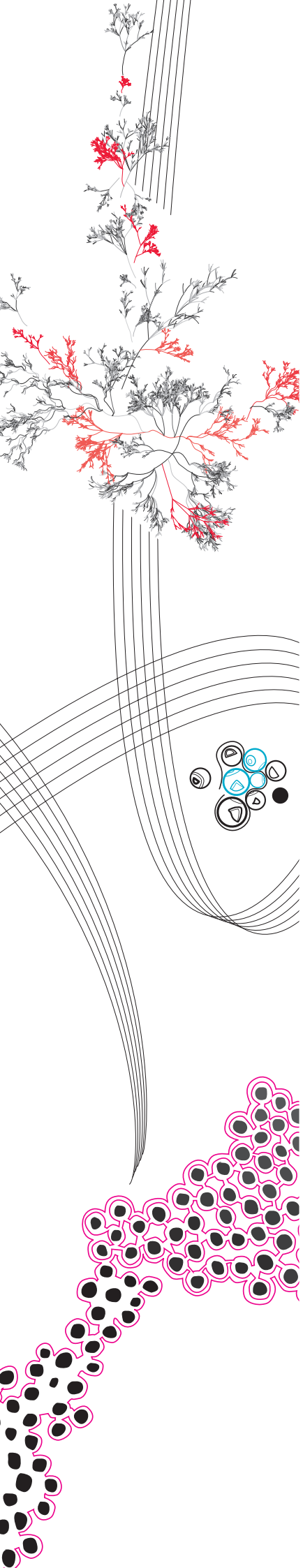

Finding the nearest positive-real system of lower order

MASTER THESIS D.F.H. SWART - S1367781

Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS), Master Applied Mathematics

Specialization: Systems Theory, Applied Analysis and Computational Science (SACS)

Chair: Hybrid Systems

Graduation committee:
Prof. dr. H.J. Zwart (UT)
Dr. ir. G. Meinsma (UT)
Dr. M. Schlottbom (UT)

Daily supervisor
Prof. dr. H.J. Zwart

Date presentation
23-11-2018

UNIVERSITY OF TWENTE.

Preface

During the past 28 weeks I have been working on this thesis. I have done my research internally at the University of Twente, where I have been studying in the past 6 years. This thesis concludes my time as a student, which has been innovating, fun, very instructive and a big part of my live. This thesis shows what I have learned in this part of my live. However, I have developed myself in multiple other ways as well and I believe that will be a really good foundation for the rest of my live. Therefore I would like to thank the following people, who have contributed a great deal of this experience.

First of all, I would like to thank Hans for supervising me throughout the whole period of working on this thesis. This topic was not easy for me and you have made it a lot better. Even more, I really appreciate it how you managed to read everything all the time, looking at how crowded your office is during the day. Also thanks to Wietske and Frank, for drinking coffee with me when we needed it and listening to me talking about generalized eigenvalues all the time.

Less research related but still very important during my period at the University of Twente, are the following people: Carolien for being able to live together with me. My parents for the mental support whenever I needed it (especially when I was a freshman). Koen, Daan, Hidde, Jan-Tino and (again) Wietske for the awesome time at our study association. Niek, Catherine, Michelle, Rik, Vera, Tamara and Niels for the 'sporty' moments at the badminton association, Diana for the oppurtunities to finance the expensive years and of course all the others that I did not mention explicitly. All together made my time here in Enschede unforgettable and I really hope to continue seeing you guys in the next period of my live!

Summary

Many physical situations can be modeled as an linear time-invariant (LTI) system. These models can become very complex, consisting of many parameters, making the order of the model (the size of the state vector) very large. Models of high order can on one hand be very realistic, but on the other hand very heavy, which makes simulation expensive. This motivates us to find accurate lower-order approximations, where they should still be a realistic image of the modelled process.

Another aspect of realistic models is the positive realness of a system. An LTI system that is postive-real (PR), which is equivalent to passivity in the case of LTI-systems, loses energy when there is no input, something that happens in nature as well. This brings us to the idea that if we approximate a physical system, the approximation should at least be positive-real to follow nature's law.

In this report, we will start by showing how to find the nearest positive-real system to a given non-PR one. This is being done for descriptor systems (system described by $E\dot{x} = Ax + Bu$ and $y = Cx + Du$). We will state the problem of finding the nearest PR-system and reformulate this problem to an equivalent problem, which has a simple convex set of solutions. Then we will formulate an algorithm to find the nearest PR-system.

The second part of this report is about model reduction. We first describe truncation and residualization for standard input/state/output LTI systems ($E = I$). Since truncation and residualization strongly depend on the initial realization, we present a way to balance the system in order to improve the model reduction techniques. This results in balanced truncation and balanced residualization, and we will give an upper bound for the error between the original and the reduced system.

Before we can combine the two previous parts to finding the nearest positive-real system of reduced order, we have to generalize model reduction for standard systems to descriptor systems. In order to do so, we will generalize classical resulsts, such as Lyapunov equations, Controllability and Observability Gramians and balanced realizations. We will conclude this part with an algorithm that gives a balanced truncation for descriptor systems.

Next follows a chapter about the implementation in MATLAB. That chapter will cover the most important MATLAB functions. Finally, the results will be presented via a numerical example. We will discuss the results, argue about the combination of finding the nearest PR system and model reduction and conclude with future research topics.

Contents

1 Introduction

Consider the following m-input m-output linear time-invariant (LTI) system Σ of the form

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

for $t \in [t_0, \infty)$, with $A, E \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{m\times n}$ and $D \in \mathbb{R}^{m\times m}$ given matrices. Σ is called a *descriptor system* if E is not invertible and is called a *standard system* if $E = I$. Sometimes we will use the matrix quituple (E, A, B, C, D) to refer to system (1.1).

Definition 1.1. The system (1.1) is called *passive* (equivalent to *positive-real (PR)* for LTI-systems) if there exists a nonnegative scalar valued function $V : X \to \mathbb{R}$, called the storage function, such that $V(0) = 0$ and the dissipation inequality

$$V(x(t_1)) - V(x(t_0)) \leq \int_{t_0}^{t_1} u(t)^{\mathrm{T}} y(t) \mathrm{d}t$$

holds for all $u(t)$, t_0 and $t_1 \geq t_0$.

The restriction to systems (1.1) with the same number of inputs and outputs is necessary to have positive-real systems [1], which follows directly from the fact that Definition 1.1 requires $u(t)$ and $y(t)$ to have the same size. Some authors use the energy function instead of the storage function. The energy function is denoted by $E(t) := V(x(t))$. Note that the E in the energy function is a different E than in (1.1). The difference can be distinguished by looking at the context. Using the energy function instead of the storage function, we can rewrite equation (1.1) into the following:

$$E(t_1) - E(t_0) \leq \int_{t_0}^{t_1} u(t)^{\mathrm{T}} y(t) \mathrm{d}t.$$

This must still hold for all $u(t)$, t_0, $t_1 \geq t_0$ and $x(t_0)$.

Lemma 1.1. *The system* (1.1) *is passive if and only if there exists a nonnegative function* $V(x(t)) : X \to \mathbb{R}$*, called the storage function, such that*

$$\frac{\mathrm{d}}{\mathrm{d}t} V(x(t)) \leq u(t)^{\mathrm{T}} y(t)$$

holds for all $u(t)$*.*

Proof. "⇒" Suppose the system of the form (1.1) is passive. Then, by Definition 1.1,

$$V(x(t_1)) - V(x(t_0)) \leq \int_{t_0}^{t_1} u(t)^{\mathrm{T}} y(t) \mathrm{d}t$$

holds for any t_0 and t_1 (such that $t_0 \leq t_1$) and $u(t)$. The left side can also be written as an integral:

$$\begin{aligned} \int_{t_0}^{t_1} \frac{\mathrm{d}}{\mathrm{d}t} V(x(t)) \mathrm{d}t = V(x(t_1)) - V(x(t_0)) &\leq \int_{t_0}^{t_1} u(t)^{\mathrm{T}} y(t) \mathrm{d}t, \\ \Rightarrow \frac{\mathrm{d}}{\mathrm{d}t} V(x(t)) &\leq u(t)^{\mathrm{T}} y(t). \end{aligned}$$

The last step follows from the fact that the inequality holds for every t_0, t_1 and $u(t)$.
"⇐" Suppose we have for a system of the form (1.1)

$$\frac{\mathrm{d}}{\mathrm{d}t} V(x(t)) \leq u(t)^{\mathrm{T}} y(t),$$

for any $u(t)$. Integrating both sides from t_0 to t_1 yields

$$V(x(t_1)) - V(x(t_0)) = \int_{t_0}^{t_1} \frac{\mathrm{d}}{\mathrm{d}t} V(x(t))\mathrm{d}t \leq \int_{t_0}^{t_1} u(t)^{\mathrm{T}} y(t)\mathrm{d}t.$$

Hence, by Definition 1.1, the system is passive. □

Note that if the input function would be $u(t) = 0$ for all $t \in [t_0, \infty)$, we have $\dot{E}(t) = \dot{V}(x(t)) \leq 0$, hence the system is dissipative.

The goal of this report is, for a given system Σ (1.1) which is not necessarily positve-real, to find the nearest PR system of lower order Σ_a. By model order, we mean the dimension of the state vector $x(t)$ (sometimes called the *McMillan degree*). This goal consists of two different sub-goals: finding the nearest postive-real system on one side and reducing the order of the system at the other side. The two sub-goals will be discussed in different chapters.

In this report we will start with solving the problem of finding the nearest PR system in Chapter 2. In Chapter 3 we will discuss model reduction of standard systems, which we will expand to model reduction of descriptor systems in Chapter 4. The implementation in MATLAB will be shown in Chapter 5 and Chapter 6 covers the results of a numerical experiment. We will concluse this report with a conclusion and a discussion.

2 Finding the nearest positive-real system

This chapter is a summary of [2], together with some extra theory. Therefore the outline is roughly the same as the outline in [2]. We assume that the reader has knowledge about norms, inner products and inner product spaces.

The goal of this chapter is, for a given system (1.1) with (E, A, B, C, D) which is not positive-real, to find the nearest positive-real (PR) system. The distance is given in terms of the Frobenius norm.

Definition 2.1. For a given matrix $A \in \mathbb{R}^{n\times n}$ the *Frobenius norm* is given as

$$||A||_F^2 := \operatorname{tr}(A^{\mathrm{T}}A) = \sum_{i,j} a_{i,j}^2, \tag{2.1}$$

where tr(A) is the trace of the matrix A.

In [2] the closest PR system to a non PR system is computed using the set of linear port-Hamiltonian (PH) systems. This is used to work with a simpler feasible set (the original one is neither open nor closed, unbounded and highly nonconvex, which will be shown later in this section).

In Section 2.1, we introduce the notations and definitions that are used throughout this chapter. Moreover, we state our problem in a more technical way. In Section 2.2 a summary of key results reported in [2] is given. These results will be used in Section 2.3 to reformulate the problem of finding the nearest positive-real system with a simpler feasible set. This chapter will be concluded by giving an algorithmical approach to the nearest PR system problem in Section 2.4.

2.1 Notation, preliminaries and (sub-)problem definition

This section contains notations and definitions used throughout the rest of the chapter. Moreover, we will introduce a more formal way of the problem that will be tackled in this section. In the following, we will write $A \succ 0$ (resp. $A \succeq 0$) if A is *symmetric positive definite* (resp. *semi-definite*). The *real part of* $s \in \mathbb{C}$ is denoted by $\operatorname{Re}(s)$.

2.1.1 Positive-real systems

Definition 2.2. The system (1.1) is called *regular* if the matrix pair (E, A) is regular, that is, if $\det(\lambda E - A) \neq 0$ for some $\lambda \in \mathbb{C}$, otherwise it is called *singular*. For a regular matrix pair (E, A) the roots of the polynomial $\det(\lambda E - A)$ are called the *finite eigenvalues* of the pencil $\lambda E - A$ or of the pair (E, A). A regular pair (E, A) has ∞ *as an eigenvalue* (with multiplicity n_∞) if E is singular.

Example 2.1. Consider the following matrix pair (E, A):

$$(E, A) = \left(\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right).$$

This matrix pair is regular, since for $\lambda = 2$ we get $\det(2 \cdot E - A) = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = -1$. However, if the second diagonal term of A would be zero, the matrix pair (E, A) is singular, since the determinant is zero for every $\lambda \in \mathbb{C}$.

Definition 2.3. The regular matrix pair (E, A) is said to be *stable* (resp. *asymptotically stable*) if all the finite eigenvalues of $\lambda E - A$ are in the closed (resp. open) left half of the complex plane and those on the imaginary axis are semisimple. A dynamical system in the form of (1.1) is called *(asymptotically) stable* if the matrix pair (E, A) is *(asymptotically) stable*.

If the system (1.1) is regular, it can be described by its *transfer function* $G(s) : \mathbb{C} \to (\mathbb{C} \cup \infty)^{m \times m}$, given by:

$$G(s) := C(Es - A)^{-1}B + D, \quad s \in \mathbb{C}. \tag{2.2}$$

The transfer function can be obtained by taking the Laplace transform with $x_0 = 0$ of

$$E\dot{x}(t) - Ax(t) = Bu(t),$$

which results in

$$\begin{aligned} &(Es - A)X(s) = BU(s), \\ &\Rightarrow \\ &Y(s) = C(Es - A)^{-1}BU(s) + DU(s), \\ &Y(s) = \big(C(Es - A)^{-1}B + D\big)\, U(s). \end{aligned}$$

Hence, if we write $Y(s) = G(s)U(s)$, we have obtained the transfer function as in (2.2).

Definition 2.4. Any representation of $G(s)$ in the form (2.2) is called a *realization* of $G(s)$. A realization is called *minimal* if the matrices A and E are of smallest possible dimension, or equivalently, the system is called *minimal* if it is both controllable and observable (see [3] for standard systems and [4] for descriptor systems).

Definition 2.5. The system (1.1) is said to be

1. *Positive real (PR)* if its transfer function $G(s)$ satisfies
   (a) $G(s)$ has no pole in $\mathrm{Re}(s) > 0$, and
   (b) $G(s) + G^*(s) \succeq 0$ for all s such that $\mathrm{Re}(s) > 0$
2. *strictly positive real (SPR)* if $G(s)$ satisfies
   (a) $G(s)$ has no pole in $\mathrm{Re}(s) \geq 0$, and
   (b) $G(\mathrm{i}\omega) + G^*(\mathrm{i}\omega) \succ 0$ for $\omega \in [0, \infty)$

Note that (a) in Definition 2.5 of PR (resp. SPR) is equivalent to (1.1) being stable (resp. asymptotically stable). System (1.1) with transfer function $G(s)$ is called *passive* if and only if it is PR [5]. Furthermore, SPR $\implies$ PR.

2.1.2 Nearest positive-real system problem

We can now define our nearest system problem:

Problem. For a given system (E, A, B, C, D) as in (1.1) and a given set $\mathcal{D}$, find the nearest system $(\tilde{E}, \tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}) \in \mathcal{D}$ to (E, A, B, C, D), that is, solve

$$\inf_{(\tilde{E}, \tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}) \in \mathcal{D}} \mathcal{F}(\tilde{E}, \tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}), \tag{2.3}$$

where $\mathcal{F}(\tilde{E}, \tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}) := ||A - \tilde{A}||_{\mathrm{F}}^2 + ||B - \tilde{B}||_{\mathrm{F}}^2 + \cdots + ||E - \tilde{E}||_{\mathrm{F}}^2$.

We consider the problem Nearest PR-system $(\mathcal{P})$: $\mathcal{D} := \mathbb{S}$, where $\mathbb{S}$ is the set of all PR-systems $(\tilde{E}, \tilde{A}, \tilde{B}, \tilde{C}, \tilde{D})$. This problem is challenging since $\mathbb{S}$ is unbounded, highly nonconvex [6] and neither open nor closed. See pages 5 and 6 of [2] for an example.

2.1.3 Port-Hamiltonian systems

A linear time-invariant input-state-output system is called a *port-Hamiltonian (PH)* system if it can be written as

$$\begin{aligned} M\dot{x}(t) &= (J-R)Qx(t) + (F-L)u(t), \\ y(t) &= (F+L)^{\mathrm{T}}Qx(t) + (S+N)u(t), \end{aligned} \tag{2.4}$$

where the following must hold:

- The matrix $Q \in \mathbb{R}^{n\times n}$ is invertible, $M \in \mathbb{R}^{n\times n}$, and $Q^{\mathrm{T}}M = M^{\mathrm{T}}Q \succeq 0$. The function $x \to \frac{1}{2}x^{\mathrm{T}}Q^{\mathrm{T}}Mx$ is the *Hamiltonian* and describes the energy of the system.
- The matrix $J^{\mathrm{T}} = -J \in \mathbb{R}^{n\times n}$ is the structure matrix and describes how the energy remains in the system.
- The matrix $R \in \mathbb{R}^{n\times n}$ with $R \succeq 0$ is the dissipation matrix and describes the energy dissipation/loss in the system.
- The matrices $F \pm L \in \mathbb{R}^{n\times m}$ are the port matrices describing the way in which energy enters and/or leaves the system.
- The matrix $S + N$, with $0 \preceq S \in \mathbb{R}^{m\times m}$ and $N^{\mathrm{T}} = -N \in \mathbb{R}^{m\times m}$, describes the direct feed-through from input to output.
- The matrices L, R and S satisfy

$$K = \begin{bmatrix} R & L \\ L^{\mathrm{T}} & S \end{bmatrix} \succeq 0.$$

Moreover, the Hamiltonian $\mathcal{H}(x) = \frac{1}{2}x^{\mathrm{T}}Q^{\mathrm{T}}Mx$ defines an energy function (see (1.1)) and satisfies

$$\mathcal{H}(x(t_1)) - \mathcal{H}(x(t_0)) \leq \int_{t_0}^{t_1} u(t)^{\mathrm{T}}y(t)\mathrm{d}t,$$

which guarantees the passivity of the system. Regular PH systems are always stable [7, Lemma 2]: the matrix pair $(E, A) = (M, (J-R)Q)$ is a so-called dissipative Hamiltonian matrix pair.

2.2 Key results for positive-real systems

In this section, we give some important results from [2] regarding the link between PR- and PH systems. For the proofs of the following theorems we refer to [2].

The positive realness of a system (1.1) can be characterized in terms of solutions X to the following linear matrix inequalities (LMIs):

$$\begin{bmatrix} A^{\mathrm{T}}X + X^{\mathrm{T}}A & X^{\mathrm{T}}B - C^{\mathrm{T}} \\ B^{\mathrm{T}}X - C & -D - D^{\mathrm{T}} \end{bmatrix} \preceq 0 \quad \text{and} \quad E^{\mathrm{T}}X = X^{\mathrm{T}}E \preceq 0. \tag{2.5}$$

Theorem 2.1 ([2], Theorem 1)**.** *Consider a system (E, A, B, C, D) in the form (1.1). If the LMIs (2.5) have a solution $X \in \mathbb{R}^{n\times n}$, then (E, A, B, C, D) is PR.*

Note that the proof of Theorem 2.1 is not given in [2], but in [1]. The converse of Theorem 2.1 is true under some additional assumptions. In fact, the positive real lemma for standard systems [8] proves that if a system is PR and minimal, then the existence of a solution to the LMIs (2.5) is also necessary. Similarly, with an additional condition, the positive real lemma for descriptor systems [1] proves that the existence of a solution to the LMIs (2.5) is also necessary for positive realness.

Theorem 2.2 ([2], Theorem 2)**.** *Every PH system* (2.4) *is PR.*

Definition 2.6. A system (E, A, B, C, D) is said to admit a *port-Hamiltonian form (PH form)* if there exists a PH system as defined in (2.4) such that

$$E = M, \quad A = (J - R)Q, \quad B = F - L, \quad C = (F + L)^{\mathrm{T}}Q, \quad \text{and} \quad D = S + N.$$

Theorem 2.3 ([2], Theorem 5)**.** *Let $\Sigma = (E, A, B, C, D)$ be a system in the form* (1.1)*. If the LMIs* (2.5) *have an invertible solution $X \in \mathbb{R}^{n\times n}$, then Σ admits a PH form.*

Corollary 2.1 ([2], Corollary 1)**.** *If the system (E, A, B, C, D) is minimal and PR, then it admits a PH form.*

Proof. If the system is minimal and PR, the extended positive real lemma for minimal PR systems [1] guarantees the existence of an invertible solution X of the LMIs (2.5). Then, by Theorem 2.3 the system admits a PH form. □

2.3 Reformulation of the nearest PR system problem

In this section the nearest PR system problem will be reformulated such that it can be solved using the results of the previous section. Define the following set:

- The set $\mathbb{S}_{\mathrm{PH}}$ containing all systems (E, A, B, C, D) in PH form, that is,

$$\begin{aligned}\mathbb{S}_{\mathrm{PH}} &:= \{(E, A, B, C, D) \mid (E, A, B, C, D) \text{ admits a PH form}\}\\ &= \Big\{(M, (J - R)Q, F - L, (F + L)^{\mathrm{T}}Q, S + N) \mid J^{\mathrm{T}} = -J, N^{\mathrm{T}} = -N,\\ &\qquad M^{\mathrm{T}}Q \succeq 0, Q \text{ invertible}, K = \begin{bmatrix} R & L\\ L^{\mathrm{T}} & S\end{bmatrix} \succeq 0\Big\}.\end{aligned}$$

  By Theorem 2.2, every system in PH form is PR but the converse is not known, hence $\mathbb{S}_{\mathrm{PH}} \subseteq \mathbb{S}$.

The set $\mathbb{S}_{\mathrm{PH}}$ is neither closed (due to the constraint that Q is invertible) nor open (due to the constraint $E^{\mathrm{T}}Q \succeq 0$). Since we want to work with a set onto which projection is easy (and possible), we work with the closure $\overline{\mathbb{S}_{\mathrm{PH}}}$ of $\mathbb{S}_{\mathrm{PH}}$ which is equal to the set $\mathbb{S}_{\mathrm{PH}}$ except that Q can be singular.

Theorem 2.4 ([2], Theorem 7)**.** *Let (E, A, B, C, D) be a system in the form* (1.1) *and $\mathcal{F}$ be defined as in* (2.3)*. Then*

$$\inf_{(M,(J-R)Q,F-P,(F+P)^{\mathrm{T}}Q,S+N)\in\overline{\mathbb{S}_{PH}}} \mathcal{F}(M, (J - R)Q, F - L, (F + L)^{\mathrm{T}}Q, S + N) \tag{2.6}$$

is an upper bound for the infimum of ($\mathcal{P}$) (See (2.3)*). Moreover, every feasible solution of* (2.6) *is a PR system.*

Proof. This follows directly from the fact that $\overline{\mathbb{S}_{\mathrm{PH}}} \subseteq \mathbb{S}$ (the proof of Theorem 2.2 does not require Q to be invertible). □

We will refer to the problem (2.6) as ($\mathcal{P}_{\mathrm{PH}}$). The solution of (2.6) lies in $\overline{\mathbb{S}_{\mathrm{PH}}}$. Since $\overline{\mathbb{S}_{\mathrm{PH}}} \subseteq \mathbb{S}$, the solution lies also in $\mathbb{S}$ and is therefore feasible for ($\mathcal{P}$). The solution of (2.3) might be in $\mathbb{S}$, while not in $\overline{\mathbb{S}_{\mathrm{PH}}}$, so the infimum of ($\mathcal{P}$) can be lower than the infimum of ($\mathcal{P}_{\mathrm{PH}}$). Hence (2.6) is an upper bound for the infimum of ($\mathcal{P}$).

2.4 Algorithmic solution to the nearest PH system problem

In this section we discuss the algorithm from [2] to tackle (2.6). The standard ($E = I_n$ and no perturbation in E) and general systems will be analyzed seperately. The main results are from [9]. Before we give an algorithm for the nearest PH system problem, we will simplify (2.6). First, define the following two sets: the set of all symmetric matrices

$$\mathfrak{R} := \{R \in \mathbb{R}^{n\times n} \mid R^T = R\}$$

and the set of all skew-symmetric matrices

$$\mathfrak{J} := \{J \in \mathbb{R}^{n\times n} \mid J^{\mathrm{T}} = -J\}.$$

2.4.1 Standard systems

For standard systems we have $M = E = I_n$ and (2.6) can therefore be simplified as follows

$$\begin{aligned}\inf_{J,R,Q,F,P,S,N} &||A - (J-R)Q||_{\mathrm{F}}^2 + ||B - (F-L)||_{\mathrm{F}}^2 + ||C - (F+L)^{\mathrm{T}}Q||_{\mathrm{F}}^2 \\ &+ ||D - (S+N)||_{\mathrm{F}}^2, \\ \text{such that} \quad & J^{\mathrm{T}} = -J, \quad Q \succeq 0, \quad N^{\mathrm{T}} = -N \text{ and } \begin{bmatrix} R & L \\ L^{\mathrm{T}} & S \end{bmatrix} \succeq 0.\end{aligned} \tag{2.7}$$

Denote $\mathcal{P}_{\mathfrak{J}}(Z)$ as the projection of Z onto the set of skew-symmetric matrices. For a given square matrix Z, $\mathcal{P}_{\mathfrak{J}}(Z)$ is given by (see Appendix A)

$$\mathcal{P}_{\mathfrak{J}}(Z) = \frac{Z - Z^{\mathrm{T}}}{2}, \tag{2.8}$$

which gives

$$\min_{J^{\mathrm{T}}=-J} ||Z - J||_{\mathrm{F}}^2 = ||Z - \mathcal{P}_{\mathfrak{J}}(Z)||_{\mathrm{F}}^2 = \left|\left|Z - \frac{Z - Z^{\mathrm{T}}}{2}\right|\right|_{\mathrm{F}}^2 = \left|\left|\frac{Z + Z^{\mathrm{T}}}{2}\right|\right|_{\mathrm{F}}^2. \tag{2.9}$$

Equation (2.9) implies that the optimal $\hat{N}$ in (2.7) for $||D - (S+N)||_{\mathrm{F}}^2$ is given by $\frac{D-D^{\mathrm{T}}}{2}$. This makes sense: since S is symmetric, the closest skew-symmetric matrix to $D - S$ is the skew-symmetric part of D. Substituting $\frac{D-D^{\mathrm{T}}}{2}$ in (2.7) gives

$$\begin{aligned}\min_{N^{\mathrm{T}}=-N} ||D - (S+N)||_{\mathrm{F}}^2 &= \left|\left|D - S - \mathcal{P}_{\mathfrak{J}}(D-S)\right|\right|_{\mathrm{F}}^2 \\ &= \left|\left|D - S - \frac{D - D^{\mathrm{T}}}{2}\right|\right|_{\mathrm{F}}^2 \\ &= \left|\left|\frac{D + D^{\mathrm{T}}}{2} - S\right|\right|_{\mathrm{F}}^2.\end{aligned}$$

Hence, equation (2.7) can be simplified to

$$\begin{aligned}\inf_{J,R,Q,F,P,S} &||A - (J-R)Q||_{\mathrm{F}}^2 + ||B - (F-L)||_{\mathrm{F}}^2 + ||C - (F+L)^{\mathrm{T}}Q||_{\mathrm{F}}^2 + \left|\left|\frac{D + D^{\mathrm{T}}}{2} - S\right|\right|_{F}^2, \\ \text{such that} \quad & J^{\mathrm{T}} = -J, \quad Q \succeq 0 \text{ and } \begin{bmatrix} R & L \\ L^{\mathrm{T}} & S \end{bmatrix} \succeq 0.\end{aligned} \tag{2.10}$$

Similarly as with the projection on skew-symmetric matrices, $\mathcal{P}_{\succeq}(Z)$ denotes the projection of Z onto the cone of positive semidefinite matrices. For a given square matrix Z, $\mathcal{P}_{\succeq}(Z)$ is given by (see Appendix A)

$$\mathcal{P}_{\succeq}(Z) = U\left(\max(\Gamma, 0)\right)U^{\mathrm{T}}, \tag{2.11}$$

where $U\Gamma U^{\mathrm{T}}$ is an eigenvalue decomposition of the symmetric matrix $\frac{Z+Z^{\mathrm{T}}}{2}$ with unitary matrix U. Using equation (2.11) gives

$$\min_{R\succeq 0}||Z-R||_{\mathrm{F}}^2 = ||Z-\mathcal{P}_{\succeq}(Z)||_{\mathrm{F}}^2 = ||\mathcal{P}_{\mathfrak{J}}(Z)||_{\mathrm{F}}^2 + \sum_{\lambda\in\Lambda(\Gamma),\lambda<0}\lambda^2.$$

In Appendix A we prove why this is true. In order to simplify the description of the algorithm, define $\mathcal{G} := \{G \in \mathbb{R}^{n\times n} \mid G = J - R, J^{\mathrm{T}} = -J, R \succeq 0\}$. Projection of a square matrix Z on $\mathcal{G}$ is equivalent to project seperately on the set of skew-symmetric matrices and the set of positive semi-definite matrices. This is shown in the following lemma.

Lemma 2.1 ([9], Lemma 7)**.** *Let $Z \in \mathbb{R}^{n\times n}$, then*

$$\min_{G\in\mathcal{G}}||Z-G||_{\mathrm{F}}^2 = ||Z-(\mathcal{P}_{\mathfrak{J}}(Z)-\mathcal{P}_{\succeq}(-Z))||_{\mathrm{F}}^2,$$

where $\mathcal{P}_{\mathfrak{J}}(Z)$ and $\mathcal{P}_{\succeq}(-Z)$ are defined as in (2.8) *and* (2.11)*, respectively.*

Proof.

$$\begin{aligned}
\min_{G\in\mathcal{G}}||Z-G||_{\mathrm{F}}^2 &= \min_{J^{\mathrm{T}}=-J,R\succeq 0}||Z-(J-R)||_{\mathrm{F}}^2 \\
&= \min_{R\succeq 0}\left(\min_{J^{\mathrm{T}}=-J}||(Z+R)-J||_{\mathrm{F}}^2\right) \\
&= \min_{R\succeq 0}\left|\left|\frac{Z+Z^{\mathrm{T}}}{2}-(-R)\right|\right|_{\mathrm{F}}^2 \\
&= \left|\left|\frac{Z+Z^{\mathrm{T}}}{2}+\mathcal{P}_{\succeq}(-\frac{Z+Z^{\mathrm{T}}}{2})\right|\right|_{\mathrm{F}}^2 \\
&= \left|\left|\frac{Z+Z^{\mathrm{T}}}{2}+\mathcal{P}_{\succeq}(-Z)\right|\right|_{\mathrm{F}}^2,
\end{aligned} \tag{2.12}$$

where the third equality follows from (2.9) and the last equality holds since $\mathcal{P}_{\succeq}(Z) = \mathcal{P}_{\succeq}(Z^{\mathrm{T}})$. Using (2.8) in (2.12) leads to

$$\begin{aligned}
\left|\left|\frac{Z+Z^{\mathrm{T}}}{2}+\mathcal{P}_{\succeq}(-Z)\right|\right|_{\mathrm{F}}^2 &= ||Z-\mathcal{P}_{\mathfrak{J}}(Z)+\mathcal{P}_{\succeq}(-Z)||_{\mathrm{F}}^2 \\
&= ||Z-(\mathcal{P}_{\mathfrak{J}}(Z)-\mathcal{P}_{\succeq}(-Z))||_{\mathrm{F}}^2.
\end{aligned}$$

□

In [2] a fast projected gradient method (FGM) is developed to solve (2.10). FGM is in general much faster than the standard projected gradient method, even in the nonconvex case, while being relatively simple to implement. They use the following:

- Compute the gradient: all the terms in the objective function are of the form $f(X) = ||AX-B||_{\mathrm{F}}^2$ whose gradient is $\nabla_X f(X) = 2A^{\mathrm{T}}(AX-B)$ (see Appendix B).
- Project onto the feasible set: the projection onto the set of skew-symmetric matrices $\{J : J = -J^{\mathrm{T}}\}$ is given in (2.8), while projection onto the set of positive semidefinite matrices $\{R : R \succeq 0\}$ is given in (2.11).

See [2] for a pseudocode for their FGM (Algorithm 1) and the parameter settings.

2.4.2 General systems

Similarly as for standard systems in (2.10), (2.6) can be simplified to

$$\begin{aligned}\inf_{J,R,Q,M,F,P,S} &||A-(J-R)Q||_{\mathrm{F}}^2+||B-(F-L)||_{\mathrm{F}}^2+||C-(F+L)^{\mathrm{T}}Q||_{\mathrm{F}}^2+\\ &\left|\left|\frac{D-D^{\mathrm{T}}}{2}+S\right|\right|_{\mathrm{F}}^2+||E-M||_{\mathrm{F}}^2,\\ \text{such that}\quad & J^{\mathrm{T}}=-J,\ \ M^{\mathrm{T}}Q\succeq 0 \text{ and } \begin{bmatrix}R & L\\ L^{\mathrm{T}} & S\end{bmatrix}\succeq 0.\end{aligned} \tag{2.13}$$

In this case, the coupling constraint $M^{\mathrm{T}}Q \succeq 0$ makes the projection on the feasible domain of (2.13) difficult, see [7, Example 3] for an example. Following the strategy used in [7], we introduce a new variable $Z := M^{\mathrm{T}}Q$ so that $M^{\mathrm{T}} = ZQ^{-1}$. In this way, the feasible set becomes simpler:

$$\begin{aligned}\inf_{J,R,Q,Z,F,P,S} &||A-(J-R)Q||_{\mathrm{F}}^2+||B-(F-L)||_{\mathrm{F}}^2+||C-(F+L)^{\mathrm{T}}Q||_{\mathrm{F}}^2+\\ &\left|\left|\frac{D-D^{\mathrm{T}}}{2}+S\right|\right|_{\mathrm{F}}^2+||E^{\mathrm{T}}-ZQ^{-1}||_{\mathrm{F}}^2,\\ \text{such that}\quad & J^{\mathrm{T}}=-J,\ \ Z\succeq 0 \text{ and } \begin{bmatrix}R & L\\ L^{\mathrm{T}} & S\end{bmatrix}\succeq 0,\end{aligned}$$

for which Algorithm 1 has been developed in [2]. The gradient of $||E^{\mathrm{T}} - ZQ^{-1}||_{\mathrm{F}}^2$ with respect to Q is given in [7, Appendix A].

2.4.3 Initializations

Since we are dealing with a non-convex optimization problem, it is very important that good initial points will be chosen. In this section, two initializations are proposed.

Standard initialization
The standard initialization uses $Q = I_n$ and $L = 0$. For these values of Q and L, the optimal solutions for the other variables are given explicitly:

$$J = (A - A^{\mathrm{T}})/2,\ \ R = \mathcal{P}_{\succeq}\left((-A - A^{\mathrm{T}})/2\right),\ \ S = \mathcal{P}_{\succeq}\left((D^{\mathrm{T}} + D)/2\right),\ \ F = (B + C^{\mathrm{T}})/2$$

and $Z = \mathcal{P}_{\succeq}(E^{\mathrm{T}}) = \mathcal{P}_{\succeq}(E)$ for general systems. $\mathcal{P}_{\succeq}(X)$ stands for the projection of a matrix X on the cone of positive semidefinite matrices, see (2.11). This initialization has the advantage that is very simple to compute while working well in many cases.

LMI-based initializations
Now we will give a initialization that uses the knowledge about the LMIs (2.5). By Theorem 2.2, we know that every PH system is PR. By Theorem 2.3 we also know that if a system does not admit a PH form, the LMIs (2.5) will not have a solution. However, since we are looking for a system that does admit a PH form (and hence that will admit a solution to the LMIs), it makes sense to initialize 'close' to the LMIs. There are multiple ways to relax the LMIs and the following is proposed in [2]:

$$\begin{aligned}\min_{\delta,X}\ & \delta^2\\ \text{such that}\ & \begin{bmatrix}A^{\mathrm{T}}X + X^{\mathrm{T}}A & X^{\mathrm{T}}B - C^{\mathrm{T}}\\ B^{\mathrm{T}}X - C & -D - D^{\mathrm{T}}\end{bmatrix} + \delta I_{n+m} \succeq 0,\\ & E^{\mathrm{T}}X + \delta I_n \succeq 0.\end{aligned} \tag{2.14}$$

Denote (δ^*, X^*) as an optimal solution of (2.14). By Theorem 2.3, if $\delta^* = 0$ and X^* is invertible, then the system (E, A, B, C, D) admits a PH form that can be constructed explicitly [2]. Moreover, as long as X^* is invertible, the matrices (J, L, R, Q, S, N, Z) can be constructed and projected onto the feasible set $\overline{\mathbb{S}_{\mathrm{PH}}}$ to obtain an initial system in PH form. We will refer to this initialization as 'LMIs + formula'. For a given $Q = X^*$, it is possible to compute the matrices (J, L, R, S, N) in order to obtain a better initial point, by solving a semidefinite program:

$$\min_{J,R,S,N,P} \; ||A - (J-R)Q||_{\mathrm{F}}^2 + ||B - (F-L)||_{\mathrm{F}}^2 + ||C - (F+L)^{\mathrm{T}}Q||_{\mathrm{F}}^2 + \left|\left|\frac{D - D^{\mathrm{T}}}{2} + S\right|\right|_F,$$
$$\text{such that } \; J^{\mathrm{T}} = -J \text{ and } \begin{bmatrix} R & L \\ L^{\mathrm{T}} & S \end{bmatrix} \succeq 0,$$

while taking $Z = \mathcal{P}_{\geq}(M^{\mathrm{T}}Q)$ (as $Q = X^*$ can be ill-conditioned). We will refer to this initialization as 'LMIs + solve'. By construction, it provides an initial point with smaller objective function value than 'LMIs + formula', at the cost of solving another program. Check [2] for additional remarks.

3 Model reduction of standard systems

The information given in this chapter is based on Chapter 11 of [10]. That chapter discusses model reduction, but the information is given in a concise way, that is why this chapter is more detailed. The outline of this chapter is roughly the same as Chapter 11 of [10].

Consider the following high-order m-input m-output causal linear time-invariant (LTI) system Σ of the form

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t), \end{aligned} \tag{3.1}$$

for $t \in [t_0, \infty)$, with $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{m\times n}$ and $D \in \mathbb{R}^{m\times m}$ given matrices. We give the following problem: find a low-order approximation Σ_a such that the difference $\Sigma - \Sigma_a$ is small. By model order, we mean the dimension of the state vector $x(t)$ in a minimal realization (sometimes called the McMillan degree) and by difference we mean the ∞-norm ($\mathcal{H}_\infty$ or $\mathcal{L}_\infty$) of $\Sigma - \Sigma_a$. The definition of the $\mathcal{H}_\infty$-norm, given in (3.3), is only for stable systems. Since the error $\Sigma - \Sigma_a$ may be unstable and the $\mathcal{L}_\infty$-norm is defined for all rational functions without poles on the imaginary axis, we will use the $\mathcal{L}_\infty$-norm instead.

Two methods for tackling this problem will be given: Section 3.2 covers truncation and residualization of standard systems. In order to obtain better realizations, balanced realizations will be discussed in Section 3.3. Section 3.4 extends truncation and residualization with the ideas of balanced realizations to balanced truncation and balanced residualization. But first, we will give some background information on the $\mathcal{L}_\infty$-norm and the $\mathcal{H}_\infty$-norm in Section 3.1.

3.1 Two system norms: the $\mathcal{L}_\infty$- and $\mathcal{H}_\infty$-norm

Recall that system Σ as in (3.1) has the transfer function

$$G(s) = C(sI - A)^{-1}B + D$$

and impulse response

$$h(t) = Ce^{At}B\mathbb{1}(t) + D\delta(t).$$

3.1.1 The $\mathcal{L}_2$- and $\mathcal{H}_2$-norm

Before we can talk about norms of a system, we first need to have an idea about the norm of a signal. Therefore, we start with the definition of the $\mathcal{L}_2$ signal norm, which is the square root of the energy of the signal:

Definition 3.1. Let $z(t) : \mathbb{R} \to \mathbb{C}^{n\times m}$ be a signal. The *$\mathcal{L}_2$-norm* of the signal $z(t)$ is defined as the square root of the sum of energies of all its entries,

$$||z(t)||_{\mathcal{L}_2} := \sqrt{\int_{-\infty}^{\infty} \sum_{i=1}^{n}\sum_{i=1}^{m} |z_{ij}(t)|^2 \,\mathrm{d}t}.$$

The $\mathcal{H}_2$ system norm is based on the $\mathcal{L}_2$ signal norm: the $\mathcal{H}_2$-norm of system Σ is the $\mathcal{L}_2$-norm of the impulse response $h(t)$ of the system Σ:

Definition 3.2. Let Σ be the system defined as in (3.1). The *$\mathcal{H}_2$-norm* of the system Σ with impulse response $h(t)$ is defined as

$$||\Sigma||_{\mathcal{H}_2} := \sqrt{\int_0^\infty \operatorname{tr}\left(h^*(t)h(t)\right) \mathrm{d}t},$$

where $h^*(t)$ is the complex conjugate of $h(t)$.

For causal systems as in (3.1) we can compute the $\mathcal{H}_2$-norm without having to solve the integral. This follows from the following two lemmas. Note that we must have $D = 0$, otherwise the impulse response would have a delta function, which has infinite $\mathcal{H}_2$-norm.

Lemma 3.1 ([11], Lemma 2.5.1)**.** *If $G(s) = C(sI - A)^{-1}B$ and if A is asymptotically stable, then*

$$A^{\mathrm{T}}P + PA + C^{\mathrm{T}}C = 0$$

has a unique symmetric solution $P \in \mathbb{R}^{n\times n}$ and

$$||\Sigma||_{\mathcal{H}_2} = \sqrt{\operatorname{tr}(B^{\mathrm{T}}PB)}.$$

Lemma 3.2 ([11], Lemma 2.5.2)**.** *If $G(s) = C(sI - A)^{-1}B$ and if A is asymptotically stable, then*

$$AQ + QA^{\mathrm{T}} + BB^{\mathrm{T}} = 0$$

has a unique symmetric solution $Q \in \mathbb{R}^{n\times n}$ and

$$||\Sigma||_{\mathcal{H}_2} = \sqrt{\operatorname{tr}(CQC^{\mathrm{T}})}.$$

3.1.2 The $\mathcal{L}_\infty$- and $\mathcal{H}_\infty$-norm

We can extend the ideas of the previous section to the $\mathcal{L}_\infty$- and $\mathcal{H}_\infty$-norm. Consider the class of harmonic input signals. It is known that the output signal is again harmonic and therefore the maximal power gain over all harmonic systems can be defined as:

Definition 3.3. Suppose the frequency response $G(\mathrm{i}\omega)$ of system Σ is defined for all real ω. The *$\mathcal{L}_\infty$-norm* of the system Σ is defined as

$$||\Sigma||_{\mathcal{L}_\infty} := \sup_{\omega\in\mathbb{R}} \bar{\sigma}(G(\mathrm{i}\omega)), \tag{3.2}$$

where $\bar{\sigma}(G(\mathrm{i}\omega))$ is the maximum singular value of $G(\mathrm{i}\omega)$.

The following theorem holds for the whole class of signals with finite $\mathcal{L}_2$-norm.

Theorem 3.1 ([11], Theorem 2.6.3)**.** *Let Σ be the system defined as in (3.1) with transfer function $G(s)$ and impulse response $h(t)$. Define $y(t) := \int_{-\infty}^{t} h(t-\tau)u(\tau)\mathrm{d}\tau$. If*

$$\sup_{0<||u(t)||_{\mathcal{L}_2}<\infty} \frac{||y(t)||_{\mathcal{L}_2}}{||u(t)||_{\mathcal{L}_2}}$$

is finite, then the frequency response $G(\mathrm{i}\omega)$ exists for all $\omega \in \mathbb{R}$ and the equality

$$\sup_{0<||u(t)||_{\mathcal{L}_2}<\infty} \frac{||y(t)||_{\mathcal{L}_2}}{||u(t)||_{\mathcal{L}_2}} = ||G||_{\mathcal{L}_\infty}.$$

holds.

Now we can finally give a definition for the $\mathcal{H}_\infty$-norm of a system Σ.

Definition 3.4. Let Σ be a causal LTI-system with finite $\mathcal{L}_2$-gain. Suppose its transfer function $G(s)$ is defined for every $s \in \mathbb{C}$ in the open right halfplane (ORHP) and analytic on the ORHP. The $\mathcal{H}_\infty$-*norm* is defined as

$$||\Sigma||_{\mathcal{H}_\infty} := \sup_{\mathrm{Re}(s)>0} \bar{\sigma}(G(s)). \tag{3.3}$$

3.2 Truncation and residualization

In this section, we assume that the system Σ of (1.1) is a minimal realization of a stable system with transfer function $G(s)$. We partition the state vector $x \in \mathbb{R}^n$ into $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, where $x_1 \in \mathbb{R}^k$ contains the states we want to keep and $x_2 \in \mathbb{R}^{n-k}$ contains the states we want to remove. Partitioning the matrices A, B and C in the appropriate way, we obtain:

$$\begin{aligned} \dot{x}_1(t) &= A_{11}x_1(t) + A_{12}x_2(t) + B_1u(t), \\ \dot{x}_2(t) &= A_{21}x_1(t) + A_{22}x_2(t) + B_2u(t), \\ y(t) &= C_1x_1(t) + C_2x_2(t) + Du(t). \end{aligned} \tag{3.4}$$

3.2.1 Truncation

The k'th-order truncation of the realization $\Sigma = (A, B, C, D)$ is given by $\Sigma_k = (A_{11}, B_1, C_1, D)$. The truncated model Σ_k is equal to Σ at infinite frequency:

$$\begin{aligned} G(s) &= C(sI - A)^{-1}B + D, \\ G_k(s) &= C_1(sI - A_{11})^{-1}B_1 + D, \\ \implies G(\infty) &= G_k(\infty) = D. \end{aligned}$$

Apart from this, there is little to say about the relationship between Σ and Σ_k. If, however, A is in Jordan form, then it is easy to order the states so that x_2 corresponds to high-frequency or fast modes. For simplicity, assume that A has been diagonalized so that

$$A = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix}, \quad B = \begin{bmatrix} b_1^{\mathrm{T}} \\ b_2^{\mathrm{T}} \\ \vdots \\ b_n^{\mathrm{T}} \end{bmatrix} \text{ and } \quad C = \begin{bmatrix} c_1 & c_2 & \dots & c_n \end{bmatrix}.$$

Then, if the λ_i's are ordered such that $|\lambda_1| < |\lambda_2| < \cdots < |\lambda_n|$, the fastest modes are removed from the model after truncation. The difference between Σ and Σ_k following a k'th-order model truncation is given by

$$\begin{aligned} G(s) - G_k(s) &= C(Is - A)^{-1}B + D - \big(C_1(sI - A_{11})^{-1}B_1 + D\big) \\ &= C(Is - A)^{-1}B - C_1(sI - A_{11})^{-1}B_1 \\ &= \sum_{i=1}^{n} \frac{c_i b_i^{\mathrm{T}}}{s - \lambda_i} - \sum_{i=1}^{k} \frac{c_i b_i^{\mathrm{T}}}{s - \lambda_i} \\ &= \sum_{i=k+1}^{n} \frac{c_i b_i^{\mathrm{T}}}{s - \lambda_i} \end{aligned}$$

and therefore, by following the definition of the $\mathcal{L}_\infty$-norm (3.2),

$$\begin{aligned}
||\Sigma - \Sigma_k||_{\mathcal{L}_\infty} &= \sup_{\omega\in\mathbb{R}} \bar{\sigma}\big(G(\mathrm{i}\omega) - G_k(\mathrm{i}\omega)\big) = \sup_{\omega\in\mathbb{R}} \left|\bar{\sigma}\left(\sum_{i=k+1}^{n} \frac{c_i b_i^{\mathrm{T}}}{\mathrm{i}\omega - \lambda_i}\right)\right| \\
&\leq \sup_{\omega\in\mathbb{R}} \sum_{i=k+1}^{n} \left|\frac{\bar{\sigma}(c_i b_i^{\mathrm{T}})}{\mathrm{i}\omega - \lambda_i}\right| = \sup_{\omega\in\mathbb{R}} \sum_{i=k+1}^{n} \frac{\bar{\sigma}(c_i b_i^{\mathrm{T}})}{|\mathrm{i}\omega - \lambda_i|} \\
&= \sup_{\omega\in\mathbb{R}} \sum_{i=k+1}^{n} \frac{\bar{\sigma}(c_i b_i^{\mathrm{T}})}{\sqrt{\mathrm{Re}(-\lambda_i)^2 + \mathrm{Im}(\omega - \lambda_i)^2}} \\
&\leq \sum_{i=k+1}^{n} \frac{\bar{\sigma}(c_i b_i^{\mathrm{T}})}{|\mathrm{Re}(\lambda_i)|}.
\end{aligned} \tag{3.5}$$

Note that, from (3.5) it follows that the error depends on the residues $c_i b_i^{\mathrm{T}}$ as well as on the λ_i's. Therefore the distance of λ_i from the imaginary axis is by itself not a very reliable indicator of whether the associated mode should be included in the truncated model or not.
An advantage of model truncation is that the poles of Σ_k are a subset of the poles of Σ, so any physical interpretation of the original model can also be found in the truncated model.

3.2.2 Residualization

Consider the partitioned system in (3.4). Where in truncation, we discard all the states and dynamics associated with x_2, in residualization we instead simply set $\dot{x}_2 = 0$ (i.e. we residualize x_2) in the state-space equations. If we assume A_{22} to be invertible, we can solve $\dot{x}_2 = 0$ in terms of x_1 and u:

$$\begin{aligned}
&\dot{x}_2(t) = A_{21}x_1(t) + A_{22}x_2(t) + B_2u(t) = 0, \\
&\implies A_{22}x_2(t) = -A_{21}x_1(t) - B_2u(t), \\
&\implies x_2(t) = -A_{22}^{-1}A_{21}x_1(t) - A_{22}^{-1}B_2u(t).
\end{aligned}$$

Substituting x_2 back in the state-space equations gives

$$\begin{aligned}
\dot{x}_1(t) &= A_{11}x_1(t) + A_{12}x_2(t) + B_1u_1(t) \\
&= A_{11}x_1(t) + A_{12}\left(-A_{22}^{-1}A_{21}x_1(t) - A_{22}^{-1}B_2u(t)\right) + B_1u_1(t) \\
&= \left(A_{11} - A_{12}A_{22}^{-1}A_{21}\right)x_1(t) + \left(B_1 - A_{12}A_{22}^{-1}B_2\right)u(t),
\end{aligned}$$

$$\begin{aligned}
y(t) &= C_1x_1(t) + C_2x_2(t) + Du(t) \\
&= C_1x_1(t) + C_2\left(-A_{22}^{-1}A_{21}x_1(t) - A_{22}^{-1}B_2u(t)\right) + Du(t) \\
&= \left(C_1 - C_2A_{22}^{-1}A_{21}\right)x_1(t) + \left(D - C_2A_{22}^{-1}B_2\right)u(t).
\end{aligned}$$

If we define

$$\begin{aligned}
A_r &:= A_{11} - A_{12}A_{22}^{-1}A_{21}, \\
B_r &:= B_1 - A_{12}A_{22}^{-1}B_2, \\
C_r &:= C_1 - C_2A_{22}^{-1}A_{21}, \\
D_r &:= D - C_2A_{22}^{-1}B_2,
\end{aligned} \tag{3.6}$$

we obtain the residualized model $\Sigma_r = (A_r, B_r, C_r, D_r)$. Again, similar as in Section 3.2.1, assume that (A, B, C, D) has been put in Jordan form, with the eigenvalues ordered such that x_2 contains the fast modes. Model reduction by residualization is then equivalent to *singular pertubational approximation* [12], where the derivatives of the fastest states are allowed to approach

zero with some parameter ϵ. An important property of residualization is that it preserves the steady-state gain of the system: $G_r(0) = G(0)$. This is a very strong difference with truncation, which retains the system properties at infinite frequency. Therefore, truncation is preferred if we want accuracy at high frequencies and residualization if we want accuracy at low frequencies.

Both methods highly depend on the original realization. A possible realization is the balanced realization which will be considered next.

3.3 Balanced realizations

A *balanced realization* is an asymptotically stable minimal realization in which the Controllability and Observability Gramians are equal and diagonal. More formally:

Definition 3.5. Let (A, B, C, D) be a minimal realization of a stable, rational transfer function $G(s)$, then (A, B, C, D) is called *balanced* if the solutions to the following Lyapunov equations

$$\begin{aligned} AP + PA^T + BB^{\mathrm{T}} &= 0, \\ A^{\mathrm{T}}Q + QA + C^{\mathrm{T}}C &= 0 \end{aligned} \tag{3.7}$$

are $P = Q = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n) =: S$, where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n > 0$.

Any minimal realization of a stable transfer function can be balanced, see Subsection 3.3.1. The matrices P and Q are the Controllability and Observability Gramians, which are the solutions of (3.7) and can also be defined by

$$\begin{aligned} P &:= \int_0^\infty \mathrm{e}^{At} BB^{\mathrm{T}} \mathrm{e}^{A^{\mathrm{T}}t} \mathrm{d}t, \\ Q &:= \int_0^\infty \mathrm{e}^{A^{\mathrm{T}}t} C^{\mathrm{T}} C \mathrm{e}^{At} \mathrm{d}t. \end{aligned} \tag{3.8}$$

If the system Σ is balanced, then $P = Q = S$ and from now on we will simply call S the *Gramian* of $G(s)$. Note that the symbol S is also used for the direct feed-through in port-Hamiltonian systems. However, it will be made clear in the context whether we are talking about the system or about the direct feed-through. The σ_i's are the ordered Hankel singular values of $G(s)$. More on the Hankel singular values can be found in Subsection 3.3.2.

The Hankel singular values provide useful information about the contribution of the states on the input-output behaviour of the system. If σ_i is small, state x_i has a small role in the input-output behaviour, see Section 3.4 for more details.

3.3.1 Balancing the system

Example 3.1. Suppose the system (3.1) has the realization $A = \begin{bmatrix} -1 & -\frac{4}{\alpha} \\ 4\alpha & -2 \end{bmatrix}$, $B = \begin{bmatrix} 1 \\ 2\alpha \end{bmatrix}$, $C = \begin{bmatrix} -1 & \frac{2}{\alpha} \end{bmatrix}$ and $D = 0$, where α is a nonzero real number. It has transfer function

$$G(s) = \frac{3s + 18}{s^2 + 3s + 18}.$$

It is easy to check that the Controllability Gramian of the realization is given by

$$P = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \alpha^2 \end{bmatrix}.$$

Since the last diagonal term of P can be made arbitrarily small by making α small, the controllability of the last state can be made arbitrarily weak. If we would remove this state, we get $A = -1$, $B = 1$ and $C = -1$ with transfer function

$$G(s) = \frac{-1}{s+1},$$

which is not close to the original one at all. To get a better understanding of the problem, we check the Observability Gramian as well:

$$Q = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{\alpha^2} \end{bmatrix}.$$

Now we can see that if we make α small, the last diagonal term of Q becomes large and therefore the last state becomes very observable (i.e. contributes more and more to the output).

This example shows that P or Q alone cannot give an accurate indication of the impact of the states on the system. This motivates the introduction of a *balanced realization*, where P and Q are equal. We will now explain how to find a realization such that it is balanced.

Suppose we do the state transformation by the nonsingular $T \in \mathbb{R}^{n\times n}$, i.e. $\hat{x}(t) = Tx(t)$. The realization is as follows

$$\begin{aligned} &\begin{cases} \dot{x}(t) = Ax(t) + Bu(t), \\ y(t) = Cx(t) + Du(t), \end{cases} \\ \Longrightarrow &\begin{cases} T^{-1}\dot{\hat{x}}(t) = AT^{-1}\hat{x}(t) + Bu(t), \\ y(t) = CT^{-1}\hat{x}(t) + Du(t), \end{cases} \\ \Longrightarrow &\begin{cases} \dot{\hat{x}}(t) = TAT^{-1}\hat{x}(t) + TBu(t), \\ y(t) = CT^{-1}\hat{x}(t) + Du(t), \end{cases} \end{aligned}$$

which gives the transformed system

$$\begin{aligned} \dot{\hat{x}}(t) &= \hat{A}\hat{x}(t) + \hat{B}u(t), \\ y(t) &= \hat{C}\hat{x}(t) + \hat{D}u(t), \end{aligned} \tag{3.9}$$

with $\hat{A} = TAT^{-1}$, $\hat{B} = TB$, $\hat{C} = CT^{-1}$ and $\hat{D} = D$. Then the Gramians are transformed to $\hat{P} = TPT^{\mathrm{T}}$ and $\hat{Q} = T^{-\mathrm{T}}QT^{-1}$. To see this for $\hat{P}$, substitute (3.9) and $\hat{P} = TPT^{\mathrm{T}}$ into (3.7) to get

$$\begin{aligned} \hat{A}\hat{P} + \hat{P}\hat{A}^{\mathrm{T}} + \hat{B}\hat{B}^{\mathrm{T}} &= TAT^{-1}TPT^{\mathrm{T}} + TPT^{\mathrm{T}}T^{-\mathrm{T}}A^{\mathrm{T}}T^{\mathrm{T}} + TBB^{\mathrm{T}}T^{\mathrm{T}} \\ &= TAPT^{\mathrm{T}} + TPA^{\mathrm{T}}T^{\mathrm{T}} + TBB^{\mathrm{T}}T^{\mathrm{T}} \\ &= T\left(AP + PA^{\mathrm{T}} + BB^{\mathrm{T}}\right)T^{\mathrm{T}} = 0. \end{aligned} \tag{3.10}$$

The last part is zero since P is the solution of (3.7), so $\hat{P}$ is the Controllability Gramian of the transformed realization. The same can be shown for $\hat{Q}$. Note that $\hat{P}\hat{Q} = TPQT^{-1}$ and therefore the eigenvalues of the product of the Gramians are invariant under a state transformation.

Now consider another, similar state transformation with nonsingular $T \in \mathbb{R}^{n\times n}$ such that T gives the eigenvector decomposition

$$PQ = T^{-1}\Lambda T, \qquad \Lambda = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n).$$

Then the columns of T^{-1} are eigenvectors of PQ corresponding to the eigenvalues $\{\lambda_i\}$. The only thing left is to make sure that the new realization is balanced. If the system (3.1) is a minimal realization, a balanced realization obtained by the state transformation $\hat{x}(t) = Tx(t)$ can be found by the following simplified procedure [13]:

1. Compute the Controllability and Observability Gramians $P > 0$, $Q > 0$,
2. Find a matrix R such that $P = R^{\mathrm{T}}R$,
3. Diagonalize RQR^{T} to get $RQR^{\mathrm{T}} = US^2U^{\mathrm{T}}$,
4. Let $T^{-1} = R^{\mathrm{T}}US^{-\frac{1}{2}}$. Then $TPT^{\mathrm{T}} = S = T^{-\mathrm{T}}QT^{-1}$ and thus $(\hat{A}, \hat{B}, \hat{C}, \hat{D}) = (TAT^{-1}, TB, CT^{-1}, D)$ is balanced.

In step 4, $S = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n)$ and $S^2 = \Lambda$. Assuming $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$ and assume that $\sigma_r \gg \sigma_{r+1}$ for some r, then the balanced realization implies that the states corresponding to the singular values $\sigma_{r+1}, \ldots, \sigma_n$ are less controllable and observable than the others. Therefore, removing the ones which are less controllable and observable will not lead to a large loss in the information about the system.

3.3.2 Hankel norm and singular values

The *Hankel norm* of a stable system Σ is obtained when one applies an input $u(t)$ up to $t = 0$, measures the output $y(t)$ for $t > 0$ and selects $u(t)$ such that the ratio of the $\mathcal{L}_2$-norm is maximal.

Definition 3.6. Let Σ be a stable system with input signal $u(t)$ and output signal $y(t)$. The *Hankel norm* is defined as

$$||\Sigma||_{\mathrm{H}} := \max_{u(t), u(t)=0, t>0} \frac{\sqrt{\int_0^\infty ||y(\tau)||_2^2 \mathrm{d}\tau}}{\sqrt{\int_{-\infty}^0 ||u(\tau)||_2^2 \mathrm{d}\tau}}.$$

It can be shown [13, Theorem 8.1] that the Hankel norm is equal to

$$||\Sigma||_{\mathrm{H}} = \max_i \sqrt{\lambda_i(PQ)},$$

where P and Q are as defined in (3.8). The corresponding *Hankel singular values* are the positive square roots of the eigenvalues of PQ,

$$\sigma_i := \sqrt{\lambda_i(PQ)}.$$

If these σ_i's are ordered such that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$, the Hankel- and $\mathcal{H}_\infty$-norm are closely related as follows [13]:

$$||\Sigma||_{\mathrm{H}} = \sigma_1 \leq ||\Sigma||_{\mathcal{H}_\infty} \leq 2\sum_{i=1}^{n} \sigma_i. \tag{3.11}$$

3.4 Balanced truncation and balanced residualization

Let (A, B, C, D) be a balanced realization of Σ, partitioned as in (3.4) together with the appropriate partitioning of S,

$$S = \begin{bmatrix} S_1 & 0 \\ 0 & S_2 \end{bmatrix},$$

where $S_1 = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_k)$, $S_2 = \mathrm{diag}(\sigma_{k+1}, \sigma_{k+2}, \ldots, \sigma_n)$ and $\sigma_k \geq \sigma_{k+1}$.

3.4.1 Balanced truncation

The reduced order model given by (A_{11}, B_1, C_1, D) in equation (3.4) is called a *balanced tuncation* of the system Σ, if Σ is a balanced realization. A balanced truncation is also a balanced realization [14] and from (3.11) it follows that the $\mathcal{H}_\infty$-norm of the error between Σ and Σ_k is bounded by twice the sum of the last $n-k$ Hankel singular values (i.e. twice the trace of Σ_k).

$$||\Sigma - \Sigma_k||_{\mathcal{L}_\infty} \leq 2 \sum_{i=k+1}^{n} \sigma_i. \tag{3.12}$$

A precise statement of the bound on the approximation error is given in Theorem 3.2 below.

3.4.2 Balanced residualization

The difference between balanced residualization and balanced truncation is the same as the difference between residualization and truncation. Where in balanced truncation the last $n-k$ states are discarded, in balanced residualization the derivatives of these last $n-k$ states are set to zero. In this way the *balanced residualization* of the balanced system Σ is $\Sigma_r = (A_r, B_r, C_r, D_r)$, where A_r, B_r, C_r, D_r are defined as in (3.6).

It is shown in [12] that balanced residualization has the same error bound (3.12) as balanced truncation. A precise statement of the error bound is given in the following theorem:

Theorem 3.2 ([10], Theorem 11.1)**.** *Let $G(s)$ be a stable rational transfer function with Hankel singular values $\sigma_1 > \sigma_2 > \cdots > \sigma_N$ where each σ_i has multiplicity r_i and let $G_a^k(s)$ be obtained by truncating or residualizing the balanced realization of $G(s)$ to the first $(r_1 + r_2 + \cdots + r_k)$ states. Then*

$$||G(s) - G_a^k(s)||_{\mathcal{L}_\infty} \leq 2 \sum_{i=k+1}^{N} \sigma_i.$$

4 Model reduction of descriptor systems

In Chapter 3 we have introduced model reduction for standard systems. In this chapter we will generealize balanced truncation of standard systems to balanced truncation of descriptor systems. In order to do so, we will have to reshape classical system analysis results, such as Lyapunov equations, Controllability and Observability Gramians and balanced realizations. Recall the LTI-system Σ (1.1):

$$\begin{aligned} E\dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t), \end{aligned} \tag{4.1}$$

with $E, A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{m\times n}$ and $D \in \mathbb{R}^{m\times m}$ given. Σ is called a descripor system if E is not invertible and a strandard system if $E = I$. The results in this chapter when Σ would be a standard system are the same as the results in Chapter 3.

System (4.1) can also be described by its *transfer function*

$$G(s) = C(Es - A)^{-1}B + D,$$

with $s \in \mathbb{C}$.

In this chapter we are interested in finding an approximation Σ_a for Σ given in (4.1) with reduced order k:

$$\begin{aligned} E_a\dot{x}(t) &= A_a x(t) + B_a u(t), \\ y(t) &= C_a x(t) + D_a u(t), \end{aligned}$$

with $E_a, A_a \in \mathbb{R}^{k\times k}$, $B_a \in \mathbb{R}^{k\times m}$, $C_a \in \mathbb{R}^{m\times k}$ and $D_a \in \mathbb{R}^{m\times m}$. Note that the input is the same in both systems. In [15] an extension of the well-known balanced truncation to descriptor systems (4.1) is given, which is closely related to the two Lyapunov equations (3.7). This chapter is based on the results of [15]. We start in Section 4.1 with generalizing the classical system analysis results to the notion of descriptor systems. These results will be used in Section 4.2 to introduce balanced realizations. When we know how to create balanced realizations of descriptor systems, we can truncate the system in Section 4.3 to obtain a balanced truncation, and present an algorithm.

4.1 Classical results of standard systems extended to descriptor systems

We refer to Definitions 2.2 – 2.4 for the notion of *regular* and *singular* systems, the *(finite) eigenvalues* of the matrix pencil $\lambda E - A$, *(asymptotically) stability*, and *(minimal) realizations*.

We assume in this chapter that the pencil $\lambda E - A$ is regular. In this case $\lambda E - A$ can be reduced to the *Weierstrass canonical form* [16], that is, there exist nonsingular matrices $W, T \in \mathbb{R}^{n\times n}$ such that

$$E = W \begin{bmatrix} I_{n_f} & 0 \\ 0 & N \end{bmatrix} T \quad \text{and} \quad A = W \begin{bmatrix} J & 0 \\ 0 & I_{n_\infty} \end{bmatrix} T, \tag{4.2}$$

where $J \in \mathbb{C}^{n_f\times n_f}$ is a matrix in *Jordan canonical form* associated with the n_f finite eigenvalues (including the multiplicity) of the pencil $\lambda E - A$ and $N \in \mathbb{C}^{n_\infty\times n_\infty}$ is a nilpotent matrix in Jordan canonical form corresponding to the n_∞ eigenvalues (including the multiplicity) of ∞. If $n_f < n$ and N has degree of nilpotency ν ($N^\nu = 0$ and $N^i \neq$ for $i < \nu$), then ν is called the *index* of the pencil $\lambda E - A$. We discuss now the generalization to the regular pencil case of the notion of invariant subspace of a matrix.

Definition 4.1. Let $\lambda E - A$ be a regular matrix pencil, with $E, A \in \mathbb{R}^{n\times n}$. The linear spaces $\mathcal{V}, \mathcal{W} \subset \mathbb{R}^n$ are called *deflating subspaces* of $\lambda E - A$ if we have

$$Ev \in \mathcal{W} \quad \text{and} \quad Av \in \mathcal{W}$$

for all $v \in \mathcal{V}$ and

$$\operatorname*{span}_{v\in\mathcal{V}} \{Ev, Av\} = \mathcal{W}.$$

Moreover, $\mathcal{V}$ is called the *right deflating subspace* and $\mathcal{W}$ is called the *left deflating subspace* of $\lambda E - A$.

We refer to [17] for more information about deflating subspaces. The matrices

$$\mathcal{P}_r := T^{-1}\begin{bmatrix} I_{n_f} & 0 \\ 0 & 0 \end{bmatrix} T \quad \text{and} \quad \mathcal{P}_l := W\begin{bmatrix} I_{n_f} & 0 \\ 0 & 0 \end{bmatrix} W^{-1}, \tag{4.3}$$

with W and T as in (4.2), are the *spectral projections* onto the right and left deflating subspaces of $\lambda E - A$ corresponding to the finite eigenvalues [18].

Consider the LTI system (4.1). If the pencil $\lambda E - A$ is regular, $u(t)$ is sufficiently smooth and the initial solution x_0 is *consistent* (i.e. if (4.1) together with x_0 has at least one solution), then (4.1) has a unique continuously differentiable solution $x(t)$, see [4], given by

$$x(t) = \mathcal{F}(t)Ex_0 + \int_0^t \mathcal{F}(t-\tau)Bu(\tau)\mathrm{d}\tau + \sum_{k=0}^{\nu-1} \mathcal{F}_{-k-1}Bu^{(k)}(t). \tag{4.4}$$

Here

$$\mathcal{F}(t) := T^{-1}\begin{bmatrix} \mathrm{e}^{tJ} & 0 \\ 0 & 0 \end{bmatrix} W^{-1} \tag{4.5}$$

is the *fundamental solution matrix* of the descriptor system (4.1) and the matrices F_k have the form

$$\mathcal{F}_k := T^{-1}\begin{bmatrix} 0 & 0 \\ 0 & -N^{-k-1} \end{bmatrix} W^{-1}, \quad k = -1, -2, \ldots. \tag{4.6}$$

Note that $\mathcal{F}_k = 0$ for $k < -\nu$, since $N^\nu = 0$. Two realizations (E, A, B, C, D) and $(\tilde{E}, \tilde{A}, \tilde{B}, \tilde{C}, \tilde{D})$ are called *equivalent* if there exist invertible matrices $\tilde{W}$ and $\tilde{T}$ such that

$$\begin{aligned} \tilde{E} &= \tilde{W}E\tilde{T}, \\ \tilde{A} &= \tilde{W}A\tilde{T}, \\ \tilde{B} &= \tilde{W}B, \\ \tilde{C} &= C\tilde{T}, \\ \tilde{D} &= D. \end{aligned} \tag{4.7}$$

4.1.1 Controllability and Observability Gramians

We can split descriptor systems into two parts: the *proper* and *improper* part. The proper part, associated with the finite eigenvalues, behaves as standard systems, where the improper part, associated with the infinite eigenvalues, has a different behaviour. This explains the two different terms in the solution $x(t)$ in (4.4). Therefore, the Controllability and Observability Gramians of the proper and improper parts are also different. They are given in this subsection.

Proper part
Assume that the pencil $\lambda E - A$ is asymptotically stable. Then the integrals

$$P_p := \int_0^\infty \mathcal{F}(t) B B^{\mathrm{T}} \mathcal{F}^{\mathrm{T}}(t) \mathrm{d}t$$

and

$$Q_p := \int_0^\infty \mathcal{F}^{\mathrm{T}}(t) C^{\mathrm{T}} C \mathcal{F}(t) \mathrm{d}t$$

exist, where $\mathcal{F}(t)$ is given in (4.5). The matrix P_p is called the *proper Controllability Gramian* and the matrix Q_p is called the *proper Observability Gramian* of the descriptor system (4.1), see [18] and [19].

If $E = I$, then P_p and Q_p are the usual Controllability and Observability Gramians for standard systems. P_p and Q_p are the unique symmetric, positive semidefinite solutions of the *projected generalized continuous-time Lyapunov equations*

$$\begin{aligned} E P_p A^{\mathrm{T}} + A P_p E^{\mathrm{T}} &= -\mathcal{P}_l B B^{\mathrm{T}} \mathcal{P}_l^{\mathrm{T}}, \\ P_p &= \mathcal{P}_r P_p \end{aligned} \tag{4.8}$$

and

$$\begin{aligned} E^{\mathrm{T}} Q_p A + A^{\mathrm{T}} Q_p E &= -\mathcal{P}_r^{\mathrm{T}} C^{\mathrm{T}} C \mathcal{P}_r, \\ Q_p &= Q_p \mathcal{P}_l \end{aligned}$$

respectivily, where $\mathcal{P}_r$ and $\mathcal{P}_l$ are given in (4.3), see [18].

Proposition 4.1. *If $\lambda E - A$ is in Weierstrass canonical form* (4.2) *and if the matrices*

$$W^{-1} B = \begin{bmatrix} B_p \\ B_i \end{bmatrix} \quad \text{and} \quad C T^{-1} = \begin{bmatrix} C_p & C_i \end{bmatrix} \tag{4.9}$$

are partitioned in blocks conformally the proper and improper parts of E and A (such that the dimensions agree), then

$$P_p = T^{-1} \begin{bmatrix} G_p & 0 \\ 0 & 0 \end{bmatrix} T^{-\mathrm{T}}, \quad Q_p = W^{-\mathrm{T}} \begin{bmatrix} H_p & 0 \\ 0 & 0 \end{bmatrix} W^{-1}, \tag{4.10}$$

where G_p and H_p satisfy the standard continuous-time Lyapunov equations

$$\begin{aligned} J G_p + G_p J^{\mathrm{T}} &= -B_p B_p^{\mathrm{T}}, \\ J^{\mathrm{T}} H_p + H_p J &= -C_p^{\mathrm{T}} C_p. \end{aligned} \tag{4.11}$$

Proof. We only provide the part of the proof for P_p. The proof for Q_p is similar.

Suppose that P_p is the solution of (4.8). We want to show that P_p is of the form (4.10), where G_p has to satisfy (4.11). From the second equation of (4.8) and (4.3) we have

$$\begin{aligned} P_p &= \mathcal{P}_r P_p, \\ P_p &= T^{-1} \begin{bmatrix} I_{n_f} & 0 \\ 0 & 0 \end{bmatrix} T P_p, \\ T P_p &= \begin{bmatrix} I_{n_f} & 0 \\ 0 & 0 \end{bmatrix} T P_p, \end{aligned}$$

which means that the matrix TP_p must be of the form

$$TP_p = \begin{bmatrix} G_{p_1} & G_{p_2} \\ 0 & 0 \end{bmatrix}.$$

Multiplying the left and right side of the equation by T^{T} gives

$$\begin{aligned} TP_pT^{\mathrm{T}} &= \begin{bmatrix} G_{p_1} & G_{p_2} \\ 0 & 0 \end{bmatrix} T^{\mathrm{T}} \\ &= \begin{bmatrix} G_{p_1} & G_{p_2} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} T_1^{\mathrm{T}} & T_3^{\mathrm{T}} \\ T_2^{\mathrm{T}} & T_4^{\mathrm{T}} \end{bmatrix} \\ &= \begin{bmatrix} G_{p_1}T_1^{\mathrm{T}} + G_{p_2}T_2^{\mathrm{T}} & G_{p_1}T_3^{\mathrm{T}} + G_{p_2}T_4^{\mathrm{T}} \\ 0 & 0 \end{bmatrix}. \end{aligned}$$

The matrix TP_pT^{T} is again symmetric, hence $G_{p_1}T_3^{\mathrm{T}} + G_{p_2}T_4^{\mathrm{T}} = 0$ and if we write $G_p := G_{p_1}T_1^{\mathrm{T}} + G_{p_2}T_2^{\mathrm{T}}$, we get

$$\begin{aligned} TP_pT^{\mathrm{T}} &= \begin{bmatrix} G_p & 0 \\ 0 & 0 \end{bmatrix}, \\ \implies P &= T^{-1} \begin{bmatrix} G_p & 0 \\ 0 & 0 \end{bmatrix} T^{-\mathrm{T}}. \end{aligned} \tag{4.12}$$

Substituting (4.12) in the left-hand side of the first equation of (4.8) and using the Weierstrass canonical form for E and A (4.2) gives

$$\begin{aligned} EP_pA^{\mathrm{T}} + AP_pE^{\mathrm{T}} &= W \begin{bmatrix} I_q & 0 \\ 0 & N \end{bmatrix} TT^{-1} \begin{bmatrix} G_p & 0 \\ 0 & 0 \end{bmatrix} T^{-\mathrm{T}}T^{\mathrm{T}} \begin{bmatrix} J & 0 \\ 0 & I_{n-q} \end{bmatrix}^{\mathrm{T}} W^{\mathrm{T}} \\ &\quad + W \begin{bmatrix} J & 0 \\ 0 & I_{n-q} \end{bmatrix} TT^{-1} \begin{bmatrix} G_p & 0 \\ 0 & 0 \end{bmatrix} T^{-\mathrm{T}}T^{\mathrm{T}} \begin{bmatrix} I_q & 0 \\ 0 & N \end{bmatrix}^{\mathrm{T}} W^{\mathrm{T}} \\ &= W \left(\begin{bmatrix} I_q & 0 \\ 0 & N \end{bmatrix} \begin{bmatrix} G_p & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} J & 0 \\ 0 & I_{n-q} \end{bmatrix}^{\mathrm{T}} + \begin{bmatrix} J & 0 \\ 0 & I_{n-q} \end{bmatrix} \begin{bmatrix} G_p & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} I_q & 0 \\ 0 & N \end{bmatrix}^{\mathrm{T}} \right) W^{\mathrm{T}} \\ &= W \begin{bmatrix} G_pJ^{\mathrm{T}} + JG_p & 0 \\ 0 & 0 \end{bmatrix} W^{\mathrm{T}}. \end{aligned} \tag{4.13}$$

Writing out the right-hand side of the first equation of (4.8) with $\mathcal{P}_l$ defined in (4.3) gives

$$EP_pA^{\mathrm{T}} + AP_pE^{\mathrm{T}} = -W \begin{bmatrix} B_pB_p^{\mathrm{T}} & 0 \\ 0 & 0 \end{bmatrix} W^{\mathrm{T}}.$$

Now evaluate the right-hand side of (4.8). Equation (4.3) with the right-hand side of (4.8) gives

$$-\mathcal{P}_lBB^{\mathrm{T}}\mathcal{P}_l^{\mathrm{T}} = -W \begin{bmatrix} I_q & 0 \\ 0 & 0 \end{bmatrix} W^{-1}BB^{\mathrm{T}}W^{-\mathrm{T}} \begin{bmatrix} I_q & 0 \\ 0 & 0 \end{bmatrix}^{\mathrm{T}} W^{\mathrm{T}}.$$

Equation (4.9) for $W^{-1}B$ gives

$$\begin{aligned} -\mathcal{P}_lBB^{\mathrm{T}}\mathcal{P}_l^{\mathrm{T}} &= -W \begin{bmatrix} I_q & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} B_p \\ B_i \end{bmatrix} \left(W^{-1}B\right)^{\mathrm{T}} \begin{bmatrix} I_q & 0 \\ 0 & 0 \end{bmatrix}^{\mathrm{T}} W^{\mathrm{T}} \\ &= -W \begin{bmatrix} B_p & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} B_p \\ B_i \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} I_q & 0 \\ 0 & 0 \end{bmatrix}^{\mathrm{T}} W^{\mathrm{T}} \\ &= -W \begin{bmatrix} B_pB_p^{\mathrm{T}} & 0 \\ 0 & 0 \end{bmatrix} W^{\mathrm{T}}, \end{aligned}$$

Equations (4.12) and (4.13) must equal eachother, which gives that G_p must satisfy

$$JG_p + G_pJ^{\mathrm{T}} = -B_pB_p^{\mathrm{T}}.$$

Hence, we conclude that

$$P_p = T^{-1}\begin{bmatrix} G_p & 0 \\ 0 & 0 \end{bmatrix} T^{-\mathrm{T}},$$

where G_p satisfies the standard continuous-time Lyapunov equation

$$JG_p + G_pJ^{\mathrm{T}} = -B_pB_p^{\mathrm{T}}.$$

□

Improper part
The *improper Controllability Gramian* of the descriptor system (4.1) is defined by

$$P_i := \sum_{k=-\nu}^{-1} \mathcal{F}_k B B^{\mathrm{T}} \mathcal{F}_k^{\mathrm{T}},$$

and the *improper Observability Gramian* of the descriptor system (4.1) is defined by

$$Q_i := \sum_{k=-\nu}^{-1} \mathcal{F}_k^{\mathrm{T}} C^{\mathrm{T}} C \mathcal{F}_k,$$

where $\mathcal{F}_k$ are defined in (4.6). P_i and Q_i are the unique symmetric, positive semidefinite solutions of the *projected generalized discrete-time Lyapunov equations*

$$\begin{aligned} AP_iA^{\mathrm{T}} - EP_iE^{\mathrm{T}} &= (I - \mathcal{P}_l)BB^{\mathrm{T}}(I - \mathcal{P}_l)^{\mathrm{T}}, \\ \mathcal{P}_r P_i &= 0 \end{aligned}$$

and

$$\begin{aligned} A^{\mathrm{T}}Q_iA - E^{\mathrm{T}}Q_iE &= (I - \mathcal{P}_r)^{\mathrm{T}}C^{\mathrm{T}}C(I - \mathcal{P}_r), \\ Q_i\mathcal{P}_l &= 0, \end{aligned}$$

respectively, where $\mathcal{P}_r$ and $\mathcal{P}_l$ are given in (4.3), see [20].

Proposition 4.2. *If $\lambda E - A$ is in Weierstrass canonical form* (4.2) *and if the matrices $W^{-1}B$ and CT^{-1} are partitioned as in* (4.11)*, then*

$$P_i = T^{-1}\begin{bmatrix} 0 & 0 \\ 0 & G_i \end{bmatrix} T^{-\mathrm{T}}, \quad Q_i = W^{-\mathrm{T}}\begin{bmatrix} 0 & 0 \\ 0 & H_i \end{bmatrix} W^{-1},$$

where G_i and H_i satisfy the standard discrete-time Lyapunov equations

$$\begin{aligned} G_i - NG_iN^{\mathrm{T}} &= B_iB_i^{\mathrm{T}}, \\ H_i - N^{\mathrm{T}}H_iN &= C_i^{\mathrm{T}}C_i. \end{aligned}$$

Proof. The outline of this proof is the same as in proposition 4.1 and is therefore omitted. □

4.1.2 Hankel singular values

For standard systems the Controllability and Observability Gramians are not system invariant, but the product of both Gramians is system invariant. The same holds for the proper and improper Controllability and Observability Gramians: under the system equivalence transformation $(\tilde{W}, \tilde{T})$, see (4.7), the Controllability Gramians are transformed to

$$\begin{aligned}\tilde{P}_p &= \tilde{T}^{-1} P_p \tilde{T}^{-\mathrm{T}},\\ \tilde{P}_i &= \tilde{T}^{-1} P_i \tilde{T}^{-\mathrm{T}},\end{aligned}$$

whereas the Observability Gramians are transformed to

$$\begin{aligned}\tilde{Q}_p &= \tilde{W}^{-\mathrm{T}} Q_p \tilde{W}^{-1},\\ \tilde{Q}_i &= \tilde{W}^{-\mathrm{T}} Q_i \tilde{W}^{-1}.\end{aligned}$$

To see why this is true, follow the same steps as in (3.10). So we can conclude that the Gramians of a descriptor system are not system invariant. However, it follows from

$$\begin{aligned}\tilde{P}_p \tilde{E}^{\mathrm{T}} \tilde{Q}_p \tilde{E} &= \tilde{T}^{-1} P_p \tilde{T}^{-\mathrm{T}} \tilde{T}^{\mathrm{T}} E^{\mathrm{T}} \tilde{W}^{\mathrm{T}} \tilde{W}^{-\mathrm{T}} Q_p \tilde{W}^{-1} \tilde{W} E \tilde{T}\\ &= \tilde{T}^{-1} P_p E^{\mathrm{T}} Q_p E \tilde{T},\\ \tilde{P}_i \tilde{A}^{\mathrm{T}} \tilde{Q}_i \tilde{A} &= \tilde{T}^{-1} P_i \tilde{T}^{-\mathrm{T}} \tilde{T}^{\mathrm{T}} A^{\mathrm{T}} \tilde{W}^{\mathrm{T}} \tilde{W}^{-\mathrm{T}} Q_i \tilde{W}^{-1} \tilde{W} A \tilde{T}\\ &= \tilde{T}^{-1} P_i A^{\mathrm{T}} Q_i A \tilde{T}\end{aligned}$$

that the spectra of the matrices $P_p E^{\mathrm{T}} Q_p E$ and $P_i A^{\mathrm{T}} Q_i A$ are system invariant. These matrices play the same role for descriptor systems as the product of the Gramians for standard state space systems.

Theorem 4.1 ([15], Theorem 2.6)**.** *Let $\lambda E - A$ be asymptotically stable. Then the matrices $P_p E^{\mathrm{T}} Q_p E$ and $P_i A^{\mathrm{T}} Q_i A$ have real, non-negative eigenvalues.*

Definition 4.2. Let n_f and n_∞ be the dimensions of the deflating subspaces of the asymptotically stable pencil $\lambda E - A$ corresponding to the finite and infinite eigenvalues, respectively. The square roots of the n_f largest eigenvalues of the matrix $P_p E^{\mathrm{T}} Q_p E$, denoted by ς_j, are called the *proper Hankel singular values* of the descriptor system (4.1). The square roots of the n_∞ largest eigenvalues of the matrix $P_i A^{\mathrm{T}} Q_i A$, denoted by θ_j, are called the *improper Hankel singular values* of the descriptor system (4.1).

We assume that the proper and improper Hankel singular values are ordered decreasingly, i.e.

$$\varsigma_1 \geq \varsigma_2 \geq \varsigma_3 \geq \cdots \geq \varsigma_{n_f}, \qquad \theta_1 \geq \theta_2 \geq \theta_3 \geq \cdots \geq \theta_{n_\infty}.$$

The proper and improper Hankel singular values form the set of Hankel singular values of the descriptor system (4.1). For $E = I$, the proper Hankel singular values are the classical Hankel singular values of standard state space systems [21], [22].

Since the proper and improper Controllability and Observability Gramians are symmetric and positive semidefinite, there exist Cholesky factorizations

$$\begin{aligned}P_p &= R_p R_p^{\mathrm{T}}, \quad & Q_p &= L_p^{\mathrm{T}} L_p,\\ P_i &= R_i R_i^{\mathrm{T}}, \quad & Q_i &= L_i^{\mathrm{T}} L_i,\end{aligned} \tag{4.14}$$

where $R_p^{\mathrm{T}}, L_p, R_i^{\mathrm{T}}, L_i \in \mathbb{R}^{n \times n}$ are upper triangular matrices. The following lemma gives a connection between the proper and improper Hankel singular values of system (4.1) and the standard singular values of the matrices $L_p E R_p$ and $L_i A R_i$.

Lemma 4.1 ([15], Lemma 2.8)**.** *Suppose the pencil $\lambda E - A$ is asymptotically stable. Consider the Cholesky factorizations* (4.14) *of the proper and improper Gramians of system* (4.1)*. Then the proper Hankel singular values of system* (4.1) *are the n_f largest singular values of the matrix L_pER_p and the improper Hankel singular values of system* (4.1) *are the n_∞ largest singular values of the matrix L_iAR_i.*

4.2 Balanced realizations

In Section 3.3 we explained how to find a balanced realization for standard state space systems. In this section we extend this to descriptor systems. We refer the reader to Definition 2.4 for the definition of *realization* and *minimal realization.*

Definition 4.3. A realization (E, A, B, C, D) of the transfer function $G(s)$ is called *balanced* if the solutions to the generalized Lyapunov equations, P_p, Q_p, P_i and Q_i, satisfy

$$P_p = Q_p = \begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad P_i = Q_i = \begin{bmatrix} 0 & 0 \\ 0 & \Theta \end{bmatrix} \tag{4.15}$$

with $S = \operatorname{diag}(\varsigma_1, \varsigma_2, \ldots, \varsigma_{n_f})$ and $\Theta = \operatorname{diag}(\theta_1, \theta_2, \ldots, \theta_{n_\infty})$.

We will now show that for a minimal realization (E, A, B, C, D) with the pencil $\lambda E - A$ being asymptotically stable, there exists a system equivalence transformation (W_b^{T}, T_b) such that the realization

$$\Sigma_b = (W_b^{\mathrm{T}} E T_b, W_b^{\mathrm{T}} A T_b, W_b^{\mathrm{T}} B, C T_b, D) \tag{4.16}$$

is balanced.
Consider the Cholesky factorizations (4.14) of the Gramians. We may assume without loss of generality that the matrices $R_p^{\mathrm{T}}, L_p, R_i^{\mathrm{T}}$ and L_i have full row rank. If (E, A, B, C, D) is minimal, it follows from Lemma 4.1 that $\varsigma_j = \sigma_j(L_pER_p) > 0, j = 1, 2, \ldots, n_f$, and $\theta_j = \sigma_j(L_iAR_i) > 0, j = 1, 2, \ldots, n_\infty$ [15]. Hence, the matrices L_pER_p and L_iAR_i are nonsingular. Let

$$L_pER_p = U_pSV_p^{\mathrm{T}}, \quad L_iAR_i = U_i\Theta V_i^{\mathrm{T}} \tag{4.17}$$

be singular value decompositions of L_pER_p and L_iAR_i, where $U_p, V_p \in \mathbb{R}^{n\times n_f}$ and $U_i, V_i \in \mathbb{R}^{n\times n_\infty}$ are orthogonal, $S = \operatorname{diag}(\varsigma_1, \varsigma_2, \ldots, \varsigma_{n_f})$ and $\Theta = \operatorname{diag}(\theta_1, \theta_2, \ldots, \theta_{n_\infty})$ are nonsingular. Consider the matrices

$$W_b = \begin{bmatrix} L_p^{\mathrm{T}} U_p S^{-\frac{1}{2}} & L_i^{\mathrm{T}} U_i \Theta^{-\frac{1}{2}} \end{bmatrix} \in \mathbb{R}^{n\times n}, \quad W_b' = \begin{bmatrix} E R_p V_p S^{-\frac{1}{2}} & A R_i V_i \Theta^{-\frac{1}{2}} \end{bmatrix} \in \mathbb{R}^{n\times n}$$

and

$$T_b = \begin{bmatrix} R_p V_p S^{-\frac{1}{2}} & R_i^{\mathrm{T}} V_i \Theta^{-\frac{1}{2}} \end{bmatrix} \in \mathbb{R}^{n\times n}, \quad T_b' = \begin{bmatrix} E^{\mathrm{T}} L_p^{\mathrm{T}} U_p S^{-\frac{1}{2}} & A^{\mathrm{T}} L_i^{\mathrm{T}} U_i \Theta^{-\frac{1}{2}} \end{bmatrix} \in \mathbb{R}^{n\times n}. \tag{4.18}$$

From [15] it follows that

$$L_pER_i = 0 \quad \text{and} \quad L_iAR_p = 0. \tag{4.19}$$

Then

$$(T_b')^{\mathrm{T}} T_b = \begin{bmatrix} S^{-\frac{1}{2}} U_p^{\mathrm{T}} L_p E R_p V_p S^{-\frac{1}{2}} & S^{-\frac{1}{2}} U_p^{\mathrm{T}} L_p E R_i V_i \Theta^{-\frac{1}{2}} \\ \Theta^{-\frac{1}{2}} U_i^{\mathrm{T}} L_i A R_p V_p S^{-\frac{1}{2}} & \Theta^{-\frac{1}{2}} U_i^{\mathrm{T}} L_i A R_i V_i \Theta^{-\frac{1}{2}} \end{bmatrix} = I_n,$$

i.e., the matrices T_b and T_b' are nonsingular and $(T_b')^{\mathrm{T}} = T_b^{-1}$. In the same way, we can show that the matrices W_b and W_b' are nonsingular and $(W_b')^{\mathrm{T}} = W_b^{-1}$. Using equations (4.14) and

(4.17)–(4.19), we obtain that the Controllability Gramian of the transformed system (4.16) has the form of (4.15):

$$\begin{aligned}
T_b^{-1}P_pT_b^{-\mathrm{T}} &= (T_b')^{\mathrm{T}}R_pR_p^{\mathrm{T}}T_b' \\
&= \begin{bmatrix} S^{-\frac{1}{2}}U_p^{\mathrm{T}}L_pE \\ \Theta^{-\frac{1}{2}}U_i^{\mathrm{T}}L_iA \end{bmatrix} R_pR_p^{\mathrm{T}} \begin{bmatrix} E^{\mathrm{T}}L_p^{\mathrm{T}}U_pS^{-\frac{1}{2}} & A^{\mathrm{T}}L_i^{\mathrm{T}}U_i\Theta^{-\frac{1}{2}} \end{bmatrix} \\
&= \begin{bmatrix} S^{-\frac{1}{2}}U_p^{\mathrm{T}}L_pER_pR_p^{\mathrm{T}}E^{\mathrm{T}}L_p^{\mathrm{T}}U_pS^{-\frac{1}{2}} & S^{-\frac{1}{2}}U_p^{\mathrm{T}}L_pER_pR_p^{\mathrm{T}}A^{\mathrm{T}}L_i^{\mathrm{T}}U_i\Theta^{-\frac{1}{2}} \\ \Theta^{-\frac{1}{2}}U_i^{\mathrm{T}}L_iAR_pR_p^{\mathrm{T}}E^{\mathrm{T}}L_p^{\mathrm{T}}U_pS^{-\frac{1}{2}} & \Theta^{-\frac{1}{2}}U_i^{\mathrm{T}}L_iAR_pR_p^{\mathrm{T}}A^{\mathrm{T}}L_i^{\mathrm{T}}U_i\Theta^{-\frac{1}{2}} \end{bmatrix}
\end{aligned}$$

Using $L_pER_p = U_pSV_p^{\mathrm{T}}$ (4.17) twice for the top-left part and $L_iAR_p = 0$ (4.19) for the other three elements gives

$$\begin{aligned}
T_b^{-1}P_pT_b^{-\mathrm{T}} &= \begin{bmatrix} S^{-\frac{1}{2}}U_p^{\mathrm{T}}U_pSV_p^{\mathrm{T}}V_pSU_p^{\mathrm{T}}U_pS^{-\frac{1}{2}} & 0 \\ 0 & 0 \end{bmatrix} \\
&= \begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix}
\end{aligned}$$

The same can be shown for $W_b^{-1}Q_pW_b^{-\mathrm{T}}$, $T_b^{-1}P_iT_b^{-\mathrm{T}}$ and $W_b^{-1}Q_iW_b^{-\mathrm{T}}$:

$$\begin{aligned}
T_b^{-1}P_pT_b^{-\mathrm{T}} &= \begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix} = W_b^{-1}Q_pW_b^{-\mathrm{T}}, \\
T_b^{-1}P_iT_b^{-\mathrm{T}} &= \begin{bmatrix} 0 & 0 \\ 0 & \Theta \end{bmatrix} = W_b^{-1}Q_iW_b^{-\mathrm{T}},
\end{aligned}$$

so the system equivalence transformation (W_b^{T}, T_b) gives the desired balanced realization. Summarizing this leads to the following simplified procedure to find a balanced realization of a descriptor system:

1. Compute the proper and improper Controllability and Observability Gramians P_p, P_i, Q_p and Q_i,

2. Find Cholesky factorizations such that

$$\begin{aligned}
P_p &= R_pR_p^{\mathrm{T}}, \quad & Q_p &= L_p^{\mathrm{T}}L_p, \\
P_i &= R_iR_i^{\mathrm{T}}, \quad & Q_i &= L_i^{\mathrm{T}}L_i,
\end{aligned}$$

   where $R_p^{\mathrm{T}}, L_p, R_i^{\mathrm{T}}, L_i \in \mathbb{R}^{n\times n}$ are upper triangular matrices,

3. Compute singular value decompositions of L_pER_p and L_iAR_i to get

$$L_pER_p = U_pSV_p^{\mathrm{T}}, \quad L_iAR_i = U_i\Theta V_i^{\mathrm{T}},$$

4. Let

$$\begin{aligned}
W_b &= \begin{bmatrix} L_p^{\mathrm{T}}U_pS^{-\frac{1}{2}} & L_i^{\mathrm{T}}U_i\Theta^{-\frac{1}{2}} \end{bmatrix}, \quad & W_b^{-1} &= \begin{bmatrix} ER_pV_pS^{-\frac{1}{2}} & AR_iV_i\Theta^{-\frac{1}{2}} \end{bmatrix}, \\
T_b &= \begin{bmatrix} R_pV_pS^{-\frac{1}{2}} & R_i^{\mathrm{T}}V_i\Theta^{-\frac{1}{2}} \end{bmatrix}, \quad & T_b^{-1} &= \begin{bmatrix} E^{\mathrm{T}}L_p^{\mathrm{T}}U_pS^{-\frac{1}{2}} & A^{\mathrm{T}}L_i^{\mathrm{T}}U_i\Theta^{-\frac{1}{2}} \end{bmatrix},
\end{aligned}$$

   then

$$\begin{aligned}
T_b^{-1}P_pT_b^{-\mathrm{T}} &= \begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix} = W_b^{-1}Q_pW_b^{-\mathrm{T}}, \\
T_b^{-1}P_iT_b^{-\mathrm{T}} &= \begin{bmatrix} 0 & 0 \\ 0 & \Theta \end{bmatrix} = W_b^{-1}Q_iW_b^{-\mathrm{T}}.
\end{aligned}$$

Hence, the system equivalence transformation (W_b^{T}, T_b) gives the balanced realization

$$(\hat{E}, \hat{A}, \hat{B}, \hat{C}, \hat{D}) = (W_b^{\mathrm{T}} E T_b, W_b^{\mathrm{T}} A T_b, W_b^{\mathrm{T}} B, C T_b, D)$$

with proper and improper Controllabilty and Observability Gramians

$$\begin{aligned} \hat{P}_p &= T_b^{-1} P_p T_b^{-\mathrm{T}}, \quad & \hat{Q}_p &= W_b^{-1} Q_p W_b^{-\mathrm{T}}, \\ \hat{P}_i &= T_b^{-1} P_i T_b^{-\mathrm{T}}, \quad & \hat{Q}_i &= W_b^{-1} Q_i W_b^{-\mathrm{T}}. \end{aligned}$$

4.3 Balanced truncation of descriptor systems

If the descriptor system (4.1) is not minimal, then it has states that are uncontrollable and/or unobservable. These states correspond to the zero proper and improper Hankel singular values and can be truncated without changing the input-output relation in the system. Note that the number of nonzero improper Hankel singular values of (4.1) is equal to $\operatorname{rank}(P_i A^{\mathrm{T}} Q_i A)$ which can in turn be estimated [15] as

$$\operatorname{rank}(P_i A^{\mathrm{T}} Q_i A) \leq \min(\nu m, \nu p, n_\infty).$$

This estimate shows that if the index ν of $\lambda E - A$ times the number of inputs or the number of outputs is much smaller than the dimension n_∞ of the deflating subspace of $\lambda E - A$ corresponding to the infinite eigenvalues, then the order of system (4.1) can be reduced significantly.

For the balanced system (4.16), P_p and $E^{\mathrm{T}} Q_p E$ are equal and, hence, they have the same invariant subspaces. In this case the truncation of the states related to the small proper Hankel singular values does not change system properties essentially. Unfortunately, this does not hold for the improper Hankel singular values. If we truncate the states that correspond to the small nonzero improper Hankel singular values, then the pencil of the reduced-order system may have finite eigenvalues in the closed right half-plane, see the following example from [23]:

Example 4.1. Consider the descriptor system

$$\begin{aligned} E\dot{x}(t) &= x(t) + Bu(t), \\ y(t) &= Cx(t), \end{aligned}$$

where

$$E = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} 0.1 \\ 0.2 \\ 1.8 \\ 2.5 \\ 3.0 \end{bmatrix}, \quad C^{\mathrm{T}} = \begin{bmatrix} 0.1 \\ 0.3 \\ 1.2 \\ 1.8 \\ 2.8 \end{bmatrix}.$$

By applying the balanced truncation technique [24] (note that this technique is different than our presented balanced truncation) to the discrete system

$$\begin{aligned} x(k+1) &= Nx(k) + Bu(k), \\ y(k) &= Cx(k), \end{aligned}$$

one can obtain the following third-order approximation [25]:

$$\begin{bmatrix} 0.5147 & 0.2445 & -0.0459 \\ -0.2445 & 0.2614 & 0.4158 \\ -0.0459 & -0.4158 & -0.5659 \end{bmatrix} \dot{x}_a(t) = x_a(t) + \begin{bmatrix} 3.9400 \\ 0.6512 \\ 0.1733 \end{bmatrix} u(t),$$
$$y(t) = \begin{bmatrix} 3.9400 & -0.6512 & 0.1733 \end{bmatrix} x_r(t).$$

We obtain that $\det(E_k) = -0.0309$, so we can conclude that the approximation has no improper part. Moreover, the three generalized eigenvalues are $2.759 + 1.667\text{i}$, $2.759 - 1.667\text{i}$ and -3.118, which shows that the approximated system is unstable. Hence, in this case the approximation is inaccurate.

Let (E, A, B, C, D) be a realization (not necessarily minimal) of the transfer function $G(s)$. Assume that the pencil $\lambda E - A$ is asymptotically stable. Consider the Cholesky factorizations (4.14). Let

$$\begin{aligned} L_p E R_p &= \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} S_1 & 0 \\ 0 & S_2 \end{bmatrix} \begin{bmatrix} V_1 & V_2 \end{bmatrix}^{\mathrm{T}}, \\ L_i A R_i &= U_3 \Theta_3 V_3^{\mathrm{T}} \end{aligned} \tag{4.20}$$

be singular value decompositions of $L_p E R_p$ and $L_i A R_i$, where $\begin{bmatrix} U_1 & U_2 \end{bmatrix}$, $\begin{bmatrix} V_1 & V_2 \end{bmatrix}$, U_3 and V_3 have orthonormal columns. Note that the matrices $L_p E R_p$ and $L_i A R_i$ are not necessarily of full rank, therefore we define $r_p = \operatorname{rank}(L_p E R_p) \leq n_f$ and $k_\infty = \operatorname{rank}(L_i A R_i) \leq n_\infty$. Then $S_1 = \operatorname{diag}(\varsigma_1, \varsigma_2, \ldots, \varsigma_{k_f})$ and $S_2 = \operatorname{diag}(\varsigma_{k_f+1}, \varsigma_{k_f+2}, \ldots, \varsigma_{r_p})$ and $\Theta_3 = \operatorname{diag}(\theta_1, \theta_2, \ldots, \theta_{k_\infty})$.

The new system balanced truncation is now given by the remaining $k = k_f + k_\infty$ states, which can be computed [15] as

$$(\tilde{E}, \tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}) = \left(W_k^{\mathrm{T}} E T_k, W_k^{\mathrm{T}} A T_k, W_k^{\mathrm{T}} B, C T_k, D\right), \tag{4.21}$$

where

$$\begin{aligned} W_k &:= \begin{bmatrix} L_p^{\mathrm{T}} U_1 S_1^{-\frac{1}{2}} & L_i^{\mathrm{T}} U_3 \Theta_3^{-\frac{1}{2}} \end{bmatrix} \in \mathbb{R}^{n \times k}, \\ T_k &:= \begin{bmatrix} R_p V_1 S_1^{-\frac{1}{2}} & R_i V_3 \Theta_3^{-\frac{1}{2}} \end{bmatrix} \in \mathbb{R}^{n \times k}. \end{aligned} \tag{4.22}$$

For model reduction of standard systems one can obtain an upper bound for the $\mathcal{L}_\infty$-norm of the error system, see Theorem 3.2. The same can be said regarding the error of model reduction of descriptor systems.

Theorem 4.2 ([25]). *Let $G(s)$ be a stable rational transfer function with proper Hankel singular values $\varsigma_1 > \varsigma_2 > \cdots > \varsigma_{k_f} > \varsigma_{k_f+1} > \cdots > \varsigma_{r_p}$ and improper Hankel singular values $\theta_1 > \theta_2 > \cdots > \theta_{k_\infty}$ and let $G_a^k(s)$ be obtained by truncating the balanced realization of $G(s)$ to the first k_f proper states, where $k_f = k - k_\infty$. Then*

$$||G(s) - G_a^k(s)||_{\mathcal{L}_\infty} \leq 2(\varsigma_{k_f+1} + \varsigma_{k_f+2} + \cdots + \varsigma_{r_p}).$$

4.3.1 Algorithm

As shown in the previous section, we have to compute the Cholesky factors of the proper and improper Controllability and Observability Gramians. For that we need system (4.1) to be in Weierstrass canonical form. Unfortunately, it is not very straightforward to find this canonical form. But the Cholesky factors can be computed using the *generalized Schur-Hammerling method* [18], [26].

Let the pencil $\lambda E - A$ be in generalized real Schur form (which can, for example, be done by the *QZ-algorithm* [27])

$$E = V \begin{bmatrix} E_f & E_u \\ 0 & E_\infty \end{bmatrix} U^{\mathrm{T}} \quad \text{and} \quad A = V \begin{bmatrix} A_f & A_u \\ 0 & A_\infty \end{bmatrix} U^{\mathrm{T}}, \tag{4.23}$$

where $U, V \in \mathbb{R}^{n \times n}$ are orthogonal, $E_f \in \mathbb{R}^{n_f \times n_f}$ is upper triangular nonsingular, $E_\infty \in \mathbb{R}^{n_\infty \times n_\infty}$ is upper triangular nilpotent, $A_f \in \mathbb{R}^{n_f \times n_f}$ is *upper quasi-triangular*[1] and $A_\infty \in \mathbb{R}^{n_\infty \times n_\infty}$ is upper triangular nonsingular and $E_u, A_u \in \mathbb{R}^{n_f \times n_\infty}$, and let the matrices

$$V^{\mathrm{T}} B = \begin{bmatrix} B_u \\ B_\infty \end{bmatrix} \quad \text{and} \quad CU = \begin{bmatrix} C_f & C_u \end{bmatrix}, \tag{4.24}$$

be partitioned conformally to E and A. Then one can show [18], [26] that the Cholesky factors of the Gramians of system (4.1) have the form

$$\begin{aligned} R_p &= U \begin{bmatrix} R_f \\ 0 \end{bmatrix}, & R_i &= U \begin{bmatrix} Y R_\infty \\ R_\infty \end{bmatrix}, \\ L_p &= \begin{bmatrix} L_f & -L_f Z \end{bmatrix} V^{\mathrm{T}}, & L_i &= \begin{bmatrix} 0 & L_\infty \end{bmatrix} V^{\mathrm{T}}, \end{aligned} \tag{4.25}$$

where $(Y, Z) \in \mathbb{R}^{n_f \times n_\infty} \times \mathbb{R}^{n_f \times n_\infty}$ is the solution of the generalized Sylvester equation

$$\begin{aligned} E_f Y - Z E_\infty &= -E_u, \\ A_f Y - Z A_\infty &= -A_u, \end{aligned} \tag{4.26}$$

the matrices $R_f^{\mathrm{T}}, L_f \in \mathbb{R}^{n_f \times n_f}$ are the upper triangular Cholesky factors of the solutions $X_{pc} = R_f R_f^{\mathrm{T}}$ and $X_{po} = L_f^{\mathrm{T}} L_f$ of the generalized continuous-time Lyapunov equations

$$E_f X_{pc} A_f^{\mathrm{T}} + A_f X_{pc} E_f^{\mathrm{T}} = -(B_u - Z B_\infty)(B_u - Z B_\infty)^{\mathrm{T}}, \tag{4.27}$$

$$E_f^{\mathrm{T}} X_{po} A_f + A_f^{\mathrm{T}} X_{po} E_f = -C_f^{\mathrm{T}} C_f, \tag{4.28}$$

while $R_\infty^{\mathrm{T}}, L_\infty \in \mathbb{R}^{n_\infty \times n_\infty}$ are the upper triangular Cholesky factors of the solutions $X_{ic} = R_\infty R_\infty^{\mathrm{T}}$ and $X_{io} = L_\infty^{\mathrm{T}} L_\infty$ of the generalized discrete-time Lyapunov equations

$$A_\infty X_{ic} A_\infty^{\mathrm{T}} - E_\infty X_{ic} E_\infty^{\mathrm{T}} = B_\infty B_\infty^{\mathrm{T}}, \tag{4.29}$$

$$A_\infty^{\mathrm{T}} X_{io} A_\infty - E_\infty^{\mathrm{T}} X_{io} E_\infty = (C_f Y + C_u)^{\mathrm{T}} (C_f Y + C_u). \tag{4.30}$$

Please note that solving the generalized Sylvester equation (4.26) is not in any way straightforward. To solve the generalized Sylvester equation, we use the *generalized Schur method* [28]. Check Appendix C for an overview of the generalized Schur method.

From (4.23) and (4.25) it follows that

$$\begin{aligned} L_p E R_p &= \begin{bmatrix} L_f & -L_f Z \end{bmatrix} V^{\mathrm{T}} V \begin{bmatrix} E_f & E_u \\ 0 & E_\infty \end{bmatrix} U^{\mathrm{T}} U \begin{bmatrix} R_f \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} L_f E_f & L_f E_u - L_f Z E_\infty \end{bmatrix} \begin{bmatrix} R_f \\ 0 \end{bmatrix} \\ &= L_f E_f R_f \end{aligned}$$

[1] Upper quasi-triangular means that the matrix has either 1×1 or 2×2 blocks on its diagonal, where the 2×2 blocks correspond to pairs of conjugate complex eigenvalues of the matrix pencil λE - A.

and in the same way, $L_i A R_i = L_\infty A_\infty R_\infty$. Since the proper and improper Hankel singular values of (4.1) are the singular values of $L_p E R_p$ and $L_i A R_i$, respectivily (Lemma 4.1), they can be computed from the singular value decompositions of the matrices $L_f E_f R_f$ and $L_\infty A_\infty R_\infty$.

Furthermore, it follows from (4.22) and (4.25) that the projection matrix W_k can be rewritten as

$$\begin{aligned} W_k &= \begin{bmatrix} L_p^{\mathrm{T}} U_1 S_1^{-\frac{1}{2}} & L_i^{\mathrm{T}} U_3 \Theta_3^{-\frac{1}{2}} \end{bmatrix} \\ &= \begin{bmatrix} L_p^{\mathrm{T}} & L_i^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} U_1 S_1^{-\frac{1}{2}} & 0 \\ 0 & U_3 \Theta_3^{-\frac{1}{2}} \end{bmatrix} \\ &= V \begin{bmatrix} L_f^{\mathrm{T}} & 0 \\ -Z^{\mathrm{T}} L_f^{\mathrm{T}} & L_\infty^T \end{bmatrix} \begin{bmatrix} U_1 S_1^{-\frac{1}{2}} & 0 \\ 0 & U_3 \Theta_3^{-\frac{1}{2}} \end{bmatrix} \\ &= V \begin{bmatrix} W_f & 0 \\ -Z^{\mathrm{T}} W_f & W_\infty \end{bmatrix}, \end{aligned}$$

with $W_f = L_f^{\mathrm{T}} U_1 S_1^{-\frac{1}{2}} \in \mathbb{R}^{n_f \times k_f}$ and $W_\infty = L_\infty^{\mathrm{T}} U_3 \Theta_3^{-\frac{1}{2}} \in \mathbb{R}^{n_\infty \times k_\infty}$. In the same way,

$$T_k = U \begin{bmatrix} T_f & Y T_\infty \\ 0 & T_\infty \end{bmatrix},$$

with $T_f = R_f V_1 S_1^{-\frac{1}{2}} \in \mathbb{R}^{n_f \times k_f}$ and $T_\infty = R_\infty V_3 \Theta_3^{-\frac{1}{2}} \in \mathbb{R}^{n_\infty \times k_\infty}$. After applying the given system equivalence transformation (W_k^{T}, T_k) we obtain the k'th-order realization (4.21) with matrices

$$\begin{aligned} \tilde{E} &= \begin{bmatrix} I_{k_f} & 0 \\ 0 & W_\infty^{\mathrm{T}} E_\infty T_\infty \end{bmatrix}, \quad \tilde{A} = \begin{bmatrix} W_f^{\mathrm{T}} A_f T_f & 0 \\ 0 & I_{k_\infty} \end{bmatrix} \\ \tilde{B} &= \begin{bmatrix} W_f^{\mathrm{T}} (B_u - Z B_\infty) \\ W_\infty^{\mathrm{T}} B_\infty \end{bmatrix}, \quad \tilde{C} = \begin{bmatrix} C_f T_f & (C_f Y + C_u) T_\infty \end{bmatrix}, \quad \tilde{D} = D. \end{aligned} \tag{4.31}$$

All computations stated above can be summarized by the *Generalized Square Root (GSR) method*:

Algorithm 4.1 ([15], Algorithm 3.1)**.** Generalized Square Root (GSR) method.

Imput: (E, A, B, C, D) such that $\lambda E - A$ is asymptotically stable.
Output: A k'th-order realization $(\tilde{E}, \tilde{A}, \tilde{B}, \tilde{C}, \tilde{D})$

1. Compute the generalized Schur form (4.23).
2. Compute the matrices (4.24).
3. Solve the generalized Sylvester equation (4.26).
4. Compute the Cholesky factors R_f and L_f of the solutions $X_{pc} = R_f R_f^{\mathrm{T}}$ and $X_{po} = L_f^{\mathrm{T}} L_f$ of (4.27) and (4.28), respectively.
5. Compute the Cholesky factors R_∞ and L_∞ of the solutions $X_{ic} = R_\infty R_\infty^{\mathrm{T}}$ and $X_{io} = L_\infty^{\mathrm{T}} L_\infty$ of (4.29) and (4.30), respectively.

6. Compute the "thin"[2] singular value decomposition

$$L_f E_f R_f = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} S_1 & 0 \\ 0 & S_2 \end{bmatrix} \begin{bmatrix} V_1 & V_2 \end{bmatrix}^{\mathrm{T}},$$

where the matrices $\begin{bmatrix} U_1 & U_2 \end{bmatrix}$ and $\begin{bmatrix} V_1 & V_2 \end{bmatrix}^{\mathrm{T}}$ have orthogonal columns, $S_1 = \mathrm{diag}(\varsigma_1, \varsigma_2, \ldots, \varsigma_{k_f})$ and $S_2 = \mathrm{diag}(\varsigma_{k_f+1}, \varsigma_{k_f+2}, \ldots, \varsigma_{r_f})$ with $r_f = \mathrm{rank}(L_f E_f R_f)$ and $\varsigma_1 \geq \varsigma_2 \geq \cdots \geq \varsigma_{k_f} > \varsigma_{k_f+1} \geq \varsigma_{k_f+2} \geq \cdots \geq \varsigma_{r_f}$.

7. Compute the "thin" singular value decomposition $L_\infty A_\infty R_\infty = U_3 \Theta_3 V_3^{\mathrm{T}}$, where U_3 and V_3 have orthonormal columns and $\Theta_3 = \mathrm{diag}(\theta_1, \theta_2, \ldots, \theta_{k_\infty})$ with $k_\infty = \mathrm{rank}(L_\infty A_\infty R_\infty)$.

8. Compute $W_f = L_f^{\mathrm{T}} U_1 S_1^{-\frac{1}{2}}, W_\infty = L_\infty^{\mathrm{T}} U_3 \Theta_3^{-\frac{1}{2}}, T_f = R_f V_1 S_1^{-\frac{1}{2}}$ and $T_\infty = R_\infty V_3 \Theta_3^{-\frac{1}{2}}$.

9. Compute the k'th-order system $(\tilde{E}, \tilde{A}, \tilde{B}, \tilde{C}, \tilde{D})$ as in (4.31).

[2]The SVD of matrix A might have $\sigma_i = 0$ for some i, the thin SVD of matrix A is given by the SVD of matrix A without those σ_i.

5 Implementation in MATLAB

In this chapter we will discuss how we implemented the algorithms in MATLAB, in order to test them and discuss the result. The result of our numerical example will be discussed in Chapter 6. We will show the implementations in MATLAB in the same order of the chapters of this report: we start with the fast projected gradient method (FGM) for finding the nearest positive-real system in Section 5.1, Section 5.2 briefly discusses model reduction for standard systems and we will conclude this chapter with the implementation of the Generalized Square Root (GSR) method in Section 5.3 for model reduction of descriptor systems.

Please not that not all MATLAB functions are given explicitly in order to save space. For the full content, please contact the author or his supervisor.

5.1 Finding the nearest positive-real system

The authors of [2] have made their code available online. All examples in that paper can be run directly from their code. Therefore, we have not adjusted the provided code in our implementation. We mainly use two functions: `stablePassiveFGM.m`, which is FGM of [2], and `getsystem.m`. The last function is used for obtaining the system matrices out of the MATLAB structure `sys` (the state-space structure).

The function `pos_real.m` returns the nearest positive-real system to a given system, following some possible preferences, given via the input variable `properties`. Our implementation starts with setting the options used in the FGM. The options maxiter, timemax, standard and initialization are straightforward, display is for displaying the evoluition of the objective function and the option weight is for setting the weights in the objective function. When the options are set, the function forms the system in MATLAB style and applies FGM to reach the nearest positive-real system. The variables `e` and `t` from the function `stablePassiveFGM` contain error and time information. In MATLAB, this looks as follows:

```
1   function [E_pr,A_pr,B_pr,C_pr,D_pr,e,t] = pos_real(E,A,B,C,D,properties)
2   %returns a positive-real system as close to the original system as possible,
3   %following given properties.
4
5   %% Getting the defined system properties
6   sort_system = properties{1,1};
7   sort_reduction = properties{1,2};
8
9   %% Options, they have to be changed inside this function
10  options.maxiter = Inf;
11  timemax = 5;
12  options.timemax = timemax;
13  options.display = 0;
14  weight = ones(5,1);
15  options.weight = weight;
16
17  %sort system
18  switch sort_system
19      case 'standard'
20         options.standard = 1; %Could be switched off if you want to allow
21                               %peturbations in E=I
22      case 'descriptor'
23          options.standard = 0;
24  end
```

```
25
26      %Initialization
27      options.init = 1; % standard initialization
28
29      %% Forming Matlab sys from matrices
30      sys.E = E_k;
31      ... %the others are the same
32
33      %% Running the algorithms
34      [PHform,e,t] = stablePassiveFGM(sys,options);
35      sysf_pr = getsystem(PHform);
36
37      %% Plotting the results
38
39      %% Obtaining the PR system
40      E_pr = sysf_pr.E;
41      ... %the others are the same
```

5.2 Model reduction of standard systems

Since model reduction of standard systems is commonly used, MATLAB has its own functions to apply model reduction to standard systems. We work with both descriptor and standard systems and therefore we use the variable `properties` to tell MATLAB if it has to deal with a standard or a descriptor system. We let MATLAB do the rest, which looks as follows:

```
1  function [E_k,A_k,B_k,C_k,D_k] = mod_red(E,A,B,C,D,k,properties)
2  %This function gives a k'th-order approximation of a given (descriptor) system. The
3  %user can choose which model reduction is used for standard systems, for descriptor
4  %systems the GSR-method is used.
5
6  %% Getting the defined system properties
7  sort_system = properties{1,1};
8  sort_reduction = properties{1,2};
9
10 %% Model reduction
11 switch sort_system
12     case 'standard'
13     E_k = eye(k); %for standard systems E = I;
14         switch sort_reduction
15             case 'truncation'
16                 [A_k,B_k,C_k,D_k] = truncation(A,B,C,D,k);
17             case 'residualization'
18                 [A_k,B_k,C_k,D_k] = residualization(A,B,C,D,k);
19         end
20     case 'descriptor'
21         [E_k,A_k,B_k,C_k,D_k] = gsr(E,A,B,C,D,k);
22 end
```

The functions `truncation.m` and `residualization.m` both use the MATLAB function `balred.m`. The only difference is that `truncation.m` executes balanced truncation (which is given as an option in `balredOptions`) and `residualization.m` executes balanced residualization (choose `Truncate` for truncation and `MatchDC` for residualization). They both are in the same form.

```
function [A_k,B_k,C_k,D_k] = truncation(A,B,C,D,k)
%This function gives the k'th-order approximation to a given standard system,
%using the balanced truncation technique.

%% Forming Matlab sys from matrices
sys.A = A;
sys.B = B;
sys.C = C;
sys.D = D;

%% Options for the balanced reduction method
opt = balredOptions('StateElimMethod','Truncate');

%% Using matlabs balanced truncation
ksys = balred(sys,k,opt);

%% Returning from Matlab sys to matrices
A_k = ksys.A;
B_k = ksys.B;
C_k = ksys.C;
D_k = ksys.D;
```

5.3 Model reduction of descriptor systems

The most challenging MATLAB function to make was the GSR method. It involves many matrix manipulations and it required to implement an advanced version of the Schur method to solve the generalized Sylvester equation. The result is the function `gsr.m`, which is stated below. The structure of this function follows the structure of the GSR method as can be found in Algorithm 4.1. We have omitted the matrix multiplications to improve readability.

```
function [E_k,A_k,B_k,C_k,D_k] = gsr(E,A,B,C,D,k)
%% Step 0: Use given properties and do initial settings
%Calculate amount of finite and infinite eigenvalues
n = size(E,1);
m = size(B,2);
p = size(C,1);
lambda = eig(A,E);
n_f = n;
for i = 1:n         %Number of finite eigenvalues (including multiplicity)
    if lambda(i) == inf
    n_f = n_f - 1;
    end
end

%% Step 1: Compute the generalized Schur form (4.24)
[A_s,E_s,V_s,U_s] = qz(A,E,'real');
%Makes the desired partition as well

%% Step 2: Compute the matrices (4.25)

%% Step 3: Solve the generalized Sylvester equation (4.27)
[Y,Z] = own_sylvester(E_f,E_inf,-E_u,A_f,A_inf,-A_u);

%% Step 4: Compute the Cholesky factors of the solutions of (4.28) and (4.29)
%solve (4.28)
B_1 = B_u - Z*B_inf;
Q_1 = B_1*B_1';
```

```
X_pc = lyap(A_f,Q_1,[],E_f);
%solve (4.29)
Q_2 = C_f'*C_f;
X_po = lyap(A_f',Q_2,[],E_f');
%compute the Cholesky factors R_f of X_pc and L_f of X_po
R_f = chol(X_pc,'lower');
L_f = chol(X_po,'upper');

%% Step 5: Compute the Cholesky factors of the solutions of (4.30) and (4.31)
%solve (4.30)
Q_3 = -B_inf*B_inf';
X_ic = dlyap(A_inf,Q_3,[],E_inf);
%solve (4.31)
C_1 = C_f*Y + C_u;
Q_4 = -C_1'*C_1;
X_io = dlyap(A_inf',Q_4,[],E_inf');
%compute the Cholesky factors R_inf of X_ic L_inf of X_io
R_inf = chol(X_ic,'lower');
L_inf = chol(X_io,'upper');

%% Step 6: Compute the thin SVD of L_f*E_f*R_f
Q_svd1 = L_f*E_f*R_f;
[U,S,V] = svd(Q_svd1,'econ');
%computing U_1, U_2, etc, will be done in the next step, when we know the
%desired sizes

%% Step 7: Compute the thin SVD of L_inf*A_inf*R_inf
Q_svd2 = L_inf*A_inf*R_inf;
[U_3,O_3,V_3] = svd(Q_svd2,'econ');
k_inf = size(O_3,1);
k_f = k - k_inf;
%compute U_1, U_2, S_1, S_2, V_1 and V_2:

%% Step 8: Compute W_f, W_inf, T_f and T_inf

%% Step 9: Compute the k'th-order system as in (4.32)
```

On line 22 we use the function `own_sylvester.m`. This function is as follows:

```
function [R,L] = own_sylvester(A,B,C,D,E,F)
%This function solves the generalized Sylvester equation
% AR - LB = C;
% DR - LE = F;
%for given matrices A,B,C,D,E and F and returns the solution (R,L).
%It makes use of the generelized Schur method.

%% Step 1: Transform (A,D) and (B,E) into the generalized Schur form
%Gives unitary matrices P,Q,U,V such that
%A_t = P*A*Q; D_t = P*D*Q;
%B_t = U*B*V; E_t = U*E*V;
[A_t,D_t,P,Q] = qz(A,D,'real');
[B_t,E_t,U,V] = qz(B,E,'real');
m = size(A_t,1);
n = size(B_t,1);

%% Step 2: Modify the right-hand sides (C,F)
C_t = P*C*V;
F_t = P*F*V;

%% Step 2.1: determine amount of blocks p and q
```

```
%returns amount of p blocks in A_t, q blocks in B_t, together with partiotioning
%blocksize_p and blocksize_q, which contain information about the size and
%whereabouts of the blokcs in A_t, B_t respectively.

%% Step 2.2: Make the partition

%% Step 3: solve the partitioned system for L_1 and R_1

%% Step 3.5: transform partition of solution back to matrix form

%% Step 4: transform solution back
L = P'*L_1*U;
R = Q*R_1*V';
```

Steps 2.1 and 2.2 follow the algorithm of Appendix C. The function makes use of the cell structure of MATLAB, where blocks $A_{ij}, B_{ij}, C_{ij}, D_{ij}, E_{ij}$ and F_{ij} are placed in the corresponding cells at position (i, j). The partitioning of these matrices (steps 2.1 and 2.2) into the cells has been omitted below. Step 3 then solves the blocks L_{ij} and R_{ij} by using the GS-algorithm, see Algorithm C.1. The solutions L_1 and R_1 can then be found in the calculated blocks of step 3, this is being done in step 3.5. Step 4 concludes the function by transforming L_1 and R_1 back to respectively L and R.

5.4 Finding the nearest positive-real system of reduced order

Combining the functions `pos_real.m` from Section 5.1, `mod_red.m` from Section 5.2 and `gsr.m` from Section 5.3 gives us the function `mod_red_pos_real.m`. This function first calculates the k'th-order approximation of a given system (the given system can be both a standard or a descriptor system) and then finds the nearest positive-real system to the k'th-order approximation.

```
function [E_pr,A_pr,B_pr,C_pr,D_pr] = mod_red_pos_real(E,A,B,C,D,k,properties)
%This function returns a positive-real, reduced system. The user can choose
%which model reduction is used for standard systems, for descriptor systems
%the GSR-method is used. This function first calculates the reduced-order
%system and then makes it positive-real.

%% Model reduction
[E_k,A_k,B_k,C_k,D_k] = mod_red(E,A,B,C,D,k);

%% Making system positive-real
%For making positive-real, we should have an m-input, m-output model
if size(B_k,2) == size(C_k,1)
    [E_pr,A_pr,B_pr,C_pr,D_pr,e,t] = pos_real(E,A,B,C,D,properties)
else
    disp('There should be m-input, m-output to achieve positive-realness')
end
```

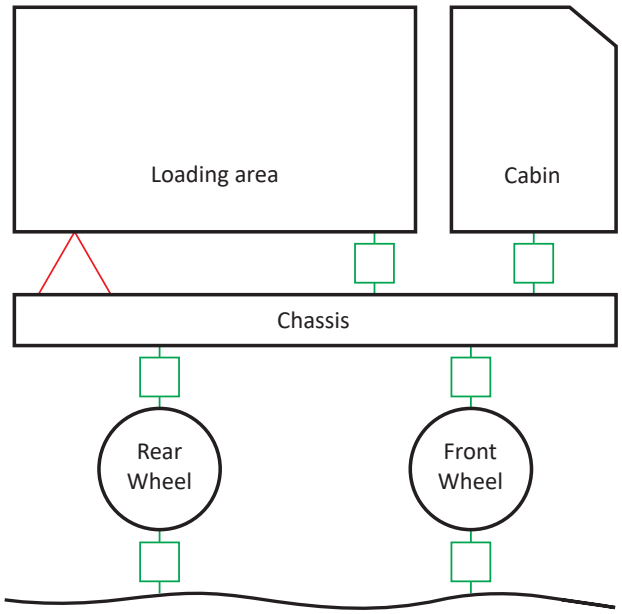

Figure 1: Sketch of the truck of Example 6.1.

6 Numerical experiment

In this chapter we discuss a numerical example to show the effects of the presented algorithms. We illustrate the effects by showing the frequency responses of the original model and its approximations, the norm of the error between the original model and its approximations. The error between the k'th-order approximation and the positive-real variant of the approximation is measured in terms of the Frobenius norm.

Example 6.1. This example is based on Example 4.1 from [15], which is a linearized model of a truck [29]. See Figure 1 for a sketch of this truck. Note that this sketch is a simplified version of the actual model. In fact, we also model the engine and the driver's seat. The red connection stands for a fixed rotation point between the loading area and the chassis of the truck, the green squares are spring-damper blocks. We model the translation of all seven parts together with four parts that can rotate as well, which leads to a state vector of size eleven. The truck is modeled by the following linearized equations:

$$\begin{aligned} \dot{p}(t) &= v(t), \\ M\dot{v}(t) &= Kp(t) - Dv(t) - G^{\mathrm{T}}\lambda(t) + B_2 u(t), \\ 0 &= Gp(t), \end{aligned}$$

where $p(t) \in \mathbb{R}^{11}$ is the position vector, $v(t) \in \mathbb{R}^{11}$ is the velocity vector, $\lambda(t) \in \mathbb{R}$ is the Lagrange multiplier, M is the positive definite mass matrix, K is the positive definite stiffness matrix, D is the positive semi-definite damping matrix, G contains the constraint information and B_2 is the input matrix. If we take $x(t) = \begin{bmatrix} p(t) & v(t) & \lambda(t) \end{bmatrix}^{\mathrm{T}}$ as state vector and output $y(t) = B_2^{\mathrm{T}} v(t)$ we obtain the descriptor system

$$\begin{aligned} \begin{bmatrix} I_{11} & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & 0 \end{bmatrix} \dot{x}(t) &= \begin{bmatrix} 0 & I_{11} & 0 \\ K & D & -G^{\mathrm{T}} \\ G & 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ B_2 \\ 0 \end{bmatrix} u(t), \\ y(t) &= \begin{bmatrix} 0 & B_2^{\mathrm{T}} & 0 \end{bmatrix} x(t), \end{aligned} \tag{6.1}$$

$\varsigma_1 = 1.209 \times 10^{-3}$	$\varsigma_6 = 6.629 \times 10^{-6}$	$\varsigma_{11} = 1.209 \times 10^{-6}$	$\varsigma_{16} = 1.469 \times 10^{-8}$
$\varsigma_2 = 1.207 \times 10^{-3}$	$\varsigma_7 = 5.125 \times 10^{-6}$	$\varsigma_{12} = 4.596 \times 10^{-7}$	$\varsigma_{17} = 1.158 \times 10^{-8}$
$\varsigma_3 = 1.561 \times 10^{-5}$	$\varsigma_8 = 3.758 \times 10^{-6}$	$\varsigma_{13} = 2.553 \times 10^{-7}$	$\varsigma_{18} = 3.769 \times 10^{-9}$
$\varsigma_4 = 1.414 \times 10^{-5}$	$\varsigma_9 = 2.365 \times 10^{-6}$	$\varsigma_{14} = 1.195 \times 10^{-7}$	$\varsigma_{19} = 9.073 \times 10^{-13}$
$\varsigma_5 = 6.850 \times 10^{-6}$	$\varsigma_{10} = 1.547 \times 10^{-6}$	$\varsigma_{15} = 3.455 \times 10^{-8}$	$\varsigma_{20} = 2.150 \times 10^{-13}$

Table 1: Proper Hankel singular values of descriptor system (6.1).

where $x(t) \in \mathbb{R}^{23\times 23}$, $u(t) \in \mathbb{R}$ and $y(t) \in \mathbb{R}$. We have chosen $y(t)$ in this way because we want the system to be positive-real. This system has 20 finite eigenvalues with negative real part ($n_f = 20$), and 3 infinite eigenvalues ($n_\infty = 3$). Following the original example, we approximate system (6.1) by a model of order $k_f + k_\infty = 15$, with $k_\infty = 3$. Since we have changed the model slightly compared to the original model, the proper Hankel singular values have changed as well. They are shown in Table 1.

Figure 2 shows the frequency responses of the original model and its 15'th-order approximation for frequencies in the rance $\omega \in [10^{-4}, 10^5]$ rad/sec. We see that the frequency respsonse of the approximated system only differs slighty at very low frequency. Therefore we can say that the approximation is accurate. Theorem 4.2 gives $||G(s) - G_k(s)||_{\mathcal{L}_\infty} \leq 8.989 \times 10^{-7}$. This error bound is shown in Figure 3, together with the absolute approximation error $|G(\mathrm{i}\omega) - G_k(\mathrm{i}\omega)|$.

After we have found the 15'th-order approximation of system (6.1), we apply FGM with standard initialization to find the nearest positive-real system to this approximated system. Figure 4 shows the frequency responses of the full-order system, its 15'th-order approximation (Σ_k) and the nearest positive-real system and Figure 5 compares the errors between the original system and both approximations. As shown in Figure 4, the frequency response of the nearest-positive real system to the lower order approximation differs from the original system and its approximation for frequencies lower than 10^{-1} rad/sec and higher than 10^{-2} rad/sec. Also, as can be expected, the error between the positive-real system and the original model is higher than the error bound.

Interpretation
In the original example of [15] the approximated system gives 12 finite eigenvalues and 3 eigenvalues of ∞. Due to the fact that we have changed the output of the system, the improper Hankel singular values in Θ_3 (4.20) are so small (the smallest in the order 10^{-36}), that Θ_3 is almost singular. Our approximated system gives 12 finite eigenvalues, 1 eigenvalue of ∞ and 2 eigenvalues of 'almost' ∞ (in the order of 10^{12}). Since they are positive, the system is not stable and therefore, by Definition 2.5, Σ_k is not positive-real, where you expect that a reduced-order approximation of a positive-real system is still positive-real. However, if you consider the two very large eigenvalues to be infinite, then the approximated system is positive-real.
FGM did not change the proper part of Σ_k, nor does it change B_k or C_k, and only changed E_k and A_k of the improper part as follows:

$$E_k = \begin{bmatrix} -0.02 & -0.02 & \epsilon \\ 0.02 & 0.02 & \epsilon \\ \epsilon & \epsilon & \epsilon \end{bmatrix}, \qquad A_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$E_{\mathrm{PR}} = \begin{bmatrix} -0,029 & -0,028 & \epsilon \\ 0,028 & 0,027 & \epsilon \\ \epsilon & \epsilon & \epsilon \end{bmatrix}, \quad A_{\mathrm{PR}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

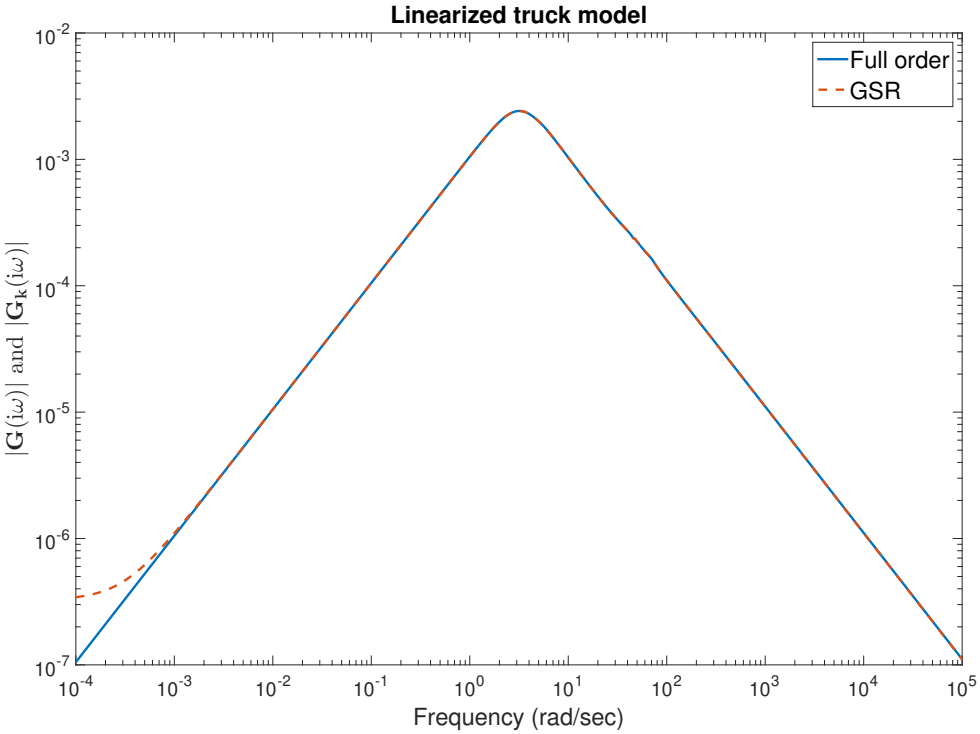

Figure 2: Frequency responses of the full-order system and the 15'th-order approximation computed by the GSR method.

where the ϵ denote almost zero (they are all different). The generalized eigenvalues of $(E_{\mathrm{PR}}, A_{\mathrm{PR}})$ are $\lambda_1 = \infty$, $\lambda_2 = -0.7$ and $\lambda_3 = -438.6$, so Σ_{PR} is asymptotically stable. This also explains the error between the original system and the positive-real approximation, and the different frequency response.

Unfortunately, we have to conclude that, for this example, the k'th-order positive-real approximation is less accurate than the regular k'th-order approximation, because the error of the PR approximation is higher and the frequency response differs more. This is mainly due to the fact that the k'th-order approximation is theoretically non-PR, but in practice it is PR. Therefore FGM changes the approximation to make it theoretically PR as well, which explains that Σ_{PR} less accurate than Σ_k.

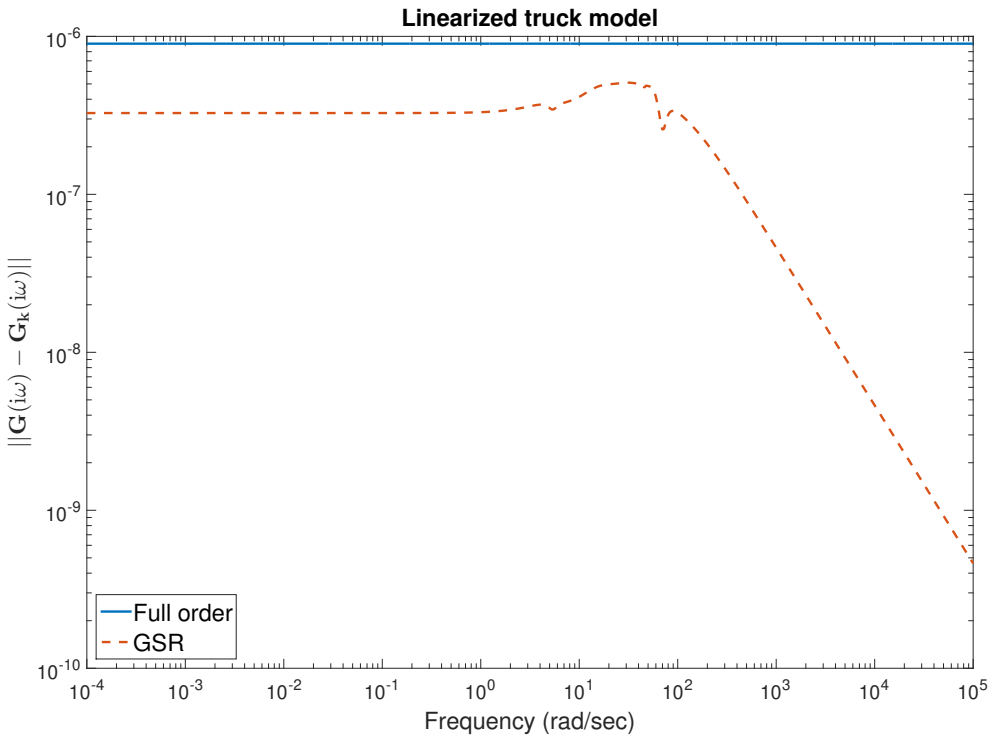

Figure 3: The error bound and the error of the approximated system.

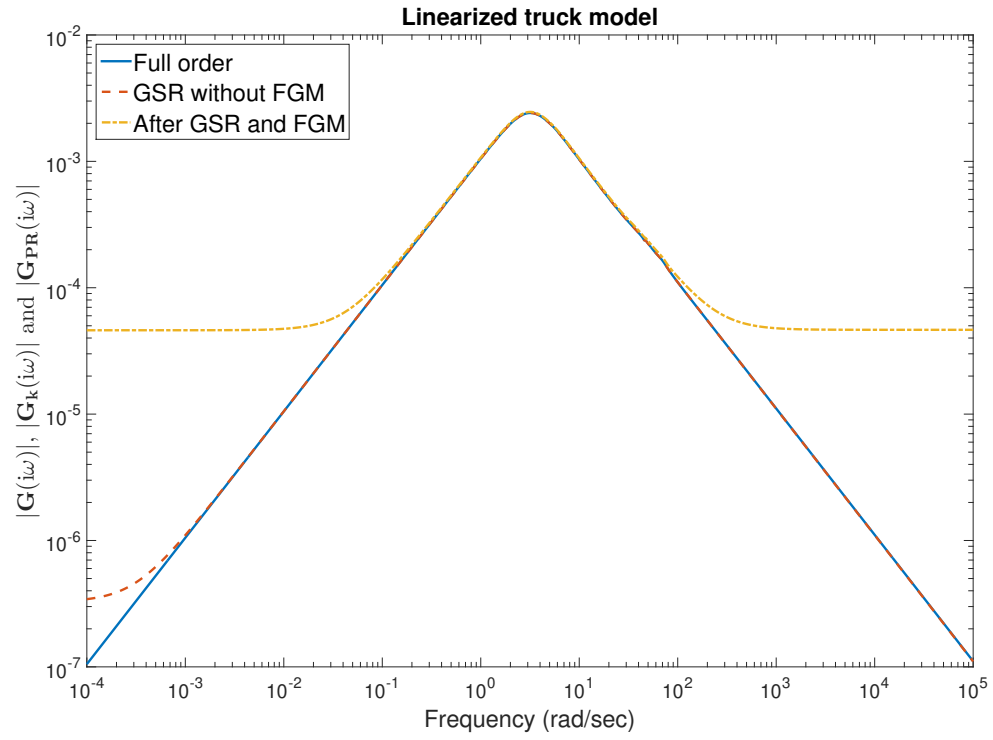

Figure 4: Frequency responses of the full-order system, the 15'th-order approximation computed by the GSR method and the nearest positive-real system computed by FGM.

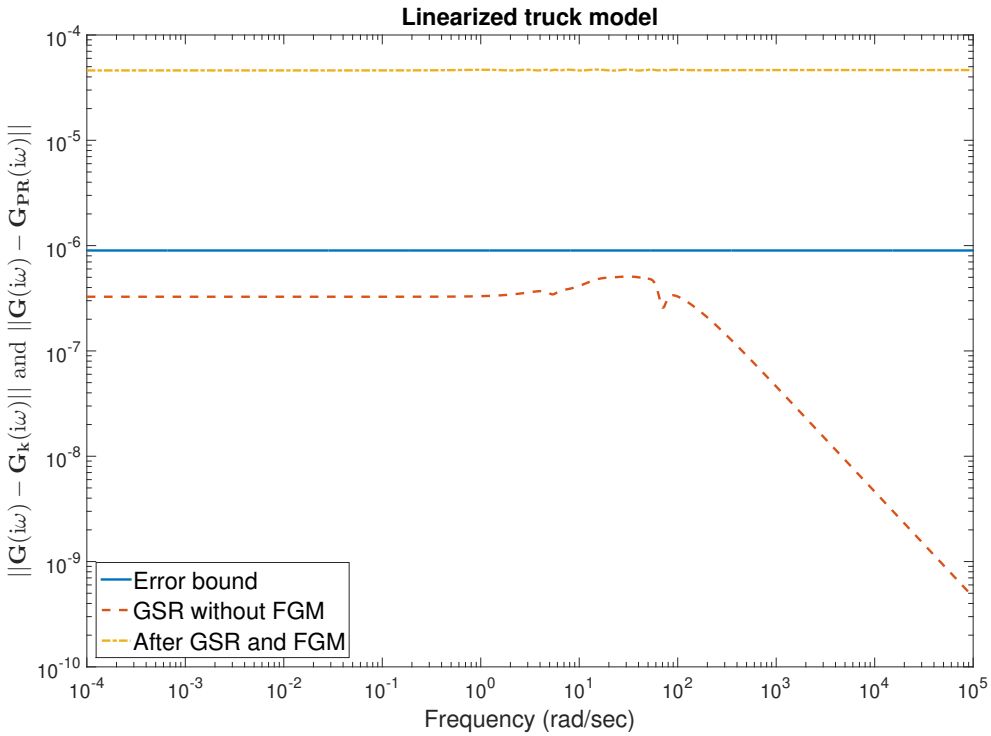

Figure 5: The error bound and the error of the approximated systems.

7 Conclusion

The goal of this report was, for a given LTI system Σ which is not necessarilty positive-real, to find the nearest positive-real system of lower order (Σ_a). In our journey to this joint problem, we have split the goal into two sub-goals: finding the nearest positive-real system and model reduction. This conclusion preserves that structure.

Nearest positive-real system
Using the results of [2], we have implemented FGM to find the nearest positive-real system to a given non-positive-real system, where the system can possibly be a descriptor system. Here we use the fact that every port-Hamiltonian system is positive-real (Theorem 2.2), since finding the nearest port-Hamiltonian system is easier than finding the nearest positive-real system.

Model reduction
Model reduction for standard systems is a well-studied subject, therefore we have only investigated the subject and use the available MATLAB functions for balanced truncation and balanced residualization.
However, model reduction for descriptor systems is much more complicated. We have studied the results of [15] and implemented the GSR method in MATLAB. This also involved implementing the Generalized Schur Method for solving generalized Sylvester equations.

Finding the nearest positive-real system of reduced order
Combining the above methods yields a method that solves our problem in a consecutive way. For a given (descriptor) system, we first use the appropriate model reduction method (balanced truncation or balanced residualization for standard systems and balanced truncation for

descriptor systems) and then we find the nearest positive-real system for the lower order approximation. Unfortunately, our numerical example did not show any improvement in the last step, mainly due to the fact that the original given system was already positive-real and because the improper Hankel singular values, used in the GSR, were to small.

8 Discussion and recommendations

Of course, as always in research projects, there are still some remarks that have to be made concerning this final project. This chapter contains the discussion and recommendations.

8.1 Discussion

Better methods
The main articles used for this project, [2] for finding the nearest positive-real system and [15] on model reduction for descriptor systems, both provide enhanced methods that have better perfomance in some cases. Due to the lack of time and/or technical knowledge, we have decided to focus ourselves on the easier understandable methods.

Where we have focussed on finding the nearest positive-real system, the authors of [2] have also considered strict positive-real systems (roughly means assymptotically stable) and extended positive-real systems (also preferable behaviour in infinite frequency). We have not studied the provided theory, but since the authors made their MATLAB functions publicly available, the FGM we have used can still find these systems. The same can be said about initializations: we have used the standard initialization in our numerical example, where the authors provide smart LMI-based initializations as well.

For model reduction of descriptor systems, we have used the GSR method, given by [15]. Also provided in the same paper is the Generalized Square Root Balancing Free (GSRBF) method. The authors show in their numerical examples that the GSRBF method gives systems with frequency response that coincide better with the original system and has a lower error throughout the frequency interval. The systems given by the GSRBF method are generally not balanced, where the GSR method gives balanced approximations.

Numerical examples
Unfortunately, to fully understand the technical aspects of this research topic, as well as the implementation in MATLAB, took more time than we expected when we started this project. Therefore, for mainly time-based reasons, we have chosen to show our results with only one numerical example. We are aware of the fact that this might not give a full image of the performance of our method.

Numerical boundaries of MATLAB
As already quickly said in Chapter 6 and Chapter 7 , Θ_3 (see (4.20)) in our numerical example consisted of three very small improper Hankel singular values (the smallest in the order 10^{-36}). Since we have to invert this Θ_3, the result has been influenced by the numerical boundaries of MATLAB. We believe that is the reason why the lower order approximation of a given positive-real system was not positive-real anymore. We think that the results can be improved by choosing a different application than MATLAB, or by manually forcing the improper part of the approximation to be positive-real.

Lack of technical knowledge
In order to have a really good feeling about the results and the used methods, you should have a decent understanding of descriptor systems. We can say that we started with little knowlegde about the behaviour of descriptor systems and gained knowledge during this project, but it is still hard to interpretate the things that happen with descriptor systems.

Literature
Our main library for the literature overview were citations in papers and Google Scholar. We realize that this not give a full overview of the available contents on this topic, we could have missed recent literature that does not have many citations yet.

8.2 Recommendations

Research would not be research if a project provided less questions than answers. Here we have given a summary of the open problems that came up during the period of this project.

Joint method versus consecutive method
Our method consists of two independent methods, applied consecutively. Since they are independent and consecutively, they could cancel each others effect or end up in local optima. It could be possible to obtain an improved method by developing a theory of finding the nearest positive-real system and reducing the order at the same time. The problem could be rewritten in a new setting, where only one different norm plays a role.

More numerical examples
As already mentioned in Section 8.1, we have only shown one numerical example and therefore our conclusion is based on only one example. The knowledge of our method could be improved a lot by applying it to more numerical examples, for example a non-positive-real system of high order, or systems of lower order.

Better implementation in MATLAB
We have not put our focus point on the implementation in MATLAB. We have used the basic computational tricks (such as computing a matrix inverse in a smart way), but strongly depend on the smartness of MATLAB. Our algorithm could be improved on this point. Moreover, in [15], T. Stykel uses the GUPTRI (generalized upper triangular) method to calculate the generalized Schur form (4.23), where we use the QZ algorithm of MATLAB. Next to this, T. Stykel calculates the Cholesky factors of the solutions of (4.27) – (4.30) without computing the solutions themself, but we just calculate first the solutions and then the Cholesky factors.

A whole different point of view
Many descriptor systems from real physical problems occur when one uses a Lagrange multiplier to model the system. In our case, (6.1), the Lagrange multiplier translates the physical aspects of the truck to the linearized model. The equation $0 = Gp(t)$ gives that some positions cannot move freely with respect to eachother, they are coupled (the Truck cannot be extended or pressed together). The singularity of our system, which is also called a differential-algebraic equation (DAE), is defined by how many times you should differentiate the DAE in order to obtain a ordinary differential equation (ODE). In our case, the index is 3 (we have 3 generalized eigenvalues of ∞).

What would we lose if we differentiate the equation $0 = Gp(t)$ such that we obtain an ODE?

This could be a whole different approach to modelling descriptor systems.

9 Notations

Symbol	Dimension	Definition
$\delta(t)$	$\mathbb{R}$	Dirac delta function
λ_i	$\mathbb{R}$	Eigenvalue such that $Ax = \lambda x$
$\Lambda(A)$	-	The set of all eigenvalues of the matrix A
$\rho(A)$	$\mathbb{R}$	Spectral radius of A: $\rho(A) := \max \lvert\lambda_i\rvert$
σ_j	$\mathbb{R}$	Hankel singular value: $\sigma_j := \sqrt{\lambda_j(PQ)}$
ς_j	$\mathbb{R}$	Proper Hankel singular value: $\varsigma_j := \lambda_j(P_p E^{\mathrm{T}} Q_p E)$
$\bar{\sigma}(A)$	$\mathbb{R}$	The maximum singular value of the matrix A
Σ	-[1]	The system: $\Sigma := (E, A, B, C, D)$
Σ_a	-[1]	The approximated system: $\Sigma_a := (E_a, A_a, B_a, C_a, D_a)$
θ_j	$\mathbb{R}$	Improper Hankel singular value: $\theta_j := \lambda_j(P_i A^{\mathrm{T}} Q_i A)$
Θ	$\mathbb{R}^{n-q\times n-q}$	Contains the improper Hankel singular values: $\Theta := \operatorname{diag}(\theta_j)$
∇_X	-	The gradient of a function with respect to the matrix X

Symbol	Dimension	Definition
A	$\mathbb{R}^{n\times n}$	State matrix
B	$\mathbb{R}^{n\times m}$	Input matrix
C	$\mathbb{R}^{m\times n}$	Observation matrix
D	$\mathbb{R}^{m\times m}$	Feedthrough matrix
E	$\mathbb{R}^{n\times n}$	System matrix
$E(t)$	$\mathbb{R}$	Energy of system at time t
F	$\mathbb{R}^{n\times m}$	$F \pm P$ are the port matrices
$\mathcal{F}_k$	$\mathbb{R}^{n\times n}$	$\mathcal{F}_k := T^{-1}\begin{bmatrix}0 & 0\\ 0 & -N^{-k-1}\end{bmatrix} W^{-1}, \quad k = -1, -2, \dots$
$\mathcal{F}(t)$	$\mathbb{R}^{n\times n}$	Fundamental solution matirx $\mathcal{F}(t) := T^{-1}\begin{bmatrix}\mathrm{e}^{tJ} & 0\\ 0 & 0\end{bmatrix} W^{-1}$
$\mathcal{F}(\dots)$	$\mathbb{R}$	Optimization goal in $\mathcal{P}$, function which gives distance from (E, A, B, C, D) to $(\tilde{E}, \tilde{A}, \tilde{B}, \tilde{C}, \tilde{D})$
G_i	$\mathbb{R}^{n-q\times n-q}$	Solution Lyapunov equation: $G_i - NG_iN^{\mathrm{T}} = B_iB_i^{\mathrm{T}}$
G_p	$\mathbb{R}^{q\times q}$	Solution Lyapunov equation: $JG_p + G_pJ^{\mathrm{T}} = -B_pB_p^{\mathrm{T}}$
$G(s)$	$\mathbb{C}^{m\times m}$	Transfer function of system Σ
$\mathcal{G}$	-	$\mathcal{G} := \{G \in \mathbb{R}^{n\times n} \mid G = J - R, J^{\mathrm{T}} = -J, R \succeq 0\}$
$h(t)$	$\mathbb{R}^{n\times n}$	Impulse response of system Σ
H_i	$\mathbb{R}^{n-q\times n-q}$	Solution Lyapunov equation: $H_i - N^{\mathrm{T}}H_iN = C_i^{\mathrm{T}}C_i$
H_p	$\mathbb{R}^{q\times q}$	Solution Lyapunov equation: $J^{\mathrm{T}}H_p + H_pJ = -C_p^{\mathrm{T}}C_p$
$\mathcal{H}(x)$	$\mathbb{R}$	$\mathcal{H}(x) = \frac{1}{2}x^{\mathrm{T}}Q^{\mathrm{T}}Ex$ is the Hamiltonian function
J	$\mathbb{R}^{n\times n}$	Structure matrix s.t. $J^{\mathrm{T}} = -J$
$\mathfrak{J}$	-	$\mathfrak{J} := \{J \in \mathbb{R}^{n\times n} \mid J^{\mathrm{T}} = -J\}$
K	$\mathbb{R}^{(n+m)\times(n+m)}$	$K = \begin{bmatrix}R & P\\ P^{\mathrm{T}} & S\end{bmatrix} \succeq 0$
L	$\mathbb{R}^{n\times n}$	$F \pm L$ are the port matrices
M	$\mathbb{R}^{n\times m}$	Projection of E on feasible set: $M := \tilde{E}$
n_f	$\mathbb{N}$	Number of finite eigenvalues of the pencil $\lambda E - A$
n_∞	$\mathbb{N}$	Number of infinite eigenvalues of the pencil $\lambda E - A$
N	$\mathbb{R}^{m\times m}$	Describes the direct feed-through from $u(t)$ to $y(t)$ s.t. $N^{\mathrm{T}} = N$
P	$\mathbb{R}^{n\times n}$	Controllability Gramian

[1] Note that the dimension of the matrix-quintuple (E, A, B, C, D) has been neglected.

Symbol	Dimension	Definition
P_i	$\mathbb{R}^{n\times n}$	Improper Controllability Gramian. $P_i = T^{-1}\begin{bmatrix}0 & 0\\ 0 & G_i\end{bmatrix}T^{-\mathrm{T}}$
P_p	$\mathbb{R}^{n\times n}$	Proper Controllability Gramian. $P_p = T^{-1}\begin{bmatrix}G_p & 0\\ 0 & 0\end{bmatrix}T^{-\mathrm{T}}$
$\mathcal{P}_l$	$\mathbb{R}^{n\times n}$	$\mathcal{P}_l := W\begin{bmatrix}I_q & 0\\ 0 & 0\end{bmatrix}W^{-1}$
$\mathcal{P}_r$	$\mathbb{R}^{n\times n}$	$\mathcal{P}_r := T^{-1}\begin{bmatrix}I_q & 0\\ 0 & 0\end{bmatrix}T$
$\mathcal{P}_{\mathfrak{S}}(X)$	$\mathbb{R}^{n\times n}$	The projection of a matrix X on the linear subspace of skew-symmetric matrices
$\mathcal{P}_{\succeq}(X)$	$\mathbb{R}^{n\times n}$	The projection of a matrix X on the cone of p.s.d. matrices
$\mathcal{P}_{\mathrm{PR}}$	-	Nearest PR-system problem
$\mathcal{P}_{\mathrm{PH}}$	-	Nearest PH-system problem
Q	$\mathbb{R}^{n\times n}$	Invertible matrix which describes energy s.t. $Q^{\mathrm{T}}E = E^{\mathrm{T}}Q \succeq 0$
Q	$\mathbb{R}^{n\times n}$	Observability Gramian
Q_i	$\mathbb{R}^{n\times n}$	Improper Observability Gramian. $Q_i = W^{-\mathrm{T}}\begin{bmatrix}0 & 0\\ 0 & H_i\end{bmatrix}W^{-1}$
Q_p	$\mathbb{R}^{n\times n}$	Proper Observability Gramian. $Q_p = W^{-\mathrm{T}}\begin{bmatrix}H_p & 0\\ 0 & 0\end{bmatrix}W^{-1}$
R	$\mathbb{R}^{n\times n}$	Dissipation matrix s.t. $R \succeq 0$
$\mathfrak{R}$	-	$\mathfrak{R} := \{R \in \mathbb{R}^{n\times n} \mid R^{\mathrm{T}} = R\}$
s	$\mathbb{C}$	Derivative function
S	$\mathbb{R}^{m\times m}$	Describes the direct feed-through from input to output s.t. $S \succeq 0$
S	$\mathbb{R}^{n\times n}$	Contains the (proper) Hankel singular values: $\Sigma := \mathrm{diag}(\sigma_j)$
$\mathbb{S}_{\mathrm{PR}}$	- [1]	The set of all PR-systems
$\mathbb{S}_{\mathrm{PH}}$	- [1]	The set of all PH-systems
T	$\mathbb{R}^{n\times n}$	State transform. $\hat{x}(t) = Tx(t)$ or part of system transform. (W, T)
$u(t)$	$\mathbb{R}^m$	Input at time t
$U(s)$	$\mathbb{C}^m$	Laplace transform of $u(t)$
$V(x(t))$	$\mathbb{R}$	Storagefunction for state $x(t)$
W	$\mathbb{R}^{n\times n}$	Part of system equivalence transformation (W, T)
x_0	$\mathbb{R}^n$	Initial state $x_0 := x(0)$
$x(t)$	$\mathbb{R}^n$	State at time t
$X(s)$	$\mathbb{C}^n$	Laplace transform of $x(t)$
$y(t)$	$\mathbb{R}^m$	Output at time t
$Y(s)$	$\mathbb{C}^m$	Laplace transform of $y(t)$
Z	$\mathbb{R}^{m\times n}$	$M := M^{\mathrm{T}}Q$
$\dot{x}(t)$	$\mathbb{R}^n$	$\frac{\mathrm{d}}{\mathrm{d}t}x(t)$

Symbol	Dimension	Definition
M_{11}	$\mathbb{R}^{k\times k}$	Truncated realization (partitioned) of matrix M
M_r	$\mathbb{R}^{n\times n}$	Residualized realization of matrix M
$\hat{M}$	$\mathbb{R}^{n\times n}$	Balanced realization of matrix M
$\tilde{M}$	$\mathbb{R}^{k\times k}$	k'th-order realization of matrix M
$\|\|A\|\|_{\mathrm{F}}$	$\mathbb{R}$	$\|\|A\|\|_{\mathrm{F}} := \mathrm{tr}(A^{\mathrm{T}}A)$
$\|\|\Sigma\|\|_{\mathrm{H}}$	$\mathbb{R}$	$\|\|\Sigma\|\|_{\mathrm{H}} = \sqrt{\rho(PQ)}$
$\|\|\Sigma\|\|_{\mathcal{H}_2}$	$\mathbb{R}$	$\|\|\Sigma\|\|_{\mathcal{H}_2} = \sqrt{\mathrm{tr}(B^{\mathrm{T}}PB)}$
$\|\|\Sigma\|\|_{\mathcal{H}_\infty}$	$\mathbb{R}$	$\|\|\Sigma\|\|_{\mathcal{H}_\infty} := \sup_{\mathrm{Re}(s)>0} \bar{\sigma}(G(s))$
$\langle A, B\rangle_{\mathrm{F}}$	$\mathbb{R}$	$\langle A, B\rangle_{\mathrm{F}} := \mathrm{tr}(A^{\mathrm{T}}B)$
$W^{\perp}$	-	$W^{\perp} := \{A \in \mathbb{R}^{n\times n} \mid \langle A, B\rangle_{\mathrm{F}} = 0 \text{ for all } B \in W\}$
$\mathrm{Im}(\omega)$	$\mathbb{R}$	Imaginary part of complex ω: $\mathrm{Im}(a+b\mathrm{i}) = b$
$\mathrm{Re}(\omega)$	$\mathbb{R}$	Real part of complex ω: $\mathrm{Re}(a+b\mathrm{i}) = a$

References

[1] R. W. Freund and F. Jarre, "An extension of the positive real lemma to descriptor systems," *Optimization methods and software*, vol. 19, no. 1, pp. 69–87, 2004.

[2] N. Gillis and P. Sharma, "Finding the nearest positive-real system," *arXiv preprint arXiv:1707.00530*, 2017.

[3] B. De Schutter, "Minimal state-space realization in linear system theory: an overview," *Journal of computational and applied mathematics*, vol. 121, no. 1-2, pp. 331–354, 2000.

[4] L. Dai, "Singular control systems," *Lecture notes in control and information science*, 1989.

[5] R. Lozano, B. Brogliato, O. Egeland, and B. Maschke, *Dissipative systems analysis and control: theory and applications*. Springer Science & Business Media, 2013.

[6] F.-X. Orbandexivry, Y. Nesterov, and P. Van Dooren, "Nearest stable system using successive convex approximations," *Automatica*, vol. 49, no. 5, pp. 1195–1203, 2013.

[7] N. Gillis, V. Mehrmann, and P. Sharma, "Computing nearest stable matrix pairs," *arXiv preprint arXiv:1704.03184*, 2017.

[8] B. D. Anderson and S. Vongpanitlerd, *Network analysis and synthesis: a modern systems theory approach*. Courier Corporation, 2013.

[9] N. Gillis and P. Sharma, "On computing the distance to stability for matrices using linear dissipative hamiltonian systems," *Automatica*, vol. 85, pp. 113–121, 2017.

[10] S. Skogestad and I. Postlethwaite, *Multivariable feedback control: analysis and design*, vol. 2. Wiley New York, 2007.

[11] G. Meinsma, *Nine Lessons on Control Systems*. Lecture Notes Robust Control, University of Twente, 2017.

[12] Y. Liu and B. D. Anderson, "Singular perturbation approximation of balanced systems," *International Journal of Control*, vol. 50, no. 4, pp. 1379–1405, 1989.

[13] K. Zhou, J. C. Doyle, K. Glover, *et al.*, *Robust and optimal control*, vol. 40. Prentice hall New Jersey, 1996.

[14] K. Fernando and H. Nicholson, "Singular perturbational model reduction of balanced systems," *IEEE Transactions on Automatic Control*, vol. 27, no. 2, pp. 466–468, 1982.

[15] T. Stykel, "Gramian-based model reduction for descriptor systems," *Mathematics of Control, Signals and Systems*, vol. 16, no. 4, pp. 297–319, 2004.

[16] F. Gantmacher, *The Theory of Matrices I.* Chelsea Publishing Company, New York, NY, 1959.

[17] G. W. Stewart, "Error and perturbation bounds for subspaces associated with certain eigenvalue problems," *SIAM review*, vol. 15, no. 4, pp. 727–764, 1973.

[18] T. Stykel, "Analysis and numerical solution of generalized lyapunov equations," *Institut für Mathematik, Technische Universität, Berlin*, 2002.

[19] D. Bender, "Lyapunov-like equations and reachability/observabiliy gramians for descriptor systems," *IEEE Transactions on Automatic Control*, vol. 32, no. 4, pp. 343–348, 1987.

[20] A. W. Bojanczyk, L. M. Ewerbring, F. T. Luk, and P. Van Dooren, "An accurate product svd algorithm," *Signal Processing*, vol. 25, no. 2, pp. 189–201, 1991.

[21] K. Glover, "All optimal hankel-norm approximations of linear multivariable systems and their l^∞-error bounds," *International journal of control*, vol. 39, no. 6, pp. 1115–1193, 1984.

[22] B. Moore, "Principal component analysis in linear systems: Controllability, observability, and model reduction," *IEEE transactions on automatic control*, vol. 26, no. 1, pp. 17–32, 1981.

[23] W. Q. Liu and V. Sreeram, "Model reduction of singular systems," *International Journal of Systems Science*, vol. 32, no. 10, pp. 1205–1215, 2001.

[24] L. Pernebo and L. Silverman, "Model reduction via balanced state space representations," *IEEE Transactions on Automatic Control*, vol. 27, no. 2, pp. 382–387, 1982.

[25] K. Perev and B. Shafai, "Balanced realization and model reduction of singular systems," *International Journal of Systems Science*, vol. 25, no. 6, pp. 1039–1052, 1994.

[26] T. Stykel, "Numerical solution and perturbation theory for generalized lyapunov equations," *Linear Algebra and its Applications*, vol. 349, no. 1-3, pp. 155–185, 2002.

[27] C. B. Moler and G. W. Stewart, "An algorithm for generalized matrix eigenvalue problems," *SIAM Journal on Numerical Analysis*, vol. 10, no. 2, pp. 241–256, 1973.

[28] B. Kagstrom and L. Westin, "Generalized schur methods with condition estimators for solving the generalized sylvester equation," *IEEE Transactions on Automatic Control*, vol. 34, no. 7, pp. 745–751, 1989.

[29] B. Simeon, F. Grupp, C. Führer, and P. Rentrop, "A nonlinear truck model and its treatment as a multibody system," *Journal of Computational and Applied Mathematics*, vol. 50, no. 1-3, pp. 523–532, 1994.

[30] S. H. Friedberg, A. J. Insel, and L. E. Spence, *Linear Algebra.* Pearson Education, Inc., 2003.

[31] N. J. Higham, "Computing a nearest symmetric positive semidefinite matrix," *Linear algebra and its applications*, vol. 103, pp. 103–118, 1988.

[32] Y. Feng and M. Yagoubi, *Robust Control of Linear Descriptor Systems*, vol. 102. Springer, 2017.

[33] R. H. Bartels and G. W. Stewart, "Solution of the matrix equation ax+ xb= c [f4]," *Communications of the ACM*, vol. 15, no. 9, pp. 820–826, 1972.

[34] K.-w. E. Chu, "The solution of the matrix equations axb- cxd= e and (ya- dz, yc- bz)=(e, f)," *Linear Algebra and its Applications*, vol. 93, pp. 93–105, 1987.

A Used projections in the Fast projected Gradient Method

In this appendix, both the projection on the linear subspace of skew-symmetric matrices and the projection on the cone of positive semi-defnite matrices will be derived.

A.1 Projection on the linear subspace of skew-symmetric matrices

Definition A.1. Let W be a nonempty subset of an inner product space V. We define $W^\perp$ to be the set of all vectors in V that are orthogonal to every vector in W, that is, $W^\perp := \{x \in V : \langle x, y\rangle = 0 \text{ for all } y \in W\}$. The set $W^\perp$ is called the *orthogonal complement* of W.

Now, define $V := \mathbb{R}^{n\times n}$ as inner product space of all square matrices with inner product $\langle A, B\rangle_{\mathrm{F}} := \mathrm{tr}(A^{\mathrm{T}}B)$. Let $\mathfrak{J}$ be the linear subspace containing all skew-symmetric matrices. We first show that $\mathfrak{J}^\perp$ is given by the linear subspace containing all symmetric matrices:

Denote the linear subspace containing all symmetric matrices by $\mathfrak{R}$. Taking $J \in \mathfrak{J}$ and $R \in \mathfrak{R}$, we have $R^{\mathrm{T}} = R$ and $J^{\mathrm{T}} = -J$. Now,

$$\begin{aligned} \langle R, J\rangle_{\mathrm{F}} = \mathrm{tr}(R^{\mathrm{T}}J) = \mathrm{tr}(RJ) = \mathrm{tr}(J^{\mathrm{T}}R) = -\mathrm{tr}(JR) = -\mathrm{tr}(RJ) \\ \implies \mathrm{tr}(RJ) = -\mathrm{tr}(RJ) \implies \langle R, J\rangle_{\mathrm{F}} = \mathrm{tr}(RJ) = 0. \end{aligned} \tag{A.1}$$

From this we can conclude that $\mathfrak{J}^\perp \supseteq \mathfrak{R}$. Take $Z \in V$. It can be checked easily that the symmetric part of Z is given by $\frac{Z+Z^{\mathrm{T}}}{2}$ and the skew-symmetric part by $\frac{Z-Z^{\mathrm{T}}}{2}$. Since for any square matrix $Z \in V$ you can write $Z = \frac{Z+Z^{\mathrm{T}}}{2} + \frac{Z-Z^{\mathrm{T}}}{2}$,

$$V = \mathfrak{J} + \mathfrak{R}. \tag{A.2}$$

Moreover, from [30, Theorem 6.6] we have that

$$V = \mathfrak{J} + \mathfrak{J}^\perp. \tag{A.3}$$

Since the only matrix that is both skew-symmetric and symmetric is the zero-matrix ($\mathfrak{J} \cap \mathfrak{R} = \{0\}$), and since the only element in both $\mathfrak{J}$ and $\mathfrak{J}^\perp$ is the zero-matrix ($\mathfrak{J} \cap \mathfrak{J}^\perp = \{0\}$), we have, together with Equations (A.2) and (A.3):

$$\mathfrak{J}^\perp = \mathfrak{R}.$$

Hence $\mathfrak{J}^\perp$ is given by the linear subspace containing all symmetric matrices. By [30, Theorem 6.6] there exist unique $J \in \mathfrak{J}$ and $R \in \mathfrak{R}$ such that $Z = J + R$. In specific: $J = \frac{Z-Z^{\mathrm{T}}}{2}$ and $R = \frac{Z+Z^{\mathrm{T}}}{2}$. Moreover, by the Corollary of [30, Theorem 6.6], $\frac{Z-Z^{\mathrm{T}}}{2}$ is the unique "closest" solution to (2.9). This gives

$$\mathcal{P}_{\mathfrak{J}}(Z) = \frac{Z - Z^{\mathrm{T}}}{2}. \tag{A.4}$$

A.2 Projection on the cone of positive semidefinite matrices

The following theory is from [31].

Theorem A.1 ([31], Theorem 2.1)**.** *Let $Z \in \mathbb{R}^{n\times n}$, and let $A = \frac{Z+Z^{\mathrm{T}}}{2}$ and $B = \frac{Z-Z^{\mathrm{T}}}{2}$ be the symmetric and skew-symmetric parts of Z respectively. Let $A = U\Gamma U^{\mathrm{T}}$ be a spectral decomposition of A $[U^{\mathrm{T}}U = I, \Gamma = \mathrm{diag}(\lambda_i)]$. Then the projection on the cone of positive semidefinite matrices is given by*

$$\mathcal{P}_{\succeq}(Z) = U\left(\max(\Gamma, 0)\right)U^{\mathrm{T}},$$

and

$$\min_{R\succeq 0}||Z-R||_{\mathrm{F}}^2 = ||\mathcal{P}_{\mathfrak{J}}(Z)||_{\mathrm{F}}^2 + \sum_{\lambda_i\in\Lambda(\Gamma),\lambda_i<0}\lambda_i^2,$$

where $\mathcal{P}_{\mathfrak{J}}(Z)$ *is given in* (A.4).

Proof. Let $R\in\mathbb{R}^{n\times n}$ be positive semidefinite. From (A.1) follows that $||R+J||_{\mathrm{F}}^2 = ||R||_{\mathrm{F}}^2+||J||_{\mathrm{F}}^2$ if $R^{\mathrm{T}}=R$ and $J^{\mathrm{T}}=-J$, so we have

$$||Z-R||_{\mathrm{F}}^2 = ||A+B-R||_{\mathrm{F}}^2 = ||A-R||_{\mathrm{F}}^2 + ||B||_{\mathrm{F}}^2 \tag{A.5}$$

and so the problem reduces to that of approximating A. Let $A=U\Gamma U^{\mathrm{T}}$ be a spectral decomposition of A, and let $X=U^{\mathrm{T}}RU$ (note that since R is positive semidefinite, X is positive semidefinite as well). Then

$$\begin{aligned}
||A-R||_{\mathrm{F}}^2 &= ||U\Gamma U^{\mathrm{T}} - UXU^{\mathrm{T}}||_{\mathrm{F}}^2 = ||U(\Gamma-X)U^{\mathrm{T}}||_{\mathrm{F}}^2 \\
&= \mathrm{tr}\left[\left(U(\Gamma-X)U^{\mathrm{T}}\right)^{\mathrm{T}}\left(U(\Gamma-X)U^{\mathrm{T}}\right)\right] \\
&= \mathrm{tr}\left[U(\Gamma-X)^{\mathrm{T}}(\Gamma-X)U^{\mathrm{T}}\right] \\
&= \mathrm{tr}\left[(\Gamma-X)U^{\mathrm{T}}U(\Gamma-X)^{\mathrm{T}}\right] \\
&= \mathrm{tr}\left[(\Gamma-X)(\Gamma-X)^{\mathrm{T}}\right] = ||\Gamma-X||_{\mathrm{F}}^2.
\end{aligned} \tag{A.6}$$

Following the definition of the Frobenius-norm (2.1) and the fact that Γ is a diagonal matrix, we get

$$\begin{aligned}
||\Gamma-X||_{\mathrm{F}}^2 &= \sum_{i\neq j}x_{ij}^2 + \sum_i(\lambda_i-x_{ii})^2 \\
&\geq \sum_i(\lambda_i-x_{ii})^2 \\
&\geq \sum_{\lambda_i<0}(\lambda_i-x_{ii})^2 \\
&\geq \sum_{\lambda_i<0}\lambda_i^2,
\end{aligned} \tag{A.7}$$

since $x_{ii}\geq 0$ because X is positive semidefinite. This lower bound is attained, uniquely, for the matrix $X=\mathrm{diag}(d_i)$, where

$$d_i = \begin{cases}\lambda_i, & \lambda_i\geq 0\\ 0 & \lambda_i<0\end{cases}. \tag{A.8}$$

Chosing X like this leads to the first inequality becoming an equality, since $x_{ij}=0$ for $i\neq j$. The second inequality becomes an equality because for all i such that $\lambda_i\geq 0$:

$$(\lambda_i-x_{ii}) = (\lambda_i-d_i) = (\lambda_i-\lambda_i) = 0$$

and the third since $x_{ii}=0$ for $\lambda_i<0$. To create $X=\mathrm{diag}(d_i)$, with d_i as in (A.8), choose

$$R = U\mathrm{diag}(d_i)U^{\mathrm{T}}.$$

Note that $\mathrm{diag}(d_i)=\max(\Gamma,0)$, which leads to

$$\mathcal{P}_{\succeq}(Z) = U(\max(\Gamma,0))U^{\mathrm{T}}. \tag{A.9}$$

Moreover, substituting (A.4) in (A.5) gives

$$\min_{R\succeq 0}||Z-R||_{\mathrm{F}}^2 = ||Z-\mathcal{P}_{\succeq}(Z)||_{\mathrm{F}}^2 = ||A-\mathcal{P}_{\succeq}(Z)||_{\mathrm{F}}^2 + ||\mathcal{P}_{\mathfrak{J}}(Z)||_{\mathrm{F}}^2$$

and equations (A.6) and (A.7), together with $X = U^{\mathrm{T}}(\mathcal{P}_{\succeq}(Z))U$, where $\mathcal{P}_{\succeq}(Z)$ is given in (A.9), give

$$\begin{aligned}\min_{R\succeq 0}||Z-R||_{\mathrm{F}}^2 &= ||\mathcal{P}_{\mathfrak{J}}(Z)||_{\mathrm{F}}^2 + ||A-\mathcal{P}_{\succeq}(Z)||_{\mathrm{F}}^2 \\ &= ||\mathcal{P}_{\mathfrak{J}}(Z)||_{\mathrm{F}}^2 + \sum_{\lambda_i\in\Lambda(\Gamma),\lambda_i<0}\lambda_i^2.\end{aligned}$$

□

B Gradient of f with respect to X

In this section, we will show how the gradient of $f(X) = ||AX-B||_{\mathrm{F}}^2$ with respect to X can be found. First, start with the definition of the gradient with respect to a matrix.

Definition B.1. The *gradient with respect to a matrix* of a function $f(X) : \mathbb{R}^{n\times m} \to \mathbb{R}$ is a function $\nabla_X f(X)$ such that

$$\lim_{||\mathrm{d}X||_{\mathrm{F}}\to 0} \frac{|f(X+\mathrm{d}X) - f(X) - \langle \nabla_X f(X), \mathrm{d}X\rangle_{\mathrm{F}}|}{||\mathrm{d}X||_{\mathrm{F}}} = 0. \tag{B.1}$$

Equation (B.1) gives

$$f(X+\mathrm{d}X) - f(X) - \langle \nabla_X f(X), \mathrm{d}X\rangle_{\mathrm{F}} \to 0 \quad \text{as} \quad \mathrm{d}X \to 0.$$

Substituting $f(X) = ||AX-B||_{\mathrm{F}}^2$ into this formula gives an expression for the gradient of $f(X)$ (note that the three different trace functions were taken into one in the first line):

$$\begin{aligned}&\mathrm{tr}\left[\big(A(X+\mathrm{d}X)-B\big)^{\mathrm{T}}\big(A(X+\mathrm{d}X)-B\big) - \big(AX-B\big)^{\mathrm{T}}\big(AX-B\big) - (\mathrm{d}X)^{\mathrm{T}}\nabla_X f(X)\right] \\ =&\mathrm{tr}\left[(A\mathrm{d}X)^{\mathrm{T}}(AX-B) + (AX-B)^{\mathrm{T}}(A\mathrm{d}X) + (A\mathrm{d}X)^{\mathrm{T}}(A\mathrm{d}X) - (\mathrm{d}X)^{\mathrm{T}}\nabla_X f(X)\right] \\ =&\mathrm{tr}\left[2(A\mathrm{d}X)^{\mathrm{T}}(AX-B)\right] + \mathrm{tr}\left[(A\mathrm{d}X)^{\mathrm{T}}(A\mathrm{d}X)\right] - \mathrm{tr}\left[(\mathrm{d}X)^{\mathrm{T}}\nabla_X f(X)\right]\end{aligned}$$

The second term goes to zero if $||\mathrm{d}X||$ goes to zero. Hence taking the limit as in (B.1) gives

$$\begin{aligned}&\mathrm{tr}\left[(\mathrm{d}X)^{\mathrm{T}}2A^{\mathrm{T}}(AX-B)\right]\Big|_{||\mathrm{d}X||\to 0} = \mathrm{tr}\left[(\mathrm{d}X)^{\mathrm{T}}\nabla_X f(X)\right]\Big|_{||\mathrm{d}X||\to 0}, \\ &\implies \nabla_X f(X) = 2A^{\mathrm{T}}(AX-B).\end{aligned}$$

C The generalized Schur method for solving generalized Sylvester equations

In this appendix, our goal is to solve the generalized Sylvester equation

$$\begin{aligned} AR - LB &= C, \\ DR - LE &= F, \end{aligned} \tag{C.1}$$

where $L, R \in \mathbb{R}^{m\times n}$ are unknown, $A, D \in \mathbb{R}^{m\times m}$, $B, E \in \mathbb{R}^{n\times n}$ and $C, F \in \mathbb{R}^{m\times n}$ are known. The following theory is based on the results of [28] and is a generalization of the Schur method for solving the Sylvester equation $AX + XB = C$. We refer to [32, Appendix A] for more details about the history and relation between the Sylvester equation and the generalized Sylvester equation. The generalized Sylvester equation (C.1) can be formulated in terms of a block-diagonalizing equivalence transformation $P^{-1}(M - \lambda N)Q$ of the matrix pencil

$$M - \lambda N = \begin{bmatrix} A & -C \\ 0 & B \end{bmatrix} - \lambda \begin{bmatrix} D & -F \\ 0 & E \end{bmatrix}.$$

Solving (C.1) is equivalent to solving the following equation for L and R:

$$\begin{bmatrix} I & -L \\ 0 & I \end{bmatrix} (M - \lambda N) \begin{bmatrix} I & R \\ 0 & I \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix} - \lambda \begin{bmatrix} D & 0 \\ 0 & E \end{bmatrix}. \tag{C.2}$$

One can show that (C.1) has a unique solution if and only if the regular pencils $A - \lambda D$ and $B - \lambda E$ have disjoint spectra [17]. If they have common spectra of if they are singular, the generalized Sylvester equation will not in general have a solution.

C.1 Algorithm

In this section, we present a generalization of the Schur method [33] for solving $AX + XB = C$. This method is based on the equivalence between (C.1) and

$$\begin{aligned} P^{\mathrm{T}}AQQ^{\mathrm{T}}RV - P^{\mathrm{T}}LUU^{\mathrm{T}}BV &= P^{\mathrm{T}}CV, \\ P^{\mathrm{T}}DQQ^{\mathrm{T}}RV - P^{\mathrm{T}}LUU^{\mathrm{T}}EV &= P^{\mathrm{T}}FV, \end{aligned} \tag{C.3}$$

where the matrices P, Q, U and V are all unitary. The solution of (C.3) involves the following four steps, which will be clarified later on.

1. Transform (A, D) and (B, E) via the *QZ-algorithm* into the simpler form, which gives A_1 and B_1 upper quasi-triangular (see (4.23)) and C_1 and D_1 upper triangular

$$\begin{aligned} (A_1, D_1) &:= (P^{\mathrm{T}}AQ, P^{\mathrm{T}}DQ), \\ (B_1, E_1) &:= (U^{\mathrm{T}}BV, U^{\mathrm{T}}EV). \end{aligned}$$

2. Modify the right-hand sides (C, F):

$$\begin{aligned} C_1 &:= P^{\mathrm{T}}CV, \\ F_1 &:= P^{\mathrm{T}}FV. \end{aligned}$$

3. Solve the transformed system for L_1 and R_1:

$$\begin{aligned} A_1R_1 - L_1B_1 &= C_1, \\ D_1R_1 - L_1E_1 &= F_1. \end{aligned} \tag{C.4}$$

4. Transform the solution back to the original system:

$$L := PL_1U^{\mathrm{T}}, \quad R := QR_1V^{\mathrm{T}}$$

Step 1 and Step 2 are straightforward to follow. Step 3 is a bit more complex to perform. Before we can dig into solving (C.4), note that, as mentioned in Section 4.3.1 for (4.23), the QZ-algorithm gives A_1 and B_1 upper quasi-triangular matrices and D_1 and E_1 upper triangular. Upper quasi-triangular means that the matrix has either 1×1 or 2×2 blocks on its diagonal, where in this case the 2×2 blocks correspond to pairs of complex conjugate eigenvalues of the matrix pencils $A - \lambda D$ and $B - \lambda E$ (See also [34]).

Assume that A_1 consists of p^2 blocks A_{ij}, where the p diagonal blocks are either 1×1 or 2×2. In the same way, B_1 has q^2 blocks, where the q B_{ii} blocks are 1×1 or 2×2. Moreover, assume that C_1, D_1, E_1, F_1, L_1 and R_1 are partitioned consistently with A_1 and B_1. Then the solution (L_1, R_1) of (C.4) can be computed by the *GS-algorithm* as follows:

Algorithm C.1. GS-algorithm.

```
1         %The GS-algorithm
2
3         for j = 1:q
4             for i = p:1
5                 %Solve the subsystem:
6                 A_ii*R_ij - L_ij*B_jj = C_ij;
7                 D_ii*R_ij - L_ij*E_jj = F_ij;
8                 %Substitute R_ij and L_ij into remaining equations
9                 for k = 1:i-1       %block-column j
10                    C_kj = C_kj - A_ki*R_ij;
11                    F_kj = F_kj - D_ki*R_ij;
12                end
13                for k = j+1:q       %block-row i
14                    C_ik = C_ik + L_ij*B_jk;
15                    F_ik = F_ik + L_ij*E_jk;
16                end
17            end
18        end
```

We have written the pseudo-code of the GS-algorithm in Matlab style for readibility purposes. Solving the subsystem in the above algorithm corresponds with following the upper quasi-triangular structure of (C.4). In each new subsystem, there are enough zero- and already solved blocks of (L_1, R_1) to solve the current subsystem.
The steps after solving the subsystem in the GS-algorithm can be associated with 'moving' the known parts of the equation to the right-hand side.

By looking carefully at (C.3) one can see that L and R have been transformed to L_1 and R_1 as

$$L_1 := P^{\mathrm{T}}LU, \quad R_1 := Q^{\mathrm{T}}RV$$

and therefore L and R can be found as stated in step 4.