High Performance Robust Control, System Identification, and Model Validation: A Literature Study

F.B.J.W.M. Hendriks

DCT 2008.143
# Contents

## 1 Introduction
- 1.1 Background ........................................... 1
- 1.2 Problem Formulation ................................. 1
- 1.3 Outline of the Report .............................. 2

## 2 Norms
- 2.1 Signals .................................................. 3
- 2.2 Systems ................................................... 6

## 3 Control
- 3.1 LQ and LQG Control ................................. 9
  - 3.1.1 LQ control ........................................... 9
  - 3.1.2 LQG control ....................................... 10
- 3.2 $H_2$ and $H_\infty$ Control ....................... 11
  - 3.2.1 General control problem formulation .......... 11
  - 3.2.2 $H_2$ optimal controller computation ........ 13
  - 3.2.3 $H_\infty$ optimal controller computation ....... 14
- 3.3 Concluding Remarks ............................... 16

## 4 Robust Control
- 4.1 Dynamic Weighting Filters ................. 19
  - 4.1.1 Signal-Based $H_\infty$ Control ............... 19
  - 4.1.2 $H_\infty$ Loop-shaping Design ............... 20
- 4.2 Model Uncertainty ............................... 22
- 4.3 Robust Stability .................................... 24
- 4.4 Robust Performance ............................. 26

## 5 Identification
- 5.1 General Considerations ....................... 31
- 5.2 Time Domain Identification ............... 32
- 5.3 Time Domain Validation ..................... 35
- 5.4 Frequency Domain Input Generation ........ 38
- 5.5 Frequency Domain Identification ......... 40
- 5.6 Frequency Domain Validation ............. 41
6 Conclusions and Recommendations 43

Bibliography 45
Chapter 1

Introduction

1.1 Background

In the modern high tech industry, speed and accuracy are very important. A good example of a machine used in the high tech industry is a wafer scanner. A wafer scanner is used for the production of integrated circuits on a silicon wafer by a photolithographic process. An important component of a wafer scanner is the wafer stage. The wafer stage has to position the silicon wafer with an accuracy in the order of nanometers and microradians.

To keep up with the competition, faster accelerations and higher accuracy is desirable. To achieve faster accelerations, the weight of wafer stages has to be reduced. As a side-effect, the wafer stages will become more flexible. Flexible modes can only be observed when considering MIMO systems. To accurately control MIMO systems, and suppress the flexible modes, MIMO control design is required.

The focus of the controllers considered in this report will be on controllers that can naturally handle MIMO systems. These controllers are calculated as a mathematical optimization problem and can naturally cope with coupling among the different inputs and outputs of the plant. Classical examples of MIMO controllers are LQ and LQG regulators. However, these controllers lack guaranteed robustness [7]. Controller design methods for MIMO systems that enable robustness against modeling errors are $H_{\infty}$-optimization and $\mu$-synthesis.

The controllers considered in this report are model based controllers, thus a model of the system has to be identified. The goal of time domain identification as discussed in [10] is to capture all dynamics of a system in the model, to create a full order model. For control purposes, this is not needed [18]. A low order model which is accurate around the bandwidth is sufficient for control. For robust control, not only the model, but also the quality of the model should be known.

1.2 Problem Formulation

The goal of the research project is formulated as:

*Obtain an accurate MIMO model of the given test setup, suitable for robust control, such that*
a high-performance robust MIMO controller can suppress flexible modes of the test setup.

The goal of this literature study is to answer some questions that arise when trying to achieve the goal mentioned above. The following questions arise:

1. What MIMO controllers are available?
2. How is uncertainty accounted for in robust control?
3. How can the performance of the controllers be quantified?
4. How can an accurate model, suitable for robust control, be obtained?

1.3 Outline of the Report

The report is organized as follows. In Chapter 2 some general notions and information on norms on signals and systems is discussed. Norms are used to quantify the performance of a controller and are very important for understanding the mathematical basics of the controllers discussed in Chapter 3. First, the classical LQ and LQG controllers will be discussed. Then a general control problem formulation is introduced. Chapter 3 will be concluded with the introduction of the more general $\mathcal{H}_2$ and $\mathcal{H}_\infty$ controllers. In Chapter 4, model uncertainty is introduced. It is also discussed how to deal with uncertainty and how to design a robust $\mathcal{H}_\infty$ controller. Finally, in Chapter 5 identification and validation in the time and frequency domain is discussed. The questions of how to identify a system for control and how to quantify the uncertainty is dealt with.
Chapter 2

Norms

2.1 Signals

A signal is a function that quantifies how a certain variable evolves in time:

\[ s : T \rightarrow W, \]

with \( s \) a signal, \( T \) the time set and \( W \) a signal space. In general \( T \subseteq \mathbb{R} \) or \( T \subseteq \mathbb{Z} \). If \( T \subseteq \mathbb{R} \), then the signal is a continuous time signal. If \( T \subseteq \mathbb{Z} \), then the signal is a discrete time signal because it evolves in discrete time. A signal \( s \) can represent multiple quantities at each time instant \( t \in T \). At each time instant \( t \in T \), signal \( s \) represents a vector

\[
s(t) = \begin{pmatrix}
s_1(t) \\
s_2(t) \\
\vdots \\
s_q(t)
\end{pmatrix}.
\]

The signal space \( W \) is in a general form a multidimensional complex-valued space. For physical system identification and control, \( W \) is only considered as a multidimensional real-valued space i.e., \( W = \mathbb{R}^q \).

To avoid proving a theorem for every signal, signals can be grouped together into classes of signals. An example of such a class are periodic signals. A signal \( s \) is periodic with period \( P \) (or \( P \)-periodic) if

\[ s(t) = s(t + P), \quad t \in T. \]

An example of periodic signals are sinusoidal signals. If a signal is not periodic for any \( P \), then the signal is a-periodic.

Another way to group the 'same' kind of signals is by using norms. A norm is a number that represents the size of a signal. A norm should satisfy the following properties ([19],[16] and [6]):

**Definition 2.1.** A norm on \( s \), denoted \( \|s\| \), is a real-valued function with the following properties:
I) \( \|s\| \geq 0; \) (positivity)

II) \( \|s\| = 0 \iff s = 0; \) (positive definiteness)

III) \( \|\alpha s\| = |\alpha|\|s\|, \alpha \in \mathbb{C}; \) (homogeneity)

IV) \( \|s_1 + s_2\| \leq \|s_1\| + \|s_2\|; \) (triangle inequality)

There are many ways to define a norm on a signal. A general form of a norm, the \( p \)-norm, is defined as follows:

**Definition 2.2. \( p \)-norm**

For continuous time signals, the \( p \)-norm is defined by:

\[
\|s\|_p = \left( \int_{t \in T} |s(t)|^p dt \right)^{1/p}.
\]

For discrete time signals, the \( p \)-norm is defined by:

\[
\|s\|_p = \left( \sum_{t \in T} |s(t)|^p \right)^{1/p}.
\]

The most commonly used norms in system identification and control are the 1-norm, the 2-norm and the \( \infty \)-norm. The 1-norm represents the area in a graphical representation of a signal (see Figure 2.1), the 2-norm represents the energy of a signal, which is the area of the square of the signal, and the \( \infty \)-norm gives the peak value (see Figure 2.1). In continuous time these norms are defined as follows:

**Definition 2.3. norms (continuous)**

\[
\|s\|_1 = \int_{t \in T} |s(t)| dt
\]

\[
\|s\|_2 = \left( \int_{t \in T} |s(t)|^2 dt \right)^{1/2}
\]

\[
\|s\|_\infty = \max_{t \in T} |s_i(t)|
\]

In discrete time the 1-norm, 2-norm and \( \infty \)-norm are defined as:
Definition 2.4. norms (discrete)

\[ \|s\|_1 = \sum_{t \in T} |s(t)| \]
\[ \|s\|_2 = (\sum_{t \in T} |s(t)|^2)^{1/2} \]
\[ \|s\|_{\infty} = \max_{i} \sup_{t \in T} |s_i(t)| \]

The 2-norm represents the energy of a signal, so a sinusoidal signal has an infinite 2-norm. In order to be able to work with this important class of signals, another norm is commonly used. This norm represents the average power of a signal and is called the power-norm. The power-norm is defined as:

Definition 2.5. power norm

For continuous time signals, the power norm is defined as:

\[ \text{pow}(s) = \left( \lim_{T \to \infty} \frac{1}{2T} \int_{-T}^{T} |s(t)|^2 \, dt \right)^{1/2}. \]

For discrete time signals, the power norm is defined as:

\[ \text{pow}(s) = \left( \lim_{T \to \infty} \frac{1}{2T} \sum_{t=-T}^{T} |s(t)|^2 \right)^{1/2}. \]

This norm is a pseudo norm, because it only satisfies I, III and IV, and not II from Definition 2.1. Take for example a signal which is 1 if \( 0 \leq t < 1 \) and 0 elsewhere. The power norm of this signal tends to 0 for \( T \to \infty \).

A way to group signals together with norms, is to define signal sets for which a certain norm is finite for all signals in that signal set. Definitions of some commonly used signals sets are ([19],[11] and [5]):

Definition 2.6. signal sets (continuous)

\[ \mathcal{L}_1(T) = \{ s : T \to W \mid \|s\|_1 < \infty \} \]
\[ \mathcal{L}_2(T) = \{ s : T \to W \mid \|s\|_2 < \infty \} \]
\[ \mathcal{L}_{\infty}(T) = \{ s : T \to W \mid \|s\|_{\infty} < \infty \} \]
\[ \mathcal{P}(T) = \{ s : T \to W \mid \text{pow}(s) < \infty \} \]

Definition 2.7. signal sets (discrete)

\[ \ell_1(T) = \{ s : T \to W \mid \|s\|_1 < \infty \} \]
\[ \ell_2(T) = \{ s : T \to W \mid \|s\|_2 < \infty \} \]
\[ \ell_{\infty}(T) = \{ s : T \to W \mid \|s\|_{\infty} < \infty \} \]
\[ \ell_{\text{pow}}(T) = \{ s : T \to W \mid \text{pow}(s) < \infty \} \]

In general these signal sets have some signals in common. But also some that are unique to the signal set. All signals in \( \mathcal{L}_2 \), also belong to \( \mathcal{P} \). A graphical interpretation of the different signal sets is given in Figure 2.2. \( \mathcal{L}_2 \) consists of all square integrable and Lebesgue measurable functions [19].
2.2 Systems

In a mathematical sense, a system is just a set of signals. In identification and control it is common to split the signals into input and output signals. Then the system is the relation between the input and the output signals,

\[ H : \mathcal{U} \rightarrow \mathcal{Y}. \]

The function, or operator, \( H \) maps the input \( u \), in signal space \( \mathcal{U} \), to output \( y \), in signal space \( \mathcal{Y} \). For linear time invariant systems, this mapping can be described by a convolution in continuous time,

\[ y(t) = (h * u)(t) = \int_{-\infty}^{\infty} h(t - \tau)u(\tau)d\tau. \]

The convolution kernel \( h(t) \) is, in system theoretic language, usually referred to as impulse response, because when the input is a Dirac impulse, \( u(t) = \delta(t) \), the output is equal to the convolution kernel, \( y(t) = h(t) \). \( H \) defines a linear map: \( H(\alpha_1 u_1 + \alpha_2 u_2) = \alpha_1 H(u_1) + \alpha_2 H(u_2) \). \( H \) is also time-invariant, it maps the shifted input \( u(t - t_0) \) to the shifted output \( y(t - t_0) \) \( \forall t_0 \).

An interesting property of a system \( H \) is its gain. This is determined by a norm on the system. A system \( H \) is in general represented by a matrix. An important observation is that a norm in the sense of Definition 2.1 on a matrix is not the same as a matrix norm. A norm on a matrix is a matrix norm if it satisfies, in addition to the properties given in definition 2.1, the multiplicative property ([16]):

**Definition 2.8.** The multiplicative property is defined as:

\[ \|AB\| \leq \|A\| \cdot \|B\| \]

The induced matrix norm is very important. Suppose we have the system depicted in Figure 2.3, with \( H \) the system, \( d \) a disturbance signal and \( e \) an error signal. Then the following relation exists: \( e = Hd \). The maximum possible gain of the system is then given by:

\[ \|H\|_{(p,q)} = \max_{d \neq 0} \frac{\|Hd\|_p}{\|d\|_q} \]
Figure 2.3: Graphical representation of a simple system.

Table 2.1: Overview of induced norms.

<table>
<thead>
<tr>
<th></th>
<th>$|u|_2$</th>
<th>$|u|_\infty$</th>
<th>$\text{pow}(u)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$|y|_2$</td>
<td>$|H(s)|_\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
</tr>
<tr>
<td>$|y|_\infty$</td>
<td>$|H(s)|_2$</td>
<td>$|h(t)|_1$</td>
<td>$\infty$</td>
</tr>
<tr>
<td>$\text{pow}(y)$</td>
<td>0</td>
<td>$\leq |H(s)|_\infty$</td>
<td>$|H(s)|_\infty$</td>
</tr>
</tbody>
</table>

Such a norm on a matrix is called an induced norm. In control engineering, linear system theory is very important. This means that the signal $e$ scales linearly with signal $d$ and that the following expression is equal to the one above.

$$\|H\|_{(p,q)} = \max_{\|d\|_q=1} \|Hd\|_p$$

All induced system norms $\|H\|_{(p,q)}$ with $p = q$ are matrix norms, i.e. they have the multiplicative property given in Definition 2.8. An overview of some induced system norms is given in table 2.1 ([6] and [16]).

Norms can have different interpretations. The 2-norm of a system can be interpreted as the gain of a system from a white noise input to an output measured in the 2-norm. Another interpretation can be given to the 2-norm of a system, consider the following equation:

$$\|H(s)\|_2 = \|h(t)\|_2 = \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} \sum_{i} \sigma^2_i(H(j\omega))d\omega\right)^{\frac{1}{2}} \tag{2.1}$$

Here the 2-norm of a system is given as the root of the singular values of the system squared, integrated over all frequencies. This can be interpreted as an average gain in all directions over all frequencies.

Looking at:

$$\|H(s)\|_\infty = \max_{\omega} \sigma(H(j\omega)) = \max_{\|d(t)\|_2} \|e(t)\|_2 \tag{2.2}$$

we see that the $\infty$-norm is the maximum singular value over all frequencies. This is the "worst-case" amplification. At all other frequencies in all other directions, the gain is less. For a SISO system, the $\infty$-norm is the same as the maximum magnitude in a Bode diagram.
Chapter 3

Control

3.1 LQ and LQG Control

3.1.1 LQ control

The L and Q of LQ control refer to Linear systems with a Quadratic performance criterion ([19],[1]). Consider the following LTI state space representation of a continuous time system:

\[
\begin{align*}
\dot{x}(t) & = Ax(t) + Bu(t) \\
y(t) & = Cx(t).
\end{align*}
\]

The control goal is to reduce a non-zero initial state \(x(0)\) to zero as fast as possible. During the control action, the input \(u\) should preferably not be too large. To achieve this, the following performance index is used:

\[
J = \int_0^\infty \left[ y^T(t)Qy(t) + u^T(t)Ru(t) \right] dt.
\]

\(J\) is a quadratic performance criterion in the input \(u\) and the output \(y\). \(Q\) and \(R\) are symmetric weighting matrices; \(Q = Q^T\), \(R = R^T\). Often, \(Q\) and \(R\) are taken as diagonal matrices. \(Q\) is the weight on the accumulated output \(y\) and \(R\) is the weight on the accumulated input \(u\). A large \(Q\) implies a good tracking behavior. A large \(R\) results in low input energy, and therefore probably in a bad tracking behavior. The size of \(Q\) and \(R\) is a trade-off because only relative values are relevant, since the absolute values do not change the minimizer. It is quite straightforward that \(Q\) and \(R\) should be chosen such that \(y^T(t)Qy(t)\) and \(u^T(t)Ru(t)\) are at least nonnegative, and in most cases positive definite, i.e. \(Q\) and \(R\) should be nonnegative-definite and positive-definite, respectively. The problem of controlling a system such that \(J\) is minimal along all possible trajectories is called the optimal linear regulator problem. The solution to this minimization problem is given by a feed-back law:

\[
u(t) = -Fy(t)\]

\(F\) is a constant matrix, given by: \(F = R^{-1}B^TX\). \(X\) is symmetric and is the nonnegative-definite solution of the algebraic matrix Riccati equation (ARE):

\[
A^TX + XA + Q + XB R^{-1} B^TX = 0.
\]
Some assumptions about the system and the performance index should be kept in mind to let the LQ controller work well. The system should be stabilizable and detectable. If the system is not stabilizable, the system cannot be stabilized by a controller. If the system is not detectable, there exist controllers that do not stabilize the system, but produce stable outputs. The unstable states are hidden from the output. Thus overall stability of the system is not guaranteed.

\( R \) should be positive-definite to prevent the input from approaching infinite. \( Q \) should also be positive-definite. If \( Q \) would not be positive-definite, unstable closed-loop modes could exist, without having any influence on the performance index \( J \).

More information about LQ control can be found in [1] and [19].

### 3.1.2 LQG control

LQG control is a lot like LQ control. The L, Q and G of LQG control refer to Linear systems with Gaussian noise and a Quadratic performance criterion. Consider the following LTI state space representation of a system:

\[
\dot{x}(t) = Ax(t) + Bu(t) + w_d \\
y(t) = Cx(t) + w_n.
\]

This system is the same as the system in Equation (3.1), but then with noise added. \( w_d \) and \( w_n \) are the disturbance (process noise) and the measurement noise respectively. These are assumed to be uncorrelated zero-mean Gaussian stochastic processes. Due to the added noise, the output \( y(t) \) and the states \( x(t) \) have become a stochastic process. As a result, also the quadratic performance criterion has become a stochastic process:

\[
J = \mathbb{E}\left\{ \lim_{T \to \infty} \frac{1}{T} \int_0^T \left[ y^T(t)Qy(t) + u^T(t)Ru(t) \right] dt \right\}.
\]

The minimization of this performance criterion is called the stochastic linear regulator problem. Because of the noise, the output will only become 0 with a very low probability. If the integral in Equation (3.4) would be taken from 0 to infinity, the integral itself would tend to infinity. The matrices \( Q \) and \( R \) should have the same properties as with the LQ controller.

To achieve an optimal controller for the system given in Equation (3.3) the problem can be split in to two separate parts, this is called the separation theorem.

First, a controller is designed for the deterministic part of the system. The system in Equation (3.3) then reduces to the system described in Equation (3.1). It is clear that the optimal controller for this system is the feedback law \( u(t) = -Fx(t) \), with \( F = R^{-1}B^TX \) and \( X \) the unique nonnegative-definite solution of the ARE given in subsection 3.1.1.

The second part of the problem consists of reconstructing the states of the system. The optimal estimation problem is actual dual to the optimal control problem. The reconstruction of the states is done with a Kalman-Bucy filter, see figure 3.1:

\[
\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + Ky(t) - C\hat{x}(t).
\]

This filter gives an optimal estimate \( \hat{x} \) of \( x \), such that \( E(e(t)^Te(t)) \) is minimized, with \( e(t) = x(t) - \hat{x}(t) \). The gain matrix \( K = YC^TV^{-1} \) minimizes the mean square reconstruction error, with \( V \) the power spectral density matrix of the white measurement noise \( w_n \) and \( Y \) the unique nonnegative-definite solution of the following ARE:

\[
YA^T + AY + W - YC^TV^{-1}CY = 0.
\]
Combining Equations (3.3), (3.5), the feedback law $u(t) = -Fx(t)$, and the following coordinate transformation
\[
\begin{bmatrix}
  x(t) \\
  e(t)
\end{bmatrix} = \begin{bmatrix} I & 0 \\ I & -I \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix},
\]
the system can also be written as:
\[
\begin{bmatrix}
  \dot{x}(t) \\
  \dot{e}(t)
\end{bmatrix} = \begin{bmatrix} A - BF & BF \\ 0 & A - KC \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix} + \begin{bmatrix} I & 0 \\ I & -K \end{bmatrix} \begin{bmatrix} w_d \\ wn \end{bmatrix}
\]
(3.7)

The eigenvalues of this system matrix consist of the eigenvalues of $(A - BF)$ and $(A - KC)$. So the controller and the observer can be designed separately, because they do not influence each others stability.

The design method for LQ(G) controllers may look very appealing, but it has no guaranteed robustness margins. More information on LQG control can be found in [1], [11], [16] and [19].

### 3.2 $\mathcal{H}_2$ and $\mathcal{H}_\infty$ Control

Although $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control have quite a different interpretation, they are very close related.

The general problem formulation prompted here is the same for both. Also the solution process is very much alike. First the general control problem formulation will be given. Then $\mathcal{H}_2$ and $\mathcal{H}_\infty$ will be briefly discussed and interpreted.

#### 3.2.1 General control problem formulation

To be able to generate a controller for every $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control problem, it is useful to have a general problem formulation. Such a general problem formulation is shown in Figure 3.2. The system of Figure 3.2 is described by
The generalized plant $P$ is given by

$$
P = \begin{bmatrix}
A & B_1 & B_2 \\
C_1 & D_{11} & D_{12} \\
C_2 & D_{21} & D_{22}
\end{bmatrix}
$$  

(3.10)

In this general control problem we also have the control variables $u$, the measured variables $v$, the exogenous signals $w$ like disturbances $w_d$ and commands $r$ and the ”error” signals $z$, which are to be minimized in some sense. The closed-loop transfer function from $w$ to $z$ is given by

$$
z = (P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21})w = F_l(P, K)w
$$  

(3.11)

Minimizing $F_l(P, K)$ in the sense of $\mathcal{H}_2$ or $\mathcal{H}_\infty$ norms is called $\mathcal{H}_2$ or $\mathcal{H}_\infty$ optimal control respectively. Both control strategies have some assumptions that are typically made ([16],[19]):

- **A.1)** $(A, B_2, C_2)$ is stabilizable and detectable.
- **A.2)** $D_{12}$ and $D_{21}$ have full rank.
- **A.3)** \[
\begin{bmatrix}
A - j\omega I & B_2 \\
C_1 & D_{12}
\end{bmatrix}
\]
  has full column rank for all $\omega$.
- **A.4)** \[
\begin{bmatrix}
A - j\omega I & B_1 \\
C_2 & D_{21}
\end{bmatrix}
\]
  has full row rank for all $\omega$.
- **A.5)** $D_{11} = 0$ and $D_{22} = 0$.

Assumption 1 is needed to guarantee the existence of a stabilizing controller $K$. Assumption 2 is sufficient to guarantee the controller is proper and thus realizable. To prevent cancelation of poles and zeros on the imaginary axis by the controller, assumptions 3 and 4 are made. $\mathcal{H}_2$ is the set of strictly proper stable transfer functions. In assumption 5, $D_{11} = 0$, guarantees
the system is strictly proper. $D_{22} = 0$ only simplifies the formulae to find the optimal $\mathcal{H}_2$ controller. This assumption can be made without loss of generality. The controller for $D_{22} \neq 0$ is given by $K_D = K(I + D_{22}K)^{-1}$ with $K$ the controller for the system with $D_{22} = 0$. For $\mathcal{H}_\infty$ control assumption 5 is not necessary, but it simplifies the computations a lot. If $D_{11}$ and $D_{22}$ are not zero, an equivalent $\mathcal{H}_\infty$ control problem can be formulated in which they are.

### 3.2.2 $\mathcal{H}_2$ optimal controller computation

The goal of $\mathcal{H}_2$ optimal control is to minimize the closed-loop transfer function from $w$ to $z$. By Parseval’s theorem, we know that $\|H\|_2 = \|h\|_2$, with $h(t)$ the impulse response of the system. So we can write the 2-norm of the closed-loop transfer function from $w$ to $z$ as follows:

$$\|F_l(s)\|_2 = \|H(s)\|_2 = \|h(t)\|_2 = \sqrt{\int_{-\infty}^{\infty} \text{Tr}(h^*(t)h(t))dt}$$

(3.12)

The impulse response is given by

$$h(t) = \begin{cases} 0 & t < 0 \\ Ce^{At}B & t \geq 0. \end{cases}$$

(3.13)

So the squared 2-norm can be written as

$$\|H\|_2^2 = \int_0^\infty \text{Tr}(B^Te^{AT}C^TCe^{At}B)dt = \text{Tr}(B^TL_oB),$$

(3.14)

with

$$L_o = \int_0^\infty \text{Tr}(e^{AT}C^TCe^{At})dt.$$  

(3.15)

$L_o$ being the observability gramian. It can be shown that the observability gramian can be computed from the following Lyapunov equation if $A$ is stable [6]:

$$L_oA + A^TL_o + C^TC = 0.$$  

(3.16)

This algebraic relation is computationally more attractive than the expression in (3.15). A dual approach computing the $\mathcal{H}_2$ norm involves the computation of the controllability gramian, $L_c = \int_0^\infty \text{Tr}(e^{AT}BB^Te^{At})dt$, which satisfies the following Lyapunov equation, $L_cA^T + AL_c + BB^T = 0$. Hence the $\mathcal{H}_2$ norm can be computed from

$$\|H\|_2 = \sqrt{\text{Tr}(B^TL_oB)} = \sqrt{\text{Tr}(CL_cC^T)}.$$  

(3.17)

Now it is clear how to compute the $\mathcal{H}_2$ norm, the question remains how to minimize it. We want to minimize the $\mathcal{H}_2$ norm from $w$ to $z$, which can be stated as the following $\mathcal{H}_2$ optimal control problem:
Synthesize a controller $K$ that internally stabilizes $P$ and minimizes $\| F_l(K) \|_2$.

Under the assumptions made in Section 3.2.1, the controller $K$ can be calculated by solving the following Riccati equations

$$
0 = A^T X_2 + X_2 A - X_2 B_2 B_2^T X_2 + C_1^T C_1 \tag{3.18}
$$

$$
0 = AY_2 + Y_2 A^T - Y_2 C_2^T C_2 Y_2 + B_1 B_1^T, \tag{3.19}
$$

with the associated Hamiltonian matrices

$$
H_2 := \begin{bmatrix} A & -B_2 B_2^T \\ -C_1^T C_1 & -A^T \end{bmatrix}, \quad J_2 := \begin{bmatrix} A^T & -C_2^T C_2 \\ -B_1 B_1^T & -A \end{bmatrix}. \tag{3.20}
$$

The unique optimal controller $K$ is then given by

$$
K := \begin{bmatrix} A + B_2 F_2 + L_2 C_2 \\ F_2 \end{bmatrix} \begin{bmatrix} -L_2 \\ 0 \end{bmatrix}, \tag{3.21}
$$

where $F_2 = -B_2^T X_2$ and $L_2 = -Y_2 C_2^T$, and $X_2$ and $Y_2$ are the solutions of the Riccati equations presented in Equations (3.18) and (3.19) respectively.

It is interesting to see that the $H_2$ optimal controller can be split into a feedback control part and a state estimation part, it is derived by using the separation principle. This can easily be seen by writing the controller as:

$$
\dot{x}_k = Ax_k + B_2 u + L_2 (C_2 x_k - y) \quad (3.22)
$$

$$
u = F_2 x_k, \quad (3.23)
$$

where $x_k$ denotes the states of the controller. This is similar to the LQG-controller, in fact the LQG-controller is a special case of the $H_2$ optimal controller, see [16] for more information.

### 3.2.3 $H_{\infty}$ optimal controller computation

The computation of the $H_{\infty}$ norm is more involved than the computation of the $H_2$ norm. When looking at Equation (2.2), we see that $\| H \|_{\infty} = \bar{\sigma}(H)(j\omega)$. One could propose to perform singular value decompositions for a lot of frequencies to find the maximum singular value. But this method is not efficient and has no known accuracy, because the maximum singular value is not known for all frequencies. In [2], a method is proposed to compute the $H_{\infty}$ norm using state-space models. This method involves a test whether $\| H \|_{\infty} < \gamma$. By means of a bisection search, and lower and upper bounds, $\underline{\gamma}$ and $\bar{\gamma}$ respectively, it is guaranteed that $\underline{\gamma} < \| H \|_{\infty} < \bar{\gamma}$. This method is considered next.

$$
\| H \|_{\infty} < \gamma \quad \text{and} \quad \| \frac{1}{\gamma} H \|_{\infty} < 1 \quad \text{are equivalent if and only if}
$$

$$
I - \frac{1}{\gamma^2} H(j\omega)^* H(j\omega) \tag{3.24}
$$

is invertible for all $\omega \in \mathbb{R}$. If we define

$$
M := I - \frac{1}{\gamma^2} H(s)^T H(s), \tag{3.25}
$$
then invertibility of (3.24) is equivalent to $M(s)^{-1}$ having no poles on the imaginary axis. A state-space realization of $M(s)$ is given by

$$M(s) = \begin{bmatrix} A & 0 & -\frac{1}{\gamma}B \\ -C^T C & -A^T & 0 \\ 0 & \frac{1}{\gamma}B^T & I \end{bmatrix}$$

and its inverse is then given by

$$M(s)^{-1} = \begin{bmatrix} A & \frac{1}{\gamma^2}BB^T & -\frac{1}{\gamma}B \\ -C^T C & -A^T & 0 \\ 0 & \frac{1}{\gamma}B^T & I \end{bmatrix}$$

When looking at the state-space realization of $M(s)^{-1}$ we can conclude that if and only if the matrix

$$H_M = \begin{bmatrix} A & \frac{1}{\gamma^2}BB^T \\ -C^T C & -A^T \end{bmatrix}$$

(3.26)

has no eigenvalues on the imaginary axis, $\|H\|_\infty < \gamma$. The $H_\infty$ norm can be computed arbitrarily accurate by means of a bisection search.

**Algorithm 3.1.** ($H_\infty$ Norm Computation [11]). **Input:** stopping criterion $\epsilon > 0$ and $\gamma$ and $\bar{\gamma}$ such that $\gamma < \|H\|_\infty < \gamma < \bar{\gamma}$.

**Step 1)** Set $\gamma = \frac{\gamma + \bar{\gamma}}{2}$.

**Step 2)** Compute the eigenvalues of $H_M$ (see Equation (3.26)).

**Step 3)** If $H_M$ has eigenvalues on the imaginary axis, set $\gamma := \gamma$, otherwise, set $\gamma := \gamma$.

**Step 4)** If $\bar{\gamma} - \gamma < \epsilon$, then stop.

**Output:** $\|H\|_\infty$ approximates $\frac{\gamma + \bar{\gamma}}{2}$, with an absolute accuracy $\epsilon$.

Similar to the computation of the $H_\infty$ norm, the computation of the $H_\infty$ optimal controller is more involved than the computation of the $H_2$ optimal controller. The $H_\infty$ optimal controller can in general not be computed exactly. Instead, a suboptimal $H_\infty$ control problem may be solved:

Find a controller $K$ that internally stabilizes $P$ and achieves $\|F_l(P, K)\|_\infty < \gamma$.

Similar to the computation of the $H_2$ optimal controller, the computation of the $H_\infty$ optimal controller is based on two Riccati equations:

$$0 = A^T X_\infty + X_\infty A - X_\infty (B_2 B_2^T - \frac{1}{\gamma^2} B_1 B_1^T) X_\infty + C_1^T C_1$$

(3.27)

$$0 = A Y_\infty + Y_\infty A^T - Y_\infty (C_2^T C_2 - \frac{1}{\gamma^2} C_1^T C_1) Y_\infty + B_1 B_1^T$$

(3.28)

with associated Hamiltonian matrices:

$$H_\infty = \begin{bmatrix} A & \frac{1}{\gamma^2} B_1 B_1^T - B_2 B_2^T \\ -C_1^T C_1 & -A^T \end{bmatrix}, \quad J_\infty = \begin{bmatrix} A^T & \frac{1}{\gamma^2} C_1^T C_1 - C_2^T C_2 \\ -B_1 B_1^T & -A \end{bmatrix}$$. (3.29)
There is an additional requirement for the existence of a $\mathcal{H}_\infty$ optimal controller, other than the non-negative definite solutions, $X_\infty$ and $Y_\infty$, of the Riccati equations. The spectral radius of $X_\infty Y_\infty$ may not be too large:

$$\rho(X_\infty Y_\infty) < \gamma^2.$$  \hfill (3.30)

If the non-negative definite solutions to the Riccati equations exist, and their spectral radius is not too large, then all stabilizing controllers $K$ of system $P$, such that $\|F_i(P,K)\|_\infty < \gamma$, are given by $K = F_i(K_\infty, Q)$, where

$$K_\infty(s) = \begin{bmatrix} A_K & B_K & Z_\infty B_2 \\ C_K & D_K & I \\ -C_2 & I & 0 \end{bmatrix}$$  \hfill (3.31)

$Q \in \mathcal{RH}_\infty$  \hfill (3.32)

$\|Q\|_\infty < \gamma$  \hfill (3.33)

$$A_K = A + \frac{1}{\gamma^2} B_1 B_1^T X_\infty + B_2 F_\infty + Z_\infty L_\infty C_2$$  \hfill (3.34)

$$B_K = -Z_\infty L_\infty$$  \hfill (3.35)

$$C_K = F_\infty$$  \hfill (3.36)

$$D_K = 0$$  \hfill (3.37)

and

$$F_\infty = -B_2^T X_\infty$$  \hfill (3.38)

$$L_\infty = -Y_\infty C_2^T$$  \hfill (3.39)

$$Z_\infty = (I - \frac{1}{\gamma^2} Y_\infty X_\infty)^{-1}.$$  \hfill (3.40)

The controller with $Q = 0$ is often called the central controller. When rewriting the equations of the $\mathcal{H}_\infty$ controller with $Q = 0$, we can write it in an observer-based form:

$$\dot{x}_k = A x_k + B_1 \underbrace{\frac{1}{\gamma^2} B_1^T X_\infty x_k + B_2 u + Z_\infty L_\infty (C_2 x_k - y)}_{\hat{w}_{\text{worst}}}$$  \hfill (3.41)

$$u = F_\infty x_k$$  \hfill (3.42)

This observer-based form is very similar to the $\mathcal{H}_2$ observer-based expression. The main difference is the addition of $\hat{w}_{\text{worst}}$, which denotes the estimate of the worst-case input. From Equation (3.42) it can be seen that also the $\mathcal{H}_\infty$ controller can be given a separation principle interpretation.

### 3.3 Concluding Remarks

In this chapter a general control problem is introduced. Both $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control problems can be stated in this general control problem. Both control strategies have some assumptions that are typically made. For $\mathcal{H}_\infty$ control, not all assumptions are needed, but they simplify...
computations a lot. The controllers are very similar. Two Riccati equations need to be solved. With the solutions of these equations, a $\mathcal{H}_2$ controller can be made. With $\mathcal{H}_\infty$ control the set of all stabilizing controllers can be defined. The $\mathcal{H}_2$ and $\mathcal{H}_\infty$ controllers can both be split into an estimator and a feedback part. In addition, the $\mathcal{H}_\infty$ controller also accounts for a worst case input. A considerable difference in $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control is the multiplicative property given in Definition 2.8. The 2-norm does not have this property, whereas the $\infty$-norm does. This property will appear very useful when considering uncertainty in robust control. Information about discrete time $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control can be found in [9], [11], [19], and references therein.
Chapter 4

Robust Control

In classic control techniques, a fixed model of the system is used. Every deviation of the output from the expected output is treated as noise. This is also the case with Prediction Error system identification ([10]). Noise can not destabilize the control-loop, whereas an error in the plant model can. To quantify the magnitude of the expected model error, an uncertainty model is needed. The use of an uncertainty model is what distinguishes Robust Control from other control techniques.

Scaling is used to improve the accuracy of the algorithms used for robust control. Scaling is also used to level the importance of signals or transfer functions in a cost function. Because it is unknown what the size of the signals will be, it is hard to scale them. Usually signals are divided by their maximum expected value or by their maximum allowable value. How to scale the signals is problem specific and closely related to the choice of dynamic weighting filters.

The methods mentioned in this chapter to design a controller can be used for $H_2$ and $H_\infty$ optimal control. In this chapter, also model uncertainty is introduced. In the case of $H_\infty$ control, the uncertainties can be added easily to the general control problem, because of the multiplicative property. Therefore the focus of this chapter will be on Robust Control based on $H_\infty$-optimization.

4.1 Dynamic Weighting Filters

Dynamic Weighting Filters are used to shape signals or transfer function to specify performance and robustness. In general the freedom of choice of filters is large. This ensures that a lot of control goals can be implemented, especially for MIMO systems. In the following subsections two main control design strategies are discussed: signal-based control and loop-shaping design.

4.1.1 Signal-Based $H_\infty$ Control

Signal-Based $H_\infty$ Control is similar to classical LQG control. In LQG control the (constant) weights $Q$ and $R$ are used to weight the output and input respectively. Similar to LQG, the frequency dependent weights $W$ and $V$ are used in signal-based control to weight the output
and input of the system, see Figure 4.1.
Weight $V$ describes the frequency content of the exogenous input signals. Weight $W$ describes the desired frequency content of the error and/or control signals. The input and output signals are

$$w = \begin{bmatrix} d \\ r \end{bmatrix} \quad \text{and} \quad z = \begin{bmatrix} e \\ u \end{bmatrix}$$

respectively.

Now a controller can be found by minimizing a norm on the input and output signals. If a 2-norm or a power norm is chosen, it can also be calculated by computing the induced $\infty$-norm of the system, with the weights included in the system:

$$z = Mw \quad \text{with} \quad M := F_l(P, K) \quad (4.1)$$

### 4.1.2 $\mathcal{H}_\infty$ Loop-shaping Design

$\mathcal{H}_\infty$ Loop-shaping Design involves the shaping of transfer functions. These transfer functions can be the open-loop transfer function $PK$, or closed loop transfer functions like $S$ and $T$. For SISO systems, control engineers often use manual loop shaping to design a controller. Some general guidelines are available how to shape the transfer functions of SISO systems. The closed loop transfer function $S$ should be small for low frequencies, so it needs a large weight at low frequencies. This ensures good performance for low frequencies. The error will consist mainly of high frequencies. The closed loop transfer function $T$ should be small for high frequencies and thus needs a large weight at high frequencies. This ensures robustness against high frequent model uncertainty.

For SISO systems, this works well. But for MIMO systems, this method is a bit more involved. It is not exactly clear how to shape the transfer functions. A major difference between SISO and MIMO systems is that inputs and outputs of MIMO systems have a direction. This implies that the minimum and maximum gain to an input can in general not be seen directly from the separate transfer functions between inputs and outputs. To cope with these
problems, singular values can be used. Singular values can be interpreted as a multivariable generalization of the gain. The maximum gain is given by $\bar{\sigma}$ and the minimum gain is given by $\sigma$. The singular values of a transfer function ($\sigma(PK)$, $\sigma(S)$ or $\sigma(T)$ for example) can be put in a Bode plot and then be shaped. The open-loop gain will in general have integral action at low frequencies and low-pass characteristics at high frequencies, see Figure 4.2 ([8]). In general the closed loop transfer functions are weighted with the inverse of some specified upper bound.

**Simulation example**

To illustrate the above control design strategies, simulations have been performed. Consider the system in Figure 4.3. The parameter values are: $C_1 = 0.8$, $D_1 = 10$, $M_1 = 1$, $C_2 = 1000$, $D_2 = 0.3$ and $M_2 = 0.01$. The bode magnitude plot of the system is shown in Figure 4.3.

To design controllers for this plant with a bandwidth of 5 Hz and 15 Hz, $H_\infty$ loop-shaping will be used. To define a certain level of performance, an upper bound for the Sensitivity function is used. The upper bounds are shown in Figure 4.4. The dynamic weighting filter for the Sensitivity function is the inverse of the upper bound shown in Figure 4.4.

To calculate the controller, the MATLAB function `hinfsyn` is used. The step responses of the system with the calculated controllers are shown in Figure 4.4. It can clearly be seen that the controller calculated with the more demanding weight on the Sensitivity function is faster.
4.2 Model Uncertainty

A robust controller is robust against model uncertainties. To guarantee robust stability and performance, uncertainty should be considered during the controller design process. Uncertainty can be easily addressed in the general control problem formulation. This is shown in Figure 4.5. The normalized uncertainties are represented by $\Delta$, such that $\bar{\Delta} = V_u \Delta W_u$ where $\bar{\Delta}$ is the actual uncertainty. The scaling matrices $V_u$ and $W_u$ are augmented with the plant model $P$. Also

$$\|\Delta\|_\infty \leq 1 \Rightarrow \sigma(\Delta(j\omega)) \leq 1, \forall \omega \in \mathbb{R}$$  \hspace{1cm} (4.3)

For different analysis, different configurations of the control problem are used. For Robust Performance (RP), the $N\Delta$ structure is used, which is shown in Figure 4.6. Where

$$N = F_l(P, K) = P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21}$$  \hspace{1cm} (4.4)

The total transfer function from $w$ to $z$ is now given by

$$z = Fw$$  \hspace{1cm} (4.5)

where

$$F = F_u(N, \Delta) = N_{22} + N_{21}\Delta(I - N_{11}\Delta)^{-1}N_{12}$$  \hspace{1cm} (4.6)
For Robust Stability (RS), the $M\Delta$ structure is used, see Figure 4.7. Here $M = N_{11}$. $N_{11}$ is dependent on the chosen form of the uncertainties. In figure 4.8 six common forms of unstructured uncertainty are shown. The transfer functions of the perturbed plants for these uncertainties are as follows:

- $\Pi_A$: $P_p = P + \Delta_A$ (4.7)
- $\Pi_I$: $P_p = P(I + \Delta_I)$ (4.8)
- $\Pi_O$: $P_p = (I + \Delta_O)P$ (4.9)
- $\Pi_{iA}$: $P_p = P(I - \Delta_{iA}P)^{-1}$ (4.10)
- $\Pi_{iI}$: $P_p = P(I - \Delta_{iI})^{-1}$ (4.11)
- $\Pi_{iO}$: $P_p = (I - \Delta_{iO})^{-1}P$ (4.12)

Where $P_p$ is a particular perturbed plant model and $\Pi$ is a set of possible perturbed plant models. If this set includes the real plant model, then conclusions for the set also hold for the true system. The element $N_{11}$ is now given by

$$N_{11} = M = W_u M_0 V_u$$ (4.13)

where $M_0$ is given by

- $P_p = P + \Delta_A$: $M_0 = K(I + PK)^{-1} = KS$ (4.14)
- $P_p = P(I + \Delta_I)$: $M_0 = K(I + PK)^{-1}P = T_I$ (4.15)
- $P_p = (I + \Delta_O)P$: $M_0 = PK(I + PK)^{-1} = T$ (4.16)
- $P_p = P(I - \Delta_{iA}P)^{-1}$: $M_0 = (I + PK)^{-1}P = SP$ (4.17)
- $P_p = P(I - \Delta_{iI})^{-1}$: $M_0 = (I + KP)^{-1} = S_I$ (4.18)
- $P_p = (I - \Delta_{iO})^{-1}P$: $M_0 = (I + PK)^{-1} = S$ (4.19)

for the different uncertainty models. For completeness, also the different LFTs of the plant with the interconnection structure of the uncertainty are given for the different uncertainty
models:

\[ P_p = F_u(Q, \Delta) \]  \hspace{1cm} (4.20)

with

\[ P_p = P + \bar{\Delta}_A : \quad Q = \begin{bmatrix} 0 & I \\ I & P \end{bmatrix} \]  \hspace{1cm} (4.21)

\[ P_p = P(I + \bar{\Delta}_I) : \quad Q = \begin{bmatrix} 0 & I \\ P & P \end{bmatrix} \]  \hspace{1cm} (4.22)

\[ P_p = (I + \bar{\Delta}_O)P : \quad Q = \begin{bmatrix} 0 & P \\ I & P \end{bmatrix} \]  \hspace{1cm} (4.23)

\[ P_p = P(I - \bar{\Delta}_{iA}P)^{-1} : \quad Q = \begin{bmatrix} P & P \\ P & P \end{bmatrix} \]  \hspace{1cm} (4.24)

\[ P_p = P(I - \bar{\Delta}_{iI})^{-1} : \quad Q = \begin{bmatrix} I & I \\ P & P \end{bmatrix} \]  \hspace{1cm} (4.25)

\[ P_p = (I - \bar{\Delta}_{iO})^{-1}P : \quad Q = \begin{bmatrix} I & P \\ I & P \end{bmatrix} \]  \hspace{1cm} (4.26)

### 4.3 Robust Stability

A major difference between representing uncertainties as uncertain dynamics \( \Delta \) or as noise injected in the control loop is that noise cannot destabilize the control loop, whereas \( \Delta \) can. This implies that not only the control loop with the nominal model and controller has to be...
checked for stability, but also the control loop with the uncertain dynamics included. The different criteria for nominal and robust stability and performance are as follows ([16]):

\[ NS \iff N \text{ is internally stable} \tag{4.27} \]
\[ NP \iff \|N_{22}\|_\infty < 1; \text{ and } NS \tag{4.28} \]
\[ RS \iff F = F_u(N, \Delta) \text{ is stable } \forall \Delta, \|\Delta\|_\infty \leq 1; \text{ and } NS \tag{4.29} \]
\[ RP \iff \|F\|_\infty < 1, \forall \Delta, \|\Delta\|_\infty \leq 1; \text{ and } NS \tag{4.30} \]

To check robust stability, it is sufficient to look at the \( M\Delta \) structure presented in Figure 4.7. The exogenous signals \( w \) and measured outputs \( z \) can not destabilize the control loop, and can thus be disregarded for stability. The small gain theorem can be used to guarantee stability of the \( M\Delta \) loop. Roughly stated the small gain theorem guarantees asymptotic stability if the loop gain is smaller than 1, i.e. \( \|M\Delta\|_\infty \leq 1 \). The \( \infty \)-norm can be interpreted as a maximum possible gain of the loop. Because this norm satisfies the multiplicative property and \( \Delta \) is unstructured and scaled such that \( \|\Delta\| \leq 1 \) the following is true:

\[ \|M\Delta\|_\infty = \|M\|_\infty \|\Delta\|_\infty = \|M\|_\infty \tag{4.31} \]

As stated in Equation (2.2) the \( \infty \)-norm equals the maximum singular value. So for unstructured uncertainty, RS is guaranteed if

\[ \|M\|_\infty < 1 \iff \bar{\sigma}(M(j\omega)) < 1, \forall \omega \in \mathbb{R} \tag{4.32} \]

All uncertainties considered previously were full-block uncertainties, unstructured uncertainties. If the uncertainty has a structure, this structure can be exploited to design a less conservative controller. Assume the structure of the uncertainty to be block-diagonal: \( \Delta = \text{diag}(\Delta_i) \). Scaling the inputs and outputs of \( M \) and \( \Delta \) as in Figure 4.9 has no effect. Thus RS is still guaranteed. Now assume \( D \) to be a block-diagonal scaling matrix, \( D = \text{diag}(d_i I_i) \), where \( d_i \) is a scalar and \( I_i \) is an identity matrix of the same dimension as the \( i \)th perturbation block, \( \Delta_i \). For this particular choice of \( D, D \) and \( \Delta \) are commutative: \( D\Delta D^{-1} = \Delta \). Thus

\[ \text{RS if } \bar{\sigma}(DMD^{-1}) < 1, \forall \omega, \tag{4.33} \]

because the scaling has no effect on stability. The least conservative RS condition is thus obtained by minimizing the scaled maximum singular value at each frequency:

\[ \text{RS if } \min_{D(\omega) \in \mathcal{D}} \bar{\sigma}(D(\omega)M(j\omega)D(\omega)^{-1}) < 1, \forall \omega \tag{4.34} \]

where \( \mathcal{D} \) is the set of block-diagonal matrices whose structure is compatible to that of \( \Delta \).

The Structured Singular Value \( \mu \) (also \( \text{Mu}, \mu \) or SSV) is a generalization of \( \sigma \). It is known that a system will be unstable if \( \det(I - M\Delta) = 0 \). \( \mu \) is defined as:

\[ \mu(M)^{-1} = \min_{\Delta} (\bar{\sigma}(\Delta) \mid \det(I - M\Delta) = 0) \tag{4.35} \]

So \( \mu \) is the inverse of the smallest maximum singular value of \( \Delta \) that destabilizes the loop. If \( \mu = 1 \), then the system is on the edge of stability. If \( \mu < 1 \), the system is stable and if \( \mu > 1 \), the system is unstable.
4.4 Robust Performance

Robust performance is achieved if $\|N(K)\|_\infty < 1$. To compute this norm under the influence of $\Delta$, the RP problem can also be cast as a RS problem. This is shown in Figure 4.10. Here $\hat{\Delta} = \text{diag}(\Delta, \Delta_p)$, where $\Delta$ is the dynamic uncertainty and $\Delta_p$ is a fictitious performance uncertainty. $\Delta_p$ is a full complex matrix. Because $\hat{\Delta}$ has structure, requiring $\|N(K)\|_\infty < 1$ can yield a conservative controller. Solving $\|N(K)\|_\mu < 1$ will generally be less conservative. So even if the dynamic uncertainty is unstructured, $\mu$-synthesis can be desirable if RP is pursued.

Simulation example

As stated above, a model mismatch with the plant can lead to instability of the control loop. The way to deal with this undesired phenomenon is by modeling the uncertain dynamics and design a robust controller. This will be illustrated in the following simulation example.

Consider the model and the calculated controllers of the former simulation example. Assume the model of the plant is not perfect. The real parameter $C_2$ is only half of the estimated value: $C_2_{\text{model}} = 1000$ and $C_2_{\text{plant}} = 500$. All other parameters are estimated perfectly. The bode magnitude plots of the model, $P$, and the real system, $P_p$, are shown in Figure 4.11. As can be seen, the main consequence of the mismatch is the shift of the resonance peak.
Figure 4.11: Bode magnitude plot of the system and the perturbed system.

Figure 4.12: Comparison of the step responses of the system and the perturbed system with the calculated controllers.

Figure 4.13: Bode plot and Nyquist plot of the nominal and perturbed loop transfer function.
The question is if this is a problem. As can be seen in Figure 4.12, the step response of the real system and the model are almost the same when the low bandwidth controller is applied. However, when the high bandwidth controller is applied, the control loop with the real system is unstable. Looking at the Bode plot and the Nyquist plot of the open-loop system (Figure 4.13), it is clear why this is the case. The notch filter in the controller does not suppress the resonance anymore, resulting in a resonance, anti-resonance sequence, which causes a loss of phase.

Thus, the next step is the modeling of an uncertainty model. Assume it is known that the value for $C_2$ is wrong and that it can be 50% higher or lower. In total 21 systems are considered with different values for $C_2$, ranging from 500 to 1500. For multiplicative uncertainty, it is known that $P_p = P(I + \Delta)$, thus $\Delta = P^{-1}P_p - I$. In Figure 4.14 the amplitude of all 21 realizations of the multiplicative uncertainty are shown.

Although this is a very accurate modeling of the uncertainty, it is not useful. It is of a too high order. To get a low order uncertainty model, a low order approximation of the upper bound is made. This is shown in Figure 4.14. The parameter uncertainty is now represented by a dynamic uncertainty, which can be used in the $H_\infty$ Robust Controller design easily.

As stated in the previous section, it is known that $M$ should contain the Complementary Sensitivity, $T$, when dealing with multiplicative uncertainty in a $M\Delta$-structure. This can easily be used in a mixed-sensitivity loop-shaping design. The multiplicative uncertainty will be used to shape the Complementary Sensitivity to ensure robustness. A Sensitivity will be used to define the level of performance. The upper bounds for $S$ and $T$ are shown in Figure 4.15. The MATLAB function \texttt{mixsyn} is used to calculate the robust controller. The step response of the closed loop system is shown in Figure 4.15. It can clearly be seen that the control loop is stable. To check whether loop is guaranteed to be stable, one should look at the achieved $S$ and $T$. These are plotted in Figure 4.16. The performance goal is achieved if $S < S_{\text{upper}}$ and the system is guaranteed to be stable if $T < T_{\text{upper}}$. As can be seen from Figure 4.16 the control loop is guaranteed to be stable and achieves its desired performance.

As we have seen, an $H_\infty$ controller of 15Hz caused the control loop to be unstable. If uncertainty is taken into account, would a robust controller be able to achieve the desired performance? To check this, another simulation is performed. The uncertainty is, of course, the same as with the low bandwidth controller. The upper bound on the sensitivity is now more demanding. The upper bounds of $S$ and $T$ are shown in Figure 4.17. Again, \texttt{mixsyn} is used to calculate the controller. The step responses of the resulting closed loop systems are
Figure 4.15: Bode magnitude plot of the upper bounds for $S$ and $T$ and the step response of the model and system with the calculated low bandwidth robust controller.

Figure 4.16: Bode magnitude plot of the upper bounds on $S$ and $T$ and their realizations.

Figure 4.17: Bode magnitude plot of the upper bounds for $S$ and $T$ and the step response of the model and system with the calculated high bandwidth robust controller.
Figure 4.18: Bode magnitude plot of the upper bounds on $S$ and $T$ and their realizations.

depicted in Figure 4.17. Both the model and the real system are stabilized by the controller. However, the performance of the real system is worse than the performance of the model. The step response contains an unwanted oscillation. When looking at the Sensitivity (Figure 4.18), it is clear why this is the case. The controller can not achieve the desired performance for the real system at 35Hz. This is exactly the frequency of the unwanted oscillation present in the step response.

When looking at the Complementary Sensitivity, we see that the achieved nominal $T$ is smaller than the upper bound. This guarantees stability for the entire model set, and thus for the real plant. This was already seen in Figure 4.17.
Chapter 5

Identification

To be able to design a robust controller, a model of the system is needed. The system has to be identified. First some general considerations are presented. These are applicable both to time domain and frequency domain identification. Then some advantages of frequency domain identification are presented. In Section 5.2, time domain identification is considered. The model structures presented in this section are generally applicable. Then time domain validation is discussed. Section 5.4 considers input generating for frequency domain identification. Frequency domain identification is discussed in Section 5.5. This chapter is concluded with frequency domain validation.

5.1 General Considerations

The identification of a system always involves three basic entities:

1. data set
2. model structure
3. rule to find ”best” model

1: A data set should be generated to be able to identify a system. To generate a good data set, attention should be paid to the experiment design. Certain signals should be chosen as inputs and outputs. For the data set to be maximally informative, also special attention should be paid to the input design.

2: A set of candidate models has to be chosen. The question which set should be chosen is a difficult one, but also a very important one. In practice, the set of candidate models is based on a priori knowledge and engineering insight. The set of candidate models can be updated in an iterative way during the identification procedure.

3: The model in the set of candidate models that fits the data set ”best” should be found. What ”best” is, depends on the chosen criterium. Also the purpose of the model is important. There are other criteria for models for open loop control than for models for closed loop control for example.

When a model is generated, it should also be validated. Is it valid for its purpose? Is it good enough? The question whether the model is good enough is an important one, because one must accept that a model will never be a perfect representation of reality (We are not considering simulation examples here). The identified model must at best be regarded as a
A very important choice in system identification is whether the identification is performed in the time domain or in the frequency domain. Both strategies have some advantages and disadvantages. In the scope of this project, the frequency domain approach is preferred. As mentioned in [14], some advantages of the frequency domain are:

1. Easy noise reduction: The non-excited (noisy) frequency lines are easily eliminated.
2. Data reduction: A large number of time-domain samples are replaced by a small number of spectral lines.
3. When using a discrete Fourier transform to calculate the spectra, the frequency domain noise is asymptotically (number of time domain samples going to infinity) complex normal distributed.
4. No initial state estimation of the system.
5. Model validation: Using periodic excitations one has very good point estimates of the frequency response function.
6. It is very easy to combine data from different experiments.

### 5.2 Time Domain Identification

A commonly used time domain identification procedure is the prediction error identification method. A general assumption is that the data collected during experiments is generated by the following system:

\[
\begin{align*}
y(t) &= G_0(z)u(t) + v(t), \\
v(t) &= H_0(z)e(t).
\end{align*}
\]

Here \(y(t)\) is the output, \(u(t)\) the input applied to the system, \(e(t)\) the collection of all effects of disturbances and non-measurable inputs and \(G_0(z)\) and \(H_0(z)\) are the real plant and real disturbance model respectively. This system can be seen in Figure 5.1.

Some assumptions on this system are that \(u(t)\) and \(e(t)\) are uncorrelated. Also, \(G_0(z)\) is a stable proper rational transfer function and \(H_0(z)\) is stable, inversely stable and monic.
Furthermore, $e(t)$ is assumed to be white noise, i.e. a sequence of independent random variables with zero mean:

\[
Ee(t) = 0 \\
R_e(\tau) \doteq Ee(t)e(t-\tau) = \sigma_e^2 \cdot \delta(\tau)
\]  

(5.2)  

(5.3)

And thus the following holds for $v(t)$:

\[
E v(t) = 0 \tag{5.4}
\]

\[
\Phi_v(\omega) = |H_0(e^{j\omega})|^2 \cdot \sigma_e^2. \tag{5.5}
\]

Consider the system in (5.1) and given observations $(y(s), u(s)), s \leq t - 1$. Can $y(t)$ be predicted one-step-ahead? By rewriting Equation (5.1) and substituting $e(t) = H_0(z)^{-1}(y(t) - G_0(z)u(t))$, the following result can be found:

\[
y(t) = G_0(z)u(t) + H_0(z)e(t) \\
= G_0(z)u(t) + (H_0(z) - 1)e(t) + e(t) \tag{5.6}
\]

\[
y(t) = G_0(z)u(t) + (H_0(z) - 1)H_0^{-1}(z)(y(t) - G_0(z)u(t)) + e(t) \tag{5.7}
\]

\[
y(t) = H_0^{-1}(z)G_0(z)u(t) + (1 - H_0^{-1}(z))y(t) + e(t) \tag{5.8}
\]

If $G_0(z)$ and $H_0(z)$ are known, then everything for the estimation of $y(t)$ is known, except for $e(t)$. So $y(t)$ can be decomposed in an estimation $\hat{y}(t|t-1)$ that is available at time instant $t-1$ and $e(t)$, which is unavailable at that time:

\[
y(t) = \hat{y}(t|t-1) + e(t) \tag{5.9}
\]

$\hat{y}(t|t-1)$ is the best estimate of $y(t)$ at time instant $t-1$. However, in practice the plant and disturbance models $G_0(z)$ and $H_0(z)$ are unknown. They are approximated by $G(z, \theta)$ and $H(z, \theta)$ respectively.

\[
\hat{y}(t|t-1; \theta) = H^{-1}(z, \theta)G(z, \theta)u(t) + (1 - H^{-1}(z, \theta))y(t) \tag{5.10}
\]

Consider the following parameter estimation

\[
\hat{\theta}_N = \min_{\theta} V_N(\theta, Z^N) \tag{5.11}
\]

with $V_N(\theta, Z^N)$ a quadratic criterion on the prediction error:

\[
V_N(\theta, Z^N) = \frac{1}{N} \sum_{t=1}^{N} e^2(t, \theta). \tag{5.12}
\]

And the prediction error defined as

\[
ev(t, \theta) \doteq y(t) - \hat{y}(t|t-1; \theta). \tag{5.13}
\]

When rewriting the prediction error, it can easily be seen that it can be separated into two parts

\[
ev(t, \theta) = y(t) - H^{-1}(z, \theta)G(z, \theta)u(t) - (1 - H^{-1}(z, \theta))y(t) \tag{5.14}
\]

\[
= H^{-1}(z, \theta)(y(t) - G(z, \theta)u(t)) \tag{5.15}
\]

\[
= H^{-1}(z, \theta)(G_0(z)u(t) + H_0(z)e(t) - G(z, \theta)u(t)) \tag{5.16}
\]

\[
= \frac{G_0(z) - G(z, \theta)}{H(z, \theta)} u(t) + \frac{H_0(z)}{H(z, \theta)} e(t). \tag{5.17}
\]

(5.18)
A graphical interpretation of the prediction error can be seen in Figure 5.2. The first part of the prediction error in Equation (5.18) vanishes if the plant estimation equals the real plant. Thus $V_N(\theta, Z^N)$ is minimal with respect to $G(z, \theta)$ if $G(z, \theta) = G_0(z)$.

When considering the second part of the prediction error, it should be reminded that $H$ is monic. Thus

$$
\frac{H_0(z)}{H(z, \theta)} e(t) = [1 + \gamma_1(\theta)z^{-1} + \gamma_2(\theta)z^{-2} + \ldots] e(t).
$$

(5.19)

When considering the quadratic criterion, the estimation equals

$$
E \varepsilon^2 = [1 + \gamma_1(\theta)^2 + \gamma_2(\theta)^2 + \ldots] \sigma^2_e.
$$

(5.20)

And this is minimal for $\gamma_1(\theta) = \gamma_2(\theta) = \ldots = 0$. Thus it is minimal for $H(z, \theta) = H_0(z)$. It can be concluded that $V_N(Z^N, \theta)$ is a useful criterion, because it is minimal for $G(z, \theta) = G_0(z)$ and $H(z, \theta) = H_0(z)$. From Equation (5.18) and Figure 5.2 it can be seen that $\varepsilon(t, \theta) = e(t)$ if $G(z, \theta) = G_0(z)$ and $H(z, \theta) = H_0(z)$.

So far, no assumptions have been made on the model structure. Polynomial fractions are commonly used to represent the models, for example:

$$
G(z, \theta) = \frac{b z^{-1}}{1 + f z^{-1}}; \quad H(z, \theta) = \frac{1 + e z^{-1}}{1 + d z^{-1}}; \quad \theta = [b; c; d; f]^T
$$

(5.21)

The most important model structures are listed in Table 5.1. Some important model properties can be distinguished. The ARX and FIR model structures are linear in the parameters [10]. A consequence is that the minimization of $V_N(\theta, Z^N)$ has an analytic solution. This is computationally very attractive (although an inverse has to be calculated, which can be very cumbersome for large data sets). The ARMAX, OE, and BJ model structures are not linear
Table 5.1: Overview of model structures.

<table>
<thead>
<tr>
<th>Model structure</th>
<th>$G(z, \theta)$</th>
<th>$H(z, \theta)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARX</td>
<td>$B(z^{-1}, \theta)$</td>
<td>$A(z^{-1}, \theta)$</td>
</tr>
<tr>
<td>ARMAX</td>
<td>$B(z^{-1}, \theta)$</td>
<td>$C(z^{-1}, \theta)$</td>
</tr>
<tr>
<td>OE</td>
<td>$B(z^{-1}, \theta)$</td>
<td>$F(z^{-1}, \theta)$</td>
</tr>
<tr>
<td>FIR</td>
<td>$B(z^{-1}, \theta)$</td>
<td>$1$</td>
</tr>
<tr>
<td>BJ</td>
<td>$B(z^{-1}, \theta)$</td>
<td>$C(z^{-1}, \theta)$</td>
</tr>
</tbody>
</table>

in the parameters. These model structures need a search to find a minimum of $V_N(\theta, Z^N)$, with the chance of finding a local minimum.

Another important property of the model structures is the independent parametrization of the plant and disturbance model. The ARX and ARMAX model structures are not independent parameterized due to the common denominator $A(z^{-1}, \theta)$. The OE, FIR and BJ model structures are independent parameterized. This is very important for the estimation of the plant if the true system is not in the model set. This issue will be discussed next.

5.3 Time Domain Validation

A certain $\hat{\theta}_N$ is found by minimizing $V_N(\theta, Z^N)$. But then the question arises if the identified model is close to the real system. To check this, a validation step is needed. The data set to validate the identified model is different from the data set used to identify the system, else validation would make no sense ([10]). To analyze the different possible outcomes of the validation, a model set is introduced.

$$\mathcal{M} := \{(G(z, \theta), H(z, \theta)) | \theta \in \Theta \subset \mathbb{R}^d\}$$ (5.22)

Let the true system, determined by $G_0(z)$ and $H_0(z)$, be denoted by $S$. If $G(z, \theta) = G_0(z)$ and $H(z, \theta) = H_0(z)$, then the prediction error will be equal to the (unknown) white noise $e(t)$. The prediction error should only contain white noise. So the autocorrelation of the prediction error should be flat and the cross correlation with the input should be zero.

$$R_e(\tau) = \sigma_e^2 \delta(\tau)$$

$$R_{eu}(\tau) = 0 \ \forall \ \tau$$ (5.24)

If we observe this then

$$\varepsilon(t, \theta) = \overbrace{G_0(z) - G(z, \theta)}^{=0} u(t) + \overbrace{H_0(z)}^{=1} e(t)$$

$$\Leftrightarrow G(z, \theta) = G_0(z) \quad \text{and} \quad H(z, \theta) = H_0$$ (5.25)

$$\Leftrightarrow S \in \mathcal{M}$$ (5.27)

And the identified model is thus validated.

If we observe the following

$$R_e(\tau) \neq \sigma_e^2 \delta(\tau)$$

$$R_{eu}(\tau) = 0 \ \forall \ \tau,$$ (5.28)

$$R_{eu}(\tau) = 0 \ \forall \ \tau,$$ (5.29)
then
\[
\varepsilon(t, \theta) = \left\{ \begin{array}{l}
\frac{G_0(z) - G(z, \theta)}{H(z, \theta)} u(t) + \frac{H_0(z)}{H(z, \theta)} e(t) = 0 \\
\neq 1
\end{array} \right.
\] (5.30)
⇔ \( G(z, \theta) = G_0(z) \) and \( H(z, \theta) \neq H_0 \) (5.31)
⇔ \( S \notin \mathcal{M} \) but \( G_0(z) \in \mathcal{G} \) for \( \mathcal{M} \) OE, BJ or FIR (5.32)

Here, \( \mathcal{G} \) is the set of plant models in the model set \( \mathcal{M} \). This situation can occur for OE, BJ and FIR because the parametrization of the plant and disturbance model is independent. For ARX and ARMAX this situation can not occur, because they are not independent parameterized. In some situations it is very useful if the plant model can be validated although the system model is invalidated. This is a big advantage of the OE, BJ and FIR model structures.

If we observe the following
\[
R_{\varepsilon}(\tau) \neq \sigma^2_\varepsilon \delta(\tau)
\] (5.33)
\[\exists \ \tau \ \text{s.t.} \ R_{\varepsilon u}(\tau) \neq 0, \] (5.34)
then
\[
\varepsilon(t, \theta) = \left\{ \begin{array}{l}
\frac{G_0(z) - G(z, \theta)}{H(z, \theta)} u(t) + \frac{H_0(z)}{H(z, \theta)} e(t) = \neq 0 \\
\neq 1
\end{array} \right.
\] (5.35)
⇔ \( G(z, \theta) \neq G_0(z) \) (5.36)
⇔ \( \{ \begin{array}{l}
either \ S \notin \mathcal{M} \text{ with } G_0(z) \in \mathcal{G} \text{ for } \mathcal{M} \text{ ARX or ARMAX} \\
or \ S \notin \mathcal{M} \text{ with } G_0(z) \notin \mathcal{G}
\end{array} \) (5.37)

And the system model is thus invalidated.

All validation situations considered up till now are situations with an infinite amount of data. In practice a finite data set is used. With a finite data set it is practically impossible to achieve conditions (5.23) and (5.24), so conditions should be loosened a bit. The autocorrelation and the cross correlation can be calculated as follows:
\[
\hat{R}^N_{\varepsilon}(\tau) = \frac{1}{N} \sum_{t=1}^{N-\tau} \varepsilon(t + \tau, \hat{\theta}_N) \varepsilon(t, \hat{\theta}_N)
\] (5.38)
\[
\hat{R}^N_{\varepsilon u}(\tau) = \frac{1}{N} \sum_{t=1}^{N-\tau} \varepsilon(t + \tau, \hat{\theta}_N) u(t).
\] (5.39)

By considering 99%-confidence regions of these estimates, it can be checked if \( \hat{R}^N_{\varepsilon}(\tau) \) and \( \hat{R}^N_{\varepsilon u}(\tau) \) lie within their confidence regions for all \( \tau \).

Now a nominal model is identified and validated. For a lot of control applications, this is sufficient. However, when identifying for robust control, an uncertainty model is needed. Let us first recap some properties of prediction error identification and use that to construct an uncertainty model.

It is assumed that the noise \( e(t) \) is white. Thus \( \varepsilon(t, \theta) \) is also white noise if the system is perfectly identified. The noise \( e(t) \) is a random variable, corrupting the finite data set \( Z^N \).
This results in the estimation \( \hat{\theta}_N \) being a random variable too. Different experiments will thus result in different estimations of \( \theta \). For the situation that \( \mathcal{S} \in \mathcal{M} \), the following can be said about the estimation [10]:

\[
\hat{\theta}_N \sim A_s N(\theta_0, P_0) \quad (5.40)
\]

\[
\hat{\theta}_N \rightarrow \theta_0 \text{ w.p. 1 when } N \rightarrow \infty \text{ (i.e. } P_0 \rightarrow 0 \text{ when } N \rightarrow \infty \).
\]

(5.41)

The covariance matrix \( P_0 \) is defined as follows:

\[
P_0 = \sigma_e^2 \big( \bar{E} \psi(t, \theta_0) \psi^T(t, \theta_0) \big)^{-1}
\]

(5.42)

\[
P_0 = \sigma_e^2 \big( \bar{E} \psi(t, \theta_0) \psi^T(t, \theta_0) \big)^{-1}
\]

(5.43)

with

\[
\psi(t, \theta_0) = \left. \frac{\partial \bar{y}(t, \theta)}{\partial \theta} \right|_{\theta=\theta_0} = - \left. \frac{\partial \epsilon(t, \theta)}{\partial \theta} \right|_{\theta=\theta_0}.
\]

(5.44)

\( P_0 \) is dependent on \( N \), as can be seen in Equation (5.43), but also on \( u(t) \). To show this, Equations (5.43) and (5.18) are used:

\[
\psi(t, \theta_0) = \left. \frac{\partial \epsilon(t, \theta)}{\partial \theta} \right|_{\theta=\theta_0} = \frac{\Lambda_G(z, \theta_0)}{H(z, \theta_0)} u(t) + \frac{\Lambda_H(z, \theta_0)}{H(z, \theta_0)} \epsilon(t),
\]

(5.45)

with

\[
\Lambda_G(z, \theta) = \frac{\partial G(z, \theta)}{\partial \theta} \quad \text{and} \quad \Lambda_H(z, \theta) = \frac{\partial H(z, \theta)}{\partial \theta}. \quad (5.46)
\]

Now defining \( \Gamma_G = \frac{\Lambda_G^\ast \Lambda_G}{HH^\ast} \) and \( \Gamma_H = \frac{\Lambda_H^\ast \Lambda_H}{HH^\ast} \), and using Parseval’s theorem

\[
P_0 = \sigma_e^2 \big( \bar{E} \psi(t, \theta_0) \psi^T(t, \theta_0) \big)^{-1} \Rightarrow
\]

\[
P_0 = \sigma_e^2 \bigg( \frac{1}{2\pi} \int_{-\pi}^{\pi} \Gamma_G(e^{j\omega}, \theta_0) \Phi_u(\omega) + \Gamma_H(e^{j\omega}, \theta_0) \sigma_e^2 \ d\omega \bigg)^{-1}.
\]

(5.47)

(5.48)

Thus \( P_0 \) is a function of \( u(t) \) and \( N \) and we can therefore influence the value of \( P_0 \) by appropriately choosing \( u(t) \) and \( N \).

\( P_0 \) can not be calculated with the expressions above, but an estimate, \( \hat{P}_0 \), can be deduced using the following expression:

\[
\hat{P}_0 = \frac{\sigma_e^2}{N} \left( \frac{1}{N} \sum_{t=1}^{N} \psi(t, \hat{\theta}_N) \psi^T(t, \hat{\theta}_N) \right)^{-1} \quad \text{with} \quad \hat{\sigma}_e^2 = \frac{1}{N} \sum_{t=1}^{N} \epsilon(t, \hat{\theta}_N)^2.
\]

(5.49)

\( \hat{P}_0 \) is a measure for the spread of \( \hat{\theta}_N \) and can be used to build an uncertainty region in the parameter space:

\[
(\theta_0 - \hat{\theta}_N)^T P_0^{-1} (\theta_0 - \hat{\theta}_N) \sim \chi^2(k), \quad (5.50)
\]

with \( k \) the dimension of \( \hat{\theta}_N \). Of course, this expression can not be calculated because \( \theta_0 \) is unknown. This expression can be used to estimate an ellipsoid \( U \) in which the true parameter \( \theta_0 \) lies with probability 99% for example:

\[
U = \left\{ \theta \in \mathbb{R} | (\theta - \hat{\theta}_N)^T P_0^{-1} (\theta - \hat{\theta}_N) \leq \alpha \right\},
\]

(5.51)

with \( \alpha \) such that \( Pr(\chi^2(k) < \alpha) = 0.99. \)

37
5.4 Frequency Domain Input Generation

Another approach to system identification is identification in the frequency domain. A very important tool to transform time domain data into frequency domain data is the Fourier transform. Most real-life systems are continuous in time. But signal processing on computers is done in discrete-time. The signals used are discretized and quantized representations of the continuous time signals. These signals are processed with a Discrete Fourier Transform to transform them to the frequency domain. All these steps introduce errors. To minimize these errors one should know the impact of each step.

A discrete time signal, $u_d(n)$, can be represented as a continuous time signal, $\tilde{u}_d(t)$, with all its power at the discrete time instances $nT_s$:

$$\tilde{u}_d(t) = u(t) \sum_{n=-\infty}^{\infty} \delta(t - nT_s) \quad (5.52)$$

Let the spectrum of the discrete time signal be defined as:

$$U_d(e^{j2\pi fT_s}) = \sum_{n=-\infty}^{\infty} u_d(n)e^{j2\pi fnT_s} \quad (5.53)$$

Then the following holds:

$$U_d(e^{j2\pi fT_s}) = \tilde{U}_d(j2\pi f) = \int_{-\infty}^{\infty} \tilde{u}_d(t)e^{-j2\pi ft}dt \quad (5.54)$$

It is known that a multiplication in time domain corresponds to a convolution in frequency domain. Thus the discrete time spectrum and the continuous time spectrum are related by a convolution:

$$U_d(e^{j2\pi fT_s}) = U(j2\pi f) * (f_s\delta_{f_s}(f)) = \frac{1}{T_s} \sum_{k=-\infty}^{\infty} U(j2\pi (f - kf_s)) \quad (5.55)$$

with $\delta_{f_s}(f) = \sum_{k=-\infty}^{\infty} \delta(f - kf_s)$.

This results in a repeated spectrum in the frequency domain with period $f_s$, as can be seen in Figure 5.3. If the measured signal contains frequencies $f > f_s/2$, aliasing occurs. The repeating spectra overlap. If continuous time performance is considered, this should be avoided as the information at those overlapping frequencies is lost. For real valued discrete time signals the continuous spectrum from 0 to $f_s/2$ corresponds to the continuous spectrum of the continuous time signal.
Up till now, all transformations are performed with an infinite data set. Of course, in practice it is impossible to capture and process an infinite amount of data. To restrict the amount of data, we only consider the samples that appear in the measurement window. This can be represented as multiplying the time domain signal with a rectangular window $w(t)$:

$$w(t) \begin{cases} = 1 & 0 < t < T \\ = 0 & \text{elsewhere} \end{cases} \quad (5.56)$$

The sampled time domain signal and its spectrum are then given by:

- **time domain** $w(t) \hat{u}_d(t)$
- **frequency domain** $W(j2\pi f) * U_d(e^{j2\pi fT}) \quad (5.57)$

with $W(j2\pi f) = T e^{-j2\pi f \frac{T}{2}} \text{sinc}(j2\pi f \frac{T}{2})$ and $\text{sinc}(x) = \frac{\sin(x)}{x}$. Due to the sinc like spectrum of the rectangular window, the power of a frequency is distributed over neighboring frequencies. This is called leakage. In Figure 5.4 the continuous spectrum of a windowed discrete time signal is given.

The calculated spectrum is still continuous. A computer can only calculate a finite number of frequencies. The number of frequencies should be restricted. Discretization in the frequency domain is needed. An equidistant frequency grid with spacing $1/T$ is selected. In general, this gives poor results, as can be seen in Figure 5.4. This is because the measured frequencies are in general not at the DFT grid. This can in fact already be seen from the time domain signal.

Sampling in the frequency domain can be interpreted as multiplication of the spectrum with a series of Dirac impulses, just like sampling in time domain. Sampling in the frequency domain, results in a repeated windowed time domain signal. If $T$ is not an integer multiple of the signal period, a discontinuity will be observed at the window border. If $T$ is an integer multiple of the signal period, no discontinuity will be observed. The resulting discrete spectrum will be exact and no leakage will occur. This is caused by the fact that if $T$ is an integer multiple of the signal period, the continuous spectrum of the windowed signal will be exactly 0 at the DFT grid, except for the true frequency. The zero crossings of the spectrum of the rectangular window of length $T$ coincide with the DFT grid with spacing $1/T$.

Considering all the above, it is only logical to use multi-sine inputs, and not stochastic/noisy signals, to identify a system. With multi-sine inputs, no leakage will occur. For linear systems, aliasing can be disregarded due to discrete input signals.
5.5 Frequency Domain Identification

To calculate the response of a system in time domain, the impulse response of the system should be convoluted with the input signal:

\[
y(t) = (g * u)(t) = \int_{-\infty}^{\infty} g(t - \tau)u(\tau)d\tau. \tag{5.59}
\]

Transformation of \(g(t)\) to the frequency domain results in \(G(\Omega k)\). A very useful property of the frequency domain is that the response of a system in frequency domain can be calculated by multiplying the input with the frequency domain representation of the impulse response ([15]):

\[
Y(k) = G(\Omega k)U(k), \tag{5.60}
\]

with \(\Omega k = j\omega \) or \(\Omega k = e^{-j\omega T_k}\). From experimental input-output data the system can now easily be identified for frequency \(f_k\) by:

\[
G(\Omega k) = \frac{Y(k)}{U(k)}, \tag{5.61}
\]

with \(f_k = k/T\) and \(k\) an integer. Of course this is an ideal situation. In practice every measurement is corrupted by noise. To reduce the influence of noise, one can perform \(M\) measurements (or perform one long measurement and divide it into \(M\) smaller data sets) to "average out" the noise. This and its consequences will be discussed next.

As stated in [15] noise causes a bias in the estimation of the system. If the noise of the input and output is uncorrelated, the bias depends on the variance of the input only:

\[
b(k) = -\exp \left( -\frac{|U_0(k)|^2}{\sigma_Y^2(k)} \right) \tag{5.62}
\]

If the noise of the input and output is correlated, the bias depends on the variances of both the input and the output:

\[
b(k) = -\exp \left( -\frac{|U_0(k)|^2}{\sigma_Y^2(k)} \right) \left( 1 - \frac{\sigma_Y^2(k)}{\sigma_Y(k)\sigma_Y(k)} \frac{U_0(k)/\sigma_U(k)}{U_0(k)/\sigma_Y(k)} \right) \tag{5.63}
\]

As our focus is on identification for control, the input is exactly known. The variance of the input will thus be zero, which results in an unbiased estimate. Another consequence of noise corrupted data is that not only the data has become a stochastic variable, but also the identified model of the system has a stochastic behavior. As stated above, the variance of the input is zero. The unbiased estimates of the true variances of the output and the system are given by ([15]):

\[
\hat{\sigma}^2_Y(k) = \frac{1}{M - 1} \sum_{i=1}^{M} |Y[i](k) - \hat{Y}(k)|^2 \tag{5.64}
\]

\[
\hat{\sigma}^2_G(k) = \frac{|\hat{G}(\Omega k)|^2}{M} \left( \frac{\hat{\sigma}^2_Y(k)}{|Y(k)|} + \frac{\hat{\sigma}^2_U(k)}{|U(k)|} - 2Re \left( \frac{\hat{\sigma}^2_{YY}(k)}{Y(k)U(k)} \right) \right). \tag{5.65}
\]

In our case the variance of the system reduces to:

\[
\hat{\sigma}^2_G(k) = \frac{|\hat{G}(\Omega k)|^2}{M} \left( \frac{\hat{\sigma}^2_Y(k)}{|Y(k)|} \right) \tag{5.66}
\]
The division by $M$ occurs due to an averaging effect if the noise is uncorrelated from one subrecord to another. With the method described above, a nonparametric model can be estimated. This model gives an estimate of the frequency response for each frequency, and the variance can also be calculated for each frequency. Although this kind of models are used frequently for controller design with loop-shaping techniques, they are useless in most model-based controller design methods, like $\mu$-synthesis. For most automated controller design methods, a parametric model is needed: $\hat{G}(\Omega_k, \theta)$. Also a cost function is needed to check whether or not the parametric model is close to the nonparametric model. A commonly used cost function is ([15]):

$$V(\theta, Z) = \frac{1}{F} \sum_{k=1}^{F} W(\Omega_k) \left| \hat{G}(\Omega_k) - G(\Omega_k, \theta) \right|^2.$$  \hspace{1cm} (5.67)

Minimizing the least square distance is intuitive. The weighting $W(\Omega_k)$ is a user defined weighting. An example is $W(\Omega_k) = \frac{1}{a\hat{z}(k)}$. Now frequencies with a small variance are more important than frequencies with a large variance. This is called the Maximum Likelihood estimator. For identification for control it can be useful to increase the weighting around the expected bandwidth.

5.6 Frequency Domain Validation

Frequency domain validation is done in a similar way as time domain validation, but now the focus is on the cost function instead of the residual. If there are no model errors left, the following holds for the cost function ([15]):

$$E(V(\hat{\theta}(Z), Z)) = V_{\text{noise}} \hspace{1cm} (5.68)$$
$$\text{var}(V(\hat{\theta}(Z), Z)) = V_{\text{noise}} \hspace{1cm} (5.69)$$

with $V_{\text{noise}} = F - n_d/2$. If the cost function equals the noise value within the uncertainty bounds, for example, with 95% confidence, then

$$V_{\text{noise}} - 2V_{\text{noise}}^{1/2} < V(\hat{\theta}(Z), Z) < V_{\text{noise}} + 2V_{\text{noise}}^{1/2}. \hspace{1cm} (5.70)$$

And also

$$R_{ee} \approx 0 \hspace{1cm} (5.71)$$

with $e(\Omega_k, \theta) = \hat{G}(\Omega_k) - G(\Omega_k, \theta)$. If the cost function is significantly larger than the noise value, systematic model errors are typically present. $R_{ee} \neq 0$ indicates there are still unmodeled dynamics in the system. If $R_{ee} \approx 0$, this can indicate the existence of nonlinear distortions.

Only the identification of the system itself is considered. The identification of an uncertainty model, needed for robust control, is disregarded. There are 2 reasons for this. The identification and validation techniques discussed here use stochastic models. Controller design in previous chapters considered deterministic models. Also, measurements always contain noise. How to separate the noise from the uncertainty is not straightforward. If all noise is treated as uncertainty, one is overly conservative. If all
uncertainty is treated as noise, one is overly optimistic. Identification of uncertainty models and applications to systems can be found in [13] and [4]. Ways to separate the noise and the model error are introduced in [12], [3], and [17].
Chapter 6

Conclusions and Recommendations

Several controllers have been discussed. One of the most important drawbacks of the LQ, LQG and $H_2$ controller designs is the lack of robustness against model errors. These controllers achieve a certain performance level at nominal operating conditions. If the operating conditions change, the performance level can decrease a lot. In fact, may even result in instability of the system.

With $H_\infty$ control, it is possible to account for modeling errors. And thus, design a robust controller. A small drawback is that an additional uncertainty model is needed. The performance of a $H_\infty$ robust controller can be optimized and quantified by using a fictitious performance uncertainty $\Delta_p$ in $\mu$-synthesis. The resulting robust controller will then have optimal performance.

System identification, both in the time domain and frequency domain, is stated as a stochastic problem. This does not correspond with deterministic controller design techniques.

The following points are recommendations for future research:

- When considering identification for control, how should the model be weighted, such that the accuracy of the model is high around the bandwidth of the final control loop?

- When considering identification for robust control, how can the noise and the uncertainty be separated? For robust control, an uncertainty model is needed. The uncertainty should be as small as possible, but it should also account for all model errors present. Obtaining an uncertainty model is difficult if also noise is present in the measurements.

- Is it possible, in practice, to suppress flexible modes of a system by controller input? A proof of principle, if possible, on a real system with a $H_\infty$ robust controller is needed.
Bibliography


