CHEMNITZ UNIVERSITY OF TECHNOLOGY

Faculty of Mathematics

Mathematics in Industry and Technology

Diploma Thesis

$\mathcal{L}_\infty$-Norm Computation for Descriptor Systems

by
Matthias Voigt
born on April 16, 1986 in Erlabrunn

1st Advisor: Prof. Dr. rer. nat. habil. Peter Benner
(Chemnitz UT, Germany)
2nd Advisor: Dr. Ing. Math. Vasile Sima
(ICI, Bucharest, Romania)

Date of Submission: June 8, 2010
Date of Defense: July 8, 2010

URL: http://archiv.tu-chemnitz.de/pub/2010/0105

Voigt, Matthias
$\mathcal{L}_\infty$-Norm Computation for Descriptor Systems
Diploma Thesis, Faculty of Mathematics
Chemnitz University of Technology, July 2010

Assignment of Tasks

The goal of this diploma thesis is to develop a method for the computation of the $\mathcal{L}_\infty$-norm for descriptor systems. For this purpose an already existing iterative method for the computation of the $\mathcal{L}_\infty$-norm of dynamical systems in state space form should be used and adapted to descriptor systems. To do so it is necessary to develop an algorithm to check the properness of the corresponding transfer functions. Additionally a structure-preserving method for the computation of the eigenvalues of the arising skew-Hamiltonian/Hamiltonian matrix pencils has to be investigated in order to guarantee reliability of the method. Finally the algorithms should be implemented in FORTRAN as `SLICOT`-style routines and their runtime and accuracy should be analyzed.

Aufgabenstellung

Das Ziel der Diplomarbeit liegt darin, ein Verfahren zur Berechnung der $\mathcal{L}_\infty$-Norm für Deskriptorsysteme zu entwickeln. Dazu soll ein bestehendes iteratives Verfahren zur Berechnung der $\mathcal{L}_\infty$-Norm dynamischer Systeme in Zustandsraumform verwendet und für Deskriptorsysteme angepasst werden. Dazu ist es nötig, ein Verfahren zur Überprüfung der Properness der entsprechenden Übertragungsfunktionen zu entwikkeln. Zusätzlich ist ein strukturerhaltendes Verfahren zur Berechnung der Eigenwerte der auftretenden schief-Hamiltonisch/Hamiltonischen Matrixbüschel zu untersuchen, um die Zuverlässigkeit des Verfahrens zu gewährleisten. Schlussendlich sollen die Algorithmen in FORTRAN in der Form von `SLICOT`-Routinen implementiert und deren Laufzeit und Genauigkeit untersucht werden.

Acknowledgement

First of all I would like to thank my advisor Prof. Dr. Peter Benner for his constant support while writing this thesis. Furthermore I am very grateful for the opportunity to work for him as student assistant and for the funding of all the travels to workshops during the last months. I would also like to thank my family for all the financial and personal support during the past six years. Last but not least I thank

- **Dr. Vasile Sima for the insane work on checking my Fortran codes, examining this thesis, and helpful advice,**
- **Dr. Timo Reis for giving me the idea for the properness test and some test examples,**
- **Dr. Tatjana Stykel for providing the codes of her test examples,**
- **Philip Losse for answering all my questions on skew-Hamiltonian/Hamiltonian matrix pencils,**
- **Prof. Dr. Christian Mehl for helping me with the theorem on the existence of real structured Schur forms,**
- **Dr. Jens Saak and Patrick Kürschner for all the nice coffee breaks, fruitful discussions and the help on forcing LaTeX to do what I want,**
- **Frederik Beuth for reading parts of this thesis and giving me advice,**
- **all my friends that made my study life to that great time.**

Abstract

In many applications from industry and technology computer simulations are performed using models which can be formulated by systems of differential equations. Often the equations underlie additional algebraic constraints. In this context we speak of descriptor systems. Very important characteristic values of such systems are the $\mathcal{L}_\infty$-norms of the corresponding transfer functions. The main goal of this thesis is to extend a numerical method for the computation of the $\mathcal{L}_\infty$-norm for standard state space systems to descriptor systems. For this purpose we develop a numerical method to check whether the transfer function of a given descriptor system is proper or improper and additionally use this method to reduce the order of the system to decrease the costs of the $\mathcal{L}_\infty$-norm computation. When computing the $\mathcal{L}_\infty$-norm it is necessary to compute the eigenvalues of certain skew-Hamiltonian/Hamiltonian matrix pencils composed by the system matrices. We show how we extend these matrix pencils to skew-Hamiltonian/Hamiltonian matrix pencils of larger dimension to get more reliable and accurate results. We also consider discrete-time systems, apply the extension strategy to the arising symplectic matrix pencils and transform these to more convenient structures in order to apply structure-exploiting eigenvalue solvers to them. We also investige a new structure-preserving method for the computation of the eigenvalues of skew-Hamiltonian/Hamiltonian matrix pencils and use this to increase the accuracy of the computed eigenvalues even more. In particular we ensure the reliability of the $\mathcal{L}_\infty$-norm algorithm by this new eigenvalue solver. Finally we describe the implementation of the algorithms in FORTRAN and test them using two real-world examples.

Contents

List of Figures

List of Tables

List of Algorithms

1 Introduction

In natural and engineering sciences, modeling and numerical simulation have become the third pillar of research besides theoretical investigations and experiments. The reason is that on the one hand many complicated problems cannot be solved analytically; just think of the solution of certain partial differential equations. On the other hand, experiments are often very costly, time-consuming and expensive or simply cannot be performed. In this way simulation has become very important in the scientific community within the last years. Modern computer architectures allow to solve even extremely difficult and large problems. But also mathematical methods for modeling and numerically solving problems have had to evolve very quickly since without these, simulation would be unimaginable. Most often, the dynamics of the modelled systems is represented by differential equations. These could be partial differential equations but here we only deal with the simple case of ordinary differential equations (with constant coefficients). Sometimes there are algebraic constraints which prevent the system to attain every possible state. Imagine for instance a simple pendulum. This pendulum is forced to move on a circle and hence cannot reach every possible position in space. In this framework we speak of differential-algebraic equations (or descriptor systems)

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

with a singular matrix E. Generally differential-algebraic equations are more difficult to analyze and to solve than usual differential equations. Besides the simulation one is often interested in optimizing or controlling the systems to obtain certain system properties. In this way we apply a control u to get the wanted behavior, i.e.,

$$E\dot{x}(t) = Ax(t) + Bu(t).$$

Quite often we do not know all values of the internal state variables x but we can do some measurements to obtain information about our system, thus we get

$$y(t) = Cx(t) + Du(t),$$

where y denotes the vector of measured outputs. A descriptor system can also be interpreted as a black box as in Figure 1.1 which gets an input and gives us an output under certain rules.

In the next section we describe some applications of descriptor systems, first we model in a detailed way the equations for an electrical circuit. Later we briefly describe the problem of robust control.

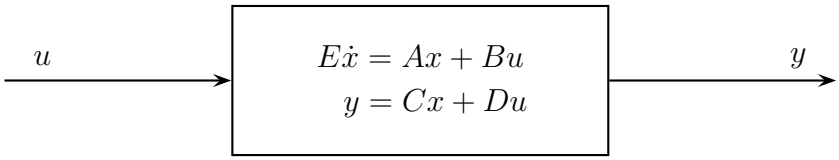


Figure 1.1: Black box interpretation of a descriptor system

1.1 Two Motivating Examples and Applications

1.1.1 Example 1: Modeling of an Electrical Circuit

Goal of our Modeling

To illustrate the importance of descriptor systems we show on an example how these arise in the modeling of electrical circuits. For this purpose we investigate the following example circuit which has been taken from [Rei09]. It contains a voltage source with $u_V(t) = u_1(t)$ and a current source $i_I(t) = i_6(t)$ which serve as inputs for our system, i.e., these parameters may be chosen by the user in an appropriate range. Furthermore the circuit consists of a coil with inductivity L, a capacitor with capacity C, a resistor with resistance R and an ideal autotransformer with the ratio $T = \frac{N_1}{N_2}$ of the number of turns in the primary and secondary winding, respectively. The goal of our modeling is to get the voltages and currents of all electrical components of the circuit as well as the voltage at the current source u_I and the current at the voltage source i_V.

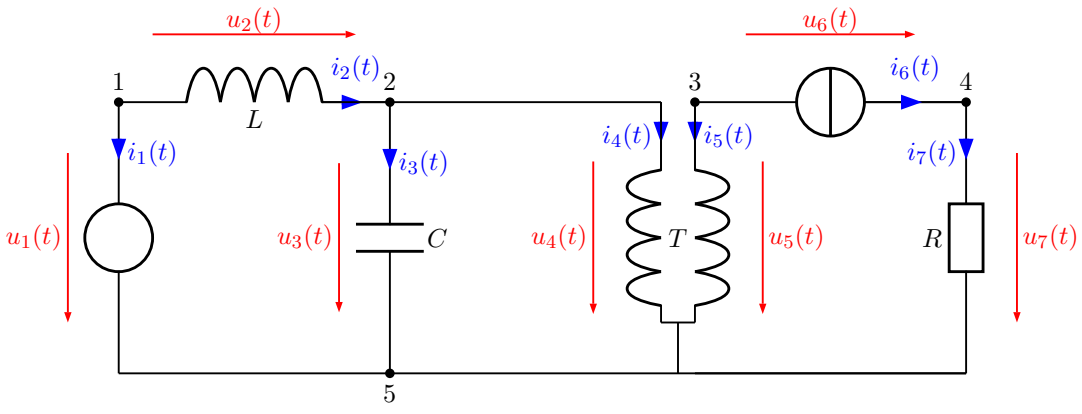


Figure 1.2: Example circuit with currents and voltages on each component

Component Laws

First we can evaluate physical laws that connect the voltage and the current of a specific component. In our example we have the following relations (see [Rei09] or any introduction to electrical engineering, e.g. [Pre09]):

$$\begin{aligned} u_1(t) &= u_V(t), \\ u_2(t) &= L \cdot \frac{\mathrm{d}}{\mathrm{d}t} i_2(t), \\ i_3(t) &= C \cdot \frac{\mathrm{d}}{\mathrm{d}t} u_3(t), \\ u_5(t) &= T \cdot u_4(t),\ i_4(t) = T \cdot i_5(t), \\ i_6(t) &= i_I(t), \\ u_7(t) &= R \cdot i_7(t). \end{aligned} \tag{1.1}$$

Application of Kirchhoff's Laws

Next we model the relations between the components. For this purpose we introduce some terms that we are going to work with (see [Sch08a]).

Definition 1.1 (Some Terms Connected to Electrical Networks)**.** (i) An *electrical network* is any interconnection of electrical elements.

(ii) A *node* of a network is a point where the connection lines between the elements of the network meet. It is pictured as a point.

(iii) A *branch* is a direct current path between two nodes of a network. It consists of the part from one node to the circuit element and the part from the element to the other node. It is pictured as a line.

(iv) A *loop or mesh* is a closed connection within the network. It consists of an arbitrary number of connected branches.

(v) A *cut-set* of an electrical network is a subcircuit which is built by removing branches of the original circuit.

Remark 1.1. The whole network we use is supposed to be electrically ideal, that means that for example the connections between the elements as well as the nodes have no resistance and there are no interactions between neighboring branches or elements without connection.

Now we can start investigating the currents of the network. For this we need the incidence matrix (see [Rei09]) of the network which expresses how the circuit elements are interconnected.

Definition 1.2 (Incidence Matrix)**.** The *incidence matrix* $A = (a_{ij})$ of the electrical circuit is constructed by

$$a_{ij} = \begin{cases} 1 & : j\text{-th branch "begins" in node } i, \\ -1 & : j\text{-th branch "ends" in node } i, \\ 0 & : \text{otherwise.} \end{cases}$$

Now we can apply the following fundamental theorem of electrical engineering.

Theorem 1.1 (Kirchhoff's Current Law (KCL))**.** *The algebraic sum of currents traversing each cut-set of the network must be equal to zero at every instant of time. Special case: The sum of currents leaving any circuit node is zero:*

$$A \cdot I(t) = 0$$

with the incidence matrix A and the corresponding vector of branch currents $I(t)$.

Example 1.1.

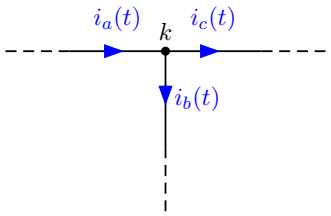


Figure 1.3: Example for KCL

Consider the example node of Figure (1.3) with one incoming current $i_a(t)$ and two outgoing currents $i_b(t)$ and $i_c(t)$. By KCL we have

$$-i_a(t) + i_b(t) + i_c(t) = 0.$$

Applying KCL to our example circuit yields

$$\underbrace{\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ -1 & 0 & -1 & -1 & -1 & 0 & -1 \end{bmatrix}}_{=: \text{ incidence matrix } A} \begin{bmatrix} i_1(t) \\ i_2(t) \\ i_3(t) \\ i_4(t) \\ i_5(t) \\ i_6(t) \\ i_7(t) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{1.2}$$

It can be easily observed that the rows in the incidence matrix in (1.2) are linearly dependent, so we can delete, e.g., the last row of A to obtain a linear system of

equations with reduced incidence matrix $\tilde{A}$

$$\underbrace{\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}}_{=:\tilde{A}} \begin{bmatrix} i_1(t) \\ i_2(t) \\ i_3(t) \\ i_4(t) \\ i_5(t) \\ i_6(t) \\ i_7(t) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{1.3}$$

In the next step we analyse the voltages of our example network. To do so we can use the following law.

Theorem 1.2 (Kirchhoff's Voltage Law (KVL))**.** *The algebraic sum of voltages within each loop of the network must be equal to zero at every instant of time. This is often used for getting a relation between branch voltage and node voltage in the form:*

$$A^T \cdot u(t) = U(t) \tag{1.4}$$

with the incidence matrix A, the corresponding vector of branch voltages $U(t)$, and the vector of node voltages $u(t)$.

Example 1.2.

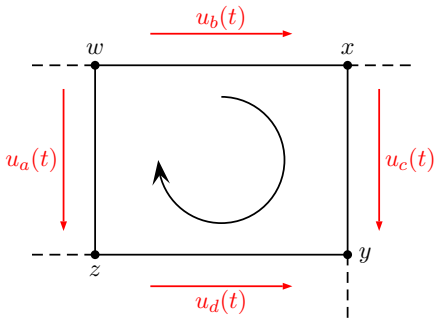


Figure 1.4: Example for KVL

Consider the example loop of Figure 1.4 with two voltages $u_b(t)$ and $u_c(t)$ in voltage direction and two voltages $u_a(t)$ and $u_d(t)$ in the opposite direction. By KVL it holds

$$-u_a(t) + u_b(t) + u_c(t) - u_d(t) = 0.$$

To obtain the relation between branch voltages and node voltages we can express the branch voltages as differences of the node voltages (potentials).

Example 1.3.

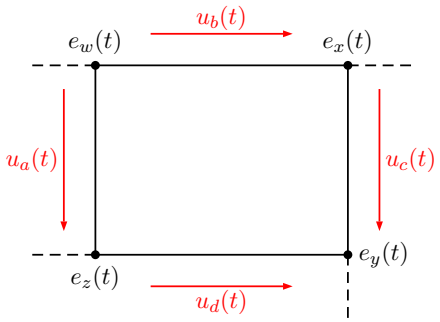


Figure 1.5: Example for KVL (equivalent formulation)

We consider the same example loop as in Figure 1.4 and receive an equivalent formulation as in Figure 1.5 of KVL by expressing the voltages in each branch by the difference of the node potentials of the corresponding outgoing and incoming nodes as follows:

$$\begin{aligned} u_a(t) &= e_w(t) - e_z(t), \\ u_b(t) &= e_w(t) - e_x(t), \\ u_c(t) &= e_x(t) - e_y(t), \\ u_d(t) &= e_z(t) - e_y(t). \end{aligned}$$

Now we assign to every node i, $i = 1, \ldots, 5$, of our example circuit a node potential $e_i(t)$ and by analyzing the network structure we get the following result for expressing the branch voltages $u_j(t)$, $j = 1, \ldots, 7$, by differences of node potentials

$$\begin{bmatrix} u_1(t) \\ u_2(t) \\ u_3(t) \\ u_4(t) \\ u_5(t) \\ u_6(t) \\ u_7(t) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & -1 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}}_{=:A^T} \begin{bmatrix} e_1(t) \\ e_2(t) \\ e_3(t) \\ e_4(t) \\ e_5(t) \end{bmatrix}. \tag{1.5}$$

Remark 1.2. In our example equation (1.4) is equivalent to (1.5).

As only differences of potentials are unique but potentials themselves are not, it is appropriate to choose one reference node and set its potential to zero. In our case we set $e_5(t) = 0$. This means that node 5 becomes a ground node. As in the considerations for the currents we obtain a reduced system of equations with $\tilde{A}^T$ as

system matrix

$$\begin{bmatrix} u_1(t) \\ u_2(t) \\ u_3(t) \\ u_4(t) \\ u_5(t) \\ u_6(t) \\ u_7(t) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{=:\tilde{A}^T} \begin{bmatrix} e_1(t) \\ e_2(t) \\ e_3(t) \\ e_4(t) \end{bmatrix}. \tag{1.6}$$

Setup of the Descriptor System

By using the component laws (1.1) and the two linear systems of equations (1.3) and (1.6) we can eliminate all voltages and some of the currents by substitutions. Since the amount of equations is quite large for explaining this in detail we just state the result. The equation

$$\underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & C & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & L & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{E} \underbrace{\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \\ \dot{e}_4 \\ \dot{i}_2 \\ \dot{i}_5 \\ \dot{i}_1 \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & -T & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & R^{-1} & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & T & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{F} \underbrace{\begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ i_2 \\ i_5 \\ i_1 \end{bmatrix}}_{x}$$
$$+ \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & -1 \\ 0 & 0 \\ 0 & 0 \\ -1 & 0 \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} u_V(t) \\ i_I(t) \end{bmatrix}}_{u} \tag{1.7}$$

is the state equation of our system. It expresses the behavior of our descriptor vector x depending on the systems input u. Note that E is singular so we have got a "real"

descriptor system. Now we still need an equation for our output vector y, that is

$$\underbrace{\begin{bmatrix} i_V(t) \\ u_I(t) \end{bmatrix}}_{y} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 \end{bmatrix}}_{G} \underbrace{\begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ i_2 \\ i_5 \\ i_1 \end{bmatrix}}_{x} \tag{1.8}$$

and expresses the current at the voltage source $i_V(t)$ and the voltage at the current source $u_I(t)$ in dependence of the state vector x. The equations (1.7) and (1.8) together form a descriptor system which has specific structural properties in this context. In the sequel we analyse those in a more general form because descriptor systems arise in a great variety of applications and do not always have certain structures.

1.1.2 Example 2: Robust Control

We now briefly describe an application of control theory (see [LMPR08, BLM+08] for a detailed problem description and solution) which also requires the system norms we consider in this thesis. Often the input of our system suffers from stochastic disturbances, this means that the input splits in two parts, so instead of $Bu(t)$ we have

$$B_1 w(t) + B_2 u(t),$$

where $w(t)$ now has the interpretation of an *exogenous* input that may include noise, linearization errors and unmodeled dynamics. Similarly, we can also split the output into two parts, i.e., instead of $y(t) = Cx + Du$ we write

$$\begin{aligned} z(t) &= C_1 x(t) + D_{11} w(t) + D_{12} u(t), \\ y(t) &= C_2 x(t) + D_{21} w(t) + D_{22} u(t), \end{aligned}$$

where $z(t)$ has the meaning of a *regulated output* or an *estimation error*. Now the control inputs are determined by the measured outputs of the system in order to react dynamically to the system's behavior. This is done by a so called *controller* or *dynamic compensator* as in Figure 1.6 which is a descriptor system of the form

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

The goal of $\mathcal{H}_\infty$- or robust control is now to find a controller such that the *closed-loop system*, resulting from the combination of the original system and the controller as in Figure 1.6 with w as input and z as output, has the following properties:

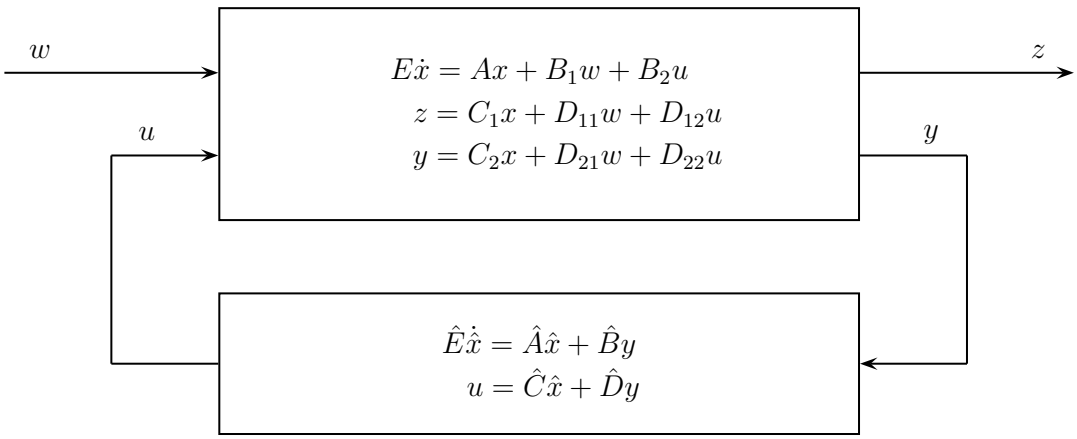


Figure 1.6: Closed-loop diagram of a descriptor system and a controller

(i) the uncontrolled system is internally stable, i.e., $\lim_{t\to\infty} \begin{bmatrix} x(t) & \hat{x}(t) \end{bmatrix}^T = 0$ for $w(t) \equiv 0$,

(ii) its "size" is reasonably small in order to make the worst-case influence of the disturbances as small as possible. In our case, the "size" is the $\mathcal{H}_\infty$-norm of the corresponding transfer function which is a special case of the $\mathcal{L}_\infty$-norm. We define these terms precisely later in this thesis.

1.2 Outline of this Work

In Chapter 2 we repeat some basic concepts of linear algebra and systems and control theory. This is immediately done in the framework of descriptor systems since these are in the focus of this thesis. In Chapter 3 we introduce a numerical method which tests if a transfer function obtained by a given state space realization is proper or not. This is important as we must know for the algorithm in Chapter 4 if a transfer function has this property. In Chapter 4 we explain in a very detailed way how our method for computing the $\mathcal{L}_\infty$-norm of a transfer function works and how we can treat the problems which do not occur in the case of standard state space systems. As our algorithm is based on computing the eigenvalues of matrix pencils with certain structure, in Chapter 5 we derive an algorithm which exploits and preserves the given matrix structures in order to achieve higher accuracy in the computed eigenvalues. In Chapter 6 we present some details of the implementation of the algorithms in Fortran and test these with respect to runtime and accuracy using some real-world examples. Finally, in Chapter 7 we summarize our findings and state some open problems and topics for further research.

2 Fundamentals from Linear Algebra and Systems and Control Theory

In this chapter we present some concepts from linear algebra and basic theory for descriptor systems. We immediately consider descriptor systems as these are the main focus of this thesis. For the theory of standard systems we refer the reader to any introduction on systems and control theory, e.g., [HP05, Dat04].

2.1 Matrices

In this section we recall some basic spectral properties of matrices and state some important matrix decompositions (see [GVL96]).

2.1.1 Eigenvalues, Eigenvectors and Invariant Subspaces

Definition 2.1 (Spectrum, Eigenvalue, Eigenvector)**.** Let $A \in \mathbb{R}^{n\times n}$. The set of numbers $\lambda \in \mathbb{C}$ for which the characteristic polynomial $P(\lambda) := \det(A - \lambda I)$ vanishes, is called *spectrum* of the matrix A. It is often denoted by $\Lambda(A)$. An element of $\Lambda(A)$ is called *eigenvalue* of A. A nonzero vector $x \in \mathbb{R}^n$ that satisfies

$$Ax = \lambda x$$

is termed *eigenvector* corresponding to the eigenvalue λ.

Definition 2.2 (Invariant Subspace)**.** A subspace $\mathcal{L} \subset \mathbb{R}^n$ is called *invariant subspace* of the matrix $A \in \mathbb{R}^{n\times n}$ if

$$A\mathcal{L} \subset \mathcal{L}$$

holds.

Remark 2.1. Every set composed of eigenvectors of the matrix A spans an invariant subspace of A.

2.1.2 Some Matrix Decompositions

Theorem 2.1 (Reduction to Jordan Canonical Form)**.** *Every matrix $A \in \mathbb{R}^{n\times n}$ can be transformed to Jordan canonical form by a change of basis, that is there exists a*

nonsingular matrix $T \in \mathbb{C}^{n\times n}$ *such that*

$$J := T^{-1}AT = \begin{bmatrix} J_1 & & & \\ & J_2 & & \\ & & \ddots & \\ & & & J_r \end{bmatrix}, \tag{2.1}$$

where each of the submatrices $J_k \in \mathbb{R}^{n_k\times n_k}$ *has the form*

$$J_k = \begin{bmatrix} \lambda_k & 1 & & & \\ & \lambda_k & 1 & & \\ & & \ddots & \ddots & \\ & & & \ddots & 1 \\ & & & & \lambda_k \end{bmatrix}, \quad k = 1, 2, \ldots, r. \tag{2.2}$$

Remark 2.2. By reducing the matrix A to Jordan canonical form it is possible to display its full eigenstructure. However, the transformation matrix T may be arbitrarily ill-conditioned, thus a numerically stable computation of (2.1) – (2.2) may be impossible.

Theorem 2.2 (Singular Value Decomposition (SVD))**.** *Let* $A \in \mathbb{R}^{m\times n}$ *with* $\operatorname{rank}(A) = r$*. Then there exist orthogonal matrices* $U \in \mathbb{R}^{m\times m}$ *and* $V \in \mathbb{R}^{n\times n}$ *such that*

$$A = U \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} V^T, \tag{2.3}$$

where $\Sigma = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r)$ *with* $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_r > 0$*. The decomposition* (2.3) *is also known as singular value decomposition or shortly as SVD.*

Theorem 2.3 (QR Decomposition)**.** *Every matrix* $A \in \mathbb{R}^{m\times n}$ *can be decomposed as*

$$A = QR$$

where $Q \in \mathbb{R}^{m\times m}$ *is orthogonal and* $R \in \mathbb{R}^{m\times n}$ *is upper triangular.*

Theorem 2.4 (Real Schur Decomposition)**.** *If* $A \in \mathbb{R}^{n\times n}$*, then there exists an orthogonal matrix* Q *such that*

$$Q^T AQ = \begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1k} \\ & R_{22} & & \vdots \\ & & \ddots & \vdots \\ & & & R_{kk} \end{bmatrix} \tag{2.4}$$

is in upper quasi-triangular form, that is a block R_{ii}*,* $i = 1, \ldots, k$ *is either* 1×1 *and a real eigenvalue of* A *or* 2×2 *and its eigenvalues are a complex conjugate pair of eigenvalues of* A*.*

2.2 Matrix Pencils

We consider now a generalization of matrices which plays an important role in the analysis of descriptor systems. We also state generalizations of some of the decompositions above.

2.2.1 Eigenvalues, Eigenvectors and Deflating Subspaces

Definition 2.3 (Matrix Pencil)**.** A pair of matrices (A, E) with $A, E \in \mathbb{R}^{m\times n}$ is called *matrix pencil*. There are many ways to denote a matrix pencil. Throughout this thesis we denote a matrix pencil by $A - \lambda E$.

Definition 2.4 (Spectrum, Eigenvalue, Eigenvector)**.** Let $A, E \in \mathbb{R}^{n\times n}$. The set of numbers $\lambda \in \mathbb{C}$ for which the characteristic polynomial $P(\lambda) := \det(A-\lambda E)$ vanishes, is called *spectrum* of the matrix pencil $A - \lambda E$. We denote the spectrum by $\Lambda(A, E)$. An element of $\Lambda(A, E)$ is called *(generalized) eigenvalue* of $A - \lambda E$. A nonzero vector $x \in \mathbb{R}^n$ that satisfies

$$Ax = \lambda Ex$$

is termed *(generalized) eigenvector* corresponding to the (generalized) eigenvalue λ.

Definition 2.5 (Regular/Singular Matrix Pencil)**.** Let $A - \lambda E$ be a square matrix pencil. The matrix pencil is called *regular*, if there exists a $\lambda \in \mathbb{C}$ such that $\det(A - \lambda E) \neq 0$. Otherwise the matrix pencil is termed *singular*.

Remark 2.3. Note that the matrix E in the matrix pencil $A - \lambda E$ may be singular. In this case the spectrum $\Lambda(A, E)$ contains infinite eigenvalues.

Example 2.1. (i) Consider the 3×3 matrix pencil

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad E = \begin{bmatrix} 1 & 2 & -2 \\ 0 & 3 & 1 \\ 0 & 0 & 0 \end{bmatrix}.$$

By simply computing the eigenvalues we obtain $\Lambda(A, E) = \{1, 2, \infty\}$.

(ii) Consider another 3×3 matrix pencil

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad E = \begin{bmatrix} 1 & 2 & -2 \\ 0 & 3 & 1 \\ 0 & 0 & 0 \end{bmatrix}.$$

Now we see that $\det(A - \lambda E) = 0$ for all $\lambda \in \mathbb{C}$. Consequently $\Lambda(A, E) = \mathbb{C}$ which means that the matrix pencil is singular.

Now we come to a generalization of invariant subspaces for matrices.

Definition 2.6 (Deflating Subspace)**.** Let A, $E \in \mathbb{R}^{n\times n}$. A subspace $\mathcal{U} \subset \mathbb{R}^n$ is said to be a *deflating subspace* of the matrix pencil $A-\lambda E$ if there exists another subspace $\mathcal{V} \subset \mathbb{R}^n$ of the same dimension as $\mathcal{U}$ such that

$$A\mathcal{U} \subset \mathcal{V}, \quad E\mathcal{U} \subset \mathcal{V},$$

or equivalently

$$A\mathcal{U} + E\mathcal{U} = \mathcal{V}.$$

Remark 2.4. Similar to the matrix case, every set composed of generalized eigenvectors of the matrix pencil $A - \lambda E$ spans a deflating subspace of $A - \lambda E$.

2.2.2 Some Basic Decompositions of Matrix Pencils

Theorem 2.5 (Reduction to Weierstraß Canonical Form (see, e.g., [Sty06]))**.** *Every regular matrix pencil $A-\lambda E$ can be reduced to Weierstraß canonical form, i.e., there exist nonsingular matrices W, $T \in \mathbb{C}^{n\times n}$ such that*

$$A = W \begin{bmatrix} J & 0 \\ 0 & I_{n_\infty} \end{bmatrix} T, \quad E = W \begin{bmatrix} I_{n_f} & 0 \\ 0 & N \end{bmatrix} T, \tag{2.5}$$

where I_m is the identity matrix of order m, J and N are in Jordan canonical form and N is nilpotent with index of nilpotency ν, where n_f and n_∞ are the dimensions of the deflating subspaces of $A-\lambda E$ corresponding to the finite and infinite eigenvalues, respectively.

Remark 2.5. By reducing the matrix pencil $A - \lambda E$ to Weierstraß canonical form it is again possible to display its full eigenstructure. In particular, the decomposition splits the matrix pencil into the subpencil $J-\lambda I_{n_f}$ that contains all finite eigenvalues and the subpencil $I_{n_\infty} - \lambda N$ that contains all infinite eigenvalues of $A - \lambda E$. As the Weierstraß canonical form contains Jordan blocks it might be impossible to compute it in a numerically stable manner.

Theorem 2.6 (Generalized Real Schur Decomposition)**.** *Let $A - \lambda E \in \mathbb{R}^{n\times n}$ be a given matrix pencil. Then there exist orthogonal matrices Q, $Z \in \mathbb{R}^{n\times n}$ such that*

$$Q^T(A - \lambda E)Z = S - \lambda T$$

where T is upper triangular and S is upper quasi triangular.

2.3 Basic Definitions for Descriptor Systems

In this thesis we consider linear time-invariant descriptor systems of the form

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

for continuous time $t \in \mathbb{R}$ or

$$\begin{aligned} Ex(t+1) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t), \end{aligned} \tag{2.7}$$

for discrete time $t \in \mathbb{Z}$, where E, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$, $D \in \mathbb{R}^{p\times m}$ and $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, $y(t) \in \mathbb{R}^p$ for any possible t. Note that the matrix E is allowed to be singular.

Definition 2.7 (Nomenclature (see, e.g., [Dat04]))**.** Consider an LTI descriptor system as in (2.6) or (2.7).

1. The first equations of (2.6) and (2.7), respectively are called *state equations* and the second ones are called *output or observer equations*.

2. The vectors are called as follows:
   - $x(t)$ - *descriptor vector*,
   - $u(t)$ - *input or control vector*,
   - $y(t)$ - *output vector*.

3. The used matrices are denoted as follows:
   - E - *descriptor matrix*,
   - A - *state matrix*,
   - B - *input or control matrix*,
   - C - *output matrix*,
   - D - *feedthrough matrix*.

4. The number of descriptor variables, i.e., the length of $x(t)$ is called the *order* of the descriptor system.

5. The index of nilpotency of the matrix N in the Weierstraß canonical form (2.5) of the matrix pencil $A - \lambda E$ is called *algebraic index* (or just *index*) of the system.

Sometimes it is suitable to denote the systems above as a 5-tuple $(E; A, B, C, D)$. Note that the matrix E may be singular, so the systems may also contain algebraic equations besides the differential or difference equations.

Definition 2.8 (Restricted System Equivalence (see [Dai89]))**.** We call two systems $(E; A, B, C, D)$ and $\left(\tilde{E}; \tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}\right)$ *restricted system equivalent (r.s.e.)* if their order, number of inputs and outputs are equal, and there exist two nonsingular matrices P and Q such that $\tilde{E} = PEQ$, $\tilde{A} = PAQ$, $\tilde{B} = PB$, $\tilde{C} = CQ$, $\tilde{D} = D$.

Remark 2.6. A transformation between two r.s.e. systems via nonsingular matrices P and Q is also called generalized state space transform.

Remark 2.7. Later we show that two restricted system equivalent systems have the same transfer function and thus behave identically.

Assumption 2.1. Throughout this thesis we assume that the considered descriptor systems are regular, i.e., the corresponding system pencils $A - \lambda E$ are regular.

A very important property of dynamical systems is asymptotic stability.

Definition 2.9 (Asymptotic Stability)**.** The descriptor system (2.6) or (2.7) is called *asymptotically stable* if $\lim_{t\to\infty} x(t) = 0$ for all solutions x of the uncontrolled system $E\dot{x}(t) = Ax(t)$ or $Ex(t+1) = Ax(t)$, respectively (see [Dai89, Sch08a]).

Asymptotic stability can also be expressed by certain conditions which have to hold for the matrix pencil $A - \lambda E$.

Theorem 2.7 (Equivalent Conditions for Asymptotic Stability)**.** *The following conditions are equivalent.*

(i) Descriptor system (2.6) or (2.7) is asymptotically stable.

(ii) In the continuous-time case, all finite eigenvalues of $A - \lambda E$ lie in the open left half-plane, i.e., $\Lambda(A, E) \subset \mathbb{C}^- := \{z \in \mathbb{C} : \mathrm{Re}(z) < 0\}$; in the discrete-time case all finite eigenvalues of $A - \lambda E$ lie on the unit disk, i.e., $\Lambda(A, E) \subset \mathring{\mathbb{D}}_1(0) := \{z \in \mathbb{C} : |z| < 1\}$ (see [Sty06]).

2.4 Solution of a Descriptor System

In this section we present some formulae for the solution trajectories of descriptor systems and state the major differences to standard systems. We need these to formulate some controllability and observability concepts in the next section. This section is again mainly based on [Sok06]. We consider the descriptor systems (2.6) and perform a generalized state space transform such that the system pencil $A-\lambda E$ is

transformed to Weierstraß canonical form (see (2.5)). Using Definition 2.8 we obtain a r.s.e. system

$$\begin{aligned} \dot{x}_1(t) &= Jx_1(t) + B_1u(t), \\ N\dot{x}_2(t) &= x_2(t) + B_2u(t), \\ y(t) &= C_1x_1(t) + C_2x_2(t) + Du(t), \end{aligned} \tag{2.8}$$

where J and N are in Jordan canonical form, N is nilpotent with index of nilpotency ν, and

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = Q^{-1}x(t), \quad \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} = PB, \quad \begin{bmatrix} C_1 & C_2 \end{bmatrix} = CQ$$

with transformation matrices P and Q. The system (2.8) now decouples into two subsystems

$$\begin{aligned} \dot{x}_1(t) &= Jx_1(t) + B_1u(t), \\ y_1(t) &= C_1x_1(t), \end{aligned} \tag{2.9}$$

and

$$\begin{aligned} N\dot{x}_2(t) &= x_2(t) + B_2u(t), \\ y_2(t) &= C_2x_2(t) + Du(t), \end{aligned} \tag{2.10}$$

which are called slow and fast subsystem of (2.8), respectively. The slow subsystem (2.9) is a standard system and therefore it has a unique solution for any initial value $x_1(0)$ and any piecewise continuous input u. This solution is

$$x_1(t) = e^{Jt}x_1(0) + \int_0^t e^{J(t-\tau)}B_1u(\tau)\mathrm{d}\tau. \tag{2.11}$$

For the fast subsystem we get an expression of different structure. Let u be ν times piecewise continuously differentiable. By continuously taking derivatives with respect to t on both sides of (2.10), and multiplying both sides by the matrix N from the left, we obtain the following equations, see [Dai89]:

$$\begin{aligned} N\dot{x}_2(t) &= x_2(t) + B_2u(t), \\ N^2\ddot{x}_2(t) &= N\dot{x}_2(t) + NB_2\dot{u}(t), \\ &\vdots \\ N^\nu x_2^{(\nu)}(t) &= N^{\nu-1}x_2^{(\nu-1)}(t) + N^{\nu-1}B_2u^{(\nu-1)}(t). \end{aligned} \tag{2.12}$$

By adding all these equations and using the fact that $N^\nu = 0$ we get the solution of the fast subsystem

$$x_2(t) = -\sum_{i=0}^{\nu-1} N^i B_2 u^{(i)}(t). \tag{2.13}$$

Comparing (2.11) and (2.13) we see that $x_1(t)$ represents a cumulative effect of $u(\tau)$ on the interval $0 \leqslant \tau \leqslant t$, whereas $x_2(t)$ only depends on the values of u and its derivatives at the same instance of time t. Because of these properties the systems (2.9) and (2.10) are called slow and fast subsystem, respectively.

Remark 2.8. For discrete-time systems of the form (2.7) the situation is quite similar. Following [Dat04] the solution of the slow subsystem with initial value $x_1(0)$ is

$$x_1(t) = J^t x_1(0) + \sum_{i=0}^{t-1} J^{t-1-i} B_1 u(i).$$

For calculating the solution of the fast subsystem we have to replace the i-th derivatives of $x_2(t)$ and $u(t)$ in (2.12) by $x_2(t+i)$ and $u(t+i)$, respectively. This yields

$$x_2(t) = -\sum_{i=0}^{\nu-1} N^i B_2 u(t+i).$$

In this way the solution $x_2(t)$ at the time instance t depends on future inputs if $N^i B_2 \neq 0$ for one $i > 0$. Then the underlying descriptor system is called *uncausal*. Otherwise, if the solution at time t depends only at past inputs, it is called *causal*.

2.5 Controllability and Observability

In this section we want to introduce some important facts about controllability and observability for descriptor systems. Since we only need the concepts of complete controllability and observability in the sequel, we just describe these in this section. A good reference is again [Sok06], a very detailed introduction on this topic can also be found in [Dai89].

Consider a continuous-time descriptor system of the form (2.6).

Definition 2.10 (C-Controllability)**.** System (2.6) is called *completely controllable (C-controllable)* if for any $w \in \mathbb{R}^n$, any initial condition $x(0) \in \mathbb{R}^n$ and any instance of time $t_1 > 0$, there exists a ν times piecewise continuously differentiable control $u \in \mathcal{C}_p^\nu$ such that $x(t_1) = w$.

In accordance with standard systems we can formulate equivalent conditions for a system to be C-controllable.

Theorem 2.8 (Equivalent Conditions for C-Controllability)**.** *The following conditions are equivalent:*

(i) System (2.6) is C-controllable;

(ii) both its slow and fast subsystems are C-controllable;

(iii) $\operatorname{rank}\begin{bmatrix} sE - A & B \end{bmatrix} = n$ *for all* $s \in \mathbb{C}$ *and* $\operatorname{rank}\begin{bmatrix} E & B \end{bmatrix} = n$.

Proof. See [Dai89]. □

Remark 2.9. It can be shown that the slow subsystem of (2.6) is C-controllable if and only if $\operatorname{rank}\begin{bmatrix} sE - A & B \end{bmatrix} = n$ for all $s \in \mathbb{C}$ and that its fast subsystem is C-controllable if and only if $\operatorname{rank}\begin{bmatrix} E & B \end{bmatrix} = n$. C-controllability of the fast subsystem is also often termed as "controllability at infinity" [Dai89].

In analogy we can now define C-observability — the dual concept to C-controllability.

Definition 2.11 (C-Observability)**.** System (2.6) is called *completely observable (C-observable)* if the initial condition $x(0)$ can be uniquely determined from $u(t)$ and $y(t)$, $0 \leqslant t < \infty$.

There exist also equivalent conditions for this property which are summarized in the next theorem.

Theorem 2.9 (Equivalent Conditions for C-Observability)**.** *The following statements are equivalent:*

(i) System (2.6) is C-observable;

(ii) Both its slow and fast subsystems are C-observable;

(iii) $\operatorname{rank}\begin{bmatrix} sE - A \\ C \end{bmatrix} = n$ *for all* $s \in \mathbb{C}$ *and* $\operatorname{rank}\begin{bmatrix} E \\ C \end{bmatrix} = n$.

Proof. See [Dai89]. □

Remark 2.10. It can also be shown that the slow subsystem of (2.6) is C-observable if and only if $\operatorname{rank}\begin{bmatrix} sE - A \\ C \end{bmatrix} = n$ for all $s \in \mathbb{C}$ and that its fast subsystem is C-observable if and only if $\operatorname{rank}\begin{bmatrix} E \\ C \end{bmatrix} = n$. C-observability of the fast subsystem is also often termed as "observability at infinity" [Dai89].

Remark 2.11. When considering discrete-time systems of the form (2.7) the differentiability condition in Definition 2.10 has to be dropped. In contrast we have to assume that at a given instance of time t the control $u(t+i)$ is already known for $i = 0, \ldots, \nu - 1$. The other statements hold without restriction for discrete-time systems as well.

2.6 Frequency Domain Analysis

Often it is very useful to treat a descriptor system in the frequency domain. This approach is especially very popular among engineers. We refer again to [Sok06, Dai89].

2.6.1 Laplace Transform and Transfer Functions

We again consider first continuous-time linear time-invariant descriptor systems.

Definition 2.12 (Laplace Transform)**.** Let $f : \mathbb{R} \longrightarrow \mathbb{R}^n$ be a given function. The function $L\{f\} : \mathbb{R} \longrightarrow \mathbb{R}^n$ defined by

$$L\{f\}(s) = \int_0^\infty e^{-st} f(t) \mathrm{d}t$$

is called *Laplace transform* of f, if the integral exists.

In the sequel we use the two following properties of the Laplace transform that immediately follow from the definition:

(1) If $h = \alpha f + \beta g$ then $L\{h\} = \alpha L\{f\} + \beta L\{g\}$ for arbitrary functions f, g and constants α, $\beta \in \mathbb{C}$.

(2) If f is a differentiable function and $g = f'$ then $L\{g\}(s) = sf - f(0)$.

Applying the Laplace transform to each of the vectors x, u and y of the descriptor system (2.6) and defining $X := L\{x\}$, $U := L\{u\}$, $Y := L\{y\}$ we get

$$\begin{aligned} sEX(s) - Ex(0) &= AX(s) + BU(s), \\ Y(s) &= CX(s) + DU(s). \end{aligned} \tag{2.14}$$

Taking the regularity of the matrix pencil $A - \lambda E$ into account we can eliminate X from (2.14) and obtain

$$Y(s) = C\left(sE - A\right)^{-1}\left(Ex(0) + BU(s)\right) + DU(s).$$

Assuming $Ex(0) = 0$, i.e., $x(0) \in \ker E$ we get the input-output relation

$$Y(s) = \left(C\left(sE - A\right)^{-1} B + D\right) U(s).$$

Definition 2.13 (Transfer Function)**.** The function

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

is called *transfer function* of the descriptor system (2.6).

Often the transfer function is evaluated at values $\mathrm{i}\omega$ where ω then has the physical interpretation of a frequency. In this thesis we often use the following lemma [Sok06].

Lemma 2.1 (Invariance of the Transfer Function)**.** *Let* $(E; A, B, C, D) \longrightarrow (PEQ; PAQ, PB, CQ, D) =: (\tilde{E}; \tilde{A}, \tilde{B}, \tilde{C}, \tilde{D})$ *be a generalized state space transform. Then the systems* $(E; A, B, C, D)$ *and* $(\tilde{E}; \tilde{A}, \tilde{B}, \tilde{C}, \tilde{D})$ *have the same transfer function. In other words, two r.s.e. systems have the same transfer function.*

Proof. The following equations hold for the corresponding transfer functions:

$$\begin{aligned}\tilde{G}(s) : &= \tilde{C}(s\tilde{E} - \tilde{A})^{-1}\tilde{B} + D \\ &= CQ\left(P(sE - A)Q\right)^{-1}PB + D \\ &= CQQ^{-1}(sE - A)P^{-1}PB + D \\ &= C(sE - A)^{-1}B + D = G(s).\end{aligned}$$

□

A special feature of descriptor systems is that even if there are no poles on the imaginary axis, the transfer function might be unbounded on $\mathrm{i}\mathbb{R}$.

Example 2.2. We consider the following transfer function.

$$\begin{aligned}G(s) &= \begin{bmatrix}1 & 1 & 1\end{bmatrix}\left(s\begin{bmatrix}1 & 0 & 0\\ 0 & 0 & 1\\ 0 & 0 & 0\end{bmatrix} - \begin{bmatrix}1 & 0 & 0\\ 0 & -1 & 0\\ 0 & 0 & -1\end{bmatrix}\right)^{-1}\begin{bmatrix}1\\ 1\\ 1\end{bmatrix} \\ &= \begin{bmatrix}1 & 1 & 1\end{bmatrix}\begin{bmatrix}\frac{1}{s-1} & 0 & 0\\ 0 & 1 & -s\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1\\ 1\\ 1\end{bmatrix} \\ &= -s + 2 + \frac{1}{s-1}.\end{aligned}$$

From that we can easily observe that $\lim_{\omega\to\infty} |G(\mathrm{i}\omega)| = \infty$.

This motivates the following definition.

Definition 2.14 ((Strictly) Proper/Improper Transfer Function)**.** A transfer function G is called *proper* if $\lim_{\omega\to\infty} \|G(\mathrm{i}\omega)\| < \infty$ and *strictly proper* if $\lim_{\omega\to\infty} \|G(\mathrm{i}\omega)\| = 0$ for any induced matrix norm $\|\cdot\|$. Otherwise it is *called improper* [Sty06].

Thus it would be desirable to have a numerical method that efficiently tests if a transfer function of a given descriptor system $(E; A, B, C, D)$ is proper or not. This a crucial issue. In the next subsection we see that if the transfer function is improper, its $\mathcal{L}_\infty$-norm is infinite and hence it is not an element of the corresponding $\mathcal{L}_\infty$-space. We establish theoretical preliminaries and develop a numerical algorithm for testing properness in the next chapter.

2.6.2 $\mathcal{L}_\infty$-Spaces and $\mathcal{L}_\infty$-Norm

Now we define some function spaces which we will work with in the sequel.

Definition 2.15 (Spaces $\mathcal{L}_\infty^{p\times m}$ and $\mathcal{RL}_\infty^{p\times m}$ (see [ZD98]))**.** $\mathcal{L}_\infty^{p\times m}(\mathrm{i}\mathbb{R})$ or shortly $\mathcal{L}_\infty^{p\times m}$ is the Banach space of all $p \times m$ matrix-valued functions that are essentially bounded on $\mathrm{i}\mathbb{R}$. The rational subspace of $\mathcal{L}_\infty^{p\times m}$, denoted by $\mathcal{RL}_\infty^{p\times m}(\mathrm{i}\mathbb{R})$ or simply $\mathcal{RL}_\infty^{p\times m}$ consists of all proper (see Definition 2.14) and real rational $p \times m$ transfer functions with no poles on the imaginary axis. As a convenience we just write $\mathcal{L}_\infty$ and $\mathcal{RL}_\infty$ if the dimensions p and m are clear by the context.

Remark 2.12. The transfer functions obtained from linear time-invariant descriptor systems are all rational, i.e., every entry of $G(s)$ is a rational function. On the other hand, every rational function can be seen as a transfer function of a linear time-invariant descriptor system (see, e.g., [Var00]). Hence, in this thesis we only work on the space $\mathcal{RL}_\infty$.

Definition 2.16 ($\mathcal{L}_\infty$-Norm (see [Ben06]))**.** For a matrix-valued function $F \in \mathcal{L}_\infty$ the $\mathcal{L}_\infty$-norm is defined by

$$\|F\|_{\mathcal{L}_\infty} = \operatorname{ess\,sup}_{\omega\in\mathbb{R}} \sigma_{\max}\left(F(\mathrm{i}\omega)\right) \tag{2.15}$$

where $\sigma_{\max}(M)$ is the maximum singular value of the matrix M and $\operatorname{ess\,sup}_{t\in N} h(t)$ is the essential supremum of a function h evaluated on the set N, that is the function's supremum on $N\backslash L$ where L is a set of Lebesgue measure zero.

Remark 2.13. For functions $G \in \mathcal{RL}_\infty$, i.e., for proper transfer function obtained from descriptor systems, equation (2.15) simplifies to

$$\|G\|_{\mathcal{L}_\infty} = \sup_{\omega\in\mathbb{R}} \sigma_{\max}\left(G(\mathrm{i}\omega)\right)$$

since G is continuous on the imaginary axis.

Now we discuss some characterizations and interpretations of the $\mathcal{L}_\infty$-norm as described in [Toi02]. For this purpose we need some facts about norms. First recall

that for a vector-valued input signal $u(t) = \big[u_1(t), \dots, u_m(t)\big]^T$ with $t > 0$, the $\mathcal{L}_2$-norm is given by

$$\|u\|_{\mathcal{L}_2} := \left(\sum_{i=1}^{m} \|u_i\|_{\mathcal{L}_2}^2\right)^{\frac{1}{2}} = \left(\int_0^\infty \sum_{i=1}^{m} u_i(t)^2 \mathrm{d}t\right)^{\frac{1}{2}} = \left(\int_0^\infty u(t)^T u(t) \mathrm{d}t\right)^{\frac{1}{2}}.$$

Similarly, for the Laplace-transformed signal $U(s) = \big[U_1(s), \dots, U_m(s)\big]^T$ we obtain

$$\begin{aligned}
\|U\|_{\mathcal{L}_2} :&= \left(\sum_{i=1}^{m} \|U_i\|_{\mathcal{L}_2}^2\right)^{\frac{1}{2}} \\
&= \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} \sum_{i=1}^{m} |U_i(\mathrm{i}\omega)|^2 \mathrm{d}\omega\right)^{\frac{1}{2}} \\
&= \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} \sum_{i=1}^{m} U_i(-\mathrm{i}\omega) U_i(\mathrm{i}\omega) \mathrm{d}\omega\right)^{\frac{1}{2}} \\
&= \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} U(-\mathrm{i}\omega)^T U(\mathrm{i}\omega) \mathrm{d}\omega\right)^{\frac{1}{2}}. \qquad (2.16)
\end{aligned}$$

Formula (2.16) can then be used to calculate the $\mathcal{L}_2$-norm of the Laplace transformed output $Y(s) = G(s)U(s)$; we obtain

$$\begin{aligned}
\|GU\|_{\mathcal{L}_2} &= \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} U(-\mathrm{i}\omega)^T G(-\mathrm{i}\omega)^T G(\mathrm{i}\omega) U(\mathrm{i}\omega) \mathrm{d}\omega\right)^{\frac{1}{2}} \\
&= \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} \|G(\mathrm{i}\omega) U(\mathrm{i}\omega)\|_2^2 \, \mathrm{d}\omega\right)^{\frac{1}{2}} \\
&\leqslant \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} \left[\|G(\mathrm{i}\omega)\|_2 \, \|U(\mathrm{i}\omega)\|_2\right]^2 \mathrm{d}\omega\right)^{\frac{1}{2}} \\
&\leqslant \sup_{\omega \in \mathbb{R}} \|G(\mathrm{i}\omega)\|_2 \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} \|U(\mathrm{i}\omega)\|_2^2 \, \mathrm{d}\omega\right)^{\frac{1}{2}} \\
&= \|G\|_{\mathcal{L}_\infty} \|U\|_{\mathcal{L}_2}.
\end{aligned}$$

Hence,

$$\|G\|_{\mathcal{L}_\infty} \geqslant \frac{\|GU\|_{\mathcal{L}_2}}{\|U\|_{\mathcal{L}_2}}, \quad \|U\|_{\mathcal{L}_2} \neq 0.$$

In fact, there exist signals which come arbitrarily close to $\|G\|_{\mathcal{L}_\infty}$. Assume that the Laplace transform $U(s)$ is concentrated to a frequency range where $\|G(\mathrm{i}\omega)\|_2$ is arbitrarily close to $\|G\|_{\mathcal{L}_\infty}$ and with components such that $\|G(\mathrm{i}\omega)U(\mathrm{i}\omega)\|_2 / \|U(\mathrm{i}\omega)\|_2$ is arbitrarily close to $\|G(\mathrm{i}\omega)\|_2$. Then it follows that the $\mathcal{L}_\infty$-norm is the operator norm induced by the $\mathcal{L}_2$-norm, i.e.,

$$\|G\|_{\mathcal{L}_\infty} = \sup \left\{ \frac{\|GU\|_{\mathcal{L}_2}}{\|U\|_{\mathcal{L}_2}} : \ U \neq 0 \right\}.$$

The $\mathcal{L}_\infty$-norm gives the maximum factor by which the system magnifies the $\mathcal{L}_2$-norm of any input. Therefore, $\|G\|_{\mathcal{L}_\infty}$ is also called *gain* of the system. There are also other interpretations, but the most important one is the one explained above, for more details see, e.g., [Toi02].

Remark 2.14. Most often it is assumed that the given system is asymptotically stable. Then we deal with the space $\mathcal{H}_\infty$ which is the (closed) subspace of $\mathcal{L}_\infty$ with functions that are analytic and bounded in the open right half-plane. For functions $F \in \mathcal{H}_\infty$ the $\mathcal{H}_\infty$-norm is given by

$$\|F\|_{\mathcal{H}_\infty} := \sup_{\mathrm{Re}(s)>0} \sigma_{\max}\left(F(s)\right) = \sup_{\omega\in\mathbb{R}} \sigma_{\max}\left(F(\mathrm{i}\omega)\right).$$

The second equality can be regarded as a generalization of the maximum modulus theorem for matrix functions (see [BD85] for the statement with proof). Again, we can define the real rational subspace of $\mathcal{H}_\infty$ which is denoted by $\mathcal{RH}_\infty$ and consists of all proper and real rational stable transfer functions (see [ZD98]). Since the definitions of the $\mathcal{L}_\infty$- and $\mathcal{H}_\infty$-norm are identical, our algorithms are also able to compute $\mathcal{H}_\infty$-norms. The theory of our algorithm does not require stability of the transfer function and so we can immediately work with the more general concept of $\mathcal{L}_\infty$-norms.

Remark 2.15. For discrete-time system we have to apply the z-transform

$$Z\{f\}(z) = \sum_{i=0}^{\infty} f(i) z^{-i}$$

to x, u, and y in order to determine the transfer function of the system [Dat04, NR07]. This transfer function is identical to the one of the continuous-time case but mostly the variable s is replaced by z. It should also be noticed that the stability region of discrete-time systems is the unit disk $\mathring{\mathbb{D}}_1(0)$, thus the $\mathcal{L}_\infty$-norm is only defined for transfer functions without any poles on the unit circle.

3 Testing Properness of a Transfer Function

In this chapter a new method for testing if the transfer function of a given descriptor system $(E; A, B, C, D)$ is proper (see Definition 2.14) is introduced. In the first part some important theoretical results are presented whereas in the second part an efficient testing routine is developed.

3.1 Theoretical Background

In this section we present and prove a theorem which establishes the basis for our properness testing routine. But first of all we need the following lemma.

Lemma 3.1 (Inverse of a 2×2 Block Matrix (see [TT09]))**.** *Let $M \in \mathbb{R}^{n \times n}$ with*

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

be a given nonsingular matrix. If the matrix A is nonsingular, the Schur complement $S_A = D - CA^{-1}B$ is also nonsingular and the inverse of M can be written in the following form:

$$M^{-1} = \begin{bmatrix} A^{-1} + A^{-1}BS_A^{-1}CA^{-1} & -A^{-1}BS_A^{-1} \\ -S_A^{-1}CA^{-1} & S_A^{-1} \end{bmatrix}.$$

In the literature this is often called the Banachiewicz inversion formula for the inverse of a nonsingular partitioned matrix.

Theorem 3.1 (Properness of a Transfer Function)**.** *Let $(E; A, B, C, D)$ be a descriptor system with C-controllable and C-observable fast subsystem and transfer function G. Let furthermore*

$$UEV = \begin{bmatrix} T & 0 \\ 0 & 0 \end{bmatrix} \tag{3.1}$$

be a decomposition of the matrix E by a generalized state space transform with nonsingular matrices U, $V \in \mathbb{R}^{n \times n}$ and a full-rank matrix $T \in \mathbb{R}^{r \times r}$. If we apply the same transformations to the matrix A and partition the blocks as in (3.1), i.e.,

$$UAV = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

with $A_{11} \in \mathbb{R}^{r \times r}$, $A_{12} \in \mathbb{R}^{r \times n-r}$, $A_{21} \times \mathbb{R}^{n-r \times r}$, $A_{22} \in \mathbb{R}^{n-r \times n-r}$, G is proper if and only if the block A_{22} is invertible.

Proof. Since U, V realizes a generalized state space transform we first have to update B and C. Define

$$UB = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \quad CV = \begin{bmatrix} C_1, & C_2 \end{bmatrix}$$

with $B_1 \in \mathbb{R}^{r\times m}$, $B_2 \in \mathbb{R}^{n-r\times m}$, $C_1 \in \mathbb{R}^{p\times r}$, $C_2 \in \mathbb{R}^{p\times n-r}$. System $(E; A, B, C, D)$ is assumed to have a C-controllable fast subsystem, so from Theorem 2.8 and the following remark it follows that

$$\operatorname{rank}\begin{bmatrix} E & B \end{bmatrix} = \operatorname{rank}\begin{bmatrix} T & 0 & B_1 \\ 0 & 0 & B_2 \end{bmatrix} = n$$

which means that the matrix B_2 must have full rank. By a similar argument from Theorem 2.9 and the corresponding remark

$$\operatorname{rank}\begin{bmatrix} E \\ C \end{bmatrix} = \operatorname{rank}\begin{bmatrix} T & 0 \\ 0 & 0 \\ C_1 & C_2 \end{bmatrix} = n$$

holds and hence C_2 has to be a full-rank matrix. Now we write the transfer function of our descriptor system in terms of the transformed matrices, that is

$$G(s) = \begin{bmatrix} C_1 & C_2 \end{bmatrix} \underbrace{\begin{bmatrix} sT - A_{11} & -A_{12} \\ -A_{21} & -A_{22} \end{bmatrix}^{-1}}_{:=K(s)} \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} + D.$$

Applying Lemma 3.1 to $K(s)$ yields the following result:

$$K(s) = \begin{bmatrix} Q(s) + Q(s)A_{12}S^{-1}(s)A_{21}Q(s) & Q(s)A_{12}S^{-1}(s) \\ S^{-1}(s)A_{21}Q(s) & S^{-1}(s) \end{bmatrix}$$

with $Q(s) = (sT - A_{11})^{-1}$ and the Schur complement $S(s) = -A_{22} - A_{21}Q(s)A_{12}$. Now we consider $\lim\limits_{s\to\infty} K(s)$. First of all we observe that $\lim\limits_{s\to\infty} Q(s) = 0$ because the matrix pencil $sT - A_{11}$ does not have any infinite eigenvalues.

Assume the matrix A_{22} is invertible. Then $\lim\limits_{s\to\infty} S^{-1}(s) = -A_{22}^{-1}$ holds and consequently

$$\begin{aligned} \lim_{s\to\infty} K(s) &= \begin{bmatrix} 0 & 0 \\ 0 & -A_{22}^{-1} \end{bmatrix} \text{ and thus} \\ \lim_{s\to\infty} \|G(s)\|_2 &= \left\|D - C_2A_{22}^{-1}B_2\right\|_2 < \infty \end{aligned} \tag{3.2}$$

which means that G is proper.

Let now A_{22} be a singular matrix. The expression $Q(s)$ can be expanded into a Laurent series at $s = \infty$ (see, e.g., [Sty06]) which yields

$$Q(s) = \sum_{i=-\infty}^{\infty} Q_i s^i$$

for constant coefficients Q_i. Since $\lim_{s\to\infty} Q(s) = 0$ the matrices Q_i for $i \geqslant 0$ have to be zero. In this way also $S(s)$ can be expanded into a Laurent series at the expansion point $s = \infty$, i.e.,

$$S(s) = \sum_{i=-\infty}^{0} S_i s^i.$$

Since A_{22} is assumed to be singular, $\lim_{s\to\infty} \lambda_{\min}(S(s)) = 0$, where $\lambda_{\min}$ denotes the smallest eigenvalue in magnitude. Hence $\lim_{s\to\infty} \left|\lambda_{\max}\left(S^{-1}(s)\right)\right| = \infty$ with the largest eigenvalue in magnitude $\lambda_{\max}$. So $S^{-1}(s)$ has a Laurent series representation at the expansion point $s = \infty$

$$S^{-1}(s) = \sum_{i=-\infty}^{\infty} \tilde{S}_i s^i$$

with degree larger or equal than 1. Consequently, the entries of $K(s)$ at the block positions (1, 1), (1, 2), and (2, 1) have a lower degree than the entry at block position (2, 2) because they contain $Q(s)$ as a factor. Because the matrices B_2 and C_2 have full rank the product $C_2 S^{-1}(s) B_2$ has the same degree as $S^{-1}(s)$. So we can write the transfer function G as

$$G(s) = H(s) + C_2 S^{-1}(s) B_2 + D,$$

where $H(s)$ contains only terms that have lower degree than S^{-1}. Since for $s \to \infty$ the maximum eigenvalue of S^{-1} tends to infinity in modulus, the maximum singular value of S^{-1} tends to infinity, thus also $\lim_{s\to\infty} \sigma_{\max}(G(s)) = \infty$ which means that G is improper. □

3.2 The Testing Routine

From the theorem above we see that we have to perform two major steps to test a transfer function for its properness:

(i) extract a subsystem of $(E; A, B, C, D)$ with both C-controllable and C-observable fast subsystem, i.e., remove all uncontrollable and unobservable infinite poles of the system;

(ii) test the matrix A_{22} from Theorem 3.1 for invertibility.

3.2.1 Removing Uncontrollable and Unobservable Infinite Poles

The method which we describe in this section is introduced in [Var90], algorithmic details can be found there. The two main steps of this method are as follows:

(i) Compute orthogonal matrices Q, $Z \in \mathbb{R}^{n\times n}$ and perform a generalized state space transform such that

$$\begin{aligned} &Q^T(A-\lambda E)Z = \begin{bmatrix} A_c^\infty - \lambda E_c^\infty & \star \\ 0 & A_{\bar{c}}^\infty - \lambda E_{\bar{c}}^\infty \end{bmatrix}, \quad Q^T B = \begin{bmatrix} B_c^\infty \\ 0 \end{bmatrix}, \\ &CZ = \begin{bmatrix} C_c^\infty & C_{\bar{c}}^\infty \end{bmatrix}, \end{aligned} \tag{3.3}$$

and the subsystem $(E_c^\infty; A_c^\infty, B_c^\infty, C_c^\infty, D)$ of order r does not contain any uncontrollable infinite poles (i.e., the fast subsystem of $(E_c^\infty; A_c^\infty, B_c^\infty, C_c^\infty, D)$ is C-controllable) and has the same transfer function as $(E; A, B, C, D)$.

(ii) Compute orthogonal matrices $\tilde{Q}$, $\tilde{Z} \in \mathbb{R}^{r\times r}$ and perform a generalized state space transform such that

$$\begin{aligned} &\tilde{Q}^T\left(A_c^\infty - \lambda E_c^\infty\right)\tilde{Z} = \begin{bmatrix} A_{co}^\infty - \lambda E_{co}^\infty & 0 \\ \star & A_{c\bar{o}}^\infty - \lambda E_{c\bar{o}}^\infty \end{bmatrix}, \quad \tilde{Q}^T B_c^\infty = \begin{bmatrix} B_{co}^\infty \\ B_{c\bar{o}}^\infty \end{bmatrix}, \\ &C_c^\infty \tilde{Z} = \begin{bmatrix} C_{co}^\infty & 0 \end{bmatrix}, \end{aligned}$$

and $(E_{co}^\infty; A_{co}^\infty, B_{co}^\infty, C_{co}^\infty, D)$ has a C-controllable and C-observable fast subsystem and the same transfer function as $(E; A, B, C, D)$.

The algorithm for the computation of the reduction (3.3) is based on the uncontrollable finite pole separation procedure (UFPSP, Algorithm 1 in [Var90]) which transforms the involved matrices to certain condensed forms. Actually the UFPSP computes a reduced system $(E_c; A_c, B_c, C_c, D)$ by orthogonal transformations with no *finite* uncontrollable poles. The participating matrices have the form

$$E_c = \begin{bmatrix} E_{11} & E_{12} & \cdots & \cdots & E_{1,k} \\ 0 & E_{22} & & & \vdots \\ \vdots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & E_{k,k} \end{bmatrix}, \quad A_c = \begin{bmatrix} A_{11} & A_{12} & \cdots & \cdots & A_{1,k} \\ A_{21} & A_{22} & & & \vdots \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & A_{k,k-1} & A_{k,k} \end{bmatrix},$$

$$B_c = \begin{bmatrix} A_{10} \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix},$$

where $E_{i,i}$, $A_{i,i} \in \mathbb{R}^{n_i \times n_i}$, $i = 1, \ldots, k$; $A_{i,i-1} \in \mathbb{R}^{n_i \times n_{i-1}}$ have ranks n_i; E_c is upper triangular and A_c is in upper block-Hessenberg form. For single-input systems, A_c is obtained in upper Hessenberg form. The reduced system $(E_c; A_c, B_c, C_c, D)$ has no finite uncontrollable poles since

$$\operatorname{rank}\begin{bmatrix} sE_c - A_c & B_c \end{bmatrix} = r \quad \forall s \in \mathbb{C}.$$

However, the subsystem $(E_c; A_c, B_c, C_c, D)$ may still contain infinite uncontrollable poles which we want to eliminate.

In order to remove the infinite uncontrollable poles of the original system we can also use the UFPSP, namely we have to apply it to the system $(A; E, B, C, D)$ which is equivalent to replacing λ by $\frac{1}{\lambda}$ in the system pencil. It can be seen from the form of the resulting matrices E_c^∞, A_c^∞, and B_c^∞ (now A_c^∞ is upper triangular, E_c^∞ is upper block-Hessenberg) that

$$\operatorname{rank}\begin{bmatrix} E_c^\infty - sA_c^\infty & B_c^\infty \end{bmatrix} = r \quad \forall s \in \mathbb{C}.$$

In particular, for $s = 0$, we obtain

$$\operatorname{rank}\begin{bmatrix} E_c^\infty & B_c^\infty \end{bmatrix} = r.$$

Hence the fast subsystem of the resulting system is C-controllable. Note that when we remove uncontrollable infinite poles using the UFPSP, the obtained system may still contain uncontrollable zero poles.

To remove the unobservable infinite poles of the system $(E_c^\infty; A_c^\infty, B_c^\infty, C_c^\infty, D)$ we apply the UFPSP to the dual system of $(A_c^\infty; E_c^\infty, B_c^\infty, C_c^\infty, D)$ which is $\left((A_c^\infty)^T; (E_c^\infty)^T, (C_c^\infty)^T, (B_c^\infty)^T, D^T\right)$. Algorithm 3.1 summarizes the main steps that have to be performed to remove all uncontrollable and unobservable infinite poles. The reason for permuting the rows and columns of the system matrices at Step 2 using the permutation transformation matrix P is to obtain the matrices $P\,(A_c^\infty)^T\,P$, and $P\,(E_c^\infty)^T\,P$ after Step 1 in upper triangular and upper block-Hessenberg form, respectively. As shown in [Var90], the UFPSP can be implemented in such a way that it can exploit efficiently the null elements structure of $P(A_c^\infty)^T P$.

It can be shown that Algorithm 3.1 can be implemented such that its computational costs are $\mathcal{O}\left(n^3\right)$, so it is reasonable for an efficient test for properness. Moreover uncontrollable and unobservable poles do not have any influence on the transfer function of a system. By calculating the subsystem $(E_{co}^\infty; A_{co}^\infty, B_{co}^\infty, C_{co}^\infty, D)$ we simultaneously reduce the order of the participating matrices in our algorithm for computing the $\mathcal{L}_\infty$-norm and hence also reduce its computational effort. The reduced system is also almost minimal because we only do not remove uncontrollable and unobservable zero poles. However, the $\mathcal{L}_\infty$-norm algorithm requires the matrix pencil $A - \lambda E$ to have no finite eigenvalues on the imaginary axis. If the matrix pencil has zero eigenvalues,

even if they are uncontrollable or unobservable, our implementation of the algorithm will return an infinite $\mathcal{L}_\infty$-norm. In this way we have to apply the UFPSP again in order to remove also finite uncontrollable or unobservable poles if we are not sure that $0 \notin \Lambda(A, E)$. Finally it should be mentioned that even if we do not test the transfer function for properness it might be useful to reduce the system order beforehand (if we have a large number of uncontrollable/unobservable poles).

Algorithm 3.1: Uncontrollable/Unobservable Infinite Pole Removal

Input: Descriptor system $(E; A, B, C, D)$.

Output: Subsystem $(E_{co}^\infty; A_{co}^\infty, B_{co}^\infty, C_{co}^\infty, D)$ whose fast subsystem is both C-controllable and C-observable.

1: Apply the UFPSP to obtain orthogonal matrices Q_1, Z_1 such that

$$Q_1^T(A - \lambda E)Z_1 = \begin{bmatrix} A_c^\infty - \lambda E_c^\infty & \star \\ 0 & A_{\bar{c}}^\infty - \lambda E_{\bar{c}}^\infty \end{bmatrix}, \quad Q_1^T B = \begin{bmatrix} B_c^\infty \\ 0 \end{bmatrix},$$
$$CZ_1 = \begin{bmatrix} C_c^\infty & C_{\bar{c}}^\infty \end{bmatrix},$$

and the fast subsystem of $(E_c^\infty; A_c^\infty, B_c^\infty, C_c^\infty, D)$ is C-controllable.

2: Compute the pertransposed system, i.e., set $E_c^\infty := PE_c^\infty P$, $A_c^\infty := PA_c^\infty P$, $B_c^\infty := PB_c^\infty$, $C_c^\infty := C_c^\infty P$ with the permutation matrix $P = \begin{bmatrix} & & 1 \\ & \iddots & \\ 1 & & \end{bmatrix}$.

3: Apply the UFPSP to obtain othogonal matrices Q_2, Z_2 such that

$$Q_2^T\left(A_c^\infty - \lambda E_c^\infty\right) Z_2 = \begin{bmatrix} A_{co}^\infty - \lambda E_{co}^\infty & 0 \\ \star & A_{c\bar{o}}^\infty - \lambda E_{c\bar{o}}^\infty \end{bmatrix}, \quad Q_2^T B_c^\infty = \begin{bmatrix} B_{co}^\infty \\ B_{c\bar{o}}^\infty \end{bmatrix},$$
$$C_c^\infty Z_2 = \begin{bmatrix} C_{co}^\infty & 0 \end{bmatrix},$$

and the fast subsystem of $(E_{co}^\infty; A_{co}^\infty, B_{co}^\infty, C_{co}^\infty, D)$ is both C-controllable and C-observable.

3.2.2 Testing Invertibility

In this section we describe a numerically reliable method for checking if the matrix A_{22} from Theorem 3.1 is invertible. For that purpose we have to determine the rank of certain matrices. This can be achieved via rank-revealing factorizations. In the sequel we state properties of these factorizations and summarize numerical methods for their computation.

Rank-Revealing QR Decomposition

The basis of our invertibility testing routine is formed by rank-revealing QR decompositions (RRQR decompositions) whose properties we briefly describe (see [BQO98b] for details). Let $A \in \mathbb{R}^{m\times n}$ be a given matrix (without loss of generality $m \geqslant n$) with singular values

$$\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_n \geqslant 0.$$

Definition 3.1 (Numerical Rank)**.** The *numerical rank* r of the matrix A with respect to a threshold τ is the number which satisfies

$$\frac{\sigma_1}{\sigma_r} \leqslant \tau < \frac{\sigma_1}{\sigma_{r+1}}.$$

Let A have a QR decomposition of the form

$$AP = Q\begin{bmatrix} R \\ 0 \end{bmatrix} = Q\begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \\ 0 & 0 \end{bmatrix} \tag{3.4}$$

where P is a permutation matrix, Q is an orthogonal matrix, R is upper triangular and R_{11} is of order r. Furthermore let $\kappa_2(A)$ denote the two-norm condition number of a matrix A.

Definition 3.2 (Rank-Revealing QR Decomposition)**.** Factorization (3.4) is said to be a *rank-revealing QR decomposition* of A if the following properties are fulfilled:

$$\kappa_2\left(R_{11}\right) \approx \frac{\sigma_1}{\sigma_r} \text{ and}$$
$$\left\|R_{22}\right\|_2 = \sigma_{\max}\left(R_{22}\right) \approx \sigma_{r+1}.$$

Whenever there is a well-determined gap between the singular values σ_r and σ_{r+1}, and hence the numerical rank of r is well defined, the RRQR decomposition (3.4) reveals the numerical rank of A by having a well-conditioned leading submatrix R_{11} and a trailing submatrix R_{22} of small norm.

Algorithms for computing RRQR factorizations all base on column pivoting in order to move columns of A with a large norm to the top. A good summary on that can be found, e.g., in [GVL96]. There the matrix Q is determined by a sequence of Householder matrices H which have the form

$$H = \begin{bmatrix} I & 0 \\ 0 & \tilde{H}(v) \end{bmatrix}, \quad \tilde{H}(v) = I - 2vv^T, \quad \|v\|_2 = 1. \tag{3.5}$$

For any given vector x we can choose a vector v such that $\tilde{H}(v)x = \alpha e_1$ where e_1 is the first canonical unit vector and $|\alpha| = \|x\|_2$ (see [GVL96] for computational

details). This can be used to successively annihilate certain elements of A to build the triangular matrix R. Assume for some k that we have computed Householder matrices $H_1, \ldots, H_{k-1}$ and permutation matrices $P_1, \ldots, P_{k-1}$ such that

$$(H_{k-1} \cdots H_1)\, A\, (P_1 \cdots P_{k-1}) = R^{(k-1)} = \begin{bmatrix} R_{11}^{(k-1)} & R_{12}^{(k-1)} \\ 0 & R_{22}^{(k-1)} \end{bmatrix},$$

where $R_{11}^{(k-1)} \in \mathbb{R}^{k-1 \times k-1}$ is a nonsingular and upper triangular matrix, and $R_{12}^{(k-1)} \in \mathbb{R}^{k-1 \times n-k+1}$, $R_{22}^{(k-1)} \in \mathbb{R}^{m-k+1 \times n-k+1}$. Now suppose that

$$R_{22}^{(k-1)} = \begin{bmatrix} z_k^{(k-1)} & \ldots & z_n^{(k-1)} \end{bmatrix}$$

is a column partitioning and let $p \geqslant k$ be the smallest index such that

$$\left\| z_p^{(k-1)} \right\|_2 = \max \left\{ \left\| z_k^{(k-1)} \right\|_2, \ldots, \left\| z_n^{(k-1)} \right\|_2 \right\}.$$

Note that if $k-1 = \operatorname{rank}(A)$, then this maximum is zero and we are finished. Otherwise let P_k be the $n \times n$ identity with columns p and k interchanged and determine a Householder matrix $H_k = \operatorname{diag}\left(I, \tilde{H}_k\right)$ such that if $R^{(k)} = H_k R^{(k-1)} P_k$ then $R^{(k)}(k+1:m, k) = 0$. In other words, P_k moves the largest column of $R_{22}^{(k-1)}$ to the leading position and $\tilde{H}_k$ zeros all of its subdiagonal elements. This method can still be refined, especially under the aspect of exploiting current computer architectures, for that we refer to [BQO98b, BQO98a].

Complete Orthogonal Decomposition

A decomposition which is related to the RRQR factorization is the complete orthogonal decomposition, see [GVL96].

Definition 3.3 (Complete Orthogonal Decomposition)**.** Let $A \in \mathbb{R}^{m \times n}$ (assume again $m \geqslant n$ without loss of generality) be a given matrix with rank r. A decomposition of the form

$$A = UTV^T = U \begin{bmatrix} T_{11} & 0 \\ 0 & 0 \end{bmatrix} V^T \tag{3.6}$$

with orthogonal matrices $U \in \mathbb{R}^{m \times m}$, $V \in \mathbb{R}^{n \times n}$ and a full-rank matrix $T_{11} \in \mathbb{R}^{r \times r}$ is called *complete orthogonal or UTV decomposition* of A. If T_{11} in (3.6) is upper or lower triangular the factorization is also called *URV or ULV decomposition*, respectively.

URV decompositions can be easily computed if one knows an RRQR factorization of A of the form (3.4). Since the matrix R_{22} has a very small norm, it can be neglected and hence we arrive at

$$Q^T A P = \begin{bmatrix} R_{11} & R_{12} \\ 0 & 0 \end{bmatrix}.$$

Then the rows of R_{12} can be successively annihilated by applying an appropriate sequence of Householder matrices of the form (3.5) to the matrix $\begin{bmatrix} R_{11} & R_{12} \end{bmatrix}$ from the right. Thereby the k-th Householder matrix eliminates the $(r-k+1)$-th row of R_{12}.

The Overall Process

With the rank-revealing factorizations from above we can now easily formulate a method which tests the matrix A_{22} from Theorem 3.1 for invertibility. This is summarized in Algorithm 3.2.

Algorithm 3.2: Invertibility Testing Procedure

Input: Descriptor system $(E; A, B, C, D)$ with C-controllable and C-observable fast subsystem and transfer function G, tolerance τ

Output: Is matrix A_{22} from Theorem 3.1 invertible?

1: Compute an URV decomposition of E with respect to the tolerance τ in the RRQR factorization, i.e., find orthogonal matrices U, $V \in \mathbb{R}^{n\times n}$ such that

$$\tilde{E} = U^T E V = \begin{bmatrix} T & 0 \\ 0 & 0 \end{bmatrix}$$

and $T \in \mathbb{R}^{r\times r}$ has full rank.

2: Apply U and V to the matrix A such that

$$\tilde{A} = U^T A V = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

has the same block partitioning as $\tilde{E}$.

3: Compute an RRQR factorization of $A_{22} \in \mathbb{R}^{k\times k}$ with respect to τ, i.e., find an orthogonal matrix Q and a permutation matrix P such that

$$A_{22} P = QR = \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix},$$

and $R_{11} \in \mathbb{R}^{s\times s}$ is well conditioned and for $R_{22} \in \mathbb{R}^{k-s\times k-s}$ the condition $\|R_{22}\|_2 \approx \sigma_{s+1} < \frac{1}{\tau}\|R\|_2 \leqslant \sigma_s$ is satisfied.

4: **if** $k = s$ **then**

5: G is proper.

6: **else**
7: G is improper.
8: **end if**

Note, that we could also have used singular value decompositions to determine the rank of the participating matrices. However, RRQR factorizations are cheaper to compute. That is the reason for using these instead of SVDs.

4 An Algorithm for Computing the $\mathcal{L}_\infty$-Norm

Until 1988 not much attention has been paid to the computation of the $\mathcal{L}_\infty$-norm of a dynamical system. The "computation" was done by a search over frequencies. The disadvantages of this approach are obvious: it cannot be used automatically within other algorithms, it takes a considerable amount of computer time, and no accuracy bound can be given [BS90]. One of the first contributions on that topic was given by Byers in [Bye88]. There he used the connection between the singular values of a certain matrix-valued function and the eigenvalues of an associated Hamiltonian matrix to measure the distance between a stable matrix and the unstable matrices. Then in 1989 Boyd, Balakrishnan, and Kabamba generalized Byers' approach to transfer functions of standard continuous linear time-invariant systems in [BBK89] and derived a bisection method for computing the $\mathcal{L}_\infty$-norm with guaranteed accuracy. This bisection algorithm is much more efficient than the search over frequencies, but for repeated use as well as for large systems, it is still not very fast. In 1990 two papers from Bruisma and Steinbuch [BS90], and Boyd and Balakrishnan [BB90] have been published which propose a much faster algorithm which converges locally quadratically and ensures a given error bound. In recent publications there were made some advances in different other directions. In this context, the work of Lawrence, Tits, and Van Dooren [LTVD00] on the computing an upper bound for the μ-norm is mentionable. The μ-norm is closely related to the $\mathcal{L}_\infty$-norm and the computational methods proposed in [LTVD00] are as well based on [BB90]. In 1998 Genin, Van Dooren and Vermaut [GVDV98] found a new iterative method for the computation of the $\mathcal{H}_\infty$-norm based on cubic interpolation which yields better convergence properties than the earlier algorithms. Just a few years ago Chahlaoui, Gallivan, and Van Dooren published articles for the estimation of the $\mathcal{H}_\infty$-norm for large sparse standard discrete-time transfer functions [CGVD04, CGVD07]. By the author's best knowledge there is still no way known to extend this method to descriptor systems.

In this thesis we extend the algorithm from [BS90, BB90] to be able to compute the $\mathcal{L}_\infty$-norm of transfer functions of descriptor systems. We also show how one can treat the arising problems during the algorithm in a proper way.

4.1 Preliminaries

First of all we state and prove some theoretical results which are essential for extending the method from [BS90, BB90] to descriptor systems. Here we focus on

continuous-time systems of the form (2.6). Furthermore we assume that the system's transfer function G is proper. This property can be tested by the algorithm introduced in the previous chapter. The computation of the $\mathcal{L}_\infty$-norm is connected to the computation of the eigenvalues of specific skew-Hamiltonian/Hamiltonian matrix pencils $M_\gamma - \lambda N$ with

$$M_\gamma = \begin{bmatrix} A & 0 \\ 0 & -A^T \end{bmatrix} + \begin{bmatrix} B & 0 \\ 0 & -C^T \end{bmatrix} \begin{bmatrix} -D & \gamma I \\ \gamma I & -D^T \end{bmatrix}^{-1} \begin{bmatrix} C & 0 \\ 0 & B^T \end{bmatrix}, \quad N = \begin{bmatrix} E & 0 \\ 0 & E^T \end{bmatrix}. \tag{4.1}$$

The matrix M_γ can also be expressed as

$$M_\gamma = \begin{bmatrix} A - BR^{-1}D^TC & -\gamma BR^{-1}B^T \\ \gamma C^TS^{-1}C & -A^T + C^TDR^{-1}B^T \end{bmatrix}$$

with the matrices $R = D^TD - \gamma^2 I$, and $S = DD^T - \gamma^2 I$. Within the next section we will consider these matrix structures in more detail. Using this we can formulate a theorem which connects the singular values of a transfer function with the eigenvalues of the associated matrix pencil (4.1) as it is done for standard systems in [BBK89].

Theorem 4.1. *Assume the matrix pencil $A - \lambda E$ is regular and has no finite eigenvalues on the imaginary axis, $\gamma > 0$ is not a singular value of D and $\omega_0 \in \mathbb{R}$. Then, γ is a singular value of $G(\mathrm{i}\omega_0)$ if and only if $M_\gamma - \mathrm{i}\omega_0 N$ is singular.*

Proof. The argumentation follows the one of the proof of Theorem 1 in [BBK89]. Let γ be a singular value of $G(\mathrm{i}\omega_0)$. Then there exist nonzero vectors $u \in \mathbb{C}^m$, $v \in \mathbb{C}^p$ such that

$$\begin{aligned} G(\mathrm{i}\omega_0)u &= \gamma v, \\ G(\mathrm{i}\omega_0)^H v &= \gamma u, \end{aligned}$$

so that

$$\begin{aligned} (C(\mathrm{i}\omega_0 E - A)^{-1}B + D)u &= \gamma v, \\ (B^T(-\mathrm{i}\omega_0 E^T - A^T)^{-1}C^T + D^T)v &= \gamma u. \end{aligned} \tag{4.2}$$

Define

$$\begin{aligned} r &= (\mathrm{i}\omega_0 E - A)^{-1}Bu, \\ s &= (-\mathrm{i}\omega_0 E^T - A^T)^{-1}C^T v. \end{aligned} \tag{4.3}$$

Now solving for u and v in terms of r and s yields

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -D & \gamma I \\ \gamma I & -D^T \end{bmatrix}^{-1} \begin{bmatrix} C & 0 \\ 0 & B^T \end{bmatrix} \begin{bmatrix} r \\ s \end{bmatrix}. \tag{4.4}$$

Note, that (4.4) guarantees that

$$\begin{bmatrix} r \\ s \end{bmatrix} \neq \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

From (4.3) we get

$$\begin{aligned} (\mathrm{i}\omega_0 E - A)r &= Bu, \\ \left(-\mathrm{i}\omega_0 E^T - A^T\right) s &= C^T v. \end{aligned} \tag{4.5}$$

From (4.5) we obtain

$$\left(\begin{bmatrix} \mathrm{i}\omega_0 E & 0 \\ 0 & \mathrm{i}\omega_0 E^T \end{bmatrix} - \begin{bmatrix} A & 0 \\ 0 & -A^T \end{bmatrix} \right) \begin{bmatrix} r \\ s \end{bmatrix} = \begin{bmatrix} B & 0 \\ 0 & -C^T \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix},$$

which, with (4.4) is equivalent to

$$\left(\begin{bmatrix} A & 0 \\ 0 & -A^T \end{bmatrix} + \begin{bmatrix} B & 0 \\ 0 & -C^T \end{bmatrix} \begin{bmatrix} -D & \gamma I \\ \gamma I & -D^T \end{bmatrix}^{-1} \begin{bmatrix} C & 0 \\ 0 & B^T \end{bmatrix} \right) \begin{bmatrix} r \\ s \end{bmatrix} = \mathrm{i}\omega_0 \begin{bmatrix} E & 0 \\ 0 & E^T \end{bmatrix} \begin{bmatrix} r \\ s \end{bmatrix}. \tag{4.6}$$

Thus

$$M_\gamma \begin{bmatrix} r \\ s \end{bmatrix} = \mathrm{i}\omega_0 N \begin{bmatrix} r \\ s \end{bmatrix}.$$

This proves one direction of Theorem 4.1.
Now we prove the converse. Suppose that the matrix pencil $M_\gamma - \lambda N$ has the eigenvalue $\mathrm{i}\omega_0$, that is, (4.6) holds for some $\begin{bmatrix} r \\ s \end{bmatrix} \neq \begin{bmatrix} 0 \\ 0 \end{bmatrix}$. Defining u and v by equation (4.4), clearly yields $\begin{bmatrix} u \\ v \end{bmatrix} \neq 0$. (Otherwise $\begin{bmatrix} r \\ s \end{bmatrix}$ would be zero, following from (4.3).) Then from (4.4) and (4.6), we conclude (4.2), which establishes that γ is a singular value of $G(\mathrm{i}\omega_0)$. □

In contrast to Theorem 2 of [BBK89] we cannot ensure $\lim_{\omega\to\infty} \sigma_{\max}(G(\mathrm{i}\omega)) = \sigma_{\max}(D)$ for descriptor systems with singular descriptor matrix E anymore, even if G is proper.

Example 4.1. Consider a continuous-time descriptor system $(E; A, B, C, D)$ with

$$E = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad A = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 2 & -3 \end{bmatrix}, \quad D = 0.$$

Now we have

$$G(s) = \frac{2}{s-2} + 1$$

and thus

$$\lim_{\omega\to\infty} \sigma_{\max}(G(\mathrm{i}\omega)) = 1 \neq 0.$$

However we can state the following simple result.

Lemma 4.1. *If E is nonsingular then* $\lim_{\omega\to\infty} \sigma_{\max}(G(\mathrm{i}\omega)) = \sigma_{\max}(D)$.

Proof. A descriptor system of the form (2.6) with nonsingular descriptor matrix E is restricted system equivalent to the standard system

$$\begin{aligned} \dot{x}(t) &= E^{-1}Ax(t) + E^{-1}Bu(t), \\ y(t) &= Cx(t) + Du(t) \end{aligned}$$

with transfer function $G(s) = C(sI - E^{-1}A)^{-1}E^{-1}B + D$. Obviously $\lim_{\omega\to\infty} G(\mathrm{i}\omega) = D$ holds which proves the assertion. □

The value of $\lim_{\omega\to\infty} \sigma_{\max}(G(\mathrm{i}\omega))$ is discussed in Section 4.3. We state a modified version of Theorem 2 from [BBK89].

Theorem 4.2. *Let $\gamma > \min_{\omega\in\mathbb{R}} \sigma_{\max}(G(\mathrm{i}\omega))$ be not a singular value of D. Then $\|G\|_{\mathcal{L}_\infty} \geqslant \gamma$ if and only if $M_\gamma - \lambda N$ has imaginary eigenvalues (i.e., at least one).*

Proof. Assume first $\|G\|_{\mathcal{L}_\infty} \geqslant \gamma$. From the definition of the $\mathcal{L}_\infty$-norm and the continuity of $\sigma_{\max}(G(\mathrm{i}\omega))$ it follows that there exists $\omega_0 \in \mathbb{R}$ such that $\sigma_{\max}(G(\mathrm{i}\omega_0)) = \gamma$. Together with Theorem 4.1 we obtain that $M_\gamma - \mathrm{i}\omega_0 N$ is singular, so the matrix pencil $M_\gamma - \lambda N$ has at least one purely imaginary eigenvalue.

If we assume on the other hand that $M_\gamma - \lambda N$ has purely imaginary eigenvalues, e.g., $\mathrm{i}\omega_0$, Theorem 4.1 yields that γ is a singular value of $(G(\mathrm{i}\omega_0))$, hence $\|G\|_{\mathcal{L}_\infty} \geqslant \gamma$. □

It should also be mentioned that in some pathologic cases, the matrix pencil $M_\gamma - \lambda N$ is singular.

Example 4.2. Consider a descriptor system $(E; A, B, C, D)$ with singular E. Furthermore choose

$$A = \frac{1}{\gamma} I_n, \quad B = C = I_n, \quad D = 0.$$

By some simple calculations we obtain

$$M_\gamma = \frac{1}{\gamma} \begin{bmatrix} I_n & I_n \\ -I_n & -I_n \end{bmatrix}$$

which has zero as only eigenvalue. Since E is singular, also N is singular and hence the matrix pencil $M_\gamma - \lambda N$ is singular.

However, if $M_\gamma - \lambda N$ is singular, every complex number is in its spectrum. In particular, $\mathrm{i}\mathbb{R} \subset \Lambda(M_\gamma, N)$. In Section 4.3 we evaluate the transfer function at certain frequencies to obtain an initial bound for Algorithm 4.2. But the singularity of

$M_\gamma - \lambda N$ yields that $\sigma_{\max}(G(\mathrm{i}\omega)) \geqslant \gamma$ for every value of ω. Furthermore, before we compute the spectrum of a skew-Hamiltonian/Hamiltonian matrix pencil the first time, the computed initial value is slightly increased. Thus we can never encounter singular matrix pencils in our computations.

4.2 The Algorithm and its Properties

4.2.1 Basic Iteration and Graphical Interpretation

Using Theorems 4.1 and 4.2 we can state the basic iteration of Algorithm 4.1 for computing the $\mathcal{L}_\infty$-norm (see [BS90]).

Algorithm 4.1: Basic Iteration for Computing the $\mathcal{L}_\infty$-Norm

Input: Continuous linear time-invariant descriptor system $(E; A, B, C, D)$ with transfer function G.

Output: $\|G\|_{\mathcal{L}_\infty}$.

1: Compute an initial value $\gamma_{lb} < \|G\|_{\mathcal{L}_\infty}$.
2: **for** $i = 1, 2, \ldots$ **do**
3: Form the matrix pencil $M_{\gamma_{lb}} - \lambda N$ and compute its eigenvalues.
4: Set $\{\mathrm{i}\omega_1, \ldots, \mathrm{i}\omega_k\}$ = finite imaginary eigenvalues with $\omega_i \geqslant 0$ for $i = 1, \ldots, k$.
5: Set $m_j = \frac{1}{2}(\omega_j + \omega_{j+1}),\ j = 1, \ldots, k-1$.
6: Compute the largest singular value of $G(\mathrm{i}m_j),\ j = 1, \ldots, k-1$.
7: Set $\gamma_{lb} = \max\limits_{1 \leqslant j \leqslant k-1} (\sigma_{\max}(G(\mathrm{i}m_j)))$.
8: **end for**

A graphical interpretation of Algorithm 4.1 is given by Figure 4.1. There the two curves illustrate the singular values of $G(\mathrm{i}\omega)$ for an interval of values ω. If we have given an iterate $\gamma := \gamma_{lb}(i)$ at iteration i we first compute all imaginary eigenvalues $\{\mathrm{i}\omega_1, \ldots, \mathrm{i}\omega_k\}$ of the matrix pencil $M_\gamma - \lambda N$ (the ω_i corresponding to squares). Then we choose the midpoints m_j of each of the intervals (ω_j, ω_{j+1}) for $j = 1, \ldots, k-1$ and compute the largest singular value of each of the matrices $G(\mathrm{i}m_j)$ (the singular values associated to triangles). In this way we obtain our new iterate $\gamma_{lb}(i+1)$ as the maximum singular value of all the matrices $G(\mathrm{i}m_j)$.

4.2.2 Convergence Properties

We briefly want to summarize the convergence properties of this algorithm which have already been investigated in [BBK89, BS90].

Theorem 4.3. *Define* $\gamma := \gamma_{lb}(i)$ *and* $V(i) = \max\limits_{0 \leqslant j \leqslant k-1} (\omega_{j+1} - \omega_j)$ *at iteration* i.

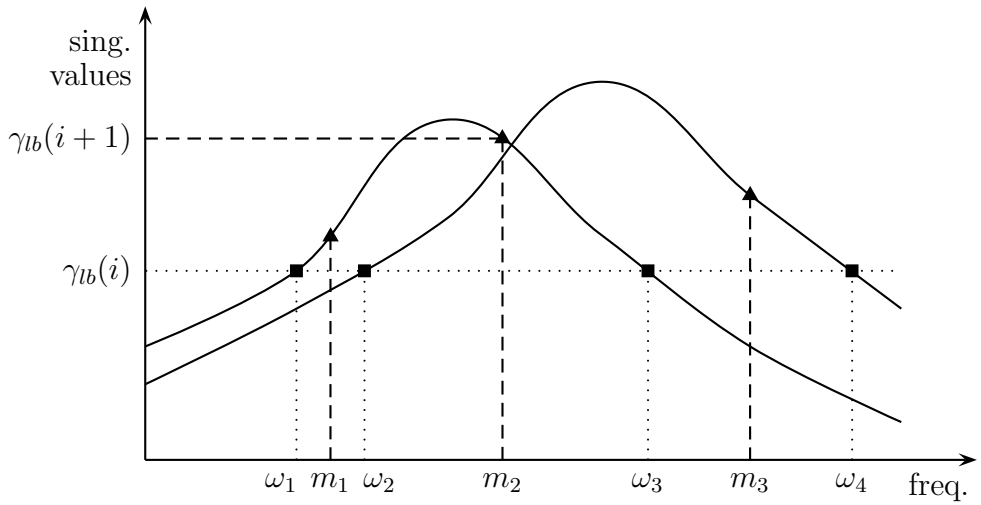


Figure 4.1: Graphical interpretation of Algorithm 4.1

Then we have

$$V(i+1) \leqslant \frac{1}{2} V(i).$$

Proof. Let $I_1^{(i)}, \ldots, I_l^{(i)}$ denote the frequency intervals (ω_j, ω_{j+1}) in which $\sigma_{\max}(G(\mathrm{i}\omega)) > \gamma$ at iteration i of Algorithm 4.1, so that

$$V(i) = \max_{1 \leqslant k \leqslant l} \operatorname{length}\left(I_k^{(i)}\right).$$

Each interval $I_k^{(i+1)}$ is contained in one of the intervals $I_1^{(i)}, \ldots, I_l^{(i)}$; moreover each interval $I_k^{(i+1)}$ cannot contain any of the midpoints of the intervals $I_k^{(i)}$ since at these frequencies we have

$$\sigma_{\max}(G(\mathrm{i}\omega)) \leqslant \gamma_{lb}(i+1),$$

whereas in the intervals $I_k^{(i+1)}$ we have

$$\sigma_{\max}(G(\mathrm{i}\omega)) > \gamma_{lb}(i+1).$$

Thus each interval at iteration $i+1$ is contained in either the left or right half of an interval from iteration i. Now the theorem follows immediately. □

Theorem 4.4 (Global Monotonic Convergence)**.** *It holds that* $\gamma_{lb}(i) \longrightarrow \|G\|_{\mathcal{L}_\infty}$ *for* $i \longrightarrow \infty$*. The convergence is global and monotonic.*

Proof. Since there are at most $\frac{1}{2}n$ intervals of the form $I_j^{(i)}$ at each iteration, it follows from Theorem 4.3 that the total length of the intervals converges to zero. The convergence of γ_{lb} to $\|G\|_{\mathcal{L}_\infty}$ follows from the uniform continuity of $\sigma_{\max}(G(\mathrm{i}\omega))$. Monotonicity and globality of the convergence follow obviously from Theorem 4.3. □

In [BBK89] also a proof of quadratic convergence is given. This proof is based on the local behavior of $\sigma_{\max}(G(\mathrm{i}\omega))$ near a local maximum.

Lemma 4.2. *Suppose $\sigma_{\max}(G(\mathrm{i}\omega))$ has a local maximum at ω_M. Then near ω_M we have*

$$\begin{aligned}\sigma_{\max}\left(G(\mathrm{i}\omega)\right) &= \sigma_{\max}\left(G\left(\mathrm{i}\omega_M\right)\right) - a(\omega-\omega_M)^{2N} \\ &\quad + \begin{cases} b_+\left(\omega-\omega_M\right)^{2N+1}, & \omega \geqslant \omega_M, \\ b_-\left(\omega-\omega_M\right)^{2N+1}, & \omega < \omega_M \end{cases} \\ &\quad + o\left(\left(\omega-\omega_M\right)^{2N+1}\right) \end{aligned} \tag{4.7}$$

for some $N \geqslant 1$, $a > 0$, and $b_- \leqslant b_+$.

Proof. See [BBK89]. □

Remark 4.1. Since $N \geqslant 1$ it follows that $\sigma_{\max}\left(G(\mathrm{i}\omega)\right)$ is at least twice continuously differentiable near a local maximum.

Theorem 4.5 (Local Quadratic Convergence)**.** *Let $\Omega_{\max} = \{\omega_1, \ldots, \omega_r\}$ be the set of maximizing frequencies (i.e., $\sigma_{\max}\left(G(\mathrm{i}\omega_j)\right) = \|G\|_{\mathcal{L}_\infty}$ for $j = 1, \ldots, r$). Let N_j, a_j, b_{+j}, b_{-j} be the constants in the local representation of $\sigma_{\max}\left(G(\mathrm{i}\omega)\right)$ near ω_j given by equation (4.7), for $j = 1, \ldots, r$. Then it holds*

$$\lim_{t\to\infty} \frac{\|G\|_{\mathcal{L}_\infty} - \gamma_{lb}(i+1)}{\left(\|G\|_{\mathcal{L}_\infty} - \gamma_{lb}(i)\right)^2} = \min_j \frac{1}{a_j}\left(\frac{b_{+j}+b_{-j}}{4a_jN_j}\right)^{2N_j},$$

i.e., Algorithm 4.1 is quadratically convergent.

Proof. See [BBK89]. □

4.2.3 Stopping Criterion and Relative Error

Algorithm 4.1 does still not contain a suitable stopping criterion. A possible stopping criterion can be introduced as follows. Let ε (e.g., the machine precision) be a predefined relative tolerance. Then the iteration can be aborted when

$$\sigma_{\max}\left(G(\mathrm{i}\omega)\right) = \gamma_{lb}(1+2\varepsilon) \tag{4.8}$$

has no solutions (see [BBK89]). Implementing this stopping criterion directly would require an additional computation of the eigenvalues of a skew-Hamiltonian/Hamiltonian matrix pencil. It is more efficient to incorporate this directly into the algorithm as in Algorithm 4.2.

Algorithm 4.2: Two-Step Algorithm for Computing the $\mathcal{L}_\infty$-Norm

Input: Continuous linear time-invariant descriptor system $(E; A, B, C, D)$ with transfer function G.

Output: $\|G\|_{\mathcal{L}_\infty}$.

1: Compute an initial value $\gamma_{lb} < \|G\|_{\mathcal{L}_\infty}$.
2: **repeat**
3: Set $\gamma := (1 + 2\varepsilon)\gamma_{lb}$.
4: Form the matrix pencil $M_\gamma - \lambda N$ and compute its eigenvalues.
5: **if** no imaginary eigenvalues **then**
6: $\gamma_{ub} = \gamma$, break.
7: **else**
8: Set $\{\mathrm{i}\omega_1, \ldots, \mathrm{i}\omega_k\}$ = finite imaginary eigenvalues with $\omega_i \geqslant 0$ for $i = 1, \ldots, k$.
9: Set $m_j = \frac{1}{2}(\omega_j + \omega_{j+1})$, $j = 1, \ldots, k-1$.
10: Compute the largest singular value of $G(\mathrm{i}m_j)$, $j = 1, \ldots, k-1$.
11: Set $\gamma_{lb} = \max\limits_{1 \leqslant j \leqslant k-1} (\sigma_{\max}(G(\mathrm{i}m_j)))$.
12: **end if**
13: **until** break
14: Set $\|G\|_{\mathcal{L}_\infty} = \frac{1}{2}(\gamma_{lb} + \gamma_{ub})$.

Theorem 4.6 (Relative Error)**.** *The stopping criterion* (4.8) *ensures a relative error of at most* ε*.*

Proof. As (4.8) is assumed not to have a solution, the inequality

$$\gamma_{lb} \leqslant \|G\|_{\mathcal{L}_\infty} \leqslant (1 + 2\varepsilon)\gamma_{lb}$$

for the exact norm $\|G\|_{\mathcal{L}_\infty}$ holds. The computed norm Γ yields

$$\Gamma = \frac{1}{2}(\gamma_{lb} + \gamma_{ub}) = \frac{1}{2}(\gamma_{lb} + (1 + 2\varepsilon)\gamma_{lb}) = (1 + \varepsilon)\gamma_{lb}.$$

This leads to

$$\left| \|G\|_{\mathcal{L}_\infty} - \Gamma \right| \leqslant \varepsilon\gamma_{lb} \leqslant \varepsilon \|G\|_{\mathcal{L}_\infty}$$

which proves the assertion. □

4.2.4 Further Remarks

We still want to briefly remark that the computation time is affected by the number of frequency points ω_j in each step (see [BS90]). The more frequency points we have, the more often the largest singular values of $G(\mathrm{i}\omega_j)$ have to be computed. Since the

number of frequencies ω_j decreases during the algorithm, the first step generally takes more time than the last one. This also shows the importance of a good initial value γ_{lb} since for those values which are "near" $\|G\|_{\mathcal{L}_\infty}$, the number of matrices $G(\mathrm{i}\omega)$ with singular values γ_{lb} will be "low".

Moreover care must be taken of the accuracy of the eigenvalue computation. Inaccuracy could cause the algorithm to fail (see [BS90]). In Section 4.4 we show how we can improve the accuracy of the eigenvalue computation by extending the matrix pencil $M_\gamma - \lambda N$ to a skew-Hamiltonian/Hamiltonian matrix pencil of larger size. Finally, in Chapter 5 we introduce a structure-preserving method which exploits the skew-Hamiltonian/Hamiltonian structure of the involved matrix pencils and computes especially purely imaginary eigenvalues very well. In this way better results for the computed $\mathcal{L}_\infty$-norms could be achieved as well.

4.3 Choice of the Initial Lower Bound

In Algorithm 4.2 we only mentioned that we have to choose an appropriate initial value $\gamma_{lb} < \|G\|_{\mathcal{L}_\infty}$. This section deals with the question how we choose this value.

4.3.1 Choice of Initial Test Frequencies

First of all, it can be observed that many continuous-time systems take their $\mathcal{L}_\infty$-norm at the frequency points $\omega = 0$ or $\omega = \infty$ (see [BS90]). These systems are called low and high pass filters, respectively. Low pass filters achieve the highest signal amplification for signals with low frequencies, whereas high pass filters best amplify signals with very high frequency. It is an important aspect to know the largest singular values of G at the boundary of the frequency interval $(0, \infty)$ since our algorithm converges to a *local* maximum. Assume for instance that the $\mathcal{L}_\infty$-norm of a transfer function is attained at $\omega = \infty$. Assume further that we have an iterate γ_i with $\hat{\gamma} < \gamma_i < \|G\|_{\mathcal{L}_\infty}$ where $\hat{\gamma}$ is the largest local maximum of $\sigma_{\max}(G(\mathrm{i}\omega))$ (if there is none, take $\hat{\gamma} = \sigma_{\max}(G(0))$). Then one can easily see that the corresponding skew-Hamiltonian/Hamiltonian matrix pencil $M_\gamma - \lambda N$ has exactly one purely imaginary eigenvalue $\mathrm{i}\omega_0$ with $\omega_0 \geqslant 0$. So it is not possible anymore to set up the midpoint of a frequency interval as in Algorithm 4.2 to force the convergence of the iteration.

Example 4.3. Consider the descriptor system $(E; A, B, C, D)$ with

$$E = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -3 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \; A = \begin{bmatrix} -3 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \; B = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \; C = \begin{bmatrix} 0 \\ 3.3 \\ 1 \\ -1 \end{bmatrix}^T, \; D = 1.$$

The singular value plot of the corresponding transfer function is shown in Figure 4.2. As one can see the situation described above occurs, e.g., for $\gamma_i = 1.9$.

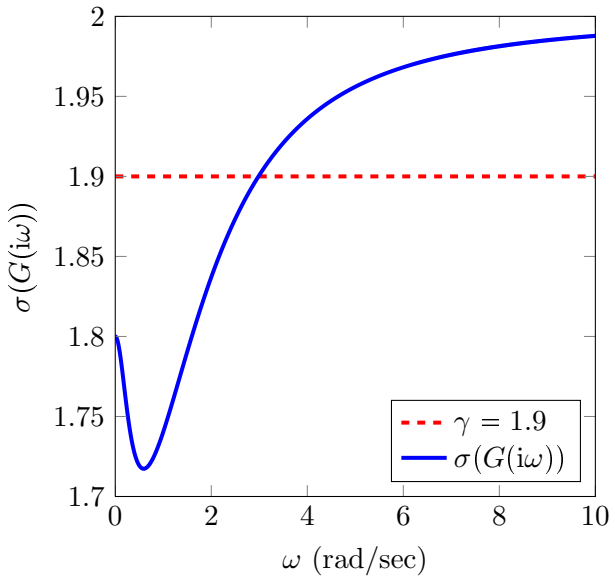


Figure 4.2: Singular value plot of $G(\mathrm{i}\omega) = C(\mathrm{i}\omega E - A)^{-1}B + D$

Additionally the transfer function is evaluated at certain test frequencies. The choice of these frequencies is rather heuristic but there are several methods which yield good results when computing the $\mathcal{L}_\infty$-norm for standard systems.

The original method (see [BS90]) proposes to compute $\sigma_{\max}(G(\mathrm{i}\omega_p))$ where ω_p depends on the poles of the transfer function. We choose $\omega_p = |\lambda_i|$ with a pole λ_i of G where λ_i is selected to

$$\text{maximize} \quad \left| \frac{\mathrm{Im}(\lambda_i)}{\mathrm{Re}(\lambda_i)} \frac{1}{|\lambda_i|} \right|$$

if G has poles with $\mathrm{Im}(\lambda_i) \neq 0$, or to

$$\text{minimize} \quad |\lambda_i|$$

if G has only real poles. The interpretation of the choice of ω_p is as follows. The corresponding pole λ_i is "dominant" in some sense. It is chosen in a way such that it is close to the imaginary axis compared to its magnitude and that it is not too large. Hence ω_p is still in a reasonable frequency range and $\mathrm{i}\omega_p$ is quite near to λ_i. In this way we hope that the pole λ_i still has a sufficiently strong impact on the value of the transfer function at $\mathrm{i}\omega_p$ and that $\sigma_{\max}(G(s))$ is near a peak in the singular value plot at $s = \mathrm{i}\omega_p$.

The second method (see [Sim06]) determines

$$\omega_p = \arg\max \sigma_{\max}(G(\mathrm{i}\omega_i)) \tag{4.9}$$

where

$$\begin{aligned}\omega_i &= \sqrt{\max\left\{\frac{1}{4}|\lambda_i|^2, \mathrm{Im}(\lambda_i)^2 - \mathrm{Re}(\lambda_i)^2\right\}} \\ &= |\lambda_i|\sqrt{\max\left\{\frac{1}{4}, 1-2r^2\right\}} \quad \text{with } r := \frac{\mathrm{Re}(\lambda_i)}{|\lambda_i|}\end{aligned} \tag{4.10}$$

for all poles λ_i with $\mathrm{Im}(\lambda_i) > 0$. This method has quite often been experienced to search initial values in larger domain than the choice of $\omega_i = |\lambda_i|$. However, this is still just a heuristic and there are of course counter examples. Note that this approach needs as many computations of the maximum singular value of the transfer function as $A - \lambda E$ has finite eigenvalues with positive imaginary part, compared to just one evaluation in the first method. But if $p, m \ll n$, the matrices whose maximum singular values should be computed are small and hence the computation is cheap compared to one eigenvalue computation of a skew-Hamiltonian/Hamiltonian matrix pencil. In this way it is not too expensive to evaluate the transfer function at many test frequencies. However, if we have a lot of inputs and outputs, the first method might lead to a better performance.

Summarizing, we obtain

$$\gamma_{lb} = \max\left\{\sigma_{\max}\left(G\left(0\right)\right), \sigma_{\max}\left(G\left(\mathrm{i}\omega_p\right)\right), \sigma_{\max}\left(G\left(\infty\right)\right)\right\}.$$

The computation of $\sigma_{\max}\left(G\left(0\right)\right)$ and $\sigma_{\max}\left(G\left(\mathrm{i}\omega_p\right)\right)$ is rather simple but evaluating $\sigma_{\max}\left(G\left(\infty\right)\right)$ is a more difficult task, since $\lim_{\omega\to\infty} \sigma_{\max}(G(\mathrm{i}\omega)) = \sigma_{\max}(D)$ does not generally hold for descriptor systems, see also Example 4.1. We propose a method for computing $G(\infty)$ in the next subsection.

4.3.2 Computation of $\sigma_{\max}\left(G(\infty)\right)$

In this subsection we show how one can additively decompose a proper transfer function

$$G(s) = G_{sp}(s) + P(s) \tag{4.11}$$

with a strictly proper part G_{sp} and a polynomial part P. Such a decomposition always exists (see [Sty06]). Since G_{sp} is strictly proper we have $\lim_{\omega\to\infty} G_{sp}(\mathrm{i}\omega) = 0$ and thus we only have to care about the polynomial part $P(s)$. As we assume that G is a proper transfer function, P has to be a constant matrix-valued polynomial. This yields

$$\lim_{\omega\to\infty} G(\mathrm{i}\omega) = P(s_0)$$

for arbitrary $s_0 \in \mathbb{C}$.

Block-Triangularization of the System Pencil

To obtain the decomposition (4.11) we have to decouple the system such that finite and infinite poles of G are separated. This can, e.g., be done as follows.

First we transform the matrix pencil $A - \lambda E$ to generalized Schur form, i.e., we apply the QZ algorithm (see [GVL96]) with a subsequent eigenvalue reordering to find othogonal matrices Q_1, $Q_2 \in \mathbb{R}^{n\times n}$ such that

$$E_1 := Q_1^T E Q_2 = \begin{bmatrix} E_f & W_E \\ 0 & E_\infty \end{bmatrix}, \quad A_1 := Q_1^T A Q_2 = \begin{bmatrix} A_f & W_A \\ 0 & A_\infty \end{bmatrix}, \tag{4.12}$$

where E_1 is upper triangular, A_1 is upper quasi triangular, $E_f \in \mathbb{R}^{n_f\times n_f}$, $A_\infty \in \mathbb{R}^{n_\infty\times n_\infty}$ are nonsingular and $E_\infty \in \mathbb{R}^{n_\infty\times n_\infty}$ has index of nilpotency ν. Here n_f denotes the number of finite eigenvalues and n_∞ the number of infinite eigenvalues of the matrix pencil $A - \lambda E$. By updating

$$B_1 := Q_1^T B = \begin{bmatrix} B_f \\ B_\infty \end{bmatrix}, \quad C_1 := C Q_2 = \begin{bmatrix} C_f & C_\infty \end{bmatrix}$$

with $B_\infty \in \mathbb{R}^{n_\infty\times m}$, $C_\infty \in \mathbb{R}^{p\times n_\infty}$ we obtain a r.s.e. descriptor system of the form

$$\begin{aligned} \begin{bmatrix} E_f & W_E \\ 0 & E_\infty \end{bmatrix} \begin{bmatrix} \dot{x}_f \\ \dot{x}_\infty \end{bmatrix} &= \begin{bmatrix} A_f & W_A \\ 0 & A_\infty \end{bmatrix} \begin{bmatrix} x_f \\ x_\infty \end{bmatrix} + \begin{bmatrix} B_f \\ B_\infty \end{bmatrix} u, \\ y &= \begin{bmatrix} C_f & C_\infty \end{bmatrix} \begin{bmatrix} x_f \\ x_\infty \end{bmatrix} + Du. \end{aligned} \tag{4.13}$$

Note that in Algorithm 4.2 we have to compute the eigenvalues of $A - \lambda E$ anyway, e.g., to obtain the frequency value ω_p, the additional update of B and C does not need a lot of extra work.

Decoupling of the System

The next step consists of block-diagonalizing the block-triangular matrix pencil (4.12). This form can be obtained by using the solution matrices Y, Z of the generalized Sylvester equation (see, e.g., [KVD90, KVD91, Ben09])

$$A_f Y + Z A_\infty + W_A = 0, \quad E_f Y + Z E_\infty + W_E = 0. \tag{4.14}$$

Numerical methods for the solution of the generalized Sylvester equation can be found in [KW87, KW89], the additive decomposition of a transfer function similar to the one in this thesis is discussed in [KVD90, KVD91].

A further system equivalence transform can be applied in order to obtain block-diagonal structures in E_1 and A_1 while keeping the upper triangular structure of E_1

and the upper quasi triangular structure of A_1, respectively. Finally we get

$$\begin{aligned}
E_2 &:= \begin{bmatrix} I & Z \\ 0 & I \end{bmatrix} \begin{bmatrix} E_f & W_E \\ 0 & E_\infty \end{bmatrix} \begin{bmatrix} I & Y \\ 0 & I \end{bmatrix} = \begin{bmatrix} E_f & 0 \\ 0 & E_\infty \end{bmatrix}, \\
A_2 &:= \begin{bmatrix} I & Z \\ 0 & I \end{bmatrix} \begin{bmatrix} A_f & W_A \\ 0 & A_\infty \end{bmatrix} \begin{bmatrix} I & Y \\ 0 & I \end{bmatrix} = \begin{bmatrix} A_f & 0 \\ 0 & A_\infty \end{bmatrix}, \\
B_2 &:= \begin{bmatrix} I & Z \\ 0 & I \end{bmatrix} \begin{bmatrix} B_f \\ B_\infty \end{bmatrix} = \begin{bmatrix} B_f + ZB_\infty \\ B_\infty \end{bmatrix}, \\
C_2 &:= \begin{bmatrix} C_f & C_\infty \end{bmatrix} \begin{bmatrix} I & Y \\ 0 & I \end{bmatrix} = \begin{bmatrix} C_f & C_fY + C_\infty \end{bmatrix},
\end{aligned}$$

and thus we obtain a descriptor system equivalent to (4.13)

$$\begin{aligned}
\begin{bmatrix} E_f & 0 \\ 0 & E_\infty \end{bmatrix} \begin{bmatrix} \dot{x}_f \\ \dot{x}_\infty \end{bmatrix} &= \begin{bmatrix} A_f & 0 \\ 0 & A_\infty \end{bmatrix} \begin{bmatrix} x_f \\ x_\infty \end{bmatrix} + \begin{bmatrix} B_f + ZB_\infty \\ B_\infty \end{bmatrix} u, \\
y &= \begin{bmatrix} C_f & C_fY + C_\infty \end{bmatrix} \begin{bmatrix} x_f \\ x_\infty \end{bmatrix} + Du.
\end{aligned} \tag{4.15}$$

Note that the transformation matrices $\mathcal{Z} := \begin{bmatrix} I & Z \\ 0 & I \end{bmatrix}$ and $\mathcal{Y} := \begin{bmatrix} I & Y \\ 0 & I \end{bmatrix}$ are generally not orthogonal anymore, hence the transformations may be ill-conditioned. However we can easily compute the 1-norm, ∞-norm, and Frobenius norm condition numbers $\kappa_1(\mathcal{Y})$, $\kappa_\infty(\mathcal{Y})$, and $\kappa_F(\mathcal{Y})$ of the transformation matrices since

$$\mathcal{Y}^{-1} = \begin{bmatrix} I & -Y \\ 0 & I \end{bmatrix}.$$

Then it can be seen that

$$\begin{aligned}
\kappa_1(\mathcal{Y}) :&= \left\|\mathcal{Y}\right\|_1 \left\|\mathcal{Y}^{-1}\right\|_1 = \left\|\begin{bmatrix} I & Y \\ 0 & I \end{bmatrix}\right\|_1 \left\|\begin{bmatrix} I & -Y \\ 0 & I \end{bmatrix}\right\|_1 \\
&= \left(1 + \max_{1 \leqslant j \leqslant n_\infty} \sum_{i=1}^{n_f} |y_{ij}|\right)^2
\end{aligned} \tag{4.16}$$

holds with the matrix elements y_{ij} of Y. Similarly it turns out that

$$\kappa_\infty(\mathcal{Y}) := \left\|\mathcal{Y}\right\|_\infty \left\|\mathcal{Y}^{-1}\right\|_\infty = \left(1 + \max_{1 \leqslant i \leqslant n_f} \sum_{j=1}^{n_\infty} |y_{ij}|\right)^2. \tag{4.17}$$

For the Frobenius norm condition number we obtain

$$\kappa_F(\mathcal{Y}) := \left\|\mathcal{Y}\right\|_F \left\|\mathcal{Y}^{-1}\right\|_F = n + \sum_{i=1}^{n_f} \sum_{j=1}^{n_\infty} |y_{ij}|^2. \tag{4.18}$$

Using these easily computable condition numbers we can measure the sensitivity of our data to perturbations. In this way we can abort the computation and return an error to the user if it exceeds a certain threshold.

Systems with Index One

We consider now the special case that the index of system (2.6) is one. This leads to some magnificent simplifications. Note first that $E_\infty = 0$. Then the generalized Sylvester equation can be reduced to the subsequent solution of $2n_\infty$ linear systems of equations

$$E_f Y = -W_E, \quad Z A_\infty = -(W_A + A_f Y).$$

Second the singular part of descriptor system (4.15) can be removed. From the second row of the state equation we obtain

$$x_\infty = -A_\infty^{-1} B_\infty u.$$

This step requires a matrix-vector-multiplication and the solution of m "small" linear systems of equations. By substituting x_∞ in the output equation, setting $B_f := B_f + Z B_\infty$, $C_\infty := C_f Y + C_\infty$, and some rearrangements we get the descriptor system

$$\begin{aligned} E_f \dot{x}_f &= A_f x_f + B_f u, \\ y &= C_f x_f + \left(D - C_\infty A_\infty^{-1} B_\infty\right) u. \end{aligned}$$

From Lemma 4.1 we obtain

$$\lim_{\omega \to \infty} \sigma_{\max}(G(\mathrm{i}\omega)) = \sigma_{\max}\left(D - C_\infty A_\infty^{-1} B_\infty\right). \tag{4.19}$$

Note that it is not necessary to compute Z. Hence we just have to compute the solution of n_∞ linear systems of equations to get Y which can be done cheaply. Then we can compute the new feedthrough matrix explicitly that requires basically n_∞ matrix-vector-multiplications and some matrix-matrix-additions.

Systems with Higher Index

Let now the index ν be greater than one. Then we have to solve the generalized Sylvester equation (4.14) in order to obtain the block-diagonal system (4.15). It can be seen that for the transfer function we obtain

$$\begin{aligned} G(s) &= \begin{bmatrix} C_f & C_\infty \end{bmatrix} \left(s \begin{bmatrix} E_f & 0 \\ 0 & E_\infty \end{bmatrix} - \begin{bmatrix} A_f & 0 \\ 0 & A_\infty \end{bmatrix} \right)^{-1} \begin{bmatrix} B_f \\ B_\infty \end{bmatrix} + D \\ &= \underbrace{C_f (sE_f - A_f)^{-1} B_f}_{:= G_{sp}(s)} + \underbrace{C_\infty (sE_\infty - A_\infty)^{-1} B_\infty + D}_{:= P(s)}. \end{aligned}$$

Note that G_{sp} is strictly proper because all eigenvalues of $A_f - \lambda E_f$ are finite. Therefore we can drop G_{sp} if we consider $\lim_{\omega\to\infty} \sigma_{\max}(G(\mathrm{i}\omega))$. Since we assume that G is proper, P has to be a constant polynomial so we can evaluate it at any point to determine its value. For $s = 0$ we obtain

$$\lim_{\omega\to\infty} G(\mathrm{i}\omega) = P(0) = D - C_\infty A_\infty^{-1} B_\infty,$$

and hence

$$\sigma_{\max}\left(G(\infty)\right) = \sigma_{\max}\left(D - C_\infty A_\infty^{-1} B_\infty\right). \tag{4.20}$$

Note that if we have performed a test for properness before, we can instantly compute $\sigma_{\max}\left(G(\infty)\right)$ by using (3.2).

The Overall Process

Summarizing the results of this section we obtain the following algorithm to compute the maximum singular value of $G(\infty)$.

Algorithm 4.3: Algorithm for Computing $\sigma_{\max}\left(G(\infty)\right)$

Input: Linear time-invariant descriptor system $(E; A, B, C, D)$ with transfer function G, tolerance τ.

Output: $\lim_{\omega\to\infty} \sigma_{\max}\left(G(\mathrm{i}\omega)\right)$.

1: Perform the QZ algorithm with eigenvalue reordering on $A - \lambda E$ to separate finite and infinite eigenvalues, i.e., compute orthogonal Q_1, $Q_2 \in \mathbb{R}^{n\times n}$ such that

$$E := Q_1^T E Q_2 = \begin{bmatrix} E_f & W_E \\ 0 & E_\infty \end{bmatrix}, \quad A := Q_1^T A Q_2 = \begin{bmatrix} A_f & W_A \\ 0 & A_\infty \end{bmatrix}.$$

2: Set $B := Q_1^T B = \begin{bmatrix} B_f \\ B_\infty \end{bmatrix}$, $\quad C := C Q_2 = \begin{bmatrix} C_f & C_\infty \end{bmatrix}$.

3: **if** $\|E_\infty\| < \tau$ **then**

4: *% The index is 1.*

5: Solve n_∞ linear systems $E_f Y = -W_E$.

6: Set $C_\infty := C_f Y + C_\infty$.

7: Compute $\lim_{\omega\to\infty} \sigma_{\max}\left(G\left(\mathrm{i}\omega\right)\right) = \sigma_{\max}\left(D - C_\infty A_\infty^{-1} B_\infty\right)$.

8: **else**

9: *% The index is higher.*

10: Solve the generalized Sylvester equation

$$A_f Y + Z A_\infty + W_A = 0, \quad E_f Y + Z E_\infty + W_E = 0.$$

11: Define $\mathcal{Y} = \begin{bmatrix} I & Y \\ 0 & I \end{bmatrix}$ and compute $\kappa(\mathcal{Y})$ using one of the formulae (4.16), (4.17), (4.18).
12: **if** $\kappa(\mathcal{Y}) > \kappa_{\max}$ **then**
13: *% The transformation matrix $\mathcal{Y}$ is very ill-conditioned.*
14: return.
15: **end if**
16: Set $C_\infty := C_f Y + C_\infty$.
17: Compute $\lim_{\omega\to\infty} \sigma_{\max}\left(G\left(\mathrm{i}\omega\right)\right) = \sigma_{\max}\left(D - C_\infty A_\infty^{-1} B_\infty\right)$.
18: **end if**

4.4 Improving the Accuracy of the Eigenvalue Computation

Naively computing the matrix M_γ in (4.1) could be very ill-advised because it contains a lot of matrix products and inverses. The matrices R and S could be ill-conditioned and even if they are not, forming "matrix-times-its-transpose" products like $BR^{-1}B^T$ suffers from the same kind of numerical instability as forming the normal equations to solve linear least square problems (see Example 5.3.2 in [GVL96]). When explicitely computing the blocks of M_γ this could easily corrupt the entries of the matrix by rounding errors and hence highly perturb the eigenvalues of the matrix pencil $M_\gamma - \lambda N$. Especially purely imaginary eigenvalues can be easily moved away from the imaginary axis by this kind of errors which forces our algorithm for computing the $\mathcal{L}_\infty$-norm to produce wrong results. Therefore it is desirable to work directly on the original data without explicitely forming matrix products and inverses. In this section we show how one can achieve this by extending the original pencil into a larger one and how one can transform this extended pencil again to skew-Hamiltonian/Hamiltonian structure. First we need the following lemma.

Lemma 4.3 (Eigenvalues of an Extended Matrix Pencil)**.** *Let $M - \lambda N = \left(A - BD^{-1}C\right) - \lambda E$ be a given matrix pencil with matrices A, $E \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{m\times n}$ and an invertible matrix $D \in \mathbb{R}^{m\times m}$. Then the extended matrix pencil $\mathcal{M} - \lambda\mathcal{N} := \begin{bmatrix} A & B \\ C & D \end{bmatrix} - \lambda \begin{bmatrix} E & 0 \\ 0 & 0 \end{bmatrix}$ has the same finite spectrum as $M - \lambda N$, i.e., $\Lambda_f(M,N) = \Lambda_f(\mathcal{M},\mathcal{N})$.*

Proof. Let $\lambda \in \Lambda_f(M,N)$, i.e., there exists a nonzero vector $x \in \mathbb{R}^n$ such that $\lambda N x = Mx$, in other words

$$\lambda E x = \left(A - BD^{-1}C\right) x. \tag{4.21}$$

Defining the vector $y := -D^{-1}Cx \in \mathbb{R}^m$ leads to

$$Cx + Dy = 0,$$

and by substitution in (4.21) we obtain

$$\lambda E x = Ax + By.$$

These two equations can be rewritten as

$$\lambda \begin{bmatrix} E & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix},$$

consequently $\lambda \in \Lambda_f(\mathcal{M},\mathcal{N})$ and hence $\Lambda_f(M,N) \subset \Lambda_f(\mathcal{M},\mathcal{N})$. The relation $\Lambda_f(M,N) \supset \Lambda_f(\mathcal{M},\mathcal{N})$ can be easily shown by going the argumentation of this proof conversely. □

Now we can apply Lemma 4.3 to extend the matrix pencil (4.1) to

$$\mathcal{M}_\gamma - \lambda\mathcal{N} = \left[\begin{array}{cc|cc} A & 0 & B & 0 \\ 0 & -A^T & 0 & -C^T \\ \hline C & 0 & D & -\gamma I_p \\ 0 & B^T & -\gamma I_m & D^T \end{array}\right] - \lambda \left[\begin{array}{cc|cc} E & 0 & 0 & 0 \\ 0 & E^T & 0 & 0 \\ \hline 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right]$$

which has the same finite eigenvalues as $M_\gamma - \lambda N$. Next we want to transform this matrix pencil to skew-Hamiltonian/Hamiltonian structure in order to be able to use sophisticated eigenvalue solvers. Following from the special block structure property of (skew-)Hamiltonian matrices (see Lemma 5.1) it is clear that the order of any skew-Hamiltonian/Hamiltonian matrix pencil is even. However, the order of $\mathcal{M}_\gamma - \lambda\mathcal{N}$ is $2n + m + p$ which is odd if and only if $m + p$ is odd. In this case we have to embed this extended pencil further into a larger matrix pencil to obtain an even order (see [BBMX99]). There are several ways to do that. We propose two possibilities. The first one works as follows:

If $m + p$ is odd we introduce one artificial input (or output) and extend each the input matrix (or output matrix) and feedthrough matrix by one zero column (or row), i.e.,

$$\tilde{B} := \begin{array}{c} \\ n \end{array} \begin{array}{c} \begin{array}{cc} m & 1 \end{array} \\ \left[\begin{array}{cc} B & 0 \end{array}\right] \end{array}, \qquad \tilde{D} := \begin{array}{c} \\ p \end{array} \begin{array}{c} \begin{array}{cc} m & 1 \end{array} \\ \left[\begin{array}{cc} D & 0 \end{array}\right] \end{array}.$$

In this way we obtain an extended transfer function

$$G_{ext}(s) := C\,(sE - A)^{-1}\,\tilde{B} + \tilde{D} = \begin{array}{c} \\ p \end{array} \begin{array}{c} \begin{array}{cc} m & 1 \end{array} \\ \left[\begin{array}{cc} G(s) & 0 \end{array}\right] \end{array}$$

which has the same singular values as G for all $s \in \mathbb{C}$ and hence the same $\mathcal{L}_\infty$-norm but the corresponding extended matrix pencil

$$\tilde{\mathcal{M}}_\gamma - \lambda\tilde{\mathcal{N}} = \left[\begin{array}{cc|cc} A & 0 & \tilde{B} & 0 \\ 0 & -A^T & 0 & -C^T \\ \hline C & 0 & \tilde{D} & -\gamma I_p \\ 0 & \tilde{B}^T & -\gamma I_{\tilde{m}} & \tilde{D}^T \end{array}\right] - \lambda \left[\begin{array}{cc|cc} E & 0 & 0 & 0 \\ 0 & E^T & 0 & 0 \\ \hline 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right] \tag{4.22}$$

with $\tilde{m} = m+1$ has even order. If $m+p$ is already even we simply set $\tilde{B} = B$, $\tilde{D} = D$ and $\tilde{m} = m$. Multiplying the second block row of (4.22) by -1 and subsequently permuting the first and second block row and the third and fourth block column leads to the *even matrix pencil* (i.e., $\hat{\mathcal{M}}_\gamma$ is symmetric and $\hat{\mathcal{N}}$ is skew-symmetric, see [Sch08b])

$$\hat{\mathcal{M}}_\gamma - \lambda\hat{\mathcal{N}} = \left[\begin{array}{cc|cc} 0 & A^T & C^T & 0 \\ A & 0 & 0 & \tilde{B} \\ \hline C & 0 & -\gamma I_p & \tilde{D} \\ 0 & \tilde{B}^T & \tilde{D}^T & -\gamma I_{\tilde{m}} \end{array}\right] - \lambda \left[\begin{array}{cc|cc} 0 & -E^T & 0 & 0 \\ E & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right].$$

Now we can exploit the symmetries of the matrix $\hat{\mathcal{M}}_\gamma$ and repartition its blocks. We define $k = \frac{\tilde{m}+p}{2}$ and obtain

$$\begin{array}{c} \\ n \\ n \end{array} \begin{array}{c} \begin{array}{cc} p & \tilde{m} \end{array} \\ \left[\begin{array}{cc} C^T & 0 \\ 0 & \tilde{B} \end{array}\right] \end{array} =: \begin{array}{c} \\ n \\ n \end{array} \begin{array}{c} \begin{array}{cc} k & k \end{array} \\ \left[\begin{array}{cc} R_{11} & R_{12} \\ R_{21} & R_{22} \end{array}\right] \end{array}$$

and further for the lower right block

$$\begin{array}{c} \\ p \\ \tilde{m} \end{array} \begin{array}{c} \begin{array}{cc} p & \tilde{m} \end{array} \\ \left[\begin{array}{cc} -\gamma I_p & \tilde{D} \\ \tilde{D}^T & -\gamma I_{\tilde{m}} \end{array}\right] \end{array} =: \begin{array}{c} \\ k \\ k \end{array} \begin{array}{c} \begin{array}{cc} k & k \end{array} \\ \left[\begin{array}{cc} S_{11} & S_{12} \\ S_{12}^T & S_{22} \end{array}\right] \end{array}.$$

with $S_{11} = S_{11}^T$ and $S_{22} = S_{22}^T$. This leads to the matrix pencil

$$\hat{\mathcal{M}}_\gamma - \lambda\hat{\mathcal{N}} = \left[\begin{array}{cc|cc} 0 & A^T & R_{11} & R_{12} \\ A & 0 & R_{21} & R_{22} \\ \hline R_{11}^T & R_{21}^T & S_{11} & S_{12} \\ R_{12}^T & R_{22}^T & S_{12}^T & S_{22} \end{array}\right] - \lambda \left[\begin{array}{cc|cc} 0 & -E^T & 0 & 0 \\ E & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right].$$

Subsequently permuting the first with second block row, the second with the third block row, the second with the fourth block row, the second with the third block column and taking the negative of the last two block rows yields again a skew-Hamiltonian/Hamiltonian matrix pencil of the form

$$\bar{\mathcal{M}}_\gamma - \lambda\bar{\mathcal{N}} = \left[\begin{array}{cc|cc} A & R_{21} & 0 & R_{22} \\ R_{12}^T & S_{12}^T & R_{22}^T & S_{22} \\ \hline 0 & -R_{11} & -A^T & -R_{12} \\ -R_{11}^T & -S_{11} & -R_{21}^T & -S_{12} \end{array}\right] - \lambda \left[\begin{array}{cc|cc} E & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \hline 0 & 0 & E^T & 0 \\ 0 & 0 & 0 & 0 \end{array}\right], \tag{4.23}$$

which contains no matrix products and no inverses and has the same eigenvalues as the original matrix pencil. This can now be treated by the structure-exploiting method proposed in Chapter 5.

We now briefly discuss a second embedding strategy which is also based on the introduction of artificial inputs or outputs and extending the corresponding input, output, and feedthrough matrices by zero blocks of adequate sizes. We assume first that $m < p$. Then we introduce $p - m$ artificial inputs and extend the matrices B and D each by $p - m$ zero columns. If on the other hand $p < m$ we introduce $m - p$ artificial outputs and extend the matrices C and D by $m - p$ zero rows. So, defining $r = \max\{0, p - m\}$ and $s = \max\{0, m - p\}$ we obtain

$$\tilde{B} := \begin{array}{c} \\ n \end{array} \begin{array}{c} \begin{array}{cc} m & r \end{array} \\ \left[\begin{array}{cc} B & 0 \end{array}\right] \end{array}, \quad \tilde{C} := \begin{array}{c} p \\ s \end{array} \begin{array}{c} \begin{array}{c} n \end{array} \\ \left[\begin{array}{c} C \\ 0 \end{array}\right] \end{array}, \quad \tilde{D} := \begin{array}{c} p \\ s \end{array} \begin{array}{c} \begin{array}{cc} m & r \end{array} \\ \left[\begin{array}{cc} D & 0 \\ 0 & 0 \end{array}\right] \end{array},$$

and the extended transfer function built by these matrices has the same singular values as the original transfer function for all values $s \in \mathbb{C}$. Now we can set up the extended pencil

$$\tilde{\mathcal{M}}_\gamma - \lambda\tilde{\mathcal{N}} = \left[\begin{array}{cc|cc} A & 0 & \tilde{B} & 0 \\ 0 & -A^T & 0 & -\tilde{C}^T \\ \hline \tilde{C} & 0 & \tilde{D} & -\gamma I_{p+s} \\ 0 & \tilde{B}^T & -\gamma I_{m+r} & \tilde{D}^T \end{array}\right] - \lambda \left[\begin{array}{cc|cc} E & 0 & 0 & 0 \\ 0 & E^T & 0 & 0 \\ \hline 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right] \tag{4.24}$$

which has the order $2(n + \max\{m, p\})$, so if $|m - p|$ is small, the difference of the sizes of (4.22) and (4.24) is not very large. Since $l := m + r = p + s = \max\{m, p\}$ we can simply multiply the last block row of (4.24) by -1, permute the second with the third block row and the second with the third block column and already obtain a skew-Hamiltonian/Hamiltonian matrix pencil

$$\bar{\mathcal{M}}_\gamma - \lambda\bar{\mathcal{N}} = \left[\begin{array}{cc|cc} A & \tilde{B} & 0 & 0 \\ \tilde{C} & \tilde{D} & 0 & -\gamma I_l \\ \hline 0 & 0 & -A^T & -\tilde{C}^T \\ 0 & \gamma I_l & -\tilde{B}^T & -\tilde{D}^T \end{array}\right] - \lambda \left[\begin{array}{cc|cc} E & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \hline 0 & 0 & E^T & 0 \\ 0 & 0 & 0 & 0 \end{array}\right]. \tag{4.25}$$

This matrix pencil is generally larger than (4.23) but it has a more special structure, i.e., the upper right and lower left 2×2 block matrices of $\bar{\mathcal{M}}_\gamma$ contain only one submatrix $\pm\gamma I_l$. So possibly we can exploit this special structure which may lead to some computational savings compared to the usage of (4.23). It might be part of further investigations to check if this is a reasonable approach.

4.5 A Brief View on Discrete-Time Systems

In this section we briefly investigate the discrete-time case and show the differences to the continuous-time case. In particular we derive the discrete-time analogue to

the skew-Hamiltonian/Hamiltonian matrix pencils that arise in the continuous-time case. Following [GVDV98] in the continuous-time case we have to consider the skew-Hamiltonian/Hamiltonian matrix pencil

$$\begin{aligned} M_\gamma - \lambda N &= \begin{bmatrix} A - \lambda E & BB^T \\ 0 & A^T + \lambda E^T \end{bmatrix} - \begin{bmatrix} BD^T \\ C^T \end{bmatrix} S^{-1} \begin{bmatrix} C & DB^T \end{bmatrix} \\ &= \begin{bmatrix} A - \lambda E - BD^T S^{-1} C & BB^T - BD^T S^{-1} DB^T \\ -C^T S^{-1} C & A^T + \lambda E^T - C^T S^{-1} DB^T \end{bmatrix} \end{aligned} \tag{4.26}$$

with $S := DD^T - \gamma^2 I$ which can be shown to be equivalent to the matrix pencil (4.1) by applying $P := \begin{bmatrix} I & 0 \\ 0 & -\gamma I \end{bmatrix}$ from the left and $Q := \begin{bmatrix} I & 0 \\ 0 & \frac{1}{\gamma} I \end{bmatrix}$ from the right.

In case of discrete-time systems we consider the matrix pencil

$$\begin{aligned} M_\gamma - \lambda N_\gamma &= \begin{bmatrix} A - \lambda E & BB^T \\ 0 & \lambda A^T - E^T \end{bmatrix} - \begin{bmatrix} BD^T \\ \lambda C^T \end{bmatrix} S^{-1} \begin{bmatrix} C & DB^T \end{bmatrix} \\ &= \begin{bmatrix} A - \lambda E - BD^T S^{-1} C & BB^T - BD^T S^{-1} DB^T \\ -\lambda C^T S^{-1} C & \lambda A^T - E^T - \lambda C^T S^{-1} DB^T \end{bmatrix} \end{aligned} \tag{4.27}$$

with S as above as described again in [GVDV98]. By applying again the matrix P from the left and Q from the right, comparing the block structures of (4.26) and (4.27), and considering (4.1) we obtain a matrix pencil of the form

$$\tilde{M}_\gamma - \lambda \tilde{N}_\gamma = \begin{bmatrix} A - \lambda E & 0 \\ 0 & E^T - \lambda A^T \end{bmatrix} + \begin{bmatrix} B & 0 \\ 0 & -\lambda C^T \end{bmatrix} \begin{bmatrix} -D & \gamma I \\ \gamma I & -D^T \end{bmatrix}^{-1} \begin{bmatrix} C & 0 \\ 0 & B^T \end{bmatrix}. \tag{4.28}$$

Note that now also the matrices N_γ and $\tilde{N}_\gamma$ depend on γ. Now we can apply an extension strategy similar to the one in Section 4.4 and obtain the extended matrix pencil

$$\begin{aligned} \mathcal{M}_\gamma - \lambda \mathcal{N} &= \left[\begin{array}{cc|cc} A - \lambda E & 0 & B & 0 \\ 0 & E^T - \lambda A^T & 0 & -\lambda C^T \\ \hline C & 0 & D & -\gamma I_p \\ 0 & B^T & -\gamma I_m & D^T \end{array}\right] \\ &= \left[\begin{array}{cc|cc} A & 0 & B & 0 \\ 0 & E^T & 0 & 0 \\ \hline C & 0 & D & -\gamma I_p \\ 0 & B^T & -\gamma I_m & D^T \end{array}\right] - \lambda \left[\begin{array}{cc|cc} E & 0 & 0 & 0 \\ 0 & A^T & 0 & C^T \\ \hline 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right] \end{aligned} \tag{4.29}$$

which has the same finite spectrum as (4.28). By subsequently permuting the first with the second block row and the third with the fourth block column, taking the

negative of the first block column and transposing the whole matrix pencil we obtain

$$\hat{\mathcal{M}}_\gamma - \lambda\hat{\mathcal{N}} = \left[\begin{array}{c|ccc} 0 & -A^T & -C^T & 0 \\ \hline E & 0 & 0 & B \\ 0 & 0 & -\gamma I_p & D \\ 0 & B^T & D^T & -\gamma I_m \end{array}\right] - \lambda \left[\begin{array}{c|ccc} 0 & -E^T & 0 & 0 \\ \hline A & 0 & 0 & 0 \\ C & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right], \tag{4.30}$$

which is called *extended symplectic matrix pencil* [Sim09] since it has symplectic eigensymmetry, i.e if $\lambda \in \Lambda(\hat{\mathcal{M}}_\gamma, \hat{\mathcal{N}})$, also $\bar{\lambda}^{-1} \in \Lambda(\hat{\mathcal{M}}_\gamma, \hat{\mathcal{N}})$. In other words, there is a symmetry with respect to the unit circle. If $\mathrm{Re}(\lambda) \neq 0$ we additionally get the eigenvalues $-\lambda$ and $-\bar{\lambda}^{-1}$. The matrix pencil above is a special case of the block structured matrix pencil

$$\mathcal{A}_D - \lambda\mathcal{E}_D = \begin{bmatrix} 0 & G \\ -F^T & D \end{bmatrix} - \lambda \begin{bmatrix} 0 & F \\ -G^T & 0 \end{bmatrix} \tag{4.31}$$

with a symmetric matrix D of appropriate size which is also called *D-type matrix pencil*. An even matrix pencil of the form

$$\mathcal{A}_C - \lambda\mathcal{E}_C = \begin{bmatrix} 0 & \tilde{G} \\ \tilde{G}^T & \tilde{D} \end{bmatrix} - \lambda \begin{bmatrix} 0 & \tilde{F} \\ -\tilde{F}^T & 0 \end{bmatrix} \tag{4.32}$$

with a symmetric matrix $\tilde{D}$ is called *C-type matrix pencil* and has a Hamiltonian spectrum, i.e., eigensymmetry with respect to the imaginary axis (see [Xu06]). We can state an equivalence transformation between D-type and C-type matrix pencils using the *Cayley transform*, $\mathbf{c} : \mathbb{C} \cup \{\infty\} \longrightarrow \mathbb{C} \cup \{\infty\}$ which is defined by

$$\mu = \mathbf{c}(\lambda) := (\lambda - 1)(\lambda + 1)^{-1}, \quad \mathbf{c}(-1) = \infty, \quad \mathbf{c}(\infty) = 1 \tag{4.33}$$

and the *generalized Cayley transform* for matrix pencils,

$$\mathcal{B} - \lambda\mathcal{F} = \mathbf{c}(\mathcal{A}, \mathcal{E}) := (\mathcal{A} - \mathcal{E}) - \lambda(\mathcal{A} + \mathcal{E}). \tag{4.34}$$

Let $\tilde{\mathcal{A}} - \lambda\tilde{\mathcal{E}} := \mathbf{c}(\mathcal{A}_D, \mathcal{E}_D)$. Then, the eigenvalue pair $\left(\lambda, \bar{\lambda}^{-1}\right)$ of $\mathcal{A}_D - \lambda\mathcal{E}_D$ is transformed into the eigenvalue pair $(\mu, -\bar{\mu})$ of $\tilde{\mathcal{A}} - \lambda\tilde{\mathcal{E}}$, with $\mu = \mathbf{c}(\lambda)$, $-\bar{\mu} = \mathbf{c}\left(\bar{\lambda}^{-1}\right)$. The same also holds for the complex conjugate eigenvalues in case of eigenvalue quadruples. In particular, eigenvalues $-1 \neq \lambda \in \Lambda(\mathcal{A}_D, \mathcal{E}_D)$ satisfying $|\lambda| = 1$ (i.e., those on the boundary of the stability domain of a corresponding discrete-time system) are mapped to eigenvalues $\mu \in \Lambda\left(\tilde{\mathcal{A}}, \tilde{\mathcal{E}}\right)$ with $\mathrm{Re}(\mu) = 0$ (i.e., those on the boundary of the stability domain of a continuous-time system). This is summarized in the following lemma.

Lemma 4.4 (Transformation of Eigenvalues on the Unit Circle)**.** *Let $-1 \neq \lambda \in \mathbb{C}$ with $|\lambda| = 1$ be given. Then $\mathrm{Re}(\mu) = 0$ with $\mu = \mathbf{c}(\lambda)$ as in (4.33) holds.*

Proof. Since $|\lambda| = 1$ we have the representation $\lambda = e^{\mathrm{i}\omega}$ with $\omega \in [0, 2\pi)$. That is,

$$\begin{aligned}\mu &:= \frac{e^{\mathrm{i}\omega} - 1}{e^{\mathrm{i}\omega} + 1} \\ &= \frac{\left(e^{\mathrm{i}\omega} - 1\right)\left(e^{-\mathrm{i}\omega} + 1\right)}{\left(e^{\mathrm{i}\omega} + 1\right)\left(e^{-\mathrm{i}\omega} + 1\right)} \\ &= \frac{e^{\mathrm{i}\omega} - e^{-\mathrm{i}\omega}}{e^{\mathrm{i}\omega} + e^{-\mathrm{i}\omega} + 2}\end{aligned}$$

which is always defined since $\omega \neq \pi$, because $\lambda \neq -1$. Now it can be easily checked that $e^{\mathrm{i}\omega} + e^{-\mathrm{i}\omega}$ is a real number and that $e^{\mathrm{i}\omega} - e^{-\mathrm{i}\omega}$ is an imaginary number. Hence also μ is an imaginary number which proves the assertion. □

Unfortunately, $\tilde{\mathcal{A}} - \lambda\tilde{\mathcal{E}}$ does not have the same block structure as $\mathcal{A}_C - \lambda\mathcal{E}_C$, and it cannot be put in the continuous-time setting. But this objective can be enforced by using a Cayley transform followed by a *drop/add transformation*

$$\mathcal{A}_C - \lambda\mathcal{E}_C = \mathbf{t}\left(\mathcal{A}_D, \mathcal{E}_D\right), \tag{4.35}$$

with $\mathbf{t}(\cdot) = \mathbf{d}\left(\mathbf{c}(\cdot)\right)$, and $\mathbf{d}$ corresponds to dropping/adding D in the $\mathcal{E}$ part. The transformation $\mathbf{d}$ is given by

$$\begin{aligned}\mathbf{d}\left(\tilde{\mathcal{A}}, \tilde{\mathcal{E}}\right) &:= \begin{bmatrix}(1-\lambda)I & 0\\ 0 & I\end{bmatrix}\left(\tilde{\mathcal{A}} - \lambda\tilde{\mathcal{E}}\right)\begin{bmatrix}I & 0\\ 0 & \frac{1}{1-\lambda}I\end{bmatrix} \\ &= \begin{bmatrix}(1-\lambda)I & 0\\ 0 & I\end{bmatrix}\left(\begin{bmatrix}0 & \tilde{G}\\ \tilde{G}^T & D\end{bmatrix} - \lambda\begin{bmatrix}0 & \tilde{F}\\ -\tilde{F}^T & D\end{bmatrix}\right)\begin{bmatrix}I & 0\\ 0 & \frac{1}{1-\lambda}I\end{bmatrix},\end{aligned} \tag{4.36}$$

similar to the transformation matrices for the palindromification of a matrix pencil in [Sch08b]. Note that the transformation matrices depend on λ and have poles for $\lambda \in \{1, \infty\}$ which are the images of the Cayley transform of $\lambda = \infty$, and $\lambda = -1$, respectively. As a consequence, the multiplicities of the eigenvalues $1, \infty \in \Lambda\left(\tilde{\mathcal{A}}, \tilde{\mathcal{E}}\right)$ may have changed. All multiplicities of all the other eigenvalues, however, are preserved. For more details see [Xu06]. The $\mathbf{t}$ *transformation diagram* is shown below:

$$\begin{array}{c}\mathcal{A}_D - \lambda\mathcal{E}_D = \begin{bmatrix}0 & G\\ -F^T & D\end{bmatrix} - \lambda\begin{bmatrix}0 & F\\ -G^T & 0\end{bmatrix} \\ \mathbf{c} \;\downarrow\uparrow\; \mathbf{c}^{-1} \\ \tilde{\mathcal{A}} - \lambda\tilde{\mathcal{E}} = \begin{bmatrix}0 & \tilde{G}\\ \tilde{G}^T & D\end{bmatrix} - \lambda\begin{bmatrix}0 & \tilde{F}\\ -\tilde{F}^T & D\end{bmatrix} \\ \text{drop } D \text{ from } \tilde{\mathcal{E}} \;\downarrow\uparrow\; \text{add } D \text{ to } \tilde{\mathcal{E}}\end{array}$$

$$\mathcal{A}_C - \lambda\mathcal{E}_C = \begin{bmatrix} 0 & \tilde{G} \\ \tilde{G}^T & \tilde{D} \end{bmatrix} - \lambda \begin{bmatrix} 0 & \tilde{F} \\ -\tilde{F}^T & 0 \end{bmatrix},$$

where $\tilde{F} := G + F$, $\tilde{G} := G - F$, and $\tilde{D} := D$. Clearly, the **t** transformation involves matrix additions and subtractions only. Applying the **t** transform to the extended symplectic matrix pencil (4.30) yields the even matrix pencil

$$\bar{\mathcal{M}}_\gamma - \lambda\bar{\mathcal{N}} = \left[\begin{array}{c|ccc} 0 & -A^T + E^T & -C^T & 0 \\ \hline -A + E & 0 & 0 & B \\ -C & 0 & -\gamma I_p & D \\ 0 & B^T & D^T & -\gamma I_m \end{array}\right] - \lambda \left[\begin{array}{c|ccc} 0 & -A^T - E^T & -C^T & 0 \\ \hline A + E & 0 & 0 & 0 \\ C & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right]. \quad (4.37)$$

Using the transformations introduced in Section 4.4 this matrix pencil can be transformed to a skew-Hamiltonian/Hamiltonian one and hence we can use the structure-exploiting algorithm from Chapter 5 in order to compute its eigenvalues.

Concerning the spectrum of **c** and **t** transformed matrix pencils we have the following statements [Xu06].

Lemma 4.5 (Regularity and Spectrum of a Cayley Transformed Matrix Pencil)**.** *Consider $\mathcal{A} - \lambda\mathcal{E}$ and $\mathcal{B} - \lambda\mathcal{F}$ where $\mathcal{B} - \lambda\mathcal{F} = \mathbf{c}(\mathcal{A}, \mathcal{E})$. Then the following holds.*

(i) $\mathcal{A} - \lambda\mathcal{E}$ is regular if and only if $\mathcal{B} - \lambda\mathcal{F}$ is regular.

(ii) $\lambda_0 \in \Lambda(\mathcal{A}, \mathcal{E})$ if and only if $\mu_0 = \mathbf{c}(\lambda_0) \in \Lambda(\mathcal{B}, \mathcal{F})$. Moreover, λ_0, μ_0 have the same partial, geometric, and algebraic multiplicities.

Proof. See [Xu06]. □

Lemma 4.6 (Regularity and Spectrum of a **t** Transformed D-Type Matrix Pencil)**.** *Let $\mathcal{A}_D - \lambda\mathcal{E}_D$ be a given D-type matrix pencil and $\mathcal{A}_C - \lambda\mathcal{E}_C = \mathbf{t}(\mathcal{A}_D, \mathcal{E}_D)$ the resulting C-type matrix pencil. Then the following holds.*

(i) $\mathcal{A}_D - \lambda\mathcal{E}_D$ is regular if and only if $\mathcal{A}_C - \lambda\mathcal{E}_C$ is regular.

(ii) $\lambda_0 \in \Lambda(\mathcal{A}_D, \mathcal{E}_D)$ $(\lambda_0 \neq -1, \infty)$ if and only if $\mu_0 = \mathbf{c}(\mathcal{A}_C, \mathcal{E}_C)$ $(\mu_0 \neq \infty, 1)$. Both λ_0, μ_0 have the same partial, geometric, and algebraic multiplicities.

Proof. See [Xu06]. □

From Lemma 4.5 it can be seen that the eigenvalue $\lambda = -1$ of a D-type matrix pencil (which is on the unit circle) requires special treatment since it is mapped on the eigenvalue $\mu = \infty$ by the Cayley transform. Additionally, the relations between $\lambda = -1 \in \Lambda(\mathcal{A}_D, \mathcal{E}_D)$, and $\mu = \infty \in \Lambda(\mathcal{A}_C, \mathcal{E}_C)$ and their multiplicities are more involved (see again [Xu06]) since these may be changed by the transformation **d**.

There exists also another way to transform the extended symplectic matrix pencil (4.30) to a more convenient structure. We first multiply the first block row of the matrix pencil by -1 and subsequently perform a two-sided transformation similar to the one in (4.36), i.e., we obtain

$$\begin{aligned}\bar{\mathcal{M}}_\gamma - \lambda\bar{\mathcal{N}}_\gamma := &\begin{bmatrix} \frac{1}{1-\lambda}I_n & & & \\ & I_n & & \\ & & I_p & \\ & & & I_m \end{bmatrix} \left(\left[\begin{array}{c|ccc} 0 & A^T & C^T & 0 \\ \hline E & 0 & 0 & B \\ 0 & 0 & -\gamma I_p & D \\ 0 & B^T & D^T & -\gamma I_m \end{array}\right] \right. \\ &\left. - \lambda \left[\begin{array}{c|ccc} 0 & E^T & 0 & 0 \\ \hline A & 0 & 0 & 0 \\ C & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right] \right) \begin{bmatrix} I_n & & & \\ & (1-\lambda)I_n & & \\ & & (1-\lambda)I_p & \\ & & & (1-\lambda)I_m \end{bmatrix} \\ = &\left[\begin{array}{c|ccc} 0 & A^T & C^T & 0 \\ \hline E & 0 & 0 & B \\ 0 & 0 & -\gamma I_p & D \\ 0 & B^T & D^T & -\gamma I_m \end{array}\right] - \lambda \left[\begin{array}{c|ccc} 0 & E^T & 0 & 0 \\ \hline A & 0 & 0 & B \\ C & 0 & -\gamma I_p & D \\ 0 & B^T & D^T & -\gamma I_m \end{array}\right]. \end{aligned} \tag{4.38}$$

This is now a *palindromic matrix pencil* which means that $\bar{\mathcal{M}}_\gamma = \bar{\mathcal{N}}_\gamma^T$. Again the transformation matrices depend on λ and have poles for $\lambda \in \{1, \infty\}$, hence the multiplicities of the eigenvalues $1, \infty$ may have changed. Note that palindromic matrix pencils have symplectic eigensymmetry [Sch08b]. Recently, structure-preserving algorithms and related software for the computation of the eigenvalues of palindromic matrix pencils became available, see [KSW09, PST09, Sch08b]. Of course, all these things are still just ideas. In future research these strategies have to be analyzed in more detail. E.g., we have to think about solving the problems with the eigenvalue $\lambda = -1$ in the first strategy or $\lambda = 1$ in the second method. Also the implementation of an algorithm for computing the $\mathcal{L}_\infty$-norm of a discrete-time descriptor system is going to be future work.

5 A New Method for the Arising Generalized Eigenvalue Problems

As mentioned in Chapter 4, the involved matrix pencils $M_\gamma - \lambda N$ have skew-Hamiltonian/Hamiltonian structure. The aim of this chapter is to illustrate the main structural properties of these matrix pencils and to develop a numerical method which exploits the matrix structures. Often structure-preserving methods are faster and more accurate than standard methods and so we want to considerably improve the speed and accuracy of the $\mathcal{L}_\infty$-norm algorithm.

5.1 Theoretical Preliminaries

5.1.1 Involved Matrix Structures

First of all we define the matrix structures that we require in the sequel (see [BBMX99, BBL+07]) and state some of their properties.

Definition 5.1 (Matrix Structures)**.** Let $\mathcal{J}_{2n} := \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}$, where I_n is the $n \times n$ identity matrix. To avoid a too complex notation, we omit the subscript $2n$ if the size of the matrix is clear.

(i) A matrix $\mathcal{H} \in \mathbb{R}^{2n\times 2n}$ is *Hamiltonian* if $(\mathcal{H}\mathcal{J})^T = \mathcal{H}\mathcal{J}$. The Lie algebra of Hamiltonian matrices in $\mathbb{R}^{2n\times 2n}$ is denoted by $\mathbb{H}_{2n}$.

(ii) A matrix $\mathcal{N} \in \mathbb{R}^{2n\times 2n}$ is *skew-Hamiltonian* if $(\mathcal{N}\mathcal{J})^T = -\mathcal{N}\mathcal{J}$. The Jordan algebra of skew-Hamiltonian matrices in $\mathbb{R}^{2n\times 2n}$ is denoted by $\mathbb{SH}_{2n}$.

(iii) A matrix pencil $\mathcal{H} - \lambda\mathcal{N} \in \mathbb{R}^{2n\times 2n}$ is *skew-Hamiltonian/Hamiltonian* if $\mathcal{N} \in \mathbb{SH}_{2n}$ and $\mathcal{H} \in \mathbb{H}_{2n}$.

(iv) A matrix $\mathcal{S} \in \mathbb{R}^{2n\times 2n}$ is *symplectic* if $\mathcal{S}\mathcal{J}\mathcal{S}^T = \mathcal{J}$. The Lie group of symplectic matrices in $\mathbb{R}^{2n\times 2n}$ is denoted by $\mathbb{S}_{2n}$.

(v) A matrix $\mathcal{U} \in \mathbb{R}^{2n\times 2n}$ is *orthogonal symplectic* if $\mathcal{U}\mathcal{J}\mathcal{U}^T = \mathcal{J}$ and $\mathcal{U}\mathcal{U}^T = I_{2n}$. The compact Lie group of orthogonal symplectic matrices in $\mathbb{R}^{2n\times 2n}$ is denoted by $\mathbb{US}_{2n}$.

For more on the algebraic structures from Definition 5.1 see, e.g., [Hal03, Jac68]. Some of the matrices above satisfy specific block structures which we want to summarize within the next lemma (see also [BK06]).

Lemma 5.1 (Matrix Block Structures)**.** *The matrix classes of Definition 5.1 satisfy the following structures:*

(i) Every matrix $\mathcal{H} \in \mathbb{H}_{2n}$ *can be written as* $\mathcal{H} = \begin{bmatrix} F & G \\ H & -F^T \end{bmatrix}$ *with* $F \in \mathbb{R}^{n\times n}$ *and symmetric matrices* G, $H \in \mathbb{R}^{n\times n}$.

(ii) Every matrix $\mathcal{N} \in \mathbb{SH}_{2n}$ *can be written as* $\mathcal{N} = \begin{bmatrix} F & G \\ H & F^T \end{bmatrix}$ *with* $F \in \mathbb{R}^{n\times n}$ *and skew-symmetric matrices* G, $H \in \mathbb{R}^{n\times n}$.

(iii) Every matrix $\mathcal{U} \in \mathbb{US}_{2n}$ *can be partioned as* $\mathcal{U} = \begin{bmatrix} U & V \\ -V & U \end{bmatrix}$ *with* U, $V \in \mathbb{R}^{n\times n}$.

Proof. The proof follows directly from the definition. □

Lemma 5.2 (Spectrum of a Skew-Hamiltonian/Hamiltonian Matrix Pencil)**.** *Let* $\mathcal{H} - \lambda\mathcal{N}$ *be a regular real skew-Hamiltonian/Hamiltonian matrix pencil and* $\lambda \in \Lambda(\mathcal{H},\mathcal{N})$*. Then also* $\bar{\lambda}, -\lambda, -\bar{\lambda} \in \Lambda(\mathcal{H},\mathcal{N})$ *(see [BBL+07]).*

Proof. Obviously the assertion is true if $\lambda = \infty$ or $\lambda = -\infty$.

If λ is finite then $\bar{\lambda} \in \Lambda(\mathcal{H},\mathcal{N})$ follows from the fact, that $\mathcal{H} - \lambda\mathcal{N}$ is a real matrix pencil. Let $x \neq 0$ be an eigenvector of $\mathcal{H} - \lambda\mathcal{N}$, i.e., $(\mathcal{H} - \lambda\mathcal{N})\, x = 0$. By taking the conjugate transpose and using $\mathcal{J}^T = -\mathcal{J}$ as well as

$$\mathcal{N}^H = \mathcal{N}^T = \mathcal{J}^T\mathcal{N}\mathcal{J}, \quad \mathcal{H}^H = \mathcal{H}^T = -\mathcal{J}^T\mathcal{H}\mathcal{J},$$

we obtain

$$0 = x^H \left(\mathcal{H}^H - \bar{\lambda}\mathcal{N}^H\right) = x^H \left(-\mathcal{J}^T\mathcal{H}\mathcal{J} - \bar{\lambda}\mathcal{J}^T\mathcal{N}\mathcal{J}\right) = -x^H\mathcal{J}^T\left(\mathcal{H} + \bar{\lambda}\mathcal{N}\right)\mathcal{J}.$$

This implies that $-\bar{\lambda} \in \Lambda(\mathcal{H},\mathcal{N})$. Since, again $\mathcal{H} - \lambda\mathcal{N}$ is real, $-\lambda \in \Lambda(\mathcal{H},\mathcal{N})$. □

The following gives a basis for structure-preserving transformations on skew-Hamiltonian/Hamiltonian matrix pencils.

Definition 5.2 ($\mathcal{J}$-Congruence for Matrix Pencils)**.** (see [Meh99, Meh00]) Two real matrix pencils $A - \lambda B$, $C - \lambda D \in \mathbb{R}^{2n\times 2n}$ are called *$\mathcal{J}$-congruent* if there exists a nonsingular matrix $P \in \mathbb{R}^{2n\times 2n}$ such that

$$\mathcal{J}P^T\mathcal{J}^T\left(A - \lambda B\right)P = C - \lambda D.$$

Lemma 5.3. *Let $\mathcal{H} - \lambda\mathcal{N}$ be a skew-Hamiltonian/Hamiltonian matrix pencil and $P \in \mathbb{R}^{2n\times 2n}$ be a nonsingular matrix. Then the $\mathcal{J}$-congruent matrix pencil $\mathcal{J}P^T\mathcal{J}^T(\mathcal{H} - \lambda\mathcal{N})P$ is again skew-Hamiltonian/Hamiltonian.*

Proof. The proof follows directly from the definitions (see [Meh99, Meh00]). □

5.1.2 Condensed Forms for Hamiltonian Matrices and Matrix Pencils

Many standard algorithms for the computation of the eigenvalues of general dense matrices and matrix pencils, e.g., the QR and QZ algorithm (see [GVL96]), apply othogonal transformations on the participating matrices to transform these to (generalized) Schur form. In this way the eigenvalue information can be acquired from the diagonal (and subdiagonal) entries. These algorithms are numerically backward stable and have a computational complexity of at most $\mathcal{O}\left(n^3\right)$. In this chapter we want to introduce a QZ-like method which additionally preserves the skew-Hamiltonian/Hamiltonian structure of the matrix pencils. For this purpose we have to state Schur-like forms for Hamiltonian matrices and matrix pencils (see [BBMX99]).

Definition 5.3 (Hamiltonian (Block) Triangular Form)**.** A Hamiltonian matrix $\mathcal{H}$ is called *Hamiltonian block triangular* if

$$\mathcal{H} = \begin{bmatrix} F & G \\ 0 & -F^T \end{bmatrix}.$$

If, furthermore, the matrix F is upper (quasi) triangular then we call the matrix $\mathcal{H}$ *Hamiltonian (quasi) triangular.*

Similar terms can be analogously defined for skew-Hamiltonian matrices.

Definition 5.4 (Hamiltonian (Skew-Hamiltonian) Schur Form)**.** If a real Hamiltonian (skew-Hamiltonian) matrix $\mathcal{H}$ can be transformed into Hamiltonian (skew-Hamiltonian) quasi triangular form by a similarity transformation with an orthogonal symplectic matrix $\mathcal{U} \in \mathbb{US}_{2n}$, where the eigenvalues of the 2×2 diagonal blocks are pairs of complex conjugate numbers, then we say that $\mathcal{U}^T\mathcal{H}\mathcal{U}$ is in *Hamiltonian (skew-Hamiltonian) Schur form.*

Unfortunately not every Hamiltonian matrix has a Hamiltonian Schur form, e.g., the matrix $\mathcal{J}$ from Definition 5.1 is invariant under arbitrary orthogonal symplectic similarity transformations but it is not in Hamiltonian Schur form. However, all real skew-Hamiltonian matrices have a skew-Hamiltonian Schur form. The problem of the existence of a Hamiltonian Schur form turns over as well to the case of skew-Hamiltonian/Hamiltonian matrix pencils. Luckily we can state a theorem which gives some conditions for the existence of a structured Schur form for skew-Hamiltonian/Hamiltonian matrix pencils (see [BBMX99, Meh99, Meh00]).

Theorem 5.1. *Let $\mathcal{H} - \lambda\mathcal{N}$ be a real regular skew-Hamiltonian/Hamiltonian matrix pencil with ν pairwise distinct, finite, nonzero, purely imaginary eigenvalues $\mathrm{i}\alpha_1, \mathrm{i}\alpha_2, \ldots, \mathrm{i}\alpha_\nu$ of algebraic multiplicities $p_1, p_2, \ldots, p_\nu$ and let $\mathcal{Q}_1, \mathcal{Q}_2, \ldots, \mathcal{Q}_\nu$ be their associated right deflating subspaces. Let furthermore p_∞ be the algebraic multiplicity of the eigenvalue infinity and let $\mathcal{Q}_\infty$ be its associated right deflating subspace. Then the following are equivalent:*

(i) There exists a nonsingular matrix $\mathcal{Y}$ such that

$$\mathcal{J}\mathcal{Y}^T\mathcal{J}^T(\mathcal{H} - \lambda\mathcal{N})\mathcal{Y} = \begin{bmatrix} H_{11} & H_{12} \\ 0 & -H_{11}^T \end{bmatrix} - \lambda \begin{bmatrix} N_{11} & N_{12} \\ 0 & N_{11}^T \end{bmatrix}, \tag{5.1}$$

where N_{11} is upper triangular, H_{11} is upper quasi triangular, N_{12} is skew-symmetric, H_{12} is symmetric and the eigenvalues of the 2×2 blocks on the diagonal of the subpencil $H_{11} - \lambda N_{11}$ are each a pair of complex conjugate numbers.

(ii) There exists an orthogonal matrix $\mathcal{Q}$ such that $\mathcal{J}\mathcal{Q}^T\mathcal{J}^T(\mathcal{H} - \lambda\mathcal{N})\mathcal{Q}$ is of the form of the right-hand side of (5.1). In this case the right-hand side of (5.1) is said to be in structured Schur form.

(iii) For $k = 1, 2, \ldots, \nu$, the matrix $\mathcal{Q}_k^H\mathcal{J}\mathcal{N}\mathcal{Q}_k$ is congruent to a $p_k \times p_k$ copy of $\mathcal{J}$ (If $\nu = 0$, i.e., if $\mathcal{H} - \lambda\mathcal{N}$ has no finite, nonzero, purely imaginary eigenvalues, then this statement holds vacuously.)

Furthermore, if $p_\infty \neq 0$ then $\mathcal{Q}_\infty^T\mathcal{J}\mathcal{H}\mathcal{Q}_\infty$ is congruent to a $p_\infty \times p_\infty$ copy of $P := \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix}$.

Proof. We give a sketch of the proof and emphasize the differences to the complex case as given in [Meh99, Meh00].
'(i) ⇒ (ii)': First we need the fact that if $\mathcal{Y}$ in (i) is nonsingular, we can decompose

$$\mathcal{Y} = \mathcal{Q}\begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22}^T \end{bmatrix}$$

with an orthogonal matrix $\mathcal{Q}$ and real upper triangular matrices R_{11}, R_{22}. This can be shown by applying Householder matrices in an appropriate order. Using this, we obtain from (5.1) that

$$\begin{aligned} &\mathcal{J}\mathcal{Q}^T\mathcal{J}^T(\mathcal{H} - \lambda\mathcal{N})\mathcal{Q} \\ =&\mathcal{J}\begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22}^T \end{bmatrix}^{-T}\mathcal{J}^T\left(\begin{bmatrix} H_{11} & H_{12} \\ 0 & -H_{11}^T \end{bmatrix} - \lambda\begin{bmatrix} N_{11} & N_{12} \\ 0 & N_{11}^T \end{bmatrix}\right)\begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22}^T \end{bmatrix}^{-1} \\ =&\begin{bmatrix} R_{22}^{-1}H_{11}R_{11}^{-1} & \star \\ 0 & -R_{11}^{-T}H_{11}^TR_{22}^{-T} \end{bmatrix} - \lambda\begin{bmatrix} R_{22}^{-1}N_{11}R_{11}^{-1} & \star \\ 0 & R_{11}^{-T}N_{11}^TR_{22}^{-T} \end{bmatrix}, \end{aligned}$$

where $R_{22}^{-1}N_{11}R_{11}^{-1}$ is still upper triangular and $R_{22}^{-1}H_{11}R_{11}^{-1}$ is still upper quasi triangular (triangular in the complex case).
'(ii) $\Rightarrow$ (i)': is trivial.
'(i) $\Rightarrow$ (iii)': This part is like in the complex case. First, we observe that (i) implies that the algebraic multiplicity of every purely imaginary eigenvalue is even. It can be shown that every skew-Hamiltonian/Hamiltonian matrix pencil with only purely imaginary eigenvalues is (up to permutations of rows and columns) a direct sum of skew-Hamiltonian/Hamiltonian matrix pencils with only one purely imaginary eigenvalue, i.e., exists a nonsingular complex matrix $\mathcal{Y}$ such that

$$\mathcal{J}\mathcal{Y}^H\mathcal{J}^T\left(\mathcal{H}-\lambda\mathcal{N}\right)\mathcal{Y} = \left[\begin{array}{ccc|ccc} H_{11} & & & H_{1,k+1} & & \\ & \ddots & & & \ddots & \\ & & H_{kk} & & & H_{k,2k} \\ \hline H_{k+1,1} & & & H_{k+1,k+1} & & \\ & \ddots & & & \ddots & \\ & & H_{2k,k} & & & H_{2k,2k} \end{array}\right] - \lambda \left[\begin{array}{ccc|ccc} N_{11} & & & N_{1,k+1} & & \\ & \ddots & & & \ddots & \\ & & N_{kk} & & & N_{k,2k} \\ \hline N_{k+1,1} & & & N_{k+1,k+1} & & \\ & \ddots & & & \ddots & \\ & & N_{2k,k} & & & N_{2k,2k} \end{array}\right],$$

where $\begin{bmatrix} H_{i,i} & H_{i,k+i} \\ H_{k+i,i} & H_{k+i,k+i} \end{bmatrix} - \lambda \begin{bmatrix} N_{i,i} & N_{i,k+i} \\ N_{k+i,i} & N_{k+i,k+i} \end{bmatrix}$ is skew-Hamiltonian/Hamiltonian and has only one purely imaginary eigenvalue. (This can also be applied if there exist non-imaginary eigenvalues, see [Meh99, Meh00] for details.) Furthermore we may assume w.l.o.g. that the bases $\mathcal{Q}_k$ and $\mathcal{Q}_\infty$ are canonical. So, if the conditions from (iii) hold for special bases $\mathcal{Q}_k$, $\mathcal{Q}_\infty$ they hold for all bases of the deflating subspace associated with the eigenvalue $\mathrm{i}\alpha_k$ or the eigenvalue ∞, respectively. Thus, if

$$\tilde{\mathcal{H}} - \lambda\tilde{\mathcal{N}} = \begin{bmatrix} \tilde{H}_{11} & \tilde{H}_{12} \\ 0 & -\tilde{H}_{11}^H \end{bmatrix} - \lambda \begin{bmatrix} \tilde{N}_{11} & \tilde{N}_{12} \\ 0 & \tilde{N}_{11}^H \end{bmatrix}$$

is in complex structured Schur form (see [Meh99, Meh00]) and has only one purely imaginary, finite eigenvalue, it remains to show that $\mathcal{J}\tilde{\mathcal{N}}$ is congruent to $\mathcal{J}$. Since

$\tilde{N}_{12}$ is skew-Hermitian, we obtain that

$$\begin{aligned}
&\begin{bmatrix} \tilde{N}_{11}^{-1} & -\frac{1}{2}\tilde{N}_{11}^{-1}\tilde{N}_{12} \\ 0 & I \end{bmatrix}^H \mathcal{J}\tilde{\mathcal{N}} \begin{bmatrix} \tilde{N}_{11}^{-1} & -\frac{1}{2}\tilde{N}_{11}^{-1}\tilde{N}_{12} \\ 0 & I \end{bmatrix} \\
&= \begin{bmatrix} \tilde{N}_{11}^{-H} & 0 \\ -\frac{1}{2}\tilde{N}_{12}^H\tilde{N}_{11}^{-H} & I \end{bmatrix} \begin{bmatrix} 0 & \tilde{N}_{11}^H \\ -\tilde{N}_{11} & -\tilde{N}_{12} \end{bmatrix} \begin{bmatrix} \tilde{N}_{11}^{-1} & -\frac{1}{2}\tilde{N}_{11}^{-1}\tilde{N}_{12} \\ 0 & I \end{bmatrix} \\
&= \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix} = \mathcal{J},
\end{aligned}$$

i.e., $\mathcal{J}\tilde{\mathcal{N}}$ is congruent to $\mathcal{J}$. In a similar way, the result for $\tilde{\mathcal{H}}$ can be proven (with real transformations, since $\mathcal{Q}_\infty$ is a real deflating subspace).
'(iii) $\Rightarrow$ (i)': For this part we need a special condensed form for real skew-Hamiltonian/Hamiltonian matrix pencils (see Theorem 4.18 in [Meh99] or Theorem 24 in [Meh00]). That is, for every regular real skew-Hamiltonian/Hamiltonian matrix pencil $\mathcal{H} - \lambda\mathcal{N} \in \mathbb{R}^{2n\times 2n}$ there exists a nonsingular matrix $\mathcal{Y} \in \mathbb{R}^{2n\times 2n}$ and $k \in \mathbb{N}$ such that

$$\begin{aligned}
&\mathcal{J}\mathcal{Y}^T\mathcal{J}^T(\mathcal{H} - \lambda\mathcal{N})\,\mathcal{Y} \\
&\qquad = \begin{bmatrix} H_{11} & H_{12} & H_{13} & H_{14} \\ 0 & 0 & H_{14}^T & H_{24} \\ 0 & 0 & -H_{11}^T & 0 \\ 0 & H_{42} & -H_{12}^T & 0 \end{bmatrix} - \lambda \begin{bmatrix} N_{11} & N_{12} & N_{13} & N_{14} \\ 0 & 0 & -N_{14}^T & N_{24} \\ 0 & 0 & N_{11}^T & 0 \\ 0 & N_{42} & N_{12}^T & 0 \end{bmatrix}, \qquad (5.2)
\end{aligned}$$

where

1. $H_{11} - \lambda N_{11} \in \mathbb{R}^{k\times k}$ is a matrix pencil in generalized Schur form having only eigenvalues with non-negative real part.

2. The matrix pencils $H_{24} - \lambda N_{24}$, $H_{42} - \lambda N_{42} \in \mathbb{R}^{(n-k)\times(n-k)}$ are block diagonal and the blocks are either of the form

$$\begin{bmatrix} \varepsilon_1 & & \\ & \ddots & \\ & & \varepsilon_m \end{bmatrix} - \lambda \begin{bmatrix} 0 & & \\ & \ddots & \\ & & 0 \end{bmatrix} \in \mathbb{R}^{m\times m}$$

for $m \in \mathbb{N}$, where $\varepsilon_i \in \{-1, 1\}$, or of the form

$$\begin{bmatrix} \tilde{h} & 0 \\ 0 & \tilde{h} \end{bmatrix} - \lambda \begin{bmatrix} 0 & \tilde{n} \\ -\tilde{n} & 0 \end{bmatrix},$$

where $\tilde{n}$, $\tilde{h} \in \mathbb{R}\backslash\{0\}$. In particular all the eigenvalues of the skew-Hamiltonian/Hamiltonian matrix pencil

$$\begin{bmatrix} 0 & H_{24} \\ H_{42} & 0 \end{bmatrix} - \lambda \begin{bmatrix} 0 & N_{24} \\ N_{42} & 0 \end{bmatrix}$$

are nonzero and purely imaginary.

3. The spectrum of $\mathcal{H} - \lambda\mathcal{N}$ is equal to the union of the spectra of the matrix pencils

$$\begin{bmatrix} H_{11} & H_{13} \\ 0 & -H_{11}^T \end{bmatrix} - \lambda \begin{bmatrix} N_{11} & N_{13} \\ 0 & N_{11}^T \end{bmatrix}, \quad \begin{bmatrix} 0 & H_{24} \\ H_{42} & 0 \end{bmatrix} - \lambda \begin{bmatrix} 0 & N_{24} \\ N_{42} & 0 \end{bmatrix}.$$

First, we assume w.l.o.g. that $\mathcal{H} - \lambda\mathcal{N}$ has only finite, purely imaginary eigenvalues. Then w.l.o.g. the matrix pencil $\begin{bmatrix} H_{11} & H_{13} \\ 0 & -H_{11}^T \end{bmatrix} - \lambda \begin{bmatrix} N_{11} & N_{13} \\ 0 & N_{11}^T \end{bmatrix}$ contains only pairs of complex conjugate eigenvalues with even algebraic multiplicities and $\begin{bmatrix} 0 & H_{24} \\ H_{42} & 0 \end{bmatrix} - \lambda \begin{bmatrix} 0 & N_{24} \\ N_{42} & 0 \end{bmatrix}$ contains only pairs of complex conjugate eigenvalues with algebraic multiplicity one (if not, i.e., it contains eigenvalues with higher algebraic multiplicities, we can transform $\mathcal{H} - \lambda\mathcal{N}$ by orthogonal $\mathcal{J}$-congruence transformations such that this condition holds). From condition (iii) it follows that the algebraic multiplicities of the purely imaginary eigenvalues are even, in other words, in (5.2) the blocks H_{24}, H_{42}, N_{24}, N_{42} are empty and so it remains the skew-Hamiltonian/Hamiltonian matrix pencil

$$\begin{bmatrix} H_{11} & H_{13} \\ 0 & -H_{11}^T \end{bmatrix} - \lambda \begin{bmatrix} N_{11} & N_{13} \\ 0 & N_{11}^T \end{bmatrix}$$

which is in real structured Schur form (5.1).

Now we assume w.l.o.g. that $\mathcal{H} - \lambda\mathcal{N}$ contains only infinite eigenvalues. Then the blocks N_{24}, N_{42} in (5.2) are zero and some other blocks vanish, and it remains the skew-Hamiltonian/Hamiltonian matrix pencil

$$\begin{bmatrix} 0 & H_{24} \\ H_{42} & 0 \end{bmatrix} - \lambda \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}. \tag{5.3}$$

Since $\mathcal{J}\mathcal{H}$ is congruent to P as defined in (iii), $\mathcal{H}$ is $\mathcal{J}$-congruent to $\begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}$ which proves that the matrix pencil (5.3) can be put into real structured Schur form (5.1). □

Shortly, there only exists a structured Schur form if the algebraic multiplicities of the finite, purely imaginary eigenvalues are even. However, we can compute the eigenvalues of skew-Hamiltonian/Hamiltonian matrix pencils in a different way, without requiring a structured Schur form. This is explained in the next section.

5.2 Computing the Eigenvalues

Here we derive a structure-preserving method for computing the eigenvalues of a skew-Hamiltonian/Hamiltonian matrix pencil $\mathcal{H} - \lambda\mathcal{N}$. As main reference serves [BBL+07]. In contrast to this publication we immediately derive the method without referring to any Cholesky-like factorization of the matrix $\mathcal{N}$. Additionally we simplify the computation of the eigenvalues such that we do not compute a structured Schur form anymore.

5.2.1 Embedded Matrix Pencils and an Associated Condensed Form

The algorithm is based on the following structural property of the involved matrix pencils.

Theorem 5.2. *Let $\mathcal{H} - \lambda\mathcal{N}$ be a real skew-Hamiltonian/Hamiltonian matrix pencil. Then there exist orthogonal matrices $\mathcal{Q}_1$, $\mathcal{Q}_2$ such that*

$$\begin{aligned} \mathcal{Q}_1^T \mathcal{N} \left(\mathcal{J}\mathcal{Q}_1\mathcal{J}^T\right) &= \begin{bmatrix} N_1 & N_2 \\ 0 & N_1^T \end{bmatrix}, \\ \left(\mathcal{J}\mathcal{Q}_2\mathcal{J}^T\right)^T \mathcal{N}\mathcal{Q}_2 &= \begin{bmatrix} M_1 & M_2 \\ 0 & M_1^T \end{bmatrix} := \mathcal{M}, \\ \mathcal{Q}_1^T \mathcal{H}\mathcal{Q}_2 &= \begin{bmatrix} H_{11} & H_{12} \\ 0 & H_{22} \end{bmatrix}, \end{aligned} \tag{5.4}$$

where N_1, M_1, H_{11} are upper triangular, H_{22}^T is upper quasi triangular, and N_2, M_2 are skew-symmetric.

Proof. In [BBL+07] this theorem is proven by using a Cholesky-like decomposition of the matrix $\mathcal{N}$. Algorithm 5.1 also gives a constructive proof for the specially structured skew-Hamiltonian/Hamiltonian matrix pencils which occur in the $\mathcal{L}_\infty$-norm algorithm. □

In the sequel, the matrices A, B, D, E are not the system matrices of a descriptor system. Here, they are square matrices that fit to the corresponding matrix structures. We first assume that the matrix pencil $\mathcal{H} - \lambda\mathcal{N}$ has the block structure

$$\mathcal{H} - \lambda\mathcal{N} = \begin{bmatrix} B & F \\ G & -B^T \end{bmatrix} - \lambda \begin{bmatrix} A & D \\ E & A^T \end{bmatrix}.$$

Then we define the double sized matrices

$$\mathcal{B}_\mathcal{N} := \begin{bmatrix} \mathcal{N} & 0 \\ 0 & \mathcal{N} \end{bmatrix}, \quad \mathcal{B}_\mathcal{H} := \begin{bmatrix} \mathcal{H} & 0 \\ 0 & -\mathcal{H} \end{bmatrix}.$$

We introduce the real orthogonal matrices

$$\mathcal{Y} = \frac{\sqrt{2}}{2}\begin{bmatrix} I_{2n} & I_{2n} \\ -I_{2n} & I_{2n} \end{bmatrix}, \quad \mathcal{P} = \begin{bmatrix} I_n & 0 & 0 & 0 \\ 0 & 0 & I_n & 0 \\ 0 & I_n & 0 & 0 \\ 0 & 0 & 0 & I_n \end{bmatrix}.$$

Then we set

$$\begin{aligned} \hat{\mathcal{B}}_{\mathcal{N}} &:= \mathcal{Y}^T \mathcal{B}_{\mathcal{N}} \mathcal{Y} = \mathcal{B}_{\mathcal{N}}, \\ \hat{\mathcal{B}}_{\mathcal{H}} &:= \mathcal{Y}^T \mathcal{B}_{\mathcal{H}} \mathcal{Y} = \begin{bmatrix} 0 & \mathcal{H} \\ \mathcal{H} & 0 \end{bmatrix}, \end{aligned} \tag{5.5}$$

and

$$\begin{aligned} \tilde{\mathcal{B}}_{\mathcal{N}} &:= \mathcal{P}^T \hat{\mathcal{B}}_{\mathcal{N}} \mathcal{P} = \mathcal{X}^T \mathcal{B}_{\mathcal{N}} \mathcal{X} = \left[\begin{array}{cc|cc} A & 0 & D & 0 \\ 0 & A & 0 & D \\ \hline E & 0 & A^T & 0 \\ 0 & E & 0 & A^T \end{array}\right], \\ \tilde{\mathcal{B}}_{\mathcal{H}} &:= \mathcal{P}^T \hat{\mathcal{B}}_{\mathcal{H}} \mathcal{P} = \mathcal{X}^T \mathcal{B}_{\mathcal{H}} \mathcal{X} = \left[\begin{array}{cc|cc} 0 & B & 0 & F \\ B & 0 & F & 0 \\ \hline 0 & G & 0 & -B^T \\ G & 0 & -B^T & 0 \end{array}\right], \end{aligned} \tag{5.6}$$

where $\mathcal{X} = \mathcal{Y}\mathcal{P}$. The $4n \times 4n$ matrix pencil $\tilde{\mathcal{B}}_{\mathcal{H}} - \lambda\tilde{\mathcal{B}}_{\mathcal{N}}$ is again a real skew-Hamiltonian/Hamiltonian matrix pencil.

Using the decomposition (5.4) it can be easily verified that

$$\left(\mathcal{J}\mathcal{Q}_2^T\mathcal{J}^T\right)\mathcal{H}\left(\mathcal{J}\mathcal{Q}_1\mathcal{J}^T\right) = \begin{bmatrix} -H_{22}^T & H_{12}^T \\ 0 & -H_{11}^T \end{bmatrix}. \tag{5.7}$$

Combining the equation (5.7) with the last equation of (5.5), we obtain

$$\begin{bmatrix} \mathcal{Q}_1 & 0 \\ 0 & \mathcal{J}\mathcal{Q}_2\mathcal{J}^T \end{bmatrix}^T \hat{\mathcal{B}}_{\mathcal{H}} \begin{bmatrix} \mathcal{J}\mathcal{Q}_1\mathcal{J}^T & 0 \\ 0 & \mathcal{Q}_2 \end{bmatrix} = \left[\begin{array}{cc|cc} 0 & 0 & H_{11} & H_{12} \\ 0 & 0 & 0 & H_{22} \\ \hline -H_{22}^T & H_{12}^T & 0 & 0 \\ 0 & -H_{11}^T & 0 & 0 \end{array}\right].$$

Applying the same transformations to the matrix $\hat{\mathcal{B}}_{\mathcal{N}}$ and using the decompositions

from (5.4) yields

$$\begin{aligned}\begin{bmatrix} \mathcal{Q}_1 & 0 \\ 0 & \mathcal{J}\mathcal{Q}_2\mathcal{J}^T \end{bmatrix}^T \hat{\mathcal{B}}_{\mathcal{N}} \begin{bmatrix} \mathcal{J}\mathcal{Q}_1\mathcal{J}^T & 0 \\ 0 & \mathcal{Q}_2 \end{bmatrix} &= \begin{bmatrix} \mathcal{Q}_1^T\mathcal{N}\mathcal{J}\mathcal{Q}_1\mathcal{J}^T & 0 \\ 0 & \mathcal{J}\mathcal{Q}_2^T\mathcal{J}^T\mathcal{N}\mathcal{Q}_2 \end{bmatrix} \\ &= \left[\begin{array}{cc|cc} N_1 & N_2 & 0 & 0 \\ 0 & N_1^T & 0 & 0 \\ \hline 0 & 0 & M_1 & M_2 \\ 0 & 0 & 0 & M_1^T \end{array}\right]. \end{aligned} \tag{5.8}$$

To understand the next transformation, we need the following little technical lemma.

Lemma 5.4. *The following equation holds:*

$$\mathcal{J}_{4n}\mathcal{P}\begin{bmatrix} \mathcal{J}_{2n}^T & 0 \\ 0 & I_{2n} \end{bmatrix} = \mathcal{P}^T\begin{bmatrix} I_{2n} & 0 \\ 0 & \mathcal{J}_{2n} \end{bmatrix}.$$

Proof. By just calculating both products, we obtain

$$\mathcal{J}_{4n}\mathcal{P}\begin{bmatrix} \mathcal{J}_{2n}^T & 0 \\ 0 & I_{2n} \end{bmatrix} = \begin{bmatrix} I_n & 0 & 0 & 0 \\ 0 & 0 & 0 & I_n \\ 0 & I_n & 0 & 0 \\ 0 & 0 & -I_n & 0 \end{bmatrix} = \mathcal{P}^T\begin{bmatrix} I_{2n} & 0 \\ 0 & \mathcal{J}_{2n} \end{bmatrix}.$$

□

Now we set

$$\tilde{\mathcal{Q}} = \mathcal{P}^T\begin{bmatrix} \mathcal{J}\mathcal{Q}_1\mathcal{J}^T & 0 \\ 0 & \mathcal{Q}_2 \end{bmatrix}\mathcal{P}.$$

By using Lemma 5.4 we obtain

$$\begin{aligned} \mathcal{J}\tilde{\mathcal{Q}}^T\mathcal{J}^T &= \mathcal{J}\mathcal{P}\begin{bmatrix} \mathcal{J}^T\mathcal{Q}_1^T\mathcal{J} & 0 \\ 0 & \mathcal{Q}_2^T \end{bmatrix}\mathcal{P}^T\mathcal{J}^T \\ &= \mathcal{J}\mathcal{P}\begin{bmatrix} \mathcal{J}^T & 0 \\ 0 & I \end{bmatrix}\begin{bmatrix} \mathcal{Q}_1^T & 0 \\ 0 & \mathcal{Q}_2^T \end{bmatrix}\begin{bmatrix} \mathcal{J} & 0 \\ 0 & I \end{bmatrix}\mathcal{P}^T\mathcal{J}^T \\ &= \mathcal{P}^T\begin{bmatrix} I & 0 \\ 0 & \mathcal{J} \end{bmatrix}\begin{bmatrix} \mathcal{Q}_1^T & 0 \\ 0 & \mathcal{Q}_2^T \end{bmatrix}\begin{bmatrix} I & 0 \\ 0 & \mathcal{J}^T \end{bmatrix}\mathcal{P} \\ &= \mathcal{P}^T\begin{bmatrix} \mathcal{Q}_1^T & 0 \\ 0 & \mathcal{J}\mathcal{Q}_2^T\mathcal{J}^T \end{bmatrix}\mathcal{P}. \end{aligned}$$

Then we can perform the orthogonal $\mathcal{J}$-congruence transformation

$$\begin{aligned}
\mathcal{J}\tilde{\mathcal{Q}}^T\mathcal{J}^T\tilde{\mathcal{B}}_{\mathcal{N}}\tilde{\mathcal{Q}} &= \mathcal{P}^T\begin{bmatrix}\mathcal{Q}_1^T & 0\\ 0 & \mathcal{J}\mathcal{Q}_2^T\mathcal{J}^T\end{bmatrix}\mathcal{P}\tilde{\mathcal{B}}_{\mathcal{N}}\mathcal{P}^T\begin{bmatrix}\mathcal{J}\mathcal{Q}_1\mathcal{J}^T & 0\\ 0 & \mathcal{Q}_2\end{bmatrix}\mathcal{P}\\
&= \mathcal{P}^T\begin{bmatrix}\mathcal{Q}_1^T & 0\\ 0 & \mathcal{J}\mathcal{Q}_2^T\mathcal{J}^T\end{bmatrix}\hat{\mathcal{B}}_{\mathcal{N}}\begin{bmatrix}\mathcal{J}\mathcal{Q}_1\mathcal{J}^T & 0\\ 0 & \mathcal{Q}_2\end{bmatrix}\mathcal{P}\\
&= \mathcal{P}^T\left[\begin{array}{cc|cc} N_1 & N_2 & 0 & 0\\ 0 & N_1^T & 0 & 0\\ \hline 0 & 0 & M_1 & M_2\\ 0 & 0 & 0 & M_1^T\end{array}\right]\mathcal{P}\\
&= \left[\begin{array}{cc|cc} N_1 & 0 & N_2 & 0\\ 0 & M_1 & 0 & M_2\\ \hline 0 & 0 & N_1^T & 0\\ 0 & 0 & 0 & M_1^T\end{array}\right],
\end{aligned} \tag{5.9}$$

by subsequently applying (5.6) and (5.8). An analogous argument yields

$$\mathcal{J}\tilde{\mathcal{Q}}^T\mathcal{J}^T\tilde{\mathcal{B}}_{\mathcal{H}}\tilde{\mathcal{Q}} = \left[\begin{array}{cc|cc} 0 & H_{11} & 0 & H_{12}\\ -H_{22}^T & 0 & H_{12}^T & 0\\ \hline 0 & 0 & 0 & H_{22}\\ 0 & 0 & -H_{11}^T & 0\end{array}\right]. \tag{5.10}$$

The block structure of the right-hand sides of (5.9) and (5.10) can be used to determine the eigenvalues of $\mathcal{H}-\lambda\mathcal{N}$ which is topic of the next subsection.

5.2.2 Extraction of the Eigenvalue Information

Spectral Properties of the Embedded Matrix Pencils

First, we summarize some statements concerning the spectrum of the embedded matrix pencils. On the one hand, we have

$$\Lambda\left(\mathcal{J}\tilde{\mathcal{Q}}^T\mathcal{J}^T\tilde{\mathcal{B}}_{\mathcal{H}}\tilde{\mathcal{Q}}, \mathcal{J}\tilde{\mathcal{Q}}^T\mathcal{J}^T\tilde{\mathcal{B}}_{\mathcal{N}}\tilde{\mathcal{Q}}\right) = \Lambda\left(\tilde{\mathcal{B}}_{\mathcal{H}}, \tilde{\mathcal{B}}_{\mathcal{N}}\right) = \Lambda\left(\mathcal{B}_{\mathcal{H}}, \mathcal{B}_{\mathcal{N}}\right),$$

since all these matrix pencils are orthogonally equivalent. As $\mathcal{B}_{\mathcal{H}}-\lambda\mathcal{B}_{\mathcal{N}}$ has a special block structure, we obtain

$$\Lambda\left(\mathcal{B}_{\mathcal{H}}, \mathcal{B}_{\mathcal{N}}\right) = \Lambda\left(\mathcal{H}, \mathcal{N}\right) \cup \Lambda\left(-\mathcal{H}, \mathcal{N}\right).$$

Since $\mathcal{H}-\lambda\mathcal{N}$ is a skew-Hamiltonian/Hamiltonian matrix pencil, its spectrum is symmetric with respect to the imaginary axis, following from Lemma 5.2. Hence,

$$\Lambda\left(\mathcal{H}, \mathcal{N}\right) = \Lambda\left(-\mathcal{H}, \mathcal{N}\right) \tag{5.11}$$

and every eigenvalue of $\mathcal{B}_\mathcal{H} - \lambda\mathcal{B}_\mathcal{N}$ has even algebraic multiplicity. On the other hand, the block structure of (5.9) and (5.10) yields

$$\Lambda\left(\mathcal{B}_\mathcal{H}, \mathcal{B}_\mathcal{N}\right) = \Lambda\left(\begin{bmatrix} 0 & H_{11} \\ -H_{22}^T & 0 \end{bmatrix}, \begin{bmatrix} N_1 & 0 \\ 0 & M_1 \end{bmatrix}\right) \cup \Lambda\left(\begin{bmatrix} 0 & H_{22} \\ -H_{11}^T & 0 \end{bmatrix}, \begin{bmatrix} N_1^T & 0 \\ 0 & M_1^T \end{bmatrix}\right). \tag{5.12}$$

We analyse formula (5.12) in more detail after we have introduced some concepts in the next paragraph.

Generalized Matrix Pencils and Periodic Schur Decomposition

In this paragraph we describe a generalization of matrix pencils of the form $A - \lambda E$ which is mainly based on [BMX02, Kre01b]. Here we use the notation of [Kre01b] in this work. Other good references are, e.g., [LVDX98, LS01]. We want to consider general matrix products of the form

$$\prod_{i=1}^{k} A_i^{s_i} := A_1^{s_1} A_2^{s_2} \cdots A_k^{s_k}, \tag{5.13}$$

where $s_i \in \{-1, 1\}$. In our considerations we also allow the matrices A_i to be singular, even if $s_i = -1$. In this case the generalized matrix product (5.13) might not exist, so it is more appropriate to keep up with the notion of matrix pencils.

Definition 5.5 (k-Matrix Pencil)**.** A *k-matrix pencil* $(\mathcal{A}, s)$ is defined by

(i) a k-tuple $\mathcal{A}$ of $n \times n$ matrices $(A_1, A_2, \ldots, A_k)$ and

(ii) a signature tuple $s = (s_1, s_2, \ldots, s_k) \in \{-1, 1\}^k$.

Definition 5.6 (Real Periodic Schur Decomposition)**.** A *real periodic Schur decomposition* of a k-matrix pencil $(\mathcal{A}, s)$ is given via the application of a k-tuple $\mathcal{Q} = (Q_1, Q_2, \ldots, Q_k)$ of transformation matrices such that the matrices

$$R_i = \begin{cases} Q_i^T A_i Q_{(i \bmod k)+1}, & \text{for } s_i = 1, \\ Q_{(i \bmod k)+1}^T A_i Q_i, & \text{otherwise,} \end{cases} \tag{5.14}$$

are all simultaneously upper triangular except one which may be upper quasi triangular. To simplify the notation, we write $(\mathcal{R}, s) = \mathcal{Q}\left(\mathcal{A}, s\right)$. The k-matrix pencil $(\mathcal{R}, s)$ is said to be in *real periodic Schur form*.

Remark 5.1. The framework above also fits for matrices and ordinary matrix pencils of the form $A - \lambda E$.

Theorem 5.3 (Existence of a Real Periodic Schur Form)**.** *For every real k-matrix pencil $(\mathcal{A}, s)$ there exists a k-tuple $\mathcal{Q}$ of orthogonal matrices such that $\mathcal{Q}(\mathcal{A}, s)$ is in real periodic Schur form (5.14).*

Proof. See, e.g., [Kre01b]. □

Now, since we know what a periodic Schur form for k-matrix pencils is, we can define what regular and singular k-matrix pencils and eigenvalues thereof are (modification of the definition in [Kre01b]).

Definition 5.7 (Regular/Singular k-Matrix Pencil)**.** Let $(\mathcal{R}, s) = \mathcal{Q}(\mathcal{A}, s)$ be in real periodic Schur form and let R_l be the upper quasi triangular matrix of $\mathcal{R}$ (If there is no quasi triangular matrix we can choose an arbitrary one.). Let

$$R_l = \begin{bmatrix} R_{11;l} & R_{12;l} & \cdots & R_{1m;l} \\ & R_{22;l} & & \vdots \\ & & \ddots & \vdots \\ & & & R_{mm;l} \end{bmatrix} \tag{5.15}$$

be partioned such that $R_{jj;l}$, $j = 1, \ldots, m$, are each either 1×1 blocks or 2×2 blocks with nonzero subdiagonal elements. Furthermore, let the other matrices R_i, $i = 1, \ldots, k$, $k \neq l$ be partioned as in (5.15). If there exists no integer j such that the product $\prod_{i=1}^{k} R_{jj;i}^{s_i}$ becomes undefined, we call $(\mathcal{A}, s)$ *regular*, otherwise *singular*.

Definition 5.8 (Eigenvalues of a k-Matrix Pencil)**.** Let $(\mathcal{R}, s) = \mathcal{Q}(\mathcal{A}, s)$ be a regular k-matrix pencil in real periodic Schur form as given in Definition 5.7.

(i) If all $R_{jj;i}$ corresponding to $s_i = -1$ are nonsingular we call the eigenvalues of $\Lambda_j := \prod_{i=1}^{k} R_{jj;i}^{s_i}$ *finite eigenvalues* of $(\mathcal{A}, s)$.

(ii) If there exists a singular $R_{jj;i}$ corresponding to $s_i = -1$ then $(\mathcal{A}, s)$ has *infinite eigenvalues*.

In the sequel we also need the following property of generalized matrix products (modification of the corresponding lemma in [Kre01b]).

Lemma 5.5 (Periodicity)**.** *Let $(\mathcal{A}, s)$ be a regular k-matrix pencil and $\mathcal{Q}$ such that $\mathcal{Q}(\mathcal{A}, s)$ is in real periodic Schur form. Then all the generalized matrix products*

$$Q_j^T A_j^{s_j} A_{j+1}^{s_{j+1}} \cdots A_k^{s_k} A_1^{s_1} \cdots A_{j-1}^{s_{j-1}} Q_j, \quad j = 1, \ldots, k,$$

are in upper quasi triangular form and have the same eigenvalues.

The periodic eigenvalue problem may also be studied via an inflated generalized eigenvalue problem [LVDX98, LS01, Kre01b]

$$\mathcal{B} - \lambda\mathcal{C} = \begin{bmatrix} B_1 & & & \\ & B_2 & & \\ & & \ddots & \\ & & & B_k \end{bmatrix} - \lambda \begin{bmatrix} & & & C_1 \\ C_2 & & & \\ & \ddots & & \\ & & C_k & \end{bmatrix}, \tag{5.16}$$

where

$$(B_i, C_i) = \begin{cases} (I_n, A_i), & \text{if } s_i = 1, \\ (A_i, I_n), & \text{if } s_i = -1. \end{cases}$$

Then for the eigenvalues of the inflated matrix pencil (5.16) we have the following result (see [LS01]).

Theorem 5.4 (Eigenvalues of the Inflated Matrix Pencil)**.** *Let $(\mathcal{A}, s)$ be a regular k-matrix pencil of $n \times n$ matrices with r eigenvalues λ_i, $i = 1, \ldots, r$. Then the generalized eigenvalue problem $\mathcal{B} - \lambda\mathcal{C}$ as defined in (5.16) has rk eigenvalues μ_{ij} where $i = 1, \ldots, r$ and $j = 1, \ldots, k$ so that*

1. *μ_{ij} are the k-th roots of λ_i if λ_i is finite, and*
2. *$\mu_{ij} = \infty$ if $\lambda_i = \infty$.*

Remark 5.2. Theorem 5.4 remains true, if we consider an inflated matrix pencil of the form (5.16) where $B_i - \lambda C_i$, $i = 1, \ldots, k$ are arbitrary regular $n \times n$ matrix pencils. Then the eigenvalues of $\mathcal{B} - \lambda\mathcal{C}$ are related to the eigenvalues of the $2k$-matrix pencil $(\{C_1, B_1, C_2, B_2, \ldots, C_k, B_k\}, \{-1, 1, -1, 1, \ldots, -1, 1\})$, see [BGVD92].

Application to Our Problem

We consider again the spectra (5.12). First we have a look at the matrix pencil

$$H - \lambda N = \begin{bmatrix} 0 & H_{11} \\ -H_{22}^T & 0 \end{bmatrix} - \lambda \begin{bmatrix} N_1 & 0 \\ 0 & M_1 \end{bmatrix}.$$

Turning over to the inverse eigenvalue problem, i.e., the problem for $N - \lambda H$ yields matrix structures as in (5.16). By applying Theorem 5.4 and the following remark we obtain

$$\Lambda(N, H) = \pm\sqrt{\Lambda\left(-H_{11}^{-1} N_1 H_{22}^{-T} M_1\right)}.$$

Then, by the spectral mapping theorem and subsequently applying Lemma 5.5 we obtain

$$\Lambda(H, N) = \pm\sqrt{\Lambda\left(-M_1^{-1} H_{22}^T N_1^{-1} H_{11}\right)} = \pm\mathrm{i}\sqrt{\Lambda\left(N_1^{-1} H_{11} M_1^{-1} H_{22}^T\right)}.$$

By similar consideration we also get

$$\begin{aligned}\Lambda\left(\begin{bmatrix} 0 & H_{22} \\ -H_{11}^T & 0 \end{bmatrix}, \begin{bmatrix} N_1^T & 0 \\ 0 & M_1^T \end{bmatrix}\right) &= \pm \mathrm{i}\sqrt{\Lambda\left(N_1^{-T} H_{22} M_1^{-T} H_{11}^T\right)} \\ &= \pm \mathrm{i}\sqrt{\Lambda\left(H_{11} M_1^{-1} H_{22}^T N_1^{-1}\right)} \\ &= \pm \mathrm{i}\sqrt{\Lambda\left(N_1^{-1} H_{11} M_1^{-1} H_{22}^T\right)} \\ &= \Lambda\left(\begin{bmatrix} 0 & H_{11} \\ -H_{22}^T & 0 \end{bmatrix}, \begin{bmatrix} N_1 & 0 \\ 0 & M_1 \end{bmatrix}\right).\end{aligned}$$

As a consequence, by using (5.11) we obtain the relation

$$\Lambda(\mathcal{H}, \mathcal{N}) = \Lambda\left(\begin{bmatrix} 0 & H_{11} \\ -H_{22}^T & 0 \end{bmatrix}, \begin{bmatrix} N_1 & 0 \\ 0 & M_1 \end{bmatrix}\right) = \pm \mathrm{i}\sqrt{\Lambda\left(N_1^{-1} H_{11} M_1^{-1} H_{22}^T\right)}$$

which can be easily determined as N_1, M_1, H_{11} are upper triangular and H_{22}^T is upper quasi triangular.

5.3 Algorithmic Details

In this section we describe our QZ-like algorithm for the computation of the eigenvalues of skew-Hamiltonian/Hamiltonian matrix pencils $\mathcal{H} - \lambda\mathcal{N}$ as described in [BBMX99, BBL+07]. This is done for the special matrix pencils we consider in this thesis, i.e., those where the skew-symmetric blocks of $\mathcal{N}$ are zero. So our method differs slightly from the one described in the given references, since we do not have to annihilate the skew-symmetric parts of $\mathcal{N}$. First we need some elementary orthogonal transformation matrices in order to manipulate the matrix pencils.

For $m = 2n$ we denote an $m \times m$ real Givens rotation matrix by

$$G(i, j, \theta) = \begin{bmatrix} I_{i-1} & & & & \\ & \cos(\theta) & & \sin(\theta) & \\ & & I_{j-i-1} & & \\ & -\sin(\theta) & & \cos(\theta) & \\ & & & & I_{m-j} \end{bmatrix},$$

where $1 \leqslant i < j \leqslant m$, $\theta \in [0, 2\pi)$. If $j = n+i$, then the rotation is real orthogonal and symplectic. In this case we drop the second argument j and use the two argument notation

$$G_s(i, \theta) := G(i, n + i, \theta).$$

Furthermore, for $0 \neq w \in \mathbb{R}^k$, we denote an $n \times n$ Householder matrix ($n \geqslant k$) by

$$H(k, w) = I_n - 2\frac{\tilde{w}\tilde{w}^T}{\tilde{w}^T\tilde{w}}, \quad \tilde{w} = \begin{bmatrix} 0 & w^T \end{bmatrix}^T,$$

where $\tilde{w}$ is obtained from w by prepending $n-k$ zeros. If $w = 0$, then we take $H(k,0) = I_n$. For the numerically stable computation of these matrices we refer to the methods described in [GVL96].

Algorithm 5.1: Eigenvalue Computation Method

Input: Real skew-Hamiltonian/Hamiltonian matrix pencil with special block structure: $\mathcal{H} - \lambda\mathcal{N} = \begin{bmatrix} B & F \\ G & -B^T \end{bmatrix} - \lambda \begin{bmatrix} A & 0 \\ 0 & A^T \end{bmatrix}$.

Output: The spectrum $\Lambda(\mathcal{H},\mathcal{N})$, optionally the decomposition (5.4) and optionally the corresponding orthogonal transformation matrices $\mathcal{Q}_1$ and $\mathcal{Q}_2$.

Step 0: Optionally set $\mathcal{Q}_1 = I_{2n}$.
Step 1: *% Reduce $\mathcal{N}$ to skew-Hamiltonian triangular form.*
for $k = 1, \ldots, n-1$ **do**
  Determine $H(n-k+1, y)$ to eliminate $\mathcal{N}(k+1 : n, k)$ (as well as $\mathcal{N}(n+k, n+k+1 : 2n)$) from the left.
  Optionally set $\tilde{\mathcal{Q}} = \mathrm{diag}(H(n-k+1, y), I_n)$.
  Update $\mathcal{N} := \tilde{\mathcal{Q}}^T\mathcal{N}\mathcal{J}\tilde{\mathcal{Q}}\mathcal{J}^T$, $\mathcal{H} := \tilde{\mathcal{Q}}^T\mathcal{H}\mathcal{J}\tilde{\mathcal{Q}}\mathcal{J}^T$, optionally set $\mathcal{Q}_1 := \mathcal{Q}_1\tilde{\mathcal{Q}}$.
end for
Set $\mathcal{M} := \mathcal{N}$, optionally set $\mathcal{Q}_2 := \mathcal{J}\mathcal{Q}_1\mathcal{J}^T$.
Step 2: *% Eliminations in $\mathcal{H}$.*
for $k = 1, \ldots, n$ **do**
  % I. Annihilate $\mathcal{H}(n+k : 2n-1, k)$.
  for $j = 1, \ldots, n-1$ **do**
    a) Use $G(n+j, n+j+1, \theta_1)$ to eliminate $\mathcal{H}_{n+j,k}$ from the left.
      Set $\mathcal{H} := G(n+j, n+j+1, \theta_1)^T\mathcal{H}$.
      Set $\mathcal{N} := G(n+j, n+j+1, \theta_1)^T\mathcal{N}\mathcal{J}G(n+j, n+j+1, \theta_1)\mathcal{J}^T$.
      Optionally set $\mathcal{Q}_1 := \mathcal{Q}_1 G(n+j, n+j+1, \theta_1)$.
    b) Use $G(j, j+1, \theta_2)$ to eliminate $\mathcal{N}_{j+1,j}$ (as well as $\mathcal{N}_{n+j,n+j+1}$) from the left.
      $\mathcal{N} := G(j, j+1, \theta_2)^T\mathcal{N}\mathcal{J}G(j, j+1, \theta_2)\mathcal{J}^T$.
      $\mathcal{H} := G(j, j+1, \theta_2)^T\mathcal{H}$.
      Optionally set $\mathcal{Q}_1 := \mathcal{Q}_1 G(j, j+1, \theta_2)$.
  end for
  % II. Annihilate $\mathcal{H}_{2n,k}$.
  Use $G_s(n, \phi_1)$ to eliminate $\mathcal{H}_{2n,k}$ from the left.
    Set $\mathcal{N} := G_s(n, \phi_1)^T\mathcal{N}G_s(n, \phi_1)$.
    Set $\mathcal{H} := G_s(n, \phi_1)^T\mathcal{H}$.
    Optionally set $\mathcal{Q}_1 := \mathcal{Q}_1 G_s(n, \phi_1)$.
  % III. Annihilate $\mathcal{H}(k+1 : n, k)$.
  for $j = n, n-1, \ldots, k+1$ **do**
    a) Use $G(j-1, j, \psi_1)$ to eliminate $H_{j,k}$ from the left.

30: Set $\mathcal{N} := G(j-1, j, \psi_1)^T \mathcal{N} \mathcal{J} G(j-1, j, \psi_1) \mathcal{J}^T$.
31: Set $\mathcal{H} := G(j-1, j, \psi_1)^T \mathcal{H}$.
32: Optionally set $\mathcal{Q}_1 := \mathcal{Q}_1 G(j-1, j, \psi_1)$.
33: b) Use $G(n+j-1, n+j, \psi_2)$ to eliminate $N_{n+j-1,n+j}$ (as well as $N_{j,j-1}$) from the left.
34: Set $\mathcal{N} := G(n+j-1, n+j, \psi_2)^T \mathcal{N} \mathcal{J} G(n+j-1, n+j, \psi_2) \mathcal{J}^T$.
35: Set $\mathcal{H} := G(n+j-1, n+j, \psi_2)^T \mathcal{H}$.
36: Optionally set $\mathcal{Q}_1 := \mathcal{Q}_1 G(n+j-1, n+j, \psi_2)$.
37: **end for**
38: *% IV. Annihilate* $\mathcal{H}(n+k, k+1 : n-1)$.
39: **for** $j = k+1, \ldots, n-1$ **do**
40: a) Use $G(j, j+1, \xi_1)$ to eliminate $\mathcal{H}_{n+k,j}$ from the right.
41: Set $\mathcal{H} := \mathcal{H} G(j, j+1, \xi_1)$.
42: Set $\mathcal{M} := \mathcal{J} G(j, j+1, \xi_1)^T \mathcal{J}^T \mathcal{M} G(j, j+1, \xi_1)$.
43: Optionally set $\mathcal{Q}_2 := \mathcal{Q}_2 G(j, j+1, \xi_1)$.
44: b) Use $G(n+j, n+j+1, \xi_2)$ to eliminate $\mathcal{M}_{n+j,n+j+1}$ (as well as $\mathcal{M}_{j+1,j}$) from the right.
45: Set $\mathcal{M} := \mathcal{J} G(n+j, n+j+1, \xi_2)^T \mathcal{J}^T \mathcal{M} G(n+j, n+j+1, \xi_2)$.
46: Set $\mathcal{H} := \mathcal{H} G(n+j, n+j+1, \xi_2)$.
47: Optionally set $\mathcal{Q}_2 := \mathcal{Q}_2 G(n+j, n+j+1, \xi_2)$.
48: **end for**
49: *% V. Annihilate* $\mathcal{H}_{n+k,n}$.
50: **if** $k \leqslant n$ **then**
51: Use $G_s(n, \eta_1)$ to eliminate $\mathcal{H}_{n+k,n}$ from the right.
52: Set $\mathcal{M} := G_s(n, \eta_1)^T \mathcal{M} G_s(n, \eta_1)$.
53: Set $\mathcal{H} := \mathcal{H} G_s(n, \eta_1)$.
54: Optionally set $\mathcal{Q}_2 := \mathcal{Q}_2 G_s(n, \eta_1)$.
55: **else**
56: Use $G_s(n, \eta_2)$ to eliminate $\mathcal{H}_{2n,n}$ from the left.
57: Set $\mathcal{N} := G_s(n, \eta_2)^T \mathcal{N} G_s(n, \eta_2)$.
58: Set $\mathcal{H} := G_s(n, \eta_2)^T \mathcal{H}$.
59: Optionally set $\mathcal{Q}_1 := \mathcal{Q}_1 G_s(n, \eta_2)$.
60: **end if**
61: *% VI. Annihilate* $\mathcal{H}(n+k, n+k+2 : 2n)$.
62: **for** $j = n, n-1, \ldots, k+2$ **do**
63: a) Use $G(n+j-1, n+j, \tau_1)$ to eliminate $\mathcal{H}_{n+k,n+j}$ from the right.
64: Set $\mathcal{M} := \mathcal{J} G(n+j-1, n+j, \tau_1)^T \mathcal{J}^T \mathcal{M} G(n+j-1, n+j, \tau_1)$.
65: Set $\mathcal{H} := \mathcal{H} G(n+j-1, n+j, \tau_1)$.
66: Optionally set $\mathcal{Q}_2 := \mathcal{Q}_2 G(n+j-1, n+j, \tau_1)$.
67: b) Use $G(j-1, j, \tau_2)$ to eliminate $M_{j,j-1}$ (as well as $\mathcal{M}_{n+j-1,n+j}$) from the right.

68: Set $\mathcal{M} := \mathcal{J}G(j-1,j,\tau_2)^T\mathcal{J}^T\mathcal{M}G(j-1,j,\tau_2)$.
69: Set $\mathcal{H} := \mathcal{H}G(j-1,j,\tau_2)$.
70: Optionally set $\mathcal{Q}_2 := \mathcal{Q}_2G(j-1,j,\tau_2)$.
71: **end for**
72: **end for**
73: *% Now, $\mathcal{M}$, $\mathcal{N}$, $\mathcal{H}$ are in block form (5.4) and H_{22}^T is upper Hessenberg.*
74: **Step 3:** *% Application of the periodic QZ algorithm (see [BGVD92, HL94] for theory and [Kre01a] for an efficient and reliable implementation).*
75: a) Apply the periodic QZ algorithm to the formal product

$$N_1^{-1}H_{11}M_1^{-1}H_{22}^T,$$

i.e., compute orthogonal matrices V_1, V_2, V_3, V_4, such that $V_1^TN_1V_3$, $V_1^TH_{11}V_4$, $V_2^TM_1V_4$ are upper triangular and $\left(V_3^TH_{22}V_2\right)^T$ is upper quasi triangular. Determine the spectrum as $\Lambda(\mathcal{H},\mathcal{N}) = \pm\mathrm{i}\sqrt{\Lambda\left(N_1^{-1}H_{11}M_1^{-1}H_{22}^T\right)}$.
76: b) Optionally set

$$\hat{\mathcal{Q}}_1 = \begin{bmatrix} V_1 & 0 \\ 0 & V_3 \end{bmatrix}, \quad \hat{\mathcal{Q}}_2 = \begin{bmatrix} V_4 & 0 \\ 0 & V_2 \end{bmatrix}.$$

77: c) Optionally update $\mathcal{H} := \hat{\mathcal{Q}}_1^T\mathcal{H}\hat{\mathcal{Q}}_2$, $\mathcal{N} := \hat{\mathcal{Q}}_1^T\mathcal{N}\mathcal{J}\hat{\mathcal{Q}}_1\mathcal{J}^T$, $\mathcal{M} := \mathcal{J}\hat{\mathcal{Q}}_2^T\mathcal{J}^T\mathcal{M}\hat{\mathcal{Q}}_2$, $\mathcal{Q}_1 := \mathcal{Q}_1\hat{\mathcal{Q}}_1$, $\mathcal{Q}_2 := \mathcal{Q}_2\hat{\mathcal{Q}}_2$.

Remark 5.3. Note, that in the algorithm for computing the $\mathcal{L}_\infty$-norm we have to transform the corresponding system pencil to generalized Schur form/Hessenberg form which is necessary to compute its eigenvalues (see also Subsection 6.1.2). In this way, the matrix A may already be in upper triangular form and so we could omit Step 1 of Algorithm 5.1. However, one might prefer to work with the original data to obtain higher accuracy. In Paragraph 6.2.2 we show how working with the transformed descriptor system affects, e.g., the infinite eigenvalues of $\mathcal{H} - \lambda\mathcal{N}$. If we work on the original data, the matrix A is generally not triangular but compared to (4.23) and (4.25) it satisfies a certain block structure. Knowing this fact it it even possible to accelerate Step 1 of Algorithm 5.1 if we operate only on the nonzero part of A. However, we would like to cover a broad range of skew-Hamiltonian/Hamiltonian matrix pencils and so we use this slightly more general version of the eigenvalue computation method.

Since this algorithm is not easy to understand from the pseudocode we give a schematic overview of the performed transformations using a 6×6 example matrix pencil. Hereby, $\star$ denotes an arbitrary entry which coincides with the given matrix structure, $\star$ denotes an element which is annihilated, and $\star$ denotes an element which is updated but not eliminated in the current step. Gray rows and columns in the

matrices indicate where the transformation matrices operate. On dark gray fields both row and column manipulations take place. However, only entries corresponding to blue and red stars are actually transformed. The other entries remain unchanged. Our initial point is a given skew-Hamiltonian/Hamiltonian matrix pencil $\mathcal{H} - \lambda\mathcal{N}$ with

$$\mathcal{N} = \left[\begin{array}{ccc|ccc} \star & \star & \star & 0 & 0 & 0 \\ \star & \star & \star & 0 & 0 & 0 \\ \star & \star & \star & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & \star & \star & \star \\ 0 & 0 & 0 & \star & \star & \star \\ 0 & 0 & 0 & \star & \star & \star \end{array}\right], \quad \mathcal{H} = \left[\begin{array}{ccc|ccc} \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \\ \hline \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \end{array}\right].$$

In **Step 1** we perform Householder reflections to make $\mathcal{N}(1:3,1:3)$ upper triangular and $\mathcal{N}(4:6,4:6)$ lower triangular, i.e., we obtain

$$k = 1: \quad \mathcal{N} = \left[\begin{array}{ccc|ccc} \star & \star & \star & 0 & 0 & 0 \\ \star & \star & \star & 0 & 0 & 0 \\ \star & \star & \star & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & \star & \star & \star \\ 0 & 0 & 0 & \star & \star & \star \\ 0 & 0 & 0 & \star & \star & \star \end{array}\right], \quad \mathcal{H} = \left[\begin{array}{ccc|ccc} \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \\ \hline \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \end{array}\right],$$

$$k = 2: \quad \mathcal{N} = \left[\begin{array}{ccc|ccc} \star & \star & \star & 0 & 0 & 0 \\ 0 & \star & \star & 0 & 0 & 0 \\ 0 & \star & \star & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & \star & 0 & 0 \\ 0 & 0 & 0 & \star & \star & \star \\ 0 & 0 & 0 & \star & \star & \star \end{array}\right], \quad \mathcal{H} = \left[\begin{array}{ccc|ccc} \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \\ \hline \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \end{array}\right].$$

Now, after setting $\mathcal{M} := \mathcal{N}$ we perform eliminations in $\mathcal{H}$. As these destroy the triangular structure of $\mathcal{N}$ or $\mathcal{M}$ we have to perform appropriate correction steps to annihilate those elements that do not fit into this structure. We illustrate **Step 2** for $k = 1$:

Sub-Step I.

$$j = 1: \ \text{(a)} \quad \mathcal{N} = \left[\begin{array}{ccc|ccc} \star & \star & \star & 0 & 0 & 0 \\ 0 & \star & \star & 0 & 0 & 0 \\ 0 & 0 & \star & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & \star & 0 & 0 \\ 0 & 0 & 0 & \star & \star & 0 \\ 0 & 0 & 0 & \star & \star & \star \end{array}\right], \quad \mathcal{H} = \left[\begin{array}{ccc|ccc} \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \\ \hline \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \end{array}\right],$$

$$\text{(b)}\quad \mathcal{N} = \left[\begin{array}{ccc|ccc} \star & \star & \star & 0 & 0 & 0 \\ \star & \star & \star & 0 & 0 & 0 \\ 0 & 0 & \star & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & \star & \star & 0 \\ 0 & 0 & 0 & \star & \star & 0 \\ 0 & 0 & 0 & \star & \star & \star \end{array}\right], \quad \mathcal{H} = \left[\begin{array}{ccc|ccc} \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \\ \hline 0 & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \end{array}\right],$$

$$j = 2:\ \text{(a)}\quad \mathcal{N} = \left[\begin{array}{ccc|ccc} \star & \star & \star & 0 & 0 & \star \\ 0 & \star & \star & 0 & 0 & \star \\ 0 & 0 & \star & \star & \star & 0 \\ \hline 0 & 0 & 0 & \star & 0 & 0 \\ 0 & 0 & 0 & \star & \star & 0 \\ 0 & 0 & 0 & \star & \star & \star \end{array}\right], \quad \mathcal{H} = \left[\begin{array}{ccc|ccc} \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \\ \hline 0 & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \end{array}\right],$$

$$\text{(b)}\quad \mathcal{N} = \left[\begin{array}{ccc|ccc} \star & \star & \star & 0 & 0 & \star \\ 0 & \star & \star & 0 & 0 & \star \\ 0 & \star & \star & \star & \star & 0 \\ \hline 0 & 0 & 0 & \star & 0 & 0 \\ 0 & 0 & 0 & \star & \star & \star \\ 0 & 0 & 0 & \star & \star & \star \end{array}\right], \quad \mathcal{H} = \left[\begin{array}{ccc|ccc} \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \\ \hline 0 & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \end{array}\right].$$

After this step, the skew-symmetric block $\mathcal{N}(1:3, 4:6)$ is still zero. However, in Sub-Step II this matrix will be filled and so in the next iteration for k this matrix is full. So we already fill this matrix with stars in order to make clearer which transformations are performed and which are not. We continue with
Sub-Step II.

$$\mathcal{N} = \left[\begin{array}{ccc|ccc} \star & \star & \star & 0 & \star & \star \\ 0 & \star & \star & \star & 0 & \star \\ 0 & 0 & \star & \star & \star & 0 \\ \hline 0 & 0 & 0 & \star & 0 & 0 \\ 0 & 0 & 0 & \star & \star & 0 \\ 0 & 0 & 0 & \star & \star & \star \end{array}\right], \quad \mathcal{H} = \left[\begin{array}{ccc|ccc} \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \\ \hline 0 & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \end{array}\right],$$

Sub-Step III.

$$j = 3:\ \text{(a)}\quad \mathcal{N} = \left[\begin{array}{ccc|ccc} \star & \star & \star & 0 & \star & \star \\ 0 & \star & \star & \star & 0 & \star \\ 0 & 0 & \star & \star & \star & 0 \\ \hline 0 & 0 & 0 & \star & 0 & 0 \\ 0 & 0 & 0 & \star & \star & 0 \\ 0 & 0 & 0 & \star & \star & \star \end{array}\right], \quad \mathcal{H} = \left[\begin{array}{ccc|ccc} \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \\ \hline 0 & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \end{array}\right],$$

$$\text{(b)} \quad \mathcal{N} = \left[\begin{array}{ccc|ccc} \star & \star & \star & 0 & \star & \star \\ 0 & \star & \star & \star & 0 & \star \\ 0 & \star & \star & \star & \star & 0 \\ \hline 0 & 0 & 0 & \star & 0 & 0 \\ 0 & 0 & 0 & \star & \star & \star \\ 0 & 0 & 0 & \star & \star & \star \end{array}\right], \quad \mathcal{H} = \left[\begin{array}{ccc|ccc} \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \\ \hline 0 & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \end{array}\right],$$

$$j = 2: \ \text{(a)} \quad \mathcal{N} = \left[\begin{array}{ccc|ccc} \star & \star & \star & 0 & \star & \star \\ 0 & \star & \star & \star & 0 & \star \\ 0 & 0 & \star & \star & \star & 0 \\ \hline 0 & 0 & 0 & \star & 0 & 0 \\ 0 & 0 & 0 & \star & \star & 0 \\ 0 & 0 & 0 & \star & \star & \star \end{array}\right], \quad \mathcal{H} = \left[\begin{array}{ccc|ccc} \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \\ \hline 0 & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \end{array}\right],$$

$$\text{(b)} \quad \mathcal{N} = \left[\begin{array}{ccc|ccc} \star & \star & \star & 0 & \star & \star \\ \star & \star & \star & \star & 0 & \star \\ 0 & 0 & \star & \star & \star & 0 \\ \hline 0 & 0 & 0 & \star & \star & 0 \\ 0 & 0 & 0 & \star & \star & 0 \\ 0 & 0 & 0 & \star & \star & \star \end{array}\right], \quad \mathcal{H} = \left[\begin{array}{ccc|ccc} \star & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \\ \hline 0 & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \end{array}\right],$$

Sub-Step IV.

$$j = 2: \ \text{(a)} \quad \mathcal{M} = \left[\begin{array}{ccc|ccc} \star & \star & \star & 0 & 0 & 0 \\ 0 & \star & \star & 0 & 0 & 0 \\ 0 & 0 & \star & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & \star & 0 & 0 \\ 0 & 0 & 0 & \star & \star & 0 \\ 0 & 0 & 0 & \star & \star & \star \end{array}\right], \quad \mathcal{H} = \left[\begin{array}{ccc|ccc} \star & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \\ \hline 0 & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \end{array}\right],$$

$$\text{(b)} \quad \mathcal{M} = \left[\begin{array}{ccc|ccc} \star & \star & \star & 0 & 0 & 0 \\ 0 & \star & \star & 0 & 0 & 0 \\ 0 & \star & \star & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & \star & 0 & 0 \\ 0 & 0 & 0 & \star & \star & \star \\ 0 & 0 & 0 & \star & \star & \star \end{array}\right], \quad \mathcal{H} = \left[\begin{array}{ccc|ccc} \star & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \\ \hline 0 & 0 & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \end{array}\right].$$

Again, as for Sub-Step II the skew-symmetric block $\mathcal{M}(1:3, 4:6)$ is only zero in this case, but is generally full for $k \geqslant 2$.

Sub-Step V.

$$\mathcal{M} = \left[\begin{array}{cccc|cc} \star & \star & \star & 0 & \star & \star \\ 0 & \star & \star & \star & 0 & 0 \\ 0 & 0 & \star & \star & 0 & 0 \\ \hline 0 & 0 & 0 & \star & 0 & 0 \\ 0 & 0 & 0 & \star & \star & 0 \\ 0 & 0 & 0 & \star & \star & \star \end{array}\right], \quad \mathcal{H} = \left[\begin{array}{cccc|cc} \star & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \\ \hline 0 & 0 & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \end{array}\right],$$

Sub-Step VI.

$$j = 3: \ \text{(a)} \quad \mathcal{M} = \left[\begin{array}{ccc|ccc} \star & \star & \star & 0 & \star & \star \\ 0 & \star & \star & \star & 0 & \star \\ 0 & 0 & \star & \star & \star & 0 \\ \hline 0 & 0 & 0 & \star & 0 & 0 \\ 0 & 0 & 0 & \star & \star & 0 \\ 0 & 0 & 0 & \star & \star & \star \end{array}\right], \quad \mathcal{H} = \left[\begin{array}{ccc|ccc} \star & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \\ \hline 0 & 0 & 0 & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \end{array}\right],$$

$$\text{(b)} \quad \mathcal{M} = \left[\begin{array}{ccc|ccc} \star & \star & \star & 0 & \star & \star \\ 0 & \star & \star & \star & 0 & \star \\ 0 & \star & \star & \star & \star & 0 \\ \hline 0 & 0 & 0 & \star & 0 & 0 \\ 0 & 0 & 0 & \star & \star & \star \\ 0 & 0 & 0 & \star & \star & \star \end{array}\right], \quad \mathcal{H} = \left[\begin{array}{ccc|ccc} \star & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \\ \hline 0 & 0 & 0 & \star & \star & 0 \\ 0 & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \end{array}\right].$$

Now $\mathcal{H}$ has the structure we require after the iteration $k = 1$. For $k = 2, 3$ we obtain

$$k = 2: \quad \mathcal{H} = \left[\begin{array}{ccc|ccc} \star & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \\ \hline 0 & 0 & 0 & \star & \star & 0 \\ 0 & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \end{array}\right], \quad k = 3: \quad \mathcal{H} = \left[\begin{array}{ccc|ccc} \star & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \\ 0 & 0 & \star & \star & \star & \star \\ \hline 0 & 0 & 0 & \star & \star & 0 \\ 0 & 0 & 0 & \star & \star & \star \\ 0 & 0 & \star & \star & \star & \star \end{array}\right],$$

i.e., $\mathcal{H}(1:3, 1:3)$ is upper triangular and $\mathcal{H}(4:6, 4:6)$ is lower Hessenberg while $\mathcal{N}$ and $\mathcal{M}$ remain in skew-Hamiltonian triangular form. In **Step 3** we use the periodic QZ algorithm to transform $\mathcal{H}(4:6, 4:6)$ to lower quasi triangular form while $\mathcal{N}(1:3, 1:3)$, $\mathcal{M}(1:3, 1:3)$, $\mathcal{H}(1:3, 1:3)$ are still upper triangular. Graphically,

$$\mathcal{N}, \mathcal{M} = \left[\begin{array}{ccc|ccc} \star & \star & \star & 0 & \star & \star \\ 0 & \star & \star & \star & 0 & \star \\ 0 & 0 & \star & \star & \star & 0 \\ \hline 0 & 0 & 0 & \star & 0 & 0 \\ 0 & 0 & 0 & \star & \star & 0 \\ 0 & 0 & 0 & \star & \star & \star \end{array}\right], \quad \mathcal{H} = \left[\begin{array}{ccc|ccc} \star & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \\ 0 & 0 & \star & \star & \star & \star \\ \hline 0 & 0 & 0 & \star & \star & 0 \\ 0 & 0 & 0 & \star & \star & \star \\ 0 & 0 & 0 & \star & \star & \star \end{array}\right].$$

6 Implementation and Numerical Tests

In this chapter we describe the implementation of the algorithms and test these on some "real-world" examples with respect to accuracy and runtime. All methods are implemented in FORTRAN in the style of `SLICOT` routines. `SLICOT` is the **S**ubroutine **L**ibrary **I**n systems and **CO**ntrol **T**heory and contains efficient and reliable implementations of the most important algorithms in systems and control (see [BMS+97]). `SLICOT` also requires the **B**asic **L**inear **A**lgebra **S**ubroutines library (`BLAS`, see [Uni01]) which contains routines for the simplest matrix and vector operations and the **L**inear **A**lgebra **PACK**age (`LAPACK`, see [ABB+99]) for more advanced algorithms in linear algebra.

Our tests run on a machine with an Intel® Pentium® 4 CPU with 3.20 GHz and 1.0 GB RAM. As operating system we use Linux 2.6.27.42-0.1-pae i686 (openSUSE 11.1 (i586)). All needed FORTRAN libraries (`LAPACK v3.2.1`, `SLICOT v5.0`) are compiled using the `gfortran` compiler and the flags `-O3 -march=native`. For efficient testing, all investigated routines are accessed from MATLAB® 2010a via mex files (see [Mat10a, Mat10b]).

6.1 Interface Description and Implementation Details

In this section we describe the interfaces and arguments of our routines as well as the most important library routines that are called.

6.1.1 Subroutine `DGEISP.F`

This subroutine can be used to check the transfer function of a descriptor system with given realization $(E; A, B, C, D)$ for properness and optionally returns the reduced system without uncontrollable and unobservable poles. Its interface is given by

```
 SUBROUTINE DGEISP( JOBSYS, JOBEIG, EQUIL, N, M, P, A, LDA, E, LDE,
$                   B, LDB, C, LDC, NR, RANKE, ISPRP, TOL, IWORK,
$                   DWORK, LDWORK, INFO )
```

The arguments have the following meanings:

- `JOBSYS`: Character that specifies whether the system $(E; A, B, C, D)$ is already in the reduced form obtained as indicated by `JOBEIG`.

- `JOBEIG`: Character that specifies if only infinite or all uncontrollable or unobservable poles should be removed. If all uncontrollable or unobservable poles should be eliminated it is assumed that after the call of `DGEISP` an $\mathcal{L}_\infty$-norm computation should be performed, otherwise the transfer function is only checked for properness.

- `EQUIL`: Character that specifies whether a balancing of the system pencil should be performed in order to make the matrices as close in norm as possible to increase reliability of the results. This is done by pre- and post-multiplying the system pencil with appropriate diagonal matrices, see also [War81].

- `N`, `M`, `P`: The number of descriptor, input and output variables, respectively.

- `A`, `E`, `B`, `C`: The arrays that contain the state, descriptor, input, and output matrices, respectively. On exit, they contain the reduced system matrices where the matrix pencil $A - \lambda E$ is in an SVD-like coordinate form

$$E = \begin{bmatrix} T & 0 \\ 0 & 0 \end{bmatrix}, \quad A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

  with an invertible triangular matrix T if all uncontrollable and unobservable poles should be removed, otherwise they contain meaningless elements.

- `LDA`, `LDE`, `LDB`, `LDC`: The leading dimensions of the arrays `A`, `E`, `B`, `C`, respectively.

- `NR`: The order of the reduced system if all uncontrollable or unobservable poles are to be removed.

- `RANKE`: The rank of the reduced matrix E if all uncontrollable and unobservable poles should be eliminated.

- `ISPRP`: Logical variable which indicates if the given system is determined as proper or improper.

- `TOL`: The tolerance which is used to determine the numerical rank of the involved matrices.

- `IWORK, DWORK`: Integer and double precision workspace, respectively.

- `LDWORK`: The length of `DWORK`.

- `INFO`: The error indicator which is negative if one of the arguments contains illegal values, otherwise it is zero.

Table 6.1: Library routine calls in DGEISP

Library	Routine	Purpose in our Routine
SLICOT	TG01AD	balancing the matrices of the system pencil
SLICOT	TG01JD	removing all uncontrollable or unobservable poles
SLICOT	TG01HX	orthogonal reduction of a descriptor system to a system with the same transfer function matrix and without uncontrollable finite poles
SLICOT	TB01XD	pertransposed dual standard state-space system
SLICOT	MA02CD	pertranspose of the central band of a square matrix
SLICOT	TG01FD	orthogonal reduction of the descriptor system to an SVD-like coordinate form

The library routines which participate in the computations are listed in Table 6.1. Note that if we only want to remove infinite uncontrollable or unobservable poles we cannot use the routine TG01JD. Instead we have to apply the routine TG01HX (the UFPSP from Subsection 3.2.1) once to the original system and once to the pertransposed dual system which is obtained by calling TB01XD and MA02CD.

6.1.2 Subroutine AB13DD.F

This routine is used to compute the $\mathcal{L}_\infty$-norm and the peak frequency (the frequency where the norm is attained) of the transfer function of a descriptor system with given realization $(E; A, B, C, D)$. This routine is already included in the current SLICOT version and in the scope of this thesis it is extended to continuous-time descriptor systems, using structure-exploiting computations. The interface is

```
 SUBROUTINE AB13DD( DICO, JOBE, EQUIL, JOBD, N, M, P, RANKE, FPEAK,
$                   A, LDA, E, LDE, B, LDB, C, LDC, D, LDD, GPEAK,
$                   TOL, BWORK, IWORK, DWORK, LDWORK, CWORK,
$                   LCWORK, INFO ).
```

The function of the arguments is as follows:

- DICO: Character that specifies if a discrete- or continuous-time system is given.
- JOBE: Character that specifies the form of the descriptor matrix E. It can be specified whether $E = I$ or E is an arbitrary nonsingular matrix. In the continuous-time case we additionally allow E to be singular or in the compressed form as obtained after calling DGEISP. In this case only the full-rank block is needed as input.

- `EQUIL`: Character that specifies whether a balancing of the system pencil should be performed.
- `JOBD`: Character that specifies if the feedthrough matrix D is nonzero or not. A zero feedthrough matrix leads to some computational savings.
- `N`, `M`, `P`: The number of descriptor, input and output variables, respectively.
- `RANKE`: The rank of the descriptor matrix E if it is passed in compressed form.
- `FPEAK`: Double precision array with 2 elements. On entry, the ratio `FPEAK(1)/FPEAK(2)` is an optional initial guess for the peak frequency, on exit this ratio is the computed peak frequency. Infinite frequencies are obtained by setting `FPEAK(2) = 0`.
- `A`, `E`, `B`, `C`, `D`: The arrays that contain the state, descriptor, input, output, and feedthrough matrices, respectively. They are unchanged on exit.
- `LDA`, `LDE`, `LDB`, `LDC`, `LDD`: The leading dimensions of the arrays `A`, `E`, `B`, `C`, `D`, respectively.
- `GPEAK`: Double precision array with 2 elements. The ratio `GPEAK(1)/GPEAK(2)` is the computed $\mathcal{L}_\infty$-norm. Infinite values are again obtained by setting `GPEAK(2) = 0`.
- `TOL`: The tolerance used to set the accuracy in determining the norm.
- `BWORK`, `IWORK`, `DWORK`, `CWORK`: Logical, integer, double precision and double complex workspace, respectively.
- `LDWORK`, `LCWORK`: The lengths of `DWORK` and `CWORK`, respectively.
- `INFO`: The error indicator which is negative if one of the arguments contains illegal values, positive if other problems like ill-conditioned subproblems or convergence problems occur, otherwise it is zero.

In Table 6.2 we summarize all important library routine calls in our extension of the routine, i.e., which are in particular important in the case of continuous-time descriptor systems.

When computing the value $\sigma_{\max}(G(\infty))$ using formula (3.2) in case of a compressed matrix E or using (4.19) or (4.20) for arbitrary singular E we solve m linear systems of equations by calling `DGESV` to obtain $A_{22/\infty}^{-1}B_{2/\infty}$. In this way we avoid forming inverses explicitely. The remaining work for forming $G(\infty)$ is done by simple matrix multiplications and additions. In the case of a compressed E we still need to compute the eigenvalues of $A - \lambda E$ by first transforming the matrix pencil to generalized

Table 6.2: Most important library routine calls in the extension of AB13DD

Library	Routine	Purpose in our Routine
LAPACK	DGESV	solving a general linear system of equations with multiple right-hand sides
LAPACK	DGESVD	computing the SVD of general rectangular matrix
SLICOT	TG01BD	orthogonal reduction of a descriptor system to the generalized Hessenberg form
LAPACK	DHGEQZ	single-/double-shift version of the QZ method for finding the generalized eigenvalues of a general matrix pencil in Hessenberg-triangular form
LAPACK	DGGES	computing the generalized eigenvalues, Schur form, and left and/or right Schur vectors of a general matrix pencil, enables eigenvalue reordering
SLICOT	SB04OD	solving a generalized Sylvester equation
SLICOT	AB13DX	computing the maximum singular value of a transfer function evaluated at a specific frequency
SLICOT	MB04BD	computing the eigenvalues of a skew-Hamiltonian/Hamiltonian matrix pencil

Hessenberg form by calling TG01BD and subsequently applying DHGEQZ. In the case of a general singular E the computation of the eigenvalues and the generalized real Schur form of $A - \lambda E$ is already done during the computation of $G(\infty)$. Note that for calling the function AB13DX to evaluate the transfer function at test frequencies we require the matrix pencil $A - \lambda E$ to be in generalized Hessenberg form. This is also done in order to achieve some computational savings.

6.1.3 Subroutine MB04BD.F

This subroutine is used to compute the eigenvalues of skew-Hamiltonian/Hamiltonian matrix pencils by the structure-preserving method proposed in Algorithm 5.1. The routine can also be used to compute the decompositions (5.4) and the corresponding orthogonal transformation matrices. We use a modified version of the routine which will be included in the next release version of SLICOT. Since Hamiltonian and skew-Hamiltonian matrices have certain block structures we use a packed storage layout proposed in [BBB00] to avoid saving redundant data. More specifically, if a $2n \times 2n$ Hamiltonian matrix $\mathcal{H} = \begin{bmatrix} A & D \\ E & -A^T \end{bmatrix}$ is given, we save the submatrix A in a convential $n \times n$ array A, the symmetric submatrices D and E are stored in an $n \times (n+1)$ array

Hamiltonian	skew-Hamiltonian
$\texttt{DE} = \begin{bmatrix} \texttt{e}_{11} & \texttt{d}_{11} & \texttt{d}_{12} & \texttt{d}_{13} & \dots \\ \texttt{e}_{21} & \texttt{e}_{22} & \texttt{d}_{22} & \texttt{d}_{23} & \dots \\ \texttt{e}_{31} & \texttt{e}_{32} & \texttt{e}_{33} & \texttt{d}_{33} & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$	$\texttt{DE} = \begin{bmatrix} \star & \star & \texttt{d}_{12} & \texttt{d}_{13} & \dots \\ \texttt{e}_{21} & \star & \star & \texttt{d}_{23} & \dots \\ \texttt{e}_{31} & \texttt{e}_{32} & \star & \star & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$

Figure 6.1: Storage layout for the (skew-)symmetric submatrices D and E

DE such that the upper triangular part of D is stored in DE(1:n,2:n+1) and the lower triangular part of E is stored in DE(1:n,1:n). The skew-symmetric parts of a skew-Hamiltonian matrix are similarly stored with the notable difference that the parts containing the diagonal and the first superdiagonal of the array DE are not referenced. See also Figure 6.1 for a visualization. The interface of our routine is given by

```
 SUBROUTINE MB04BD( JOB, COMPQ1, COMPQ2, N, A, LDA, D, LDD, C1,
$                   LDC1, VW, LDVW, Q1, LDQ1, Q2, LDQ2, B, LDB, F,
$                   LDF, C2, LDC2, ALPHAR, ALPHAI, BETA, IWORK,
$                   LIWORK, DWORK, LDWORK, INFO ).
```

We again briefly describe the input and output parameters which are:

- JOB: Character that specifies if only the eigenvalues or also the decomposition (5.4) is wanted.
- COMPQ1, COMPQ2: Characters that specify if the orthogonal transformation matrices $\mathcal{Q}_1$ and $\mathcal{Q}_2$ in (5.4) should be computed, initialized to the identity, or updated.
- N: The dimension of the input skew-Hamiltonian/Hamiltonian matrix pencil.
- A, D: The arrays that contain the submatrices which determine the skew-Hamiltonian matrix $\mathcal{N}$. On entry, the array A defines $\mathcal{N} = \begin{bmatrix} A & 0 \\ 0 & A^T \end{bmatrix}$. On exit, we have $\mathcal{N} = \begin{bmatrix} A & D \\ 0 & A^T \end{bmatrix}$.
- C1, VW, C2: The arrays that contain the submatrices which determine the matrix $\mathcal{H}$. On entry, they define the Hamiltonian input matrix $\mathcal{H} = \begin{bmatrix} C_1 & V \\ W & -C_1^T \end{bmatrix}$.

On exit, they determine the transformed matrix $\mathcal{Q}_1^T \mathcal{H} \mathcal{Q}_2 = \begin{bmatrix} C_1 & V \\ 0 & C_2^T \end{bmatrix}$ as in (5.4).

- `B`, `F`: The arrays that determine the skew-Hamiltonian triangular output matrix $\mathcal{M} = \begin{bmatrix} B & F \\ 0 & B^T \end{bmatrix}$.

- `Q1`, `Q2`: The arrays that contain the orthogonal transformation matrices $\mathcal{Q}_1$ and $\mathcal{Q}_2$ from (5.4), respectively.

- `LDA`, `LDD`, `LDC1`, `LDVW`, `LDQ1`, `LDQ2`, `LDB`, `LDF`, `LDC2`: The leading dimensions of the arrays `A`, `D`, `C1`, `VW`, `Q1`, `Q2`, `B`, `F`, `C2`, respectively.

- `ALPHAR`, `ALPHAI`, `BETA`: The arrays that contain information about the eigenvalues. That is, the j-th eigenvalue λ_j is represented by the ratio $\lambda_j =$ `(ALPHAR(j) + i*ALPHAI(j))/BETA(j)`. This product should not be computed explicitely since it could easily cause over- or underflow. Infinite eigenvalues λ_j satisfy `BETA(j) = 0`.

- `IWORK`, `DWORK`: Integer and double precision workspace, respectively.

- `LIWORK`, `LDWORK`: The lengths of `IWORK` and `DWORK`, respectively.

- `INFO`: The error indicator which is negative if one of the arguments contains illegal values, positive if there occur problems during the computation of the eigenvalues (e.g., during the periodic QZ algorithm), otherwise it is zero.

In Table 6.3 all needed library routines are listed. The routines `DLARFG` and `DLARF` are used to generate a Householder reflection and to update the corresponding parts of the matrices A and C_1. However, the situation for the matrix V is more difficult, since it is symmetric and so we have to take this structure into account. We explain this in more detail. Assume we are in iteration k of Step 1 in Algorithm 5.1. In this step we compute and apply a Householder reflection

$$H = \begin{bmatrix} I_{k-1} & 0 \\ 0 & \tilde{H}(v) \end{bmatrix}, \quad \tilde{H}(v) = I - \tau v v^T.$$

Furthermore we partition

$$V = \left[\begin{array}{c|c} V(1:k-1,1:k-1) & V(1:k-1,k:m) \\ \hline V(k:m,1:k-1) & V(k:m,k:m) \end{array}\right] =: \left[\begin{array}{c|c} V_{11} & V_{12} \\ \hline V_{21} & V_{22} \end{array}\right].$$

Table 6.3: Library routine calls in MB04BD

Library	Routine	Purpose in our Routine
LAPACK	DLARFG	generating a Householder reflection
LAPACK	DLARF	applying a Householder reflection
BLAS	DSYMV	symmetric matrix-vector multiply
BLAS	DDOT	dot product
BLAS	DAXPY	performing the vector operation $y = ax + y$
BLAS	DSYR2	performing the symmetric rank-2 operation $A := \alpha xy^T + \alpha yx^T + A$
LAPACK	DLARTG	generating a Givens rotation
BLAS	DROT	applying a Givens rotation
SLICOT	MB03BD	performing the periodic QZ algorithm

First we set $V_{12} := V_{12}\tilde{H}(v)$, $V_{21} := \tilde{H}(v)^T V_{21}$ by calling DLARF. Now we need a symmetric update of V_{22}, i.e.,

$$\begin{aligned} V_{22} &:= \tilde{H}(v)^T V_{22}\tilde{H}(v) \\ &= \left(I - \tau vv^T\right) V_{22} \left(I - \tau vv^T\right) \\ &= V_{22} - \tau vv^T V_{22} - \tau V_{22} vv^T + \tau^2 vv^T V_{22} vv^T \\ &= V_{22} - vx^T - xv^T + \frac{1}{2}\tau \left(x^T v\right) vv^T + \frac{1}{2}\tau \left(x^T v\right) vv^T, \quad x := \tau V_{22} v \\ &= V_{22} - vw^T - wv^T, \quad w := x - \frac{1}{2}\tau \left(x^T v\right) v, \end{aligned}$$

so we first compute the auxiliary vectors x using DSYMV and w using DDOT and DAXPY. Finally, we perform a rank-2 update by applying DSYR2. In Step 2 we only need Givens rotations as provided by DLARTG and DROT. Note that we do not update those parts of the matrices which are theoretically zero in order to obtain some computational savings. Finally, by calling the routine MB03BD we compute the imaginary positive part of the spectrum of $\mathcal{H} - \lambda\mathcal{N}$. Because of the Hamiltonian eigensymmetry it is not necessary to compute the imaginary negative part as well.

6.2 Numerical Experiments

6.2.1 Test Examples

Constrained Damped Mass-Spring System

The first example is a damped mass-spring system with holonomic constraints illustrated in Figure 6.2 (see [MS05, Sok06]). The i-th mass m_i is connected to the

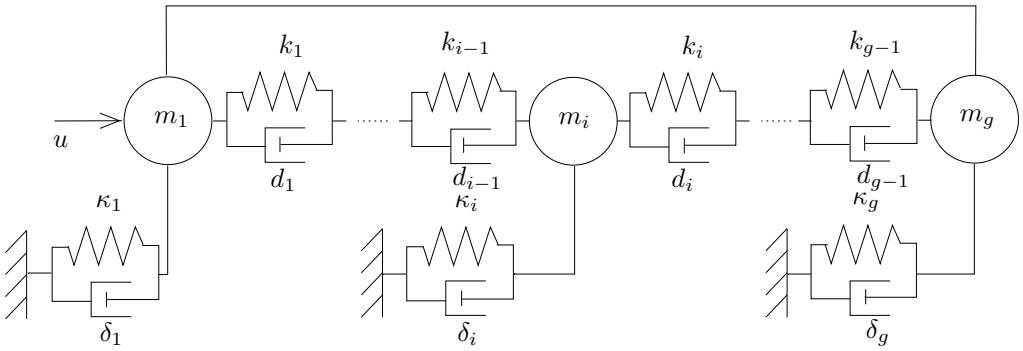


Figure 6.2: Constrained damped mass-spring system with g masses

$(i+1)$-st mass by a spring and a damper with constants k_i and d_i, respectively, and also to the ground by a spring and a damper with constants κ_i and δ_i, respectively. Additionally, the first mass is connected to the last one by a rigid bar and it is influenced by the control $u(t)$. The vibration of this system can be described by a second order descriptor system. By standard linearization methods we obtain a first order descriptor system of the form

$$\begin{aligned}
\dot{\mathbf{p}}(t) &= \mathbf{v}(t), \\
M\dot{\mathbf{v}}(t) &= -K\mathbf{p}(t) - D\mathbf{v}(t) + F^T\boldsymbol{\lambda}(t) + B_2u(t), \\
0 &= F\mathbf{p}(t), \\
y(t) &= C_1\mathbf{p}(t),
\end{aligned}$$

with algebraic index 3, where $\mathbf{p} \in \mathbb{R}^g$ is the position vector , $\mathbf{v}(t) \in \mathbb{R}^g$ is the velocity vector, $\boldsymbol{\lambda}(t) \in \mathbb{R}$ is the Lagrange multiplier, $M = \operatorname{diag}(m_1, \ldots, m_g)$ is the mass matrix, D and K are the tridiagonal damping and stiffness matrices, respectively. For our experiments we take $m_1 = \ldots = m_g = 100$ and

$$\begin{aligned}
k_1 = \ldots = k_{g-1} = \kappa_2 = \ldots = \kappa_{g-1} = 2, \qquad \kappa_1 = \kappa_g = 4, \\
d_1 = \ldots = d_{g-1} = \delta_2 = \ldots = \delta_{g-1} = 5, \qquad \delta_1 = \delta_g = 10.
\end{aligned}$$

Furthermore we assume that we can accelerate the first mass of the system, i.e., $B_2 = \begin{bmatrix} 1 & 0 & \ldots & 0 \end{bmatrix}^T$ and that we observe the position of the first mass, i.e., $C_1 = \begin{bmatrix} 1 & 0 & \ldots & 0 \end{bmatrix}$.

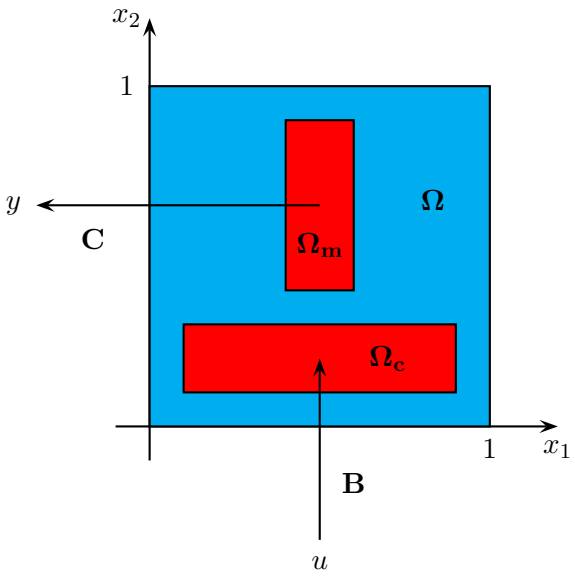


Figure 6.3: Test configuration: Stokes equation on a square with homogeneous Dirichlet boundary conditions

Semidiscretized Stokes Equation

Next we consider the d-dimensional ($d = 2, 3$) instationary Stokes equation [Sty02, MS05] describing the flow of an incompressible fluid

$$\begin{aligned} \frac{\partial v}{\partial t} &= \Delta v - \nabla \rho + f, && (x,t) \in \Omega \times (0, t_f), \\ 0 &= \nabla \cdot v, && (x,t) \in \Omega \times (0, t_f), \end{aligned}$$

with appropriate initial conditions and homogeneous Dirichlet boundary conditions. Here $v(t,x) \in \mathbb{R}^d$ is the velocity vector, $\rho(t,x) \in \mathbb{R}$ is the pressure, and $f(t,x) \in \mathbb{R}^d$ is the vector of external forces. For our tests we assume $d = 2$ and $\Omega = [0,1] \times [0,1]$. Further we assume that we can influence the flow by means of a volume force which is assumed to be free of potential force components on the control domain $\Omega_c = (0.1, 0.9) \times (0.1, 0.3)$. We measure the x_1-averaged distribution in x_2-direction of the two velocity components in the measurement domain $\Omega_m = (0.4, 0.6) \times (0.4, 0.9)$ (see [Sch07]). See also Figure 6.3 for a graphical interpretation of our configuration. Using a finite volume semidiscretization method on a

uniform staggered $k \times k$ grid [Wei97, Bew01], we obtain the descriptor system

$$\begin{aligned} \dot{\mathbf{v}}(t) &= A_{11}\mathbf{v}(t) + A_{12}\boldsymbol{\rho}(t) + B_1 u(t), \\ 0 &= A_{12}^T \mathbf{v}(t) + B_2 u(t), \\ y(t) &= C_1 \mathbf{v}(t) + C_2 \boldsymbol{\rho}(t). \end{aligned} \tag{6.1}$$

Here $\mathbf{v}(t) \in \mathbb{R}^{n_\mathbf{v}}$ is the semidiscretized vector of velocities, $\boldsymbol{\rho}(t) \in \mathbb{R}^{n_\rho}$ is the semidiscretized vector of pressures, $A_{11} = A_{11}^T \in \mathbb{R}^{n_\mathbf{v} \times n_\mathbf{v}}$ is the discrete Laplace operator, and $-A_{12} \in \mathbb{R}^{n_\mathbf{v} \times n_\rho}$ and $-A_{12}^T \in \mathbb{R}^{n_\rho \times n_\mathbf{v}}$ are the discrete gradient and divergence operators, respectively. Due to the non-uniqueness of the pressure, the matrix A_{12} has a rank defect one. In this case, instead of A_{12} we can take a full column rank matrix obtained from A_{12} by disregarding the last column. Therefore, in the following we assume without loss of generality that A_{12} has full column rank. In this case, system (6.1) is of index 2. The matrices $B_1 \in \mathbb{R}^{n_\mathbf{v} \times m}$, $B_2 \in \mathbb{R}^{n_\rho \times m}$ and the control input $u(t) \in \mathbb{R}^m$ are resulting from the external forces, the output $y(t) \in \mathbb{R}^p$ is the vector of interest. The order $n = n_\mathbf{v} + n_\rho$ of system (6.1) depends on the level of refinement of the discretization and is usually very large, whereas the number m of inputs and the number p of outputs are typically small. For our tests we set $m = 4k$ and $p = k$.

6.2.2 Numerical Results

In this subsection we present some numerical results based on our Fortran implementations of the algorithms using the test examples introduced above.

Subroutine `DGEISP.F`

Now we test the subroutine `DGEISP`. First we analyze the routine with the constrained mass-spring system from Paragraph 6.2.1. In Tables 6.4 and 6.5 we state the number of masses g, the order of the original system n, the order of the reduced system n_r, and if the associated transfer function is determined as proper or improper. Furthermore we verify if the fast subsystems of the reduced systems are C-controllable and C-observable. This can be done by checking if the matrices

$$\mathcal{C}_\infty := \begin{bmatrix} E_r & B_r \end{bmatrix}, \quad \mathcal{O}_\infty := \begin{bmatrix} E_r \\ C_r \end{bmatrix},$$

composed by E_r, B_r, C_r corresponding to the reduced system have full rank, i.e., we consider the matrices as singular if their 2-norm condition numbers exceed a certain threshold, e.g., $1/\varepsilon$. Finally we measure the runtime, once if we remove all infinite uncontrollable or unobservable poles and once if we remove all uncontrollable or unobservable poles of the system. In the routine we perform scaling of the system pencil and we take a tolerance of $n^2\varepsilon$ for determining the numerical ranks.

Table 6.4: Results of `DGEISP` for constrained damped mass-spring system when removing only uncontrollable or unobservable nonzero finite and infinite poles

g	n	n_r	proper?	time in s	$\kappa_2(\mathcal{C}_\infty)$	$\kappa_2(\mathcal{O}_\infty)$
5	11	6	yes	0.00058	10.0000	10.0000
10	21	10	yes	0.00130	10.0000	10.0000
20	41	41	no	0.0077	∞	∞
50	101	101	no	0.0743	∞	∞
100	201	201	no	0.3615	∞	∞
200	401	401	no	2.7148	∞	∞
300	601	601	no	9.5858	∞	∞
500	1001	1001	no	46.6024	∞	∞

Table 6.5: Results of `DGEISP` for constrained damped mass-spring system when removing all uncontrollable or unobservable poles

g	n_r	proper?	time in s	$\kappa_2(\mathcal{C}_\infty)$	$\kappa_2(\mathcal{O}_\infty)$
5	6	yes	0.00063	10.0000	10.0000
10	10	yes	0.00127	10.0000	10.0000
20	23	no	0.0060	∞	∞
50	53	no	0.0405	∞	∞
100	103	no	0.2196	2.4142e+17	∞
200	203	no	1.3087	5.6636e+29	∞
300	303	no	4.4915	∞	∞
500	503	no	20.5077	2.4140e+17	∞

Unfortunately for the constrained damped mass-spring system we obtain rather unsatisfactory results. Even for very small system sizes the transfer functions are determined to be improper which does not match the theoretical results. Note also that the conditions for C-controllability and C-observability of the fast subsystems are not fulfilled since the matrices $\mathcal{C}_\infty$ and $\mathcal{O}_\infty$ are singular as their condition numbers are very high or even infinity. As reason we suppose a problem in the `SLICOT` routine `TG01HX` which is responsible for the finite/infinite controllability/observability form reductions (UFPSP from [Var90]). Note that the smallest singular values of $\mathcal{C}_\infty$ and $\mathcal{O}_\infty$ are mostly exactly zero (when κ_2 is infinity). In this way even for small tolerances the rank should be correctly estimated. This problem can also not be solved by simply increasing the tolerances.

Generally we observe cubic runtime. But we also remark that the elapsed time

for removing all uncontrollable or unobservable nonzero finite and infinite poles is much higher than the time for removing all uncontrollable or unobservable poles. The reason is that if we only want to remove infinite poles there is no reduction of the system. I.e., in this case the UFPSP is called twice with a quite high system order. On the other hand, if we want to remove all uncontrollable or unobservable poles, in the first reduction phase (removing all finite uncontrollable poles) the system is already reduced to the final reduced order n_r. Consequently, the following three reduction phases only have to deal with a much smaller system which saves of course a lot of computational time.

Table 6.6: Results of `DGEISP` for semidiscretized Stokes equation when removing only uncontrollable or unobservable nonzero finite and infinite poles

k	n	n_r	τ	proper?	time in s	$\kappa_2(\mathcal{C}_\infty)$	$\kappa_2(\mathcal{O}_\infty)$
4	39	6	100ε	yes	0.0086	80.7838	17.6794
6	95	25	1000ε	yes	0.0281	1.6139e+03	10.1050
8	175	43	30000ε	yes	0.0853	4.4593e+03	2.2043e+15
10	279	88	$10^5\varepsilon$	yes	0.4090	5.9232e+12	1.4944e+15
12	407	134	$5 \cdot 10^7\varepsilon$	yes	1.1411	7.2773e+09	1.2025e+15
14	559	186	$10^8\varepsilon$	yes	2.8604	4.0062e+11	1.1407e+15
16	735	240	$5 \cdot 10^9\varepsilon$	yes	6.1766	5.6533e+05	4.8757e+11

We also apply `DGEISP` to the semidiscretized Stokes equation as described in Paragraph 6.2.1. For this example we get much better results as listed in Table 6.6. However we have to choose appropriate tolerances τ to obtain good results. If τ is too low, we again have to consider $\mathcal{C}_\infty$ and $\mathcal{O}_\infty$ as singular since their condition numbers become too large. To demonstrate this behavior we choose the semidiscretized Stokes equation with $k = 16$ and compute n_r, $\kappa_2(\mathcal{C}_\infty)$, and $\kappa_2(\mathcal{O}_\infty)$ for different values of τ as shown in Table 6.7. From this table we can see that we obtain nonsingular matrices $\mathcal{C}_\infty$ and $\mathcal{O}_\infty$ the first time for $\tau \approx 10^{10}\varepsilon$. But the most important observation is that for every value of τ we get different orders of the reduced system. This is not even monotone, i.e., for increasing values of τ sometimes the values of n_r increase. In this way it is very difficult to make robust rank decisions, especially if the systems have high orders. So it should be preferred to check properness of a transfer function by analytic methods, however the reduction of the systems' orders is a nice feature which could reduce the computational effort of the $\mathcal{L}_\infty$-norm algorithm drastically.

Table 6.7: Results of DGEISP for semidiscretized Stokes equation with $k = 16$ and different values of τ when removing only uncontrollable or unobservable nonzero finite and infinite poles

τ	n_r	$\kappa_2(\mathcal{C}_\infty)$	$\kappa_2(\mathcal{O}_\infty)$
$10^2\varepsilon$	288	8.8237e+16	2.3802e+64
$10^3\varepsilon$	297	1.7143e+17	4.0136e+239
$10^4\varepsilon$	270	3.7454e+16	2.5881e+37
$10^5\varepsilon$	267	1.9342e+16	3.5819e+18
$10^6\varepsilon$	261	2.0334e+15	3.1946e+16
$10^7\varepsilon$	261	9.2423e+13	1.9860e+16
$10^8\varepsilon$	259	1.4908e+13	2.4501e+16
$10^9\varepsilon$	249	3.5273e+09	1.2403e+16
$10^{10}\varepsilon$	241	2.2650e+04	2.4739e+12

Subroutine AB13DD.F

In this paragraph we analyze the numerical behavior of our modified version of AB13DD. Again we perform our first test with the constrained damped mass spring system from Paragraph 6.2.1. In Table 6.8 we list the number of masses, the order of the system, the peak frequency $\hat{\omega}$, the computed $\mathcal{L}_\infty$-norm $\|G\|_{\mathcal{L}_\infty} := \sigma_{\max}(G(\mathrm{i}\hat{\omega}))$, and the time needed for the computation.

Table 6.8: Results of AB13DD for the constrained damped mass-spring system

g	n	$\hat{\omega}$	$\|G\|_{\mathcal{L}_\infty}$	time in s
5	11	0.1475	0.1590	0.0147
10	21	0.1693	0.1508	0.0297
20	41	0.1579	0.1511	0.0756
50	101	0.1581	0.1511	0.3863
100	201	0.1581	0.1511	2.8206
200	401	0.1581	0.1511	25.1296
300	601	0.1581	0.1511	85.2102
500	1001	0.1581	0.1511	427.4301

We do the same for the reduced systems obtained by DGEISP. The corresponding results are summarized in Table 6.9. Since DGEISP does not remove all uncontrollable or unobservable infinite poles of the system, we are not allowed to use formula (3.2) to compute $G(\infty)$. Hence we specify JOBE such that AB13DD deals with singular matrices

E (not in compressed form). In our tests we do not perform scaling and the relative error for the computed $\mathcal{L}_\infty$-norm is set to $n\varepsilon$, where ε denotes the machine precision. We use the version of Algorithm 5.1 that requires the skew-Hamiltonian matrix to be already in skew-Hamiltonian triangular form.

Table 6.9: Results of `AB13DD` for the constrained damped mass-spring system (reduced model)

g	n_r	$\hat{\omega}_r$	$\|G_r\|_{\mathcal{L}_\infty}$	time in s
5	6	0.1475	0.1590	0.0130
10	10	0.1693	0.1508	0.0218
20	23	0.1579	0.1511	0.0312
50	53	0.1581	0.1511	0.0988
100	103	0.1581	0.1511	0.3574
200	203	0.1581	0.1511	2.7218
300	303	0.1581	0.1511	9.2196
500	503	0.1581	0.1511	50.7836

In Figure 6.4 we also plot the runtimes once for computing the $\mathcal{L}_\infty$-norm for the original system and once for computing the reduced model and subsequently computing its system norm. A look on the numbers yields that first computing the reduced system and then computing the norms roughly takes only 1/6 of the time needed for the computation of the norm for the original system (if g is sufficiently large). Generally the algorithm requires $\mathcal{O}\left(n^3\right)$ real floating point operations which can be also seen from Figure 6.4.

We also compare the distance between the peak frequencies and the $\mathcal{L}_\infty$-norms of the transfer functions of the original and the reduced systems. For this purpose we take the corresponding results for the original system as reference values and compute the relative errors by

$$\varepsilon_{\text{freq}} := \frac{|\hat{\omega} - \hat{\omega}_r|}{|\hat{\omega}|}, \quad \varepsilon_{\text{norm}} := \frac{\left|\|G\|_{\mathcal{L}_\infty} - \|G_r\|_{\mathcal{L}_\infty}\right|}{\|G\|_{\mathcal{L}_\infty}}.$$

The results are listed in Table 6.10. Especially the errors in the norms are all less than 10^{-8} which is quite satisfactory. In particular, if we do not need a very high accuracy it is worth to accept a slightly larger error to obtain lots of computational savings.

We also demonstrate the quadratic convergence of the method. Consider the constrained damped mass-spring system with $g = 10$ masses. First we have to compute the maximum singular value of the transfer function evaluated at certain test frequencies. In our case we obtain $\sigma_{\max}(G(0)) = 9.55056179775282260 \cdot 10^{-2}$,

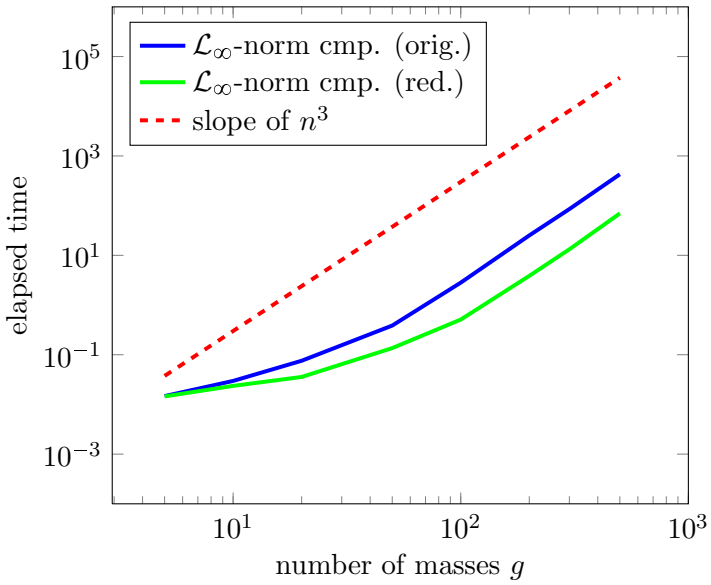


Figure 6.4: Comparison of the runtimes of `AB13DD` for the original models; and `DGEISP` and `AB13DD` for the reduced models of the constrained damped mass-spring system

$\sigma_{\max}(G(\infty)) = 3.80134287067529736 \cdot 10^{-19}$, $\sigma_{\max}(G(\mathrm{i}\omega_p)) = 0.15048033177512740$, using (4.9) and (4.10). In this way we already obtain a good initial estimation for the $\mathcal{L}_\infty$-norm which is $\|G\|_{\mathcal{L}_\infty} = 0.15080691648129951$ for our example. We show the Bode singular value plot and the relative error between the current iterate γ_i and the "exact" value of $\|G\|_{\mathcal{L}_\infty}$ at each iteration i in Figure 6.5. When setting the maximum relative error to $\tau = 1000\varepsilon$ we already obtain convergence after four iterations, i.e., we have to compute four times the eigenvalues of a skew-Hamiltonian/Hamiltonian matrix pencil.

We also apply the routine `AB13DD` to the semidiscretized Stokes equation. The results are summarized in Tables 6.11 and 6.12. Here the $\mathcal{L}_\infty$-norm is always attained at $\omega = 0$, so we only have to compute the eigenvalues of a skew-Hamiltonian/Hamiltonian matrix pencil once. We illustrate one example transfer function by the Bode singular value plot in Figure 6.6. Again the $\mathcal{L}_\infty$-norm computation for a reduced system takes only approximately 1/6 of the corresponding time for the original system. Also the relative errors between the norms of the original and reduced systems are very satisfactory as shown in Table 6.13.

We want to give one additional remark on the accuracy of the eigenvalues of the extended skew-Hamiltonian/Hamiltonian matrix pencils $\bar{\mathcal{M}}_\gamma - \lambda\bar{\mathcal{N}}$ from (4.23) or

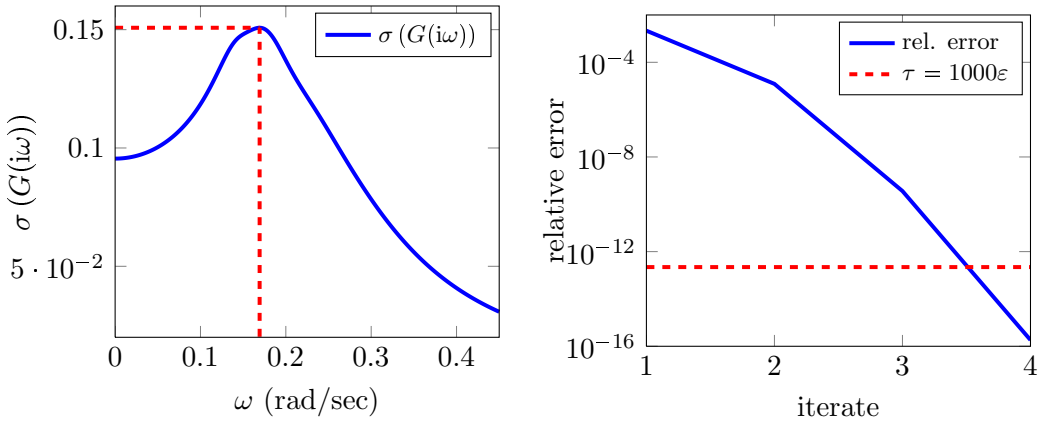
Table 6.10: Relative error of the peak frequencies and $\mathcal{L}_\infty$-norms between original and reduced systems for constrained damped mass-spring system

g	$\varepsilon_{\text{freq}}$	$\varepsilon_{\text{norm}}$
5	2.2581e-15	8.2047e-15
10	1.1805e-14	2.6135e-14
20	1.9100e-06	8.0590e-12
50	4.5722e-05	4.6485e-09
100	9.0950e-06	1.2433e-09
200	8.0773e-15	1.7266e-14
300	1.2818e-14	5.3268e-15
500	2.0969e-06	1.3485e-11

Table 6.11: Results of `AB13DD` for the semidiscretized Stokes equation (original model)

k	n	$\hat{\omega}$	$\|G\|_{\mathcal{L}_\infty}$	time in s
4	39	0	0.8479	0.0353
6	95	0	0.5815	0.2562
8	175	0	0.4939	1.4809
10	279	0	0.4348	5.8893
12	407	0	0.9956	19.3335
14	559	0	0.8691	50.0174
16	735	0	0.7664	114.2369

(4.25). As already mentioned in `AB13DD` the system pencil $A - \lambda E$ is transformed to generalized upper Schur or Hessenberg form. In this way we can build the extended matrix pencils by using the transformed system matrices and hence omit Step 1 in Algorithm 5.1. However, the transformation to generalized Schur form is done by using the QZ algorithm [GVL96] which is a kind of iterative method, i.e., the "exact" generalized Schur form cannot in general be obtained in a finite number of steps. So, besides possible rounding errors we also get an approximation error from the QZ method. Step 1 of Algorithm 5.1 also transforms the matrix $\bar{\mathcal{N}}$ to skew-Hamiltonian triangular form but this is done using a finite number of Householder reflections, so we get at most some rounding errors, in particular we can apply Algorithm 5.1 to the extended matrix pencils (4.23) or (4.25) built by the original system matrices. We can observe this difference also in the computed eigenvalues. We computed these for the first iteration of γ in `AB13DD` for the semidiscretized Stokes equation with $k = 4$. The maximum relative error between the finite eigenvalues is $2.0715 \cdot 10^{-14}$ which is still acceptable. However, almost all infinite eigenvalues take finite values in the method



(a) Bode plot

(b) Convergence history

Figure 6.5: Bode plot and convergence history of `AB13DD` of the constrained damped mass-spring system with $g = 10$ masses

without considering Step 1 of Algorithm 5.1. Some of these are also purely imaginary. In our case these are, e.g., $\pm 79023174364.764297\mathrm{i}$ and $\pm 74527725084.653229\mathrm{i}$. So it is very important to define a tolerance that specifies when an eigenvalue with finite numerical value is declared as infinite. This tolerance should ideally be coupled to the "size" of the matrix pencil $\bar{\mathcal{M}}_\gamma - \lambda\bar{\mathcal{N}}$, e.g., via the norms of the matrices $\bar{\mathcal{M}}_\gamma$ and $\bar{\mathcal{N}}$. When using the original system matrices for setting up (4.23) or (4.25) all theoretically infinite eigenvalues are also numerically infinite. In this case we do not need to define a tolerance as above but for other examples this could also be required by this method.

Subroutine `MB04BD.F`

Finally we test the routine `MB04BD` which is the implementation of our structure-preserving algorithm for the computation of the spectrum of a skew-Hamiltonian/Hamiltonian matrix pencil. Here we test the version that includes Step 1 of Algorithm 5.1, i.e., we assume that the skew-Hamiltonian matrix is block-diagonal with full blocks. First we compute the purely imaginary eigenvalues of a matrix pencil associated to the constrained damped mass-spring system with $g = 10$ masses and $\gamma = 0.1$. As seen before it holds that $\sigma_{\max}(G(0)) < \gamma < \|G\|_{\mathcal{L}_\infty}$, in particular this guarantees the existence of at least four purely imaginary eigenvalues. As second example we compute the imaginary eigenvalues of a matrix pencil associated to the semidiscretized Stokes equation with $k = 10$ and $\gamma = 0.1$. Again, we can ensure the existence of four purely imaginary eigenvalues. For testing we set up the matrix

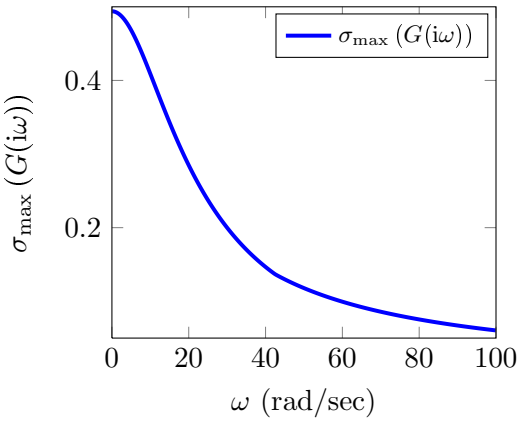


Figure 6.6: Bode plot for the semidiscretized Stokes equation for $k = 8$

Table 6.12: Results of `AB13DD` for the semidiscretized Stokes equation (reduced model)

k	n_r	$\hat{\omega}_r$	$\|G_r\|_{\mathcal{L}_\infty}$	time (`AB13DD`) in s	time (`DGEISP`+`AB13DD`) in s
4	6	0	0.8479	0.0096	0.0128
6	25	0	0.5815	0.0277	0.0517
8	43	0	0.4939	0.1010	0.2403
10	88	0	0.4348	0.3525	0.9263
12	134	0	0.9956	1.0648	3.0242
14	186	0	0.8691	2.8634	8.1392
16	240	0	0.7664	6.1732	19.8795

pencils (4.25), directly built by the system matrices. We compare the results of the structure-preserving algorithm with the eigenvalues computed by the QZ algorithm which is the standard algorithm for generalized eigenvalue problems. The results are shown in Figure 6.7.

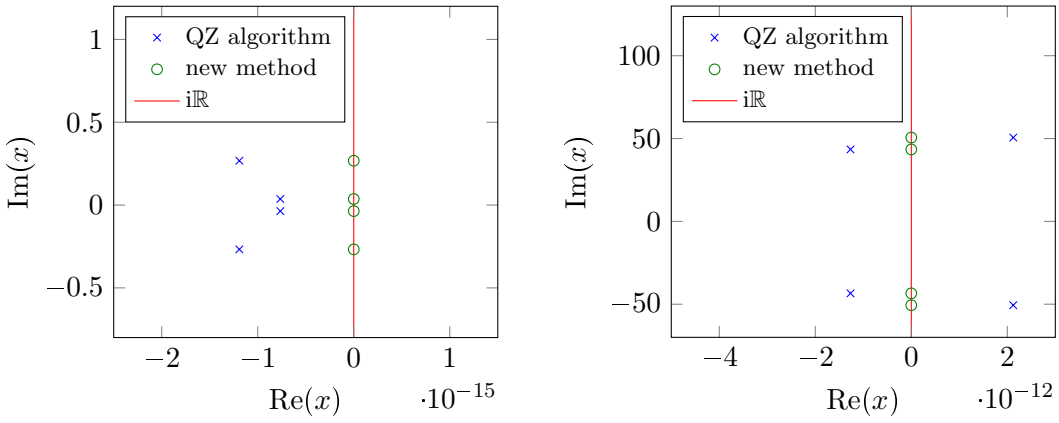
We observe that the eigenvalues computed by our new method corresponding to purely imaginary eigenvalues, are purely imaginary. This also coincides with the theory since there are only structured perturbations allowed. In our example all purely imaginary eigenvalues are simple. If there was a perturbation of such an eigenvalue λ away from the imaginary axis there would not exist an eigenvalue which could take the role of $-\bar{\lambda}$ which is required by the Hamiltonian eigensymmetry. These imaginary eigenvalues correspond to real positive eigenvalues of the 4-matrix pencil obtained in Step 3 of Algorithm 5.1. Finally their roots are multiplied by $\pm\mathrm{i}$ and they become

Table 6.13: Relative error of the $\mathcal{L}_\infty$-norms between original and reduced systems for semidiscretized Stokes equation

k	$\varepsilon_{\text{norm}}$
4	6.5468e-16
6	2.2910e-15
8	2.5851e-15
10	6.0000e-15
12	2.0295e-14
14	8.5589e-15
16	2.3759e-14

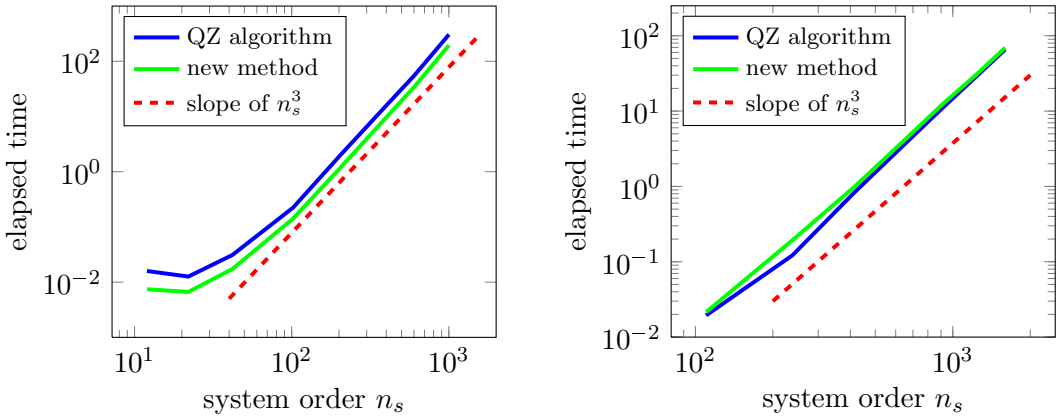
the purely imaginary eigenvalues of the considered skew-Hamiltonian/Hamiltonian matrix pencil. However, the QZ algorithm does not consider any structure and so also the purely imaginary eigenvalues could be perturbed in any direction. Consequently, as seen in Figure 6.7, the imaginary eigenvalues are moved away from the imaginary axis. Referring to [BBL+07] also the non-imaginary eigenvalues are less perturbed by the new method compared to the QZ algorithm.

Finally we compared the runtimes of the method with the QZ algorithm. For this purpose we called the `LAPACK` driver `DGGEV` compiled with the same compiler options as `MB04BD` by MATLAB® via a mexfile. The results are plotted in Figure 6.8 and besides the cubic growth it turns out that the structure-preserving method takes only about 2/3 of the time the QZ algorithm needs for the mass-spring system example. However, for the Stokes equation example the new method takes about 10% more time. The reason is simple. Before the QZ algorithm starts the actual QZ iteration, the matrix pencil is reduced to generalized Hessenberg form. In our example there are a lot of very small entries or zeros on the subdiagonal of the Hessenberg matrix. This is very advantageous for the QZ iteration as it has to spend much less effort to achieve deflation in the eigenvalues. But in general the exploitation of the structure leads to some savings as skew-Hamiltonian/Hamiltonian matrix pencils have less degrees of freedom in the choice of their entries. It can be concluded that the new structure-preserving algorithm generally beats the QZ algorithm under every aspect.



(a) Constrained damped mass-spring system (b) Semidiscretized Stokes equation

Figure 6.7: Computed purely imaginary eigenvalues of two skew-Hamiltonian/Hamiltonian example matrix pencils



(a) Constrained damped mass-spring system (b) Semidiscretized Stokes equation

Figure 6.8: Comparison of the runtimes between the QZ algorithm and the new eigenvalue solver

7 Conclusion and Outlook

Finally we summarize the results of our work and state some open problems which have to be analyzed by future research.

First we motivated the thesis topic by some examples and applications in Chapter 1 and introduced the basic concepts of linear algebra and systems and control theory in Chapter 2, in particular the notion of the transfer function of a descriptor system and its $\mathcal{L}_\infty$-norm. We encountered the difficulty that not every descriptor system has a proper transfer function. To check this property we derived a numerical method in Chapter 3. Besides the test for properness this method also reduces the order of the involved descriptor system by removing uncontrollable or unobservable poles of the system which may reduce the costs for the computation of the $\mathcal{L}_\infty$-norm. In Chapter 4 we extended an existing numerical method for the computation of the $\mathcal{L}_\infty$-norm of standard state space systems to descriptor systems. For standard systems there exists a connection between the singular values of the transfer function and the eigenvalues of certain Hamiltonian matrices. This framework was extended to skew-Hamiltonain/Hamiltonian matrix pencils in the descriptor system case. The derived algorithm carries over all the advantageous properties from the standard system algorithm. However, we have to spend more effort in the computation of the lower bound of the iterates calculated by the $\mathcal{L}_\infty$-norm algorithm as described in Section 4.3. To improve the stability and accuracy of the required eigenvalue computation we proposed an extension strategy for the skew-Hamiltonian/Hamiltonian matrix pencils in Section 4.4 to work directly with the original data without explicitely forming matrix products and inverses. We also had a brief view on discrete-time systems in Section 4.5. Here we have to work with symplectic matrix pencils instead of skew-Hamiltonian/Hamiltonian ones. We presented some ideas how we can transform the symplectic matrix pencils to skew-Hamiltonian/Hamiltonian or palindromic matrix pencils which allow a more accurate computation of their eigenvalues since there exist algorithms which exploit the matrix structures. However, there are still some open questions concerning the transformation of certain eigenvalues. In Chapter 5 we presented a structure-preserving algorithm for the accurate computation of the eigenvalues of skew-Hamiltonian/Hamiltonian matrix pencils. As only structured perturbations are allowed in this method and hence the Hamiltonian eigensymmetry is preserved, especially all simple imaginary eigenvalues are computed exactly as purely imaginary. The imaginary eigenvalues are of very high importance in the $\mathcal{L}_\infty$-norm computation method and so the reliability and the accuracy of the results were additionally increased. Finally, in Chapter 6 we explained how the considered

algorithms are implemented in Fortran as `SLICOT`-style routines. All routines are tested with two example systems illustrated in Section 6.2. In the experiments it turned out that the properness testing procedure works only with some restrictions. First we encountered some problems with the `SLICOT` routine `TG01HX` which does not properly remove all uncontrollable or unobservable infinite poles. And second it is very important to choose an appropriate tolerace to determine the numerical rank of the involved matrices. We observed quite large problems with robustness during rank determinations. However, this routine still provides a good reduction of system orders which reduces the costs of the $\mathcal{L}_\infty$-norm computation for our examples drastically. The results of the $\mathcal{L}_\infty$-norm algorithm are very satisfactory. We observe quick convergence of the iterates and obtain very high accuracy by the improvements explained in this thesis. Also our structure-preserving method for the eigenvalue computation of skew-Hamiltonian/Hamiltonian matrix pencils generally behaves much better than standard methods with respect to speed and accuracy.

There are still many open problems and questions which have to be analyzed in future research. First we have to investigate discrete-time systems in more detail. As already mentioned there are still some problems with the transformation of some eigenvalues of the associated extended symplectic matrix pencils. In this context we also think of an efficient implementation of the discrete-time part of the algorithm in Fortran. We also remark that the methods analyzed in this thesis are only reasonable for fairly small systems because we rely on algorithms for dense matrices. There exist some iterative algorithms for the computation of the $\mathcal{H}_\infty$-norm for large-scale standard state space systems but by the author's best knowledge there is still no way known to extend this to the descriptor system case.

Bibliography

[ABB+99] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov and D. Sorensen: *LAPACK's User's Guide - Release 3.0*, Philadelphia, PA, USA, Aug. 1999.
URL http://www.netlib.org/lapack/lug/

[BB90] S. Boyd and V. Balakrishnan: *A regularity result for the singular values of a transfer matrix and a quadratically convergent algorithm for computing its L_∞-norm*, *Syst. Control Lett.*, vol. 15(1):pp. 1–7, 1990, ISSN 0167-6911.

[BBB00] P. Benner, R. Byers and E. Barth: *Algorithm 800: Fortran 77 subroutines for computing the eigenvalues of Hamiltonian matrices I: the square-reduced method*, *ACM Trans. Math. Softw.*, vol. 26(1):pp. 49–77, 2000, ISSN 0098-3500.

[BBK89] S. Boyd, V. Balakrishnan and P. Kabamba: *A Bisection Method for Computing the H_∞ Norm of a Transfer Matrix and Related Problems*, *Mathematics of Control, Signals, and Systems*, vol. 2(3):pp. 207–219, Sept. 1989.

[BBL+07] P. Benner, R. Byers, P. Losse, V. Mehrmann and H. Xu: *Numerical Solution of Real Skew-Hamiltonian/Hamiltonian Eigenproblems*, Nov. 2007.

[BBMX99] P. Benner, R. Byers, V. Mehrmann and H. Xu: *Numerical Computation of Deflating Subspaces of Embedded Hamiltonian Pencils*, Tech. Rep., Chemnitz University of Technology, Faculty of Mathematics, Germany, Jun. 1999, SFB393-Preprint 99-15.

[BD85] S. Boyd and C. A. Desoer: *Subharmonic Functions and Performance Bounds on Linear Time-Invariant Feedback Systems*, *IMA J. Math. Control I.*, vol. 2:pp. 153–170, 1985.

[Ben06] P. Benner: *Control Theory*, in *Handbook of Linear Algebra* (ed. by L. Hogben), Discrete Mathematics and Its Applications, chap. 57, Chapman & Hall/CRC, Boca Raton, Florida, 2006, ISBN 978-1-58488-510-8.

[Ben09] P. Benner: *Advances in Balancing-Related Model Reduction for Circuit Simulation*, in *Scientific Computing in Electrical Engineering SCEE 2008* (ed. by J. Roos and L. R. J. Costa), vol. 14 of *Mathematics in Industry*, Springer-Verlag, Berlin/Heidelberg, 2009.

[Bew01] T. R. Bewley: *FLOW CONTROL: New Challenges for a New Renaissance*, in *Progress in Aerospace Sciences 37*, pp. 21–58, American Elsevier Publishing Company, Inc, 2001.

[BGVD92] A.l Bojanczyk, G. Golub and P. Van Dooren: *The periodic Schur decomposition. Algorithms and applications*, in *In Proc. SPIE Conference*, pp. 31–42, 1992.

[BK06] P. Benner and D. Kressner: *Algorithm 854: Fortran 77 subroutines for computing the eigenvalues of Hamiltonian matrices II*, *ACM Trans. Math. Softw.*, vol. 32(2):pp. 352–373, 2006, ISSN 0098-3500.

[BLM+08] P. Benner, P. Losse, V. Mehrmann, L. Poppe and T. Reis: *γ-Iteration for Descriptor Systems Using Structured Matrix Pencils*, in *Proceedings of the International Symposium on Mathematical Theory of Networks and Systems*, Blacksburg VA, USA, 2008.

[BMS+97] P. Benner, V. Mehrmann, V. Sima, S. van Huffel and A. Varga: *SLICOT – A Subroutine Library in Systems and Control Theory*, NICONET, Jun. 1997, Report 97-3.

[BMX02] P. Benner, V. Mehrmann and H. Xu: *Perturbation Analysis for the Eigenvalue Problem of a Formal Product of Matrices*, *BIT*, vol. 42(1):pp. 1–43, 2002, ISSN 0006-3835.

[BQO98a] C. H. Bischof and G. Quintana-Ortí: *Algorithm 782: codes for rank-revealing QR factorizations of dense matrices*, *ACM Trans. Math. Softw.*, vol. 24(2):pp. 254–257, 1998, ISSN 0098-3500.

[BQO98b] C. H. Bischof and G. Quintana-Ortí: *Computing rank-revealing QR factorizations of dense matrices*, *ACM Trans. Math. Softw.*, vol. 24(2):pp. 226–253, 1998, ISSN 0098-3500.

[BS90] N. A. Bruinsma and M. Steinbuch: *A fast algorithm to compute the H_∞-norm of a transfer function matrix*, *Syst. Control Lett.*, vol. 14(4):pp. 287–293, 1990, ISSN 0167-6911.

[Bye88] R. Byers: *A Bisection Method for Measuring the Distance of a Stable Matrix to the Unstable Matrices*, *SIAM J. Sci. Stat. Comput.*, vol. 9:pp. 875–881, Sept. 1988.

[CGVD04] Y. Chahlaoui, K. Gallivan and P. Van Dooren: *$\mathcal{H}_\infty$-norm calculations of large sparse systems*, in *Proceedings International Symposium Math. Th. Netw. Syst.*, Leuven, Belgium, 2004.

[CGVD07] Y. Chahlaoui, K. Gallivan and P. Van Dooren: *Calculating the $\mathcal{H}_\infty$ norm of a large sparse system via Chandrasekhar iterations and extrapolation*, in *ESAIM Proceedings*, vol. 20, pp. 83–92, Rabat, Algeria, Oct. 2007.

[Dai89] L. Dai: *Singular Control Systems*, vol. 118 of *Lecture Notes in Control and Information Sciences*, Springer-Verlag, Heidelberg, 1989.

[Dat04] B. N. Datta: *Numerical Methods for Linear Control Systems*, Elsevier Academic Press, San Diego/London, 2004.

[GVDV98] Y. Genin, P. Van Dooren and V. Vermaut: *Convergence of the calculation of $\mathcal{H}_\infty$-norms and related questions*, in *Proceedings MTNS-98*, pp. 429–432, Jul. 1998.

[GVL96] G. H. Golub and C. F. Van Loan: *Matrix Computations*, The John Hopkins University Press, Baltimore/London, third edition, 1996.

[Hal03] B. Hall: *Lie Groups, Lie Algebras, and Representations - An Elementary Introduction*, vol. 222 of *Graduate Texts in Mathematics*, Springer-Verlag, first edition, 2003, ISBN 978-0-387-40122-5.

[HL94] J. J. Hench and A. J. Laub: *Numerical solution of the discrete-time periodic Riccati equation*, *IEEE Trans. Automat. Control*, vol. 39(6):pp. 1197–1210, Jun. 1994.

[HP05] D. Hinrichsen and A. J. Pritchard: *Mathematical Systems Theory I: Modelling, State Space Analysis, Stability and Robustness*, vol. 48 of *Texts in Applied Mathematics*, Springer-Verlag, Berlin, Heidelberg, New York, 15th edition, 2005, ISBN 3-540-44125-5.

[Jac68] N. Jacobson: *Structure and Representations of Jordan Algebras*, vol. 39 of *AMS Colloquium Publications*, American Mathematical Society, Providence, 1968, ISBN 978-0-8218-4640-7.

[Kre01a] D. Kressner: *An efficient and reliable implementation of the periodic QZ algorithm*, in *IFAC Workshop on Periodic Control Systems*, 2001.

[Kre01b] D. Kressner: *Numerical Methods for Structured Matrix Factorizations*, Diploma Thesis, Chemnitz University of Technology, Faculty of Mathematics, Germany, 2001.

[KSW09] D. Kressner, C. Schröder and D. S. Watkins: *Implicit QR algorithms for palindromic and even eigenvalue problems*, *Numer. Algorithms*, vol. 51(2):pp. 209–238, 2009.

[KVD90] B. Kågström and P. Van Dooren: *Additive Decomposition of a Transfer Function with respect to a Specified Region*, in *Proceedings of MTNS-89*, vol. 3, pp. 469–477, Birkhäuser Boston Inc., Boston, 1990.

[KVD91] B. Kågström and P. Van Dooren: *A Generalized State-space Approach for the Additive Decomposition of a Transfer Matrix*, Tech. Rep., Institute of Information Processing, University of Umeå, Sweden, Apr. 1991, Report UMINF-91.12.

[KW87] B. Kågström and L. Westin: *GSYLV - Fortran Routines for the Generalized Schur Method with Dif Estimators for Solving the Generalized Sylvester Equation*, Tech. Rep., Institute of Information Processing, University of Umeå, Sweden, Jul. 1987, Report UMINF-132.86.

[KW89] B. Kågström and L. Westin: *Generalized Schur Methods with Condition Estimators for Solving the Generalized Sylvester Equation*, *IEEE Trans. Automat. Control*, vol. 34:pp. 745–751, 1989.

[LMPR08] P. Losse, V. Mehrmann, L. Poppe and T. Reis: *The modified optimal $\mathcal{H}_\infty$ control problem for descriptor systems*, *SIAM J. Control Optim.*, vol. 47(6):pp. 2795–2811, 2008.

[LS01] W. W. Lin and J.-G. Sun: *Perturbation analysis for the eigenproblem of periodic matrix pairs*, *Lin. Alg. Appl.*, vol. 337(1-3):pp. 157–187, 2001, ISSN 0024-3795.

[LTVD00] C. T. Lawrence, A. L. Tits and P. Van Dooren: *A fast algorithm for the computation of an upper bound on the μ-norm*, *Automatica*, vol. 36:pp. 449–456, 2000.

[LVDX98] W. W. Lin, P. Van Dooren and Q. F. Xu: *Equivalent characterizations of periodical invariant subspaces*, Tech. Rep., National Centre of Theoretical Sciences, National Hsinghua University, Hsinchu, Taiwan, 1998.

[Mat10a] The MathWorks™, Inc., Natick, MA, USA: *MATLAB® 7 C/C++ and Fortran API Reference*, Mar. 2010.
URL http://www.mathworks.com/access/helpdesk/help/pdf_doc/matlab/apiref.pdf

[Mat10b] The MathWorks™, Inc., Natick, MA, USA: *MATLAB® 7 External Interfaces*, Mar. 2010.
URL http://www.mathworks.com/access/helpdesk/help/pdf_doc/matlab/apiext.pdf

[Meh99] C. Mehl: *Compatible Lie and Jordan algebras and applications to structured matrices and pencils*, Ph.D. Thesis, Chemnitz University of Technology, Faculty of Mathematics, Germany, Berlin, 1999.

[Meh00] C. Mehl: *Condensed Forms for Skew-Hamiltonian/Hamiltonian Pencils*, *SIAM J. Matrix Anal. Appl.*, vol. 21(2):pp. 454–476, 2000, ISSN 0895-4798.

[MS05] V. Mehrmann and T. Stykel: *Balanced Truncation Model Reduction for Large-Scale Systems in Descriptor Form*, in *Dimension Reduction of Large-Scale Systems* (ed. by P. Benner, V. Mehrmann and D. Sorensen), vol. 45 of *Lecture Notes in Computational Science and Engineering*, chap. 3, pp. 89–116, Springer-Verlag, Berlin, Heidelberg, New York, 2005, ISBN 978-3-540-24545-2.

[NR07] T. Noda and A. Ramirez: *z-Transform-Based Methods for Electromagnetic Transient Simulations*, *IEEE Trans Power Del.*, vol. 22(3):pp. 1799–1805, Jul. 2007.

[Pre09] R. Pregla: *Grundlagen der Elektrotechnik*, Hüthig Verlag, Heidelberg, 8th edition, 2009.

[PST09] L. Poppe, C. Schröder and I. Thies: *PEPACK: A Software Package for computing the Numerical Solution of palindromic and even eigenvalue problems using the Pencil Laub Trick*, Tech. Rep., Institut für Mathematik, Technische Universität Berlin, Germany, Oct. 2009, Preprint 22-2009.

[Rei09] T. Reis: *Model Reduction of Electrical Circuits*, Sept. 2009, Casa Autumn School on Future Developments in Model Order Reduction, Terschelling, The Netherlands.
URL http://www.win.tue.nl/casa/meetings/special/mor09/reis.pdf

[Sch07] M. Schmidt: *Systematic Discretization of Input/Output Maps and other Contributions to the Control of Distributed Parameter Systems*, Ph.D. Thesis, Institut für Mathematik, Technische Universität Berlin, Germany, May 2007.

[Sch08a] A. Schneider: *Matrix decomposition based approaches for model order reduction of linear systems with a large number of terminals*, Diploma Thesis, Chemnitz University of Technology, Faculty of Mathematics, Germany, 2008.

[Sch08b] C. Schröder: *Palindromic and Even Eigenvalue Problems — Analysis and Numerical Methods*, Ph.D. Thesis, Institut für Mathematik, Technische Universität Berlin, Germany, May 2008.

[Sim06] V. Sima: *Efficient Algorithm for L_∞-Norm Calculations*, Jul. 2006, Preprints of 5th IFAC Symposium on Robust Control Design, Toulouse, France. Invited Session MD001 - "Advances in numerical algorithms for robust control and its applications".

[Sim09] V. Sima: *Report on the research visit at the Technische Universität Chemnitz*, Jul. 2009.

[Sok06] V. I. Sokolov: *Contributions to the Minimal Realization Problem for Descriptor Systems*, Ph.D. Thesis, Chemnitz University of Technology, Faculty of Mathematics, Germany, 2006.

[Sty02] T. Stykel: *Analysis and Numerical Solution of Generalized Lyapunov Equations*, Ph.D. Thesis, Institut für Mathematik, Technische Universität Berlin, Germany, Jun. 2002.

[Sty06] T. Stykel: *On some norms for descriptor systems*, *IEEE Trans. Automat. Control*, vol. 51(5):pp. 842–847, May 2006.

[Toi02] H. Toivonen: *Signal and system norms*, in *Lecture Notes on Robust Control by State-Space Methods*, chap. 2, Åbo, 2002.
URL http://users.abo.fi/htoivone/courses/robust/rob2.pdf

[TT09] Y. Tian and Y. Takane: *The inverse of any two-by-two nonsingular partitioned matrix and three matrix inverse completion problems*, *Computers Math. Appl.*, vol. 57(8):pp. 1294–1304, Apr. 2009.

[Uni01] University of Tennessee, Knoxville, Tennessee, USA: *Basic Linear Algebra Subprograms Technical (BLAST) Forum Standard*, Aug. 2001.
URL http://www.netlib.org/blas/blast-forum/blas-report.pdf

[Var90] A. Varga: *Computation of irreducible generalized state-space realizations*, *Kybernetika*, vol. 26(2):pp. 89–106, 1990.

[Var00] A. Varga: *A Descriptor System Toolbox for MATLAB*, in *Proc. of the IEEE International Symposium on Computer Aided Control System Design*, pp. 150–155, CACSD'2000, Anchorgae, Alaska, 2000.

[War81] R. C. Ward: *Balancing the Generalized Eigenvalue Problem*, *SIAM J. Sci. Comput.*, vol. 2:pp. 141–152, 1981.

[Wei97] J. Weickert: *Applications of the Theory of Differential-Algebraic Equations to Partial Differential Equations of Fluid Dynamics*, Ph.D. Thesis, Chemnitz University of Technology, Faculty of Mathematics, Germany, 1997.

[Xu06] H. Xu: *On equivalence of pencils from discrete-time and continuous-time control*, *Lin. Alg. Appl.*, vol. 414(1):pp. 97–124, Apr. 2006, ISSN 0024-3795.

[ZD98] K. Zhou and J. D. Doyle: *Essentials of Robust Control*, Prentice Hall, 1st edition, 1998, ISBN 978-0-13525-833-0.

Theses

1. In this diploma thesis we extend an existing algorithm for the computation of the $\mathcal{L}_\infty$-norm of standard state space systems to descriptor systems and present different approaches to increase reliability and accuracy of the results.

2. We derive a numerical method which tests if the transfer function obtained by a given descriptor system is proper or improper. This method is additionally used to reduce the order of the given descriptor system in order to reduce the costs for computing the $\mathcal{L}_\infty$-norm.

3. For standard state space systems the computation of the $\mathcal{L}_\infty$-norm is related to the computation of the eigenvalues of certain Hamiltonian matrices. We extend this approach and use skew-Hamiltonian/Hamiltonian matrix pencils in the descriptor system case.

4. By extending the considered skew-Hamiltonian/Hamiltonian matrix pencils to skew-Hamiltonian/Hamiltonian matrix pencils of larger dimensions it is possible to built these directly from the original data without explicitely forming matrix products and inverses. In this way we increase reliability and accuracy of the computed eigenvalues.

5. In the discrete-time case we have to consider symplectic matrix pencils instead of skew-Hamiltonian/Hamiltonian ones. We apply the extension strategy from the skew-Hamiltonian/Hamiltonian case and transform the resulting matrix pencils to more convenient structures in order to apply structure-exploiting eigenvalue solvers.

6. We derive and explain a new structure-preserving algorithm to compute the eigenvalues of skew-Hamiltonian/Hamiltonian matrix pencils. In particular, simple, purely imaginary eigenvalues do not experience any numerical error in their imaginary parts. Hence, by applying this method we can ensure reliability of the $\mathcal{L}_\infty$-norm algorithm since there purely imaginary eigenvalues are the ones of interest.

7. The numerical results of the properness testing procedure are not completely satisfactory. One the one hand we experience some problems with the `SLICOT` routine `TG01HX` as not all uncontrollable or unobservable poles are removed. On the other

hand, rank decisions are observed not to be robust. We have to choose the tolerances for determining the numerical ranks very carefully to obtain good results. However, this algorithm yields a good reduction of the systems' orders for our examples which can be used to accelerate the computation of the $\mathcal{L}_\infty$-norm.

8. The numerical results for the $\mathcal{L}_\infty$-norm algorithm are very satisfactory. We observe fast convergence and very high accuracy.

9. Also the experimental results of the new structure-preserving method for the computation of the eigenvalues of skew-Hamiltonian/Hamiltonian matrix pencils are very satisfactory. Purely imaginary eigenvalues do not experience any error in the imaginary parts and the new algorithm is in general faster than the QZ algorithm.

Declaration of Authorship

Hereby I certify that I have completed the present thesis independently. I have not submitted it previously for examination purposes and I have used no others than the stated references. All consciously used excerpts, quotations and contents of other authors have been properly marked as those.

Chemnitz, 9th June 2010

Selbstständigkeitserklärung

Hiermit erkläre ich, daß ich die vorliegende Arbeit selbstständig angefertigt, nicht anderweitig zu Prüfungszwecken vorgelegt und keine anderen als die angegebenen Hilfsmittel verwendet habe. Sämtliche wissentlich verwendete Textausschnitte, Zitate oder Inhalte anderer Verfasser wurden ausdrücklich als solche gekennzeichnet.

Chemnitz, den 9. Juni 2010

Matthias Voigt