{pmatrix}.$$

This implies that

$$\begin{pmatrix} u \\ y_1 \end{pmatrix} = M_1\widetilde{H}\alpha \tag{5.2}$$

and thus

$$y(L-1) = ZM_1\widetilde{H}\alpha = Z\begin{pmatrix} u \\ y_1 \end{pmatrix} = Z_1u + Z_2y_1 = Z_2\begin{pmatrix} y(0) \\ \vdots \\ y(L-2) \end{pmatrix} + Z_1u.$$

Now we define the state $\hat{y}$ and control $\hat{u}$ at time k to be

$$\hat{y}(k) = \begin{pmatrix} y(k) \\ \vdots \\ y(k+L-2) \end{pmatrix} \quad \text{and} \quad \hat{u}(k) = \begin{pmatrix} u(k) \\ \vdots \\ u(k+L-1) \end{pmatrix}.$$

Then we obtain

$$\hat{y}(k+1) = \begin{pmatrix} y(k+1) \\ \vdots \\ y(k+L-1) \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & \mathrm{Id} & 0 & \dots & \dots & 0 \\ 0 & 0 & \mathrm{Id} & 0 & \dots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & 0 & \mathrm{Id} & 0 \\ 0 & \dots & \dots & \dots & 0 & \mathrm{Id} \\ & & & Z_2 & & \end{pmatrix}}_{=:\widehat{A}} \underbrace{\begin{pmatrix} y(k) \\ \vdots \\ y(k+L-2) \end{pmatrix}}_{=\hat{y}(k)} + \underbrace{\begin{pmatrix} 0 \\ \vdots \\ Z_1 \end{pmatrix}}_{=:\widehat{B}} \hat{u}(k), \tag{5.3}$$

where all identity matrices Id are $m \times m$ matrices. In conjunction with $C = (\mathrm{Id}\ 0\ \dots\ 0)$, which implies $Cx(k) = y(k)$, this yields the standard description of a linear control system.

It is important to emphasize that one aspect of this control system is not entirely standard. This is because $\hat{u}(k)$ in this description is not a single control vector but a sequence of length L. Particularly, we have that

$$\hat{u}(k) = \begin{pmatrix} u(k) \\ \vdots \\ u(k+L-1) \end{pmatrix} \quad \text{and} \quad \hat{u}(k+1) = \begin{pmatrix} u(k+1) \\ \vdots \\ u(k+L) \end{pmatrix},$$

which implies that the first $L-1$ (vector) entires of $u(k+1)$ are already determined at time k by $\hat{u}(k)$ and only $u(k+L)$ can be chosen freely. At time $k = 0$, the values $\hat{u}(0) = u = (u(0), \dots, u(L-1))^T$ can be chosen freely.

The way of defining the control system shown in this section has the advantage that reconstructing the state $x(k)$ from the output measurements $y(k)$ is almost trivial. It suffices to collect $L-1$ measurements $y(k), \dots, y(k+L-2)$ and stack them together to obtain the state of the system. However, it requires the computation of the implicitly defined matrices Z_1, Z_2, E_1 and E_2. Moreover, the way the control function enters is somewhat unusual.

The minimality of L is not crucial for the construction. We would end up with a similar system if we chose a larger L. In particular, $L = n+1$ would always satisfy the rank requirement.

If we look at system (5.3) componentwise, then we see that it computes $y(k+L-1)$ from $y(k), \dots, y(k+L-2)$ and $u(k), \dots, u(k+L-1)$. If we carry out the construction with a larger $L' > L$ but the same matrix M_1, then the matrix M_2 has $(L'-L+1)m$ rows. Then system (5.3) changes to

$$\begin{pmatrix} y(k+1) \\ \vdots \\ y(k+L-1) \\ y(k+L) \\ \vdots \\ y(k+L'-1) \end{pmatrix} = \begin{pmatrix} 0 & \mathrm{Id} & 0 & \dots & \dots & 0 \\ 0 & 0 & \mathrm{Id} & 0 & \dots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & 0 & \mathrm{Id} & 0 \\ 0 & \dots & \dots & \dots & 0 & \mathrm{Id} \\ & & & Z_2' & & \end{pmatrix} \begin{pmatrix} y(k) \\ \vdots \\ y(k+L-2) \end{pmatrix} + \begin{pmatrix} 0 \\ \vdots \\ Z_1' \end{pmatrix} \begin{pmatrix} u(k) \\ \vdots \\ u(k+L'-1) \end{pmatrix}, \tag{5.4}$$

With this equation, we can compute $y(k+L-1), \dots, y(k+L'-1)$ from $y(k), \dots, y(k+L-2)$ and $u(k), \dots, u(k+L'-1)$ in one shot, without having to iterate (5.3). As before, for $k \geq 1$ the control values $u(k), \dots, u(k+L-2)$ are fixed from the previous time step. Note that the persistency of excitation requirement on u^d and y^d becomes more demanding the larger L' becomes. Particularly, in order to compute Z_1' and Z_2', the larger L' becomes, the longer the sequences (u^d) and (y^d) must be.

5.4 Data-driven model predictive control

In this section we want to use the data-driven approach from the previous sections in model predictive control (MPC). We recall from [4], that in MPC in each time step an optimal control problem over a time horizon N is solved, and the first element is applied to the control system for the next time step. When combining this with the data-driven approach, one could in principle use the standard model from the previous section. However, the explicit derivation of this model can actually be avoided. In this section we describe the approach from the paper [2], in which the explicit model is not needed. Here we first assume that all measured data is exact, in order to explain the basic methodology. However, as this is a highly idealised assumption, in the next section we explain an extension of the approach that can deal with noisy data. For simplicity, the following exposition is for the case of scalar y and u, i.e., $m = l = 1$.

Let us assume that $(u^d, y^d) = (u^d(k), y^d(k))_{k=0,\ldots,N-1}$ is an input-output trajectory of a transfer function G and $u^d = (u^d(k))_{k=0,\ldots,N-1}$ is persistently exciting of order $L + n$, where n is the dimension of the minimal realisation of G. Then, in view of Theorem 5.4, we can write a cost functional for the input-output trajectories of the corresponding system as

$$J_T(y_0(\cdot), u_0(\cdot), \bar{y}(\cdot), \bar{u}(\cdot), \alpha) = \sum_{k=0}^{T-1} \ell(\bar{y}(k), \bar{u}(k))$$

subject to the conditions

- $y_0(k)$ is defined for $k = -n, \ldots, -1$
- $u_0(k)$ is defined for $k = -n, \ldots, -1$,
- $\bar{y}(k)$ and $\bar{u}(k)$ are defined for $k = -n, \ldots, T-1$ and satisfy $\bar{y}_{[-n,-1]} = y_{0[-n,-1]}$ as well as $\bar{u}_{[-n,-1]} = u_{0[-n,-1]}$.
- $\alpha \in \mathbb{R}^{N-T-n+1}$,

and

$$\begin{pmatrix} H_{T+n}(u^d) \\ H_{T+n}(y^d) \end{pmatrix} \alpha = \begin{pmatrix} \bar{u} \\ \bar{y} \end{pmatrix}.$$

The interpretation of the ingredients of this formulation are as follows:

- $T \in \mathbb{N}$ is the optimization horizon of the optimal control problem. This is usually denoted by "N" in MPC, but N already has a different meaning in this chapter as the length of the sequences (u^d, y^d).
- $y_0(\cdot)$ are the past values of the output. In the representation of Section 5.3, y_0 plays the role of the vector $\hat{y}(k)$
- $u_0(\cdot)$ is the control sequence from the previous steps. It represents the part of the control sequence that cannot be chosen freely.
- (u, y) is the input-output trajectory from (5.1) for $\alpha \in \mathbb{R}^{N-T-n+1}$ in the argument of J_T. For past times, the trajectory coincides with (u_0, y_0).

Theorem 5.4 guarantees that the constraints are satisfied exactly by all possible input-output trajectories (u, y) that start in y_0 and are compatible with the previously chosen control values u_0. The setting corresponds to the control system (5.4) with $L = n + 1$ and $L' = T + n$. Since Theorem 5.4 requires persistency of excitation of order $L' + n$, for the optimization problem to be well defined we need a sequence u^d that is persistently exciting of order $T + 2n$.

We only consider a relatively simple MPC scheme, in which the goal of the optimization is to drive the output y to a desired equilibrium value y^s with corresponding control u^s. We say that (y^s, u^s) is an equilibrium of the system, if the sequences $(y^s_T) := (y^s, y^s, \dots, y^s)$, $(u^s_T) := (u^s, u^s, \dots, u^s)$ of length $T \in \mathbb{N}$ form an input-output trajectory for each $T \in \mathbb{N}$.

The cost function is then chosen as

$$\ell(y, u) = \|y - y^s\|_R^2 + \|u - u^s\|_Q^2,$$

where $\|x\|_P = \sqrt{x^T P x}$ for a matrix P of appropriate dimension and R and Q are symmetric and positive definite matrices. In addition, we require constraints $y \in \mathbb{Y}$, $u \in \mathbb{U}$.

In order to guarantee stability of the scheme, we impose an equilibrium terminal contraint, i.e., we optimize over those trajectories that end up in the equilibrium. This is the simplest way to design an MPC scheme with provable stability properties. Inequality terminal constraints and also schemes entirely without terminal constraints are also possible (cf. [4]), but require a more involved stability analysis, which we would like to avoid here for the sake of brevity. Since the initial condition y_0 and the initial controls u_0 are sequences of length n, we also require the equilibrium condition for n consecutive outputs and controls. Overall, this leads to the following optimal control problem.

At time $t \in \mathbb{N}$, consider the past inputs and outputs $u_{MPC}(0), \dots, u_{MPC}(t-1)$ and $y_{MPC}(0), \dots, y_{MPC}(t-1)$ of the MPC closed loop. Set $u_0(k) := u_{MPC}(t+k)$ and $y_0(k) := y_{MPC}(t+k)$ for $k = -n, \dots, -1$. Then solve the optimization problem

$$\min_{\alpha, \bar{u}, \bar{y}} J_T(y_0(\cdot), u_0(\cdot), \bar{y}(\cdot), \bar{u}(\cdot), \alpha) = \sum_{k=0}^{T-1} \ell(\bar{y}(k), \bar{u}(k)) \tag{5.5}$$

with $\alpha \in \mathbb{R}^{N-T-n+1}$, $(\bar{u}(-n), \dots, \bar{u}(T-1)) \in \mathbb{R}^{m(T+n)}$, $(\bar{y}(-n), \dots, \bar{y}(T-1)) \in \mathbb{R}^{l(T+n)}$, subject to the conditions

$$\begin{pmatrix} H_{T+n}(u^d) \\ H_{T+n}(y^d) \end{pmatrix} \alpha = \begin{pmatrix} \bar{u} \\ \bar{y} \end{pmatrix},$$

$$\begin{pmatrix} \bar{u}_{[-n,-1]} \\ \bar{y}_{[-n,-1]} \end{pmatrix} = \begin{pmatrix} u_{0[-n,-1]} \\ y_{0[-n,-1]} \end{pmatrix},$$

$$\begin{pmatrix} \bar{u}_{[T-n,T-1]} \\ \bar{y}_{[T-n,T-1]} \end{pmatrix} = \begin{pmatrix} u^s_n \\ y^s_n \end{pmatrix}, \tag{5.6}$$

$\bar{u}(k) \in U$ and $\bar{y}(k) \in \mathbb{Y}$ for all $k = 0, \dots, T-1$.

As usual, the first element of the optimal control sequence $\bar{u}^\star$ is used as control value for the next step. This means that we set

$$u_{MPC}(t) := \bar{u}^\star(0) \quad \text{and} \quad y_{MPC}(t) := \bar{y}^\star(0). \tag{5.7}$$

We note that we need to start the process to be controlled at time $t = -n$, in order to enable the initialisation of $u_0(0), \ldots, u_0(n-1)$. This requires to define a rule for initialising the values $u_{MPC}(-n), \ldots, u_{MPC}(-1)$. A standard choice would be to set these values to u^s.

From now on, we fix an arbitrary minimal realisation of the control system with state $x \in \mathbb{R}^n$. Then there is a unique state trajectory $x_{MPC}(\cdot)$ corresponding to $u_{MPC}(\cdot)$ and $y_{MPC}(\cdot)$[1].

We now consider the situation (and notation) of Definition 5.1 for this minimal realisation. The following lemma gives a property of the map mapping x_0 and $u(\cdot)$ to $y(\cdot)$ and $u(\cdot)$.

Lemma 5.5 Let $p \geq n-1$ and let $(u(\cdot), y(\cdot))$ be the input-output trajectory of length p corresponding to initial value $x_0 \in \mathbb{R}^n$ and $u(\cdot)$ according to Definition 5.1. Then the map

$$E(x_0, u(0), \ldots u(p)) := (y(0), \ldots, y(p), u(0), \ldots, u(p))$$

is linear and injective. Particularly, there is a constant $\eta > 0$ with

$$\|y(0), \ldots, y(p), u(0), \ldots, u(p)\| \geq \eta \|x_0, u(0), \ldots u(p)\|.$$

Proof: Linearity of E is immediate from the fact that the solution $x(k; x_0, u)$ of a linear control system depends linearly on $(x_0, u(\cdot))$. We abbreviate $v = (x_0, u(0), \ldots u(p))$, $w = (y(0), \ldots, y(p), u(0), \ldots, u(p))$, implying $Ev = w$. Now observe that injectivity, i.e., the fact that $Ev \neq 0$ for all $v \neq 0$, implies that $\eta := \min_{\|v\|=1} \|E(v)\| \neq 0$. For $v \neq 0$ this implies

$$\|Ev\| = \|v\| \left\| E\frac{v}{\|v\|} \right\| \geq \|v\|\eta$$

and thus the claimed inequality. For $v = 0$, this inequality is immediately clear.

Injectivity now follows from the fact that by observability $y(0) = \ldots = y(p) = 0$ and $u(0) = \ldots = u(p) = 0$ imply $x_0 = 0$. Hence, if $v \neq 0$ then either $u(k) \neq 0$ for some $k \in \{0, \ldots, p\}$ and thus $Ev \neq 0$, or $x_0 \neq 0$, implying $y(k) \neq 0$ for some $k \in \{0, \ldots, p\}$ and thus again $Ev \neq 0$. □

This lemma has two important consequences. First, it shows that for each input-output trajectory (u, y) of length $\geq n$ there is a unique initial state x_0 for any minimal realization of the control system. This also implies that for each pair u_0, y_0 in the argument of J_T there is a unique x_0 such that the solution $x(\cdot)$ corresponding to u_0 and y_0 (which is defined for $k = -n, \ldots, 0$) satisfies $x(0) = x_0$. Conversely, from $x(0)$ and $\bar{u}(\cdot)$ the output

[1] In the Mathematical Control Theory lecture [4], we have always expressed $u_{MPC}(t)$ as $\mu_T(x_{MPC}(t))$, in order to emphasise its feedback character. Since the state is only an auxiliary quantity here and not present in the MPC scheme, we use the different notation here.

$\bar{y}(\cdot)$ can be uniquely determined. We may therefore write $J_T(x_0, \bar{y}(\cdot), \bar{u}(\cdot), \alpha)$ instead of $J_T(y_0(\cdot), u_0(\cdot), \bar{y}(\cdot), \bar{u}(\cdot), \alpha)$.

Second, the inequality from the lemma implies that for $T \geq n$ we get

$$\begin{aligned} J_T(x_0, \bar{y}(\cdot), \bar{u}(\cdot), \alpha) &= \sum_{k=0}^{T-1} \ell(\bar{y}(k), \bar{u}(k)) \geq \sum_{k=0}^{n-1} \ell(\bar{y}(k), \bar{u}(k)) \\ &\geq C\|\bar{y}(0) - y^s, \ldots, \bar{y}(n-1) - y^s, \bar{u}(0) - u^s, \ldots, \bar{u}(n-1) - u^s\|^2 \\ &\geq \eta^2 C\|x_0 - x^s\|^2. \end{aligned}$$

Here we used that E maps $v - (x^s, u_p^s)$ to $w - (y_p^s, u_p^s)$, where x^s is the state equilibrium corresponding to u^s, y^s. We abbreviate $C_1 = \eta^2 C$.

We now assume that there is also a similar lower bound. To this end, we first need to make the following definition.

Definition 5.6 The feasible set $\mathbb{X}_T$ for the control system is the set of all initial conditions $x_0 \in \mathbb{R}^n$ for which there is a control sequence u of length T with $u(k) \in \mathbb{U}$, $y(k) = Cx(k, x_0, u) \in \mathbb{Y}$ for all $k = 0, \ldots, T-1$, and (5.6). □

Assumption 5.7 We assume that there is a constant $C_2 > 0$ such that optimal value function

$$V_T(x_0) \coloneqq \min_{\alpha, \bar{u}, \bar{y}} J_T(x, \bar{y}(\cdot), \bar{u}(\cdot), \alpha),$$

where the minimisation is subject to all the constraints defined after (5.5), satisfies the inequality

$$V_T(x_0) \leq C_2\|x_0 - x^s\|^2$$

for all $x_0 \in \mathbb{X}_T$. □

Without any constraints, this inequality can easily be concluded from the stabilisability of the system, which follows from its controllability. However, with the constraints on y and u it needs to be assumed separately.

The following theorem shows the main qualitative properties of the MPC closed loop.

Theorem 5.8 Consider the MPC scheme based on the optimisation problem (5.5), with u^d being persistently exciting of order $T + 2n$. Assume that the optimal value function satisfies Assumption (5.7). Then on the feasible set $\mathbb{X}_T$ the MPC closed loop is recursively feasible, satisfies all constraints, and is exponentially stable.

Proof: Observe that for any trajectory satisfying the constraints, the terminal condition together with observability implies that $x(T-1, x_0, \bar{u}) = x^s$. This implies that by prolonging the control sequence with $\bar{u}(T) = u^s$, we obtain $x(T, x_0, \bar{u}) = x^s$ and thus $(\bar{u}(T), \bar{y}(T)) = (u^s, y^s)$. Hence, the control sequence $\tilde{u}(\cdot) = (\bar{u}(1), \ldots, \bar{u}(T))$ satisfies all the constraints for initial condition $\tilde{x}_0 = x(1, x_0, \bar{u})$. This shows that $x_{MPC}(t) \in \mathbb{X}_T$ if $x_{MPC}(t-1) \in \mathbb{X}_T$ and thus recursive feasibility. Constraint satisfaction of the MPC closed loop is then straightforward.

In order to show asymptotic stability, consider the optimal value function V_T. From the considerations after Lemma 5.5 and Assumption 5.7, V_T has quadratic upper and lower bounds. Moreover, using the control function $\tilde{u}^\star$ from the previous construction with the optimal control $\bar{u}^\star$ in place of $\bar{u}$, we obtain

$$\begin{aligned}
&V_T(x_{MPC}(t+1)) - V_T(x_{MPC}(t))\\
&\quad\le\; J_T(x(1,x_0,\bar{u}^\star),\tilde{u}^\star) - J_T(x_0,\bar{u}^\star)\\
&\quad\le\; \sum_{j=0}^{T-1} \ell(Cx(k+1,x_0,\bar{u}^\star),\bar{u}^\star(k+1)) - \sum_{j=0}^{T-1} \ell(Cx(k,x_0,\bar{u}^\star),\bar{u}^\star(k))\\
&\quad=\; \underbrace{\ell(Cx(T,x_0,\bar{u}^\star),\bar{u}^\star(T))}_{=\ell(x^s,u^s)=0} - \underbrace{\ell(Cx(0,x_0,\bar{u}^\star),\bar{u}^\star(0))}_{=\ell(y_{MPC}(t),u_{MPC}(t))}\\
&\quad= -\ell(y_{MPC}(t),u_{MPC}(t)) \;\le\; 0.
\end{aligned}$$

Thus, $t \mapsto V_T(x_{MPC}(t))$ is non-increasing. By Lemma 5.5 and Assumption 5.7 we moreover obtain

$$\begin{aligned}
V_T(x_{MPC}(t+n)) - V_T(x_{MPC}(t)) \;&\le\; -\sum_{k=t}^{t-n-1} \ell(y_{MPC}(t),u_{MPC}(t))\\
&\le\; -\eta^2 C\|x_{MPC}(t) - x^s\|^2 \;\le\; -C_3 V_T(x_{MPC}(t)).
\end{aligned}$$

By induction this implies that $p \mapsto V_T(x_{MPC}(np))$ decreases exponentially and since V_T is non-increasing along the solution, $t \mapsto V_T(x_{MPC}(t))$ also decreases exponentially fast, i.e., there are $C > 0$ and $\sigma \in (0,1)$ with

$$V_T(x_{MPC}(t)) \le C\sigma^t V_T(x_{MPC}(0)).$$

Then the quadratic bounds on V_T imply that

$$\begin{aligned}
\|x_{MPC}(t) - x^s\| \;&\le\; \sqrt{\frac{1}{C_1}V_T(x_{MPC}(t))} \;\le\; \sqrt{\frac{1}{C_1}C\sigma^t V_T(x_{MPC}(0))}\\
&\le\; \sqrt{\frac{C_2}{C_1}C\sigma^t\|x_{MPC}(0)) - x^s\|^2} \;=\; \sqrt{\frac{C_2}{C_1}C}\sqrt{\sigma}^t\|x_{MPC}(0)) - x^s\|,
\end{aligned}$$

i.e., the claimed exponential stability. □

Besides the fact that the proposed MPC scheme runs entirely on the basis of data, i.e., without using any first principle laws (e.g., from physics), another remarkable feature of this MPC scheme is that it only requires output values for setting up the optimal control problem, but no state information. This means, there is no need to construct an observer in order to estimate the state $x_{MPC}(t)$ from the output values $y_{MPC_{[0,t|}}$.

A potential disadvantage is that in order to start the scheme we first need n measurements and n control values from $t = -n$ to $t = -1$ to start the first optimisation. As already mentioned, the control values $u_{MPC}(-n),\ldots,u_{MPC}(-1)$ need to be determined by some other rule. We note that the choice of these values determines the value $x_{MPC}(0)$, which appears on the right hand side of the exponential stability estimate. A bad choice of these values can lead to a large norm $\|x_{MPC}(0)) - x^s\|$ and thus to a large constant in the exponential stability estimate.

With similar techniques as in the lecture on Mathematical Control Theory [4], one could also derive estimates for the performance

$$\sum_{t=0}^{\infty} \ell(y_{MPC}(t), u_{MPC}(t)),$$

but we will not go into details about this aspect here.

5.5 A robust variant

In practice, the data obtained from measurements of real systems is never absolutely exact, but subject to so-called *measurement noise*. In the approach here, this concerns the entries of the vector y^d (and thus of the matrix $H_{T+n}(y^d)$ in the constraints of the optimal control problem) and the past values y_{MPC}, which in practice are not obtained from simulation as in (5.7) but also from real measurements. In order to reflect this effect in the notation, we denote the noisy values with $\tilde{y}^d$ and $\tilde{y}_{MPC}$. Since the values of $\tilde{y}_{MPC}$ determine the initial values y_0, we also denote these by $\tilde{y}_0$.

The effect of the noisy values on the optimal control problem (5.5) is that the constraints may not be feasible anymore, because $y_{MPC}(\cdot)$ is not compatible with y^d. This implies that the first constraint of the problem, i.e.,

$$\begin{pmatrix} H_{T+n}(u^d) \\ H_{T+n}(\tilde{y}^d) \end{pmatrix} \alpha = \begin{pmatrix} \bar{u} \\ \bar{y} \end{pmatrix}$$

may not be compatible with the second constraint, i.e., with

$$\begin{pmatrix} \bar{u}_{[-n,-1]} \\ \bar{y}_{[-n,-1]} \end{pmatrix} = \begin{pmatrix} u_{0[-n,-1]} \\ \tilde{y}_{0[-n,-1]} \end{pmatrix}.$$

However, since the deviation of the measured values $\tilde{y}$ from the exact values y are typically small, it should still be possible to satisfy these two equations approximately. This idea is the basis for the following "robust" modification of (5.5).

The modifications compared to (5.5) are the following:

- A slack variable σ with the same dimension as $\bar{y}$ is introduced, which allows for a relaxation of the first constraint.
- The slack variable is included in the objective, in order to ensure it remains small.
- The coefficient vector α is also included in the objective, in order to avoid the "overfitting" phenomenon. The penalisation of α involves a parameter $\varepsilon > 0$, which should satisfy $\varepsilon \geq \|y - \tilde{y}\|$ for all measurements used in the computation.
- A "compatibility condition" between σ and α is added to the constraints. This depends on the same ε as the penalisation of α.

At time $t \in \mathbb{N}$, consider the past inputs and outputs $u_{MPC}(0), \dots, u_{MPC}(t-1)$ and $\tilde{y}_{MPC}(0), \dots, \tilde{y}_{MPC}(t-1)$ of the MPC closed loop. Set $u_0(k) := u_{MPC}(t+k)$ and $\tilde{y}_0(k) := \tilde{y}_{MPC}(t+k)$ for $k = -n, \dots, -1$. Then solve the optimization problem

$$\min_{\alpha,\bar{u},\bar{y},\sigma} J_T(\tilde{y}_0(\cdot), u_0(\cdot), \bar{y}(\cdot), \bar{u}(\cdot), \alpha, \sigma) = \sum_{k=0}^{T-1} \left[\ell(\bar{y}(k), \bar{u}(k)) + \lambda_\sigma \|\sigma(k)\|\right] + \lambda_\alpha \varepsilon \|\alpha\| \tag{5.8}$$

with $\alpha \in \mathbb{R}^{N-T-n+1}$, $(\bar{u}(-n), \dots, \bar{u}(T-1)) \in \mathbb{R}^{m(T+n)}$, $(\bar{y}(-n), \dots, \bar{y}(T-1)) \in \mathbb{R}^{l(T+n)}$, subject to the conditions

$$\begin{pmatrix} H_{T+n}(u^d) \\ H_{T+n}(y^d) \end{pmatrix} \alpha = \begin{pmatrix} \bar{u} \\ \bar{y} + \sigma \end{pmatrix},$$

$$\begin{pmatrix} \bar{u}_{[-n,-1]} \\ \bar{y}_{[-n,-1]} \end{pmatrix} = \begin{pmatrix} u_{0[-n,-1]} \\ \tilde{y}_{0[-n,-1]} \end{pmatrix},$$

$$\begin{pmatrix} \bar{u}_{[T-n,T-1]} \\ \bar{y}_{[T-n,T-1]} \end{pmatrix} = \begin{pmatrix} u^s_n \\ y^s_n \end{pmatrix},$$

$$\|\sigma(k)\|_\infty \le \varepsilon(1 + \|\alpha\|_1), \quad k = 0, \dots, T-1$$

$\bar{u}(k) \in U$ and $\bar{y}(k) \in \mathbb{Y}$ for all $k = 0, \dots, T-1$.

For this optimal control problem one can prove that the optimal value function satisfies quadratic bounds, similar to the ones given before and in Assumption 5.7, of the form

$$C_1 \|x_0 - x^s\|^2 \le V_T(x_0) \le C_2 \|x_0 - x^s\|^2 + C_3 \varepsilon \lambda_\sigma.$$

Similarly to the upper bound, the bound on the decrease $V_T(x_{MPC}(t+n)) - V_T(x_{MPC}(t))$ also has an additional term of the form $C_4(\varepsilon + \varepsilon^2)$. However, this estimate is only obtained if the first n entries of the optimal control sequence are applied in the MPC scheme in each step (as opposed to only the first entry, as usual). This way of implementing MPC is called "n-step MPC".

For the n-step MPC scheme this implies a practical exponential convergence estimate of the form

$$\|x_{MPC}(t) - x^s\| \le \max\{C\sigma^t \|x_{MPC}(0) - x^s\|, \beta(\varepsilon)\},$$

where $\beta(\varepsilon)$ is a polynomial in ε. This inequality holds for all initial values with $\|x_{MPC}(0) - x^s\| \le \Delta(\varepsilon)$, where $\Delta(\varepsilon) \to \infty$ as $\varepsilon \to 0$.

This means that the measurement noise, represented here by its amplitude $\varepsilon > 0$, determines both the neighborhood of x^s to which the solutions converge and the set of initial conditions for which the convergence holds.

Bibliography

[1] P. Benner and H. Fassbender, *Modellreduktion. Eine systemtheoretisch-orientierte Einführung*, Springer-Verlag, Erscheint demnächst in der Reihe Studium Mathematik — Master.

[2] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, *Data-driven model predictive control with stability and robustness guarantees*, IEEE Transactions on Automatic Control, 66 (2021), pp. 1702–1717.

[3] L. Grüne, *Vertiefung der Numerischen Mathematik.* Vorlesungsskript, Universität Bayreuth, 2019. Erhältlich von https://num.math.uni-bayreuth.de/de/team/lars-gruene/skripten/.

[4] ——, *Mathematical Control Theory.* Lecture Notes, University of Bayreuth, 2021. Available from https://num.math.uni-bayreuth.de/de/team/lars-gruene/skripten/.

[5] D. Werner, *Funktionalanalysis*, Springer, 5. ed., 2005.

[6] J. C. Willems, P. Rapisarda, I. Markovsky, and B. L. M. De Moor, *A note on persistency of excitation*, Systems Control Lett., 54 (2005), pp. 325–329.