On Fixed Order $\mathcal{H}_\infty$-Controller Design for Delay Systems

Masterarbeit

Paul Schwerdtner

Gutachter
Prof. Dr. Volker Mehrmann
Dr. Matthias Voigt (Betreuer)

eingereicht am 20. April 2018

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und eigenhändig sowie ohne unerlaubte fremde Hilfe und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.

Berlin, den 20. April 2018

Paul Schwerdtner

Contents

Abstract

We develop and implement an algorithm to compute fixed order controllers for delay systems that minimize the $\mathcal{H}_\infty$-norm of the corresponding closed loop transfer function. Specifically, we alter the controller parameters of a given fixed order controller with possibly structural constraints to optimize the $\mathcal{H}_\infty$-performance of the resulting closed loop transfer function using direct numerical optimization. We use a recently developed algorithm for the computation of $\mathcal{H}_\infty$-norms of irrational functions such as transfer functions of systems with delay. It is based on rational interpolation using the Loewner matrix framework. We illustrate the efficiency of the developed method with several numerical test examples and show that our method outperforms the currently available algorithm for this purpose.

After that, the applicability of the developed method for a broader class of systems is shown and the realization independence of the utilized algorithm for the $\mathcal{H}_\infty$-norm computation is exploited to also impose performance requirements on transfer functions of closed loop systems using weight functions in the delay setting.

Zusammenfassung

Gegenstand dieser Masterarbeit ist die Entwicklung und Implementierung eines Algorithmus zur Berechnung von Reglern mit festgelegter Dimension für Systeme mit Zeitverzögerung mit dem Ziel, die $\mathcal{H}_\infty$-Norm der Übertragungsfunktion des geschlossenen Regelkreises zu minimieren. Insbesondere werden die Reglerparameter unter Berücksichtigung möglicher Strukturvorgaben entsprechend eines Verfahrens zur numerischen Optimierung so variiert, dass die $\mathcal{H}_\infty$-Performance des geschlossenen Regelkreises optimiert wird. Es kommt ein kürzlich entwickelter Algorithmus für die Berechnung der $\mathcal{H}_\infty$-Norm irrationaler Funktionen, zu denen auch Übertragungsfunktionen von Systemen mit Zeitverzögerung gehören, zum Einsatz. Dieser basiert auf rationaler Interpolation, die mit Loewner-Matrizen durchgeführt wird. Die Effizienz der entwickelten Methode wird anhand mehrerer numerischer Beispiele und im Vergleich mit der aktuell für die Regleroptimierung bei Systemen mit Zeitverzögerung verfügbaren Methode gezeigt.

Anschließend wird die Anwendbarkeit des Verfahrens für eine allgemeinere Klasse von Systemen mit Zeitverzögerung gezeigt und die Realisierungsunabhängigkeit des Verfahrens, das für die Berechnung der $\mathcal{H}_\infty$-Norm verwendet wird, ausgenutzt, um mit Hilfe von Gewichtsfunktionen Anforderungen an die Übertragungsfunktionen im geschlossenen Regelkreis für Systeme mit Zeitverzögerung zu definieren.

1 Introduction

Mathematical models play an important role throughout the natural and engineering sciences when it comes to the investigation of system behavior. A mathematical model is the representation of a system and can be used to predict how it is affected by external influences or how the system evolves under certain initial conditions. When the focus of the investigation is on the dynamic behavior of a system, in many cases differential equations are used. Therein lies the assumption that the change of the system's state depends only on its current state as well as the current value of the external influences.

However, many phenomena such as transport phenomena in fluid dynamics, communication lags or the incubation period when modeling the spread of an infectious disease suggest that in order to obtain a more realistic model, the influence of past states needs to be considered, as well. Even when no intrinsic influence of past states is present in a given system, it is often introduced as feedback delay when applying control.

One of the main reasons for creating a model in engineering science is to obtain a controller that is used to make the system behave in a desired way. The desired behavior is often specified in terms of $\mathcal{H}_\infty$-norms of closed loop transfer functions. For systems that are modeled with differential equations there exist algorithms that are used to compute controllers that fulfill the given requirements using Riccati-equations or linear matrix inequalities as shown in [48]. The dimension of the controller obtained with these methods is larger than or equal to the dimension of the given model which is undesired in the practical use of controllers. A design of fixed order controllers that locally minimize the $\mathcal{H}_\infty$-norm can be achieved by using numerical optimization. This is done in [11, 21] utilizing a nonsmooth and nonconvex numerical optimization algorithm. In this way, it is possible for the user to set the controller dimension and even specify fixed parameters within the controller. The rest of the controller parameters is then tuned such that the $\mathcal{H}_\infty$-norm of the closed loop transfer function is minimized.

This direct optimization approach can be extended to find controllers for delay systems, as well. First investigations towards that goal were carried out in [22, 23] however in order to have an efficient optimization approach, it is necessary to use a fast method for the computation of the objective, i. e. the $\mathcal{H}_\infty$-norm of a closed loop transfer function of a system with delay. In [38] the efficiency of the computation of $\mathcal{H}_\infty$-norms of irrational functions such as transfer functions of systems with delay is sharply increased using rational interpolation.

The emphasis of this thesis is to create an efficient algorithm for $\mathcal{H}_\infty$-controller design for delay systems with the direct optimization approach using the method developed in [38] to perform the $\mathcal{H}_\infty$-norm computation inside the optimization loop. Therefore,

at first in Chapter 2, basic results for delay differential equations are presented and linear control systems as well as methods for numerical optimization are introduced. In Chapter 3, an introduction to the topic of $\mathcal{H}_\infty$-control is given and the recent advances in $\mathcal{H}_\infty$-norm computation are explained followed by a description of the developed method in Chapter 4. After that, insights into implementation details are given in Chapter 5. Also, in this chapter numerical experiments are conducted and a comparison between the method developed in this thesis and previous approaches taken in [22, 23] is made. Finally, in Chapter 6, a generalization of the method to a broader class of delay systems is given and possible applications for the developed algorithm are shown.

2 Preliminaries

In this chapter basic notions and results concerning delay differential equations and control systems are introduced, followed by a brief overview of numerical optimization.

2.1 Delay Differential Equations

Delay differential equations (DDEs) play an important role in the modeling of technical or biological systems. In [40] some examples for applications of delay differential equations are given. A profound introduction into the theory of delay differential equations can be found in [25]. Most of the results and notions presented in this section are therefore taken from [25].

Definition 2.1. *Let $\tau \geq 0$ and $C(\mathcal{I}, \mathbb{C}^n)$ be the Banach space of continuous functions mapping the interval $\mathcal{I} \subseteq \mathbb{R}$ into $\mathbb{C}^n$ equipped with the supremum-norm, i. e. $\|\phi\|_{C(\mathcal{I},\mathbb{C}^n)} = \sup_{\theta\in\mathcal{I}} \|\phi(\theta)\|_2$. Then, if $x \in C([\sigma-\tau, \sigma+\alpha], \mathbb{C}^n)$ for $\sigma \in \mathbb{R}, \alpha > 0$, we define $x_t \in C([-\tau, 0], \mathbb{C}^n)$ by $x_t(\theta) = x(t+\theta)$ for $-\tau \leq \theta \leq 0$. If further $f : \mathbb{R} \times C([-\tau, 0], \mathbb{C}^n) \to \mathbb{C}^n$ is a given function and $\dot{x}$ represents the derivative of x with respect to t, the equation*

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

is a retarded functional differential equation *(RFDE).*

Note that (2.1) is a special case of a more general class of *neutral functional differential equations* of the form

$$\frac{\mathrm{d}}{\mathrm{d}t} D(t, x_t) = f(t, x_t),$$

where $D : \mathbb{R} \times C([-\tau, 0], \mathbb{C}^n) \to \mathbb{C}^n$ is a given function that is atomic (as defined in [25, Definition 5.3]) at zero. In this thesis, we are only concerned with RFDEs.

A function x is a solution of (2.1) on $[\sigma-\tau, \sigma+\alpha]$ if $x \in C([\sigma-\tau, \sigma+\alpha], \mathbb{C}^n)$ and $x(t)$ satisfies equation (2.1) for $t \in [\sigma, \sigma+\alpha)$. For given $\sigma \in \mathbb{R}, \phi \in C([-\tau, 0], \mathbb{C}^n)$, we say $x(\cdot, \sigma, \phi) \in C([-\tau+\sigma, \sigma+\alpha], \mathbb{C}^n)$ is a solution of (2.1) with initial function ϕ.

Equation (2.1) is called *linear autonomous* if f is given by $f(t, x_t) = Lx_t$ where $L : C([-\tau, 0], \mathbb{C}^n) \to \mathbb{C}^n$ is linear. If we look for nontrivial exponential solutions

$$x(t) = \mathrm{e}^{\lambda t} v, \quad v \in \mathbb{C}^n, \quad v \neq 0$$

and use the notation (as in [40]) $\exp_\lambda : [-\tau, 0] \to \mathbb{C}$ for the function $\exp_\lambda(\theta) = \mathrm{e}^{\lambda\theta}$, we get

$$\dot{x}(t) = \lambda \mathrm{e}^{\lambda t} v = L(\mathrm{e}^{\lambda t} \exp_\lambda v) = \mathrm{e}^{\lambda t} L(\exp_\lambda v).$$

Since $e^{\lambda t}$ is nonzero for all values of λ and t, this is equivalent to

$$\lambda v = L(\exp_\lambda v). \tag{2.2}$$

Expressing v in the standard basis for $\mathbb{C}^n$ we get $v = \sum_{k=1}^{n} v_k e_k$ and have that

$$L(\exp_\lambda v) = \sum_{k=1}^{n} v_k L(\exp_\lambda e_k).$$

Thus, we can define a matrix $L_\lambda \in \mathbb{C}^{n\times n}$ as

$$[L_\lambda]_{ij} = L_i(\exp_\lambda e_j)$$

where $L_i(\cdot)$ denotes the i-th entry of $L(\cdot)$. Then we have that $L(\exp_\lambda v) = L_\lambda v$ and therefore, a nontrivial function x of the form $x(t) = \mathrm{e}^{\lambda t} v$ is a solution of the *linear autonomous* RFDE, if and only if λ is a solution of its *characteristic equation* given by

$$\det(\lambda I - L_\lambda) = 0. \tag{2.3}$$

We are concerned with the solutions of the characteristic equation since they are used to investigate the stability of the trivial solution of a *linear autonomous* RFDE.

Definition 2.2. *Let $f(t, \hat{0}) = 0$ for all $t \in \mathbb{R}$, where $\hat{0}$ denotes the zero function in $C([-\tau, 0], \mathbb{R}^n)$. The zero solution of $\dot{x} = f(t, x_t)$ is called* stable *if for any $\sigma \in \mathbb{R}$, $\varepsilon > 0$, there exists a $\delta > 0$ such that for $\|\phi\|_{C([-\tau,0],\mathbb{C}^n)} < \delta$ all solutions x of $\dot{x} = f(t, x_t)$ satisfy $\|x_t(\cdot, \sigma, \phi)\|_{C([-\tau+\sigma,\sigma],\mathbb{C}^n)} < \varepsilon$ for all $t \geq \sigma$. The zero solution is called* asymptotically stable *if it is stable and for all solutions $x(\cdot, \sigma, \phi)$ it holds that $\lim_{t\to\infty} x(t, \sigma, \phi) = 0$. If the solution is not stable, it is called* unstable.

The connection between the solutions of the characteristic equation *(characteristic roots)* and the stability properties of the trivial solution is established in the following theorem which is proved in [25, Chapter 1, Theorem 5.2].

Theorem 2.3. *Let $\Lambda(L_\lambda)$ be the set of all solutions of $\det(\lambda I - L_\lambda)$. If $\max_{\lambda\in\Lambda(L_\lambda)} \mathrm{Re}(\lambda) < \mu$, then there is a $K > 0$ such that for the solution of the RFDE*

$$\dot{x} = Lx_t, \quad x_\sigma = \phi,$$

it holds that

$$\|x(t, \sigma, \phi)\|_2 \leq K \mathrm{e}^{\mu t} \|\phi\|_{C([-\tau,0],\mathbb{C}^n)}$$

for all $t \geq \sigma$ and $\phi \in C([-\tau, 0], \mathbb{C}^n)$.

This implies that the trivial solution of a linear autonomous RFDE is asymptotically stable if all its characteristic roots are in the open left half plane or, more formally, if $\Lambda(L_\lambda) \subset \mathbb{C}^-$, where $\mathbb{C}^- := \{s \in \mathbb{C} \mid \mathrm{Re}(s) < 0\}$. This result for linear autonomous systems carries over from ordinary differential equations (ODEs). Also, we can deduce that stability of the trivial solution is not dependent on initial conditions but a system property in the linear autonomous case (the same holds for ODEs). However, to find the characteristic roots, a nonlinear eigenvalue problem must be solved which generically has infinitely many solutions. In the following we are mostly concerned with the asymptotic stability of a system. The next theorems about the properties of $\Lambda(L_\lambda)$ are essential for the approach to influence the stability of a system that is described in Chapter 4.

Theorem 2.4. *Given $r \in \mathbb{R}$, there are at most finitely many characteristic roots $\lambda \in \Lambda(L_\lambda)$ satisfying $\mathrm{Re}(\lambda) > r$. If there are infinitely many distinct characteristic roots $\{\lambda_n\}_{n\in\mathbb{N}}$ sorted increasingly by absolute value, then*

$$\mathrm{Re}(\lambda_n) \to -\infty \text{ as } n \to \infty$$

A proof can be found in [40, Lemma 4.2]. This implies that if one is concerned with the stability of a linear RFDE, only finitely many characteristic roots need to be considered. The other roots do not affect the stability of the system because their real parts are too small. In what follows, we will focus on *linear autonomous* RFDEs with a discrete delay, such that f in equation (2.1) is of the form

$$f(t, x_t) = A_1 x(t) + A_2 x(t - \tau).$$

In this case, the characteristic equation is given by

$$\det\left(\lambda I - A_1 - \mathrm{e}^{-\lambda\tau} A_2\right) = 0. \tag{2.4}$$

A relation of the solutions of (2.4) to the eigenvalues of $A_1 + A_2$ which coincide with the characteristic roots of the system with $\tau = 0$ is presented in the following theorem [40, Theorem 4.4].

Theorem 2.5. *Let Λ_0 be set of eigenvalues of $A_1 + A_2$. Let $\delta > 0$ and let $\eta \in \mathbb{R}$ such that $\eta < \min_{\lambda\in\Lambda_0} (\mathrm{Re}(\lambda))$. Then there exists a $\tau_0 \in \mathbb{R}^+$ such that if $\tau \in (0, \tau_0]$ and $\det\left(\lambda I - A_1 - \mathrm{e}^{-\lambda\tau} A_2\right) = 0$, then either $\mathrm{Re}(\lambda) < \eta$ or $|\lambda - \lambda_0| < \delta$ for some $\lambda_0 \in \Lambda_0$.*

This implies that for small delay, the characteristic roots are either close to the eigenvalues of the system without delay or they are left of the spectrum Λ_0 of the system without delay. Therefore, small delays do not affect the asymptotic stability of the system. This is further investigated in Section 4.3.1 and the delay at which a system is destabilized is computed. The effect of large delays on the stability of a system is studied in [29].

2.2 Control Problems

As in [15] consider the set of equations

$$\mathcal{P}: \begin{cases} \dot{x}(t) = Ax(t) + B_1 w(t) + Bu(t), \quad x(0) = x_0, \\ z(t) = C_1 x(t) + D_{11} w(t) + D_{12} u(t), \\ y(t) = Cx(t) + D_{21} w(t) + D_{22} u(t), \end{cases} \tag{2.5}$$

where $x(t) \in \mathbb{R}^{n_x}, u(t) \in \mathbb{R}^{n_u}, y(t) \in \mathbb{R}^{n_y}, z(t) \in \mathbb{R}^{n_z}, w(t) \in \mathbb{R}^{n_w}$ describe the state, control input, measured output, regulated output, and noise input, respectively, and $A \in \mathbb{R}^{n_x \times n_x}, B_1 \in \mathbb{R}^{n_x \times n_w}, B \in \mathbb{R}^{n_x \times n_u}, C_1 \in \mathbb{R}^{n_z \times n_x}, D_{11} \in \mathbb{R}^{n_z \times n_w}, D_{12} \in \mathbb{R}^{n_z \times n_u}, C \in \mathbb{R}^{n_y \times n_x}, D_{21} \in \mathbb{R}^{n_y \times n_w}$, and $D_{22} \in \mathbb{R}^{n_y \times n_u}$. The control problem is to choose an input $u : \mathbb{R}^+ \to \mathbb{R}^{n_u}$ such that the regulated output $z : \mathbb{R}^+ \to \mathbb{R}^{n_z}$ behaves in a desired manner. Since the input noise $w : \mathbb{R}^+ \to \mathbb{R}^{n_w}$ is generically unknown it is not advisable to define u a priori but instead make u a function of the measured output $y : \mathbb{R}^+ \to \mathbb{R}^{n_y}$ which leads to the concept of *feedback control*.

The problem can be divided into the *regulator problem* which is to choose u such that the magnitude of the regulated output is minimized and thus counteract the effect of the noise input and the *servo problem* where the control input is chosen such that the regulated output is kept close to a given path $r_1 : \mathbb{R}^+ \to \mathbb{R}^{n_z}$. Since the latter problem can be transformed into the prior problem by introducing the reference r_1 into the system equations and in this way minimizing $\|z - r_1\|_{\mathcal{L}_2}$ we are mostly concerned with the *regulator problem*.

In this work *output feedback control* is used to address the regulator problem, which is one of many methods that were developed to determine values for u using the measured output y in order to control z.

When applying output feedback control, the control input u is defined as the output of the dynamical system given by

$$\mathcal{K}: \begin{cases} \dot{x}_c(t) = A_c x_c(t) + B_c y(t), \quad x_c(0) = 0, \\ u(t) = C_c x_c(t) + D_c y(t), \end{cases} \tag{2.6}$$

where $x_c(t) \in \mathbb{R}^{n_c}$ is the controller state and $A_c \in \mathbb{R}^{n_c \times n_c}, B_c \in \mathbb{R}^{n_c \times n_y}, C_c \in \mathbb{R}^{n_u \times n_c}$, and $D_c \in \mathbb{R}^{n_u \times n_y}$. The input to the controller system is the measured output of (2.5). If the controller state dimension n_c is equal to zero, we have that $u(t) = D_c y(t)$ which is considered *static feedback control*. In this case (if further $D_{22} = 0$ for simplicity) we can rewrite (2.5) such that

$$\begin{aligned} \dot{x} &= \underbrace{(A + BD_cC)}_{\tilde{A}} x(t) + B_1 w(t), \\ z &= (C_1 + D_{12} D_c C) x(t) + (D_{11} + D_{12} D_c D_{21}) w(t). \end{aligned} \tag{2.7}$$

In this way, the *closed loop system* is created and its properties can be influenced by adjusting the controller matrices. For example, for $n_c = 0$ it is a common objective to find a D_c such that the eigenvalues of $\tilde{A}$ in (2.7) are at desired locations.

However, the static controller that leads to the closed loop system (2.7) is often not sufficient to obtain the desired results such as asymptotic stability of the closed loop system which is why we are concerned with the general case for output feedback, where $n_c \geq 0$. For this, the system can be rewritten, as in [30], as

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}x(t)\\ x_c(t)\end{bmatrix} &= \underbrace{\begin{bmatrix}A + BD_cZ_1C & B_2Z_2C_c\\ B_cZ_1C_2 & A_c + B_cZ_1D_{22}C_c\end{bmatrix}}_{\tilde{A}}\underbrace{\begin{bmatrix}x(t)\\ x_c(t)\end{bmatrix}}_{\tilde{x}} + \underbrace{\begin{bmatrix}B + B_1D_cZ_1D_{21}\\ B_cZ_1D_{21}\end{bmatrix}}_{\tilde{B}} w(t),\\ z(t) &= \underbrace{\begin{bmatrix}C + D_{12}Z_2D_cC_1 & D_{12}Z_2C_c\end{bmatrix}}_{\tilde{C}}\begin{bmatrix}x(t)\\ x_c(t)\end{bmatrix} + \underbrace{(D_{11} + D_{12}D_cZ_1D_{21})}_{\tilde{D}} w(t),\end{aligned} \tag{2.8}$$

with $Z_1 = (I - D_{22}D_c)^{-1}$ and $Z_2 = (I - D_cD_{22})^{-1}$. Note that if either the matrix $I - D_{22}D_c$ or $I - D_cD_{22}$ is not invertible, then the feedback system is not well-posed by [48, Definition 5.2]. Therefore, we are only concerned with the case that the matrices $I - D_{22}D_c$ and $I - D_cD_{22}$ are invertible. We denote the arrangement of controller and plant matrices which is given in (2.8) by $\mathbf{CL}(\mathcal{P}, \mathcal{K})$. The transfer function of the closed loop system, which is introduced in the following section, is denoted as $H_{\mathcal{P},\mathcal{K}}$. The main goal of output feedback control is that the closed loop system is internally stable; i. e., the trivial solution of the homogeneous system

$$\dot{\tilde{x}} = \tilde{A}\tilde{x}$$

is asymptotically stable.

Also, it is desired that the regulated output is not sensitive to the noise input w for all possible bounded noise input signals. To express this objective in a meaningful way we use the transfer function of the closed loop system.

2.3 Laplace Transformation and Transfer Functions

The state space description (2.8) of the closed loop dynamical system does not directly reveal the reaction of the regulated output under the effect of some noise input. However, we wish to minimize the magnitude of z, also called system response, for any w. Therefore, it is desirable to estimate the magnitude of z as a function of w. This can be done using the *Laplace transform*.

Definition 2.6. *If a function $g : [0, \infty) \to \mathbb{R}$ is locally integrable and there exist constants $\alpha, \beta \in \mathbb{R}$ such that $|g(t)| \leq \alpha \mathrm{e}^{\beta t}$ for all $t \in [0, \infty)$, then the* Laplace transform *$\mathcal{L}(g)$ is defined by*

$$\mathcal{L}(g)(s) = \int_0^\infty \mathrm{e}^{-st} g(t)\,\mathrm{d}t.$$

The Laplace transform is particularly useful for differential equations, since $\mathcal{L}(\dot{g})(s) = s\mathcal{L}(g)(s) - g(0)$ which can easily be verified using integration by parts. In this way, differential equations are transformed into algebraic equations. Consider again the closed loop system (2.8). If we apply the Laplace transformation to both equations, we obtain

$$\begin{aligned} sX(s) &= \tilde{A}X(s) + \tilde{B}W(s) \\ Z(s) &= \tilde{C}X(s) + \tilde{D}W(s), \end{aligned}$$

where X, W, and Z denote the Laplace transforms of $\tilde{x}, w$, and z, respectively, if $\tilde{x}(0) = 0$, which is assumed throughout this section. Solving the first equation for $X(s)$ and inserting $X(s)$ into the second equation yields

$$Z(s) = \underbrace{\left(\tilde{C}\big(sI - \tilde{A}\big)^{-1}\tilde{B} + \tilde{D}\right)}_{H(s)} W(s), \tag{2.9}$$

where H is called the *transfer function* from the noise input to the regulated output. In order to classify different transfer functions we define the Banach spaces

$$\mathcal{L}_\infty^{p\times m} := \left\{ H|_{\mathrm{i}\mathbb{R}} \;\middle|\; H : \Omega \to \mathbb{C}^{p\times m} \text{ is analytic for an open domain } \Omega \subseteq \mathbb{C} \text{ with } \mathrm{i}\mathbb{R} \subset \Omega \text{ and } \sup_{\omega\in\mathbb{R}} \|H(\mathrm{i}\omega)\|_2 < \infty \right\},$$

$$\mathcal{H}_\infty^{p\times m} := \left\{ H : \mathbb{C}^+ \to \mathbb{C}^{p\times m} \;\middle|\; H \text{ is analytic and } \sup_{s\in\mathbb{C}^+} \|H(s)\|_2 < \infty \right\},$$

where $\mathbb{C}^+ := \{s \in \mathbb{C} \mid \mathrm{Re}(s) > 0\}$.

Transfer functions of a linear dynamical systems such as (2.8) with the structure of H in (2.9) are rational functions which can be shown by inverting $sI - \tilde{A}$ using Cramer's rule. They can only have a pole at $\lambda \in \mathbb{C}$ when λ is an eigenvalue of $\tilde{A}$ or, in the terminology of the previous section, if λ is a characteristic root of the homogeneous system $\dot{x} = \tilde{A}x$. Thus, all transfer functions $H : \Omega \to \mathbb{C}^{p\times m}$, with $\mathrm{i}\mathbb{R} \subset \Omega$, of linear systems with no characteristic root on the imaginary axis are in $\mathcal{L}_\infty^{p\times m}$. For such, the $\mathcal{L}_\infty$-norm is given by

$$\|H\|_{\mathcal{L}_\infty} := \sup_{\omega\in\mathbb{R}} \|H(\mathrm{i}\omega)\|_2 = \sup_{\omega\in\mathbb{R}} \sigma_{\max}(H(\mathrm{i}\omega)), \tag{2.10}$$

where $\sigma_{\max}(\cdot)$ denotes the largest singular value of its matrix argument. If furthermore, all characteristic roots are in $\mathbb{C}^-$, then we have $H \in \mathcal{H}_\infty^{p\times m}$. The corresponding norm is given by

$$\|H\|_{\mathcal{H}_\infty} := \sup_{s\in\mathbb{C}^+} \|H(s)\|_2 = \sup_{s\in\mathbb{C}^+} \sigma_{\max}(H(s)). \tag{2.11}$$

It is important to note that for transfer functions of asymptotically stable systems, the $\mathcal{H}_\infty$-norm and the $\mathcal{L}_\infty$-norm are equal, i. e.

$$\|H\|_{\mathcal{H}_\infty} := \sup_{s\in\mathbb{C}^+} \|H(s)\|_2 = \sup_{\omega\in\mathbb{R}} \sigma_{\max}(H(\mathrm{i}\omega)) = \|H\|_{\mathcal{L}_\infty}.$$

The concept of transfer functions is not limited to linear first order systems as (2.8) or (2.5). Since we focus on systems with a discrete delay, note that

$$\begin{aligned}\mathcal{L}(x(\cdot-\tau))(s) &= \int_0^\infty \mathrm{e}^{-st}x(t-\tau)\,\mathrm{d}t = \int_{-\tau}^\infty \mathrm{e}^{-s(t+\tau)}x(t)\,\mathrm{d}t\\ &= \underbrace{\int_{-\tau}^0 \mathrm{e}^{-s(t+\tau)}x(t)\,\mathrm{d}t}_{=0,\text{ if } x\big|_{t\in[-\tau,0]}\equiv 0} + \int_0^\infty \mathrm{e}^{-s(t+\tau)}x(t)\,\mathrm{d}t = \mathrm{e}^{-s\tau}\mathcal{L}(x)(s).\end{aligned}$$

In this way, for zero initial conditions we have that the transfer function of a system with internal delay such as

$$\begin{aligned}\dot{x}(t) &= A_1x(t) + A_2x(t-\tau) + Bw(t), \qquad x|_{[-\tau,0]} \equiv 0,\\ z(t) &= Cx(t) + Dw(t),\end{aligned}$$

is given by

$$H(s) = C(sI - A_1 - \mathrm{e}^{-s\tau}A_2)^{-1}B + D.$$

Note that again H is analytic in λ, if λ is not a characteristic root of the homogeneous system

$$\dot{x}(t) = A_1x(t) + A_2x(t-\tau)$$

such that for an asymptotically stable system we still have $H \in \mathcal{H}_\infty^{p\times m}$.

The usefulness of the $\mathcal{H}_\infty$-norm as a measure for the influence of noise input on the regulated output of asymptotically stable systems is established in the following theorem (cf. [48, Theorem 4.4]).

Theorem 2.7. *Let $H \in \mathcal{H}_\infty^{p\times m}$ be a transfer function of the dynamical system (2.8) such that $Z(s) = H(s)W(s)$. Then it holds that*

$$\|z\|_{\mathcal{L}_2} \le \|H\|_{\mathcal{H}_\infty} \cdot \|w\|_{\mathcal{L}_2}.$$

Proof. Using Plancherel's theorem (cf. [36]), we obtain

$$\|f\|_{\mathcal{L}_2}^2 = \frac{1}{2\pi}\int_{-\infty}^\infty \|F(\mathrm{i}\omega)\|_2^2\,\mathrm{d}\omega,$$

where F denotes the FOURIER transform of f, which is defined by

$$F(\mathrm{i}\omega) := \int_{-\infty}^{\infty} f(t)\mathrm{e}^{-\mathrm{i}\omega t}\,\mathrm{d}t.$$

Taking into account that for functions $f \in \mathcal{L}_1(\mathbb{R}, \mathbb{R}^n) \cap \mathcal{L}_2(\mathbb{R}, \mathbb{R}^n)$ with $f(t) = 0$ for all $t < 0$ the FOURIER transform of f coincides with its Laplace transform restricted to the imaginary axis, we have that

$$\begin{aligned}
\|z\|_{\mathcal{L}_2}^2 &= \frac{1}{2\pi}\int_{-\infty}^{\infty} \|Z(\mathrm{i}\omega)\|_F^2\,\mathrm{d}\omega = \frac{1}{2\pi}\int_{-\infty}^{\infty} \|Z(\mathrm{i}\omega)\|_2^2\,\mathrm{d}\omega \\
&= \frac{1}{2\pi}\int_{-\infty}^{\infty} \|H(\mathrm{i}\omega)W(\mathrm{i}\omega)\|_2^2\,\mathrm{d}\omega \leq \int_{-\infty}^{\infty} \|H(\mathrm{i}\omega)\|_2^2 \cdot \|W(\mathrm{i}\omega)\|_2^2\,\mathrm{d}\omega \\
&\leq \sup_{\omega\in\mathbb{R}} \|H(\mathrm{i}\omega)\|_2^2 \cdot \frac{1}{2\pi}\int_{-\infty}^{\infty} \|W(\mathrm{i}\omega)\|_2^2\,\mathrm{d}\omega \leq \|H\|_{\mathcal{H}_\infty}^2 \cdot \|w\|_{\mathcal{L}_2}^2.
\end{aligned}$$

□

Furthermore, it is shown in [48] that

$$\|H\|_{\mathcal{H}_\infty} = \sup_{w\in\mathcal{L}_2} \frac{\|z\|_{\mathcal{L}_2}}{\|w\|_{\mathcal{L}_2}}.$$

This shows that the $\mathcal{H}_\infty$-norm of the closed loop transfer function provides an upper bound on the gain between the noise input and the regulated output.

2.4 Numerical Optimization

An unconstrained optimization problem is given by

$$\min_{x\in\mathbb{R}^n} f(x), \tag{2.12}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuous function. Various numerical methods have been established to numerically determine a minimizer for the given function. While in general the global minimizer is desired, depending on the given function and initial value, such algorithms may only determine a local minimizer. Throughout this thesis, we use the extension of a quasi-Newtonian approach that is established in [13] especially addressing optimization problems that arise in controller design. An introduction to quasi-Newtonian optimization methods is given in [7]. In the following, we will briefly describe this optimization approach and reference software packages that are used throughout this thesis to perform numerical optimization.

2.4.1 Line Search

Consider the case when only an optimal scalar value for the minimization of f needs to be found. This is the case when $n = 1$ or a direction $d \in \mathbb{R}^n$ is determined and a *line search* in that direction is performed. Then a function

$$q : \mathbb{R} \to \mathbb{R}, \quad t \mapsto f(x + t \cdot d), \tag{2.13}$$

with $q'(0) < 0$, can be defined and a one dimensional optimization problem is solved to find a value t^* that minimizes q. This can be addressed by imposing the conditions

$$q(t^*) < q(0) \text{ and } |q'(t^*)| < \varepsilon,$$

where $\varepsilon > 0$ is chosen arbitrarily small. In this way a candidate for a local minimizer for q can be found. However, this t^* may be hard to compute, since many evaluations of q may be necessary until $|q'(t^*)|$ has reached a sufficiently small value. Furthermore, these conditions also apply for a local maximizer of q. Therefore, in state-of-the-art optimization methods the line search by Wolfe that is established in [42] is used. Two coefficients $0 < m_1 < m_2 < 1$ are chosen and $t^* \in [t_L, t_R]$ is bracketed in the following way:

1. if $q(t) \leq q(0) + m_1 t q'(0)$ and $q'(t) \geq m_2 q'(0)$, terminate,
2. if $q(t) > q(0) + m_1 t q'(0)$, set $t_R = t$,
3. if $q(t) \leq q(0) + m_1 t q'(0)$ and $q'(t) < m_2 q'(0)$, set $t_L = t$.

In this way, it is ensured that $q(t^*) < q(0)$ and for the derivative it holds that $q'(t^*) > q'(0)$ which is characteristic for a local minimizer. In [7, Theorem 3.7] it is shown, that this line search terminates for continuously differentiable functions q.

2.4.2 Quasi-Newton Methods

Note that in order to perform a line search on a multidimensional optimization problem, the function q as in (2.13) needs to be defined, first. For this, a *descent direction* d has to be computed. To address this task, we use the quasi-Newton method, which is described in [7, Chapter 4] and which is briefly outlined here.

Newton's method in optimization is built on a second order approximation of a given function $f : \mathbb{R}^n \to \mathbb{R}$, which is assumed to be twice continuously differentiable. Its Taylor expansion can then be given by

$$f(x_i + \Delta x) = f(x_i) + \nabla f(x_i)^\mathsf{T} \Delta x + \frac{1}{2} (\Delta x)^\mathsf{T} H(x_i) \Delta x + \mathcal{O}(|\Delta x|^3) \tag{2.14}$$

where ∇f and H denote the gradient of f and the Hessian matrix, respectively. Setting the gradient of the quadratic approximation to zero yields

$$0 = \nabla f(x_i) + H(x_i) \Delta x \quad \Leftrightarrow \quad \Delta x = -H(x_i)^{-1} \nabla f(x_i), \tag{2.15}$$

Algorithm 1 Quasi-Newtonian optimization

Input: Objective function $f : \mathbb{R}^n \to \mathbb{R}$, initial point $x \in \mathbb{R}^n$, tolerance $\varepsilon > 0$, initial symmetric, positive definite matrix $W \in \mathbb{R}^{n \times n}$

Output: Local minimizer x

1: Compute the initial gradient $g := \nabla f(x)$.
2: **while** $\|g\| > \varepsilon$ **do**
3: Compute $d := -Wg$.
4: Perform a line search initialized on $t = 1$ to obtain t^*.
5: Compute $x := x + t^* d$.
6: Compute the gradient $g := \nabla f(x)$.
7: Update the matrix W.
8: **end while**

if $H(x_i)$ is invertible. From this an algorithm can be constructed to find a local minimizer of f by letting $x_{i+1} = x_i + \Delta x$ or $x_{i+1} = x_i + t^* \Delta x$, if line search is applied as well. Since the evaluation of f can be computationally expensive and the derivatives often have to be computed using a finite difference scheme, which takes $n+1$ evaluations for the gradient and $(n+1) \cdot n$ evaluations for the Hessian matrix, the computation of the Hessian matrix is avoided for higher dimensional and computationally more involved problems. For this, quasi-Newtonian methods were introduced in which the Hessian matrix of f is approximated based on information of the gradient.

In Algorithm 1 the quasi-Newtonian method is outlined. In line 7 the inverse Hessian matrix of the objective function or its inverse, respectively, has to be approximated. We use the BFGS method that was developed independently and justified using different arguments by Broyden, Fletcher, Goldfarb, and Shanno. Here, the updated matrix W_{i+1} is given by

$$W_{i+1} = W_i - \frac{\ell_i y_i^\mathsf{T} W_i + W_i y_i \ell_i^\mathsf{T}}{y_i^\mathsf{T} \ell_i} + \left(1 + \frac{y_i^\mathsf{T} W y_i}{y_i^\mathsf{T} \ell_i}\right) \frac{\ell_i \ell_i^\mathsf{T}}{y_i^\mathsf{T} \ell_i}, \tag{2.16}$$

where $\ell_i = x_i - x_{i-1}$ and $y_i = g_i - g_{i-1}$. Note that this approximation ensures that the matrix W in Algorithm 1 is always symmetric, positive definite. An elegant derivation of this method can be found in [14].

2.4.3 Nonsmooth Optimization

In Example 3.1 it is shown that the objective function that we are concerned with throughout this thesis may not be continuously differentiable on the entire domain. In that case the line search in Algorithm 1 can be replaced by an *inexact line search* that is introduced in [28]. When performing the inexact line search as proposed in [28], the Wolfe termination conditions are replaced by the conditions

$$\begin{aligned} A(t): &\quad h(t) < m_1 b t \\ W(t): &\quad h \text{ is differentiable at } t \text{ with } h'(t) > m_2 b, \end{aligned} \tag{2.17}$$

Algorithm 2 Inexact line search

Input: Line search objective function h
Output: Minimizer t^*

Initialize $t_L := 0$, $t_R := \infty$, $t^* := 1$.
while True **do**
  if $A(t^*)$ fails **then**
    Update $t_R := t^*$.
  else if $W(t^*)$ fails **then**
    Update $t_L := t^*$.
  else
    break
  end if
  if $t_R < \infty$ **then**
    Update $t^* := (t_L + t_R)/2$.
  else
    Update $t^* := 2t_L$.
  end if
end while

where b and h are defined by

$$h : \mathbb{R}^+ \to \mathbb{R}, \quad t \mapsto f(x + td) - f(x), \tag{2.18}$$

$$b := \limsup_{t \to 0} \frac{h(t)}{t}. \tag{2.19}$$

The line search is then performed as outlined in Algorithm 2. In every iteration, the interval $[t_L, t_R]$, in which t^* is located, is decreasing in size, narrowing the interval $[t_L, r_R]$ until the modified termination conditions given in (2.17) are satisfied. The convergence properties of this line search are discussed in [28]. The algorithm outlined here is implemented in MATLAB programming language as software package `hanso`[1]. It has been extended to perform constrained optimization in [13]. This extension is implemented in the software package `granso`[2]. The latter is used for the implementation of the algorithm that is developed in this thesis.

[1] available at `https://cs.nyu.edu/overton/software/hanso/`
[2] available at `https://gitlab.com/timmitchell/GRANSO/`

3 $\mathcal{H}_\infty$-Control

Theorem 2.7 implies that the $\mathcal{H}_\infty$-norm provides an upper bound for the influence of disturbances on the regulated output and gives reason to find a controller that minimizes the $\mathcal{H}_\infty$-norm of the closed loop transfer function in order to make the influence of the noise input on the regulated output small.

3.1 Classical $\mathcal{H}_\infty$-Controller Design

The problem of finding a feedback controller that minimizes the sensitivity to noise excitation is addressed since the early 1980s ([44, 45, 46]) to explicitly address the robustness issue in controller design. In [15] a method is introduced that computes all controllers which ensure that the $\mathcal{H}_\infty$-norm of the closed loop transfer function is strictly less than a given $\gamma > 0$.

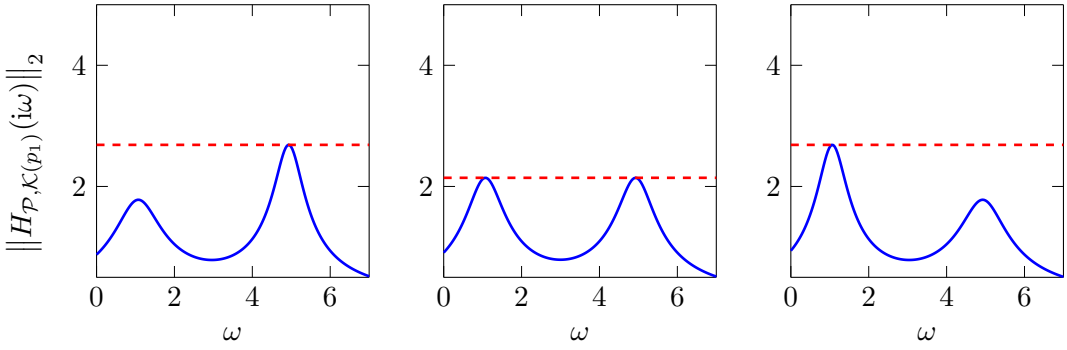
The approach is based on solving two algebraic Riccati equations of size $n_x \times n_x$. Numerical robustness issues that limited applicability of this method were addressed in [3] such that the method is well established among control engineers. However, one drawback of this method is the dimension of the computed controller. Using the algorithm suggested in [15] the computed controller has the same number of states as the system which is not acceptable for the actual implementation of the controller when the dimension of the system model becomes large.

This problem can partly be addressed by using model order reduction, which is the approximation of a system with a large number of states by a system with a smaller number of states. An introduction to the field of model order reduction is given in [2, 4] and controller order reduction is specifically addressed in [18, 49]. However, the reduced order controllers are generically not optimal with regard to the minimization of the $\mathcal{H}_\infty$-norm of the closed loop system or might not even stabilize the closed loop system.

Furthermore, in the large scale case the solution of the two algebraic Riccati equations becomes numerically expensive. Another problem with the practicability of the approach is that for some use cases studied in [20, 43] it is desired to have a certain controller structure, i. e. a structure that is imposed on the controller matrices $[A_c, B_c, C_c, D_c]$.

3.2 Fixed Order $\mathcal{H}_\infty$-Controller Design

The demands concerning structure and dimension of the controller for $\mathcal{H}_\infty$-optimization are met in [11] using a numerical optimization algorithm. In Example 3.1 it is shown



(a) $p_1 = 0.4$ (b) $p_1 = 0.4998$ (c) $p_1 = 0.6$

Figure 3.1: Transfer function of (3.2) for different values of p_1

that the $\mathcal{H}_\infty$-norm of the closed loop transfer function may in general not depend smoothly on the controller parameters everywhere.

Example 3.1. *Consider the system given by*

$$\mathcal{P}: \begin{cases} \dot{x}(t) = \begin{bmatrix} -1+\mathrm{i} & 0 \\ 0 & -1+5i \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(t) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} w(t) \\ y(t) = z(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x(t) \end{cases}$$

that is controlled by a static output feedback controller

$$\mathcal{K}: \quad u(t) = \begin{bmatrix} p_1 & 1 - p_1 y(t) \end{bmatrix}, \tag{3.1}$$

where p_1 is a tunable controller parameter. The closed loop system is given by

$$\begin{aligned} \dot{x}(t) &= \left(\begin{bmatrix} -1+\mathrm{i} & 0 \\ 0 & -1+5i \end{bmatrix} + \begin{bmatrix} p_1 & 1-p_1 \\ p_1 & 1-p_1 \end{bmatrix} \right) x(t) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} w(t), \\ z(t) &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x(t). \end{aligned} \tag{3.2}$$

The closed loop transfer function of (3.2) for different values of p_1 is shown in Figure 3.1. The maximum of $\|H_{\mathcal{P},\mathcal{K}(p_1)}(\mathrm{i}\cdot)\|_2$ is indicated with a red line. It can be seen in Figure 3.1 that the two local maxima change in size as p_1 increases. At $p_1 = 0.4998$ the left and right local maximum have the same value. The global maximum is at the left local maximum for $p_1 < 0.4998$ while the right local maximum is the global maximum for $p_1 > 0.4998$.

The dependency of the $\mathcal{H}_\infty$-norm of the transfer function on p_1 is shown in Figure 3.2. It can be observed that the $\mathcal{H}_\infty$-norm does not depend smoothly on p_1 everywhere, since the two distinct local maximizers are global maximizers for different values of p_1.

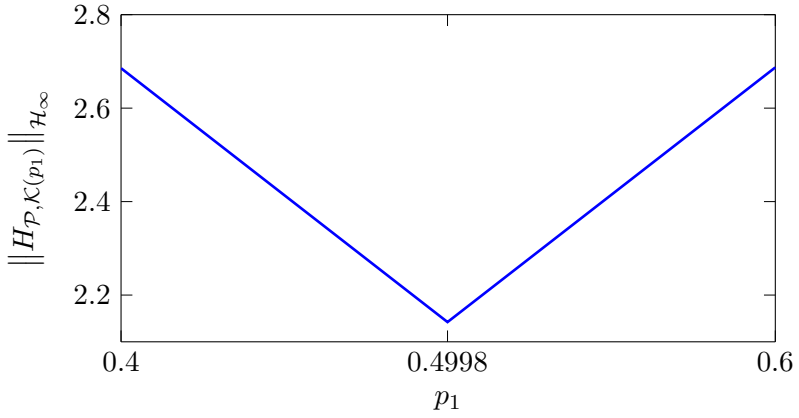


Figure 3.2: $\mathcal{H}_\infty$-norm of the closed loop transfer function of (3.2) over changing controller parameter

Algorithm 3 Fixed order $\mathcal{H}_\infty$-optimization

Input: Plant $\mathcal{P}$, initial (stabilizing) controller $\mathcal{K}$
Output: Locally $\mathcal{H}_\infty$-optimal controller $\mathcal{K}_{\text{opt}}$

1: **for** $k = 1, 2, \ldots$ **do**
2: Construct $\mathbf{CL}(\mathcal{P}, \mathcal{K})$, the closed loop model from $\mathcal{P}, \mathcal{K}$ as in (2.8).
3: Compute $f(\mathcal{K}) := \|H_{\mathcal{P},\mathcal{K}}\|_{\mathcal{H}_\infty}$.
4: Compute $\nabla f(\mathcal{K})$, the gradient of $\|H_{\mathcal{P},\hat{\mathcal{K}}}\|_{\mathcal{H}_\infty}\Big|_{\hat{\mathcal{K}}=\mathcal{K}}$.
5: Update $\mathcal{K}$ using the method in [13].
6: **end for**

Due to advances in computational speed and improved algorithms for nonconvex and nonsmooth optimization problems it is possible to simply parametrize controllers and find a minimizer for the $\mathcal{H}_\infty$-norm. In this way, given structured controller matrices of fixed order, the free parameters are iteratively changed according to a numerical optimization scheme to minimize the $\mathcal{H}_\infty$-norm of the closed loop system. The method presented in [11] is available in MATLAB as software package `hifoo`[1]. It is described in Algorithm 3. Inside the optimization loop, the $\mathcal{H}_\infty$-norm of the closed loop transfer function is iteratively computed and based on the gradient with respect to the controller parameters these parameters can be optimized using the method in [13]. The computation of the gradient that is performed in line 4 of Algorithm 3 is described in [11], however, that method to compute the gradient does not carry over for delay systems, which is why it is not described here. The computation of the gradient is further discussed in Section 4.3.2.

Note that an initial stabilizing controller must be provided in order to perform $\mathcal{H}_\infty$-optimization. This is addressed in [11] by using numerical optimization to obtain

[1] available at `https://cs.nyu.edu/overton/software/hifoo`

controller parameters that minimize the rightmost eigenvalue of the closed loop system until all eigenvalues are in the open left half plane, which is also a nonsmooth optimization problem. Then Algorithm 3 can be executed.

3.3 Recent Advances in $\mathcal{H}_\infty$-Norm Computation

It can be seen that in Algorithm 3 a couple of $\mathcal{H}_\infty$-norm evaluations are performed in order to determine the $\mathcal{H}_\infty$-optimal controller. Therefore, it is necessary to have an efficient algorithm for the computation of the $\mathcal{H}_\infty$-norm at hand.

The computation of the $\mathcal{H}_\infty$-norm is studied since the early 1990s when an algorithm to compute the distance to instability of a matrix developed in [12] was adapted for the computation of the $\mathcal{L}_\infty$-norm of a linear dynamical system in [8] and, independently, in [10]. These approaches were generalized to transfer functions of the form

$$H(s) = C(sE - A)^{-1}B + D \tag{3.3}$$

with a possibly singular matrix E in [5]. They use the connection between the singular values of $H(\mathrm{i}\omega)$ and the imaginary axis eigenvalues of a Hamiltonian matrix that can be constructed from the matrices $[E, A, B, C, D]$. These methods are based on bisection and require the solution of several Hamiltonian eigenvalue problems of size around $2n_x$. That is why they are not suitable for systems exceeding medium scale. Systems with irrational transfer functions such as systems with delay are also not directly encompassed by these algorithms.

In [35] the performance of the $\mathcal{L}_\infty$-norm computation was improved for large scale systems using model order reduction. Therein, the fact that in applications typically $n \gg m, p$ is exploited and subspace projections are used to reduce the computational cost of computing the $\mathcal{L}_\infty$-norm. After choosing the set of initial interpolation points $\mathbb{W} \subset \mathrm{i}\mathbb{R}$, orthonormal matrices $V, W \in \mathbb{R}^{n\times k}$ are computed such that

$$CV(\lambda W^\mathsf{T} EV - W^\mathsf{T} AV)^{-1} W^\mathsf{T} B = C(\lambda E - A)^{-1} B \text{ for all } \lambda \in \mathbb{W}.$$

Then the $\mathcal{L}_\infty$-norm of $\tilde{H}$, with $\tilde{H}(s) = \tilde{C}(s\tilde{E} - \tilde{A})^{-1}\tilde{B} + D$, where $\tilde{C} := CV$, $\tilde{E} := W^\mathsf{T} EV$, $\tilde{A} := W^\mathsf{T} AV$, and $\tilde{B} := W^\mathsf{T} B$ is computed at a low cost, since a moderate number of interpolation points typically leads to $k \ll n$ and $\tilde{A} \in \mathbb{R}^{k\times k}$. Then, the set of interpolation points $\mathbb{W}$ and the projection matrices are expanded such that $\mathrm{i}\omega_*$, the point on the imaginary axis where the $\mathcal{L}_\infty$-norm of $\tilde{H}$ is attained, is added to the set of interpolation points $\mathbb{W}$. Repeating this expansion of the set of interpolation points leads to an iterative algorithm with a locally superlinear rate of convergence. The results of the numerical experiments conducted in [35] indicate that despite the fact that only convergence to a local maximizer of $\sigma_{\max}(H(\mathrm{i}\cdot))$ is guaranteed, the global maximum is often found using this method.

The subspace projection method can also be extended to certain irrational functions, i. e. functions H defined as

$$H(s) := C(s)D(s)^{-1}B(s)$$

with meromorphic functions $B : \Omega \to \mathbb{C}^{n\times m}$, $C : \Omega \to \mathbb{C}^{p\times n}$, $D : \Omega \to \mathbb{C}^{n\times n}$ that satisfy

$$B(s) := f_1(s)B_1 + \ldots + f_{\kappa_B}(s)B_{\kappa_B}, \tag{3.4}$$
$$C(s) := g_1(s)C_1 + \ldots + g_{\kappa_C}(s)C_{\kappa_C}, \tag{3.5}$$
$$D(s) := h_1(s)D_1 + \ldots + h_{\kappa_D}(s)D_{\kappa_D}. \tag{3.6}$$

The subspace projection is applied to the given matrices $B_1, \ldots, B_{\kappa_B} \in \mathbb{C}^{n\times m}$, $C_1, \ldots, C_{\kappa_C} \in \mathbb{C}^{p\times n}$, $D_1, \ldots, D_{\kappa_D} \in \mathbb{C}^{n\times n}$. The meromorphic coefficient functions $f_1, \ldots, f_{\kappa_B}, g_1, \ldots, g_{\kappa_C}, h_1, \ldots, h_{\kappa_D} : \Omega \to \mathbb{C}$ remain unchanged. After the subspace projection the $\mathcal{L}_\infty$-norm of the reduced transfer function $\tilde{H}$ with

$$\tilde{H}(s) := C(s)V(W^\mathsf{T}D(s)V)^{-1}W^\mathsf{T}B(s)$$

is computed and the same expansion strategy as in [35] is applied.

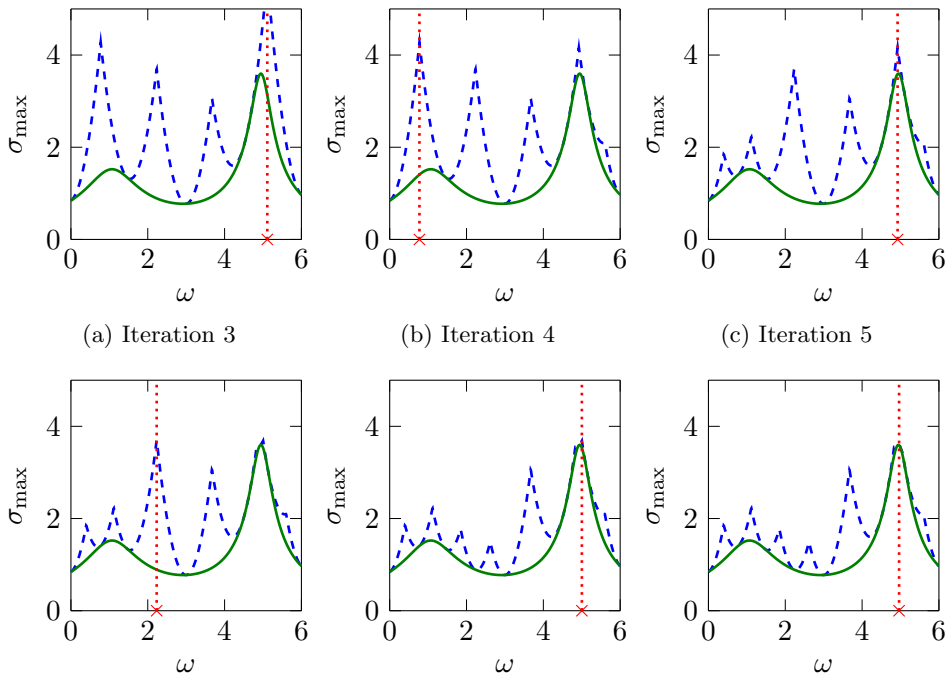
The reduced transfer function still is a irrational function. Therefore, an algorithm that is capable of computing the $\mathcal{L}_\infty$-norm of a irrational function must be used to perform the $\mathcal{L}_\infty$-norm computation of $\tilde{H}$.

In [35] the `eigopt`[2]-algorithm that is presented in [32] is used for this purpose. The algorithm uses piecewise quadratic upper bounds of the original function to find its global maximum. In Figure 3.3 the final iterations before convergence of `eigopt` are shown. In each iteration a new interpolation point is added at the maximum value of the upper bound. In this way the $\mathcal{L}_\infty$-norm of the upper bound converges to the $\mathcal{L}_\infty$-norm of the original function. A drawback of this method is that the critical interval in which the $\mathcal{L}_\infty$-norm is attained as well as a global lower bound on the second derivative of $-\sigma_{\max}(H(\mathrm{i}\cdot))$ to ensure that the piecewise quadratic functions are in fact upper bounds of the given function H need to be specified a priori but are often hard to obtain at a low computational cost.

Another method to compute the $\mathcal{L}_\infty$-norm specifically of transfer functions of delay systems is presented in [34], where a connection between the imaginary axis eigenvalues of an infinite dimensional operator $\mathcal{L}_\xi$ and the $\mathcal{L}_\infty$-norm of the transfer function is derived. Then a predictor-corrector algorithm to compute the $\mathcal{L}_\infty$-norm based on discretizations of $\mathcal{L}_\xi$ is established and made available in MATLAB programming language as software package `tds_hinfn`[3]. Since in the prediction step the imaginary axis eigenvalues of the discretization of $\mathcal{L}_\xi$ need to be computed and in the correction step a set of nonlinear equations needs to be solved, this method is computationally demanding.

Both methods can benefit from subspace projection if $n \gg p, m$ because by using projection, the problem size is reduced. This is demonstrated for the `eigopt`-algorithm in [35]. However, the drawback of this method which is the a priori specification of the critical interval and the lower bound on the second derivative of $-\sigma_{\max}(H(\mathrm{i}\cdot))$ is not overcome by subspace projection. Furthermore, even in the medium scale case the

[2] available at `https://github.com/rothos/eigopt`
[3] available at `http://twr.cs.kuleuven.be/research/software/delay-control/hinf/`



(a) Iteration 3 (b) Iteration 4 (c) Iteration 5

(d) Iteration 6 (e) Iteration 7 (f) Iteration 8

Figure 3.3: Intermediate piecewise quadratic upper bounds that are obtained by `eigopt`. The original function is depicted in green while the upper bounds are represented by blue dotted lines. The red crosses on the base line indicate the location of the maximizers for the $\mathcal{L}_\infty$-norm of the upper bounds.

algorithms that compute the $\mathcal{L}_\infty$-norm of irrational functions are far less efficient than the methods that perform the computation for rational ones.

That is why in [38] the subspace projection is replaced by rational interpolation. Specifically, a matrix quadruple $\left[\tilde{E}, \tilde{A}, \tilde{B}, \tilde{C}\right]$ is constructed such that its transfer function $\tilde{H}(s) = \tilde{C}(s\tilde{E} - \tilde{A})^{-1}\tilde{B}$ matches the given function H and its derivative at the interpolation points given in $\mathbb{W} = \{\mathrm{i}\omega_1, \ldots, \mathrm{i}\omega_r\} \subset \mathrm{i}\mathbb{R}$. In [1] this task is addressed by using so-called Loewner matrices. When the input or the dimension of the given system is greater than one, *block Loewner matrices* are used to perform the interpolation. The *Loewner quadruple* $[\mathbf{E}, \mathbf{A}, \mathbf{B}, \mathbf{C}]$ consisting of the block Loewner matrix $\mathbf{E} = [\mathbf{E}_{ij}] \in \mathbb{C}^{rp\times rm}$, the *shifted* block Loewner matrix $\mathbf{A} = [\mathbf{A}_{ij}] \in \mathbb{C}^{rp\times rm}$, and two data matrices $\mathbf{B} \in \mathbb{C}^{rp\times m}$ and $\mathbf{C} \in \mathbb{C}^{p\times rm}$, where r is the number of interpolation points, is given by

$$\begin{aligned}
\mathbf{E}_{ij} &= \begin{cases} \frac{1}{\mathrm{i}\omega_i - \mathrm{i}\omega_j}(H(\mathrm{i}\omega_i) - H(\mathrm{i}\omega_j)), & \text{if } i \neq j, \\ H'(\mathrm{i}\omega_i), & \text{else,} \end{cases} \\
\mathbf{A}_{ij} &= \begin{cases} \frac{1}{\omega_i - \omega_j}(\omega_i H(\mathrm{i}\omega_i) - \omega_j H(\mathrm{i}\omega_j)), & \text{if } i \neq j, \\ H(\mathrm{i}\omega_i) + \mathrm{i}\omega_i H'(\mathrm{i}\omega_i), & \text{else,} \end{cases}
\end{aligned} \tag{3.7}$$

and moreover,

$$\mathbf{B} = \begin{bmatrix} H(\mathrm{i}\omega_1) \\ \vdots \\ H(\mathrm{i}\omega_r) \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} H(\mathrm{i}\omega_1) & \ldots & H(\mathrm{i}\omega_r) \end{bmatrix}.$$

From the matrices $\mathbf{E}$, $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ the system matrices of the interpolating system can be computed if for all $\mu \in \mathbb{W}$ it holds that

$$\operatorname{rank}(\mu\mathbf{E} - \mathbf{A}) = \operatorname{rank}\begin{bmatrix} \mathbf{E} & \mathbf{A} \end{bmatrix} = \operatorname{rank}\begin{bmatrix} \mathbf{E} \\ \mathbf{A} \end{bmatrix}. \tag{3.8}$$

In order to obtain a regular system which is important for the investigation of the transfer function, the economic singular value decomposition of the vertical and horizontal concatenations of $\mathbf{E}$ and $\mathbf{A}$ that are given by

$$\begin{bmatrix} \mathbf{E} & \mathbf{A} \end{bmatrix} = Y\Sigma_{\mathrm{l}}\tilde{X}^{\mathsf{H}}, \quad \begin{bmatrix} \mathbf{E} \\ \mathbf{A} \end{bmatrix} = \tilde{Y}\Sigma_{\mathrm{r}}X^{\mathsf{H}}, \tag{3.9}$$

where we have $\Sigma_{\mathrm{l}} = \operatorname{diag}(\sigma_{\mathrm{l},1}, \ldots, \sigma_{\mathrm{l},k}) \in \mathbb{R}^{k\times k}$, $\Sigma_{\mathrm{r}} = \operatorname{diag}(\sigma_{\mathrm{r},1}, \ldots, \sigma_{\mathrm{r},k}) \in \mathbb{R}^{k\times k}$ with $\sigma_{\mathrm{l},1} \geq \ldots \geq \sigma_{\mathrm{l},k} > 0$, $\sigma_{\mathrm{r},1} \geq \ldots \geq \sigma_{\mathrm{r},k} > 0$, and $\tilde{Y}$, Y, $\tilde{X}$, $X \in \mathbb{C}^{n\times k}$ have orthonormal columns, are computed. The system matrices of the regularized system are then given by

$$\tilde{E} = -Y^{\mathsf{H}}\mathbf{E}X, \quad \tilde{A} = -Y^{\mathsf{H}}\mathbf{A}X, \quad \tilde{B} = Y^{\mathsf{H}}\mathbf{B}, \quad \tilde{C} = \mathbf{C}X. \tag{3.10}$$

Note that the condition specified in (3.8) is only fulfilled and the regularization carried out in (3.9), (3.10) can only be applied, if the pencil $s\mathbf{E} - \mathbf{A}$ is either regular or

Algorithm 4 Computation of the $\mathcal{L}_\infty$-norm using rational interpolation

Input: A function $H \in \mathcal{L}_\infty^{p\times m}$, initial interpolation points $\mathbb{W} \subset \mathrm{i}\mathbb{R}$.
Output: The $\mathcal{L}_\infty$-norm of H, the maximizer ω_* with $f_* := \|H\|_{\mathcal{L}_\infty} = \|H(\mathrm{i}\omega_*)\|_2$.

1: **while** not converged **do**
2: Construct $\left[\tilde{E}, \tilde{A}, \tilde{B}, \tilde{C}\right]$ as in (3.7), (3.9), and (3.10).
3: Define $\tilde{H}(s) := \tilde{C}(s\tilde{E} - \tilde{A})^{-1}\tilde{B}$.
4: Compute $\omega_* := \arg \max_{\omega\in\mathbb{R}\cup\{\infty\}} \sigma_{\max}(\tilde{H}(\mathrm{i}\omega))$.
5: Set $\mathbb{W} := \mathbb{W} \cup \{\mathrm{i}\omega_*\}$.
6: **end while**
7: Set $f_* := \|H(\mathrm{i}\omega_*)\|_2$.

its singular Kronecker blocks are of size at most 0×1 or 1×0, see [19, Chapter 7.3]. Regularization for higher-index singular Kronecker blocks is addressed in [6]. However, the method developed in [6] makes use of feedback regularization which is why it cannot be used here.

Using the same expansion strategy as in [35] a local maximizer of $\sigma_{\max}(H(\mathrm{i}\cdot))$ is found. This method is summarized in Algorithm 4 and implemented in Matlab as software package `linorm_subsp`[4].

In [38] it is shown that this interpolation approach can lead to a significant increase of computational efficiency for irrational transfer functions. This is because the computation of the $\mathcal{L}_\infty$-norm of the interpolated systems is more efficient, since then the established algorithms for the computation of the $\mathcal{L}_\infty$-norm of rational functions can be used. Furthermore, no structural constraints as given in (3.4) have to be imposed on the function of interest when rational interpolation is applied.

This sharp increase in efficiency, which makes the repeated computation of the $\mathcal{H}_\infty$-norm of delay systems within an optimization loop feasible, is the main impetus for this thesis.

[4] available at `http://www.math.tu-berlin.de/index.php?id=186267` (version 1.1), with the projection approach from [35] included

4 $\mathcal{H}_\infty$-Controller Design for Delay Systems

The stabilization of delay systems is extensively studied and various methods using *finite spectrum assignment* (cf. [9, 31]) or modifications of the *Smith predictor* (cf. [47]) to stabilize systems with delay have been developed. Also the $\mathcal{H}_\infty$-controller synthesis for delay systems is discussed in [47] using the Smith predictor to reformulate the delay controller design problem such that the methods already described in [15] can be applied. However, these methods do not allow for fixed order controllers or structural constraints imposed on the controller. In order to obtain an initial stabilizing controller we will therefore only consider methods for the stabilization of delay systems that allow for fixed order controllers as starting point for our development such as the method described in [33] which uses numerical optimization to tune controller parameters in order to stabilize a given system.

4.1 Stabilization of Delay Systems

In [33] a method for the stabilization of delay systems called *continuous pole placement* is introduced. It is formulated as state feedback control (which can be interpreted as a special case of output feedback control with $y(t) = x(t)$ and $u(t) = Kx(t)$ implying that the state can be used directly to determine the control signal). However, it can easily be extended to output feedback control. The method relies on the computation of the rightmost characteristic roots of the closed loop system which is addressed in [17] and implemented in [37] as software package `dde-biftool`[1]. We briefly illustrate the method with a basic stabilization problem for a delay system.

Consider the delay equation

$$\dot{x}(t) = Ax(t) + Bu(t - \tau), \tag{4.1}$$

where $x(t) \in \mathbb{R}^{n_x}$, $u(t) \in \mathbb{R}^{n_u}$ and $A \in \mathbb{R}^{n_x \times n_x}, B \in \mathbb{R}^{n_x \times n_u}$. Find a matrix $K \in \mathbb{R}^{n_u \times n_x}$ such that for $u(t) = Kx(t)$ the closed loop system

$$\dot{x}(t) = Ax(t) + BKx(t - \tau)$$

is stabilized, i. e. the characteristic equation

$$\det(\lambda I - A - BK\mathrm{e}^{-\lambda}) = 0 \tag{4.2}$$

only has solutions λ with a negative real part.

[1] available at `http://sourceforge.net/projects/ddebiftool`

In [33] this problem is addressed by using numerical optimization to minimize the rightmost characteristic root of (4.2) until its real part is negative by changing the entries of the matrix K. Applications of this method for state feedback control are given in [41]. As mentioned above, this approach is quite naturally extendable to output feedback control. Simply expand (4.1) by introducing an output equation

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

where $C \in \mathbb{R}^{n_y \times n_x}$ and minimize the rightmost solution of

$$\det\left(\lambda I - A - BKC\mathrm{e}^{-\lambda}\right) = 0,$$

which is the characteristic equation of the closed loop system for $u(t) = KCx(t)$, with $K \in \mathbb{R}^{n_u \times n_y}$. This method also enables the user to impose a structure on K by fixing some of its entries and only passing the remaining entries to the numerical optimizer. However, in the current formulation with only one controller matrix K it is limited to static controllers, that is $n_c = 0$. Therefore, the closed loop models that result from higher order controllers are determined in the following section.

4.2 Closed Loop Model of the Delay System

In the following, we consider control systems with delay in the measured output which is equivalent to introducing an input delay to the controller equations. This can be reflected within the controller equations as shown in (4.3). We derive the closed loop system that results from introducing output delay to the plant in this section since we introduce delay to our numerical examples in this way, as well. The derivation of the closed loop system for a more general class of delay systems is described in section 6.2.

Consider the controller equations with the delayed measured output from the system $y(t-\tau)$ as input

$$\begin{aligned} \dot{x}_c(t) &= A_c x_c(t) + B_c y(t-\tau), \quad x_c(0) = 0, \\ u(t) &= C_c x_c(t) + D_c y(t-\tau), \end{aligned} \tag{4.3}$$

where τ is the controller input delay, or plant output delay, respectively. Then we need to derive a new form for the closed loop system, since the form given in (2.8) cannot be applied here due to the delay that is introduced.

The closed loop system can be expressed with a delay differential-algebraic equation (DDAE). Consider the set of equations

$$\begin{aligned} \dot{x}(t) &= Ax(t) + B_1 w(t) + Bu(t), \quad x(0) = x_0, \\ \dot{x}_c(t) &= A_c x_c(t) + B_c y(t-\tau), \quad x_c(0) = 0, \\ z(t) &= C_1 x(t) + D_{11} w(t) + D_{12} u(t), \\ y(t) &= Cx(t) + D_{21} w(t) + D_{22} u(t), \\ u(t) &= C_c x_c(t) + D_c y(t-\tau), \end{aligned} \tag{4.4}$$

which describes the dynamics of a closed loop system with a delayed controller feedback. This can be rewritten into the form

$$\begin{aligned}\underbrace{\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} I & 0 & 0 & 0\\ 0 & I & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix}}_{\mathcal{E}}\begin{bmatrix} x(t)\\ x_c(t)\\ u(t)\\ y(t)\end{bmatrix} &= \underbrace{\begin{bmatrix} A & 0 & B & 0\\ 0 & A_c & 0 & 0\\ 0 & C_c & -I & 0\\ C & 0 & D_{22} & -I\end{bmatrix}}_{\mathcal{A}_0}\begin{bmatrix} x(t)\\ x_c(t)\\ u(t)\\ y(t)\end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0\\ 0 & 0 & 0 & B_c\\ 0 & 0 & 0 & D_c\\ 0 & 0 & 0 & 0\end{bmatrix}}_{\mathcal{A}_1}\begin{bmatrix} x(t-\tau)\\ x_c(t-\tau)\\ u(t-\tau)\\ y(t-\tau)\end{bmatrix}\\ &\quad + \underbrace{\begin{bmatrix} B_1\\ 0\\ 0\\ D_{21}\end{bmatrix}}_{\mathcal{B}} w(t),\\ z(t) &= \underbrace{\begin{bmatrix} C_1 & 0 & D_{12} & 0\end{bmatrix}}_{\mathcal{C}}\begin{bmatrix} x(t)\\ x_c(t)\\ u(t)\\ y(t)\end{bmatrix} + \underbrace{D_{11}}_{\mathcal{D}} w(t).\end{aligned} \tag{4.5}$$

We use this form to compute the $\mathcal{L}_\infty$-norm of the transfer function of the closed loop system. Assume that $x(t) = x_0, x_c(t) = 0, u(t) = 0, y(t) = 0$ for all $t < 0$. We denote the system given in (4.5) as $\mathbf{CL}^\tau(\mathcal{P}, \mathcal{K})$. The transfer function of $\mathbf{CL}^\tau(\mathcal{P}, \mathcal{K})$ is given by

$$H^\tau_{\mathcal{P},\mathcal{K}}(s) := \mathcal{C}\big(s\mathcal{E} - \mathcal{A}_0 - \mathrm{e}^{-\tau s}\mathcal{A}_1\big)^{-1}\mathcal{B} + \mathcal{D}.$$

The computation of the $\mathcal{L}_\infty$-norm of $H^\tau_{\mathcal{P},\mathcal{K}}$ can be performed using the method described in [38] since it is fit to compute the $\mathcal{L}_\infty$-norm of DDAEs. However, in [16] it is shown that for DDAEs it does not hold in general that if all characteristic solutions are in the open left half plane the homogeneous part of the system is asymptotically stable. Therefore, it is in general not sufficient to compute the solutions λ of

$$\det\left(\lambda\mathcal{E} - \mathcal{A}_0 - \mathcal{A}_1\mathrm{e}^{-\lambda\tau}\right) = 0$$

and check if their real part is negative. Furthermore, it is necessary that the system is *strangeness free* following the concept of the *strangeness index* in [26]. By [16, Definition 2.5] the DDAE we are concerned with is strangeness free if there exists a nonsingular matrix $W \in \mathbb{C}^{\tilde{n}\times\tilde{n}}$, where $\tilde{n} = n_x + n_c + n_u + n_y$ is the state dimension of (4.5), such that

$$W^{-1}\mathcal{E} = \begin{bmatrix}\mathcal{E}_1\\ 0\\ 0\end{bmatrix}, \quad W^{-1}\mathcal{A}_0 = \begin{bmatrix}\mathcal{A}_{01}\\ \mathcal{A}_{02}\\ 0\end{bmatrix}, \quad W^{-1}\mathcal{A}_1 = \begin{bmatrix}\mathcal{A}_{11}\\ \mathcal{A}_{12}\\ \mathcal{A}_{13}\end{bmatrix},$$

where $\mathcal{E}_1, \mathcal{A}_{01}, \mathcal{A}_{11} \in \mathbb{C}^{d\times n}$, $\mathcal{A}_{02}, \mathcal{A}_{12} \in \mathbb{C}^{a\times n}$, $\mathcal{A}_{13} \in \mathbb{C}^{h\times n}$ with $d+a+h = \tilde{n}$, rank $\mathcal{E}_1 =$ rank $\mathcal{E} = d$, rank $\mathcal{A}_{02} = a$, and

$$\operatorname{rank} \begin{bmatrix} \mathcal{E}_1 \\ \mathcal{A}_{02} \\ \mathcal{A}_{13} \end{bmatrix} = \tilde{n}.$$

In order to see that (4.5) is in fact strangeness free, note that

$$\begin{aligned} \operatorname{rank} \begin{bmatrix} \mathcal{E}_1 \\ \mathcal{A}_{02} \end{bmatrix} &= \tilde{n}, \text{ with} \\ \mathcal{E}_1 &= \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \end{bmatrix}, \\ \mathcal{A}_{02} &= \begin{bmatrix} 0 & C_c & -I & 0 \\ C & 0 & D_{22} & -I \end{bmatrix}, \end{aligned}$$

where $d = n_x + n_c$, $a = n_u + n_y$, and $h = 0$. Therefore by [16, Theorem 3.1] the trivial solution of the homogeneous is asymptotically stable if and only if for all solutions λ of

$$\det\left(\lambda\mathcal{E} - \mathcal{A}_0 - \mathrm{e}^{-\lambda\tau}\mathcal{A}_1\right) = 0$$

it holds that $\lambda \in \mathbb{C}^-$.

However, using the software `dde-biftool` it is only possible to perform stability analysis for DDEs. That is why the algebraic part of (4.5) needs to be eliminated for the computation of the eigenvalues. In the following theorem, this task is addressed.

Theorem 4.1. *Let ν be the maximum of the nilpotency indices of D_cD_{22} and $D_{22}D_c$. Then all solutions $\lambda \in \mathbb{C}$ of $\det\left(\lambda\mathcal{E} - \mathcal{A}_0 - \mathrm{e}^{-\lambda\tau}\mathcal{A}_1\right) = 0$, with $\mathcal{E}, \mathcal{A}_0$, and $\mathcal{A}_1$ as in (4.5), are also solutions of*

$$\det\left(\lambda I - A_0 - \sum_{k=1}^{\nu} \mathrm{e}^{-k\lambda\tau} A_k\right) = 0,$$

with

$$A_k = \begin{cases} \begin{bmatrix} A & BC_c \\ 0 & A_c \end{bmatrix} & \text{for } k = 0, \\ \begin{bmatrix} BD_c(D_{22}D_c)^{k-1}C & BD_cD_{22}(D_cD_{22})^{k-1}C_c \\ B_c(D_{22}D_c)^{k-1}C & B_cD_{22}(D_cD_{22})^{k-1}C_c \end{bmatrix} & \text{for } k > 0 \end{cases}$$

and vice versa.

Proof. First, note that

$$\begin{bmatrix} I & 0 \\ -D_{22} & I \end{bmatrix}^{-1} = \begin{bmatrix} I & 0 \\ D_{22} & I \end{bmatrix}.$$

Assume that $\lambda \in \mathbb{C}$ and $v = \left[v_1^\mathsf{T}, v_2^\mathsf{T}, v_3^\mathsf{T}, v_4^\mathsf{T}\right]^\mathsf{T} \in \mathbb{C}^{\tilde{n}}$ satisfy

$$\lambda \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} - \begin{bmatrix} A & 0 & B & 0 \\ 0 & A_c & 0 & 0 \\ 0 & C_c & -I & 0 \\ C & 0 & D_{22} & -I \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} - \mathrm{e}^{-\lambda\tau} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & B_c \\ 0 & 0 & 0 & D_c \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{4.6}$$

Then we can rewrite the two lower block rows of (4.6) i. e.

$$\begin{bmatrix} 0 & C_c \\ C & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + \begin{bmatrix} -I & 0 \\ D_{22} & -I \end{bmatrix} \begin{bmatrix} v_3 \\ v_4 \end{bmatrix} + \begin{bmatrix} 0 & D_c \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathrm{e}^{-\lambda\tau} v_3 \\ \mathrm{e}^{-\lambda\tau} v_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

as

$$\begin{bmatrix} v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} 0 & C_c \\ C & D_{22}C_c \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + \begin{bmatrix} 0 & D_c \\ 0 & D_{22}D_c \end{bmatrix} \begin{bmatrix} \mathrm{e}^{-\lambda\tau} v_3 \\ \mathrm{e}^{-\lambda\tau} v_4 \end{bmatrix}. \tag{4.7}$$

In this way, v_3 and v_4 can be substituted in the first two rows of equation (4.6) such that we have

$$\begin{aligned} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} &= \begin{bmatrix} A & 0 \\ 0 & A_c \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + \begin{bmatrix} B & 0 \\ 0 & 0 \end{bmatrix} \left(\begin{bmatrix} 0 & C_c \\ C & D_{22}C_c \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + \begin{bmatrix} 0 & D_c \\ 0 & D_{22}D_c \end{bmatrix} \begin{bmatrix} \mathrm{e}^{-\lambda\tau} v_3 \\ \mathrm{e}^{-\lambda\tau} v_4 \end{bmatrix} \right) \\ &\qquad + \begin{bmatrix} 0 & 0 \\ 0 & B_c \end{bmatrix} \begin{bmatrix} \mathrm{e}^{-\lambda\tau} v_3 \\ \mathrm{e}^{-\lambda\tau} v_4 \end{bmatrix} \\ &= \begin{bmatrix} A & BC_c \\ 0 & A_c \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + \begin{bmatrix} 0 & BD_c \\ 0 & B_c \end{bmatrix} \begin{bmatrix} \mathrm{e}^{-\lambda\tau} v_3 \\ \mathrm{e}^{-\lambda\tau} v_4 \end{bmatrix}. \end{aligned} \tag{4.8}$$

By shifting (4.7) we get

$$\begin{bmatrix} \mathrm{e}^{-\lambda\tau} v_3 \\ \mathrm{e}^{-\lambda\tau} v_4 \end{bmatrix} = \begin{bmatrix} 0 & C_c \\ C & D_{22}C_c \end{bmatrix} \begin{bmatrix} \mathrm{e}^{-\lambda\tau} v_1 \\ \mathrm{e}^{-\lambda\tau} v_2 \end{bmatrix} + \begin{bmatrix} 0 & D_c \\ 0 & D_{22}D_c \end{bmatrix} \begin{bmatrix} \mathrm{e}^{-2\lambda\tau} v_3 \\ \mathrm{e}^{-2\lambda\tau} v_4 \end{bmatrix}.$$

Thus, we have that

$$\begin{bmatrix} 0 & BD_c \\ 0 & B_c \end{bmatrix} \begin{bmatrix} \mathrm{e}^{-\lambda\tau} v_3 \\ \mathrm{e}^{-\lambda\tau} v_4 \end{bmatrix} = \begin{bmatrix} BD_cC & BD_cD_{22}C_c \\ B_cC & B_cD_{22}C_c \end{bmatrix} \begin{bmatrix} \mathrm{e}^{-\lambda\tau} v_1 \\ \mathrm{e}^{-\lambda\tau} v_2 \end{bmatrix} + \begin{bmatrix} 0 & BD_cD_{22}D_c \\ 0 & B_cD_{22}D_c \end{bmatrix} \begin{bmatrix} \mathrm{e}^{-2\lambda\tau} v_3 \\ \mathrm{e}^{-2\lambda\tau} v_4 \end{bmatrix}.$$

This is inserted into (4.8) and gives

$$\begin{aligned} \begin{bmatrix} \lambda v_1 \\ \lambda v_2 \end{bmatrix} &= \begin{bmatrix} A & BC_c \\ 0 & A_c \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + \begin{bmatrix} BD_cC & BD_cD_{22}C_c \\ B_cC & B_cD_{22}C_c \end{bmatrix} \begin{bmatrix} \mathrm{e}^{-\lambda\tau} v_1 \\ \mathrm{e}^{-\lambda\tau} v_2 \end{bmatrix} \\ &\qquad + \begin{bmatrix} 0 & BD_cD_{22}D_c \\ 0 & B_cD_{22}D_c \end{bmatrix} \begin{bmatrix} \mathrm{e}^{-2\lambda\tau} v_3 \\ \mathrm{e}^{-2\lambda\tau} v_4 \end{bmatrix}. \end{aligned}$$

Repeating the process of shifting and inserting we get that

$$\begin{bmatrix} \lambda v_1 \\ \lambda v_2 \end{bmatrix} = \begin{bmatrix} A & BC_c \\ 0 & A_c \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + \sum_{k=1}^{\nu} \begin{bmatrix} BD_c(D_{22}D_c)^{k-1}C & BD_cD_{22}(D_cD_{22})^{k-1}C_c \\ B_c(D_{22}D_c)^{k-1}C & B_cD_{22}(D_cD_{22})^{k-1}C_c \end{bmatrix} \begin{bmatrix} \mathrm{e}^{-k\lambda\tau} v_1 \\ \mathrm{e}^{-k\lambda\tau} v_2 \end{bmatrix}.$$

It remains to prove that $[v_1^\mathsf{T}, v_2^\mathsf{T}]^\mathsf{T} \neq 0$ to show that

$$\det\left(\lambda I - \underbrace{\begin{bmatrix} A & BC_c \\ 0 & A_c \end{bmatrix}}_{=:A_0} - \sum_{k=1}^{\nu} \mathrm{e}^{-k\lambda\tau} \underbrace{\begin{bmatrix} BD_c(D_{22}D_c)^{k-1}C & BD_cD_{22}(D_cD_{22})^{k-1}C_c \\ B_c(D_{22}D_c)^{k-1}C & B_cD_{22}(D_cD_{22})^{k-1}C_c \end{bmatrix}}_{=:A_k \text{ for } k>0}\right) = 0.$$

Assume $[v_1^\mathsf{T}, v_2^\mathsf{T}]^\mathsf{T} = 0$. Then (4.7) reads

$$\begin{bmatrix} v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} 0 & D_c \\ 0 & D_{22}D_c \end{bmatrix} \begin{bmatrix} \mathrm{e}^{-\lambda\tau}v_3 \\ \mathrm{e}^{-\lambda\tau}v_4 \end{bmatrix}.$$

This is equivalent to

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} I & -\mathrm{e}^{-\lambda\tau}D_c \\ 0 & I - \mathrm{e}^{-\lambda\tau}D_{22}D_c \end{bmatrix} \begin{bmatrix} v_3 \\ v_4 \end{bmatrix},$$

which can only have a nonzero solution, if

$$\det(I - \mathrm{e}^{-\lambda\tau}D_{22}D_c) = 0. \tag{4.9}$$

Let $\gamma := \mathrm{e}^{\lambda\tau}$ and note that (4.9) is equivalent to

$$\det(\gamma I - D_{22}D_c) = 0. \tag{4.10}$$

Since we have that $D_{22}D_c$ is nilpotent, $D_{22}D_c$ only has zero eigenvalues and thus (4.10) is only satisfied for $\gamma = 0$. However, $\mathrm{e}^{\lambda\tau} \neq 0$ for all $\lambda \in \mathbb{C}$ and $\tau \in \mathbb{R}^+$. This implies that if $[v_1^\mathsf{T}, v_2^\mathsf{T}]^\mathsf{T} = 0$, then also $[v_3^\mathsf{T}, v_4^\mathsf{T}]^\mathsf{T} = 0$. This is a contradiction to $v \neq 0$ and therefore $[v_1^\mathsf{T}, v_2^\mathsf{T}]^\mathsf{T} \neq 0$.

For the converse, let $(\lambda, [w_1^\mathsf{T}, w_2^\mathsf{T}]^\mathsf{T})$ be an eigenpair of $\lambda I - A_0 - \sum_{k=1}^{\nu} \mathrm{e}^{-k\lambda\tau}A_k$ and verify that then

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} := \left(\begin{bmatrix} I & 0 \\ 0 & I \\ 0 & C_c \\ 0 & 0 \end{bmatrix} + \sum_{k=1}^{\nu} \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ D_c(D_{22}D_c)^{k-1}C\mathrm{e}^{-\lambda k\tau} & D_cD_{22}(D_cD_{22})^{k-1}C_c\mathrm{e}^{-\lambda k\tau} \\ (D_{22}D_c)^{k-1}C\mathrm{e}^{-\lambda(k-1)\tau} & D_{22}(D_cD_{22})^{k-1}C_c\mathrm{e}^{-\lambda(k-1)\tau} \end{bmatrix} \right) \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

is an eigenvector of (4.6) corresponding to λ. □

We have thus shown that for systems with a structure given in (2.5) that are controlled by a controller with a delayed input the stability can be determined by solving a delay eigenvalue problem. If $D_{22} = 0$, $D_c = 0$, or if both $D_{22}D_c$ and D_cD_{22} are nilpotent with maximum index of nilpotency ν as assumed in Theorem 4.1, then we

only have finitely many summands in the delay eigenvalue problem. Therefore, in this case we can use the method described in [37] to compute the rightmost eigenvalues of the closed loop system to determine if it is asymptotically stable. Note that in the case that $D_{22} \neq 0$, the controller matrix D_c can be chosen to be 0, or such that $D_{22}D_c$ and D_cD_{22} is nilpotent. In this way, the method described in [37] may be used for arbitrary D_{22}.

Finding a stabilizing controller of fixed order is the first step of fixed order $\mathcal{H}_\infty$-controller design, since the minimization of the $\mathcal{H}_\infty$-norm using numerical optimization is only possible when an initial stabilizing controller is available. Otherwise, the closed loop transfer function would not be in $\mathcal{H}_\infty^{n_z \times n_w}$ and a gradient of the $\mathcal{H}_\infty$-norm with respect to the controller parameters could not be computed.

4.3 Fixed Order $\mathcal{H}_\infty$-Controller Design for Delay Systems

In this section we give details on our method for fixed order $\mathcal{H}_\infty$-controller design aimed at systems of the form given in equation (2.5) that have a controller with input delay as described in (4.3). We specifically address this setting because it allows providing initial stabilizing controllers using stabilization methods for the system without delay as shown in the following section.

4.3.1 Finding an Initial Controller

The immediate way to determine an initial controller that stabilizes the delay system is to use continuous pole placement from [33]. Since this method is computationally expensive we want to give another approach to determine initial controllers for delay systems that can be expressed in the form of (4.5).

From Theorem 2.5 we get that an asymptotically stable system $\mathbf{CL}^0(\mathcal{P}, \mathcal{K})$ is not destabilized by a sufficiently small delay, i. e. if $\mathbf{CL}^0(\mathcal{P}, \mathcal{K})$ is asymptotically stable, then there exists a $\tau > 0$ such that $\mathbf{CL}^\tau(\mathcal{P}, \mathcal{K})$ is asymptotically stable as well. We denote $\tau_{\max} \in \mathbb{R}^+ \cup \{\infty\}$ as maximal delay such that the controller that is designed to stabilize the closed loop system without delay $\mathbf{CL}^0(\mathcal{P}, \mathcal{K})$ still stabilizes $\mathbf{CL}^{\tau_{\max}}(\mathcal{P}, \mathcal{K})$. In the case that a system is to be stabilized with a controller delay of a given τ^*, we must distinguish two cases for τ^*:

1. The given τ^* is small enough such that $\mathbf{CL}^{\tau^*}(\mathcal{P}, \mathcal{K})$ is still asymptotically stable. Then, the initial controller for the $\mathcal{H}_\infty$-optimization of the delay system is already given by the controller for the system without delay.

2. The given τ^* is such that $\tau^* \geq \tau_{\max}$. Then $\mathbf{CL}^{\tau^*}(\mathcal{P}, \mathcal{K})$ is not asymptotically stable and another initial controller needs to be computed before minimizing the $\mathcal{H}_\infty$-norm of the closed loop system. This can be done using continuous pole placement as in [33] or an iterative optimization of the $\mathcal{H}_\infty$-norm that is discussed in Section 6.1.

Algorithm 5 A lower bound for $\tau_{\max}$

Input: Step size $\Delta\tau$, τ^*, plant $\mathcal{P}$, initial controller $\mathcal{K}$
Output: Delay $\tau = \min\{\Delta\tau \cdot \left\lfloor \frac{\tau_{\max}}{\Delta\tau} \right\rfloor, \tau^*\}$

1: Initialize $\tau := 0$, $\Delta f := 1$.
2: Construct $\mathbf{CL}^\tau(\mathcal{P}, \mathcal{K})$ as in (4.5).
3: Compute $f := \left\| H^\tau_{\mathcal{P},\mathcal{K}} \right\|_{\mathcal{L}_\infty}$.
4: **while** $(\Delta f > 0)$ and $(\tau < \tau^*)$ **do**
5: Set $f_{\text{prev}} := f$.
6: Update $\tau := \tau + \Delta\tau$.
7: Construct $\mathbf{CL}^\tau(\mathcal{P}, \mathcal{K})$ as in (4.5).
8: Compute $f := \|\mathcal{H}^\tau_{\mathcal{P},\mathcal{K}}\|_{\mathcal{L}_\infty}$.
9: Update $\Delta f := f - f_{\text{prev}}$.
10: **end while**
11: Set $\tau := \tau - \Delta\tau$.

It is important to find a numerical method for the distinction of these two cases. A straightforward way to perform the distinction would be to compute the DDE associated with $\mathbf{CL}^\tau(\mathcal{P}, \mathcal{K})$ for the given τ^* as described in Theorem 4.1 to compute the eigenvalues of $\mathbf{CL}^\tau(\mathcal{P}, \mathcal{K})$ using the method provided in [37], but for large scale problems this is prohibitively expensive, because in order to compute the rightmost eigenvalues an initial value problem of the size of the given DDE needs to be solved. Also, the method is not applicable, when $D_{22}D_c$ is not nilpotent. Furthermore, in order to compute $\tau_{\max}$ using this method, bisection or another iterative method must be used which implies multiple computations of the rightmost eigenvalues.

Therefore, we provide a method that only needs iterative computations of the $\mathcal{L}_\infty$-norm to perform the distinction or even to compute $\tau_{\max}$. This method is based on the heuristic that the $\mathcal{L}_\infty$-norm of the transfer function of an asymptotically stable system increases, when the rightmost eigenvalues move toward the imaginary axis, which can be seen in Figure 4.1. It decreases again, when the rightmost eigenvalues have passed the imaginary axis and thus the system is destabilized. In Algorithm 5 a lower bound of $\tau_{\max}$ is computed using these considerations.

In Figure 4.1 the position of eigenvalues of $\mathbf{CL}^\tau(\mathcal{P}, \mathcal{K})$ and the corresponding transfer function $H^\tau_{\mathcal{P},\mathcal{K}}$ are shown for different values of τ. It can be seen, that as τ increases, the eigenvalues move to the right and peaks develop in the transfer function. The most noticeable peaks occur at values of ω that are imaginary parts of the rightmost couple of eigenvalues. In Algorithm 5 the delay is increased within the while loop starting in line 4 as long as the $\mathcal{L}_\infty$-norm of the closed loop system is increasing. Once the eigenvalues have crossed the imaginary axis, the $\mathcal{L}_\infty$-norm of $H^\tau_{\mathcal{P},\mathcal{K}}$ decreases again which is used in Algorithm 5 as termination condition to determine $\tau_{\max}$. This is a heuristic approach and the step size needs to be chosen sufficiently small in order not to miss eigenvalues crossing the imaginary axis. However, it is computationally efficient even for large scale systems because the evaluation of the $\mathcal{L}_\infty$-norm can be performed using the method described in [38] and the maximizer from previous iterations can be

passed to the initial interpolation points for the next computation which can increase the computational speed since for small changes of τ the maximizer of the $\|H(\mathrm{i}\cdot)\|_2$ often does not change drastically.

4.3.2 Computing the Gradients

In the software package `hifoo` the gradient of the closed loop transfer function is computed exploiting the structure of the given transfer function, that is, as in (2.9), given by

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

Specifically, once the $\mathcal{H}_\infty$-norm of H is computed and the critical point on the imaginary axis $\hat{\omega}$ where the $\mathcal{H}_\infty$-norm is attained is approximately known, the derivative of H at $\hat{\omega}$ with respect to the different closed loop matrices $[A, B, C, D]$ is computed analytically.

Since the transfer functions we are concerned with are not of the form of (2.9) the method used in `hifoo` cannot be applied directly. In [22, 23] transfer functions are covered that also incorporate delay terms and an extension of the gradient computation proposed in `hifoo` is given. However, since our method for the computation of the $\mathcal{H}_\infty$-norm is realization independent and we make use of that property in Chapter 6, we do not want to introduce a realization dependence during the computation of the derivative. That is why we use a numerical approximation for the computation of the gradient. There are two ways to compute the gradient numerically. Both of them are presented in Algorithm 6.

They incorporate a finite difference scheme in which all variable controller parameters are varied one after the other and the differences in the $\mathcal{L}_\infty$-norm are used to compute the derivative with a central difference scheme as seen in line 20 of Algorithm 6. In lines 8 and 15, respectively, the fact that the critical point, where the $\mathcal{L}_\infty$-norm is attained, only changes insignificantly, when the system parameters are changed by the step size h. Despite the fact that this only holds, if the function $\left\|H^\tau_{\mathcal{P},\mathcal{K}}(\mathrm{i}\cdot)\right\|_2$ is differentiable at ω^*, in Table 5.1 it is demonstrated, that this heuristic leads to a considerable increase in computational speed and no systematic decrease in the quality of the controller design; on the contrary the $\mathcal{H}_\infty$-norm of the resulting systems is even often significantly lower.

4.3.3 Minimization of the $\mathcal{H}_\infty$-Norm

The minimization of the $\mathcal{H}_\infty$-norm for delay systems is addressed in [22, 23]. The numerical optimization scheme used in the implementation is actually the same that is also used for the controller synthesis in [24], namely the software package `hanso`. We use an enhanced version of this general purpose optimization software package called `granso` which is explicitly addressed at optimization problems that arise in controller synthesis i. e. nonsmooth optimization problems and is described in [13].

The objective function for the optimization is the function that returns the $\mathcal{H}_\infty$-norm of the given system and the optimization parameters are the variable controller

Algorithm 6 Numerical gradient computation with respect to controller parameters

Input: Plant $\mathcal{P}$, controller $\mathcal{K}$, step size h

Output: Gradient $g \in \mathbb{R}^m$ of $\left\|H^\tau_{\mathcal{P},\mathcal{K}}\right\|_{\mathcal{H}_\infty}$ with respect to the m free parameters in the controller $\mathcal{K}$

1: Construct $\mathbf{CL}^\tau(\mathcal{P},\mathcal{K})$ as in (4.5).
2: Compute $f := \left\|H^\tau_{\mathcal{P},\mathcal{K}}\right\|_{\mathcal{H}_\infty}$.
3: Define ω_*, as critical frequency, where $\left\|H^\tau_{\mathcal{P},\mathcal{K}}(\mathrm{i}\omega_*)\right\|_2 = \left\|H^\tau_{\mathcal{P},\mathcal{K}}\right\|_{\mathcal{H}_\infty}$.
4: **for** $j = 1, \ldots, m$ **do**
5: Increase parameter j of the controller $\mathcal{K}$ by h.
6: Construct $\mathbf{CL}^\tau(\mathcal{P},\mathcal{K})$ as in (4.5).
7: **if** Fast Computation **then**
8: Compute $f^+ := \left\|H^\tau_{\mathcal{P},\mathcal{K}}(\mathrm{i}\omega_*)\right\|_2$.
9: **else**
10: Compute $f^+ := \left\|H^\tau_{\mathcal{P},\mathcal{K}}\right\|_{\mathcal{H}_\infty}$.
11: **end if**
12: Decrease parameter j of the controller $\mathcal{K}$ by $2h$.
13: Construct $\mathbf{CL}^\tau(\mathcal{P},\mathcal{K})$ as in (4.5).
14: **if** Fast Computation **then**
15: Compute $f^- := \left\|H^\tau_{\mathcal{P},\mathcal{K}}(\mathrm{i}\omega_*)\right\|_2$.
16: **else**
17: Compute $f^- := \left\|H^\tau_{\mathcal{P},\mathcal{K}}\right\|_{\mathcal{H}_\infty}$.
18: **end if**
19: Increase parameter j of the controller $\mathcal{K}$ by h.
20: Set $g_j := \frac{f^+ - f^-}{2h}$.
21: **end for**

parameters, that are entries of the controller matrices. However, the objective function is slightly modified to ensure that the resulting controller always stabilizes the system. Note that we can only compute the $\mathcal{L}_\infty$-norm of the closed loop system although we wish to minimize its $\mathcal{H}_\infty$-norm. As described in Section 3.3, we can compute the $\mathcal{L}_\infty$-norm in a computationally efficient way and the two norms have the same value if the system is asymptotically stable.

Thus, we need to ensure in every optimization step, that the controller is always stabilizing the closed loop system. This task is addressed within the line search of the optimization. The objective function only returns the $\mathcal{L}_\infty$-norm of the given transfer function if the system is asymptotically stable. Otherwise, `nan` is returned. In that case, the step size t^* is set to $t^*/2$ until all eigenvalues are in $\mathbb{C}^-$ and the $\mathcal{L}_\infty$-norm, which then coincides with the $\mathcal{H}_\infty$-norm is returned. Only then, a new descent direction is computed and the optimization is continued.

This ensures that the computed controllers always stabilize the given system because the step size can become arbitrarily small and the eigenvalues of the closed loop system depend continuously on the controller parameters. However, the computation time

increases, since `dde_biftool` must be used to check whether there are eigenvalues in the closure of the right half plane. Using profiling during the numerical experiments, it can be observed that between 40% and 60% of the computation time is spent to check for asymptotic stability of the closed loop system. In Figure 4.2 a couple of iterates of t^* that are obtained during the reduction of the step size due to instability is shown. Note that not all iterates are displayed but only a selection.

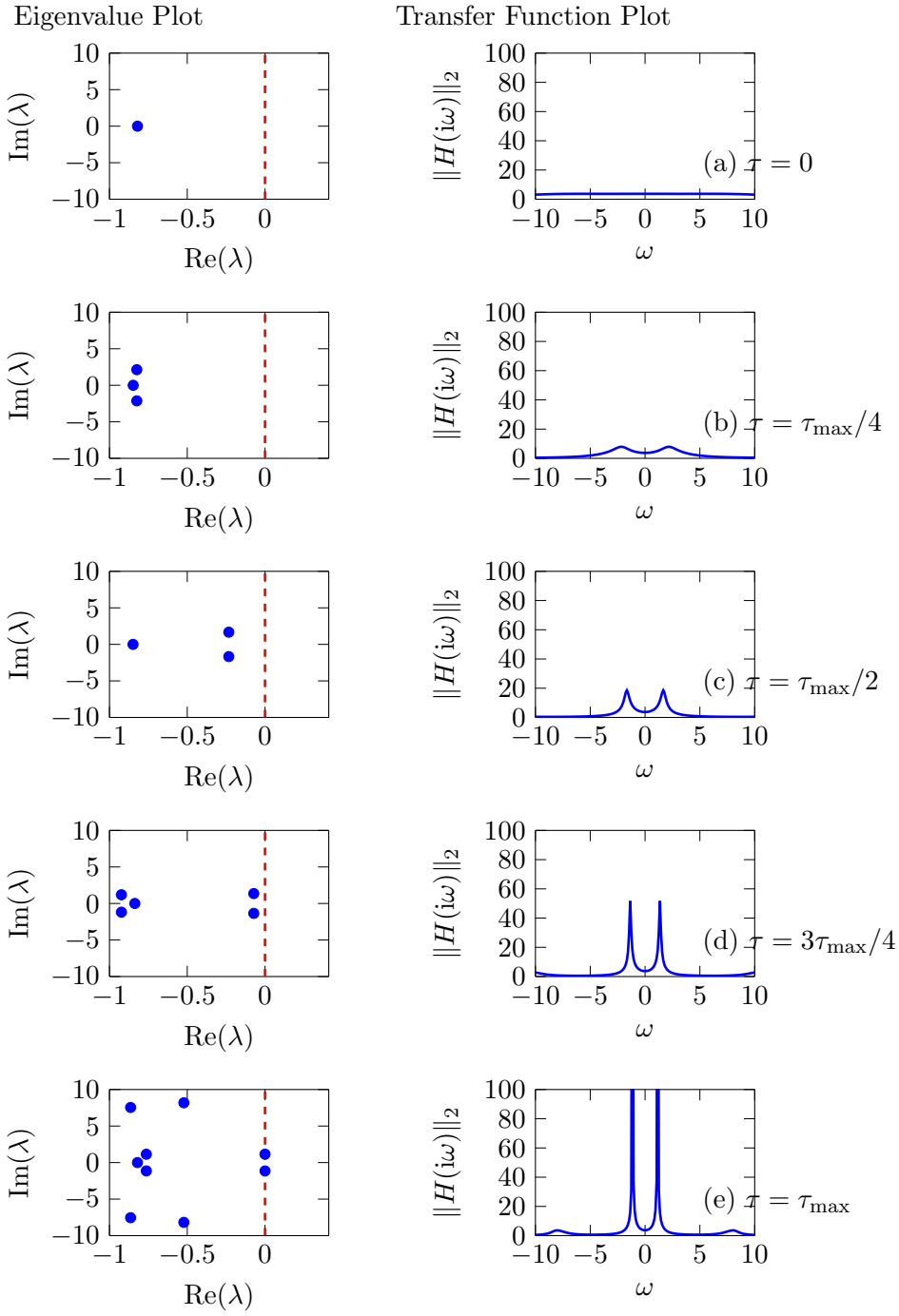
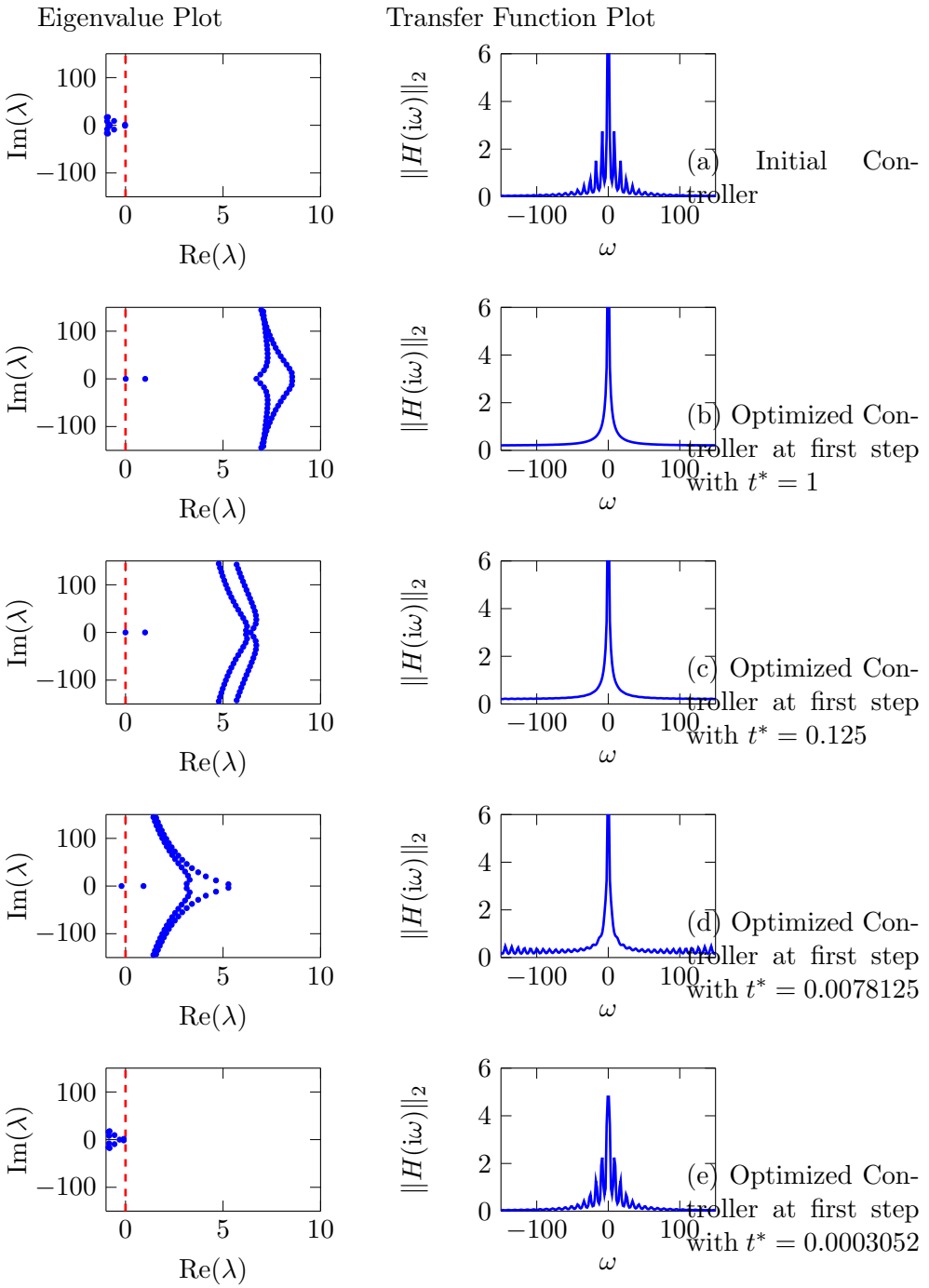


Figure 4.1: Plot of eigenvalues and transfer function to illustrate the connection between $\mathcal{L}_\infty$-norm and position of rightmost eigenvalues as the delay increases



(a) Initial Controller

(b) Optimized Controller at first step with $t^* = 1$

(c) Optimized Controller at first step with $t^* = 0.125$

(d) Optimized Controller at first step with $t^* = 0.0078125$

(e) Optimized Controller at first step with $t^* = 0.0003052$

Figure 4.2: Plot of eigenvalues and transfer function for different controllers that are obtained during the line search of the first optimization step with iteratively reduced step size

5 Implementation Details and Numerical Experiments

In this chapter we give insights into the implementation of the proposed method and compare it to the existing method for $\mathcal{H}_\infty$-optimal controller design for delay systems developed in [23]. Even though both algorithms yield similar results, Table 5.1 indicates that the method described in this thesis is computationally more efficient.

5.1 Implementation Details

The method developed in this work was implemented in the MATLAB programming language, since the used software packages `granso`, `dde_biftool`, and `linorm_subsp` are implemented in MATLAB as well. In the following, we share some implementation details and show, how the software packages for $\mathcal{L}_\infty$-norm computation and numerical optimization, respectively, are connected to perform controller design.

At the center of the implementation is the objective function. It is used to connect the software packages for $\mathcal{L}_\infty$-norm computation and numerical optimization. In order to be passed to a numerical optimization algorithm, the objective function must only depend on the free controller parameters. However, the $\mathcal{L}_\infty$-norm of the closed loop transfer function and the stability of the closed loop system are also dependent on parameters that may not be changed in numerical optimization such as the non-free controller parameters and the given plant. This conflict is resolved by using a function that is built upon a general objective function. This is described in Section 5.1.2. In the following, we show how the distinction between fixed and free controller parameters is made, i. e. how the position of free controller parameters within the given controller matrices $\left[A_c, B_c, C_c, D_c\right]$ is extracted from the user input.

5.1.1 Fixed Structure Controller

In accordance with the convention used in the software package `hifoo`, the free parameters are given as `nan` values within the controller matrices and the fixed parameters are just given within the controller matrices at the desired value. As an example the matrix A of such a controller can be given as

$$\texttt{K.A} = \begin{bmatrix} 1 & 2 & \texttt{nan} \\ 0 & 1 & \texttt{nan} \\ 0 & 0 & \texttt{nan} \end{bmatrix}. \tag{5.1}$$

In order to store the position of the free parameters, we build the block matrix

$$\texttt{BK} = \begin{bmatrix} \texttt{K.A} & \texttt{K.B} \\ \texttt{K.C} & \texttt{K.D} \end{bmatrix} \tag{5.2}$$

and find the position of the `nan`-valued entries. In this way, only one vector containing information on the position of the free parameters needs to be passed instead of one for each controller matrix. In order to construct the controller from a parameter vector, its entries can simply be inserted at the previously determined positions.

5.1.2 Objective Function

In order to perform optimization, an objective function needs to be provided that is only dependent on the controller parameters. However, as mentioned above, the asymptotic stability and the $\mathcal{L}_\infty$-norm of the closed loop system, or its transfer function, respectively, are not only dependent on the free parameters, but the fixed parameters and, naturally, the given plant as well. This is addressed using two different functions. Specifically, a function where Algorithm 6 is executed is implemented as a MATLAB function that is dependent on the plant and all controller parameters. Additionally, options for the computation of the $\mathcal{L}_\infty$-norm using the method from [38] as well as the previously determined controller structure may be passed. This gives a function with the function call

```
[ f, gradf ] = objectiveFunction( plant, freeParameters, options ).
```

In order to obtain a function that may be passed to `granso`, in the main routine another function `specificObjective` is created using

```
specificObjective = @( freeParameters ) objectiveFunction( plant,...
                    freeParameters, options ).
```

To increase the computational speed in our implementation we use that the maximizer for the $\mathcal{L}_\infty$-norm often does not to change drastically when the controller parameters are varied during numerical optimization. This can be exploited using the method from [38], since good starting values lead to a faster convergence to the maximizer and therefore, the $\mathcal{L}_\infty$-norm can also be computed in a faster way. This is implemented using a *persistent* variable for the maximizer $\mathrm{i}\omega^*$. In this way, $\mathrm{i}\omega^*$ can be saved for the next call of the objective function and added to the initial interpolation points that are used in `linorm_subsp` for the computation of the $\mathcal{L}_\infty$-norm.

5.2 Numerical Experiments

In order to test the developed algorithm and its implementation, we consider a couple of benchmark systems from the COMPLEIB library[1] (cf. [27]) that were also used to prove the effectiveness of `hifoo` and add delay to the control systems as described in

[1] available at `http://www.compleib.de/`

Section 4.1. The COMPLEIB library contains systems of the form described in (2.5) with state space dimension between 4 and 40. The stability properties of the resulting systems have been investigated in Theorem 4.1. In this way, we obtain test examples and, more importantly, using Algorithm 5, initial controllers that already stabilize the system, which is necessary to perform $\mathcal{H}_\infty$-norm optimization. For our experiments we restrict the controller state dimension n_c to one. Therefore, the number of optimizable controller parameters is between six and seventeen. We compare our implementation to the software package `tds_hiopt`. Each test example is created as follows:

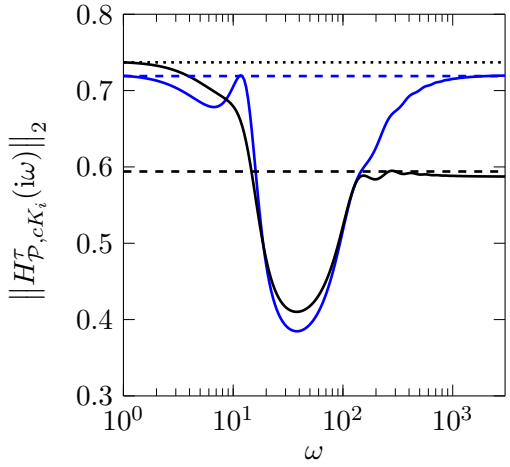
1. Find an initial controller for the given plant with no delay using `hifoo`.
2. Determine a bound for $\tau_{\max}$ at which the initial controller still stabilizes the system.
3. Set the controller's input delay to $0.9 \cdot \tau_{\max}$. At $\tau_{\max}$ the problem may be ill conditioned since `linorm_subsp` finds the maximum of the transfer function on the imaginary axis which can be infinite if there exists an imaginary axis eigenvalue. Then both $\mathcal{H}_\infty$-optimization methods are started for the delay system using the same initial controller.

This procedure creates the same starting conditions for the comparison of both optimization methods. Furthermore, existing benchmarks that were created for $\mathcal{H}_\infty$-optimization of systems of the standard form given in (2.5) can be reused to test procedures for $\mathcal{H}_\infty$-optimization of delay systems which increases the number of available test examples.

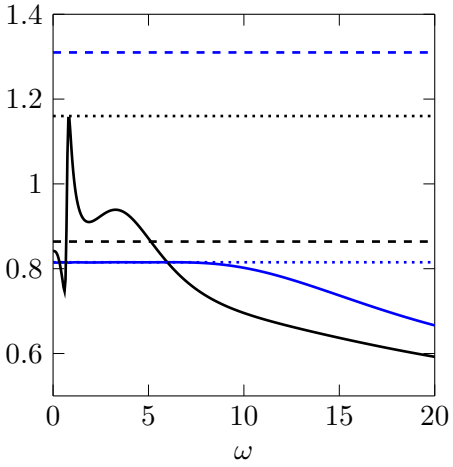
Note that we do not consider example AC10. This is due to the fact that no initial stabilizing controller of the desired order one was found for the system without delay in the stabilization phase of `hifoo`.

In Table 5.1 a comparison between `tds_hiopt` and our method is presented with respect to the runtime and the $\mathcal{H}_\infty$-performance of the closed loop system. Another distinction is made between the two types of gradient approximation methods that we propose in Algorithm 6. Our experiments have been performed on a machine with an AMD Phenom™ II X6 1090T processor with 6 cores that can run up to 3200 MHz with 8 GB RAM running MATLAB 2017a (9.2.0.538062). As expected, the computation time of our method is much lower. This is due to the fact that the computation of the $\mathcal{L}_\infty$-norm within the optimization steps is much faster with `linorm_subsp` than using `tds_hinfn` (run with standard options) which is used within `tds_hiopt` to compute the $\mathcal{L}_\infty$-norm during optimization. The ratio between the runtime of `tds_hiopt` and our method (with fast gradient computation enabled) that is shown in the last column of Table 5.1 indicates that our method is from around 60 times up to around 14500 times faster than `tds_hiopt`.

However, more important than a small computation time is the closed loop performance, i. e. the $\mathcal{H}_\infty$-norm of the closed loop transfer function. The best performance for each test example is highlighted in Table 5.1. It is remarkable, that only for three test examples, all methods have lead to approximately the same closed loop performance



(a) Closed Loop Transfer Functions for AC4



(b) Closed Loop Transfer Functions for HE3

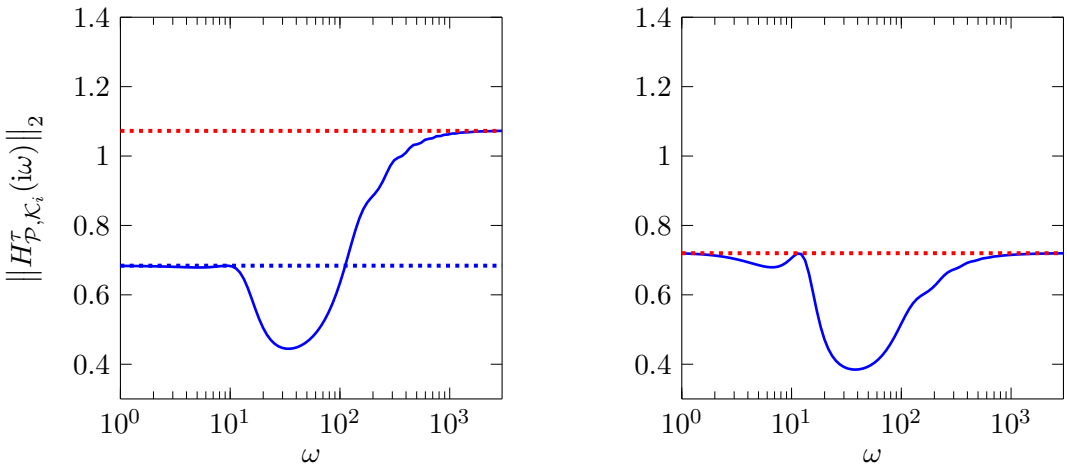
Figure 5.1: A comparison of closed loop transfer functions with controllers obtained using either our method (depicted in blue) or with `tds_hiopt` (depicted in black) is shown. Furthermore, the actual $\mathcal{H}_\infty$-norms computed with `linorm_subsp` are shown as dotted lines in the corresponding color. The $\mathcal{H}_\infty$-norms computed with `tds_hinfn` are depicted as dashed lines.

despite the fact that all methods are given the same initial value for the optimization. However, in the remaining cases the closed loop performance is mostly of the same order. An explanation for this is that the optimization problems in fixed order controller synthesis are nonconvex such that even slight differences in the gradient computation may lead to convergence to different local optima. For almost every test example, our method has found better or equally performing controllers than `tds_hiopt`.

Since `linorm_subsp` only converges locally to a maximizer of the Euclidean norm of a given transfer function, in Table 5.1 we also show the $\mathcal{H}_\infty$-norm computed with `tds_hinfn` to make sure that in fact the controllers determined with our method achieve a better performance. The computed values differ slightly for most examples which is due to a less accurate computation in `tds_hinfn`, however the ranking with regard to performance is not affected by these changes except for Examples AC4 and HE3. Here, `tds_hinfn` seems to indicate that `tds_hiopt` is actually providing a controller with a better performance. However, the transfer functions that are shown in Figure 5.1 reveal that `tds_hinfn` just does not compute the $\mathcal{H}_\infty$-norm correctly for these examples with the standard options enabled and the values computed with `linorm_subsp` are in fact correct. The overall lower accuracy of `tds_hinfn` can be seen as reason for the weaker performance of the controllers found with `tds_hiopt`, since the gradient based optimization algorithm is designed for exact optimization problems.

	$\mathcal{H}_\infty$-norm `linorm_subsp`			$\mathcal{H}_\infty$-norm `tds_hiopt`			computation time in s			
Name	`hiopt`	new	new, FG	`hiopt`	new	new, FG	`hiopt`	new	new, FG	ratio
AC1	1.55e-02	1.16e-02	9.86e-03	1.55e-02	1.16e-02	9.85e-03	3.3e+04	2.9e+02	8.8e+01	373.4
AC2	1.28e-01	1.30e-01	1.11e-01	1.28e-01	1.30e-01	1.11e-01	2.7e+04	2.8e+02	7.0e+01	380.3
AC3	4.67e+00	8.03e+00	8.15e+00	4.66e+00	8.03e+00	8.14e+00	3.8e+04	7.6e+02	1.7e+01	2265.5
AC4	7.37e-01	7.64e-01	7.20e-01	5.94e-01	7.64e-01	7.19e-01	3.0e+04	5.6e+01	1.5e+01	2017.7
AC5	6.77e+02	5.86e+03	6.76e+02	6.76e+02	5.86e+03	6.76e+02	3.0e+04	1.2e+01	1.1e+01	2787.9
AC6	5.83e+00	5.62e+00	5.16e+00	5.83e+00	5.62e+00	5.15e+00	4.1e+04	5.7e+02	2.9e+02	141.2
AC7	1.16e-01	1.16e-01	1.16e-01	1.16e-01	1.16e-01	1.16e-01	4.8e+04	6.1e+01	1.2e+01	3996.5
AC8	2.13e+00	2.07e+00	2.07e+00	2.13e+00	2.07e+00	2.07e+00	4.2e+04	1.3e+02	7.9e+01	530.0
AC9	1.08e+00	1.08e+00	1.05e+00	1.08e+00	1.08e+00	1.05e+00	4.9e+04	3.1e+02	2.3e+02	218.6
AC11	2.84e+00	2.83e+00	2.83e+00	2.83e+00	2.83e+00	2.83e+00	2.6e+04	7.1e+01	2.1e+01	1241.4
AC12	3.14e-01	3.14e-01	3.13e-01	3.14e-01	3.14e-01	3.13e-01	3.2e+04	1.6e+02	9.2e+01	351.6
AC13	2.92e+02	1.78e+02	1.71e+02	2.92e+02	1.78e+02	1.71e+02	7.9e+04	1.0e+04	1.0e+03	78.2
AC14	1.04e+02	1.34e+02	1.34e+02	1.04e+02	1.33e+02	1.34e+02	1.1e+05	2.0e+03	1.8e+03	60.9
AC15	1.54e+01	1.54e+01	1.51e+01	1.53e+01	1.54e+01	1.51e+01	3.3e+04	8.7e+01	2.4e+01	1384.2
AC16	1.52e+01	1.53e+01	1.50e+01	1.52e+01	1.53e+01	1.50e+01	3.6e+04	9.4e+01	1.3e+01	2709.9
AC17	6.72e+00	6.70e+00	6.70e+00	6.70e+00	6.70e+00	6.70e+00	3.5e+04	2.5e+02	2.7e+01	1285.3
AC18	7.77e+00	7.77e+00	7.77e+00	7.76e+00	7.77e+00	7.77e+00	3.4e+04	3.2e+01	6.9e+00	4986.5
HE1	1.26e-01	1.26e-01	1.26e-01	1.26e-01	1.26e-01	1.26e-01	3.1e+04	7.4e+00	2.1e+00	14502.3
HE2	3.52e+00	2.88e+00	2.48e+00	3.52e+00	2.88e+00	2.48e+00	2.6e+04	1.6e+02	5.2e+01	498.1
HE3	1.16e+00	8.27e-01	8.15e-01	8.46e-01	1.11e+00	1.31e+00	3.9e+04	3.2e+03	5.4e+02	72.5
HE4	2.36e+01	2.35e+01	2.28e+01	2.36e+01	2.35e+01	2.28e+01	5.3e+04	5.7e+02	7.5e+01	709.2
HE5	7.03e+00	2.56e+00	2.50e+00	6.22e+00	2.56e+00	2.50e+00	4.3e+04	9.1e+02	2.0e+02	216.5

Table 5.1: A comparison of runtime and $\mathcal{H}_\infty$-performance between the method presented in [23] and our method is made. We distinguish the two different methods to compute the gradient as in Algorithm 6. The fast gradient computation is indicated by FG. Furthermore, the best $\mathcal{H}_\infty$-performance is highlighted in columns 2–4. In the column entitled ratio, the quotient of the computation time of `hiopt` and our method with fast gradient computation enabled, is given.



(a) Closed loop transfer function of AC4 with a controller locally minimizing $\left\|H^{\tau}_{\mathcal{P},\mathcal{K}}(\mathrm{i}\cdot)\right\|_2$

(b) Closed loop transfer function of AC4 with a controller globally minimizing $\left\|H^{\tau}_{\mathcal{P},\mathcal{K}}(\mathrm{i}\cdot)\right\|_2$

Figure 5.2: A comparison between a controller, that only locally minimizes $\left\|H^{\tau}_{\mathcal{P},\mathcal{K}_i}(\mathrm{i}\cdot)\right\|_2$, and another controller, that minimizes $\left\|H^{\tau}_{\mathcal{P},\mathcal{K}_i}(\mathrm{i}\cdot)\right\|_2$ globally, is shown. The $\mathcal{H}_\infty$-norm of the two transfer functions is indicated with a red dotted line, while the local maximum of $\left\|H^{\tau}_{\mathcal{P},\mathcal{K}_i}(\mathrm{i}\cdot)\right\|_2$ that is actually optimized in the left plot is depicted as a blue dotted line. Note that in the right plot a balance between the frequencies for which $\left\|H^{\tau}_{\mathcal{P},\mathcal{K}}(\mathrm{i}\cdot)\right\|_2$ has high values is found.

5.3 Limitations of the Method

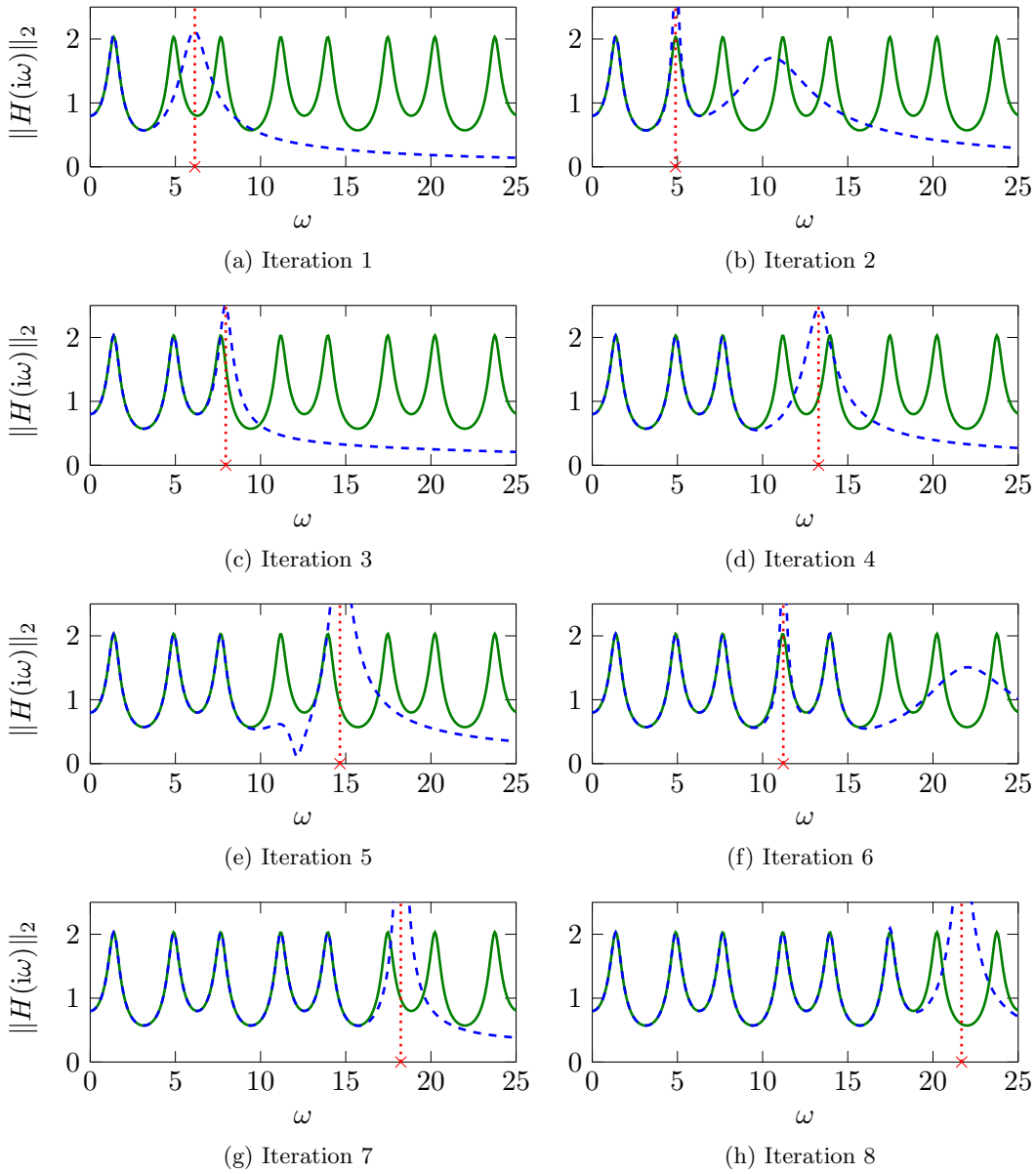
As mentioned in Section 3.3, the algorithm for the computation of the $\mathcal{L}_\infty$-norm developed in [35, 38] converges only to a local maximizer and may miss the global maximizer for $\|H(\mathrm{i}\cdot)\|_2$. An example for this behavior is given in [35]. Therefore, the controllers that are chosen to minimize the $\mathcal{H}_\infty$-norm of the closed loop transfer function may also just minimize the value of a local maximizer, when not enough interpolation points are given or if they are chosen within a disadvantageous frequency range. An example for this is shown in Figure 5.2. Here, first a controller for Example AC4 is determined using only initial interpolation points within an interval between 0 and 20. Since the magnitude of the transfer function is low in an interval between 10 and 100, `linorm_subsp` is then not taking the high values of the norm of the transfer function that occur at values for ω at around 1000 into account. Therefore, only a locally optimizing controller is determined. On the other hand, when another initial interpolation point at 1000 is passed to `linorm_subsp`, these high values are considered during the optimization as well and a controller minimizing the global maximum that is a controller minimizing $\left\|H^{\tau}_{\mathcal{P},\mathcal{K}}(\mathrm{i}\omega)\right\|_2$ on the entire domain $\omega \in \mathbb{R}$ is found.

Another drawback of using the interpolation approach is that transfer functions of

delay systems can have infinitely many peaks as shown in Figure 5.3 for the transfer function of a delay system given in [23]

$$H_{\text{periodic}}(s) := \frac{1}{1 - 0.25\mathrm{e}^{-s} + 0.5\mathrm{e}^{-2s}}.$$

In this case, the interpolation approach for $\mathcal{L}_\infty$-norm computation from [38] does not converge, since infinitely many local maximizers of similar and nondecreasing magnitude are found and lead to infinitely many candidates for the global optimum. This is shown in Figure 5.3 for the first few iterations.



(a) Iteration 1 (b) Iteration 2

(c) Iteration 3 (d) Iteration 4

(e) Iteration 5 (f) Iteration 6

(g) Iteration 7 (h) Iteration 8

Figure 5.3: Plot of the magnitude of the given periodic function (depicted in green) and its rational interpolation (depicted in blue) with diverging interpolation points when Algorithm 4 is applied.

6 A Generalization and Applications

In this chapter we show how the presented method can be used to compute initial stabilizing controllers for delay systems, even at values of τ where the original controller that is designed for the system without delay does not stabilize the system, anymore. Furthermore, we show how our method can be extended in order to solve control problems for a larger set of plants. In Section 6.2 we show that Theorem 4.1 can be extended to rewrite a DDAE to a DDE even in the case of multiple delays.

Thereafter, we show how a *weighted synthesis* in which not just the $\mathcal{H}_\infty$-norm of the closed loop transfer function is minimized, but also critical frequency ranges, where the magnitude of the transfer function should not exceed a previously specified value, can be introduced without changing the given model, which is usually necessary in order to perform a *weighted synthesis*.

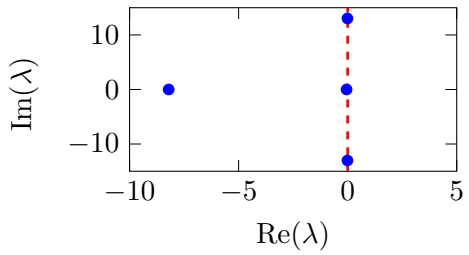
6.1 Iterative Method to Find an Initial Stabilizing Controller

The developed method for $\mathcal{H}_\infty$-norm optimization of closed loop systems with delay can also be applied to find initial controllers. Recall that we distinguished two different cases for a given τ^*. It is either small enough such that the controller that is designed for the system without delay still stabilizes the delayed system or it can be larger. When τ^* is larger, stabilization has to be performed prior to the $\mathcal{H}_\infty$-norm optimization. This can be addressed with continuous pole placement from [33]. However, using a controller to minimize the $\mathcal{H}_\infty$-norm of a closed loop transfer function can have the side effect of moving the real parts of the eigenvalues of the closed loop system to the left. This yields a controller that is designed to minimize the closed loop transfer function at τ^* may still stabilize the system for delays $\tau > \tau^*$.

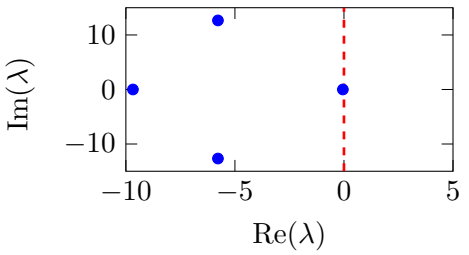
In this way, a sequence of controllers that stabilize the system for increasing values of τ is used for the stabilization of a system at a given delay value τ^* in the following example.

Example 6.1. *Consider the system* `AC4` *from the* COMPLEIB *library. The controller* $\mathcal{K}_0$ *computed with the method in [21] to minimize the* $\mathcal{H}_\infty$*-norm of the closed loop transfer function at* $\tau = 0$ *still stabilizes the system until* $\tau_{\max} = 0.0459$*. Assume an* $\mathcal{H}_\infty$*-optimization for* $\tau^* := 2\tau_{\max}$ *is wanted. Then, an intermediate controller* $\mathcal{K}_1$ *is computed at* $0.9 \cdot \tau_{\max}$ *such that the* $\mathcal{H}_\infty$*-norm of the closed loop transfer function of* $\mathbf{CL}^{0.9\tau_{\max}}(\mathcal{P}_{\texttt{AC4}}, \mathcal{K}_1)$ *is minimized. The eigenvalues of the closed loop system at* $\tau_{\max}$ *are shown in Figure 6.1 for both* $\mathcal{K}_0$ *(a) and* $\mathcal{K}_1$ *(b).*

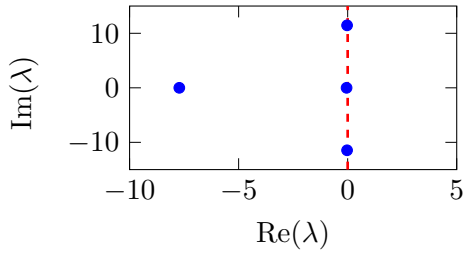
Next, τ *is increased until the closed loop* $\mathbf{CL}^{\tau_{\max}}(\mathcal{P}_{\texttt{AC4}}, \mathcal{K}_1)$ *is destabilized, which is at* $\tau = 1.88\tau_{\max}$*. Another controller* $\mathcal{K}_2$ *is then computed by using* $\mathcal{H}_\infty$*-norm minimization*



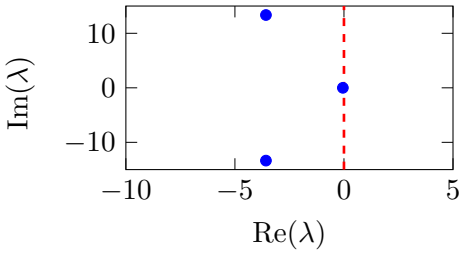
(a) Eigenvalues of closed loop system with controller $\mathcal{K}_0$ at $\tau = \tau_{\max}$



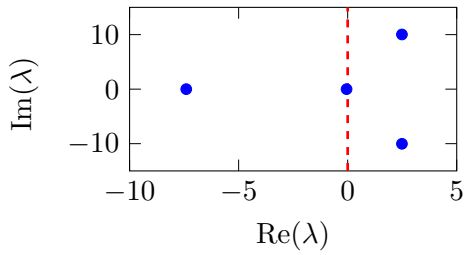
(b) Eigenvalues of closed loop system with controller $\mathcal{K}_1$ at $\tau = \tau_{\max}$



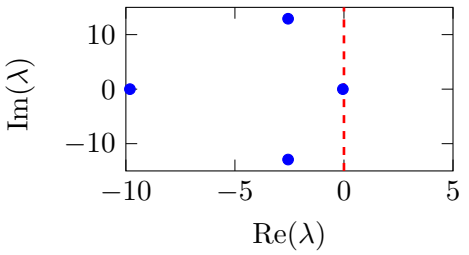
(c) Eigenvalues of closed loop system with controller $\mathcal{K}_1$ at $\tau = 1.88\tau_{\max}$



(d) Eigenvalues of closed loop system with controller $\mathcal{K}_2$ at $\tau = 1.88\tau_{\max}$



(e) Eigenvalues of closed loop system with controller $\mathcal{K}_0$ at $\tau = 2\tau_{\max}$



(f) Eigenvalues of closed loop system with controller $\mathcal{K}_2$ at $\tau = 2\tau_{\max}$

Figure 6.1: Eigenvalues of the closed loop system for increasing delay and adjusted controllers (the imaginary axis is depicted as red dashed line)

and the resulting positions of the eigenvalues of the closed loop system at $\tau = 1.88\tau_{\max}$ *is shown in Figure 6.1 for both* $\mathcal{K}_1$ *(c) and* $\mathcal{K}_2$ *(d). The controller* $\mathcal{K}_2$ *still stabilizes the closed loop for the given* τ^* *as shown in Figure 6.1 (f) and can be chosen as initial controller for a last* $\mathcal{H}_\infty$*-norm minimization, while the original controller* $\mathcal{K}_0$ *could not have been chosen as starting point for optimization since it is not stabilizing the plant at* τ^* *as shown in Figure 6.1 (e).*

Note that this method of iteratively increasing the delay cannot always be applied, since there is no guarantee that the maximal delay at which a controller still stabilizes the system is increased by using a controller that minimizes the $\mathcal{H}_\infty$-norm of the closed loop transfer function at $0.9 \cdot \tau_{\max}$. Furthermore, there might not even exist a stabilizing

controller if τ is too large.

6.2 Multiple Delays

The developed controller design method can immediately be extended to a broader class of systems. This is shown in the following section using systems with multiple delays, that may include systems with not only input or output delay but also systems with state/internal delay such as

$$\begin{aligned}\dot{x}(t) &= A_0 x(t) + \sum_{k=1}^{m} A_k x(t-\tau_k) + Bu(t),\\ y(t) &= Cx(t) + Du(t).\end{aligned}$$

In the following, we show how the generalization of the developed approach for such systems can be done. The general form of a plant $\mathcal{P}$ from (2.5) is expanded to

$$\mathcal{P}_e : \begin{cases}\dot{x}(t) = A_0 x(t) + \sum\limits_{k=1}^{m} A_k x(t-\tau_k) + B_1 w(t) + Bu(t),\\ z(t) = C_1 x(t) + D_{11} w(t) + D_{12} u(t),\\ y(t) = Cx(t) + D_{21} w(t) + D_{22} u(t).\end{cases} \tag{6.1}$$

The form of the controller $\mathcal{K}$ is not affected by this generalization. It remains to determine the closed loop $\mathbf{CL}(\mathcal{P}_e, \mathcal{K})$ and to generalize Theorem 4.1. The DDAE that is used to describe $\mathbf{CL}(\mathcal{P}_e, \mathcal{K})$ is similar to (4.5). The closed loop system for $\mathbf{CL}(\mathcal{P}_e, \mathcal{K})$ is given by

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\underbrace{\begin{bmatrix} I & 0 & 0 & 0\\ 0 & I & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix}}_{\mathcal{E}}\begin{bmatrix} x(t)\\ x_c(t)\\ u(t)\\ y(t)\end{bmatrix} &= \underbrace{\begin{bmatrix} A & 0 & B & 0\\ 0 & A_c & 0 & 0\\ 0 & C_c & -I & 0\\ C & 0 & D_{22} & -I\end{bmatrix}}_{\mathcal{A}_0}\begin{bmatrix} x(t)\\ x_c(t)\\ u(t)\\ y(t)\end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0\\ 0 & 0 & 0 & B_c\\ 0 & 0 & 0 & D_c\\ 0 & 0 & 0 & 0\end{bmatrix}}_{\mathcal{A}_1}\begin{bmatrix} x(t-\tau)\\ x_c(t-\tau)\\ u(t-\tau)\\ y(t-\tau)\end{bmatrix}\\ &\quad + \sum_{k=1}^{m}\left(\underbrace{\begin{bmatrix} A_k & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix}}_{\mathcal{A}_{k+1}}\begin{bmatrix} x(t-\tau_k)\\ x_c(t-\tau_k)\\ u(t-\tau_k)\\ y(t-\tau_k)\end{bmatrix}\right) + \underbrace{\begin{bmatrix} B_1\\ 0\\ 0\\ D_{21}\end{bmatrix}}_{\mathcal{B}} w(t),\\ z(t) &= \underbrace{\begin{bmatrix} C_1 & 0 & D_{12} & 0\end{bmatrix}}_{\mathcal{C}}\begin{bmatrix} x(t)\\ x_c(t)\\ u(t)\\ y(t)\end{bmatrix} + \underbrace{D_{11}}_{\mathcal{D}} w(t).\end{aligned} \tag{6.2}$$

Since the additional matrices $\mathcal{A}_2, \ldots, \mathcal{A}_{m+1}$ do not affect the algebraic part, the proof of Theorem 4.1 caries over to the following extension.

Theorem 6.2. *Let ν be the maximum of the nilpotency indices of $D_c D_{22}$ and $D_{22} D_c$. Then all solutions $\lambda \in \mathbb{C}$ of $\det\left(\lambda\mathcal{E} - \mathcal{A}_0 - \mathrm{e}^{-\lambda\tau}\mathcal{A}_1 - \sum_{k=1}^{m} \mathrm{e}^{-\lambda\tau_k}\mathcal{A}_{k+1}\right) = 0$, with all matrices defined as in (6.2), are also solutions of*

$$\det\left(\lambda I - A_0 - \sum_{j=1}^{\nu} \mathrm{e}^{-j\lambda\tau} A_j - \sum_{k=1}^{m} \tilde{A}_k \mathrm{e}^{-\lambda\tau_k}\right) = 0,$$

with

$$A_j = \begin{cases} \begin{bmatrix} A & BC_c \\ 0 & A_c \end{bmatrix} & \text{for } j = 0, \\ \begin{bmatrix} BD_c(D_{22}D_c)^{j-1}C & BD_cD_{22}(D_cD_{22})^{j-1}C_c \\ B_c(D_{22}D_c)^{j-1}C & B_cD_{22}(D_cD_{22})^{j-1}C_c \end{bmatrix} & \text{for } j > 0 \end{cases}$$

and

$$\tilde{A}_k = \begin{bmatrix} A_k & 0 \\ 0 & 0 \end{bmatrix} \text{ for } k = \{1, \ldots, m\}$$

and vice versa.

In this way, we can compute the $\mathcal{L}_\infty$-norm of the closed loop transfer function and, by Theorem 6.2, using the method described in [17], check the stability of the closed loop system. Therefore, the algorithm is applicable for systems with internal delay as well.

6.2.1 Strong $\mathcal{H}_\infty$-Norm

In [23] it is shown, that the $\mathcal{H}_\infty$-norm is not always continuous with respect to changes in the delay when the system contains multiple delays. Therefore, the *strong* $\mathcal{H}_\infty$-norm (cf. [23, Definition 4.4]), which is robust with respect to small delay changes, needs to be considered as well.

Since the discontinuity is due to the behavior of the transfer function for large frequencies, we take a look at the *asymptotic* transfer function of a system of the form (6.2), i. e.

$$\begin{aligned} E\dot{x}(t) &= \sum_{k=0}^{m} A_k x(t - \tau_k) + Bw(t) \\ z(t) &= Cx(t) + Dw(t), \end{aligned} \tag{6.3}$$

with $\operatorname{rank} E = \tilde{n} - \nu$ and $\nu \leq \tilde{n}$. Let $U, V \in \mathbb{R}^{\nu \times \tilde{n}}$ be a basis for the left and right nullspaces of E, respectively, such that

$$U^{\mathsf{T}} E = 0, \quad EV = 0.$$

Then, if $\tau_0 = 0$ and A_0 is nonsingular, the asymptotic transfer function of (6.3) is given by

$$H_a(s) := -CV \left(U^{\mathsf{T}} A_0 V + \sum_{k=1}^{m} U^{\mathsf{T}} A_k V \mathrm{e}^{-s\tau_k} \right)^{-1} U^{\mathsf{T}} B + D,$$

which is the projection of (6.3) into the nullspace of E. It is shown in [23] that for large frequencies, the transfer function converges to the asymptotic transfer function.

In order to study infinitesimally small delay changes, we define *rationally independent* numbers as in [23].

Definition 6.3. *Real numbers* $\{\tau_1, \ldots, \tau_m\}$ *are* rationally independent*, if and only if* $\sum_{k=1}^{m} z_k \tau_k = 0$ *with* $z_k \in \mathbb{Z}$ *implies* $z_k = 0$ *for all* k*. If real numbers are not rationally independent, they are called* rationally dependent*.*

Note that rationally dependent numbers are always arbitrarily close to rationally independent ones (cf. [23]).

It is now easy to see that for rationally independent τ_k, the image of $H_a(\mathrm{i}\cdot)\big|_{\mathbb{R}}$, is the same as the image of

$$\mathbb{H}_a(v) := -CV \left(U^{\mathsf{T}} A_0 V + \sum_{k=1}^{m} U^{\mathsf{T}} A_k V \mathrm{e}^{\mathrm{i}v_k} \right)^{-1} U^{\mathsf{T}} B + D,$$

with $v = \begin{bmatrix} v_1, \ldots, v_m \end{bmatrix}^{\mathsf{T}} \in [0, 2\pi]^m$. However, in the case of rationally dependent τ_k, the image of the asymptotic transfer function is only a subset of the image of $\mathbb{H}_a$ and therefore its $\mathcal{H}_\infty$-norm can be smaller. Since rationally independent numbers are always arbitrarily close to rationally dependent ones, the $\mathcal{H}_\infty$-norm of H_a is sensitive to arbitrarily small delay changes. This is shown in [23, Example 4.1]. Therefore, the maximum value of $\sigma_{\max}(\mathbb{H}_a(v))$, which can be computed by evaluating $\mathbb{H}_a$ on a grid in $[0, 2\pi]^m$, must also be taken into account. The *strong* $\mathcal{H}_\infty$-norm is then determined by computing the $\mathcal{H}_\infty$-norm of the given transfer function and the maximum value of $\sigma_{\max}(\mathbb{H}_a(v))$ for all $v \in [0, 2\pi]^m$. Then the strong $\mathcal{H}_\infty$-norm is the maximum of these two values. Since our objective function can be extended in the implementation such that not only `linorm_subsp` is run but also $\mathbb{H}_a$ can be computed and evaluated on a grid in $[0, 2\pi]^m$ the computation of the strong $\mathcal{H}_\infty$-norm can easily be added to the developed algorithm.

6.2.2 Test Examples

We show the applicability of the implementation of our method for systems with internal delay and for the multiple delay case based on an example given in [22]. This

system is defined as

$$\dot{x}(t) = -x(t) - 0.5x(t-1) + w(t) + u(t), \tag{6.4}$$
$$z(t) = x(t) + u(t), \tag{6.5}$$
$$y(t) = x(t) + w(t),$$

and the task stated in [22] is to design a first-order controller with the constraint $D_c = 0$.

Using our method, we design the first-order controller

$$\dot{x}_c(t) = -3.61x_c(t) + 1.07y(t),$$
$$u(t) = -1.07x_c(t).$$

The transfer function of our controller coincides with the transfer function of the one computed in [22] (note that in [22] a minus sign in front of $x_c(t)$ is missing), and their $\mathcal{H}_\infty$-performance is the same which implies that both methods converge to the same local optimizer. The closed-loop transfer function has an $\mathcal{H}_\infty$-norm of 0.064. If we extend the system by a second internal delay, i. e. replace (6.4) and (6.5) by

$$\dot{x}(t) = -x(t) - 0.3x(t-0.5) - 0.5x(t-1) + w(t) + u(t),$$
$$z(t) = x(t) + u(t) - 0.1w(t),$$

and use our method to find a controller for this modified plant with multiple delays, we obtain a similar controller given by

$$\dot{x}_c(t) = -2.90x_c(t) + 0.90y(t),$$
$$u(t) = -0.90x_c(t),$$

and a closed loop $\mathcal{H}_\infty$-performance of 0.1307. Note that the strong $\mathcal{H}_\infty$-norm of the asymptotic transfer function is 0.10 which can be computed analytically from the closed loop DDAE defined by

$$\begin{aligned}
\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathcal{E}} \dot{\tilde{x}}(t) &= \begin{bmatrix} -1 & 0 & 1 & 0 \\ 0 & -2.90 & 0 & 0.90 \\ 0 & -0.90 & -1 & 0 \\ 1 & 0 & 0 & -1 \end{bmatrix} \tilde{x}(t) + \sum_{i=1}^{2} \left(\underbrace{\begin{bmatrix} a_i & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathcal{A}_i} \tilde{x}(t-\tau_i) \right) \\
&\qquad + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} w(t), \\
z(t) &= \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix} \tilde{x}(t) - 0.1w(t),
\end{aligned} \tag{6.6}$$

with $\tilde{x} = [x^\mathsf{T}, x_c^\mathsf{T}, u^\mathsf{T}, y^\mathsf{T}]$, $a_1 = -0.3$, $a_2 = -0.5$, $\tau_1 = 0.5$, and $\tau_2 = 1$. Since in (6.6) the left and right nullspace of $\mathcal{E}$ is also in the kernel of both $\mathcal{A}_1$ and $\mathcal{A}_2$, we have that the asymptotic transfer function is given by

$$H_a(s) := -\begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}-1 & 0\\ 0 & -1\end{bmatrix}\begin{bmatrix}0\\ 1\end{bmatrix} - 0.1,$$

which is constantly -0.1 and therefore its $\mathcal{H}_\infty$-norm 0.1, independent of both τ_1 and τ_2. This is common for delay systems with only internal delay. In practical use, the asymptotic transfer function and perturbations in the delay parameters need only be considered in case of direct feedthrough over the control loop (cf. [23]). However, this often leads to a transfer function with countably many nondecreasing peaks, where as shown in Section 5.3 the interpolation approach and thus the method developed in this thesis cannot be applied.

6.3 Weighted Controller Synthesis

In this section the $\mathcal{H}_\infty$-sensitivity design problem is addressed. We show how weight functions that are used to improve the controller performance in a relevant frequency range, can be included into our implementation.

6.3.1 Weighted Sensitivity Function and Mixed Synthesis

We briefly introduce notions and common design goals in feedback control. The sensitivity function S is the transfer function from the noise input to the control error. Consider the block diagram in Figure 6.3. In the following, we investigate the influence of the noise input w on the output y. From the block diagram we have that

$$Y(s) = W(s) - G(s)K(s)Y(s), \tag{6.7}$$

which can be rearranged such that we have

$$Y(s) = (I + G(s)K(s))^{-1}W(s), \tag{6.8}$$

and define the sensitivity as $S(s) := (I + G(s)K(s))^{-1}$, which is well defined on a domain enclosing the imaginary axis if the closed loop system is asymptotically stable. If its magnitude $\|S(\mathrm{i}\omega)\|_2$ is low for an $\omega \in \mathbb{R}$, so is the influence of the disturbance w on y for that ω. Since the design objectives are different in different frequency ranges, it is necessary to not only consider the $\mathcal{H}_\infty$-norm of S when synthesizing a controller for the plant but also use a *performance weight* that reflects the different demands for S in the different frequency ranges. A typical *performance weight function* used in [39] is given by

$$w_P(s) = \frac{s/e_\infty + \omega_B^*}{s + \omega_B^* e_0}. \tag{6.9}$$

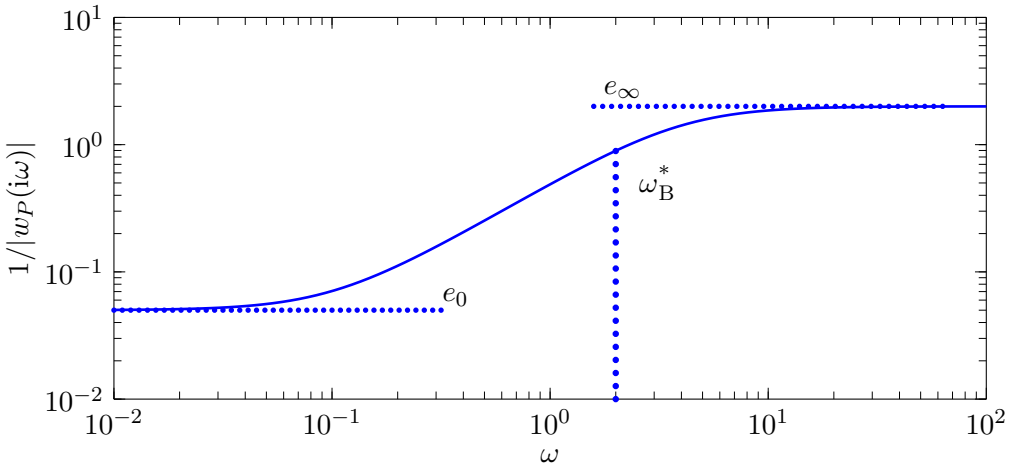


Figure 6.2: Inverse of $|w_P|$ and its asymptotic behavior

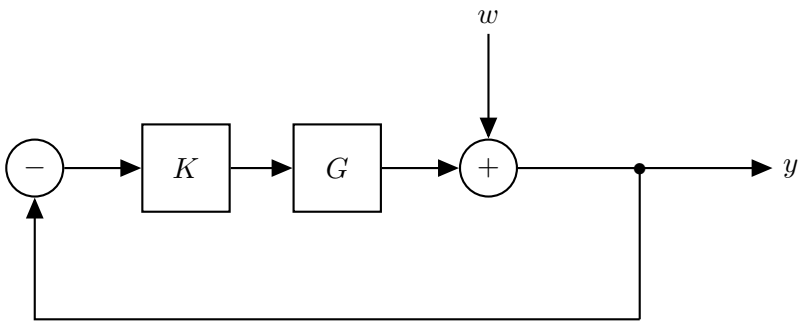


Figure 6.3: Block diagram to visualize the sensitivity function

The inverse $1/|w_P|$ is shown in Figure 6.2.

Note that if $\|w_P S\|_{\mathcal{H}_\infty} < 1$, then the maximum singular value of the sensitivity function is always below the bound that is given by $1/|w_P|$. In this way, choosing constants e_0, e_∞, and ω_B^*, allows to specify control objectives such as a maximum steady state tracking error e_0 or a minimum bandwidth ω_B^*.

In order to restrict the magnitude of the control signal $\|u\|_{\mathcal{L}_2}$ an upper bound $1/|w_u|$ can be placed on the transfer function KS in order to synthesize a controller that is actually implementable and does lead to prohibitively high values of $\|u\|_{\mathcal{L}_2}$ when controlling the given system. When both $\|w_P S\|_{\mathcal{H}_\infty}$ and $\|w_u KS\|_{\mathcal{H}_\infty}$ are supposed to be minimized, then both transfer functions are stacked to form one *mixed sensitivity* function that is given by

$$N = \begin{bmatrix} w_P S \\ w_u K S \end{bmatrix}. \tag{6.10}$$

In this way, the $\mathcal{H}_\infty$-optimal controller is synthesized to minimize the $\mathcal{H}_\infty$-norm of N. Note that when using `linorm_subsp` for the computation of the $\mathcal{H}_\infty$-norm of N

no realization of N needs to be computed since the method uses interpolation and is therefore realization independent.

6.3.2 Weighted Synthesis for a Delay System with Reference Tracking

In order to show how the realization independence facilitates the use of weight functions consider the system shown as block diagram in Figure 6.4. As H-block, we are given a dynamical system

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

with the transfer function as in [39, Example 2.17] given by

$$H(s) := \frac{200}{0.025s^3 + 1.002s^2 + 10.1s + 1}.$$

We obtain the general control problem as defined in (2.5) from the block diagram and by shifting the delay block represented by $\mathrm{e}^{-\tau}$ to the controller input equation as shown in (4.3). In this way, $\mathcal{P}$ is given by

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t),\\ y(t) &= -Cx(t) + w(t),\\ z(t) &= -Cx(t) + w(t).\end{aligned}$$

Note that with this setup, the transfer function from the input w, which in this case is the reference, to w_P is in fact the sensitivity function, if we replace the K-block by $-K$ and set the delay to 0. Therefore, in order to specify the bandwidth and steady state tracking error, the weight function w_P is chosen such that the desired specifications are met, if the closed loop transfer function from w to z_P has an $\mathcal{H}_\infty$-norm less than 1 which means that for all $\omega \in \mathbb{R}$ the gain from noise input to the regulated output is below the given bound defined by $1/|w_P|$. This is also true for systems with delay, since the performance specification is based on the transfer function and not on the realization.

The computation of that $\mathcal{H}_\infty$-norm and, therefore, the optimization of the controller parameters are directly implementable into the developed routine, since the algorithm for the $\mathcal{H}_\infty$-norm computation from [38] is realization independent, such that no structural constraints are imposed on the system. The closed loop system and its transfer function $H^\tau_{\mathcal{P},\mathcal{K}}$ can be computed as before in Section 4.1. Now, for each interpolation point $\mathrm{i}\omega_k$ with $k \in (1, \ldots, r)$ we pass

$$w_P(\mathrm{i}\omega_k) \cdot H^\tau_{\mathcal{P},\mathcal{K}}(\mathrm{i}\omega_k)$$

as interpolation data to Algorithm 4 to construct the Loewner matrices as defined in (3.7). This slight change enables the use of weight functions without the need of

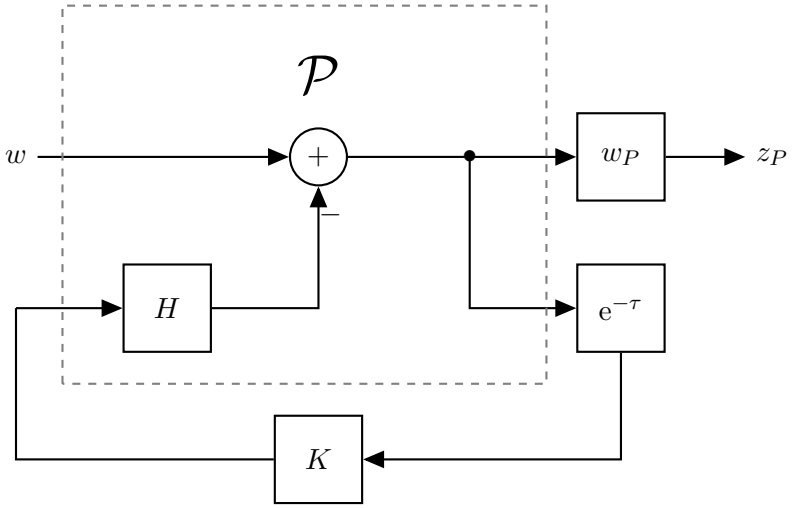


Figure 6.4: Block diagram representing the weighted sensitivity setup

integrating them into $\mathcal{P}$ as in classical $\mathcal{H}_\infty$-norm optimization which is useful especially in the delay case, when the structure of the plant cannot always be given as in (2.5).

We proceed with [39, Example 2.17], by introducing an output delay $\tau := 0.1$ as shown in Figure 6.4 and defining performance specifications using w_P. The maximum steady state tracking error A is chosen to be 0.001 and the bandwidth ω_B^* is set to 1.

First, a stabilizing controller $\mathcal{K}_1$ with $n_c = 1$ is designed. This can be done using continuous pole placement as mentioned in Section 4.1 or even by hand, since the system is already asymptotically stable without control. It is chosen as

$$\mathcal{K}_1 : \begin{cases} \dot{x}_c(t) = -0.1x_c(t) + 0.01y(t), \\ u(t) = x_c(t). \end{cases}$$

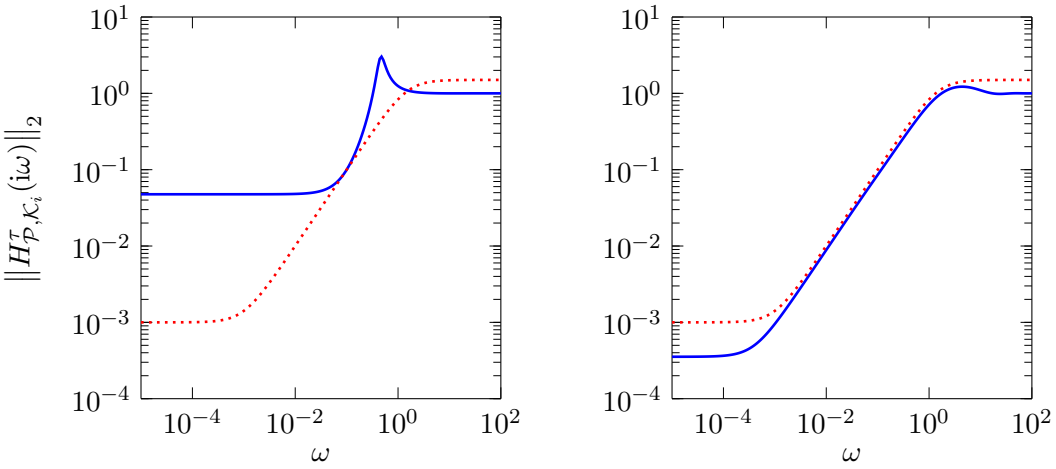
After that, we use our method with the modified objective function to minimize $\|w_P H^\tau_{\mathcal{P},\mathcal{K}}\|_{\mathcal{H}_\infty}$ by changing the controller parameters. The resulting controller is called $\mathcal{K}_2$ and is given by

$$\mathcal{K}_2 : \begin{cases} \dot{x}(t) = -0.00061x(t) + 0.0057y(t), \\ u(t) = 0.99x(t) + 0.061y(t). \end{cases}$$

In Figure 6.5 the absolute values of the transfer functions $H^\tau_{\mathcal{P},\mathcal{K}_1}$, $H^\tau_{\mathcal{P},\mathcal{K}_2}$, and of the inverted weight which defines the upper bound for the magnitude of the transfer function for a desirable controller are shown.

It can be seen, that using $\mathcal{K}_1$ both the steady state tracking requirement and the bandwidth requirement are not met. However, after optimization a controller of the same order is found such that both requirements are satisfied.

Considering the realization independence of `linorm_subsp`, the other typical requirements such as noise rejection or disturbance rejection can be introduced just like the



(a) Comparison of the inverted weight function and $H^{\tau}_{\mathcal{P},\mathcal{K}_1}$

(b) Comparison of the inverted weight function and $H^{\tau}_{\mathcal{P},\mathcal{K}_2}$

Figure 6.5: Comparison of the performance of the controller designed by hand with the optimized controller

sensitivity requirements by introducing the design objectives as additional outputs of the plant as shown in [39, Section 9.3].

7 Conclusion and Outlook

We have developed and implemented an algorithm to compute fixed order controllers for delay systems that minimize the $\mathcal{H}_\infty$-norm of the closed loop transfer function using direct numerical optimization. The developed method benefits from an interpolation approach that is used to compute the $\mathcal{H}_\infty$-norm of the irrational transfer functions that arise in delay systems.

We have shown in numerical examples that our method outperforms a previously developed algorithm for $\mathcal{H}_\infty$-optimization of delay systems with respect to both computational speed and $\mathcal{H}_\infty$-performance. After showing the effect of the new method on a class of systems that were obtained by introducing output delay to benchmark systems for classical $\mathcal{H}_\infty$-optimization, we have demonstrated how a more general class of delay systems can be handled with our method.

Another benefit from using interpolation is that no structural constrains need to be imposed on the considered transfer function. In this way it is easy to integrate performance requirements as additional weight functions, even in the delay case.

We have established an equivalence of the resulting closed loop DDAEs to certain DDEs in order to use a robust algorithm to compute the rightmost eigenvalues to check if the resulting closed loop system is asymptotically stable. However, this transformation can be avoided when an efficient and robust algorithm for the computation of the rightmost eigenvalues of the strangeness free DDAEs is available. This leaves room for future research.

One drawback of the interpolation approach for the $\mathcal{H}_\infty$-norm computation compared with level-set methods is that it often does not converge when transfer functions with infinitely many peaks of similar magnitude are investigated. In order to extend the developed method such that it can also determine optimal controllers for systems with such transfer functions this issue must be addressed. A first approach is to combine the interpolation approach with the idea in [23] for the computation of the strong $\mathcal{H}_\infty$-norm of the asymptotic transfer function. For low frequencies interpolation can be used and the high frequency behavior may be estimated using the strong $\mathcal{H}_\infty$-norm of the asymptotic transfer function that is obtained using subspace projection.

Bibliography

[1] A. J. Mayo and A. C. Antoulas. A framework for the solution of the generalized realization problem. *Linear Algebra and its Applications*, 425(4):634–662, 2007.

[2] A. Antoulas. *Approximation of Large-Scale Dynamical Systems*, volume 6 of *Advances in Design and Control*. SIAM Publications, Philadelphia, PA, 2005.

[3] P. Benner, R. Byers, V. Mehrmann, and H. Xu. A robust numerical method for the γ-iteration in $\mathcal{H}_\infty$ control. *Linear Algebra and its Applications*, 425(2):548–570, 2007.

[4] P. Benner, M. Ohlberger, A. Cohen, and K. Willcox, editors. *Model Reduction and Approximation: Theory and Algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2017.

[5] P. Benner, V. Sima, and M. Voigt. $\mathcal{L}_\infty$-norm computation for continuous-time descriptor systems using structured matrix pencils. *IEEE Transactions on Automatic Control*, 57(1):233–238, 2012.

[6] A. Binder, V. Mehrmann, A. Miedlar, and P. Schulze. A Matlab toolbox for the regularization of descriptor systems arising from generalized realization procedures. Preprint 24-2015, Institut für Mathematik, TU Berlin, Germany, 2016. Available at `http://www3.math.tu-berlin.de/preprints/files/BinMMS16_ppt.pdf`.

[7] J. F. Bonnans, J. C. Gilbert, C. Lemerechal, and C. A. Sagastizábal. *Numerical Optimization: Theoretical and Practical Aspects*. Springer-Verlag, Berlin, New York, second edition, 2006.

[8] S. Boyd and V. Balakrishnan. A regularity result for the singular values of a transfer matrix and a quadratically convergent algorithm for computing its $\mathcal{L}_\infty$-norm. *Systems & Control Letters*, 15(1):1–7, 1990.

[9] D. Brethé and J. J. Loiseau. An effective algorithm for finite spectrum assignment of single-input systems with delays. *Mathematics and Computers in Simulation*, 45(3):339–348, 1998.

[10] N. A. Bruinsma and M. Steinbuch. A fast algorithm to compute the $\mathcal{H}_\infty$-norm of a transfer function matrix. *Systems & Control Letters*, 14(4):287–293, 1990.

[11] J. V. Burke, D. Henrion, A. S. Lewis, and M. L. Overton. HIFOO - a MATLAB package for fixed-order controller design and $\mathcal{H}_\infty$ optimization. In *Proceedings of the 5th IFAC Symposium on Robust Control Design*, pages 339–344, Toulouse, 2006.

[12] R. Byers. A bisection method for measuring the distance of a stable matrix to the unstable matrices. *SIAM Journal on Scientific and Statistical Computing*, 9(5):875–881, 1988.

[13] F. E. Curtis, T. Mitchell, and M. L. Overton. A BFGS-SQP method for nonsmooth, nonconvex, constrained optimization and its evaluation using relative minimization profiles. *Optimization Methods and Software*, 32(1):148–181, 2017.

[14] J. E. Dennis and R. B. Schnabel. A new derivation of symmetric positive definite secant updates. In O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, editors, *Nonlinear Programming 4*, pages 167–199. Academic Press, 1981.

[15] J. C. Doyle, K. Glover, P. P. Khargonekar, and B. A. Francis. State-space solutions to standard $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control problems. *IEEE Transactions on Automatic Control*, 34(8):831–847, 1989.

[16] N. H. Du, V. H. Linh, V. Mehrmann, and D. D. Thuan. Stability and robust stability of linear time-invariant delay differential-algebraic equations. *SIAM Journal on Matrix Analysis and Applications*, 34(4):1631–1654, 2013.

[17] K. Engelborghs and D. Roose. Numerical computation of stability and detection of Hopf bifurcations of steady state solutions of delay differential equations. *Advances in Computational Mathematics*, 10(3):271–289, 1999.

[18] J. L. Figeuroa, A. C. Desages, J. A. Romagnoli, and A. Palazoglu. Controller order reduction for process control systems. *Computers & Chemical Engineering*, 16(6):593–603, 1992.

[19] F. R. Gantmacher. *The Theory of Matrices*, volume 2. AMS Chelsea Publishing, Providence, RI, USA, 1959.

[20] Josep M Guerrero, Mukul Chandorkar, Tzung-Lin Lee, and Poh Chiang Loh. Advanced control architectures for intelligent microgrids—part i: Decentralized and hierarchical control. *IEEE Transactions on Industrial Electronics*, 60(4):1254–1262, 2013.

[21] S. Gumussoy, D. Henrion, M. Millstone, and M. L. Overton. Multiobjective robust control with HIFOO 2.0. In *Proceedings of the 6th IFAC Symposium on Robust Control Design*, pages 144–149, Haifa, 2009.

[22] S. Gumussoy and W. Michiels. Fixed-order $\mathcal{H}_\infty$ optimization of time-delay systems. In M. Diehl, F. Glineur, E. Jarlebring, and W. Michiels, editors, *Recent Advances in Optimization and its Applications in Engineering: The 14th*

Belgian-French-German Conference on Optimization, pages 103–112. Springer-Verlag, Berlin, Heidelberg, 2010.

[23] S. Gumussoy and W. Michiels. Fixed-order $\mathcal{H}_\infty$ control for interconnected systems using delay differential algebraic equations. *SIAM Journal on Control and Optimization*, 49(5):2212–2238, 2011.

[24] S. Gumussoy and M. L. Overton. Fixed-order $\mathcal{H}_\infty$ controller design via HIFOO, a specialized nonsmooth optimization package. In *Proceedings of 2008 American Control Conference*, pages 2750–2754, Seattle, 2008.

[25] J. K. Hale and S. M. V. Lunel. *Introduction to Functional Differential Equations*. Springer-Verlag, New York, 1993.

[26] P. Kunkel and V. Mehrmann. *Differential-Algebraic Equations: Analysis and Numerical Solution*. EMS Textbooks in Mathematics. European Mathematical Society, Zürich, 2006.

[27] F. Leibfritz. Compleib: Constraint Matrix-optimization Problem library - a collection of test examples for nonlinear simidefinite programs, control system design and related problems, 2004. Available at `http://www.friedemann-leibfritz.de/COMPlib_data/COMPlib_Main_Paper.pdf`.

[28] A. S. Lewis and M. L. Overton. Nonsmooth optimization via quasi-Newton methods. *Mathematical Programming*, 141(1):135–163, 2013.

[29] M. Lichtner, M. Wolfrum, and S. Yanchuk. The spectrum of delay differential equations with large delay. *SIAM Journal on Mathematical Analysis*, 43(2):788–802, 2011.

[30] P. Losse, V. Mehrmann, L. K. Poppe, and T. Reis. The modified optimal $\mathcal{H}_\infty$ control problem for descriptor systems. *SIAM Journal on Control and Optimization*, 47(6):2795–2811, 2009.

[31] A. Manitius and A. Olbrot. Finite spectrum assignment problem for systems with delays. *IEEE Transactions on Automatic Control*, 24(4):541–552, 1979.

[32] E. Mengi, E. A. Yildirim, and M. Kiliç. Numerical optimization of eigenvalues of hermitian matrix functions. *SIAM Journal on Matrix Analysis and Applications*, 35(2):699–724, 2014.

[33] W. Michiels, K. Engelborghs, P. Vansevenant, and D. Roose. Continuous pole placement for delay equations. *Automatica*, 38(5):747–761, 2002.

[34] W. Michiels and S. Gumussoy. Characterization and computation of $\mathcal{H}_\infty$ norms for time-delay systems. *SIAM Journal on Matrix Analysis and Applications*, 31(4):2093–2115, 2010.

[35] N. Aliyev, P. Benner, E. Mengi, P. Schwerdtner, and M. Voigt. Large-scale computation of $\mathcal{L}_\infty$-norms by a greedy subspace method. *SIAM Journal on Matrix Analysis and Applications*, 34(4):1496–1516, 2017.

[36] M. Plancherel and M. Leffler. Contribution à l'étude de la représentation d'une fonction arbitraire par des intégrales définies. *Rendiconti del Circolo Matematico di Palermo*, 30(1):289–335, 1910.

[37] D. Roose, T. Luzyanina, K. Engelborghs, and W. Michiels. Software for stability and bifurcation analysis of delay differential equations and applications to stabilization. In S. I. Niculescu and K. Gu, editors, *Advances in Time-Delay Systems*, pages 167–181. Springer-Verlag, Berlin, Heidelberg, 2004.

[38] P. Schwerdtner and M. Voigt. Computation of the $\mathcal{L}_\infty$-norm using rational interpolation. 2017. Submitted for Publication, Available at `http://www.math.tu-berlin.de/fileadmin/i26/Bilder_Webseite/AG_ModNumDiff/FG_NumMath/mvoigt/SchV17.pdf`.

[39] S. Skogestad and I. Postlethwaite. *Multivariable Feedback Control: Analysis and Design*. Wiley, Chichester, second edition, 2009.

[40] H. Smith. *An Introduction to Delay Differential Equations with Applications to the Life Sciences*, volume 57 of *Texts in Applied Mathematics*. Springer-Verlag, New York, 2011.

[41] J. Vanbiervliet, K. Verheyden, W. Michiels, and S. Vandewalle. A nonsmooth optimisation approach for the stabilisation of time-delay systems. *ESAIM: Control, Optimisation and Calculus of Variations*, 14(3):478–493, 2008.

[42] P. Wolfe. Convergence conditions for ascent methods. *SIAM Review*, 11(2):226–235, 1969.

[43] Yongfang Xie, Weihua Gui, and Zhaohui Jiang. Decentralized robust h-infinity descriptor output feedback control for value-bounded uncertain descriptor large-scale systems. *Journal of Control Theory and Applications*, 4(2):193–200, 2006.

[44] G. Zames. Feedback and optimal sensitivity: Model reference transformations, multiplicative seminorms, and approximate inverses. *IEEE Transactions on Automatic Control*, 26(2):301–320, 1981.

[45] G. Zames and D. Bensoussan. Multivariable feedback, sensitivity, and decentralized control. *IEEE Transactions on Automatic Control*, 28(11):1030–1035, 1983.

[46] G. Zames and B. Francis. Feedback, minimax sensitivity, and optimal robustness. *IEEE Transactions on Automatic Control*, 28(5):585–601, 1983.

[47] Q. C. Zhong. *Robust Control of Time-Delay Systems*. Springer-Verlag, London, first edition, 2006.

[48] K. Zhou, J. C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice Hall, Upper Saddle River, NJ, 1996.

[49] K. Zhou, C. D'Souza, and J. R. Cloutier. Structurally balanced controller order reduction with guaranteed closed loop performance. *Systems & Control Letters*, 24(4):235–242, 1995.