MASTER

Inversion-based MIMO feedforward design beyond rigid body systems

Lunenburg, J.J.M.

Award date:
2010

Inversion-Based MIMO Feedforward Design Beyond Rigid Body Systems

J.J.M. Lunenburg

CST Report 2010.061

Report APT536-10-0422

Master of Science Thesis

Committee: Ir. G.E. van Baars†
Prof. ir. O.H. Bosgra‡ (coach)
Dr. ir. D.J.H. Bruijnen† (coach)
Dr. ir. M.F. Heertjes‡♮
Prof. dr. ir. M. Steinbuch‡
Dr. ir. M.M.J. van de Wal♮ (coach)

† Philips Applied Technologies
Mechatronics Program
Drives and Control Group

‡ Eindhoven University of Technology
Department of Mechanical Engineering
Control Systems Technology Group

♮ ASML
Research Mechatronics

Eindhoven, September 2010

Inversion-Based MIMO Feedforward Design Beyond Rigid Body Systems
Janno Johan Maria Lunenburg

Master of Science Thesis

Philips Applied Technologies
Mechatronics Program
Drives and Control Group

Eindhoven University of Technology
Department of Mechanical Engineering
Control Systems Technology Group

Summary

At Philips Applied Technologies, research is carried out on the motion control of wafer scanners. A wafer scanner performs the lithography process within the production of integrated circuits (ICs). Due to the fine patterns on the ICs and the high throughput requirements, extremely high positioning accuracy as well as very aggressive motion profiles are demanded.

In the near future, lightweight stage construction becomes crucial to avoid an infeasible design for actuators and amplifiers. As a result, these stages will be less stiff than traditional designs and will have more pronounced resonant dynamics. Traditional feedforward design methods do not take these resonant dynamics explicitly into account or do not lead to satisfactory results. The goal of this research is to develop an inversion based feedforward controller design method which deals with the more pronounced resonant dynamics of future wafer stages in an effective manner and which is robust for model uncertainties.

Before developing a feedforward controller design method, inverse systems are investigated, since a feedforward filter usually is an approximation of the inverse system model. Special attention is given to non-minimum phase systems, since these have unstable inverses. Furthermore, the initial conditions of the plant are shown to have a substantial effect on the achievable servo performance. Commonly, model uncertainty is addressed by optimizing the worst case performance in a model set. In this thesis, feedforward is optimized for the entire model set instead of the worst case.

The lifted system representation proves to be a useful system description in the design and analysis of feedforward. The inverse of a convolution matrix can be approximated by the Moore-Penrose generalized inverse. Using the lifted system description, it can be shown that non-minimum phase systems cannot track an arbitrary reference trajectory, unless the plant starts with suitable initial conditions. Pre-actuation can be used to bring the system into the correct initial conditions, but this introduces a servo error during the pre-actuation interval.

Using this knowledge, it is shown how, regardless of the presence of non-minimum phase zeros, an inverse state-space system can be used for feedforward control by decomposing the inverse system into a stable and an unstable part. Subsequently, the boundary conditions for the stable part are defined at the start of the interval under consideration and the boundary conditions for the unstable part at the end of this interval. This method is called stable inversion.

This feedforward design method is subsequently applied to the NXT-A7 wafer stage. It is

shown that stable inversion feedforward results in a smaller transient error and less oscillatory behavior than velocity and acceleration feedforward and that the high-frequent energy content of the servo error is significantly reduced. Applying sufficient pre- and post-actuation, nevertheless, is a necessity. To further improve the servo performance, more accurate models of the low-frequent behavior of the system are required. Robust stable inversion remains a subject to further research: the selection of the nominal and uncertainty model in this research were shown to be too conservative, hence leading to poor servo performance.

Contents

Nomenclature

Roman Symbols

A	state matrix
B	input matrix
C	output matrix / controller
D	direct feedthrough matrix
F	force / convolution matrix of feedforward filter
I	rotational inertia
J	convolution matrix
M	matrix with basis functions
P	plant
P_o	nominal model
R	Toeplitz matrix containing reference trajectory
S_{ff}	feedforward sensitivity
S_o	output sensitivity
S_r	reference sensitivity
T	transformation matrix
T_s	sampling frequency
U	matrix containing output singular vectors
V	matrix containing input singular vectors
a	acceleration
d_i	damping constant
e	servo error
f	frequency
j	jerk / imaginary unit
k	gain / sample instant
k_i	spring constant
l	length
m	mass
n	number
p_i	pole
r	position reference / radius
r_a	acceleration reference
s	s-operator / snap

t	time
u	input signal
v	velocity
x	system state / displacement
y	output signal
z	z-operator
z_i	zero

Greek Symbols

Δ	difference / variation / uncertainty
Σ	diagonal matrix containing singular values
δ	difference / variation / uncertainty
θ	angle
κ	condition number
λ	eigenvalue
ω	radial frequency
ϕ	rotation
ρ	relative degree
σ	singular value

Other Symbols

$\mathcal{P}$	model set

Subscripts

0	initial condition
c	causal
c	continuous
d	discrete
dec	decoupled
eq	equivalent
f	end condition / filtered
fa	acceleration feedforward
fb	feedback
ff	feedforward
fv	velocity feedforward
i	inverse / index number
j	index number
nc	non-causal

nmp	non-minimum phase
o	nominal /output
p	perturbed / pole
pre	pre- (actuation)
post	post- (actuation)
r	reduced
s	stable / sampling
u	unstable
z	zero

Superscripts

$*$	complex conjugate
$\dagger$	generalized inverse
z	zero

Abbreviations and Acronyms

BRB	Beyond Rigid Body
COG	Center-Of-Gravity
DOF	Degree-Of-Freedom
FF	Feedforward
FIR	Finite Impulse Response
FRF	Frequency Response Function
IC	Integrated Circuit
ILC	Iterative Learning Control
LHP	Left-Half-Plane
LTI	Linear Time Invariant
LTV	Linear Time Varying
MIL	Machine-In-the-Loop
MIMO	Multi-Input Multi-Output
MP	Minimum Phase
MSD	Mass-Spring-Damper
NMP	Non-Minimum Phase
PI	Proportional - Integral
PSD	Power Spectral Density
RHP	Right-Half-Plane
SISO	Single-Input Single-Output
SVD	Singular Value Decomposition
ZOH	Zero-Order-Hold

CHAPTER 1

Introduction

1.1 Background

At Philips Applied Technologies, research is carried out on the motion control of wafer scanners. A wafer scanner performs the lithography process within the production of integrated circuits (ICs). The goal of this process is to project a certain image on the light-sensitive top of a silicon disc, see Figure 1.1. Ultra-violet light (a) passes through the reticle, which is a mask containing the image, and imaging optics (c) onto the silicon disc, also called a wafer. The wafer and reticle are both placed on motion stages (b and d), that position them with respect to the imaging optics. Due to the fine patterns on the ICs, extremely high position accuracy is required: the servo errors of the wafer stage are in the order of magnitude of 1 nm. Next to accuracy, the stages are also required to position the wafer and reticle very quickly. Typical throughput requirements of 150 wafers per hour imply very aggressive motion profiles: velocities of 1 m/s and accelerations in the order of 100 m/s^2 are customary.

1.2 Motivation

The position control of motion systems generally consists of a feedforward and a feedback controller. While feedback control is mainly concerned with the (robust) stability of the closed loop system and disturbance rejection, feedforward control is essential to achieve the required settling performance: in Hennekens (2009) it is argued that, during maximum acceleration, feedforward is responsible for approximately 99.97% of the total control signal for wafer stages. In Lunenburg (2009) a number of commonly used feedforward methods are discussed. In current wafer stages, low order mass and snap feedforward (Boerlage, 2006; Boerlage et al., 2004) is used to handle the rigid body behavior and the low-frequent behavior of the flexible modes. It is assumed that these flexible modes appear only at high frequencies and are therefore not excited by the reference trajectory. The servo performance of this method can be optimized by means of Machine-In-the-Loop (MIL) control optimization (Van der Meulen, 2005b; Van der Meulen et al., 2008). Further improvements can be achieved by adding Finite Impulse Response (FIR) filters (Baggen et al., 2008; Heertjes and Van de Molengraft, 2009; Hennekens, 2009).

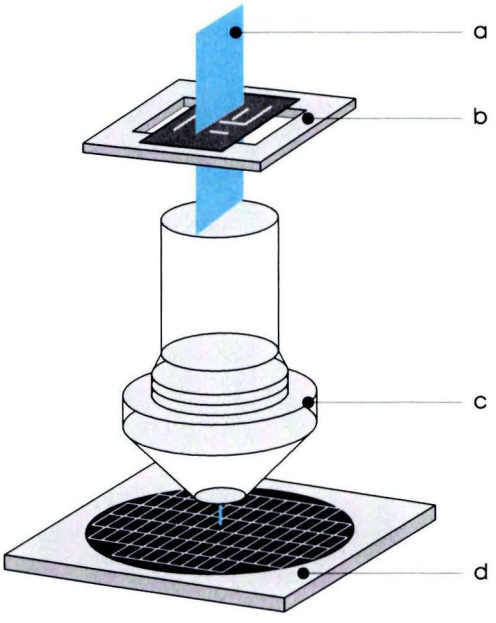


Figure 1.1: Schematic layout of the lithography process, where a: ultra-violet light, b: reticle stage with reticle, c: imaging optics and d: wafer stage with wafer.

In the near future, lightweight stage construction becomes crucial to avoid an infeasible design for actuators and amplifiers. Furthermore, increasing throughput requirements demand for bigger stages, hence the wafer size is bound to increase from 300 mm to 450 mm. As a result, these stages will be less stiff than traditional designs and will therefore display more pronounced resonant dynamics. In summary, future lightweight wafer stages will become less stiff and hence resonances will occur at lower frequencies, while increasingly aggressive motion profiles have more energy content at higher frequency. Contrary to the current method described above, which essentially treats resonant behavior merely as parasitic, this asks for a feedforward method that effectively deals with these flexibilities. Within Philips Applied Technologies, this is referred to as Beyond Rigid Body (BRB) control.

Iterative Learning Control (ILC) is a MIL optimization method which iteratively updates the feedforward *signal* such that the servo error for a particular reference trajectory decreases (Bristow et al., 2006; Van de Wijdeven, 2008). Recently, ILC is often implemented using the lifted system representation, which provides large design flexibility and superior analysis properties (Van der Meulen, 2005a). Although ILC can lead to excellent servo performance for BRB systems, the fact that a new feedforward signal has to be 'learned' for every reference trajectory is a major drawback.

In theory, exact nominal tracking is achieved if the inverse of the plant is used as a feedforward controller. The use of the inverse, however, is generally assumed to be of limited practical use: firstly, motion stages are strictly proper and hence result in an improper feedforward filter. In discrete time, this implies a non-causal filter. In Lunenburg (2009), however, it is already shown that this can be easily solved by adding a certain amount of delay to the filter and delaying the reference trajectory with the corresponding number of samples. Secondly, the presence of non-minimum phase transmission zeros leads to an unstable feedforward filter and, in general, to an unbounded feedforward signal which cannot be implemented in practice.

However, by suitably addressing this issue, a useful feedforward signal may be obtained, as will be shown.

A third remark concerning the use of the inverse of the plant for feedforward control is that a plant model never represents the true system exactly. The previously mentioned MIL methods are able to handle model uncertainty. Furthermore, the general plant setup, which is commonly used in combination with $\mathcal{H}_\infty$ or $\mathcal{H}_2$ optimization is very well suited for dealing with model uncertainty. Although very good results are obtained with $\mathcal{H}_\infty$ or $\mathcal{H}_2$ feedback control, the result is often not satisfactory for feedforward control: since the feedforward controller essentially is a trade-off over the entire frequency range, a low-frequent mismatch between feedforward filter and inverse system introduces a significant servo error (Van de Wal, 2001). Furthermore, order reduction of the controller to meet implementation restrictions deteriorates the performance. In practice, a nominal $\mathcal{H}_\infty$ or $\mathcal{H}_2$ feedforward controller often outperforms the robust solution.

1.3 Overview of the Research

From the issues mentioned in the previous section, the problem statement of this research is formulated as:

Develop an inversion-based feedforward controller design method that deals with the more pronounced resonant dynamics of future wafer stages in an effective manner and that is robust for model uncertainties.

With respect to this problem statement, a number of subproblems are identified.

Q1) In order to understand how to address certain properties of (inverse) systems and to be able to interpret results, it is important to gain physical insight in (inverse) systems.

Inverting a system can be viewed as interchanging the poles and zeros, indicating their importance in feedforward design. Particularly non-minimum phase zeros are important, since these result in unstable inverse systems. Computing a feedforward signal from an inverse system essentially means solving a differential equation. Hereto, the initial conditions are often tacitly assumed to be zero. However, by choosing other boundary conditions completely different results are obtained. Before that, however, it is important to understand what the states of inverse systems actually represent. For these reasons, Q1 is further split into the following questions:

Q1a) What do the states of an inverse system represent?
Q1b) How can poles and zeros be physically interpreted?
Q1c) What is the (physical) explanation for the fact that systems with non-minimum phase zeros are difficult to control in a feedforward sense?
Q1d) How do initial conditions affect this?

Next to these questions it is important to realize that the plant model never exactly resembles the true plant. It is essential to account for this in feedforward control, hence:

Q2) What is a suitable way of dealing with model uncertainty in inversion-based feedforward control?

To be able to address uncertainty, it has to be quantified. Furthermore, it should be realized that a feedforward controller is usually not implemented in open loop but always accompanied by a (robust) feedback controller which partly attenuates the servo error introduced by a mismatch between model and true system. Finally, robust control commonly aims at optimizing the worst case performance. However, as mentioned in Skogestad and Postlethwaite (2005, p. 259), this does not always lead to optimal results, which suggests the following subquestions:

Q2a) What is the most suitable way to model uncertainty for feedforward control design?
Q2b) To what extent can feedback attenuate mismatches between the true inverse system and the feedforward controller?
Q2c) Is the common robust approach of optimizing the worst-case performance suitable for feedforward design?

By answering Q1 and Q2, it becomes clear what is important in feedforward control design and the gained knowledge is used to develop a design methodology for feedforward control. Both the rather conventional state-space system representation as well as the lifted system representation are investigated. A state-space solution is advantageous, since it can be easily implemented in the current software. As mentioned above the lifted system representation on the other hand, provides large design flexibility and excellent analysis possibilities and is therefore an interesting system description for feedforward control design.

Q3) Is it possible to use inverse state-space systems for feedforward control design, regardless of the presence of non-minimum phase zeros?
Q4) How can the lifted system representation be used for feedforward control design?

A good starting point for Q3 is stable inversion, see, *e.g.*, Devasia et al. (1996); Sogo (2002); Zou and Devasia (1999). A preliminary example using a simple minimum phase system for feedforward in the lifted system representation is discussed in Lunenburg (2009), but a general approach has not been provided yet. Furthermore, addressing uncertainty is an open research question in both approaches.

1.4 Outline of the Report

This report is organized as follows. In Chapter 2 it is recapitulated why inverting either the plant, the open loop or the closed loop for feedforward control design can in theory lead to exact tracking. Furthermore, the difference between SISO and MIMO systems is addressed and some concepts are defined which are used throughout this report.

Next, item Q1 is addressed in Chapter 3, followed by item Q2 in Chapter 4. The gathered knowledge is subsequently used in Chapters 5 and 6 to develop feedforward control methods using the lifted system representation and the state-space description.

Thereafter, the NXT-A7 wafer stage setup is introduced in Chapter 7, which will be used to validate the developed theories. The results of SISO experiments are discussed in Chapter 8 and a model of this wafer stage is used to apply the developed feedforward methods in MIMO simulations (Chapter 9).

Finally, conclusions and recommendations for further research are given in Chapter 10.

CHAPTER 2

The Use of Inverses in Feedforward

In this chapter, the use of inverse systems in feedforward control is discussed and some definitions are given that are used throughout this thesis. As is shown in Lunenburg (2009), using the plant inverse for feedforward control leads to exact tracking if a standard 2DOF control architecture is used. In Section 2.1 it is shown that, in theory, exact tracking can also be achieved by using the inverse of either the open loop or the closed loop if a different control architecture is used. This holds for both SISO as well as MIMO systems. However, inverting MIMO systems is in general not as easy and less intuitive than inverting SISO systems. This is explained in Section 2.2. Another issue arising when inverting systems is the relative degree, which is therefore discussed in Section 2.3.

2.1 Feedforward Architectures

In theory, exact tracking can be achieved by using either the inverse plant, the inverse open loop or the inverse closed loop for feedforward control in different control architectures. This can be explained by determining the transfer functions from the reference r to the servo error e (the arguments s and t or z and k are omitted to enhance readability).

In Figure 2.1, the standard 2DOF control architecture is displayed. In this case, the transfer function from reference trajectory r to tracking error e is given by:

$$e = (I + PC_{\text{fb}})^{-1} (I - PC_{\text{ff}})\, r \tag{2.1}$$

where P denotes the plant, C_{fb} the feedback controller and C_{ff} the feedforward controller. In this configuration, the error e is zero if $C_{\text{ff}} = P^{-1}$. Note the difference with Iterative Learning Control: an ILC signal is usually also injected between the feedback controller and the plant. In that case the inverse process sensitivity is used to update the ILC signal. However, (2.1) shows why the inverse plant instead of the inverse process sensitivity should be used for feedforward control.

In literature, architectures different from the one in Figure 2.1 are sometimes used. A slightly different configuration is displayed in Figure 2.2. Here, e is given by:

$$e = (I + PC_{\text{fb}})^{-1} (I - PC_{\text{fb}}C_{\text{ff}})\, r \tag{2.2}$$

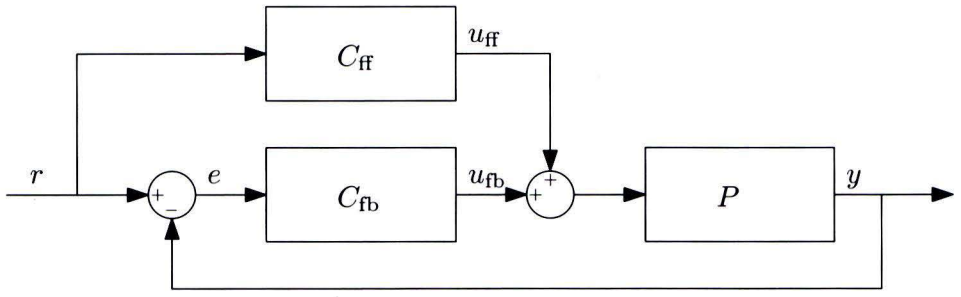


Figure 2.1: Plant inversion feedforward.

where exact tracking is obtained when $C_{\text{ff}} = (PC_{\text{fb}})^{-1}$, meaning that the feedforward controller equals the inverse of the open loop. This configuration, however, is not very common for feedforward control design.

It is also possible to use the inverse of the closed loop as C_{ff}. Hereto, the configuration in Figure 2.3 is used. Note that in this case, the input to C_{fb} does *not* equal the servo error. The error can be expressed as:

$$e = \left(I - (I + PC_{\text{fb}})^{-1} PC_{\text{fb}}C_{\text{ff}}\right) r \tag{2.3}$$

Exact tracking results if $C_{\text{ff}} = \left((I + PC_{\text{fb}})^{-1} PC_{\text{fb}}\right)^{-1}$ which boils down to inverting the closed loop. This architecture is more common than open loop inversion and is used in, *e.g.*, Butterworth et al. (2008); Tomizuka (1987). In Rigney et al. (2006) plant inversion and closed loop inversion are compared for settle time applications with uncertainty. In this thesis, however, only the first architecture will be considered since this is the only one that is independent of the feedback controller.

The relation in (2.1) denotes the closed loop transfer from reference trajectory to servo error. This makes it a suitable measure to assess the effect of the feedforward controller in the frequency domain.

Definition 2.1 *The reference sensitivity S_r is the* closed *loop transfer from reference to error:*

$$S_r = (I + PC_{\text{fb}})^{-1} (I - PC_{\text{ff}}) = S_o S_{\text{ff}} \tag{2.4}$$

where S_o denotes the output sensitivity.

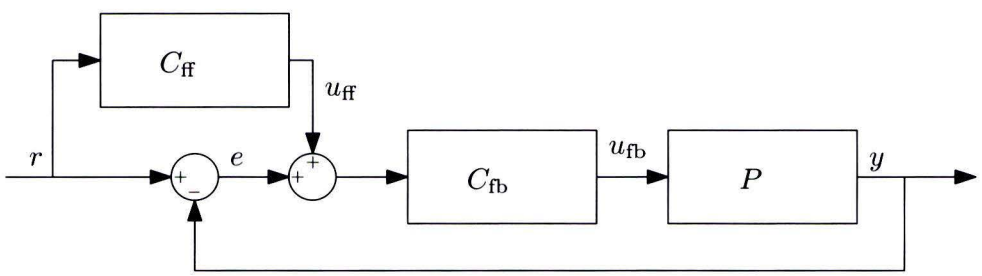


Figure 2.2: Open loop inversion feedforward.

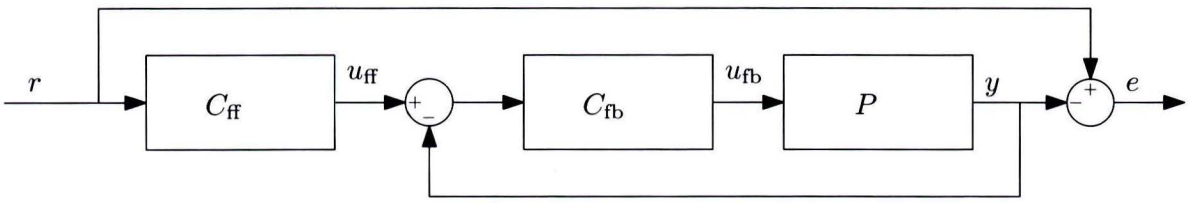


Figure 2.3: Closed loop inversion feedforward.

This definition, however, does not depend solely on feedforward but also on feedback. In open loop, the following definition can be given:

Definition 2.2 *The feedforward sensitivity S_{ff} is the* open *loop transfer from reference to error:*

$$S_{\text{ff}} = 1 - PC_{\text{ff}} \tag{2.5}$$

Hence, S_{ff} indicates the effect of feedforward. Ideally, $S_{\text{ff}}(j\omega) = 0\ \forall\omega$, but this is hardly possible in practice. A mismatch between $P^{-1}(j\omega)$ and $C_{\text{ff}}(j\omega)$ can be partly attenuated by the feedback controller. This is also taken into account in $S_r(j\omega)$. Since feedback controllers usually contain integral action, $\lim_{\omega\to 0} S_o(j\omega) = 0$ and hence $S_r(j\omega)$ will also approach zero at low frequencies. Note that the concepts of feedforward sensitivity (2.4) and reference sensitivity (2.5) are not used consistently in literature, see, *e.g.*, Hennekens (2009); Peeters et al. (2000); Skogestad and Postlethwaite (2005).

The control architectures discussed in this section can be used for both SISO and MIMO systems. However, inverting MIMO systems is in general not as easy and less intuitive than inverting SISO systems. This is addressed in the next section.

2.2 SISO and MIMO Systems

The flexible behavior of the next generation wafer stages inherently introduces multivariable behavior. In a rigid body approach, the various degrees of freedom (DOFs) are statically decoupled and feedforward is subsequently applied separately in each DOF.

In Figure 2.5 the Bode diagrams from F_1 to y_2 of a 2 Mass-Spring-Damper system (see Figure 2.4) and its inverse are plotted. The magnitude and phase of the plant and its inverse are simply related by:

$$|P^{-1}(j\omega)| = \quad |P(j\omega)|^{-1} \tag{2.6a}$$

$$\angle(P^{-1}(j\omega)) = \quad -\angle(P(j\omega)) \tag{2.6b}$$

The low-frequent -2 slope corresponding to the rigid body mode of a motion system hence appears as a $+2$ slope in the inverse system. This implies that there is no force required to hold the system at rest at a certain position. However, as can be seen in Figure 2.6, this does not hold for a 2×2 MIMO 2 MSD system.

All entries of the MIMO 2 MSD system have low-frequent -2 slopes, but the inverse system shows 0 slopes. This can be explained by the fact that the spring is applying a force on both

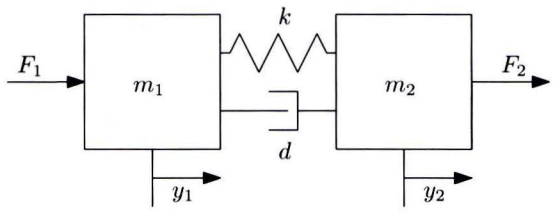


Figure 2.4: Schematic 2 Mass-Spring-Damper System.

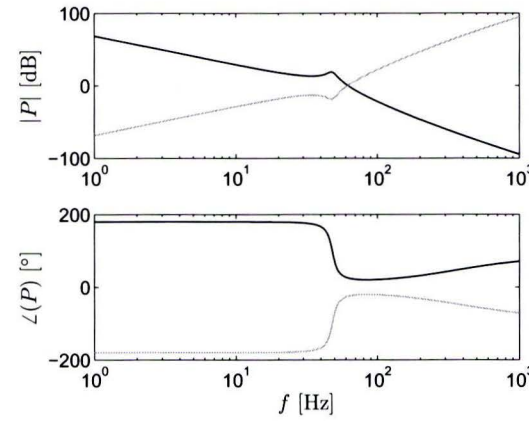


Figure 2.5: Bode diagram of a SISO 2 MSD System (black) and its inverse (gray).

m_1 as well as m_2 whilst holding the system at a position $y_1 \neq y_2$. In order for the system to stay at rest, this force has to be compensated by both F_1 and F_2, hence $\lim_{\omega \to 0} P_{i,j}^{-1}(j\omega) \neq 0$.

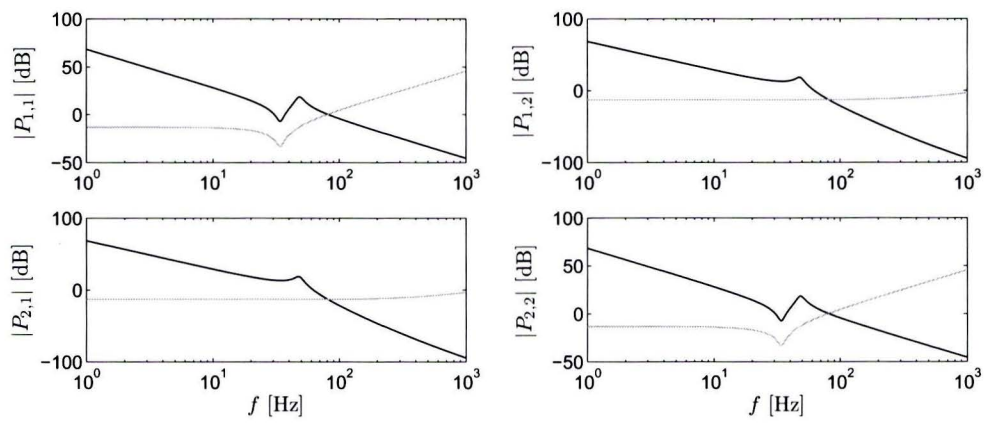


Figure 2.6: Bode magnitude diagram of a MIMO 2 Mass-Spring-Damper System (black) and its inverse (gray).

2.2.1 Decoupling

In order to enable the possibility to synthesize diagonal feedforward and feedback controllers and thus significantly simplifying control design, the 2×2 2MSD system can be decoupled into a rigid body mode and a flexible mode. The resulting Bode-diagram is plotted in Figure 2.7, where also the inverse system is shown. It appears that on the diagonal entries, the relations in (2.6a) and (2.6b) hold. The off-diagonal entries are zero for both P_{dec} and P_{dec}^{-1}, hence no MIMO feedforward design is required for this particular example.

However, the 2MSD system has two inputs and two outputs, but only one rigid body mode. This is different from many practical applications such as wafer stages, that usually have

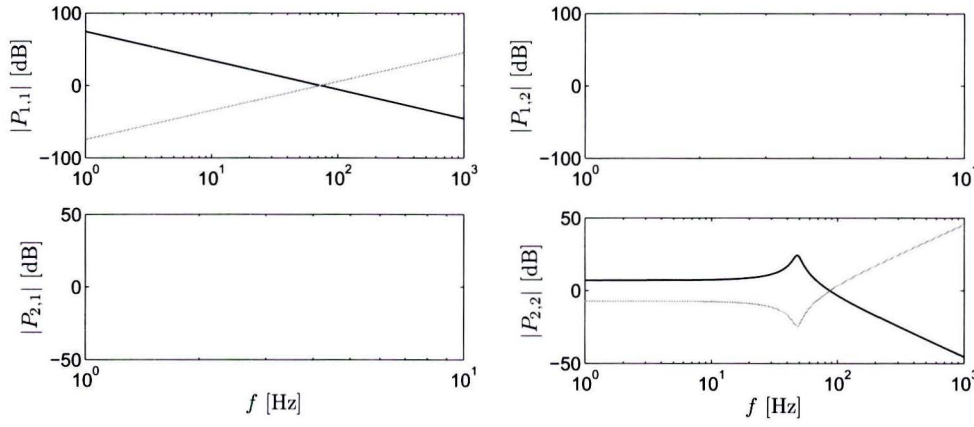


Figure 2.7: Bode magnitude diagram of a decoupled MIMO 2 Mass-Spring-Damper System (black) and its inverse (gray).

an equal number of rigid body modes and DOFs. Thus, wafer stages are usually statically decoupled according to their rigid body modes. This is in general not sufficient to make the off-diagonal entries equal to zero. Take, *e.g.*, the lumped mass model of an H-bridge in Figure 2.8. The original model is rigid body decoupled, resulting in:

$$\begin{bmatrix} y \\ \phi \end{bmatrix} = P \begin{bmatrix} F_y \\ T_z \end{bmatrix} \tag{2.7}$$

The corresponding Bode diagram is plotted in Figure 2.9. At low frequencies, the rigid body behavior is dominant so the diagonal entries of $P(j\omega)$ are much larger than the off-diagonal entries. This also holds for $P_{1,1}^{-1}(j\omega)$ and $P_{2,2}^{-1}(j\omega)$, which are much larger than $P_{1,2}^{-1}(j\omega)$ and $P_{2,1}^{-1}(j\omega)$ at low frequencies. However, at higher frequencies, the inexact decoupling due to flexibilities causes the off-diagonal entries to have the same order of magnitude as the diagonal entries, hence requiring multivariable feedforward control.

In present feedforward control designs, MIMO FIR filters are tuned to address the high-frequent multivariable behavior (Baggen et al., 2008; Heertjes and Van de Molengraft, 2009; Hennekens, 2009). Nevertheless, future wafer stages will have flexibilities at lower frequencies and the current FIR filters will not be able to compensate for this if the inverse plant possesses resonances. Furthermore, FIR tuning does not give any insight in what is physically happening and is not useful in case of inferential control. Therefore, a fully multivariable inversion-based feedforward control design is required for BRB control design.

In the next section, the relative degree and properness for SISO and MIMO systems are discussed.

2.3 The Relative Degree

The relative degree of a SISO system is defined as the number of poles n_p minus the number of zeros n_z of a system, *i.e.*,

$$\rho = n_p - n_z \tag{2.8}$$

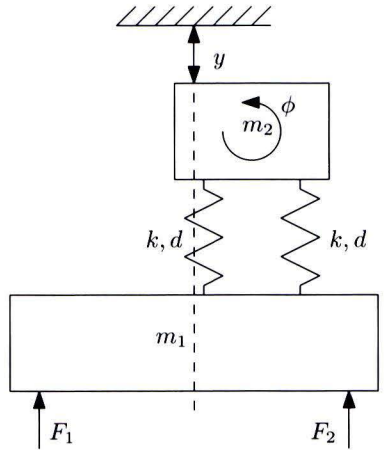


Figure 2.8: Lumped mass model of an H-Bridge.

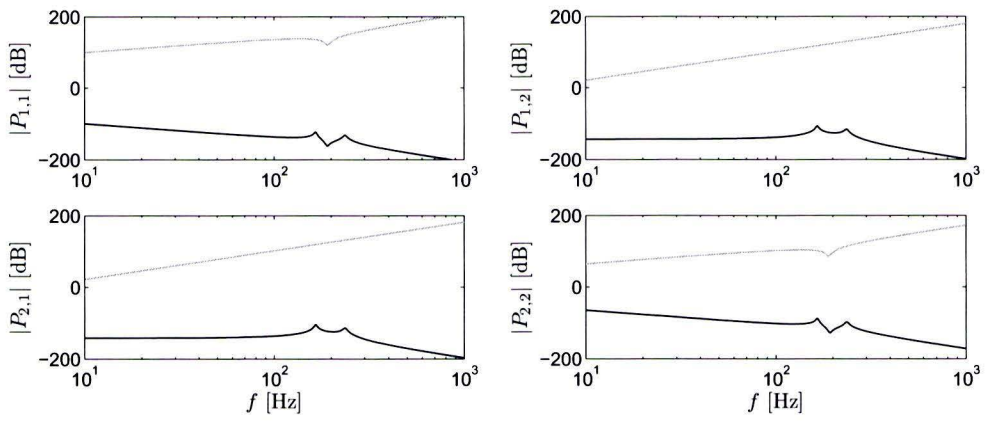


Figure 2.9: Bode magnitude diagram of a H-bridge (black) and its inverse (gray).

In discrete time, causality and properness of a system are directly related:

Definition 2.3

- *A discrete system $P(z)$ is* strictly proper, *or strictly causal, if the output depends only on past inputs,* i.e., $\rho > 0$;
- *A discrete system $P(z)$ is* bi-proper *if the output depends on both past and present inputs,* i.e., $\rho = 0$;
- *A discrete system $P(z)$ is* proper, *or causal, if it is either strictly proper or proper,* i.e., $\rho \geq 0$;
- *A discrete system $P(z)$ is* improper, *or a-causal (non-causal), if the output also depends on future inputs,* i.e., $\rho < 0$.

In literature, it is often argued that a system is only invertible if it is bi-proper, since inverting a strictly proper system results in an improper system. This is generally not true: although an improper system cannot be described in common state-space, a descriptor state-space system (Luenberger, 1977) or transfer function matrix can be used instead. In practice, however, an improper inverse system cannot be implemented as a causal feedforward filter.

In continuous time, an improper inverse system with relative degree $-\rho$ can be made proper by multiplying it by $\frac{1}{s^\rho}$ and subsequently $\frac{d^\rho r(t)}{dt^\rho}$ should be used to calculate $u_{\text{ff}}(t)$. This corresponds to inverting a system according to Silverman (1969), where the inverse system is viewed as a bank of differentiators followed by a proper dynamic system.

In discrete time, an improper system is non-causal. Hence, an improper inverse system can be made proper by multiplication by $\frac{1}{z^\rho}$, in which case $r(k)$ can still be used to calculate $u_{\text{ff}}(k)$ but should be delayed by ρ samples before being fed into the feedback loop (Lunenburg, 2009).

In a MIMO system, the relative degree is not defined as the number of poles minus the number of transmission zeros. Instead, a MIMO system has a vector relative degree $[\rho_1, \ldots, \rho_p]$, with p the number of outputs and ρ_i the number of times one has to differentiate the output $y_i(t)$ in order to have at least one component of the input vector explicitly appearing (Isidori, 1995).

The properness of a MIMO system depends on the individual entries of the system. According to Zhang et al. (2006), a MIMO system is proper if all of its elements are proper and a system is strictly proper if all its elements are strictly proper.

In the next chapter, the states of (inverse) systems and the physical interpretation of poles and zeros are discussed.

CHAPTER 3

States, Poles and Zeros

In this chapter, a number of properties of systems and their inverses are discussed. To begin with, the states of the original and the inverted system are discussed in Section 3.1. Next, the mathematical description as well as the physical interpretation of poles and zeros is discussed. If a SISO system is represented by a gain k and a number of poles and zeros, the inverse can be described by inverting k and interchanging the poles and zeros. This illustrates the importance of poles and zeros in feedforward. It also implies that a system containing non-minimum phase (NMP) zeros has an unstable inverse. A physical explanation for this is discussed in Section 3.4, followed by the implications of the initial conditions of the system and its inverse. Finally, the zeros due to the sampling of a continuous time system are discussed in Section 3.5.

3.1 The States of the Inverse System

If a system is physically modeled, the states of a state-space system have a straightforward interpretation: in case of mechanical systems, these usually express the (angular) positions and velocities of components of the system. A question that arises while working with inverse systems is how to interpret the states of the inverse system.

Consider the bi-proper dynamic system:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \quad x(0) = x_0 \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{3.1}$$

with $x(t) \in \mathbb{R}^n$ the states of the system and $u(t), y(t) \in \mathbb{R}^p$ the inputs and outputs of the system. The inverse of this system is given by (Lunenburg, 2009):

$$\begin{aligned} \dot{x}(t) &= \left(A - BD^{-1}C\right)x(t) + BD^{-1}y(t), \quad x(0) = x_0 \\ u(t) &= -D^{-1}Cx(t) + D^{-1}y(t) \end{aligned} \tag{3.2}$$

Essentially, only the inputs and outputs are interchanged and the states remain the same, which implies that the states of the inverse system represent the same quantities as for the original system. If D is rank deficient, Silverman (1968, 1969) can be used, in which case the states of the resulting inverse system are still the same.

If a continuous time system is viewed in combination with ZOH and sampler, the states of the original system and the resulting discrete system are the same. This follows directly from the definition of the matrices of the discrete system: while the discrete state and input matrices (with a sampling period T_s) are given by (Franklin et al., 2002):

$$A_d = e^{A_c T_s} \tag{3.3}$$

$$B_d = \left(\int_0^{T_s} e^{A_c \eta} d\eta \right) B_c \tag{3.4}$$

the C and D matrices of the discrete system are equal to those of the continuous system. Furthermore, a similar reasoning as with continuous time systems can be made: the states of the inverse system equal those of the original system. This can be illustrated by an example:

Example 3.1 *As an example, a system consisting of a mass ($m = 1$ kg) connected to a wall by means of a spring ($k = 100$ N/m) and damper ($d = 0.5$ Ns/m) is used with a second order reference trajectory. The inverse system is used as the feedforward controller. In Figures 3.1 and 3.2 the states of the inverse and the original system are shown, which are exactly the same.*

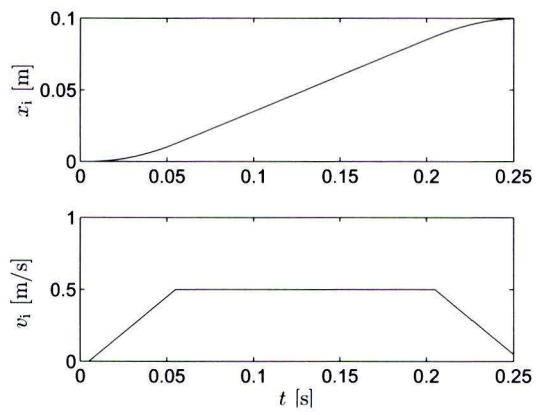


Figure 3.1: Position (upper plot) and velocity (lower plot) of the inverse system. These exactly match the position and velocity of the original system (see Figure 3.2).

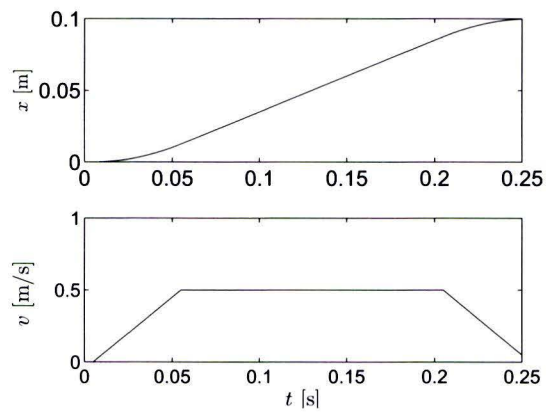


Figure 3.2: Position (upper plot) and velocity (lower plot) of the mass-spring-damper system.

In practice, a plant model is often obtained by system identification rather than by physical modeling. Therefore, it is more difficult to assign a physical measure to the states. However, via similarity transformations it is always possible to obtain a modal representation where the states represent velocities and positions. Therefore, it can be concluded that it is not desired to directly use unstable feedforward filters, since these may result in unbounded velocities and/or positions of the system to be controlled.

3.2 Poles

In Skogestad and Postlethwaite (2005), poles are loosely defined as the finite values $s = p_i$ (for continuous systems) or $z = p_i$ (for discrete systems) where $P(p_i)$ has a singularity ('is infinite'). If a plant is represented by a minimal state-space system, the poles p_i can be computed as the eigenvalues λ_i of the state matrix A. The locations of poles in the complex plane solely depend on the plant to be controlled, *e.g.*, mass, stiffness and damping characteristics. In the lifted system description, unstable poles turn up as very large singular values of the convolution matrix, see Appendix B.

In case of a MIMO transfer function matrix, the multivariable poles essentially are the poles of the various entries of the transfer function matrix, although the multiplicity of the poles cannot be determined by looking at the individual entries. In case of a state-space description, the poles are still the eigenvalues of the state matrix. Next to a location, multivariable poles also have directions associated with them. Using

$$P(p_i)\bar{u}_{p_i} = \infty \bar{y}_{p_i} \tag{3.5}$$

$\bar{u}_{p_i}$ is defined as the input pole direction and $\bar{y}_{p_i}$ is defined as the output pole direction of the pole located at $s = p_i$ (Skogestad and Postlethwaite, 2005).

The poles of a system are associated with the eigenmodes (natural modes) of the system. The motion systems considered throughout this thesis typically contain a rigid body mode for every degree-of-freedom (DOF), which implies having two poles at $s = 0$ (continuous time) or $z = 1$ (discrete time). Furthermore, systems usually contain a number of flexible modes (resonance modes).

Example 3.2 *Consider the system in Figure 3.3. This system has one rigid body mode, in which the two masses move as if they were connected through a rigid rod. The flexible mode corresponds to the masses moving in opposite directions. The frequency and damping of this mode depend on the values for m_1, m_2, k and d and are directly related to the locations of the poles in the complex plane.*

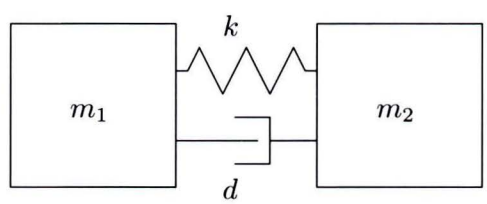


Figure 3.3: A 2 mass-spring-damper system.

3.3 Zeros

Corresponding to the loose definition of poles in the previous section, zeros can be defined as the values $s = z_i$ (continuous time) or $z = z_i$ (discrete time) where $P(z_i)$ loses rank[1]. Unlike

[1]Note that z (without subscript) denotes the z-operator, while z_i (with subscript) denotes a zero in either continuous or discrete time.

poles, the values of zeros vary mainly according to the locations of the sensors and actuators (Miu, 1991). Only if sensors and actuators introduce mass and additional dynamics, the A matrix and therefore the values of the poles may very according to sensor and actuator locations. If the state-space equations of a dynamical system are written as:

$$\begin{bmatrix} sI - A & -B \\ C & D \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} \tag{3.6}$$

then the zeros can be computed as the values for which the matrix in (3.6) loses rank. In the lifted system description, NMP zeros turn up as very small singular values of the convolution matrix, see Appendix B.

Contrary to poles, the zeros of a MIMO system cannot be computed by looking at the individual entries of a transfer function matrix, as will be illustrated by an example (Skogestad and Postlethwaite, 2005, Section 4.5).

Example 3.3 *Consider the 2×2 system given by the transfer function matrix:*

$$P(s) = \frac{1}{s+2} \begin{bmatrix} s-1 & 4 \\ 4.5 & 2(s-1) \end{bmatrix} \tag{3.7}$$

In this system, two individual entries have a zero at $s = 1$. However, substituting $s = 1$ in (3.7) does not cause the matrix to lose rank. On the other hand, substituting $s = 4$ does cause the matrix to lose rank, hence $s = 4$ actually is a multivariable zero.

As is the case with MIMO poles, multivariable zeros also have directions associated with them. In this case, they are given by (Skogestad and Postlethwaite, 2005):

$$P(z_i)\bar{u}_{z_i} = 0\bar{y}_{z_i} \tag{3.8}$$

$\bar{u}_{z_i} \neq 0$ is defined as the input pole direction and $\bar{y}_{z_i} \neq 0$ is defined as the output pole direction of the pole located at $s = z_i$.

In the time domain, the presence of zeros implies that certain input signals are blocked. More specifically, if $s = z_i$ is a zero of $P(s)$, then there exists an input signal of the form $ue^{z_i t}1_+(t)$ ($1_+(t)$ denotes a unit step) and initial conditions x_0 such that the output $y(t) = 0$ for $t > 0$.

A physical interpretation of zeros of mechanical systems is given in Miu (1991). Here, it is shown that the zeros of a system represent the resonances of the remaining sub-structure, when artificial constraints are put at the sensor/actuator location (see also Verhoeven (2010)). This indicates why the zeros of a system are essential for feedforward. In point-to-point moves, the output, *i.e.*, the sensor location, is supposed to stay at rest after the move. However, the remaining states of the system are not necessarily in rest, therefore requiring post-actuation.

Example 3.4 *Take for example the mass-spring-damper system from Figure 3.4. Now, a feedforward signal is applied such that y exactly tracks a point-to-point move. After arrival m_2 is at rest so $\dot{y} = 0$, but m_1 might still be moving. Typically, it will oscillate at the frequency of the zeros of the total system. In order for y to stay at rest, F should not only contain a steady state component but also compensate for the oscillating force of k_2 and d_2 on m_2.*

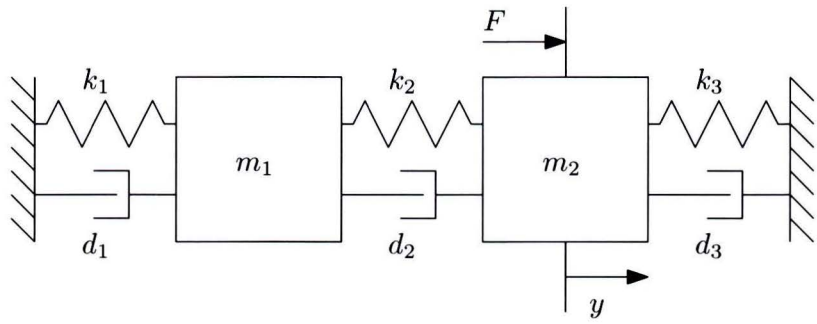


Figure 3.4: Mass-spring-damper system from an example in Miu (1991).

This can be further generalized: forces on m_2 due to possible oscillations of m_1 have to be compensated during any reference trajectory.

Remark 3.1 *From the examples in the previous and current section, it appears that, unlike the locations of the poles, the locations of the zeros depend on the placement of sensors and actuators.*

Unlike unstable poles, unstable zeros (also called non-minimum phase zeros) do not imply that a system is unstable. However, it is well known that they do pose limits on the achievable performance of feedback control, see, *e.g.*, Skogestad and Postlethwaite (2005). In feedforward control, unstable zeros have to be dealt with in a suitable way (Butterworth et al., 2008; Lunenburg, 2009; Tomizuka, 1987), since a NMP system has an unstable inverse. In the next section, it is discussed why the presence of unstable zeros limits servo performance if they are not addressed in a suitable way. Furthermore, special attention is given to sampling zeros in Section 3.5.

3.4 Non-Minimum Phase Behavior

In mechanical systems, NMP zeros may arise when sensors and actuators are placed on opposite sides of the center of gravity. Furthermore, since in practice digital controllers are used, additional NMP zeros may be introduced by sampling, see Åström et al. (1984). In order to illustrate the implications of NMP zeros, an exemplary system is introduced. Thereafter, inertia feedforward is used on a second order reference trajectory, which clearly shows the difference between minimum phase and non-minimum phase behavior.

3.4.1 The Flexible Cart System

Throughout this thesis, the flexible cart system shown in Figure 3.5 will often be used to illustrate the difference between minimum and non-minimum phase systems. It consists of a mass resting on two springs and dampers (not shown in the picture). A force u is applied in horizontal direction at the top of the mass, resulting in a displacement x and a rotation ϕ of the block. The displacement can be measured at the right bottom (y_1) or the right top (y_2); the difference between two systems will be discussed below. The dynamics of the system in

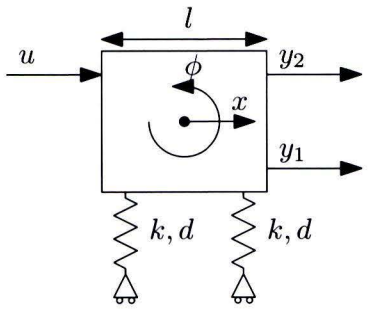


Figure 3.5: The Flexible Cart system.

Figure 3.5 can be described by the following equations:

$$m\ddot{x} = u \tag{3.9a}$$

$$I\ddot{\phi} = \frac{1}{2}lu - \frac{1}{2}kl^2\sin(\phi) - \frac{1}{2}dl^2\sin(\dot{\phi}) \tag{3.9b}$$

$$y_1 = x + \frac{l}{2}\sin(\phi) \tag{3.9c}$$

$$y_2 = x - \frac{l}{2}\sin\phi \tag{3.9d}$$

where k and d are the spring and damper constants, m is the mass of the block, and I is the mass moment of inertia. The various parameters are given in Table 3.1[2]. Linearizing the

Table 3.1: System parameters of the flexible cart.

Parameter	Value	Unit
m	8	kg
I	0.013	kgm^2
k	10	kN/m
d	333	Ns/m
l	0.1	m

equations of motion around $\phi = 0$ results in the following transfer functions:

$$P_1(s) = \frac{y_1(s)}{u(s)} = \frac{1}{ms^2} - \frac{l^2}{4\left(Is^2 + \frac{1}{2}dl^2s + \frac{1}{2}kl^2\right)} \tag{3.10a}$$

$$P_2(s) = \frac{y_2(s)}{u(s)} = \frac{1}{ms^2} + \frac{l^2}{4\left(Is^2 + \frac{1}{2}dl^2s + \frac{1}{2}kl^2\right)} \tag{3.10b}$$

Both systems $P_1(s)$ and $P_2(s)$ have two poles at the origin and poles at $s = -48.1$ and $s = -80.0$. Furthermore, $P_1(s)$ has a LHP zero at $s = -27.0$ and a RHP zero at $s = 265$ and is thus non-minimum phase. This is also visible in the Bode diagram in Figure 3.6, where $P_1(s)$ does not comply with the Bode gain-phase relationship. System $P_2(s)$ has both its zeros in the LHP at $s = -25.2 \pm 29.6j$ and thus is minimum phase.

[2]Note that in Lunenburg (2009) $d = 10$ Ns/m. However, the alternative value in this section is chosen such that the response is supercritically damped and hence the NMP behavior can be seen more clearly.

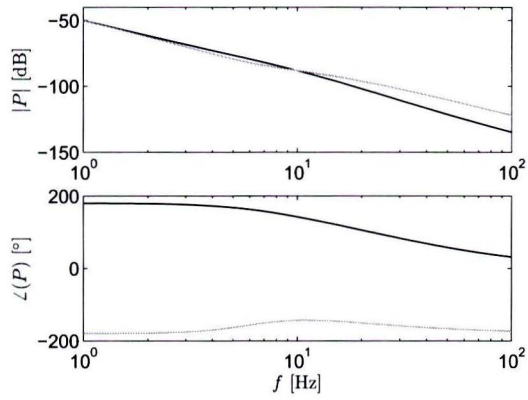


Figure 3.6: Bode-diagrams of $P_1(s)$ (black) and $P_2(s)$ (gray).

In a state-space description with $\boldsymbol{x} = \begin{bmatrix} x & \dot{x} & \phi & \dot{\phi} \end{bmatrix}^T$, matrices A and B are equal for P_1 and P_2:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{1}{2}\frac{kl^2}{I} & -\frac{1}{2}\frac{dl^2}{I} \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ \frac{1}{m} \\ 0 \\ \frac{1}{2}\frac{l}{I} \end{bmatrix} \tag{3.11}$$

The difference appears in the C matrices:

$$C_1 = \begin{bmatrix} 1 & 0 & \frac{l}{2} & 0 \end{bmatrix} \tag{3.12a}$$

$$C_2 = \begin{bmatrix} 1 & 0 & -\frac{l}{2} & 0 \end{bmatrix} \tag{3.12b}$$

and in both cases $D = 0$.

To assess the sampled data system, the MATLAB command C2D is used to obtain the system including ZOH and sampler. Hereto, a sampling time of $T_s = 0.005$ s is used.

The resulting discrete time system has two poles at $z = 1$ and poles at $z = 0.786$ and $z = 0.670$, where the poles at $z = 1$ correspond to the rigid body mode and the remaining poles with the rotation of the flexible cart. Furthermore the sampled minimum phase system $P_2(z)$ has zeros at $z = 0.872 \pm 0.130j$ and a sampling zero at $z = -0.879$ which are all located inside the unit circle. The non-minimum phase system $P_1(z)$ has its zeros at $z = 0.874$, $z = 3.87$ and the sampling zero at $z = -0.575$. The non-minimum phase behavior manifests itself here in the zero outside the unit circle. Note that by adding the ZOH and the sampler the discrete systems have three zeros instead of the two of the continuous time flexible cart systems. This zero is therefore also referred to as sampling zero (see Section 3.5).

3.4.2 Inertia Feedforward

To illustrate the difficulty in feedforward control for NMP plants, it is first studied what happens if a step input is applied to the flexible cart system. This step input corresponds to using mass feedforward for a second order reference trajectory. To ensure closed-loop stability, a low bandwidth feedback controller is used. In Figure 3.7, it is shown that the rotation ϕ

of the flexible cart system converges to steady state. As long as ϕ is constant, the system essentially behaves as a rigid body. As a result of this rotation, y_1 lags behind the reference trajectory and y_2 on the other hand leads in front of the reference trajectory, as can be seen in Figure 3.8. As ϕ has reached its steady state value, the difference between y_1, y_2 and a simple mass is also constant, being $e = \pm\frac{l}{2}\phi$.

This result is well-known: mass feedforward does not lead to exact tracking in case of flexible systems. Intuitively, one could expect that y_2 achieves exact tracking if the feedforward signal $u_{\text{ff},2}$ is not a step response but increased from 0 to 1 somewhat gradually, *e.g.*, like a filtered step response. This would cause both x and $-\frac{l}{2}\phi$ and hence the output y to increase less quickly. In case of the non-minimum phase system, however, this is not that straightforward: since it is lagging, it seems that the force should initially be increased. However, this would also increase the initial rotation and hence the lag. A feedforward control strategy has to be designed that effectively deals with this apparent contradiction.

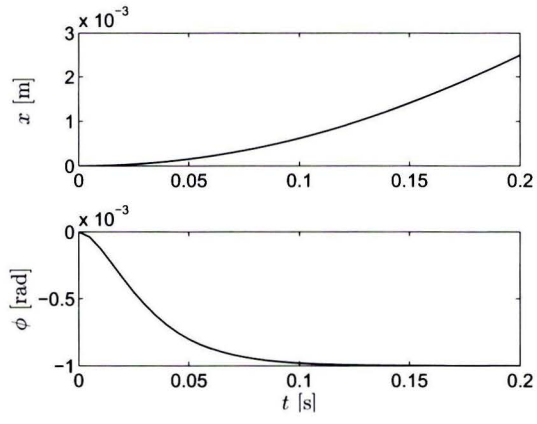


Figure 3.7: Displacement of the COG x and rotation ϕ of the flexible cart system due to a step input on u. The rotation ϕ converges to steady state.

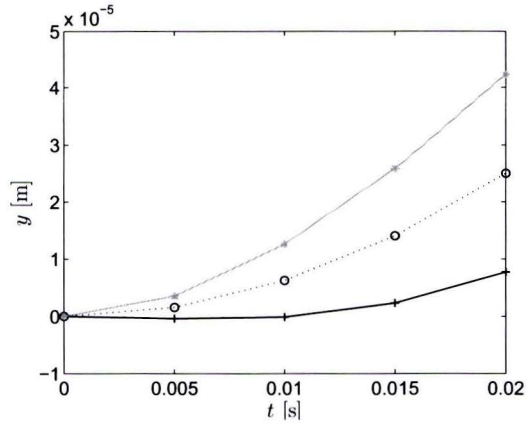


Figure 3.8: Reference trajectory (black, dotted), y_1 (black) and y_2 (gray). It appears that y_1 is lagging and y_2 is leading the reference trajectory.

3.4.3 The Inverse System

With this contradiction introduced, the question arises how it is possible that, in theory, an unstable inverse leads to exact tracking. In Figure 3.9, $u_{\text{ff},1}$ and $u_{\text{ff},2}$ are computed using the exact inverse systems. Since both $P_1(z)$ and $P_2(z)$ have a relative degree of $\rho = 1$, the inverse systems are improper so a pole at $z = 0$ is added to enable a causal implementation. Therefore, the reference trajectory is delayed with one sample before being fed into the feedback loop (Lunenburg, 2009, Section 3.1).

As can be seen in the lower plot $u_{\text{ff},2}$ converges to 1 as expected. The feedforward signal for the non-minimum phase system $u_{\text{ff},1}$ is quite different: it becomes negative, hence moving the COG in negative direction, while the flexible cart is supposed to move in positive direction. This can be explained by looking at the rotations in Figure 3.10: again, ϕ_2 nicely converges to the same steady state value as in Figure 3.7. However, due to the continuous negative force

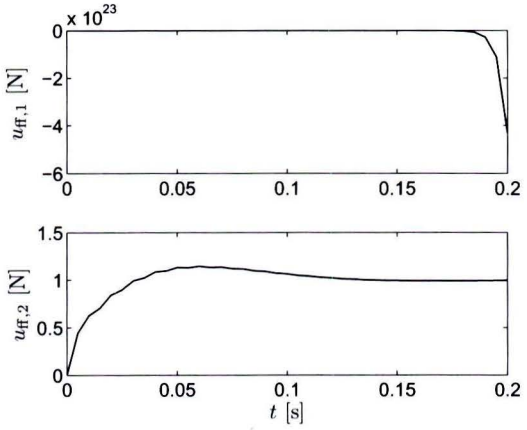


Figure 3.9: Feedforward control signals for y_1 (upper plot) and y_2 (lower plot). A theoretical feedforward force in the order of 10^{23} N (upper plot) is unrealistic.

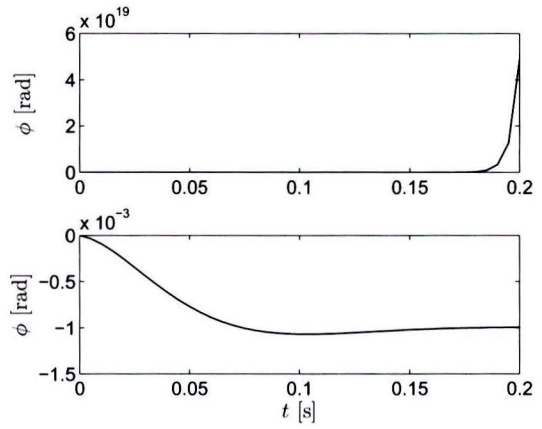


Figure 3.10: Rotation of the flexible cart system due to inversion-based feedforward: ϕ_1 (upper plot) and ϕ_2 (lower plot). An angle in the order of 10^{19} rad (upper plot) cannot be achieved in practice.

$u_{\text{ff},1}$, the increase of ϕ_1 is unbounded. Although the simulated response y_1 exactly matches the reference trajectory, this result for $P_1(z)$ obviously cannot be realized in practice.

3.4.4 Initial Conditions

The output of a dynamic system consists of the sum of the forced response, which is due to the input signal, and the free response, which is due to the initial conditions of the system. In the design of control systems, these initial conditions are often tacitly assumed to be zero, resulting in a zero free response[3]. This seems sensible since the systems under consideration are at rest before and after moves.

However, in Section 3.4.2 it is shown that the rotation ϕ converges to a steady state value if a constant force is applied. Therefore, it is interesting to see what happens if the simulation starts using these initial conditions, *i.e.*, $x_0 = \begin{bmatrix} 0 & 0 & -1 \times 10^{-3} & 0 \end{bmatrix}^T$. This is plotted in Figures 3.11 and 3.12. It appears that ϕ stays constant and both $P_1(z)$ and $P_2(z)$ exactly track the reference trajectory. This is an important result: contrary to the situation with zero initial conditions, a non-minimum phase system can actually exactly track a second order reference trajectory using inertia feedforward if the correct initial conditions are used.

Remark 3.2 *Note that the minimum phase system also achieves exact tracking if mass feedforward is used in combination with the correct initial condition. Nevertheless, exact tracking can also be obtained by using the inverse system (Section 3.4.3).*

A detailed discussion about a single or multiple (complex) NMP zeros can be found in Hoagg and Bernstein (2007), where undershoot, overshoot and zero crossings in servo systems due to NMP zeros are addressed.

[3]A system which is not connected to the world through a spring element will also remain in rest if the initial velocity equals zero and no external force is applied.

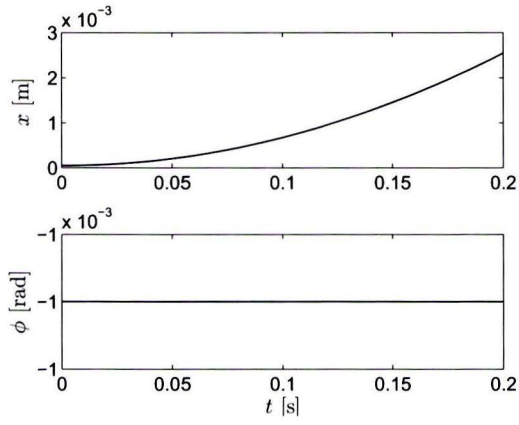


Figure 3.11: Displacement of the COG and rotation of the flexible cart system due to a step input on u, using $x_0 = 0$ m and $\phi_0 = -1 \times 10^{-3}$ rad. The rotation ϕ stays constant.

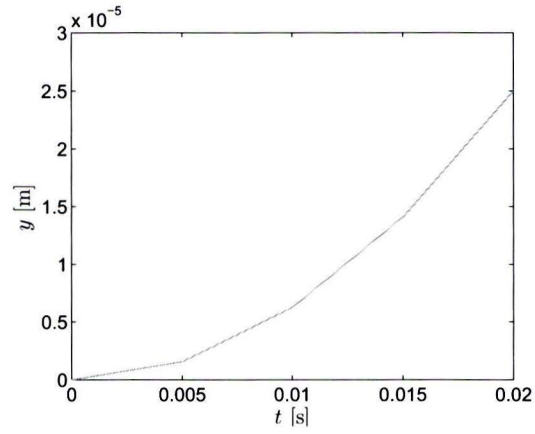


Figure 3.12: Reference trajectory (black, dotted), y_1 (black) and y_2 (gray) using $x_0 = 0$ m and $\phi_0 = -1 \times 10^{-3}$ rad. It appears that all three are exactly the same.

3.5 Sampling Zeros

Next to the non-minimum phase zeros discussed in the previous section, sampling zeros require some additional attention. In Åström et al. (1984), it is shown that additional zeros are introduced when sampling systems with a relative degree $\rho > 1$. If a continuous SISO system has a relative degree $\rho = 2$, the sampled data system will have an additional zero at $z \approx -1$ for a sufficiently small sampling period. If the inverse model is used for feedforward control, this will have a pole at $z \approx -1$. This is not desired for feedforward design, since it may introduce undamped oscillations of the resulting feedforward signal at the Nyquist frequency ($f_s/2$), see, *e.g.*, Franklin et al. (2002, Section 8.2.3).

This behavior is highly undesirable: not only might these vibrations turn up in the system output due to a mismatch between the model and the true system, but it might also have a significant impact on the intersample behavior. In Oomen (2010); Oomen et al. (2009) it is shown in an ILC context that excellent onsample performance does not necessarily imply good intersample behavior and that this effect may be caused by sampling zeros.

To prevent this oscillatory effect, the pole at $z = z_s$ should be taken out of the feedforward filter:

$$C_{\text{ff}}(z) = P^{-1}(z)\frac{z - z_s}{z\,(1 - z_s)} \tag{3.13}$$

The numerator of the transfer function cancels the pole at $z = z_s$, while the factor $1 - z_s$ assures that the steady state gain does not change. Furthermore, an additional pole at $z = 0$ is added to retain the relative degree. This modification of the feedforward filter can be seen as filtering of the feedforward path. Therefore, in order to maintain exact tracking, the reference trajectory should be filtered equivalently, see Figure 3.13.

Remark 3.3 *Note that this filtering of the reference trajectory can be seen as a subsample*

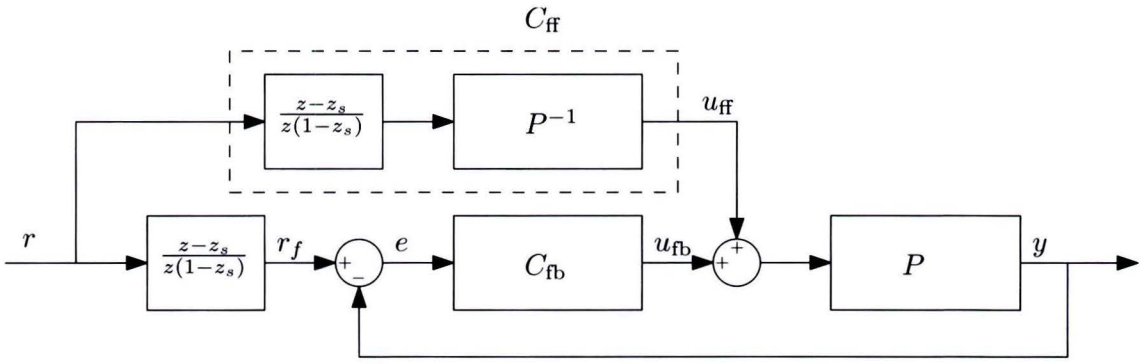


Figure 3.13: Filtering to avoid vibrations of the feedforward signal at the Nyquist frequency.

delay correction, see, e.g., *Van Donkelaar (2003):*

$$r_f(k+1) = \frac{1}{1-z_s} r(k+1) - \frac{z_s}{1-z_s} r(k) \tag{3.14}$$

Note, however, that in Van Donkelaar (2003) a subsample delay is tuned to improve the servo performance, while in this case it is used to avoid oscillations of the feedforward signal at the Nyquist frequency.

To illustrate the issue of sampling zeros and the proposed solution, a step response is performed using a simple mass, similar to Example 1.1 in Oomen (2010, 8.5.2).

Example 3.5 *The transfer function of a moving mass in combination with a ZOH and sampler is given by:*

$$P(z) = \frac{T_s^2}{2m} \frac{z+1}{(z-1)^2} \tag{3.15}$$

which has a zero at exactly $z = -1$. Using the inverse system to compute the feedforward signal for a step reference trajectory $r(t)$ results in the upper plot of Figure 3.14. In Figure 3.15 it appears that the output $y(t)$ exactly equals $r(t)$ at the sample instants but shows oscillatory behavior in between samples. This corresponds to the ILC result in Oomen (2010).

Removing the pole of the inverse system at $z = -1$ results in the feedforward signal in the lower plots of Figure 3.14. The oscillatory behavior of the tracking error has been removed, but the at-sample error at $t = 0.02$ s is not zero. Finally, filtering the reference trajectory as well (lower plot of Figure 3.15) results once again in exact at-sample tracking as well as optimized intersample behavior.

In Chapters 2 and 3, inverse systems and their properties are discussed. However, a model of a system never exactly represents the true system. Therefore, model uncertainty is addressed in Chapter 4.

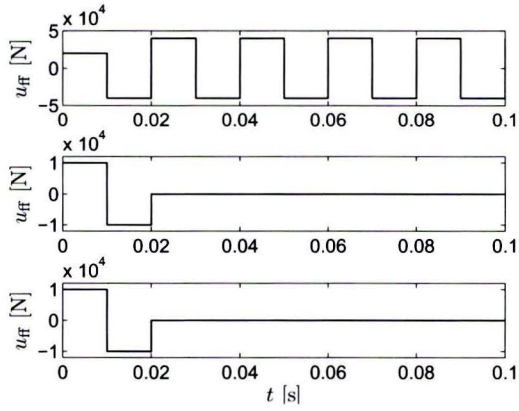


Figure 3.14: Comparison of u_{ff} for a step in the setpoint resulting from the exact inverse system (upper plot) and with the sampling zero removed (middle and lower plot). The upper plot shows oscillations of u_{ff} at the Nyquist frequency, while the lower two plots are the same.

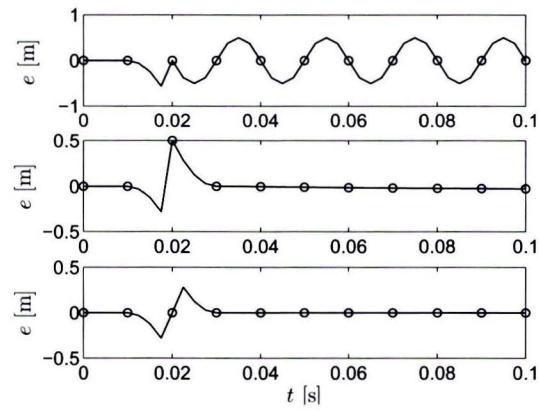


Figure 3.15: Comparison of the servo error at low (open dots) and high (solid line) sampling rate, with u_{ff} resulting from the exact inverse system (upper plot), the sampling zero removed (middle plot) and the sampling zero removed plus the setpoint filtered (lower plot). The upper plot shows good onsample but poor intersample behavior, while the middle plot has an onsample error at $t = 0.02$ s. The lower plot shows the best onsample and intersample behavior.

CHAPTER 4

Uncertainty and Feedforward

As shown in Chapter 2 exact tracking can be achieved if the feedforward filter is chosen as $C_{\text{ff}} = P^{-1}$. Although current system identification techniques result in accurate models, these will never *exactly* resemble the real plant. A certain amount of model uncertainty will always remain due to, *e.g.*, production tolerances, nonlinearities, wear and position dependent dynamics. Furthermore, a model cannot be made overly complex, so a finite model order is also an important source of uncertainty. In this chapter, the consequences of model uncertainty for feedforward control are addressed. In Section 4.1, various uncertainty structures used to describe these uncertainties are discussed. Next, a few examples of mismatches between the true plant and the model are investigated to gain insight in the effect these have. The role of feedback is also taken into account. There exist robust feedforward solutions which explicitly deal with model uncertainty. These are introduced in Section 4.3. These commonly optimize the 'worst-case' performance, but one could question whether this approach is suited for feedforward design. This is discussed in Section 4.4.

4.1 Uncertainty Modeling

In Skogestad and Postlethwaite (2005), two main classes of model uncertainty are distinguished:

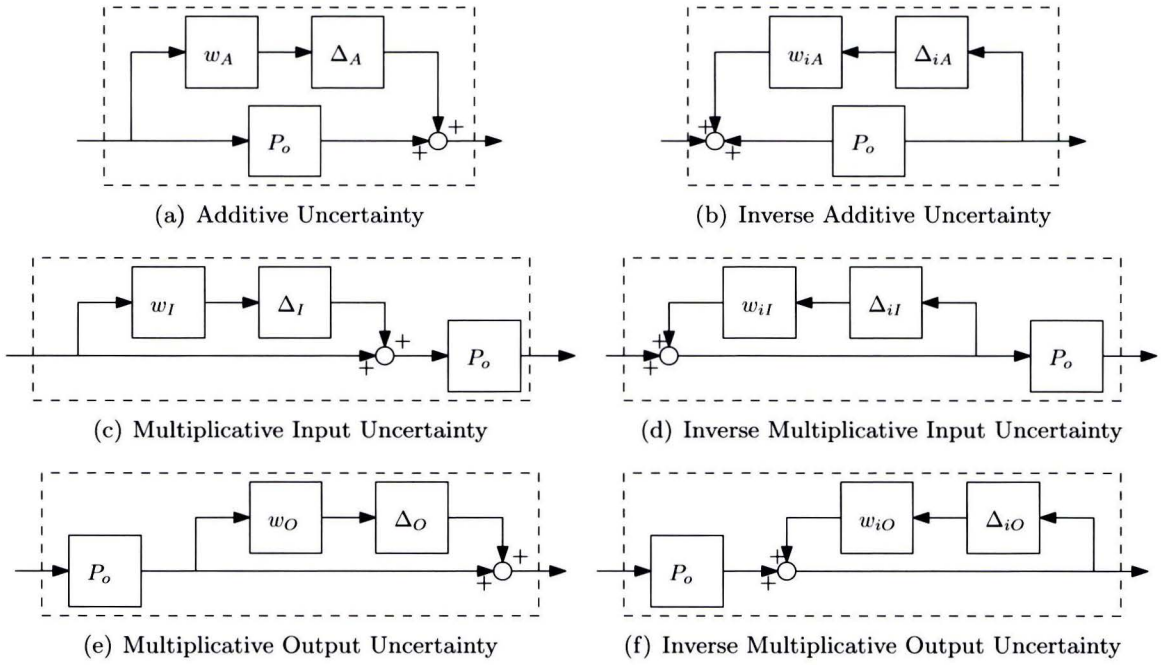
- Parametric (real) uncertainty: the model structure is completely known, but (some of) the parameters are uncertain. In this case, an uncertain parameter $\alpha_p \in [\alpha_{\min}, \alpha_{\max}]$ can be described by:
$$\alpha_p = \alpha_o \left(1 + r_\alpha \Delta\right) \tag{4.1}$$
where α_o is the nominal value and $r_\alpha = \left(\alpha_{\max} - \alpha_{\min}\right) / \left(\alpha_{\max} + \alpha_{\min}\right)$ is the relative uncertainty and Δ is any real scalar $\Delta \in [-1, 1]$.
- Dynamic (frequency-dependent) uncertainty: Since a true system is too complex to be represented exactly by a model of finite order, there is model uncertainty due to missing dynamics.

The models of wafer stages are obtained by system identification rather than by physical modeling and therefore the model parameters do not directly represent physical variables. This means that dynamic uncertainty models are used for the control design. There are a

number of uncertainty structures which are commonly used for control design. These are discussed in the next section.

4.1.1 Uncertainty Structures

In Figure 4.1 a number of commonly used uncertainty structures is displayed. Here, Δ can be any stable system satisfying $\|\Delta\|_\infty \leq 1$. Furthermore, w denotes a stable, minimum phase weighting filter. Of these six alternatives, multiplicative uncertainty weights are often preferred since these are more informative: if $|w_I(j\omega)| > 1$, the uncertainty exceeds the magnitude of the plant, which is important in both feedback and feedforward control. This will be further discussed in Section 4.3.



(a) Additive Uncertainty

(b) Inverse Additive Uncertainty

(c) Multiplicative Input Uncertainty

(d) Inverse Multiplicative Input Uncertainty

(e) Multiplicative Output Uncertainty

(f) Inverse Multiplicative Output Uncertainty

Figure 4.1: Various commonly used uncertainty models

A disadvantage of the uncertainty structures displayed in Figure 4.1 is that poles and zeros cannot cross from the LHP into the RHP and vice-versa (in continuous time). This is important since in case of only LHP zeros, the inverse model can be used directly for feedforward control while RHP zeros result in an unstable inverse system. Therefore, a coprime-factor-based uncertainty description (Skogestad and Postlethwaite, 2005) or Dual-Youla-Kučera model uncertainty structure (Oomen, 2010) are more suited. These less common uncertainty descriptions are, however, beyond the scope of this thesis.

4.1.2 Uncertainty in the Lifted System Description

In the lifted system description, a static map J, representing convolution, is used to describe the input-output behavior of a dynamic system (Bamieh et al., 1991; Lunenburg, 2009;

Van de Wijdeven, 2008). The lifted system description will be elaborated in Section 5.1. In ILC, addressing uncertainty is mainly important to ensure robust convergence. Hereto, an uncertainty description in the lifted domain is important. Consider the singular value decomposition of the convolution matrix J:

$$J = U\Sigma V^T \tag{4.2}$$

where V and U contain the input and output singular vectors and Σ is a diagonal matrix with the singular values $\sigma_1, \ldots, \sigma_N$ ordered in decreasing magnitude. One could expect that uncertainty concerns the smaller singular values but this turns out to be not true.

Looking more carefully at the singular values and singular vectors, it can be concluded that the singular vectors contain harmonic oscillations. The corresponding singular values denote the magnitude of the Bode-diagram of the underlying system. This is confirmed by Dijkstra (2003), and a more elaborate discussion of the interpretation of singular values and singular vectors can be found in Appendix B.

In Van de Wijdeven (2008), an uncertainty description like the ones above is used in the lifted domain. In this case, the weighting filters and Δ are lower triangular (block) Toeplitz matrices.

4.2 Feedback

Model uncertainty implies that the model is not an exact representation of the plant. It is interesting to see how a mismatch between model and plant affects the tracking performance, and what the effect of feedback is. Hereto, a number of possibilities is discussed. First, the presence of friction is investigated, followed by a low-frequent gain mismatch. Finally, phase delays are discussed.

4.2.1 Damping

Industrial motion stages are often modeled with a low-frequent -2 slope, which implies that there is no damping or stiffness with respect to the fixed world. In practice there may be some friction or stiffness present due to, *e.g.*, a cable slab or Eddy current damping, which results in a -1 or 0 slope at low frequencies. A low-frequent mismatch between $C_{\text{ff}}(s)$ and $P^{-1}(s)$ will, to a certain extent, be attenuated by the integral action of $C_{\text{fb}}(s)$. Here, the remaining error will be quantified in the frequency domain using $S_{\text{ff}}(s)$ and $S_r(s)$.

Assume that the true system is a simple mass with $m = 1$ kg, given by:

$$P(s) = \frac{1}{s(s + \omega_f)} \tag{4.3}$$

while the nominal model is given by:

$$P_o(s) = \frac{1}{s^2} \tag{4.4}$$

Now, if the inverse of the model is used as feedforward controller, $C_{\text{ff}}(s) = P_o^{-1}(s)$, the resulting feedforward sensitivity is given by:

$$S_{\text{ff}}(s) = 1 - P(s)P_o(s)^{-1} = 1 - \frac{1}{s(s+\omega_f)}\frac{s^2}{1} = \frac{\omega_f}{s+\omega_f} \tag{4.5}$$

At high frequencies, this is a minus one slope, meaning that the feedforward controller improves tracking performance, but at low frequencies ($\omega < \omega_f$), $S_{\text{ff}}(s) \approx 1$ so it does not improve performance with respect to the situation without feedforward, as was expected. Furthermore, $S_{\text{ff}}(s)$ increases linearly with ω_f. A simple proportional-integral (PI) controller is introduced to illustrate to what extent the feedback controller is able to attenuate this:

$$C_{\text{fb}}(s) = k\frac{s+\omega_i}{s} \tag{4.6}$$

Note that in practice derivative action is required to stabilize the system, but this occurs at a higher frequency than ω_i and is therefore omitted for simplicity. The resulting output sensitivity is given by:

$$S_o(s) = \frac{s^2(s+\omega_f)}{s^3 + \omega_f s^2 + ks + k\omega_i} \tag{4.7}$$

Now, the reference sensitivity is given by:

$$S_r(s) = S_o(s)S_{\text{ff}}(s) = \frac{\omega_f s^2}{s^3 + \omega_f s^2 + ks + k\omega_i} \tag{4.8}$$

An exemplary $S_r(s)$ with $\omega_f = 0.2\pi$ rad/s, $\omega_i = 6\pi$ rad/s and $k = 9$ is displayed in Figure 4.2. At low and high frequencies, $S_r(s)$ asymptotically approaches:

$$\lim_{s\to 0} S_r(s) = \frac{\omega_f}{k\omega_i}s^2 \tag{4.9}$$

$$\lim_{s\to\infty} S_r(s) = \frac{a}{s} \tag{4.10}$$

These asymptotes intersect at $\omega = \sqrt[3]{k\omega_i}$ rad/s with a magnitude of $|S_r(j\sqrt[3]{k\omega_i})| \approx \frac{\omega_f}{\sqrt[3]{k\omega_i}}$. In this example, $\max|S_r(s)| = -20.3$ dB, indicating that although ω_f manifests itself a factor 30 below ω_i it still has a significant impact on the eventual tracking performance. In summary, the extent to which the feedback controller is able to attenuate such an uncertainty depends on the locations of the integrator frequency and the uncertainty.

4.2.2 Gain Mismatch

In Lunenburg (2009), it was shown that a low-frequent gain mismatch leads to large tracking errors if only a proportional controller is used. Here, an analysis similar to the previous section is performed for such a mismatch. Take, *e.g.*, the system:

$$P(s) = \frac{1}{ms^2} \tag{4.11}$$

which is approximated by the nominal model:

$$P_o(s) = \frac{1}{(m+\delta)s^2} \tag{4.12}$$

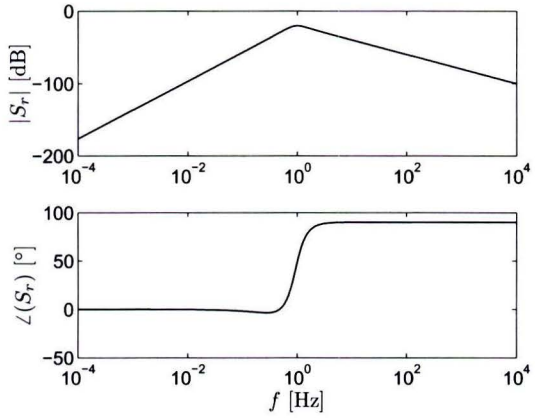


Figure 4.2: $S_r(s)$ in case of a system which contains damping which is not present in the model.

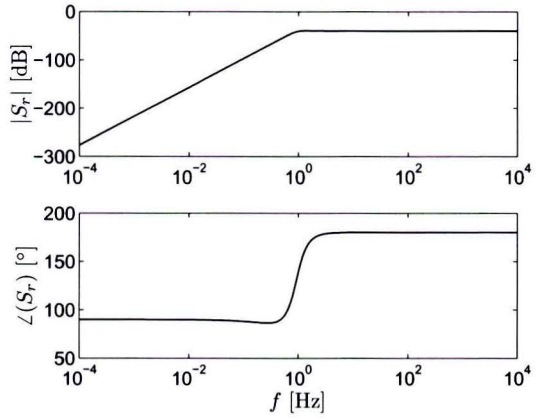


Figure 4.3: $S_r(s)$ in case of a mismatch between the gain of the inverse of the true system and the feedforward controller.

If $C_{\text{ff}} = P_o^{-1}$, the feedforward sensitivity is given by:

$$S_{\text{ff}}(s) = 1 - PP_o^{-1} = -\frac{\delta}{m} \tag{4.13}$$

which is constant over the entire frequency range. Using the PI controller of (4.6), $S_o(s)$ is given by:

$$S_o(s) = \frac{ms^3}{ms^3 + ks + k\omega_i} \tag{4.14}$$

Hence, the reference sensitivity equals:

$$S_r(s) = -\frac{\delta}{m}\frac{ms^3}{ms^3 + ks + k\omega_i} = -\frac{\delta s^3}{ms^3 + ks + k\omega_i} \tag{4.15}$$

which is plotted for $m = 1$ kg and $\delta = 0.01$ kg in Figure 4.3. The limits of $S_r(s)$ are given by:

$$\lim_{s\to 0} S_r(s) = -\frac{\delta}{k\omega_i}s^3 \tag{4.16}$$

$$\lim_{s\to\infty} S_r(s) = -\frac{\delta}{m} \tag{4.17}$$

These asymptotes intersect at $\omega = \sqrt[3]{\frac{k\omega_i}{m}}$, which is independent of the actual mismatch δ. Note that $\lim_{s\to 0} S_r(s)$ has a low-frequent +3 slope. This is due to the integrator. If a controller did not contain integral action ($\omega_i = 0$), (4.15) would become:

$$S_r(s) = -\frac{\delta s^3}{ms^3 + ks} \tag{4.18}$$

with

$$\lim_{s\to 0} = -\frac{\delta s^2}{k} \tag{4.19}$$

This is a +2 slope hence the transfer function from the desired acceleration $r_a(s)$ to the error $e(s)$ would give a good indication of the tracking error during constant acceleration.

4.2.3 Phase Lag

The examples discussed above both concern a gain mismatch between $C_{\text{ff}}(s)$ and $P_o^{-1}(s)$, but also a phase mismatch can cause severe tracking errors: a phase shift of ϕ implies a time delay of ϕ/ω for a sinusoidal input with frequency ω (Tomizuka, 1987). In turn, this causes $S_r(s)$ to be unequal to zero. Take again a simple mass ($m = 1$ kg). If C_{ff} lags P_0^{-1} with only 1°, this results in $|S_{\text{ff}}| = -35.2$ dB over the entire frequency range, which equals a gain mismatch of 1.7%. As discussed before, this is far too much for accurate servo performance.

In Appendix A, it is argued that using Euler integrators in series to compute reference trajectories leads to a transfer function from reference to output:

$$Y(z) = \frac{z+1}{2z^2} R(z) \tag{4.20}$$

Although this is not a plant property and cannot be attributed to model uncertainty, this difference can be seen as a phase delay of $1\frac{1}{2}$ samples. If the sample delay due to the relative degree of $\rho = 1$ is compensated, a subsample delay of $\frac{1}{2}$ remains. This corresponds to a time delay of:

- $0.5/2500 = 0.0002$ s with $f_s = 2.5$ kHz
- $0.5/5000 = 0.0001$ s with $f_s = 5.0$ kHz

The resulting servo errors during a constant scanning motion with a velocity of 1 m/s are 0.2 mm and 0.1 mm. This is obviously too large for the feedback controller to attenuate, indicating the importance of tuning the required delay correction (Van der Meulen, 2005b; Van der Meulen et al., 2008).

4.3 Robust Feedforward

Now that the uncertainty is modeled, it should be suitably addressed in feedforward design. In feedback control design, $\mathcal{H}_\infty$ and $\mathcal{H}_2$ optimization are commonly used to address model uncertainty. According to Skogestad and Postlethwaite (2005), the presence of non-parametric and unstructured uncertainty is the *raison d'être* for $\mathcal{H}_\infty$ optimization. Similarly, these tools can be used to synthesize feedforward controllers, see, *e.g.*, Hoyle et al. (1991); Lee and Salapaka (2009); Lunenburg (2009); Prempain and Postlethwaite (2001). These methods typically result in a feedforward controller which gives a guaranteed performance for every plant P in the set $\mathcal{P}$. This result is commonly obtained by reducing the gain, *i.e.*, $|C_{\text{ff}}(j\omega)| \leq |P^{-1}(j\omega)|\ \forall\omega$, thus also deteriorating the nominal performance.

Another reference which is often quoted in discussing robust feedforward is Devasia (2002), which investigates the use of the exact inverse model for feedforward. With the additive uncertainty defined as:

$$\Delta(j\omega) := P_o(j\omega) - P_p(j\omega) \tag{4.21}$$

where P_o denotes the nominal model and P_p the perturbed plant, and for multivariable plants the condition number:

$$\kappa_{P_o}(j\omega) = \frac{\bar{\sigma}(j\omega)}{\underline{\sigma}(j\omega)} \tag{4.22}$$

Table 4.1: Comparison of reference sensitivity $S_r(\Delta, j\omega)$ and output sensitivity $S_o(\Delta, j\omega)$ for different uncertainty size $\Delta(j\omega)$.

Size of Uncertainty	Comparison of Tracking Performance
$\|\Delta(j\omega)\|_2 \leq \frac{P_o(j\omega)}{\kappa_{P_o}(j\omega)}$	For all controllers and any uncertainty $\Delta(j\omega)$ $S_r(\Delta, j\omega) \leq S_o(\Delta, j\omega)$
$\frac{\|P_o(j\omega)\|_2}{\kappa_{P_o}(j\omega)} < \|\Delta(j\omega)\|_2$	There exists a controller and an uncertainty $\tilde{\Delta}(j\omega)$ such that $S_r(\Delta, j\omega) > S_o(\Delta, j\omega)$
$\|P_o(j\omega)\|_2 < \|\Delta(j\omega)\|_2$	For any controller, there exists an uncertainty $\hat{\Delta}(j\omega)$ such that $S_r(\Delta, j\omega) > S_o(\Delta, j\omega)$

the conclusions are summarized in Table 4.1.

In words, this means that at those frequencies ω where the additive uncertainty $\Delta(j\omega)$ is smaller than the the nominal plant $P_o(j\omega)$ divided by its condition number κ_{P_o}, using the inverse model as feedforward controller leads to improved servo performance compared to the situation without feedforward. Therefore, the robust feedforward filter proposed in Devasia (2002) equals zero when $\|\Delta(j\omega)\|_2 > \frac{\|P_0(j\omega)\|}{k_{P_o}(j\omega)}$ and equals the inverse model when $\|\Delta(j\omega)\|_2 \leq \frac{\|P_0(j\omega)\|}{k_{P_o}(j\omega)}$ with $\Delta(j\omega)$ as defined in (4.21).

In Wu and Zou (2009) a more sophisticated gain modulated system inverse is used for feedforward:

$$C_{\text{ff}}(j\omega) = \frac{2\cos\left(|\Delta\theta|_{\max}(\omega)\right)}{\Delta r_{\min} + \Delta r_{\max}} P_o^{-1}(j\omega) \tag{4.23}$$

where Δr and $\Delta\theta$ are defined by:

$$\frac{P_p(j\omega)}{P_o(j\omega)} = \Delta r(\omega) e^{j\Delta\theta(\omega)} \tag{4.24}$$

This approach may yield less conservative results since both magnitude and phase information of the uncertainty are taken into account. Similar to $\mathcal{H}_\infty$ and $\mathcal{H}_2$ feedforward, it is an optimization of the worst case. In the next section it is discussed whether this approach is suited for feedforward.

4.4 Discussion

All methods discussed in the previous section have in common that they optimize performance for the worst-case plant P present in $\mathcal{P}$, leading to a guaranteed performance for every $P \in \mathcal{P}$. In feedback control, this approach makes sense, since the feedback controller is supposed to stabilize the system and the closed loop system must be guaranteed to be stable. However, this worst-case approach may not always achieve optimal performance, *e.g.*, if the worst-case plant rarely or never occurs (Skogestad and Postlethwaite, 2005, p. 259). Therefore, it can be argued that better servo performance may be obtained when a different approach is used for feedforward, *e.g.*, optimizing the average performance of all $P \in \mathcal{P}$. This is illustrated by means of a simple example with a constant plant subject to real uncertainty.

Example 4.1 *Consider a system with a nominal model*

$$P_o = 1 \tag{4.25}$$

which is subject to a large parameter uncertainty such that

$$\mathcal{P} = P_o\,(1 + 1.1\Delta)\,, \quad \Delta \in [-1, 1] \tag{4.26}$$

Since the magnitude of the uncertainty exceeds the magnitude of the plant, robust feedforward implies that $C_{\text{ff}} = 0$ *(Devasia, 2002). However, in* $1/1.1 = 91\%$ *of the plants* $P \in \mathcal{P}$*, better performance is achieved when the inverse of the nominal system is used as feedforward:* $C_{\text{ff}} = 1/\hat{P} = 1$.

To compute an optimal solution, a cost function is defined as:

$$H(\Delta) = S_{\text{ff}}(\Delta)^2 = (1 - P(\Delta)C_{\text{ff}})^2 \tag{4.27}$$

Instead of optimizing the worst case ($\Delta = -1$)*, the optimal solution for the entire model set* $\mathcal{P}$ *is defined as:*

$$C_{\text{ff}} = \arg\min_{C_{\text{ff}}} \int_{-1}^{1} H(\Delta)d\Delta \tag{4.28}$$

which can be computed to be $C_{\text{ff}} = 0.71$.

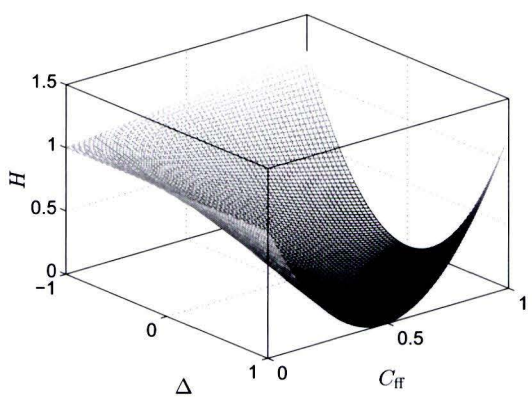


Figure 4.4: The cost function $H = (1 - P(\Delta)C_{\text{ff}})^2$ for various values of Δ and C_{ff}.

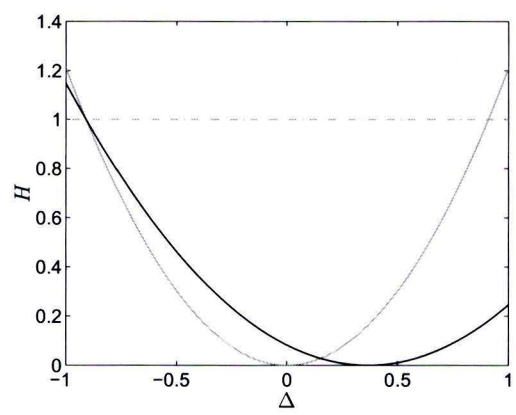


Figure 4.5: The cost function H as a function of Δ for $C_{\text{ff}} = 0$ (grey, dash-dotted), $C_{\text{ff}} = 1$ (grey, solid) and $C_{\text{ff}} = 0.71$ (black) The area under the gray, dash-dotted line is the largest, while the area under the black line is smallest, indicating that $C_{\text{ff}} = 0.71$ is the optimal solution.

This is graphically depicted in Figures 4.4 and 4.5. It can be seen that if $C_{\text{ff}} = 0$*,* $H(\Delta)$ *will never exceed* 1*, but the overall performance,* i.e.*, the area under the gray, dash-dotted line in Figure 4.5 is much larger than the area under the solid gray line, obtained by using* $C_{\text{ff}} = P_o^{-1}$ *(see also Table 4.2). This area is minimized by using* $C_{\text{ff}} = 0.71$*, resulting in the black line.*

This is quite an extreme example where the multiplicative uncertainty exceeds the nominal model. In practice, this will usually only occur at high frequencies. However, also if the

Table 4.2: Integral of H on $\Delta \in [-1, 1]$.

C_{ff}	$\int_{-1}^{1} H d\Delta$
0.00	2.00
1.00	0.80
0.71	0.58

uncertainty is smaller smaller ($w_I < 1$), the result is not optimal for $C_{\text{ff}} = P_o^{-1}$, showing that a robust feedforward approach optimizing C_{ff} for the entire model set $\mathcal{P}$ can improve servo performance. This result is taken into account in robust feedforward synthesis in Chapters 5 and 6.

CHAPTER 5

The Use of the Lifted System Representation for Feedforward Controller Synthesis

The lifted system description is an increasingly used way of describing linear systems, *e.g.*, in Iterative Learning Control Dijkstra (2003). According to Van der Meulen (2005a, p. 97), the main advantages of the lifted system representation are the large design flexibility and the excellent analysis properties. In Lunenburg (2009), it is shown how the lifted system representation can be used for feedforward control design. The proposed method is relatively straightforward: the impulse response matrix used in the lifted system representation is inverted and the resulting matrix can be used to compute the feedforward signal. This technique was successfully applied to the minimum phase flexible cart system introduced in Section 3.4.1. A number of issues, however, remained unsolved:

- Non-minimum phase systems lead to convolution matrices with singular values $\sigma \approx 0$ for every NMP zero (Dijkstra, 2003; Hashemi and Hammond, 1996). This means that using the inverse convolution matrix leads to large (unbounded) feedforward signals, which cannot be applied in practice. How can the inverse convolution matrix be approximated and what is the error that results from this approximation?
- In Section 3.4.4, it is shown that certain initial conditions other than $x(0) = 0$ can lead to exact tracking for NMP systems. What requirements do these initial conditions have to meet? How can this be explained in the lifted domain? Furthermore, what is the effect of pre-actuation in the lifted domain?
- How can model uncertainty be dealt with in a suitable way?

These issues are investigated in this chapter. First, the approach in Lunenburg (2009) is extended to include non-minimum phase systems in Section 5.1. Thereafter, a lower bound on the resulting servo error is discussed in Section 5.2, where the initial conditions and pre-actuation are also taken into account. It appears that the solutions of Sections 5.1 and 5.2 are Linear Time Varying (LTV) rather than Linear Time Invariant (LTI). In Section 5.3, it is explained how LTI solutions can be enforced if this is desired. Thereafter, in Section 5.4, it is discussed how uncertainty can be handled in the spirit of Section 4.4. Despite the advantages of the lifted system description, numerical issues form a drawback on these concepts. These are discussed in Section 5.5. The various issues are illustrated by means of an example with

the flexible cart system in Section 5.6.

5.1 Concepts

In the lifted system representation, a static map, representing convolution, is used to describe the system dynamics. Commonly, a convolution matrix (or impulse response matrix) with finite dimensions is used, leading to the following description:

$$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(N-1) \end{bmatrix} = \begin{bmatrix} D & 0 & \cdots & 0 \\ CB & D & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{N-2}B & CA^{N-3}B & \cdots & D \end{bmatrix} \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(N-1) \end{bmatrix} + \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{N-1} \end{bmatrix} x_0 \tag{5.1}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{m\times n}$ and $D \in \mathbb{R}^{m\times m}$ denote the matrices of an nth order square state-space description. Furthermore, N represents the length of input- and output vectors, *i.e.*, the length of the actuation and observation intervals. Although not a general requirement, these are chosen equal in this case, since this closely resembles the situation of an online feedforward filter. As a result, the convolution matrix is square. The last term of (5.1) represents the contribution of the initial conditions x_0. For now, these are assumed to be zero. Matrix J_0 resembles an 'extended' observability matrix, *i.e.*, the observability matrix of a state-space system is defined as the first $m \times n$ rows of J_0. The short notation used for (5.1) is:

$$\boldsymbol{y} = J\boldsymbol{u} + J_0 x_0 \tag{5.2}$$

If a system has a relative degree $\rho > 0$, the first ρ rows of matrix J are zero. In order to prevent the resulting loss of rank, the output vector can be shifted over ρ samples. With a relative degree $\rho = 1$, which is typical for the systems under consideration (discrete systems having force input and position output), (5.1) becomes:

$$\begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(N) \end{bmatrix} = \begin{bmatrix} CB & 0 & \cdots & 0 \\ CAB & CB & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{N-1}B & CA^{N-2}B & \cdots & CB \end{bmatrix} \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(N-1) \end{bmatrix} + \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^N \end{bmatrix} x_0 \tag{5.3}$$

In case of MIMO systems, there is a vector relative degree (Section 2.3) and this shift is not this easy in general. However, for the systems under consideration D typically is a zero matrix and the product CB of full rank, in which case this shifted representation can be used directly.

Now, with the initial conditions $x_0 = 0$, designing the feedforward signal $\boldsymbol{u}$ implies computing the solution to $\boldsymbol{y} = J\boldsymbol{u} = \boldsymbol{r}$. Ideally, $\boldsymbol{u} = J^{-1}\boldsymbol{r}$ such that $\boldsymbol{y} = JJ^{-1}\boldsymbol{r} = \boldsymbol{r}$. In Hashemi and Hammond (1996), however, it is mentioned that J has a singular value $\sigma \approx 0$ for every NMP zero of the underlying system. This is confirmed by, *e.g.*, Dijkstra (2003). This means that the exact inverse of J cannot be used in general to calculate $\boldsymbol{u}$, since it results in large (unbounded) $\boldsymbol{u}$. As an alternative, the Moore-Penrose generalized inverse (also called Moore-Penrose pseudo-inverse) can be used:

Proposition 5.1 *Consider a system with $n_{z,u}$ unstable zeros and the singular value decomposition of the convolution matrix of* (5.2)*:*

$$J = U\Sigma V^T = \left[\begin{array}{c|c} U_1 & U_2 \end{array}\right] \left[\begin{array}{c|c} \Sigma_1 & 0 \\ \hline 0 & \Sigma_2 \end{array}\right] \left[\begin{array}{c} V_1^T \\ \hline V_2^T \end{array}\right] \tag{5.4}$$

Here, Σ_1 contains the $N - n_{z,u}$ 'clustered' singular values, while Σ_2 contains the $n_{z,u}$ singular values $\sigma \approx 0$. By clustered it is meant that the diagonal entries of Σ_1 are large compared to the diagonal entries of Σ_2 (see also Figure 5.10). The matrices U and V, containing the singular vectors, are partitioned accordingly.

A bounded feedforward signal $\boldsymbol{u}$ can now be computed by:

$$\boldsymbol{u} = V_1\Sigma_1^{-1}U_1^T\boldsymbol{r} = J^\dagger\boldsymbol{r} \tag{5.5}$$

This bounded feedforward signal $\boldsymbol{u}$ minimizes the 2-norm (en hence the energy) of the servo error $\boldsymbol{e} = \boldsymbol{r} - J\boldsymbol{u}$.

If the underlying system is minimum phase, Σ_2 is empty and $J^\dagger = J^{-1}$. In case of a non-minimum phase system, $J^\dagger = V_1\Sigma_1^{-1}U_1^T$ is neither lower-triangular nor Toeplitz, implying that the underlying system is neither causal nor linear time invariant (LTI). This can be seen in Figures 5.1 and 5.2, where the first row and the diagonal of the generalized inverse of the 200×200 convolution matrix ($N = 200$) of the NMP flexible cart system are plotted. While using J^{-1} in case of MP systems leads to exact tracking, replacing the exact inverse J^{-1} by the Moore-Penrose pseudo-inverse $J^\dagger$, however, introduces a servo error. The lower bound on the resulting servo error is analyzed in the next section.

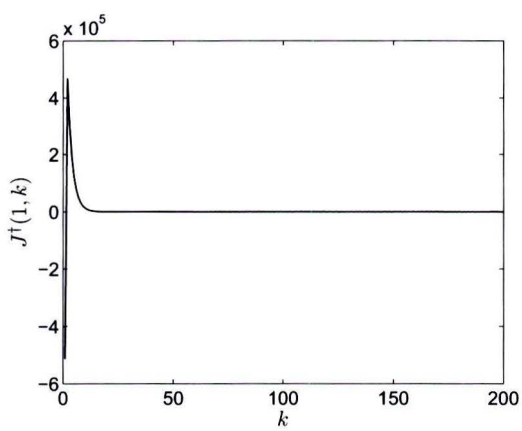


Figure 5.1: First row of the 200×200 $J^\dagger$ of the NMP flexible cart system. The observation that $J^\dagger(1,k) \neq 0$ for $k > 1$ indicates that the underlying system is non-causal.

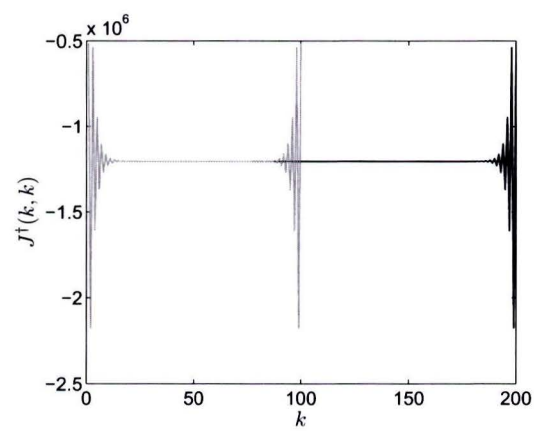


Figure 5.2: Diagonal of the 100×100 (gray) and 200×200 (black) $J^\dagger$ of the NMP flexible cart system. Since $J^\dagger(k,k)$ is not equal for all k, it is concluded that the underlying system is not LTI. For different sizes of $J^\dagger$, the effects on the edges are the same, hence the black and gray lines coincide on the left side. If $J^\dagger$ is sufficiently large, the center part of this matrix is approximately Toeplitz hence LTI.

5.2 Performance Limitations in the Lifted Domain

As shown in Section 5.1, a feedforward signal can be computed by inverting the convolution matrix of a system and multiplying this with the reference trajectory. In case of a minimum phase system, this leads to exact tracking of an arbitrary reference trajectory. In case of a NMP system, however, the inverse convolution matrix is approximated by the Moore-Penrose pseudo-inverse. As a result, the feedforward signal $\boldsymbol{u}$ is bounded, but a servo error is introduced as well. In Section 5.2.1, this error is quantified.

However, in Section 3.4.4 it is shown that a NMP system can achieve exact tracking with a bounded feedforward signal if it has suitable initial conditions. This is investigated in the lifted domain in Section 5.2.2. The question naturally arising from this observation is whether it is possible to use pre-actuation on a limited interval $k \in [-n_{\text{nc}}, 0]$ to bring the system from $x(k = -n_{\text{nc}}) = 0$ to *any* $x(k = 0) = x_0$ while keeping the output $y(k \leq 0) = 0$. This is discussed in Section 5.2.3.

Nevertheless, it appears that this two-step approach does not lead to a clear bound on the achievable performance. In order to quantify this bound, a feedforward signal including pre-actuation should be computed without explicitly defining x_0. This way, a bound on the achievable performance can be given which is discussed in Section 5.2.4.

5.2.1 Quantifying the Error in the Lifted Domain

Recall that the feedforward problem stated at the start of this chapter is defined as computing a feedforward signal $\boldsymbol{u}$ such that the 2-norm of the servo error defined by:

$$\boldsymbol{e} = \boldsymbol{r} - \boldsymbol{y} = \boldsymbol{r} - J_N \boldsymbol{u} \tag{5.6}$$

is minimized. In this equation, $\boldsymbol{e}$ is the $N \times 1$ error vector, $\boldsymbol{y}$ is an $N \times 1$ output vector, J_N the $N \times N$ convolution matrix of the plant (the subscript N denotes the size) and $\boldsymbol{r}$ the $N \times 1$ reference trajectory. Furthermore, zero initial conditions are assumed. Now, if (5.5) is used to compute the feedforward signal and given zero initial conditions, the output of the system is given by:

$$\begin{aligned} \boldsymbol{y} &= J_N J_N^{\dagger} \boldsymbol{r} \\ &= \left[\begin{array}{c|c} U_1 & U_2 \end{array}\right] \left[\begin{array}{c|c} \Sigma_1 & 0 \\ \hline 0 & \Sigma_2 \end{array}\right] \left[\begin{array}{c} V_1^T \\ \hline V_2^T \end{array}\right] [V_1] \left[\Sigma_1^{-1}\right] \left[U_1^T\right] \boldsymbol{r} \\ &= U_1 U_1^T \boldsymbol{r} \end{aligned} \tag{5.7}$$

Hence, the resulting servo error is given by:

$$\boldsymbol{e} = \left(I - U_1 U_1^T\right) \boldsymbol{r} = U_2 U_2^T \boldsymbol{r} \tag{5.8}$$

This expression is the servo error for an arbitrary reference trajectory if a non-causal, time-varying feedforward signal is computed in the lifted domain for an exact model. Hence, it forms a lower bound on the achievable servo performance if other limitations such as causality and time-invariance are enforced. Basically, it implies that exact tracking can only be achieved

if the reference trajectory $\boldsymbol{r}$ is perpendicular to U_2. On the other hand, the open loop servo error equals $\boldsymbol{r}$ if this is a linear combination of the columns of U_2.

As discussed in Appendix B and Dijkstra (2003); Skogestad and Postlethwaite (2005), the vectors in U_2 are exponentially decreasing with the rate determined by the location of the NMP zeros. In Figure 5.3 the output singular vector U_2 corresponding to the single, real NMP zero of the flexible cart system is plotted. In general, $\boldsymbol{r}$ is not perpendicular to U_2, since the location of the NMP zeros of the system is usually not considered in the design of a reference trajectory. This implies that exact tracking cannot be achieved for NMP systems with initial conditions $x(0) = 0$ and a bounded feedforward signal.

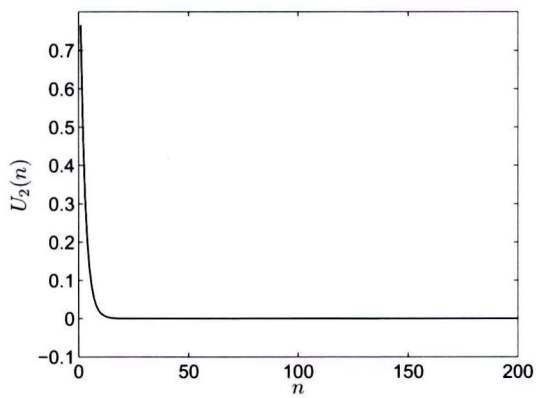


Figure 5.3: Output singular vector corresponding to the NMP zero of the flexible cart system, which is a decreasing exponential function.

5.2.2 Initial Conditions

The previous section confirms that a NMP system cannot achieve exact tracking on $k \in [0, N]$ if a bounded feedforward signal and zero initial conditions are used. This raises the question what happens if nonzero initial conditions are used. In Section 3.4.4 it is shown that the NMP flexible cart system can achieve exact tracking if it starts with suitable initial conditions. In that particular case, suitable initial conditions and a corresponding feedforward were determined pragmatically, but in general this is not that straightforward. In the lifted domain it can be confirmed that exact tracking can be achieved.

Theorem 5.1 *Given a system with convolution matrix J_N and extended observability matrix J_0, there exists a bounded feedforward signal $\boldsymbol{u}$ and initial conditions x_0 such that exact tracking of an arbitrary reference trajectory is achieved,* i.e.*:*

$$\boldsymbol{e} = \boldsymbol{r} - \boldsymbol{y} = \boldsymbol{r} - (J_N \boldsymbol{u} + J_0 x_0) = 0 \tag{5.9}$$

To prove this, (5.9) is rewritten into:

$$\boldsymbol{e} = \boldsymbol{r} - \left[\begin{array}{c|c} J_N & J_0 \end{array}\right] \left[\begin{array}{c} \boldsymbol{u} \\ \hline x_0 \end{array}\right] = 0 \tag{5.10}$$

The loss of rank of J_N due to NMP zeros equals the number of NMP zeros n^z_{nmp}. On the other hand, adding the initial conditions of the system to (5.6) adds n variables, while keeping the number of equations equal. Hence, the matrix $\left[\begin{array}{c|c} J_N & J_0 \end{array}\right]$ in (5.9) is rectangular with more columns than rows. For proper systems, it holds that $n \geq n^z_{\text{nmp}}$, so there exist $\boldsymbol{u}$ and x_0 such that (5.10) is satisfied. Note that this solution is generally not unique. An approach to compute a unique solution for $\boldsymbol{u}$ without explicitly calculating x_0 is discussed in Section 5.2.4.

This proves that for every system there exist a bounded feedforward signal $\boldsymbol{u}$ and initial conditions x_0 such that exact tracking is achieved, *i.e.*, $\boldsymbol{y} = \boldsymbol{r}$ (see Figure 5.4). To start tracking a reference trajectory with initial conditions x_0, however, the system has to be brought from $x = 0$ to $x = x_0$ using pre-actuation during a bounded interval. Therefore, it is important to know whether every x_0 can actually be reached while keeping $y = 0$, which is discussed in the next section.

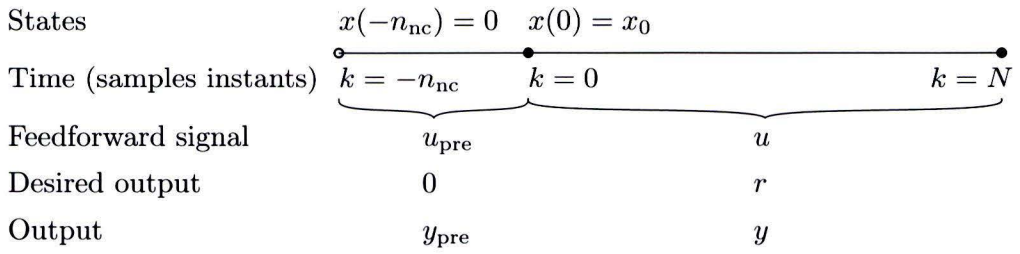


Figure 5.4: A timeline to illustrate pre-actuation and initial conditions (n_{nc} denotes the number of pre-actuation samples).

5.2.3 Pre-actuation

As explained in Section 5.2.2, pre-actuation should be used to bring a system to suitable initial conditions at $k = 0$. To analyse this in the lifted domain, two matrices are defined. First, a matrix is defined which maps the input signal onto the states of the system. Essentially, this equals a convolution matrix except that the entries have not been premultiplied with the output matrix C:

$$\begin{bmatrix} x(-n_{\text{nc}}+1) \\ x(-n_{\text{nc}}+2) \\ \vdots \\ x(0) \end{bmatrix} = \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{n_{\text{nc}}-1}B & A^{n_{\text{nc}}-2}B & \cdots & B \end{bmatrix} \begin{bmatrix} u(-n_{\text{nc}}+1) \\ u(-n_{\text{nc}}+2) \\ \vdots \\ u(0) \end{bmatrix} \tag{5.11}$$

$$\boldsymbol{x}_{\text{pre}} = J_I \boldsymbol{u}_{\text{pre}}$$

This matrix maps the input $\boldsymbol{u}_{\text{pre}}$ onto the states on the entire pre-actuation interval of n_{nc} samples. However, only $x(0)$ and therefore the bottom n rows of J_I are of interest:

$$\begin{aligned} x_0 &= \begin{bmatrix} A^{n_{\text{nc}}-1}B & A^{n_{\text{nc}}-2}B & \cdots & B \end{bmatrix} \begin{bmatrix} u(-n_{\text{nc}}+1) \\ u(-n_{\text{nc}}+2) \\ \vdots \\ u(0) \end{bmatrix} \\ &= J_{I,0}\boldsymbol{u}_{\text{pre}} \end{aligned} \tag{5.12}$$

Note that $J_{I,0}$ resembles a reversed extended controllability matrix, *i.e.*, the controllability matrix of a state-space system is defined as the right $m \times n$ columns of $J_{I,0}$ in reverse order. If the underlying system is controllable, the rank of $\text{rank}(J_{I,0}) = n$.

It appears that it is not possible to bring $P(z)$ to $x(k = 0) = x_0$ on a bounded pre-actuation interval while keeping the output zero (see also Zou (2009)):

Theorem 5.2 *A system $P(z)$ cannot be brought from $x(k = -n_{\text{nc}}) = 0$ to an arbitrary $x(k = 0) = x_0$ on a bounded interval $k \in [-n_{\text{nc}}, 0]$ with a bounded input signal $\boldsymbol{u}_{\text{pre}}$ while keeping its output $\boldsymbol{y}_{\text{pre}}$ equal to zero,* i.e., *it is not possible to compute a $\boldsymbol{u}_{\text{pre}}$ satisfying both:*

$$J_{I,0}\boldsymbol{u}_{\text{pre}} = x_0 \tag{5.13}$$

and

$$\boldsymbol{y}_{\text{pre}} = J_{n_{\text{nc}}}\boldsymbol{u}_{\text{pre}} = \boldsymbol{0} \tag{5.14}$$

where $J_{n_{\text{nc}}}$ denotes a convolution matrix of size n_{nc}.

To prove this, the two equations in Theorem 5.2 are written into one matrix equation:

$$\left[\begin{array}{c} J_{n_{\text{nc}}} \\ \hline J_{I,0} \end{array}\right] \boldsymbol{u}_{\text{pre}} = \left[\begin{array}{c} \boldsymbol{0} \\ \hline x_0 \end{array}\right] \tag{5.15}$$

The matrix in this equation is rectangular with more rows than columns, hence it can be seen as having $n_{\text{nc}} + n$ equations for n_{nc} unknowns. Contrary to the situation in the previous section, there is no $\boldsymbol{u}_{\text{pre}}$ satisfying this equation. However, if a large amount of pre-actuation is utilized, *i.e.*, $n_{\text{nc}} \gg n$, then $n_{\text{nc}} + n \approx n_{\text{nc}}$ and a solution which only introduces a small error can be found. This qualitatively shows that the states x of a system $P(z)$ can be brought from $x(k = -n_{\text{nc}})$ to $x(0) = x_0$ while keeping $y(k \leq 0) \approx 0$ if n_{nc} is selected sufficiently large.

Next, it will be shown how a quantitative measure on the total error can be given if pre-actuation is taken into account implicitly, *i.e.*, without explicitly computing the initial conditions.

5.2.4 Implicit Pre-actuation

In Section 5.2.2 it is shown that it is possible to compute a feedforward signal $\boldsymbol{u}$ and initial conditions x_0 such that exact tracking is achieved. However, in Section 5.2.3 it appeared that it is not possible to bring system $P(z)$ into state x_0 using pre-actuation on a limited interval while keeping $y(k \leq 0) = 0$. In this section, the performance that can be achieved when pre-actuation is implicitly added to (5.6) is investigated. This implies that pre-actuation is used, but no x_0 is explicitly computed. Essentially, this means that one feedforward signal $[\boldsymbol{u}_{\text{pre}}^T, \boldsymbol{u}^T]^T$ is computed for the entire interval $k \in [-n_{\text{nc}}, N]$.

Hereto, the expressions for $\boldsymbol{y}$ in (5.9) and $\boldsymbol{y}_{\text{pre}}$ in (5.14) are rewritten into one matrix equation:

$$\left[\begin{array}{c} \boldsymbol{y}_{\text{pre}} \\ \hline \boldsymbol{y} \end{array}\right] = \left[\begin{array}{c} J_{n_{\text{nc}}}\boldsymbol{u}_{\text{pre}} \\ \hline J_N\boldsymbol{u} + J_0 x_0 \end{array}\right] = \left[\begin{array}{c|c} J_{n_{\text{nc}}} & 0 \\ \hline J_0 J_{I,0} & J_N \end{array}\right] \left[\begin{array}{c} \boldsymbol{u}_{\text{pre}} \\ \hline \boldsymbol{u} \end{array}\right] \tag{5.16}$$

In this equation, the product $J_0 J_{I,0}$ is given by:

$$\begin{aligned} J_0 J_{I,0} &= \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{N-1} \end{bmatrix} \begin{bmatrix} A^{n_{\mathrm{nc}}-1}B & A^{n_{\mathrm{nc}}-2}B & \cdots & B \end{bmatrix} \\ &= \begin{bmatrix} CA^{n_{\mathrm{nc}}-1}B & CA^{n_{\mathrm{nc}}-2}B & \cdots & CB \\ CA^{n_{\mathrm{nc}}}B & CA^{n_{\mathrm{nc}}-1}B & \cdots & CAB \\ \vdots & \vdots & \ddots & \vdots \\ CA^{N+n_{\mathrm{nc}}-2}B & CA^{N+n_{\mathrm{nc}}-3}B & \cdots & CA^{N-1}B \end{bmatrix} \end{aligned} \tag{5.17}$$

This shows that the matrix of (5.16) can be rewritten:

$$\left[\begin{array}{c|c} J_{n_{\mathrm{nc}}} & 0 \\ \hline J_0 J_{I,0} & J_N \end{array}\right] = J_{N+n_{\mathrm{nc}}} \tag{5.18}$$

which simply is a $(N+n_{\mathrm{nc}}) \times (N+n_{\mathrm{nc}})$ convolution matrix. Hence, the total output is given by:

$$\boldsymbol{y}_{N+n_{\mathrm{nc}}} = J_{N+n_{\mathrm{nc}}} \boldsymbol{u}_{N+n_{\mathrm{nc}}} \tag{5.19}$$

Subsequently, the feedforward problem implies computing the feedforward signal $\boldsymbol{u}$ that minimizes the 2-norm of the error $\boldsymbol{e}$:

$$\boldsymbol{e}_{N+n_{\mathrm{nc}}} = \boldsymbol{r}_s - J_{N+n_{\mathrm{nc}}} \boldsymbol{u}_{N+n_{\mathrm{nc}}} \tag{5.20}$$

with:

$$\boldsymbol{r}_s = \left[\frac{0^{n_{\mathrm{nc}} \times 1}}{\boldsymbol{r}}\right] \tag{5.21}$$

the shifted reference trajectory.

Similar to (5.4), the error is given by:

$$\boldsymbol{e} = U_2 U_2^T \boldsymbol{r}_s \tag{5.22}$$

where U_2 has $N+n_{\mathrm{nc}}$ rows instead of N, while the number of columns still equals the number of NMP zeros. To investigate why this shift of the reference trajectory leads to a reduced servo error, one has to take a closer look at U_2. This matrix contains the output singular vector(s) corresponding to the smallest singular values of J and hence to the NMP zeros of $P(z)$. From now on, the product $U_2 U_2^T$ will be denoted as E:

$$E = U_2 U_2^T \tag{5.23}$$

In order to quantify a lower bound on the achievable servo error, the following lemma, inspired by Dijkstra (2003), is required:

Lemma 5.1 *Consider a vector $U_2 = \alpha z_u^{-k}$ with $|z_u| > 1$, $k \in [1, N+n_{\mathrm{nc}}]$ and α such that $\|U_2\|_2 = 1$, which is the basic shape for a system with one NMP zero (see Appendix B). Then, $\|E\|_2 = \|U_2 U_2^T\|_2 = 1$. If E is partitioned:*

$$E = \left[\begin{array}{c|c} E_{\mathrm{pre}} & E_{\mathrm{res}} \end{array}\right] \tag{5.24}$$

with E_{pre} the first n_{nc} columns of E and E_{res} the remaining N columns of E, then

$$\|E_{\mathrm{res}}\|_2 \leq \alpha z_u^{-n_{\mathrm{nc}}} \tag{5.25}$$

To prove this, it should be realized that the ith column of E equals the ith entry of U_2 times U_2. As a result, the 2-norm of the ith column of E equals the ith entry of U_2, since $\|U_2\|_2 = 1$. Now, the 2-norm of E is given by:

$$\|E\|_2 = \sqrt{\sum_{k=1}^{N+n_{\text{nc}}} \left(\alpha z_u^{-k}\right)^2} \tag{5.26}$$

Furthermore, the 2-norm of E_{res} can be written as:

$$\begin{aligned}
\|E_{\text{res}}\|_2 &= \sqrt{\sum_{k=1+n_{\text{nc}}}^{N+n_{\text{nc}}} \left(\alpha z_u^{-k}\right)^2} \\
&= \sqrt{\left(\alpha z_u^{-n_{\text{nc}}}\right)^2 \sum_{k=1}^{N} \left(\alpha z_u^{-k}\right)^2} \\
&= \alpha z_u^{-n_{\text{nc}}} \sqrt{\sum_{k=1}^{N} \left(\alpha z_u^{-k}\right)^2} \\
&< \alpha z_u^{-n_{\text{nc}}} \sqrt{\sum_{k=1}^{N+n_{\text{nc}}} \left(\alpha z_u^{-k}\right)^2} \\
&< \alpha z_u^{-n_{\text{nc}}} \|E\|_2 \\
&< \alpha z_u^{-n_{\text{nc}}}
\end{aligned} \tag{5.27}$$

This leads to the following theorem:

Theorem 5.3 *If a system $P(z)$ contains one non-minimum phase zero at $z = z_u$, the minimum of the 2-norm of the servo error $\|\boldsymbol{e}\|_2$ is bounded by $\alpha z_u^{-n_{\text{nc}}}\|\boldsymbol{r}\|_2$, where n_{nc} denotes the number of preview samples.*

To prove this, (5.22) is rewritten:

$$\boldsymbol{e} = E\boldsymbol{r}_s = \left[\begin{array}{c|c} E_{\text{pre}} & E_{\text{res}} \end{array}\right] \left[\begin{array}{c} 0^{n_{\text{nc}}\times 1} \\ \hline \boldsymbol{r} \end{array}\right] \tag{5.28}$$

Next, using the triangle inequality, it appears that:

$$\|E\boldsymbol{r}_s\|_2 = \|E_{\text{pre}}0^{n_{\text{nc}}\times 1} + E_{\text{res}}\boldsymbol{r}\|_2 \leq \|E_{\text{pre}}0^{n_{\text{nc}}\times 1}\|_2 + \|E_{\text{res}}\boldsymbol{r}\|_2 = \|E_{\text{res}}\boldsymbol{r}\|_2 \tag{5.29}$$

The proof is completed by using Lemma 5.1:

$$\|E_{\text{res}}\boldsymbol{r}\|_2 \leq \|E_{\text{res}}\|_2\|\boldsymbol{r}\|_2 \leq \alpha z_u^{-n_{\text{nc}}}\|\boldsymbol{r}\|_2 \tag{5.30}$$

In theory, this analysis can be extended for systems with multiple (complex) NMP zeros. This is subject to further research.

Remark 5.1 *If the rank loss of a convolution matrix J_N equals the number of NMP zeros n^z_{nmp}, one could see pre-actuation as simply adding columns on the left side of J_N until its*

column rank equals N. This way, exact tracking on $k \in [k_0, k_f]$ is achieved with only n^z_{nmp} preview samples. In that case, however, the output y during the pre-actuation interval is not taken into account. This may lead to undesirable values of the output. Furthermore, the feedback controller which is usually present in practical applications will respond to this excitation of the output. Therefore, it is essential to take the output y during the pre-actuation interval into account.

5.2.5 Summary

Summarizing, the use of pre-actuation is supposed to overcome the performance limitations of non-minimum phase systems:

- In Section 5.2.2 it is shown that a non-minimum phase system can indeed achieve exact tracking of an arbitrary reference trajectory if the right initial conditions and a corresponding feedforward signal are used.
- Nevertheless, it is not possible to use pre-actuation on a bounded interval to bring a system $P(z)$ from $x(-n_{\text{nc}}) = 0$ to $x(0) = x_0$ while keeping the output equal to zero, as discussed in Section 5.2.3.
- However, if sufficient pre-actuation time is used, the lower bound of the servo error decreases depending on the locations of the NMP zeros, see Section 5.2.4.

In, *e.g.*, Jemaa and Davison (2003); Ooi et al. (2006); Qiu and Davison (1993), time and frequency domain integral constraints have been used to show that the presence of non-minimum phase zeros imposes inherent performance limitations on the achievable servo performance in the context of feedforward control. In these references, however, it is assumed that no preview or pre-actuation is employed. In Middleton et al. (2001, 2004), these limitations are extended to the preview control case. One of the most important results in Middleton et al. (2004) is that the achievable weighted $\mathcal{H}_\infty$ performance is approximately proportional to $e^{-\zeta T_{\text{pre}}}$, where ζ denotes the smallest real part of any NMP (continuous time) zero and T_{pre} is the available preview time (note the correspondence between $e^{-z_c T_s}$ (with z_c a continuous time zero) and $z_d^{-n_{\text{nc}}}$ (with z_d a discrete time zero)). Furthermore, it is shown how a small undershoot during the pre-actuation period can prevent large transient errors. For a formal proof of these results and further performance limitations in both time and frequency domain the reader is referred to Middleton et al. (2004).

5.3 Linear Time Invariant Lifted Feedforward

In the previous sections it appeared that feedforward in the lifted domain may be Linear Time Varying (LTV), since $J^\dagger$ is not Toeplitz in case of a non-minimum phase system. It is, however, possible to obtain LTI feedforward in the lifted domain. The advantage of LTI filters is that they can easily be implemented online as FIR filters. To enforce LTI solutions, two different approaches can be used:

- Compute a larger impulse response matrix and use the center part to compute the feedforward signal.
- Make use of the commutative property of a convolution.

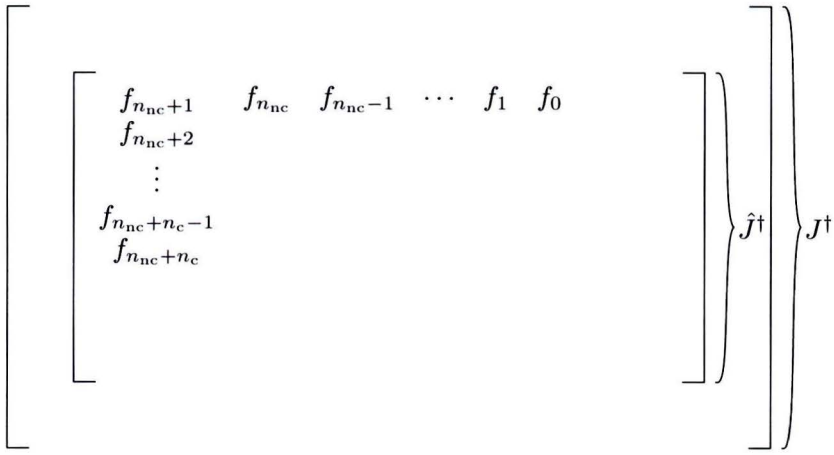


Figure 5.5: The entries of a truncated pseudo-inverse of a convolution matrix can be implemented as a FIR filter.

The first approach is more straightforward and pragmatic, while the latter method is theoretically better funded and offers more possibilities, *e.g.*, to include basis functions. Pre-actuation can be applied in both cases, as will be shown. Both approaches are elaborated below.

5.3.1 Truncated Convolution Matrices

In Figure 5.2 the magnitudes of the diagonal entries of the generalized inverse $J^\dagger$ of a 100×100 and a 200×200 impulse response matrix J of the NMP flexible cart system are plotted. It can be seen that the time varying behavior exhibits itself mainly near the start and the end of the trajectory, which is emphasized by the difference of the 100×100 and 200×200 $J^\dagger$ examples. This is explained by the fact that convolution matrices of finite sizes are used.

The most straightforward way to obtain an LTI filter is to use the center part of the $J^\dagger$ to compute $\boldsymbol{u}$. In Figure 5.2, it can be seen that the middle part $\hat{J}^\dagger$ of $J^\dagger$, *e.g.*, between 30 and 170, is Toeplitz ($J^\dagger(k,k)$ is approximately constant for $k \in [30, 170]$) and thus LTI.

Now, if one wants to implement a FIR filter of n_{nc} preview samples and n_{c} causal samples, one has to take the first n_{nc} entries of the first row and the first n_{c} entries of the first column of $\hat{J}^\dagger$ and implement them as one FIR filter (see Figure 5.5). Note that the reference trajectory $r(k)$ has to be delayed with n_{nc} samples to synchronize $r(k)$ and the feedforward signals $u_{\mathrm{ff}}(k)$. According to Heertjes and Van de Molengraft (2009), choosing n_c and n_{nc} is often a matter of trial and error: choosing the filter order too small limits the ability to describe the inverse dynamics, while choosing it too large may result in overfitting.

5.3.2 Commutation of Convolution

The approach to obtain FIR filters introduced in the previous section is somewhat pragmatic. Another way of computing FIR filters follows from utilizing the commutative property of convolution. This property essentially means that the order of the convolution operands does not affect the result of the convolution. Next, it is discussed how commutation can

be used to obtain LTI feedforward. Thereafter, it is discussed how pre-actuation can be achieved and finally it is briefly discussed how basis functions can be employed to enforce other properties.

Concept

Essentially, the concept introduced in Section 5.1 boils down to computing a convolution matrix F and subsequently a feedforward signal $\boldsymbol{u} = F\boldsymbol{r}$. In Section 5.1, $F = J^{\dagger}$ such that the 2-norm of

$$\boldsymbol{e} = \boldsymbol{r} - \boldsymbol{y} = \boldsymbol{r} - JF\boldsymbol{r} \tag{5.31}$$

is minimized. Using the commutative property, this can be rewritten as minimizing

$$\boldsymbol{e} = \boldsymbol{r} - JR\boldsymbol{f} \tag{5.32}$$

since:

$$\begin{bmatrix} f_1 & 0 & \cdots & 0 \\ f_2 & f_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ f_N & f_{N-1} & \cdots & f_1 \end{bmatrix} \begin{bmatrix} r(1) \\ r(2) \\ \vdots \\ r(N) \end{bmatrix} = \begin{bmatrix} r(1) & 0 & \cdots & 0 \\ r(2) & r(1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ r(N) & r(N-1) & \cdots & r(1) \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix} \tag{5.33}$$

Similar to Section 5.1, this can be solved for $\boldsymbol{f}$ using the pseudo-inverse, *i.e.*:

$$\boldsymbol{f} = (JR)^{\dagger}\,\boldsymbol{r} \tag{5.34}$$

In this case, the reference trajectory $\boldsymbol{r}$ and hence also the matrix R represent an auxiliary trajectory which is only used to compute the FIR parameters, which actually are supposed to be trajectory independent. However, using a suitable $\boldsymbol{r}$ can be shown to have a significant effect on the obtained result. Similar to system identification, $\boldsymbol{r}$ should be informative enough (Pintelon and Schoukens, 2001), hence one could use (white) noise, a chirp signal or multisines. On the other hand, better results may be achieved if the auxiliary trajectory closely resembles realistic reference trajectories. However, choosing a suitable auxiliary $\boldsymbol{r}$, is is not further discussed in this thesis, since it is a system identification rather than a feedforward issue and this approach will not be used in the remainder of this report.

The FIR parameters $\boldsymbol{f}$ resulting from (5.34) can either be implemented online as a FIR filter or the commutative property can be used to compute F and subsequently $\boldsymbol{u}$ offline. The length of the FIR filter is determined by the number of columns of (JR) that is used. This technique has a strong resemblance with Heertjes and Van de Molengraft (2009), where an ILC signal is mapped onto the optimal FIR parameters.

Pre-actuation

In case of a non-minimum phase system, the approach in Section 5.1 results in feedforward that is not only LTV, but also non-causal. Pre-actuation can also be achieved using the LTI

approach described here. In this case, the difference between the output $\boldsymbol{y}$ and a shifted reference signal should be minimized, similar to pre-actuation in Section 5.2. Hence:

$$\boldsymbol{f} = (JR)^{\dagger}\,\boldsymbol{r}_s \tag{5.35}$$

Here, J is the $(N + n_{\text{nc}}) \times (N + n_{\text{nc}})$ convolution matrix, R is the matrix containing the original reference trajectory $\boldsymbol{r}$ and $\boldsymbol{r}_s$ is the shifted reference signal. Contrary to Sections 5.1 and 5.2, $\boldsymbol{r}$ and $\boldsymbol{r}_s$ have equal length in this case. The final n_{nc} entries of $\boldsymbol{r}$ are equal, while $\boldsymbol{r}_s$ contains n_{nc} leading zero entries.

Basis Functions

A further expansion to this approach is to introduce basis functions. In Hennekens (2009), a connection between the FIR coefficients and the gains of time derivatives of the input signal is established. Here, the polynomial $(z-1)^n$ plays an important role, since it directly connects FIR coefficients to velocity, acceleration, jerk and snap feedforward gains. This knowledge can be used to define basis functions for the impulse response parameters. These can be used to enforce certain properties of the resulting FIR filter.

The FIR parameters can be expressed by:

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots \\ 0 & -1 & -2 & \cdots \\ 0 & 0 & 1 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \vdots \end{bmatrix} \tag{5.36}$$
$$\boldsymbol{f} = M\boldsymbol{\alpha}$$

Now, the feedforward gains $\boldsymbol{\alpha}$ can be computed by changing (5.35) into:

$$\boldsymbol{\alpha} = (JRM)^{\dagger}\,\boldsymbol{r}_s \tag{5.37}$$

and the resulting FIR parameters are computed according to (5.36).

Example 5.1 *Take,* e.g., *a moving mass. In case this system is in rest, there should be no feedforward force acting on the mass. This is the case if the sum of the FIR parameters is zero, which can be obtained by discarding the first column of matrix M.*

5.3.3 Summary

In the lifted domain, LTI feedforward can be obtained by either using a truncated inverse convolution matrix or using the commutative property of convolution. Using the latter approach, a number of design choices have to be made:

- A suitable number of causal and non-causal FIR parameters have to be selected.
- The size N of the convolution matrices used is important. In theory, increasing N leads to a more accurate computation of $\boldsymbol{f}$. However, it also increases the condition number of JR so choosing it too large results in numerical inaccuracies (see also Section 5.5). Therefore, a suitable trade-off has to be made.

- The final design choice regards the auxiliary reference trajectory ($\boldsymbol{r}$, $\boldsymbol{r}_s$ and R), which is used for the computation of $\boldsymbol{f}$. This shows strong resemblance to designing an excitation signal for system identification. Therefore, one could think of, *e.g.*, (white) noise, multisines or a chirp signal. In practice, however, these will not be used as reference trajectories, so it could also be argued that realistic trajectories or (filtered) step responses are used. These, on the other hand, may not be informative enough.

Despite the increased implementation possibilities, the LTI lifted approach is not further elaborated in this thesis. In the next chapter, stable inversion feedforward will be introduced. Since this is LTI, it chosen to use prefer the LTV approach of Section 5.1 above the LTI concept of Section 5.3, hence enabling the possibility to compare time varying and time invariant feedforward. Furthermore, a drawback of the lifted LTI method is that there are too many design choices involved, so the obtained performance depends to a large extent on 'tuning' of the parameters. Especially designing a suitable auxiliary $\boldsymbol{r}$ is essential for the end result, but this is more a system identification issue and therefore beyond the scope of this thesis.

5.4 Uncertainty

All concepts in this chapter so far assume the model is an exact representation of the system dynamics. Nevertheless, this is never the case in practice. In this section, robust feedforward in the lifted domain is discussed.

In Section 4.4 it is argued that for robust feedforward design better results may be obtained if it is optimized for the entire model set $\mathcal{P}$ instead of optimizing the worst case performance. Here, the large design flexibility of the lifted system description proves to be very useful.

The main idea behind robust feedforward in the lifted domain is to stack a number n_J of convolution matrices of random realizations of $\mathcal{P}$ such that the total matrix represents the entire model set. Subsequently, the least-squares solution to the stacked matrix equation is computed by:

$$\boldsymbol{u} = \mathcal{J}^{\dagger}\underline{\boldsymbol{r}_s} \tag{5.38}$$

where

$$\mathcal{J} = \begin{bmatrix} J_1^T & J_2^T & \cdots & J_{n_J}^T \end{bmatrix}^T$$

$$\underline{\boldsymbol{r}_s} = \begin{bmatrix} \boldsymbol{r}_s^T & \boldsymbol{r}_s^T & \cdots & \boldsymbol{r}_s^T \end{bmatrix}^T$$

In this equation, n_J has to be selected such that $\mathcal{J}$ is sufficiently representative for the entire model set. No direct guidelines of how to select n_J are available. A pragmatic approach is to increase n_J incrementally and compare the resulting $\boldsymbol{u}$ with the previous one. If the resulting $\boldsymbol{u}$ has converged to its optimal value for the entire model set, it does not change significantly if n_J is increased any further. It is assumed that the realizations are equally distributed over $\mathcal{P}$.

The n_J impulse response matrices can be obtained either directly by doing multiple identification experiments, *e.g.*, with different input signals or at different locations, or by fitting an overbound and taking random samples from the resulting uncertainty set. The first method omits the intermediate step of fitting an overbound, while the advantage of the latter method

is that no additional experiments are required if n_J is increased. An indirect benefit of this method is that it can be used incrementally to determine a suitable n_J: one keeps adding impulse response matrices until $\boldsymbol{u}$ does not change significantly anymore, as discussed above. Since increasing n_J also increases the size of $\mathcal{J}$, numerical issues become more important. These are addressed in the next section.

5.5 Numerical Issues

While the lifted system description has advantages over other system representations, there is one serious drawback: numerical implementation. If the length of the trajectory increases, so does the size of J. Furthermore, if feedforward signals for MIMO systems are computed or if robust feedforward is required, multiple impulse response matrices are stacked. According to Heath (2002), the required number of computations to calculate the SVD of an $m \times n$ matrix J is proportional to $mn^2 + n^3$. This shows that computation time quickly increases for larger trajectories, MIMO systems and uncertain systems.

Note that there are other methods than the singular value decomposition to calculate a least squares solution. For example, the MATLAB command `mldivide` selects one out of ten alternative algorithms, depending on the structure of the matrix. Computing the singular value decomposition and subsequently the generalized inverse, however, enables the possibility to explicitly discard those singular values that are associated with the non-minimum phase zeros. Furthermore, the SVD method shows superb robustness and reliability (Heath, 2002).

The latter is important since the condition number of J is another issue, besides its size. The rigid body modes, which are commonly present in the systems under consideration, cause the impulse response to increase linearly with time after the (assumed stable) flexible modes have converged. This causes the condition number of the Toeplitz impulse response matrix to increase dramatically. Take, *e.g.*, the minimum phase flexible cart system, which has no small singular values due to NMP zeros (and thus only 'clustered' singular values). In Figure 5.6 the condition number is plotted for various sizes of J and it appears that this also increases quickly with increasing size of J. The latter issue can be resolved if the impulse response matrix is computed for a system with acceleration as output rather than position, *i.e.*, a system with the rigid body dynamics removed. Subsequently, the feedforward signal can be calculated by using the reference acceleration instead of the reference position. In practice, however, it is common to measure position rather than acceleration and the systems in this report therefore represent the transfer from force to position. Next, the issues discussed in this chapter are illustrated by means of an example.

5.6 Example

5.6.1 Introduction

The system used in this example is the flexible cart system, see Section 3.4.1, with the difference that the damping is changed to $d = 10$ Ns/m. This leads to the Bode-diagrams in Figure 5.7. The continuous time NMP system has two poles at $s = 0$ due to the rigid body

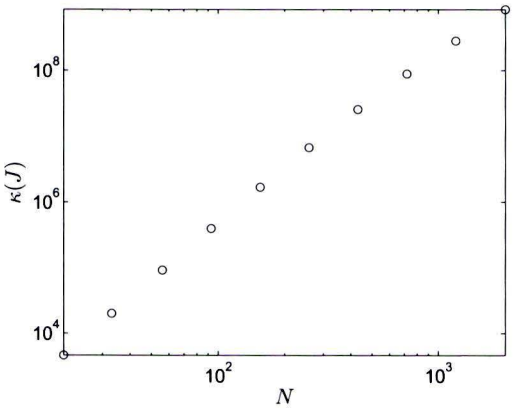


Figure 5.6: Condition number of the impulse response matrix of the minimum phase flexible cart system.

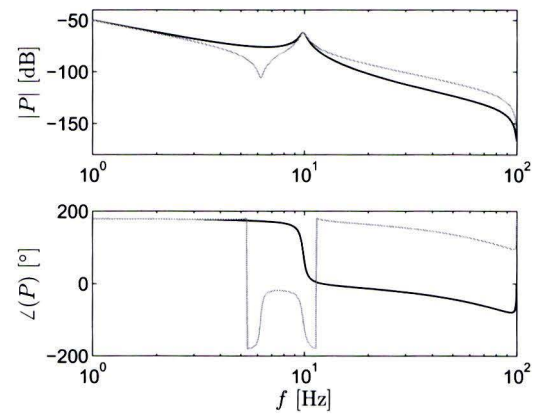


Figure 5.7: Bode-diagrams of $P_1(z)$ (black) and $P_2(z)$ (gray).

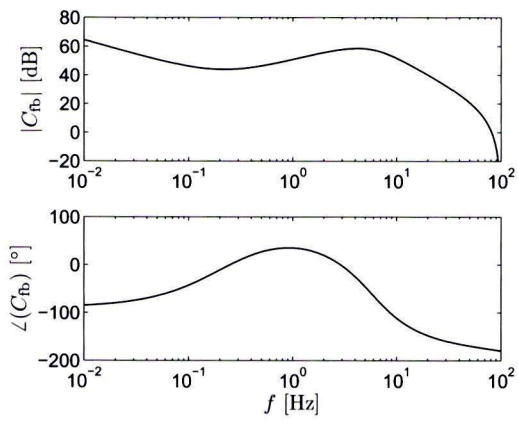


Figure 5.8: Bode-diagram of the feedback controller.

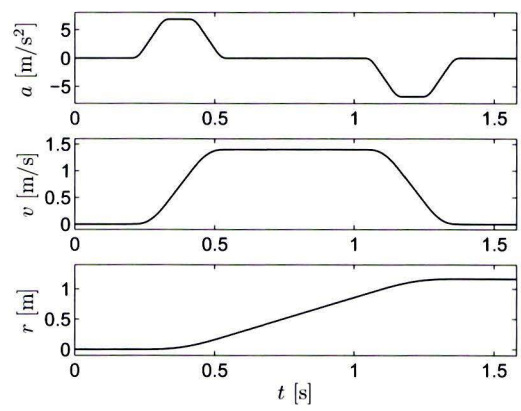


Figure 5.9: The reference acceleration (upper plot), velocity (middle plot) and position (lower plot) for the flexible cart system.

mode. Furthermore, the flexible mode, having a resonance frequency of 9.9 Hz, introduces two stable poles at $s = -1.92 \pm 62.0j$. The zeros of the NMP system are located at $s = -81.0$ and $s = 88.2$.

Including the ZOH and sampler with a sampling frequency of $T_s = 0.005$ s, a discrete representation is obtained. Then, the poles of the rigid body mode are located at $z = 1$ and the poles of the flexible mode at $z = 0.94 \pm 0.30j$. The zeros are mapped to $z = 0.67$ and $z = 1.55$. In addition, a sampling zero is introduced at $z = -0.98$.

In Figure 5.7, the Bode-diagram of the minimum phase system is also plotted. This has its poles at the same location, but the zeros are located at $s = 0.76 \pm 38.9j$ (continuous time) and $z = 0.98 \pm 0.19j$ and $z = -1.00$ (discrete time).The latter zero is a sampling zero. Next to the resonance, the MP flexible cart also has an anti-resonance, which is located at 6.2 Hz.

In order to have a stable closed loop, the feedback controller displayed in Figure 5.8 is used.

It consists of a lead-lag filter, an integrator and a low-pass filter, leading to a bandwidth of 1 Hz. It is decided to use only a 1 Hz bandwidth controller to emphasize the performance of feedforward control.

5.6.2 Nominal Feedforward

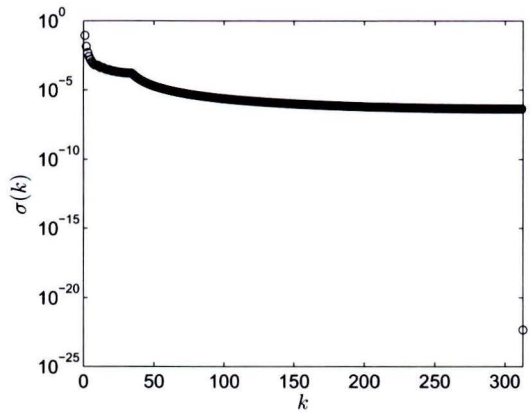


Figure 5.10: Singular values of the 313×313 convolution matrix of the NMP flexible cart system. The first 312 singular values are 'clustered' and $\sigma_{313} \approx 0$ due to the single NMP zero.

In this example, the flexible cart system is supposed to track a fourth order reference trajectory, see Figure 5.9. Including 0.2 s pre- and post-actuation, this reference trajectory counts 313 samples. The corresponding singular values of the 313×313 impulse response matrix are plotted in Figure 5.10. As expected, $\sigma_{313} \approx 0$ due to the presence of a single NMP zero. This singular value is discarded when the generalized inverse is computed, and the resulting $J^{\dagger}$ is neither lower triangular nor Toeplitz as discussed in Section 5.1.

Using $J^{\dagger}$, the nominal feedforward signal is calculated and plotted in Figure 5.11. To a certain extent, the feedforward signal resembles the (scaled) acceleration signal but is somewhat smoother. Especially in the lower plot of Figure 5.11 it can be seen that u_{ff} increases before the acceleration is unequal to zero. This is the pre-actuation, discussed in Section 5.2, to enable exact tracking in the presence of NMP zeros. Likewise, the end of the feedforward signal is also smoother than the reference acceleration. Since the zeros are almost equal but of opposite sign ($s = -81 \approx -s = -88$), the symmetric increase and decrease of reference acceleration seems to result in a symmetric increase and decrease of u_{ff} as well. Closer examination, however, shows that this is not the case. Since there are only real zeros, the pre- and post-actuation do not show oscillatory behavior.

The error after simulating this feedforward signal with the nominal system is shown in Figure 5.12. A servo error in the order of magnitude of 10^{-12} m is very small compared to the velocity and traveled distance. The most striking about the error plot is the sharp peak at the start of the interval. This can be attributed to the finite preview time: to compensate for this, the feedforward signal shows a minor peak of 1×10^{-6} N. Had more preview been

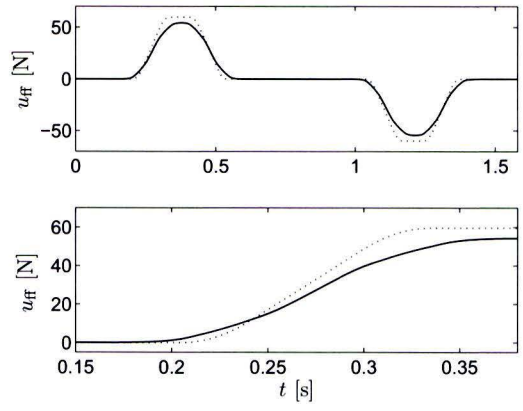


Figure 5.11: Scaled acceleration (dotted) and feedforward signal (solid) resulting from lifted feedforward. The lower plot is a zoom of the upper plot, showing that u_{ff} starts increasing before the reference acceleration.

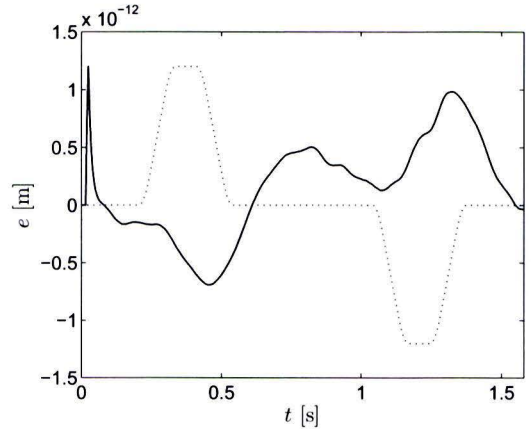


Figure 5.12: Scaled acceleration (dotted) and error (solid) using the feedforward signal of Figure 5.11.

available, this peak would have been even smaller and the resulting servo error would also have decreased. This is further investigated in the next section.

According to Section 5.2, the states of the system do not equal zero at x_0. In this case, $x_{cg} = 4.0 \times 10^{-5}$ m, $\dot{x}_{cg} = 3.4 \times 10^{-3}$ m/s, $\phi = -7.9 \times 10^{-4}$ rad and $\dot{\phi} = -6.8 \times 10^{-2}$ rad/s. Nevertheless, this combination of x_{cg} and ϕ only introduces a servo error in the order of magnitude of 10^{-12} m.

5.6.3 Pre- and Post-Actuation

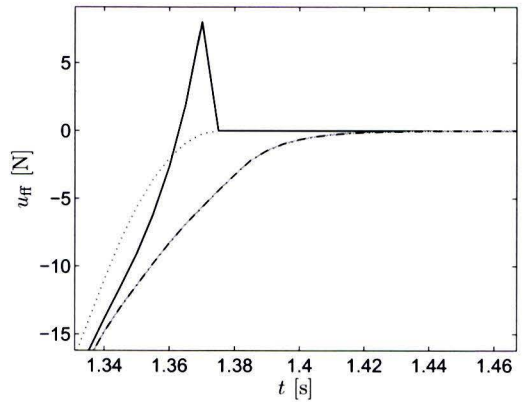


Figure 5.13: Scaled acceleration (black, dotted) and u_{ff} using $T_{post} = 0.0$ s (black, solid), $T_{post} = 0.1$ s (gray) and $T_{post} = 0.2$ s (black, dash-dotted). $T_{pre} = 0.2$ s in all cases. The gray and the dash-dotted line can hardly be distinguished by visual inspection.

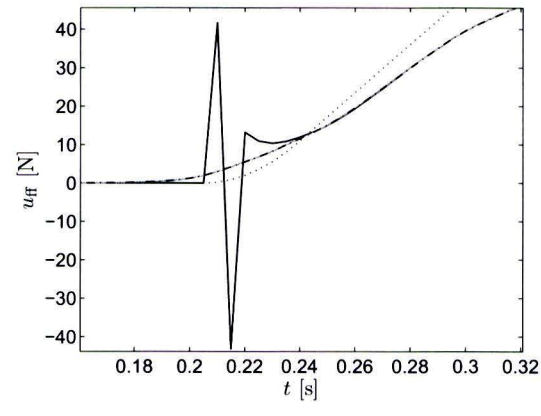


Figure 5.14: Scaled acceleration (black, dotted) and u_{ff} using $T_{pre} = 0.0$ s (black, solid), $T_{pre} = 0.1$ s (gray) and $T_{pre} = 0.2$ s (black, dash-dotted). $T_{post} = 0.2$ s in all cases. The gray and the dash-dotted line can hardly be distinguished by visual inspection.

In order to demonstrate the effect of pre- and post-actuation, simulations have been performed with different values of T_{pre} and T_{post}. Ideally, u_{ff} is only defined during the interval of the move, hence $T_{\text{post}} = T_{\text{pre}} = 0$ s. However, in Sections 3.3 and 5.2 it is argued that pre- and post-actuation is required to prevent transient errors due to NMP zeros at the start and MP zeros after the end of the move. This is illustrated in this section.

In Figures 5.11 and 5.12, pre- and post-actuation of $T_{\text{pre}} = T_{\text{post}} = 0.2$ s is used. In Figure 5.13 the effect of varying the post-actuation on the feedforward signal is shown, while in Figure 5.14 the amount of pre-actuation is varied. Varying T_{post} mainly affects the end of u_{ff}: if $T_{\text{post}} = 0$ s, u_{ff} does not converge smoothly to zero but shows an aggressive peak in opposite direction. The FF signals using $T_{\text{post}} = 0.1$ s and $T_{\text{post}} = 0.2$ s do not show any visible difference.

A similar observation is made when varying T_{pre}. In case $T_{\text{pre}} = 0$ s, sharp peaks of -40 N and 40 N are visible. Again, there is no visible difference for the feedforward signals using $T_{\text{pre}} = 0.1$ s and $T_{\text{pre}} = 0.2$ s.

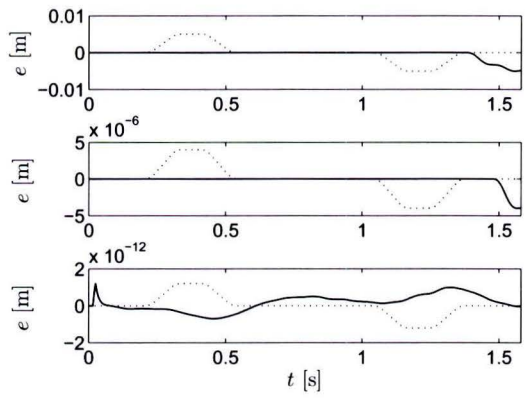


Figure 5.15: Scaled acceleration (dotted), and e using $T_{\text{pre}} = 0.2$ s and $T_{\text{post}} = 0.0$ s (upper plot), $T_{\text{post}} = 0.1$ s (middle plot) and $T_{\text{post}} = 0.2$ s (lower plot).

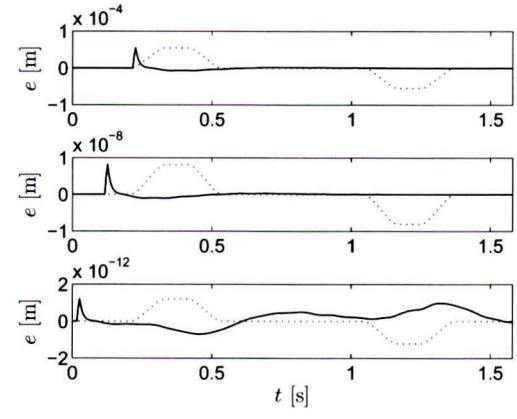


Figure 5.16: Scaled acceleration (dotted), and e using $T_{\text{pre}} = 0.0$ s and $T_{\text{post}} = 0.2$ s (upper plot), $T_{\text{pre}} = 0.1$ s (middle plot) and $T_{\text{pre}} = 0.2$ s (lower plot).

The resulting servo errors are plotted in Figures 5.15 and 5.16. In Figure 5.15, it appears that if insufficient post-actuation is applied, not all velocities are smoothly brought to zero. This is confirmed by Figure 5.17, where the angular velocity is plotted. As a result, the states do not equal zero at this point and an error is introduced after the feedforward signal has ended. This error can take substantial magnitudes.

Varying the amount of pre-actuation, a narrow peak is introduced in the error profile, at the start of the feedforward signal. If sufficient pre-actuation is used, however, this error tends to decrease and can actually be made arbitrarily small by selecting T_{pre} sufficiently large.

5.6.4 Robust Feedforward

To illustrate the use of robust feedforward, the parameters of the minimum phase flexible cart system are perturbed. It is chosen to use the minimum phase system because this shows an anti-resonance at 6.2 Hz. This is interesting because the anti-resonance causes

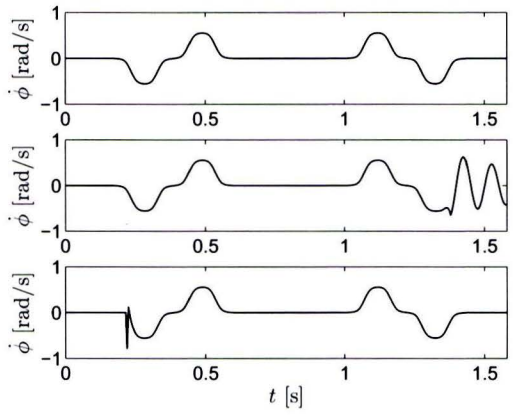


Figure 5.17: Angular velocity using $T_{\text{pre}} = T_{\text{post}} = 0.2$ s (upper plot), $T_{\text{post}} = 0$ s (middle plot), introducing oscillations after the movement and $T_{\text{pre}} = 0$ s (lower plot), introducing an error before the movement.

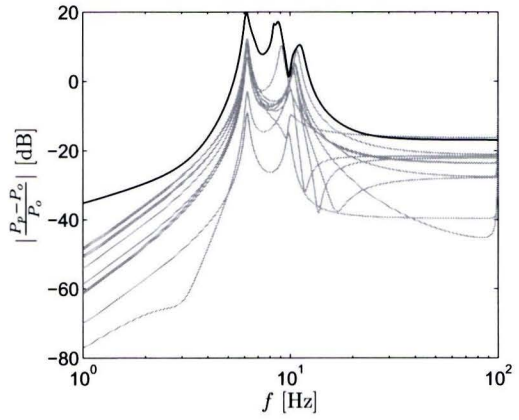


Figure 5.18: Multiplicative uncertainty of 10 random realizations of the uncertain plant (gray) and uncertainty weighting filter $w_m(z)$ (black).

a severe oscillation of the nominal u_{ff}, which can be seen in Figure 5.19. The parameters are perturbed with an uncertainty of 10% of their nominal value, with exception of the mass. This uncertainty is rather large, hence magnifying the effect of uncertainty and robust feedforward. In practice, wafer stages are expected to show much less uncertainty. In order to fit a multiplicative overbound $w_m(z)$, 10 random realizations $P_p(z)$ of the uncertain model set $\mathcal{P}$ have been taken and the multiplicative difference $\frac{P_p - P_o}{P_o}$ is computed. This is plotted in Figure 5.18.

To compute the robust feedforward signal, the convolution matrices of 10 realizations of the uncertain minimum phase system have been stacked according to Section 5.4. The resulting u_{ff} is plotted in Figure 5.19. The robust feedforward signal appears to be smoother than the nominal signal, with oscillations at the anti-resonance frequency (6.2 Hz) having a smaller amplitude. A small dip is visible at the start of the signal, again to compensate for the fact that only a limited amount of preview time is available.

With both the nominal and the robust feedforward signal, simulations have been performed with 5 different random realizations of the perturbed system. Similar to the nominal case, it is interesting to look at the states of the system to understand what is essentially happening. The states of the simulations above are plotted in Figure 5.20. The red lines denote the states of the nominal system using nominal feedforward. It appears that robust feedforward uses pre-actuation to bring the states of the system to values at the start of the interval, such that less oscillations are introduced during the remainder of the move.

Looking at the error (Figure 5.21), robust feedforward only shows a small improvement compared to nominal feedforward during the move, *i.e.*, between $t = 0.21$ s and $t = 1.38$ s. This improvement means that the amplitude of the 6.2 Hz oscillation is slightly reduced, mainly during constant velocity (lower plot). The error during the pre-actuation period ($T < 0.21$ s), however, is significant. This is confirmed by the cumulative PSD spectra (Figure 5.22): if a large mismatch between inverse plant and feedforward is present, the increase of the cu-

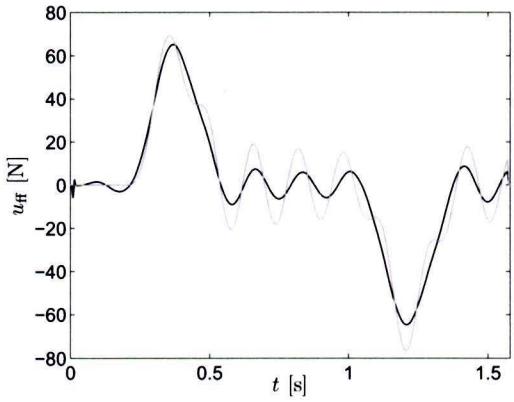


Figure 5.19: Nominal (gray) and robust (black) feedforward signal. The amplitude of the oscillation of the robust feedforward signal is much smaller than that of the nominal feedforward signal.

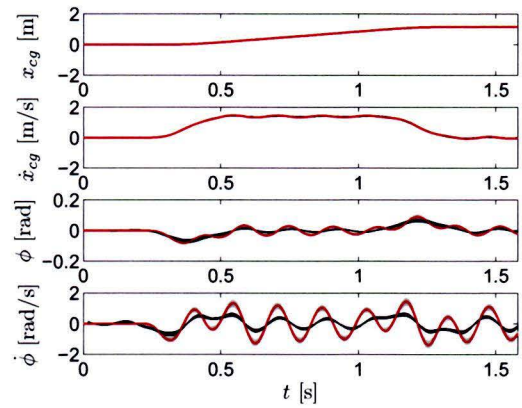


Figure 5.20: System states of 10 realizations of the uncertain system using nominal (gray) and robust (black) feedforward and states of the nominal system using nominal feedforward (red). Using robust feedforward, the states clearly show less oscillatory behavior.

mulative PSD at 6.2 Hz using nominal feedforward is larger than using robust feedforward. This manifests itself most clearly in the green case. Nevertheless, the total error using robust feedforward is slightly larger for most simulations.

5.7 Summary

Using the generalized inverse of the convolution matrix, a bounded feedforward signal can be computed for both minimum and non-minimum phase systems. A number of issues that are discussed in this chapter are:

- In case of NMP systems, the convolution matrix J has a singular value $\sigma \approx 0$ for every NMP zero. Therefore, it is undesired to use the inverse convolution matrix J^{-1} to compute the feedforward signal. If, however, the Moore-Penrose pseudo-inverse $J^{\dagger}$ is used instead, the 2-norm of the resulting servo error $\boldsymbol{e} = \boldsymbol{r} - J\boldsymbol{u}$ is minimized.
- If the Moore-Penrose pseudo-inverse is used to compute a feedforward signal for a NMP system, a bound on the achievable servo performance can be given. This bound depends on the amount of pre-actuation and the location of the NMP zeros. The feedforward signal should be calculated including the pre-actuation feedforward, *i.e.*, without explicitly computing the initial conditions at the start of the move.
- The lifted system description can be used to obtain both LTV and LTI feedforward. In order to find out the possible performance improvement through LTV feedforward, the LTV approach is used throughout this thesis.
- Uncertainty can be addressed by stacking multiple realizations of the convolution matrix. A drawback of this method is that this may result in numerical difficulties. Simulation results, however, show only a limited improvement of servo performance during

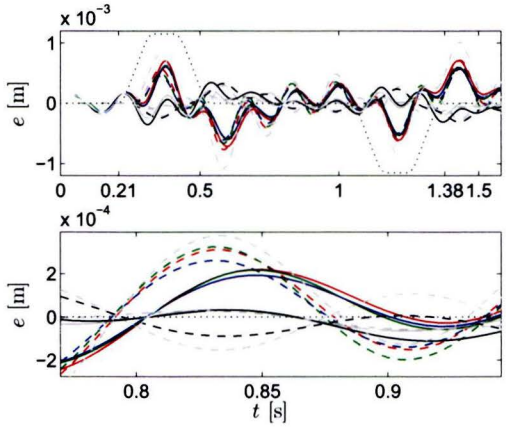


Figure 5.21: Scaled acceleration (dotted) and simulation error for 5 random realizations of the uncertain system using nominal (dashed) and robust (solid) feedforward. The lower plot is a zoom during the constant velocity interval.

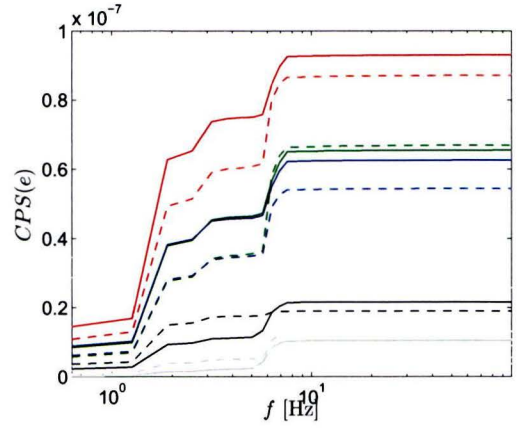


Figure 5.22: Cumulative PSD of the servo errors using nominal (dashed) and robust (solid) feedforward.

the move and a deterioration before and after the move. A general conclusion cannot be drawn yet.

- Numerical issues are important in the lifted domain and may be a limiting factor. In case of stacking due to MIMO systems and / or uncertainty, matrix sizes tend to increase and also the condition number increases dramatically. The latter could be improved by using systems with acceleration as output rather than position.

CHAPTER 6

Stable Inversion

One of the main reasons why the exact inverse of the plant model is hardly ever used for feedforward is the possible presence of non-minimum phase zeros, since these lead to an unstable inverse system. In this chapter, it is discussed how this inverse model can still be used for feedforward control design and how stable inversion leads to bounded feedforward signals (Section 6.1). The connection between stable inversion and the lifted domain concepts in Chapter 5 is discussed in Section 6.2. Furthermore, since the model does not exactly resemble the actual plant, uncertainty should be addressed, which is discussed in Section 6.3. Finally, the use of stable inversion feedforward is illustrated by means of an example in Section 6.4.

6.1 Stability, Causality and Boundary Conditions

6.1.1 Concept

Consider the state-space representation of the feedforward controller $C_{\text{ff}}(z) = P^{-1}(z)$ for a square system:

$$\begin{aligned} x(k+1) &= Ax(k) + Br(k) \\ u(k) &= Cx(k) + Dr(k) \end{aligned} \tag{6.1}$$

with $r(k) \in \mathbb{R}^m$ the reference trajectory, $u(k) \in \mathbb{R}^m$ the feedforward signal and $x(k) \in \mathbb{R}^n$ the system states. In order to come to a unique solution to (6.1), the boundary conditions to the difference equation, *i.e.*, the upper part of (6.1), have to be defined as well. These are typically defined at the start of the interval under consideration and are therefore called initial conditions, *i.e.*:

$$x(k_0) = x_0 \tag{6.2}$$

Often, it is tacitly assumed that $x_0 = 0$. Using initial conditions seems a sensible choice, since it enables the possibility to implement (6.1) as a causal filter.

Now, $C_{\text{ff}}(z)$ is stable if and only if it has all its poles in the open unit circle, *i.e.*, $|\lambda_i(A)| < 1$, $\forall i$. Roughly speaking, stability of $C_{\text{ff}}(z)$ implies that injection of a bounded signal $r(k)$ on the interval $k \in [k_0, \ldots, k_f]$ with initial conditions $x(k_0) = x_0$ results in a bounded $u(k)$ on the same interval. Conversely, an unstable $C_{\text{ff}}(z)$ results in an unbounded $u(k)$, which is

undesirable (see Section 3.1). However, instead of specifying initial conditions, one could also define boundary conditions at the end of the interval (final conditions):

$$x(k_f) = x_f \tag{6.3}$$

Remark 6.1 *Similar results are obtained in case of continuous time systems. In that case, $C_{\mathrm{ff}}(s)$ is stable iff it has all its poles in the open left half plane,* i.e., $\mathrm{Re}\,(\lambda_i(A)) < 0,\ \forall i$.

Example 6.1 *As an example, we want to compute the solution to:*

$$\dot{x} = ax, \quad a > 0 \tag{6.4}$$

on the interval $t \in [t_0, t_f]$. If the boundary condition $x(t_0) = x_0$ is specified, the solution is given by:

$$x(t) = x_0 e^{a(t-t_0)} \tag{6.5}$$

which increases exponentially since $a\,(t - t_0) > 0$ on $t \in [t_0, t_f]$. The solution $x(t)$ can therefore become very large. If, however, boundary conditions are specified on the end of the interval, e.g., *$x(t_f) = x_f$, the solution is:*

$$x(t) = x_f e^{a(t-t_f)} \tag{6.6}$$

which is bounded, since $a\,(t - t_f) < 0$ on $t \in [t_0, t_f]$.

Summarizing, if $C_{\mathrm{ff}}(z)$ has all its poles inside the open unit circle, a bounded $u(k)$ results if boundary conditions are defined on the start of the interval under consideration. Conversely, if $C_{\mathrm{ff}}(z)$ has its poles outside the unit circle, the resulting $u(k)$ is bounded if boundary conditions are defined on the end of the interval.

This is the main idea behind stable inversion feedforward (Devasia et al., 1996; Peeters et al., 2000; Sogo, 2010; Zou, 2009; Zou and Devasia, 1999):

Proposition 6.1 *In stable inversion feedforward, $C_{\mathrm{ff}}(z)$ is decomposed into a stable and an unstable part. The boundary conditions of the stable part $C_{\mathrm{ff,s}}(z)$ are subsequently defined on the start of the interval and the boundary conditions of the unstable part $C_{\mathrm{ff,u}}(z)$ are defined on the end of the interval (the subscripts $\cdot_s$ and $\cdot_u$ denote the stable and unstable part).*

From this proposition, it appears that choosing suitable boundary conditions is essential using stable inversion, which is therefore discussed next. Thereafter, a number of issues regarding implementation are discussed.

6.1.2 Boundary Conditions

As mentioned before, boundary conditions on the start of an interval are often assumed $x(k_0) = 0$. This also seems a suitable choice for $C_{\mathrm{ff,s}}(z)$. Choosing x_f for $C_{\mathrm{ff,u}}(z)$, on the other hand, requires more attention.

To enable a suitable choice, the reference trajectories considered are in rest at both the start and the end of the interval, *i.e.*, $r(k < k_0) = r(k_0)$ and $r(k > k_f) = r(k_f)$ (see Figure 6.1). Given this assumption, it makes sense to choose the boundary conditions such that the feedforward signal converges to a constant value for $k < k_0$ and $k > k_f$. Hereto, $x_{\mathrm{u},f}$

has to be chosen such that $C_{\mathrm{ff,u}}(z)$ is in equilibrium at $k = k_f$. A discrete time system is in equilibrium if $x(k+1) = x(k)$, hence:

$$x_u(k+1) = A_u x_u(k) + B_u r(k) = x_u(k) \tag{6.7}$$

which is solved by:

$$x_{\mathrm{u},f} = (I - A_u)^{-1} B_u r(k_f) \tag{6.8}$$

Since $r(k > k_f) = r(k_f)$, it holds that $x_u(k > k_f) = x_u(k_f)$ and therefore $u_u(k > k_f) = u_u(k_f)$. Furthermore, $C_{\mathrm{ff,s}}(z)$ is stable and therefore converges to steady state, so that the total feedforward signal $u_{\mathrm{ff}}(k) = u_s(k) + u_u(k)$ also converges to steady state after the interval under consideration $k \in [k_0, \ldots, k_f]$.

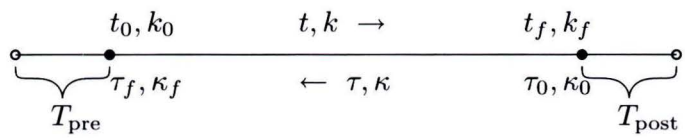


Figure 6.1: A timeline to illustrate time-reversal and pre- and post-actuation.

A similar reasoning can be made regarding the start of the interval: here, $u_s(k < 0) = 0$ and $u_u(k)$ will converge to a constant value for $k < k_0$. Note that this has implications for the servo behavior: if one wants to achieve exact tracking, actuation of the system should already start *before* the system is supposed to move. This pre-actuation is required to prevent transient errors at the start of the move, as discussed in the lifted domain in Section 5.2.

6.1.3 Implementation Aspects

In order to obtain a bounded solution to (6.1) if $C_{\mathrm{ff}}(z)$ contains both stable and unstable poles, $C_{\mathrm{ff}}(z)$ should be decomposed into a stable $C_{\mathrm{ff,s}}(z)$ and unstable part $C_{\mathrm{ff,u}}(z)$. As argued in Wortelboer (1994), a robust matrix form from which the eigenvalues can be read out is the complex Schur form. This is used in Peeters et al. (2000) to decompose the system into a stable and an unstable part.

First, matrix A is Schur transformed into an upper triangular form, where the diagonal contains the eigenvalues of A. In MATLAB , the `schur` is used to compute the Schur decomposition. Furthermore, the eigenvalues are ordered using `ordschur`, such that $\tilde{A}_{11}$ contains the eigenvalues inside the unit disc and $\tilde{A}_{22}$ the eigenvalues outside the unit disc. As a result:

$$\tilde{A} = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ 0 & \tilde{A}_{22} \end{bmatrix} \tag{6.9}$$

with $\tilde{A}_{11}$ and $\tilde{A}_{22}$ both upper triangular. Next, a state transformation has to be applied such that:

$$T^{-1}\tilde{A}T = \begin{bmatrix} A_s & 0 \\ 0 & A_u \end{bmatrix} \tag{6.10}$$

with:

$$T = \begin{bmatrix} I & X \\ 0 & I \end{bmatrix} \tag{6.11}$$

where X can be computed by solving the Sylvester equation (using `lyap` in MATLAB):

$$\tilde{A}_{11}X - X\tilde{A}_{22} + \tilde{A}_{12} = 0 \tag{6.12}$$

Using T and the Schur transformation matrix, $C_{\text{ff}}(z)$ can be brought into the form:

$$C_{\text{ff}} = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \left[\begin{array}{cc|c} A_s & 0 & B_s \\ 0 & A_u & B_u \\ \hline C_s & C_u & D \end{array}\right] \tag{6.13}$$

Hence,

$$C_{\text{ff}} = C_{\text{ff,s}} + C_{\text{ff,u}} = \left[\begin{array}{c|c} A_s & B_s \\ \hline C_s & D \end{array}\right] + \left[\begin{array}{c|c} A_u & B_u \\ \hline C_u & 0 \end{array}\right] \tag{6.14}$$

where D can be included in either $C_{\text{ff,s}}$ or $C_{\text{ff,u}}$.

As mentioned before, $u_s(k)$ can be computed straightforwardly since boundary conditions are defined on the start of the interval and $C_{\text{ff,s}}(z)$ can therefore be implemented as a causal filter. This is, however, not the case for $u_u(k)$. In order to calculate this component $r(k)$ is fed into $C_{\text{ff,u}}$ in reverse time. This way, $u_u(k)$ can be computed offline.

In continuous time, the time-reversed system can be found by using $\dot{x}(-t) = -\dot{x}(t)$. Therefore:

$$\begin{aligned} \dot{x}(\tau) &= -A_u x(\tau) - B_u r(\tau) \\ u(\tau) &= C_u x(\tau) + D_u r(\tau) \end{aligned} \tag{6.15}$$

In discrete time, time-reversal is somewhat more complicated. First, z is replaced by z^{-1}, which essentially means the time reversal is performed: $k+1$ is replaced by $k-1$. To emphasize the backward direction, the symbol κ is used instead of k.

$$\begin{aligned} x(\kappa - 1) &= A_u x(\kappa) + B_u r(\kappa) \\ u(\kappa) &= C_u x(\kappa) + D_u r(\kappa) \end{aligned} \tag{6.16}$$

This is not a standard discrete state-space description. The first step to come to a standard description is rewriting the state equation in (6.16) such that:

$$\begin{aligned} x(\kappa) &= A_u^{-1} x(\kappa - 1) - A_u^{-1} B_u r(\kappa) \\ u(\kappa) &= C_u x(\kappa) + D_u r(\kappa) \end{aligned} \tag{6.17}$$

Next, $x(\kappa)$ is shifted one sample such that $\tilde{x}(\kappa + 1) = x(\kappa)$, hence:

$$\begin{aligned} \tilde{x}(\kappa + 1) &= A_u^{-1} \tilde{x}(\kappa) - A_u^{-1} B_u r(\kappa) \\ u(\kappa) &= C_u \tilde{x}(\kappa + 1) + D_u r(\kappa) \end{aligned} \tag{6.18}$$

Now the state equation is in the standard form, but the output equation contains $\tilde{x}(\kappa + 1)$ instead of $\tilde{x}(\kappa)$. This is solved by substituting the state equation into the output equation.

$$\begin{aligned} \tilde{x}(\kappa + 1) &= A_u^{-1} \tilde{x}(\kappa) - A_u^{-1} B_u r(\kappa) \\ u(\kappa) &= C_u A_u^{-1} \tilde{x}(\kappa) + \left(D_u - C_u A_u^{-1} B_u\right) r(\kappa) \end{aligned} \tag{6.19}$$

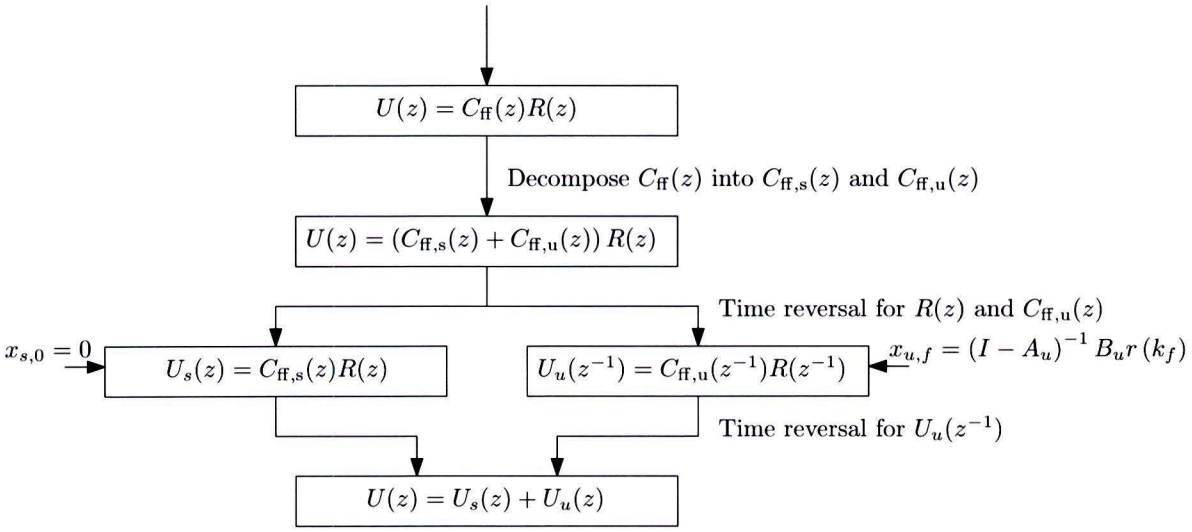


Figure 6.2: Flow-diagram of stable inversion.

From this expression, it immediately becomes clear why this leads to a bounded solution: since A_u has all its eigenvalues outside the unit circle, A_u^{-1} will have all its eigenvalues inside the unit circle and the system in (6.19) is therefore stable. The concept of stable inversion feedforward is graphically depicted in Figure 6.2.

Although this concept seems very different from the lifted approach of Chapter 5, the results show strong resemblance. This is shown in the next section.

6.2 Stable Inversion in the Lifted Domain

In the concept introduced in the previous section, the solution to the unstable part of the inverse system is computed in reverse time, hence it can be seen as a-causal. The feedforward signal that is obtained by using the Moore-Penrose pseudoinverse is also a-causal. Therefore, it is interesting to see if there is a connection between these two results. This is investigated in this section.

Recall the lifted system description including initial conditions of (5.1). Now, consider the lifted system description of an *inverse* system which is decomposed into a stable and an unstable part, see (6.14). Taking initial conditions into account, the lifted system description is given by:

$$\boldsymbol{u} = (J_s + J_u)\,\boldsymbol{r} + J_{s,0}x_{s,0} + J_{u,0}x_{u,0} \tag{6.20}$$

with:

$$
\begin{aligned}
\boldsymbol{u} &= \begin{bmatrix} u^T(0) & u^T(1) & \cdots & u^T(N-1) \end{bmatrix}^T \\
\boldsymbol{r} &= \begin{bmatrix} r^T(0) & r^T(1) & \cdots & r^T(N-1) \end{bmatrix}^T \\
J_s &= \begin{bmatrix} D & 0 & \cdots & 0 \\ C_s B_s & D & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C_s A_s^{N-2} B_s & C_s A_s^{N-3} B_s & \cdots & D \end{bmatrix} \\
J_u &= \begin{bmatrix} 0 & 0 & \cdots & 0 \\ C_u B_u & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C_u A_u^{N-2} B_u & C_u A_u^{N-3} B_u & \cdots & 0 \end{bmatrix} \\
J_{s,0} &= \begin{bmatrix} C_s^T & (C_s A_s)^T & \cdots & (C_s A_s^{N-1})^T \end{bmatrix}^T \\
J_{u,0} &= \begin{bmatrix} C_u^T & (C_u A_u)^T & \cdots & (C_u A_u^{N-1})^T \end{bmatrix}^T
\end{aligned}
$$

Since A_u has its eigenvalues outside the unit circle, the entries of the first column of J_u increase exponentially and may grow unbounded. Similarly to stable inversion in Section 6.1.1, a boundary condition $x_{u,f}$ is defined on the end of the trajectory:

$$
x_u(N) = A_u^N x_{u,0} + \sum_{i=0}^{N-1} A_u^i B_u r(N-1-i) = x_{u,f} \tag{6.21}
$$

Solving this equation for $x_{u,0}$ leads to:

$$
\begin{aligned}
x_{u,0} &= -A_u^{-N} \sum_{i=0}^{N-1} A_u^i B_u r(N-1-i) + A_u^{-N} x_{u,f} \\
&= -A_u^{-N} \begin{bmatrix} A_u^{N-1} B_u & A_u^{N-2} B_u & A_u^{N-3} B_u & \cdots & B_u \end{bmatrix} \begin{bmatrix} r(0) \\ r(1) \\ r(2) \\ \vdots \\ r(N-1) \end{bmatrix} + A_u^{-N} x_{u,f}
\end{aligned} \tag{6.22}
$$

Substituting this result into Equation 6.20 results in:

$$
\boldsymbol{u} = (J_s + \tilde{J}_u)\boldsymbol{r} + J_{s,0} x_{s,0} + \tilde{J}_{u,0} x_{u,f} \tag{6.23}
$$

with:

$$
\begin{aligned}
\tilde{J}_u &= J_u - J_{u,0}A_u^{-N}\begin{bmatrix}A_u^{N-1}B_u & A_u^{N-2}B_u & A_u^{N-3}B_u & \cdots & B_u\end{bmatrix} \\
&= \begin{bmatrix}
-C_uA_u^{-1}B_u & -C_uA_u^{-2}B_u & -C_uA_u^{-3}B_u & \cdots & -C_uA_u^{-N}B_u \\
0 & -C_uA_u^{-1}B_u & -C_uA_u^{-2}B_u & \cdots & -C_uA_u^{-(N-1)}B_u \\
0 & 0 & -C_uA_u^{-1}B_u & \cdots & -C_uA_u^{-(N-2)}B_u \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & -C_uA_u^{-1}B_u
\end{bmatrix} \\
\tilde{J}_{u,0} &= J_{u,0}A_u^{-N} \\
&= \begin{bmatrix}\left(C_uA_u^{-N}\right)^T & \left(C_uA_u^{-N+1}\right)^T & \left(C_uA_u^{-N+2}\right)^T & \cdots & \left(C_uA_u^{-1}\right)^T\end{bmatrix}^T
\end{aligned}
$$

This illustrates why the feedforward signals resulting from the lifted approach and stable inversion show a strong resemblance, since the first row of $\tilde{J}_u$ is similar to the first row of $J^\dagger$ (see Figure 5.1). Although not shown here, $\tilde{J}_u$ corresponds with the upper-diagonal part of a truncated pseudo-inverse (Section 5.3.1). The observation that the lifted solution is time-varying cannot be explained by the LTI concepts of this chapter. This can be attributed to the fact that the lifted system description, contrary to state-space systems or transfer functions, is a *finite* time system description. This can be seen if the size of the impulse response matrix J of a system with one unstable zero is increased: it turns out that the time-varying behavior of $J^\dagger$ manifests itself more at the edges, as is already discussed in Section 5.3.1.

In the next section, it is discussed how uncertainty can be addressed if stable inversion is used for feedforward.

6.3 Uncertainty

In the discussion above about stable inversion feedforward it is assumed that $C_{\text{ff}}(z)$ exactly resembles the inverse plant. However, as argued in Chapter 4, this is never the case in practice and there will always be a certain amount of model uncertainty. Handling model uncertainty in case of stable inversion feedforward is addressed in this section.

In Section 4.4, it is shown why methods like $\mathcal{H}_\infty$ and $\mathcal{H}_2$ feedforward, which optimize the performance of the worst case plant P present in $\mathcal{P}$, may not be suited for feedforward control. In this section, a method is developed which calculates the optimal feedforward filter for the entire model set $\mathcal{P}$, which is directly applicable using stable inversion feedforward. The model uncertainty is quantified by a multiplicative uncertainty model, *i.e.*:

$$P(z) = P_o(z)\left(I + w_m(z)\Delta(z)\right) \tag{6.24}$$

where $P_o(z)$ denotes the nominal model, $w_m(z)$ a multiplicative uncertainty bound, $C_{\text{ff}}(z)$ the feedforward filter and $\Delta(z)$ a complex uncertainty such that $\|\Delta(z)\|_\infty \leq 1$. A multiplicative uncertainty description is chosen, since if $|w_m| > 1$, this straightforwardly shows that the magnitude of the uncertainty exceeds the magnitude of the plant (see Chapter 4). The

feedforward sensitivity is given by:

$$\begin{aligned} S_{\text{ff}}(z) &= I - P(z)C_{\text{ff}}(z) \\ &= I - P_o(z)\left(I + w_m(z)\Delta(z)\right)C_{\text{ff}}(z) \\ &= I - P_o(z)C_{\text{ff}}(z) - P_o(z)w_m(z)\Delta(z)C_{\text{ff}}(z) \end{aligned} \tag{6.25}$$

The complex uncertainty $\Delta(z)$ essentially forms a circle with radius $r \in [0,1]$ and angle $\theta \in [-\pi, \pi]$ at each frequency f. With $\cdot^*$ defining the complex conjugate, $\Delta(z)$ and $\Delta^*(z)$ can be written as:

$$\Delta(z) = r(z)\left(\cos\theta(z) + j\sin\theta(z)\right) \tag{6.26a}$$

$$\Delta^*(z) = r(z)\left(\cos\theta(z) - j\sin\theta(z)\right) \tag{6.26b}$$

$$\Delta(z)\Delta^*(z) = r^2(z) \tag{6.26c}$$

In Section 4.4, the optimal feedforward is defined as the value for C_{ff} which minimizes the integral of $S_{\text{ff}}^2(\Delta, C_{\text{ff}})$ on the interval $\Delta \in [-1, 1]$. A similar reasoning can be used to compute the optimal $C_{\text{ff}}(z)$ in case $P(z)$, $w_m(z)$ and $\Delta(z)$ are dynamic systems instead of real constants. Since $S_{\text{ff}}(z)$ is complex, the square of the magnitude is computed by $|S_{\text{ff}}(z)|^2 = S_{\text{ff}}(z)S_{\text{ff}}^*(z)$ and the cost function $H(z)$ is defined as:

$$\begin{aligned} H(z) &= \iint_{\Delta} S_{\text{ff}}(z)S_{\text{ff}}^*(z)d\Delta \\ &= \int_0^{2\pi}\int_0^1 S_{\text{ff}}(z)S_{\text{ff}}^*(z)r\,dr\,d\theta \end{aligned} \tag{6.27}$$

Now, using:

$$\int_0^{2\pi} \Delta(z)d\theta = \int_0^{2\pi} \Delta^*(z)d\theta = 0 \tag{6.28}$$

and

$$\int_0^{2\pi} = \Delta(z)\Delta^*(z)d\theta = 2\pi r^2 \tag{6.29}$$

it can be shown that:

$$H(z) = \frac{\pi}{2}P_o(z)P_o^*(z)C_{\text{ff}}(z)C_{\text{ff}}^*(z)w_m(z)w_m^*(z) + \pi\left(P_o(z)C_{\text{ff}}(z) - I\right)\left(P_o^*(z)C_{\text{ff}}^*(z) - I\right) \tag{6.30}$$

Finally, the optimal $C_{\text{ff}}(z)$ can be calculated by setting the derivative of $H(z)$ with respect to $C_{\text{ff}}^*(z)$ to zero:

$$\frac{\partial H(z)}{\partial C_{\text{ff}}^*(z)} = \frac{\pi}{2}P_o(z)P_o^*(z)C_{\text{ff}}(z)C_{\text{ff}}^*(z)w_m(z)w_m^*(z) + \pi\left(P_o(z)P_o^*(z)C_{\text{ff}}(z) - P_o^*(z)\right) = 0 \tag{6.31}$$

which results in:

$$C_{\text{ff}}(z) = \left(P_o(z)\left(I + \frac{1}{2}w_m(z)w_m^*(z)\right)\right)^{-1} \tag{6.32}$$

From this expression, it is clear that in case no uncertainty is present, *i.e.*, $w_m(z) = 0$, the feedforward filter equals the inverse of the nominal plant. Furthermore, it can be seen that model uncertainty can only decrease the gain at certain frequencies, which is consistent with other robust feedforward methods. A final remark is that $\angle(\frac{1}{2}w_m(z)w_m^*(z)) = 0$, which implies

that robustness does not introduce any phase angle. As discussed in Subsection 4.2.3, this is essential to achieve good servo performance. Since $w_m(z)$ is chosen as a stable weighting filter, $w_m^*(z)$ is typically unstable. Therefore, pre-actuation is not only applied in case of NMP systems but, is also present in case of an uncertain MP system. This also appeared in the example using robust feedforward in the lifted domain (Section 5.6.4).

In the next section, the use of nominal and robust stable inversion feedforward is illustrated by means of an example using the flexible cart introduce in Section 3.4.1.

6.4 Example

The stable inversion concepts discussed in this chapter are illustrated by repeating the example of Chapter 5, hence the flexible cart is supposed to track the reference trajectory in Figure 5.9. Again, nominal feedforward is discussed, as well as the effect of pre- and post-actuation and robust feedforward.

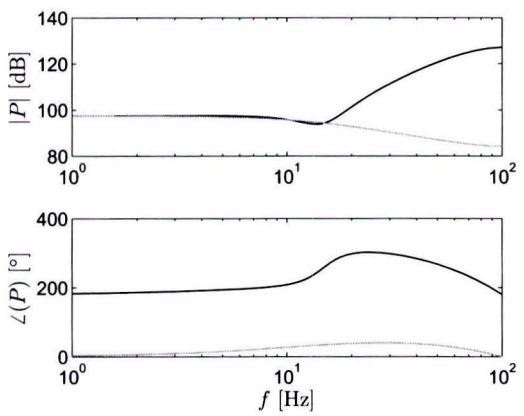


Figure 6.3: Bode-diagrams of the stable (black) and unstable part (gray) of the inverted NMP flexible cart model.

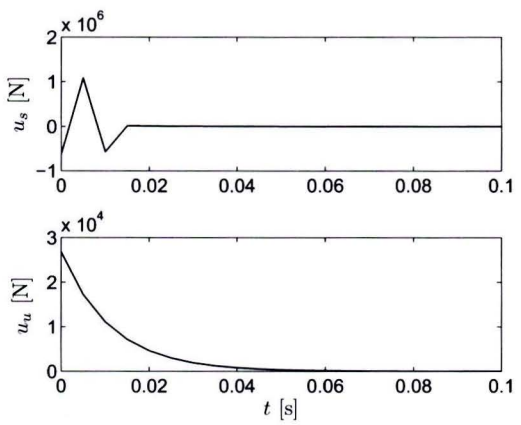


Figure 6.4: Impulse response of the stable part (upper plot) and time-reversed unstable part (lower plot) of the inverted NMP flexible cart model.

6.4.1 Nominal Feedforward

To compute the feedforward signal, the inverse system $P^{-1}(z)$ is first decomposed into a stable part $P_s^{-1}(z)$ and an unstable part $P_u^{-1}(z)$. The Bode-diagrams of both systems are shown in Figure 6.3. At low frequencies, both $P_s^{-1}(z)$ and $P_u^{-1}(z)$ have the same magnitude and a zero slope. The phase difference is exactly 180°. Note that the time-reversed $P_u^{-1}(z^{-1})$ (not shown) has the same magnitude as $P_u^{-1}(z)$ but a negative phase.

To get an impression about the required pre- and post-actuation time, the impulse responses of $P_s^{-1}(z)$ and $P_u^{-1}(z^{-1})$ are plotted in Figure 6.4. In this figure, both impulse responses seem to have converged to zero after 0.05 s. However, as will be shown in Section 6.4.2, using more pre- and post-actuation results in smaller servo error. Using the entire trajectory in

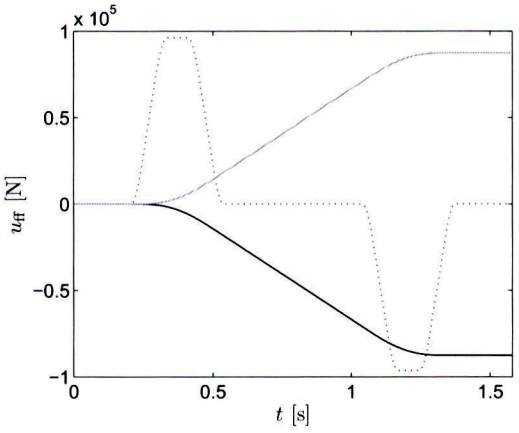


Figure 6.5: Scaled acceleration (dotted) and stable (black) and unstable (gray) part of the feedforward signal.

Figure 5.9, *i.e.*, with $T_{\text{pre}} = T_{\text{post}} = 0.2$ s, the stable ($u_{\text{ff,s}}$) and unstable ($u_{\text{ff,u}}$) component of the feedforward signal are computed. The results are plotted in Figure 6.5. Both components strongly resemble the reference trajectory, with the stable component being negative. This can be explained by the fact that the Bode-diagrams both have a zero slope, the same magnitude but a 180° phase difference. Adding $u_{\text{ff,s}}$ and $u_{\text{ff,u}}$, it appears that both components nearly cancel each other, resulting in the total feedforward signal shown in Figure 6.6.

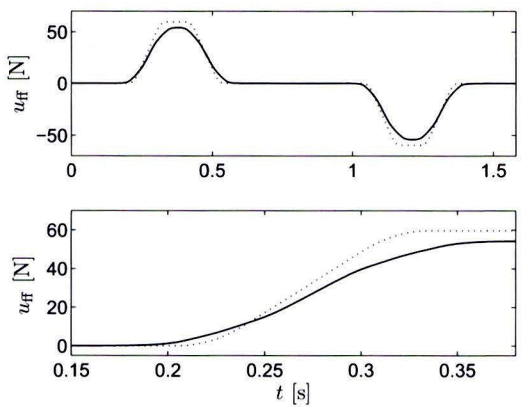


Figure 6.6: Scaled acceleration (dotted) and feedforward signal (solid) resulting from stable inversion feedforward. The bottom plot is a zoom of the upper plot, showing that u_{ff} starts increasing before the reference acceleration.

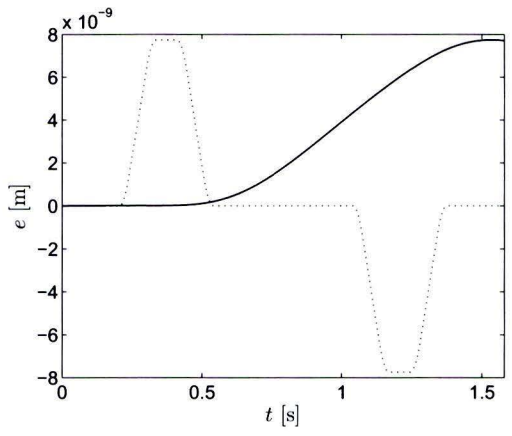


Figure 6.7: Scaled acceleration (dotted) and servo error (solid) using the feedforward signal of Figure 6.6.

The feedforward signal is very similar to that of Chapter 5, so instead of being a scaled version of the acceleration, it is much smoother. It can be seen that the pre-actuation starts exciting the system while r_a is still zero and hence before the system is supposed to move. Since there is one real NMP zero rather than a complex pair, the pre-actuation shows no oscillations. Furthermore, the transitions from constant jerk to constant acceleration phase

and vice-versa are smooth in the feedforward signal. This u_{ff} results in the servo error shown in Figure 6.7. It appears that the error looks like a scaled version of $r(k)$, but there are neither significant transient errors nor oscillations at the resonance frequency. The magnitude of the error, however, is significantly larger than in Figure 5.12, where lifted feedforward is used. The sharp peak in Figure 5.12 is completely absent in this figure.

In these figures, the entire time before and after the start of the move is used for pre- and post-actuation. In the next section, the effect of the amount of pre- and post-actuation is investigated, followed by simulations using an uncertain system.

6.4.2 Pre- and Post-Actuation

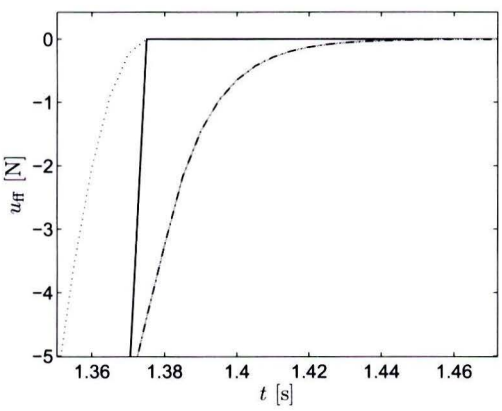


Figure 6.8: Scaled acceleration (black, dotted) and u_{ff} using $T_{\text{post}} = 0.0$ s (black, solid), $T_{\text{post}} = 0.1$ s (gray) and $T_{\text{post}} = 0.2$ s (black, dash-dotted). $T_{\text{pre}} = 0.2$ s in all cases. The gray and the dash-dotted line can hardly be distinguished by visual inspection.

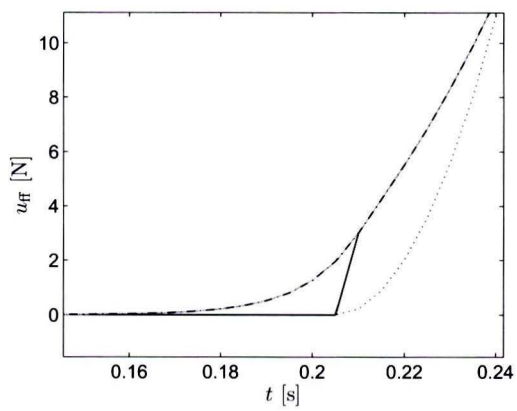


Figure 6.9: Scaled acceleration (black, dotted) and u_{ff} using $T_{\text{pre}} = 0.0$ s (black, solid), $T_{\text{pre}} = 0.1$ s (gray) and $T_{\text{pre}} = 0.2$ s (black, dash-dotted). $T_{\text{post}} = 0.2$ s in all cases. The gray and the dash-dotted line can hardly be distinguished by visual inspection.

When implementing stable inversion feedforward, a suitable interval has to be selected on which the feedforward signal is defined, and boundary conditions are defined at the start and end of this interval. Similar to Section 5.6.3, T_{post} and T_{pre} have been varied to show the effect of pre- and post-actuation. The difference between the various feedforward signals is plotted in Figures 6.8 and 6.9. Contrary to Figures 5.13 and 5.14, there is no compensation for the fact that only a limited amount of T_{pre} or T_{post} is used. This means that the feedforward signals are essentially cut off if insufficient pre- or post-actuation time is available. Similar to the lifted approach, the difference between the feedforward signals with $T_{\text{pre}} = 0.1$ s and $T_{\text{pre}} = 0.2$ s and between $T_{\text{post}} = 0.1$ s and $T_{\text{post}} = 0.2$ s seems negligible by visual inspection, but does have a significant effect on the servo error.

While in Figure 6.7 pre- and post-actuation times of $T_{\text{pre}} = T_{\text{post}} = 0.2$ s are used, less post-actuation leads to servo errors after the move (Figure 6.10) and less pre-actuation leads to transient errors at the start of the move (Figure 6.11). This is similar to the observation in Section 5.6.3.

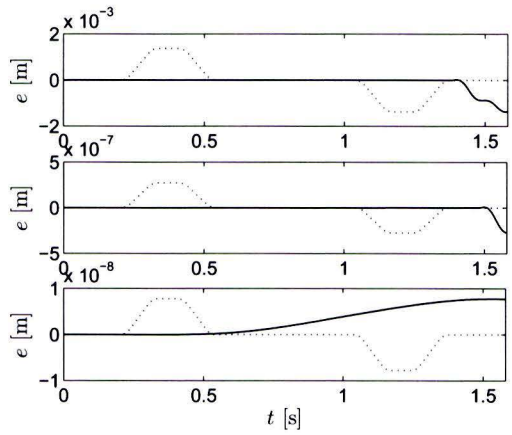


Figure 6.10: Scaled acceleration (dotted), and e using $T_{\text{pre}} = 0.2$ s and $T_{\text{post}} = 0.0$ s (upper plot), $T_{\text{post}} = 0.1$ s (middle plot) and $T_{\text{post}} = 0.2$ s (lower plot).

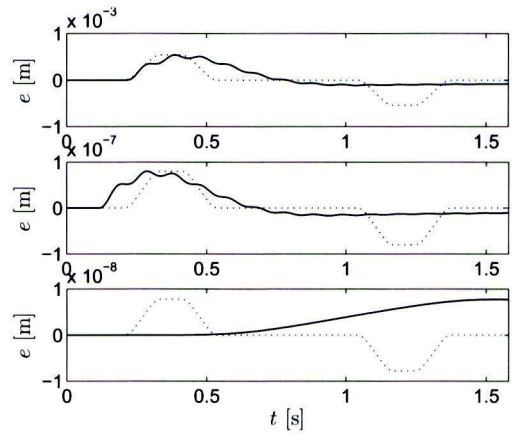


Figure 6.11: Scaled acceleration (dotted), and e using $T_{\text{post}} = 0.2$ s and $T_{\text{pre}} = 0.0$ s (upper plot), $T_{\text{pre}} = 0.1$ s (middle plot) and $T_{\text{pre}} = 0.2$ s (lower plot).

Again, these results can be explained by looking at the states of the system, in particular to the angle ϕ and angular velocity $\dot{\phi}$ (Figure 6.12). If no post-actuation is used, the feedforward input u_{ff} turns zero as the output y reaches its final value. The states ϕ and $\dot{\phi}$ do not equal zero at that point, hence setting $u_{\text{ff}} = 0$ causes ϕ and $\dot{\phi}$ to show an oscillation with decreasing amplitude.

Similarly, using sufficient pre-actuation ensures that the cart is gradually pushed into the angle that will be reached during the constant acceleration phase. Not using sufficient pre-actuation causes a sudden push, hence introducing a transient error. Comparing Figures 5.17 and 6.12, it appears that $\dot{\phi}$ shows a larger deviation from the 'ideal' trajectory if LTV lifted feedforward is used.

Although the differences between the signals in Figures 6.8 and 6.9 seem small, they do have a major impact on the servo error as can be seen in Figures 6.10 and 6.11. Comparing Figures 5.16 and 6.11, the difference between LTV and LTI feedforward can be seen. In Figure 5.16, insufficient pre-actuation introduces a sharp peak in the error signal, so the error is concentrated in a small time-interval. In Figure 6.11, however, the error starts increasing at the same moment, but only slowly returns towards zero.

In Figure 6.13 the 2-norm of the servo error on the entire interval is plotted for various values of T_{pre} and T_{post}. Note that $\|e\|_2$ is plotted on a logarithmic scale. The straight red line in Figure 6.13 thus indicates that $\|e\|_2$ depends exponentially on T_{pre}. The dependency of $\|e\|_2$ on T_{pre} depends on the location of the unstable zeros, as discussed in Section 5.2. A similar observation is made if the maximum servo error is plotted for various values of T_{pre} and T_{post}. The bound on the error due to insufficient post-actuation has not been proven yet, but it seems likely that this is a bound similar to bound related to pre-actuation.

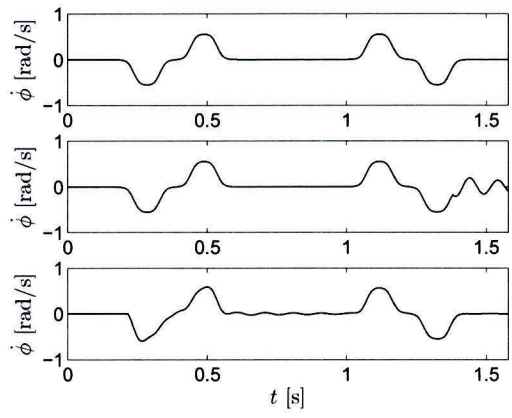


Figure 6.12: Angular velocity using $T_{\text{pre}} = T_{\text{post}} = 0.2$ s (upper plot), $T_{\text{post}} = 0$ s (middle plot) , introducing oscillations after the movement and $T_{\text{pre}} = 0$ s (lower plot), introducing an error before the movement.

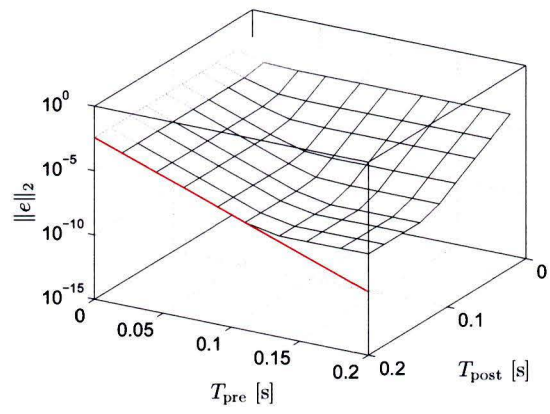


Figure 6.13: 2-norm of the tracking error versus T_{pre} and T_{post}. The red line shows the underbound for $T_{\text{post}} = 0.2$ s as discussed in Section 5.2.4.

6.4.3 Robust Feedforward

To assess the performance of robust stable inversion in Section 6.3, the same model set $\mathcal{P}$ is used as in Chapter 5. Using the same overbound $w_m(z)$, a robust feedforward is computed according to Section 6.3. The resulting feedforward signal is also plotted in Figure 6.14. It can be seen that, contrary to nominal feedforward, pre- actuation is used and the resulting feedforward signal is slightly smoother. Furthermore, the oscillations during the constant velocity interval and after the move are much smaller. Both feedforward signals are used to simulate with 10 realizations of the uncertain system. The resulting servo errors are plotted in Figure 6.16.

It appears that the pre-actuation introduces a low-frequent servo error before the start of the move. This is due to the limited pre-actuation time. In Figure 6.14 it can be seen that the robust feedforward signal does not start entirely from rest. It can be expected that an increased preview period leads to a smaller pre-actuation error. During the move, however, the error using robust feedforward is smaller than using nominal feedforward on the same perturbed system. The dominant component of approximately 6.2 Hz is partially removed.

These findings are confirmed by looking at the cumulative power spectral densities of the servo error in Figure 6.17. It seems that robust feedforward generally shows larger values at low frequencies; if a large mismatch between inverse plant and feedforward is present, nominal feedforward shows a large increase at 6.2 Hz (the frequency of the anti-resonance of the nominal system), while this difference is much smaller if robust feedforward is used. It is most clear in the gray case. This shows the advantage of robust feedforward: the low-frequent error can be attenuated by the feedback controller, while this is much more difficult for high-frequent errors.

The state trajectories of the simulations using nominal and robust stable inversion feedforward also show similar behavior as using lifted feedforward. Using robust feedforward, oscillations

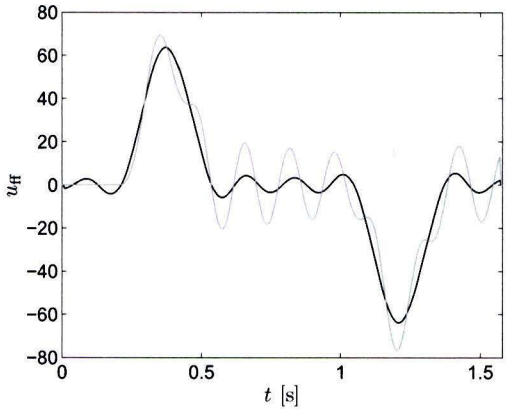


Figure 6.14: Nominal (gray) and robust (black) feedforward signal for the MP flexible cart system. The amplitude of the oscillation of the robust feedforward signal is much smaller than that of the nominal feedforward signal.

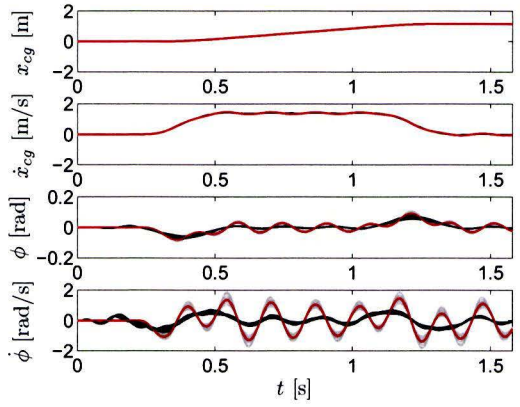


Figure 6.15: System states of 10 realizations of the uncertain system using nominal (gray) and robust (black) feedforward and states of the nominal system using nominal feedforward (red). Using robust feedforward, the states clearly show less oscillatory behavior.

of the flexible modes have a smaller amplitude. This is possible due to the use of pre-actuation.

In short, it can be stated that in case of a highly uncertain system, robust feedforward leads to improved servo performance during the movement, especially during the constant velocity phase (see lower plot of Figure 6.16). This improvement can be attributed to the reduced oscillation of the feedforward signal at the frequency of the anti-resonance of the system. The improvement during the constant velocity phase comes at the expense of an increased servo error during pre-actuation.

6.5 Summary

Regarding stable inversion feedforward, a number of concluding remarks can be made:

- Using stable inversion, a bounded feedforward signal is obtained despite unstable poles of the inverse system.
- The results of stable inversion and lifted feedforward methods show strong resemblance, but since time-varying solutions are possible in the lifted domain, lifted feedforward is able to partly compensate for the limited available pre-actuation time while this is not possible using stable inversion. However, in both cases, sufficient pre- and post-actuation is required to obtain good servo tracking in case of non-minimum phase systems.
- The required pre- and post-actuation time depends on the locations of the corresponding unstable and stable system zeros. Pre-actuation is used to bring the states of the system to values at the start of the interval such that exact tracking is possible, while post-actuation is used to bring the states to rest after the move. The nominal servo error can

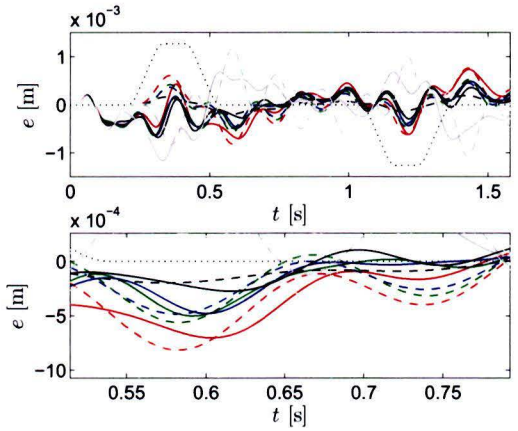


Figure 6.16: Scaled acceleration (dotted) and simulation error for 5 random realizations of the uncertain system using nominal (dashed) and robust (solid) feedforward. The lower plot is a zoom during the constant velocity interval.

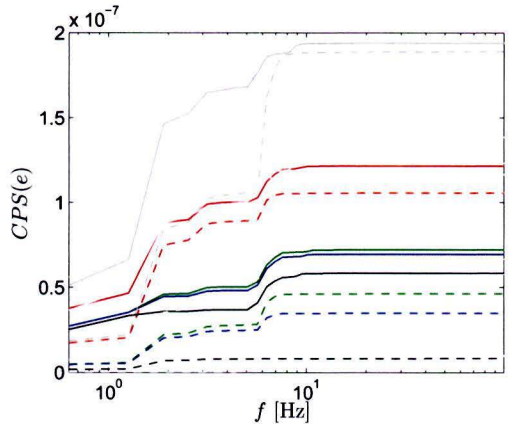


Figure 6.17: Cumulative PSD of the servo errors using nominal (dashed) and robust (solid) feedforward.

be made arbitrarily small by taking T_{pre} and T_{post} sufficiently large. The exact amount of T_{pre} and T_{post} required depends on the demanded servo accuracy.

- Robust stable inversion leads to less oscillatory feedforward signals and state trajectories. Furthermore, it reduces the high-frequent error during the constant velocity interval, which is otherwise present due to a mismatch between the inverse plant and feedforward filter. Nevertheless, this comes at the expense of a low-frequent servo error.

The examples discussed so far concerned a fictitious flexible cart system. In the next chapters, the developed concepts are validated on the NXT-A7 wafer stage.

CHAPTER 7

Experimental Setup

The performance of the feedforward strategy discussed in the previous chapter is validated on the NXT-A7 wafer stage, which is introduced here. First, an overview of this stage is given, followed by a description of the dynamics of the A7 wafer stage. Next, the feedback controllers used to obtain closed loop stability are described in Section 7.3.

7.1 The NXT-A7 Wafer Stage

The NXT-A7 wafer stage was made in the context of ASML's NXT project to learn about planar motor artifacts. Its main components are shown in Figures 7.1 and 7.2. Future wafer stages are required to operate in vacuum to prevent absorption of the laser beam. Therefore, contactless operation is necessary since mechanical and air bearings are not or hardly feasible in vacuum. Furthermore, this removes friction, thus enabling higher positioning accuracy. In order to achieve contactless operation, the NXT-A7 utilizes a planar motor to position the stage (Compter, 2004).

A planar motor has two main parts: a plate with an array of permanent magnets and a set of coils fixed to the wafer stage. By controlling the current through these coils, horizontal and vertical electromechanical forces are generated which enable position control of the stage in six degrees-of-freedom. The NXT-A7 has four forcers, each providing one vertical and one horizontal force. These forcers are mounted to an interface plate. A mirror block is mounted to the interface plate using four leaf springs. The position of the chuck is measured using an interferometry measurement system. This is fixed on a metrology stone, which in turn is mounted on three airmounts to isolate it from floor vibrations.

The main direction of movement of the NXT-A7 wafer stage is the x-direction, having a stroke of approximately 300 mm. End-of-stroke dampers are present to protect the wafer stage and the devices mounted on the metrology stone. In y-direction, the stroke is limited to 5 mm due to the presence of boundary rails (see Figure 7.2). The zero of the coordinate system is defined such that the stage can make symmetric strokes, see Figure 7.3.

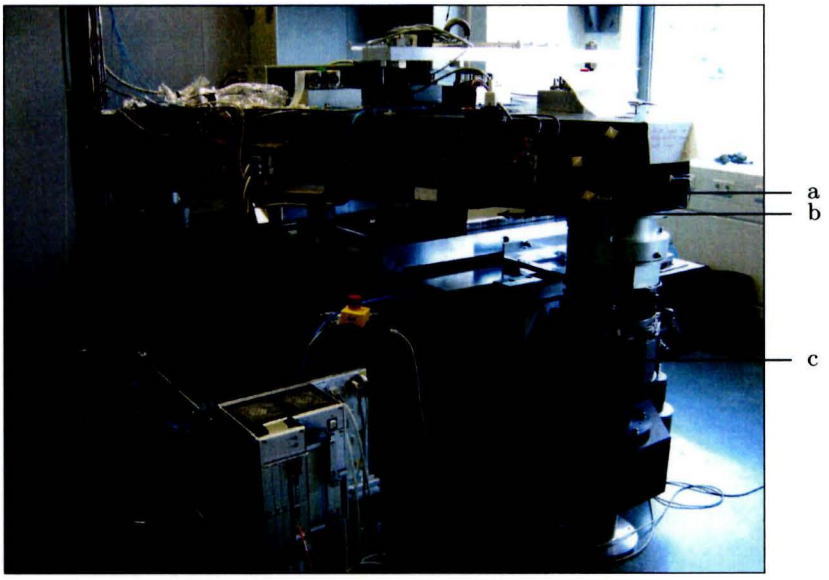


Figure 7.1: Photograph of the NXT-A7 wafer stage, where a: granite block, b: mover, c: airmount

7.2 Dynamics

7.2.1 MIMO Systems and Equivalent Plants

Before continuing to the discussion of the dynamics of the NXT-A7 wafer stage, a note on MIMO systems and equivalent plants is given. Of the six DOFs shown in Figure 7.3, only the x-direction is used for the experiments and models of the x- and y-direction are used for simulations. This implies that the remaining DOFs are not explicitly taken into consideration. This implies that actually the equivalent plants are used. This is illustrated by means of an example using a 2×2 MIMO system.

Example 7.1 *Consider a 2×2 MIMO system P having SISO entries P_{11}, P_{12}, P_{21} and P_{22} (see Figure 7.4). If one wants to implement feedforward in the direction of y_1, not only P_{11} should be taken into account, since there is interaction between the various DOFs involved. Now, the SISO transfer function from u_1 to y_1, while the loop with C_{22} is closed, is also called the equivalent plant $P_{11,\mathrm{eq}}$. This is given by:*

$$P_{11,\mathrm{eq}} = P_{11} - P_{12}\left(I + C_{22}P_{22}\right)^{-1} C_{22}P_{21} \tag{7.1}$$

In case of SISO feedforward of the NXT-A7 wafer stage, the equivalent plant in x-direction is similarly defined by (7.1) with:

- P_{11} the SISO upper left entry of the total 6×6 system, having both its input and output in the x-direction.
- P_{12} the MISO (1×5) upper right part of the total system, having its output in x-direction and inputs in $\mathrm{y}, \mathrm{R_z}, \mathrm{z}, \mathrm{R_x}, \mathrm{R_y}-$directions.
- P_{21} the SIMO (5×1) lower left part of the total system, having its input in the x-direction and outputs in the $\mathrm{y}, \mathrm{R_z}, \mathrm{z}, \mathrm{R_x}, \mathrm{R_y}-$directions.

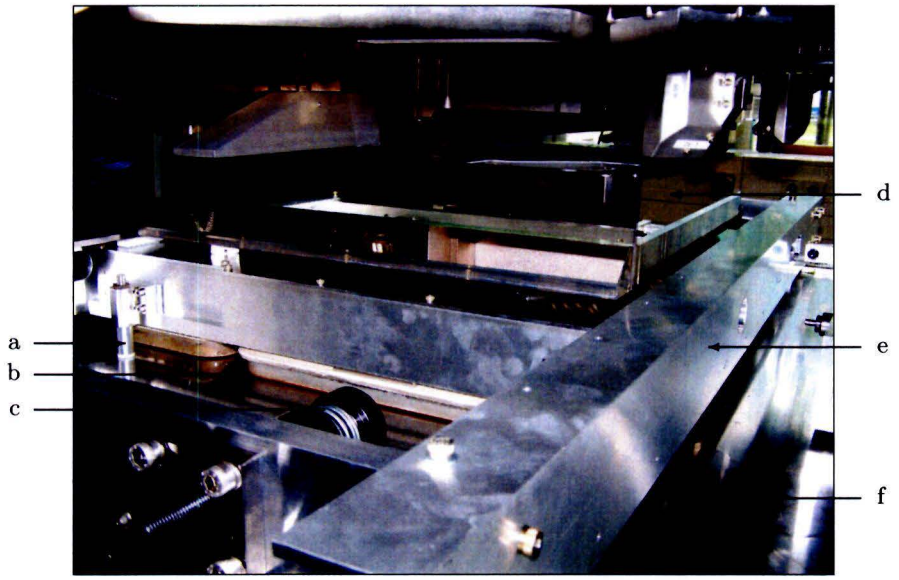


Figure 7.2: Detail photograph of the wafer stage test rig, where a: landing foot, b: forcer coil, c: end-of-stroke damper, d: zerodure mirror block, e: boundary rail and f: permanent magnet plate.

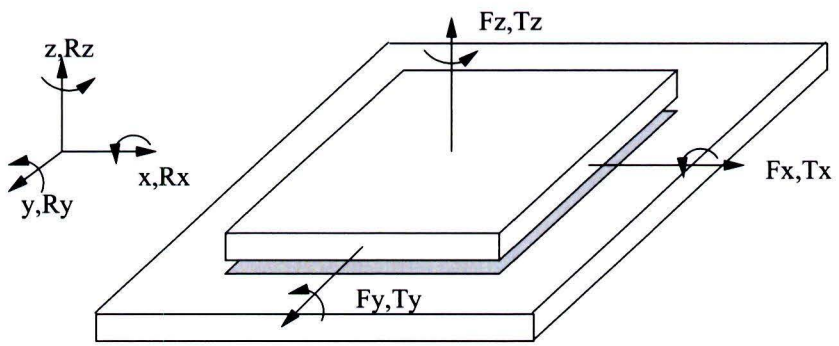


Figure 7.3: Schematic overview of the wafer stage with the coordinate system.

- P_{22} the MIMO (5×5) lower right part of the total system, having both its inputs in $\mathrm{y}, \mathrm{R_z}, \mathrm{z}, \mathrm{R_x}, \mathrm{R_y}-$directions.
- C_{11} the feedback controller in x-direction.
- C_{22} is the diagonal feedback controller in $\mathrm{y}, \mathrm{R_z}, \mathrm{z}, \mathrm{R_x}, \mathrm{R_y}-$directions.

This implies that the equivalent plant $P_{11,\mathrm{eq}}$ is independent of the feedback controller in x-direction C_{11} but *does* depend on the feedback controller used in the other 5 DOFs.

7.2.2 SISO Identification

In Oomen (2010); Quist (2010); Van Herpen (2009a), robust feedback control relevant system identification of the NXT-A7 wafer stage is discussed. The models obtained in these references are not used for the experiments in this thesis because:

- The dynamics have changed over time: in Quist (2010) the dominant resonance was

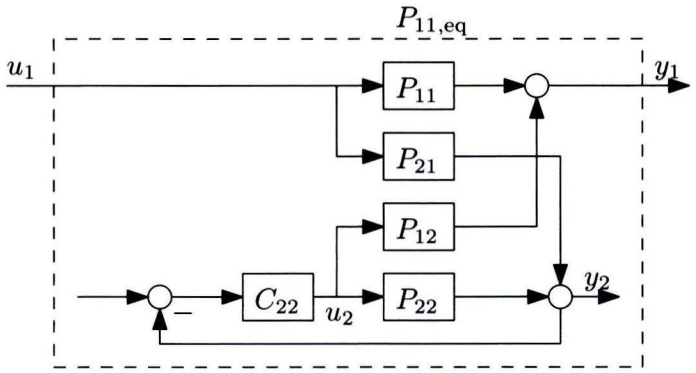


Figure 7.4: The transfer from u_1 to y_1 with the loop containing C_{22} closed is also called the equivalent plant $P_{11,\text{eq}}$.

located around 198 Hz. After renewing the glue connection between the leaf springs and the mirror block, this is now located at a higher frequency, as will be shown.

- In Quist (2010) a 2×2 MIMO model is identified in x- and y-direction. For the feedforward experiments, however, a SISO model of the equivalent x-direction is required. Due to interaction, this is not the same as taking the x-direction of the model by Quist (2010).
- The experiments are aimed at identifying feedback relevant models, whereas here feedforward relevant models are required. As will be explained below, feedback controllers with a lower bandwidth have been used for the identification of feedforward relevant models.

In this section, system identification of the equivalent x-direction is discussed, followed by the MIMO identification in Section 7.2.3.

A common way of measuring frequency response functions (FRFs) is to add noise to the controller output. By subsequently measuring the plant input and the servo error, the input sensitivity $S_i = (I + C_{\text{fb}}P)^{-1}$ and negative process-sensitivity $-PS = -(I + PC_{\text{fb}})^{-1}P$ can be computed. Next, the FRF of the plant is extracted by using:

$$\begin{aligned}(PS)(S_i)^{-1} &= (I + PC_{\text{fb}})^{-1}P(I + C_{\text{fb}}P) \\ &= P(I + C_{\text{fb}}P)^{-1}(I + C_{\text{fb}}P) \\ &= P\end{aligned} \tag{7.2}$$

By injecting and measuring the plant input and servo error only in the x-direction the equivalent plant of this DOF is obtained. In the references mentioned above, it is argued that better results are achieved by using multisine excitation signals rather than noise, since these enable:

- a reduction of the effect of disturbances without introducing systematic errors,
- a straightforward combination of multiple measurements.

Therefore, the multisine signals used in Quist (2010) are also used for the identification experiments in this research.

Besides the excitation signal, the feedback controller used during identification is of importance for the obtained model quality. In Oomen (2010); Van Herpen (2010) it is shown that

for feedback control relevant identification, it is advantageous if the feedback controller used during the identification experiments is as close to the desired situation as possible. This means, *e.g.*, that the bandwidth for the experimental feedback controller should be chosen as high as possible, and in Quist (2010) a controller with 40 Hz bandwidth in x- and y-direction is used. Nevertheless, this is not necessarily true for identification for feedforward control: if a controller has a high bandwidth, the input sensitivity at low frequencies will be very small, subsequently leading to a small signal-to-noise ratio and therefore a significant variance.

This is confirmed by initial experiments with 40 Hz and 25 Hz bandwidth feedback controllers. Using controllers with a lower bandwidth (10 Hz) did not show a significant improvement. Therefore, controllers with a 25 Hz bandwidth in all DOFs are used for the identification experiments. The resulting FRFs and 3σ variances (Van Herpen, 2009b) are displayed in Figures 7.5 and 7.6.

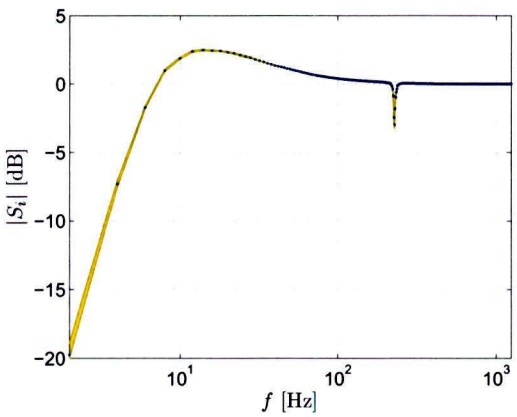


Figure 7.5: Identified FRF of the input sensitivity S_i with 3σ-variance bound.

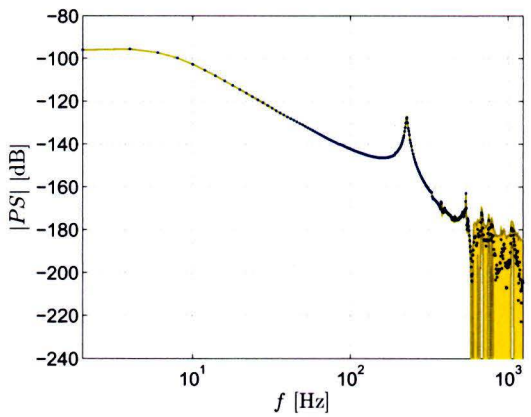


Figure 7.6: Identified FRF of the process sensitivity PS with 3σ-variance bound.

Remark 7.1 *Alternatively, the multisines could have been injected at the controller input rather than at the controller output. By measuring the plant input and the servo error, the negative complementary sensitivity* $-T = -(I + PC_{\text{fb}})^{-1} PC_{\text{fb}}$ *and control sensitivity* $CS = (I + C_{\text{fb}}P)^{-1} C_{\text{fb}}$ *can be determined. Contrary to exciting at the controller output, these transfers do not have small magnitudes and are therefore expected to have a better signal-to-noise ratio and hence a smaller variance. Nevertheless, CS showed a very large variance at low frequencies and it is therefore decided to use excitation at the controller output for system identification.*

The equivalent FRF resulting from the multisine excitation is shown in Figure 7.7. A low-frequent -2 slope is visible corresponding to the rigid body mode of the system. The first resonance appears at 228 Hz, which can be attributed to the limited stiffness of the leaf-springs connecting the mirror block to the interface plate. Furthermore, a resonance and anti-resonance appear at 540 Hz and 588 Hz. At higher frequencies, the variance of PS is very large (see Figure 7.6), hence there is a lot of model uncertainty above approximately 590 Hz.

A tenth order model is fitted on the FRF-data, which is also shown in Figure 7.7. It corresponds very well with the measurement data up to the second resonance, at higher frequencies

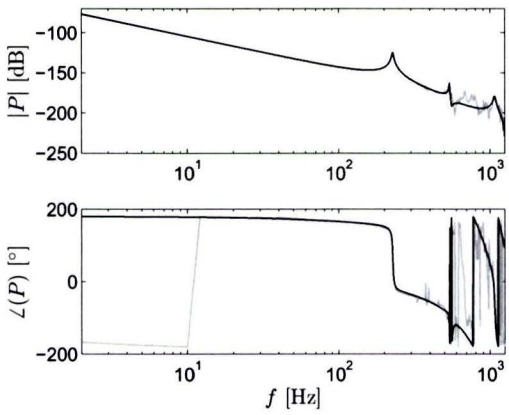


Figure 7.7: Measured FRF (gray) and 10th order model (black) of the plant in x-direction.

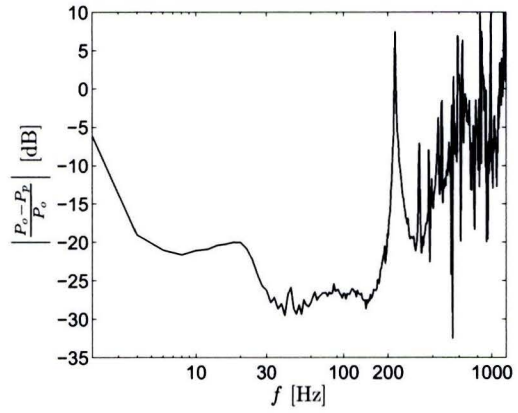


Figure 7.8: Relative difference between two FRFs measured at $x = 0$ cm and $x = 5$ cm.

the difference is much larger.

Position Dependency

In Hamers (2006) actuator-related servo error sources of the NXT-A7 wafer stage are discussed. Due to, *e.g.*, phenomena based on the interaction between the magnetic field and the electric field and the design of the actuator and permanent magnet plate, the wafer stage possesses position dependent dynamics. In Chapter 8 scanning motions are performed from $x = 0$ cm to $x = 5$ cm. In order to get an impression of the position dependency of the NXT-A7, an identification experiment has been performed at $x = 5$ cm. The result is shown in Figure 7.8. Here, the relative difference between this measurement and the original measurement at $x = 0$ cm is plotted.

It appears that between 30 Hz and 200 Hz, -30 dB $< \left|\frac{P_o-P_p}{P_o}\right| < -25$ dB. This means that there is a relative difference of more than 5% is this frequency region, and in the remaining frequencies the uncertainty exceeds 10%. These are significant differences and it is expected that this has a significant effect on the servo performance, as discussed in Section 4.2.

7.2.3 MIMO Identification

In Chapter 9, simulations are performed with a 2×2 MIMO model of the x- and y-direction of the NXT-A7 wafer stage. The used model is obtained in Quist (2010) and the corresponding Bode-diagram is plotted in Figure 7.9.

This model is identified before the glue connections were renewed, which explains why the dominant resonance is located at 198 Hz rather than 228 Hz. Since this model is only of 8th order, only the first resonance is fitted. One of the most striking features of this model are the low-frequent zero slopes. The excitation signal used for identification contains frequencies down to 2 Hz, and it appears that the resulting model has zero slopes below 2 Hz. It can be argued whether this is an accurate representation of the real plant; since the only connection

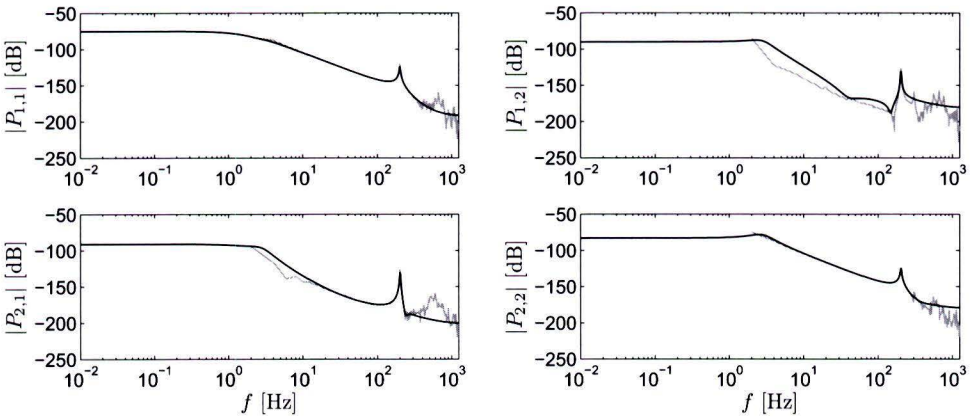


Figure 7.9: Measured FRF (gray) and 8th order model (black) of the NXT-A7 wafer stage in x- and y-direction.

to the base stone is the cable slab, which is not expected to have a significant stiffness with respect to the outside world. A zero slope might also be caused by parasitic stiffness effect of K-factor variations (Van de Wal et al., 2007), but this manifests itself mainly in the vertical plane. Hence, a -2 or -1 slope seems more appropriate. At steady state, the magnitude of the transfer function is given by:

$$P(\omega = 0) = \begin{bmatrix} -0.177 \times 10^{-3} & -0.032 \times 10^{-3} \\ -0.028 \times 10^{-3} & 0.073 \times 10^{-3} \end{bmatrix} \tag{7.3}$$

Especially compared to $P_{yy}(z)$, the off-diagonal entries have a significant magnitude.

The purpose of this model is to demonstrate the effect of MIMO feedforward compared to SISO feedforward. Although this model has some deficiencies, *i.e.*, the fact that the system has changed after the identification experiment and the low-frequent zero slopes, it is well suited for its purpose and is therefore used in Chapter 9.

7.3 Feedback Control

All feedforward experiments and simulations in Chapters 8 and 9 are performed with 25 Hz bandwidth diagonal feedback controllers. The Bode-diagram of the feedback controller in x-direction is shown in Figure 7.10. In all six DOFs, this controller consists of:

- a lead-lag filter with the zero and pole located at 8 Hz and 75 Hz.
- an integrator with the zero located at 5 Hz.
- a second order low-pass filter. In x- and y-direction, this is situated at 115 Hz, while in the other DOFs this is 125 Hz. For all DOFs, the damping is 0.7.

This 25 Hz bandwidth feedback controller does not push performance to the limit: in Quist (2010) a 90 Hz bandwidth MIMO feedback controller is designed. By using a low-bandwidth feedback controller, however, the performance of the feedforward controller expected to be more manifest in the experimental results.

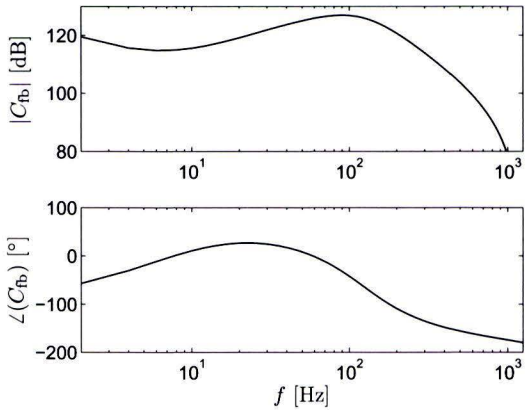


Figure 7.10: Bode-diagram of the feedback controller in x-direction.

In the next chapters, stable inversion feedforward is implemented in SISO experiments and MIMO simulations on the NXT-A7 wafer stage.

CHAPTER 8

Application to the NXT-A7 Wafer Stage: SISO Experiments

8.1 Introduction

In this chapter, the performance of stable inversion feedforward is validated. To start with, the model identified in Section 7.2.2 is validated. Thereafter, the performance of stable inversion feedforward on a second order reference trajectory is discussed in Section 8.3, followed by the effect of pre- and post-actuation in Section 8.4. Furthermore, it is investigated whether the robust stable inversion method of Section 6.3 can be applied to deal with the position dependent dynamics of the NXT-A7 wafer stage. This is addressed in Section 8.5. Finally, some concluding remarks are given.

8.2 Model Validation

In this section, the model of the equivalent x-direction of the NXT-A7 wafer stage that is identified in Section 7.2.2 is validated. Hereto, a fourth order reference trajectory and mass feedforward is used for both a simulation and an experiment to evaluate the performance of the obtained model. The parameters of the reference trajectory, shown in Table 8.1, are selected by gradually increasing them from zero until they were found to be suitable for model validation. The result is displayed in Figure 8.1. In this figure, the green line denotes the simulation result and the blue line is the experimental result. Furthermore, the dotted line represents the (scaled) reference acceleration. The upper plot of Figure 8.1 shows the move from 0 cm to 5 cm while the lower plot shows the error while moving in the opposite direction. It appears that there is quite a significant mismatch between simulation, with $e(t) \approx 0$, and experiment.

To further investigate the cause of the mismatch, the measured feedback signal is plotted in Figure 8.2. Here, two main features can be distinguished:

- During standstill, *i.e.*, when $t < 0.32$ s, 0.63 s $< t < 1.26$ s and $t > 1.58$ s, the feedback signal does not equal zero. This implies that there is a force required to keep the system at a certain position, which may be caused by the cable slab. Nevertheless, this is not

expected to be of any significance for feedforward control due to the integral action of the feedback controller.

- The feedback signal during the constant velocity part, *i.e.*, between 0.38 s and 0.56 s and between 1.33 s and 1.51 s, deviates significantly from the signal during standstill. Therefore, it is expected that there is in fact (viscous) damping present. This can also be caused by the cable slab or by electro-magnetic damping in the planar motor.

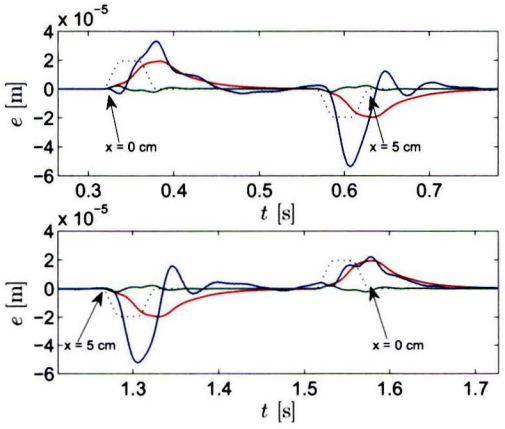


Figure 8.1: Scaled acceleration (dotted), measured error (blue) and simulated error using the original (green) and modified (red) models. Upper plot: forward movement. Lower plot: backward movement.

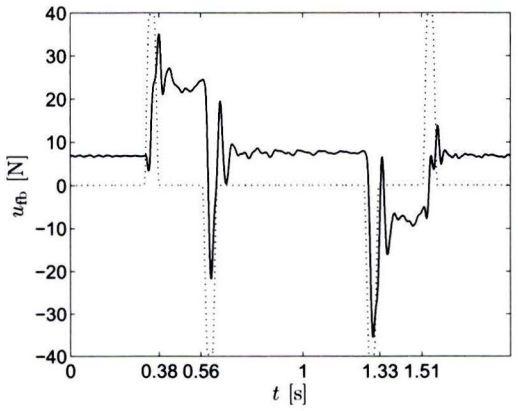


Figure 8.2: Scaled acceleration (dotted) and feedback signal using mass feedforward during the validation experiment.

To investigate the damping of the stage, experiments with fourth order trajectories with $v_{\max} = 0.016$ m/s, $v_{\max} = 0.020$ m/s, $v_{\max} = 0.024$ m/s, $v_{\max} = 0.028$ m/s and $v_{\max} = 0.032$ m/s have been conducted. It is chosen to use small velocities since these require only a short constant acceleration interval and the constant velocity part on $x \in [0, 5]$ cm is thus maximized.

From the measurements, the mean feedback signal during the constant velocity part is computed. This is subsequently corrected for the fact that a nonzero feedback signal is required during standstill, as discussed above. The results are plotted in Figure 8.3, where $\circ$ denotes the result for the constant velocity in positive and $\times$ in negative direction. Next, a line is fitted through the measurement results using a least-squares method. This result shows that there is a static friction of 0.32 N and a dynamic (viscous) damping of 74.2 Ns/m. This shows that this effect is caused by Eddy current damping; the EUV AD wafer stage was shown to have a static friction in x-direction of 3.8 N and a damping of 81 Ns/m.

While static friction typically is nonlinear, the viscous damping is a linear effect and can thus be included in the linear plant model. The Bode-diagram of the original and the modified model is shown in Figure 8.4.

Instead of having a low-frequent -2 slope, the modified model has a low-frequent -1 slope and a pole at 0.27 Hz. This shows why the system identification did not show any damping: the multisine excitation signals did not contain frequencies below 2 Hz. One could argue that multisine signals containing frequencies below 2 Hz should have been used in the identification experiments. However, since S_i already showed a significant variance at 2 Hz, a multisine

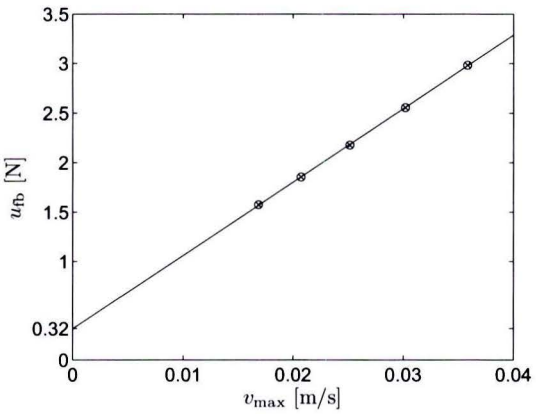


Figure 8.3: Mean feedback force u_{fb} during constant velocity v_{max} for various values for v_{max} (○ for positive, × for negative velocity) and first order fit (line).

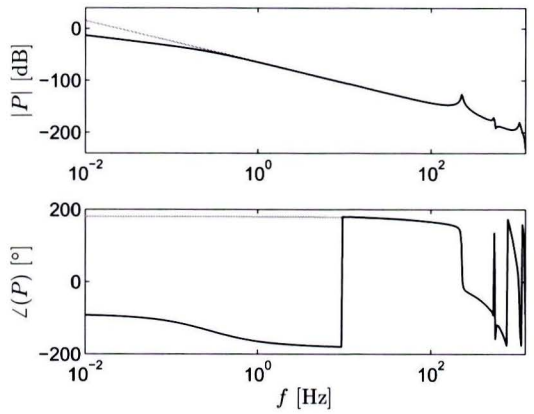


Figure 8.4: Bode-diagrams of the original (gray) and modified (black) models.

signal with frequencies below 0.27 Hz cannot be expected to lead to accurate identification of the low-frequent dynamics.

The simulated response of the modified model is also plotted in Figure 8.1. It appears that the modified model (black line) is a much better representation of the system dynamics (dashed) than the original model (gray). There is, however, still a large difference between the accuracy of the model at $x = 0$ cm and $x = 5$ cm. This follows from the fact that the upper plot (the forward movement) shows a better correspondence at the start of the move (at $x = 0$ cm), while during the backward movement (the lower plot), better correspondence can be observed at the end of the move, hence also at $x = 0$ cm. This is explained by the fact that the model is fitted on FRF data obtained by an identification experiment at $x = 0$ cm, while in reality motor-induced and position-dependent dynamics and disturbances are present. In the the FRFs measured in Section 7.2.2 a significant position dependency was shown, and here this also manifests itself in the time domain. As a consequence, the performance of nominal feedforward should be evaluated around $x = 0$ cm, *i.e.*, during acceleration of the forward motion or deceleration of the backward motion.

Remark 8.1 *The fact that the FRF is measured at $x = 0$ cm is also the reason that the scans are performed from $x = 0$ cm to $x = 5$ cm; this way, the model is accurate at either the start (forward scan) or the end (backward scan) of the movement. Had symmetrical scans from $x = -2.5$ cm to $x = 2.5$ cm been made, the model would not be that accurate during either acceleration or deceleration.*

The correction to the model and the remaining position-dependency are confirmed by Figure 8.5. Here, position and feedback are plotted if only k_{fa} is applied (43.5 kg) and if both k_{fa} and k_{fv} (74.2 Ns/m) are applied. In the latter case, there is less feedback required. Nevertheless, there is still some hysteresis visible, which can be attributed to static friction. Since these experiments are performed with small acceleration and velocity, the remaining shape is entirely due to position dependency of the stage and/or position dependency of the external disturbances.

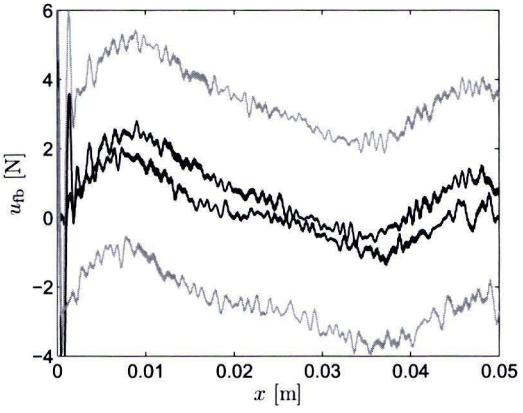


Figure 8.5: Position - feedback plot with only k_{fa} (gray) and k_{fa} and k_{fv} (black). The feedback u_{fb} is much smaller if k_{fv} is applied.

Concluding, it is remarked that the model is sufficiently accurate for stable inversion feedforward at $x = 0$ cm, but not at $x = 5$ cm. Therefore, good performance using stable inversion feedforward is expected at $x = 0$ cm while a decreased performance and hence a significantly larger servo error will be visible at $x = 5$ cm.

8.3 Stable Inversion Feedforward

Now that damping is added to the identified model and the resulting model is validated, it can be used for stable inversion feedforward. In Figure 8.6, a pole-zero map of the system is shown. The zeros of the model are located at:

- $z = -1.59$: this NMP zero is not expected to have a substantial effect since it is located at 1260 Hz.
- $z = -1.03 \pm 1.36j$: these NMP zeros are located at 907 Hz so are expected to have only a moderate effect.
- $z = 5.05$: with a frequency of 644 Hz, this is expected to be the dominant NMP zero, *i.e.*, its effect will be clearly visible in the feedforward signal.
- $z = -1$: this is a sampling zero, which is removed from the inverse system as discussed in Section 3.5.
- $z = 0.164 \pm 0.968j$: this MP complex pair is very lightly damped with a damping of 0.0129, which can introduce oscillations of the feedforward signal at the corresponding frequency (558 Hz).
- $z = 0.60 \pm 0.13j$: with a damping of 0.916, these MP zeros are nearly critically damped and will therefore not cause any vibrations at 213 Hz.

The Bode-diagrams of the decomposed inverse model are shown in Figure 8.7. The observations are similar to those made in Section 6.4: at low frequency, both $P_s^{-1}(z)$ and $P_u^{-1}(z)$ have a zero slope, are equal in magnitude but have a phase difference of 180°. Surprisingly, the resonance at 228 Hz does not turn up in either $P_s^{-1}(z)$ or $P_u^{-1}(z)$. In $P_s^{-1}(z)$, a resonance and anti-resonance turn up at 553 Hz and 566 Hz, hence they do not exactly coincide with those

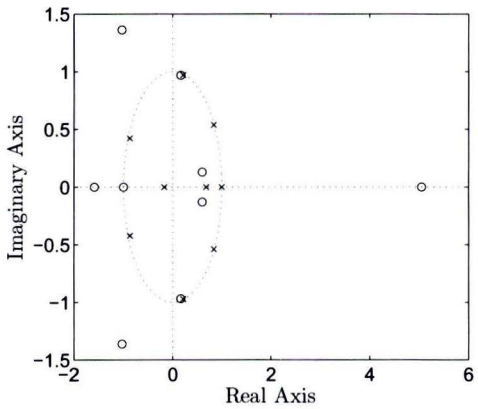


Figure 8.6: Pole-zero map of the 10th order plant model.

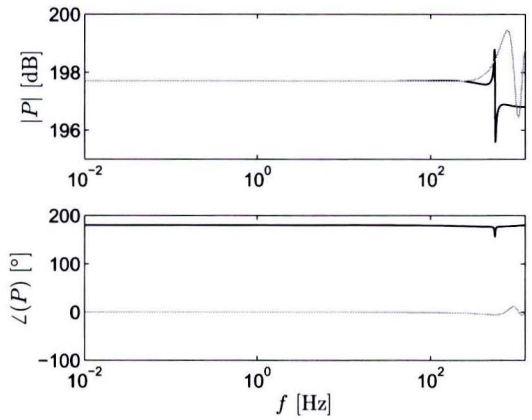


Figure 8.7: Bode-diagrams of the stable part $P_s^{-1}(z)$ (black) and unstable part $P_u^{-1}(z)$ (gray) of the decomposed inverse model $P^{-1}(z)$.

of the original system. Furthermore, the damping that is present in the model is not visible in the decomposed inverse model. There is no connection between the zeros of the inverse system $P^{-1}(z)$ on the one hand and the zeros of the separate parts $P_s^{-1}(z)$ and $P_u^{-1}(z)$ on the other hand. Using this decomposition, feedforward signals have been computed for various trajectories. Next, the resulting u_{ff} for a fourth order reference trajectory is discussed.

Fourth Order Reference Trajectory

In Table 8.1, the move parameters of second and fourth order reference trajectories are given. The acceleration, velocity and position of the fourth order trajectory are plotted in Figure 8.8. For this reference trajectory, two feedforward signals are computed: one with acceleration k_{fa}

Table 8.1: Parameters for the validation trajectory as well as the fourth and second order reference trajectories.

	validation	4th order	2nd order
$s_{\max}$ [m/s^4]	48×10^3	48×10^3	∞
$j_{\max}$ [m/s^3]	320	320	∞
$a_{\max}$ [m/s^2]	4.8	4.8	5.0
$v_{\max}$ [m/s]	0.032	0.2	0.2
$x_{\max}$ [m]	0.05	0.05	0.05

and velocity k_{fv} feedforward and one using the stable inversion approach of Chapter 6. The results are plotted in Figure 8.9, where it appears that there are only minor differences between both feedforward signals. Only during the constant acceleration interval, there is a small difference in magnitude visible, but the shape is the same in both cases. As a result, the performance of both signals is also approximately equal, and the servo error (not shown here) is dominated by other effects, such as a mismatch between model and real plant and disturbances.

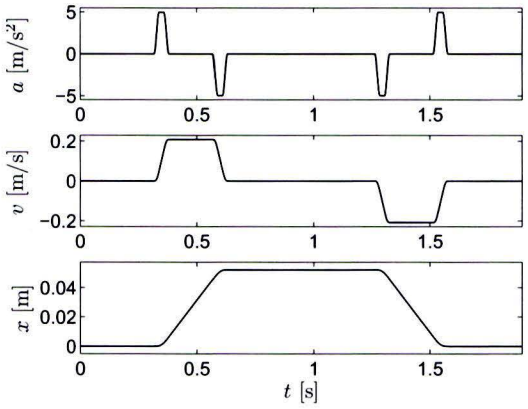


Figure 8.8: Acceleration (upper plot), velocity (middle plot) and position (lower plot) of a fourth order reference trajectory.

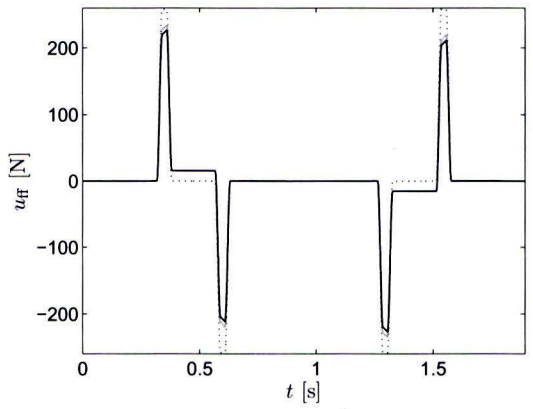


Figure 8.9: Scaled acceleration (dotted), u_{ff} using k_{fa} and k_{fv} (gray) and u_{ff} using stable inversion (black) corresponding to the trajectory in Figure 8.8. Both feedforward signals are nearly the same, the only difference appears during the constant acceleration part.

To be able to demonstrate the difference between stable inversion and acceleration and velocity feedforward, a more aggressive reference trajectory was required. However, it was undesired to increase acceleration and velocity because:

- Increasing the maximum acceleration leads to large rotations around the R_y axis with the used low-bandwidth feedback controller. One could consider increasing the controller bandwidth in R_y direction only. However, this may lead to a shift of the poles and zeros of the equivalent x-direction, hence suppressing the NMP behavior.
- Using the same maximum acceleration, increasing the maximum velocity leads to either a longer trajectory or a shorter constant velocity part. The first one is not desired due to the position dependent dynamics (Hamers, 2006), while a substantial constant velocity part is desired, since this resembles the scanning interval of a wafer stage.

Therefore, only snap and jerk have been gradually increased. Eventually, it has been decided to switch to third ($s_{\max} = \infty$) and, since these were still not aggressive enough, to second order trajectories ($s_{\max} = j_{\max} = \infty$). These results are discussed next.

Second Order Reference Trajectory

In Figure 8.10 the second order reference trajectory with the parameters of Table 8.1 is plotted. The feedforward signal for the forward motion is computed using stable inversion feedforward with 0.24 s pre- and post-actuation. This means that u_{ff} is defined on $t \in [0.052, 0.82]$ s. It is assumed that this is sufficient preview time and that the oscillations of the flexible dynamics of $P^{-1}(z)$ have converged at $t = 0.82$ s. Due to the specific choice of boundary conditions as discussed in Section 6.1, $u_{\text{ff,s}}$ and $u_{\text{ff,u}}$ are supposed to cancel each other for $t \notin [0.052, 0.82]$ s and the leading and trailing zeros can thus be zero. As a result, both components are 'reset' at 0.82 s and 1.70 s, as can be seen in Figure 8.11. This assumption is confirmed in Figure 8.12, where no steps are observed at either 0.052 s or 0.82 s. The components of the feedforward

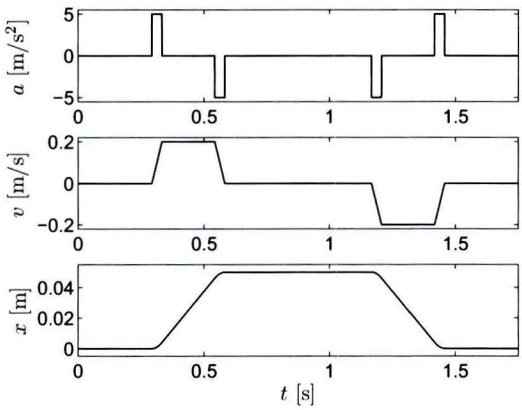


Figure 8.10: Acceleration (upper plot), velocity (middle plot) and position (lower plot) of a second order reference trajectory.

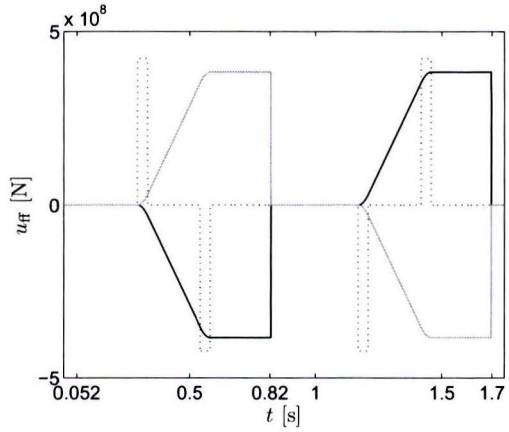


Figure 8.11: Scaled acceleration (dotted) and stable (black) and unstable (gray) part of the feedforward signal corresponding to the trajectory in Figure 8.10.

signal for the backward motion are simply the negative of those of the forward motion. This way of implementing stable inversion thus relies on linearity of the system. Alternatively, the feedforward signal could have been computed for the entire trajectory, but then it would not have been symmetrical for the forward and backward motion.

The components of the stable part $u_{\mathrm{ff,s}}$ and the unstable part $u_{\mathrm{ff,u}}$ of the feedforward signal both strongly resemble the reference trajectory, $u_{\mathrm{ff,u}}$ with positive and $u_{\mathrm{ff,s}}$ with negative sign. Furthermore, it appears that both components are very large, *i.e.*, in the order of magnitude of 10^8 N. This was expected due to the large magnitude of the corresponding Bode-diagrams (Figure 8.11). However, they nearly cancel each other and the resulting feedforward signal is plotted in Figure 8.12.

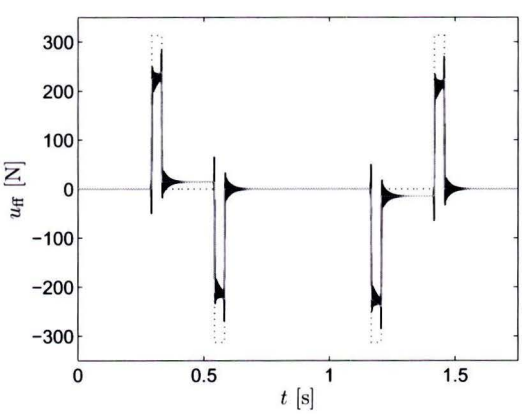


Figure 8.12: Scaled acceleration (dotted), u_{ff} using k_{fa} and k_{fv} (gray) and u_{ff} using stable inversion (black) corresponding to the trajectory in Figure 8.10 (see Figure 8.13 for details).

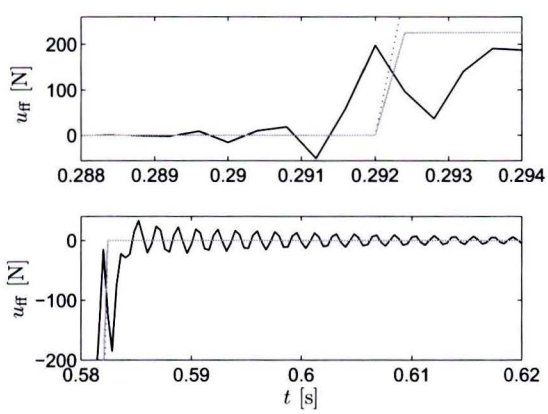


Figure 8.13: Zoom of Figure 8.12 during pre-actuation (upper plot) and post-actuation (lower plot).

The total feedforward signal is plotted in Figures 8.12 and 8.13, along with a k_{fa}, k_{fv} feedforward signal. It appears that in case of an aggressive second order trajectory, the difference between both feedforward signals is substantial, although the effect of the mass and damping is clearly visible in the feedforward signal computed by stable inversion. One of the most striking features is the pre-actuation (see upper plot of Figure 8.13). While a forward motion is requested, an oscillating pre-actuation force is present. The frequency of this oscillation is 909 Hz, hence can be attributed to the pair of complex NMP zeros at 907 Hz. A few samples before the system is supposed to start moving, this reaches its minimum: $u_{\text{ff}}(t = 0.2912) = -50$ N.

Furthermore, severe oscillations with a frequency of 565 Hz are present during the constant acceleration and constant velocity parts. These are caused by the anti-resonance of the model at 558 Hz, and should not be visible in the output.

Results

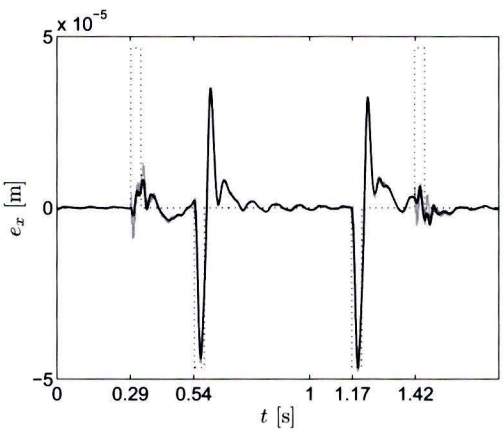


Figure 8.14: Scaled acceleration (dotted), e_x using k_{fa} and k_{fv} (gray) and e_x using stable inversion (black) (see Figure 8.15 for details).

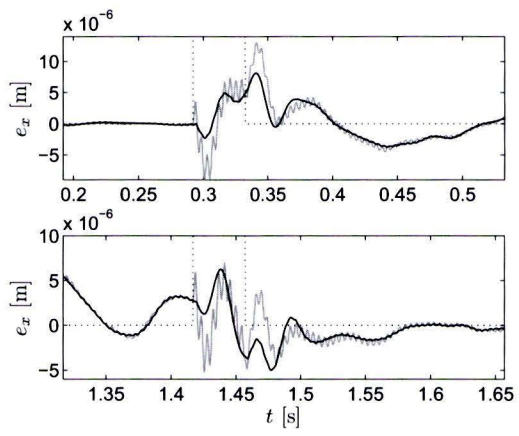


Figure 8.15: Zoom of Figure 8.14 during acceleration of the forward motion (upper plot) and deceleration of the backward motion (lower plot), both around $x = 0$ cm.

The servo errors in x-direction resulting from these feedforward signals are plotted in Figure 8.15. The most striking feature of the error is the large difference between on the one hand, the error during acceleration and deceleration at $x = 0$ cm, *i.e.*, around $t = 0.29$ s and $t = 1.42$ s, and on the other hand, around $x = 5$ cm, that is, at $t = 0.54$ s and $t = 1.17$ s. This can be attributed to the large difference between the low-frequent dynamic behavior of the NXT-A7 at $x = 0$ cm and $x = 5$ cm, as discussed in Section 7.2.2. In Quist (2010), it is argued that in case of feedback control this can easily be handled by robust control. This result, however, clearly shows that the effect of the position dependency on the performance of feedforward control is a big issue and it remains to be seen if this can be effectively addressed by robust feedforward control.

As mentioned before, this position dependency is addressed in Hamers (2006), but the corresponding compensation algorithms were not available during the experiments. Due to this

position dependency, one should assess feedforward performance around $x = 0$, see Figure 8.15. In this figure, the difference of the servo error resulting from the two feedforward signals of Figure 8.12 is substantial. A number of observations from this figure are:

- In the upper plot of Figure 8.15, the error during the acceleration and constant velocity part of the forward motion is shown. It appears that the pre-actuation of stable inversion does not lead to a significant servo error at $t < 0.292$ s. This confirms that the pre-actuation is in the zero direction of the system.
- The transient error using k_{fa}/k_{fv} feedforward is much larger than using stable inversion feedforward. During the constant acceleration phase, $|e_{\max}| = 9.1 \times 10^{-6}$ m using k_{fa}/k_{fv} feedforward and $|e_{\max}| = 4.9 \times 10^{-6}$ m using stable inversion.
- The use of k_{fa}/k_{fv} feedforward causes the system to oscillate at the frequency of the dominant resonance of the plant (228 Hz). This oscillation is hardly visible using stable inversion, since the inverse system has an anti-resonance at that frequency.
- The error during the constant velocity part, *i.e.*, between $t = 0.33$ s and $t = 0.54$ s, is approximately equal for both feedforward methods, except that stable inversion does not show oscillations at 228 Hz. The remaining error is attributed to the position dependency.
- The lower plot of Figure 8.15 shows the servo error during deceleration of the backward motion. Here, similar observations are made as during acceleration in positive direction: less oscillations around 228 Hz and smaller transient errors.

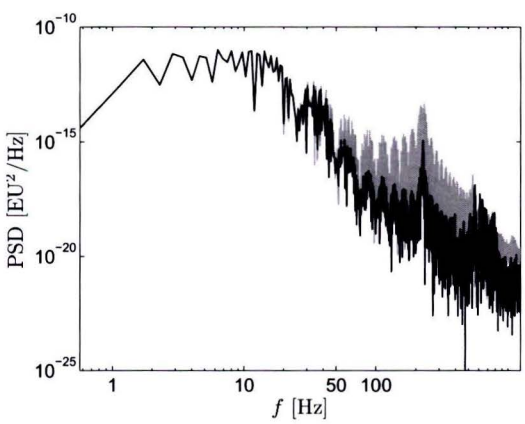


Figure 8.16: Power Spectral Density of e_x using k_{fa} and k_{fv} (gray) and stable inversion (black).

The power spectral densities (PSDs) of the entire error signals are plotted in Figure 8.16. Above approximately 50 Hz, stable inversion performs better than k_{fa}/k_{fv} feedforward. Only around the second resonance frequency, the difference is small, which can be attributed to the mismatch between the real plant and the plant model at that frequency. This shows that advanced feedforward is able to improve servo performance above the bandwidth of the feedback controller, which has only a small gain at high frequencies due to controller roll-off.

The feedback signals in x-direction applied during these experiments are plotted in Figure 8.17. Again, a force of 17 N is required to keep the stage at $x = 0$. Although the force is constant during one experiment, it will be different after the stage has been terminated and

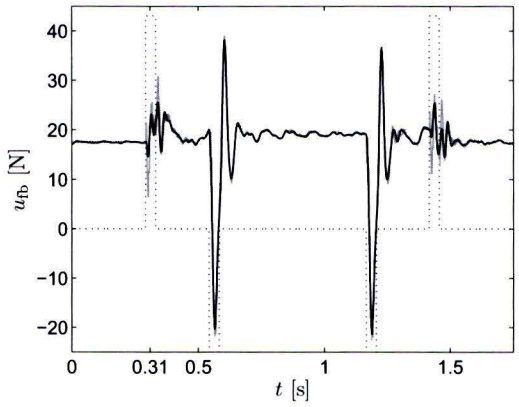


Figure 8.17: Scaled acceleration (dotted) and feedback in x-direction using k_{fa} and k_{fv} (gray) and stable inversion (black).

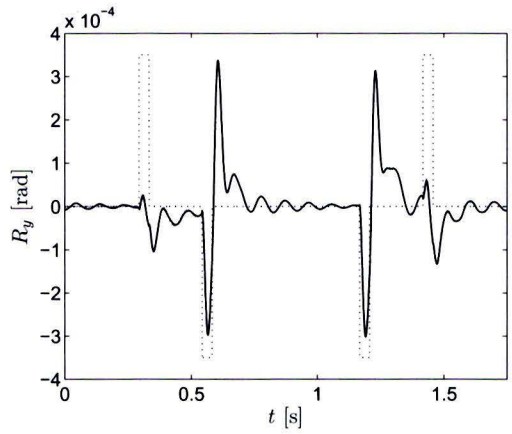


Figure 8.18: Rotation R_y using k_{fa} and k_{fv} (gray) and stable inversion (black). The rotation R_y, which is nearly the same for both feedforward signals, strongly resembles the servo error in Figure 8.14.

re-initiated. As mentioned before, it is expected that this is caused by the cable slab, but the reason for the differences between experiments is unclear at this point.

Furthermore, the feedback signals strongly resemble the servo errors. Using stable inversion, the maximum feedback action during acceleration in positive direction, *i.e.*, for 0.29 s $< t <$ 0.33 s, is approximately 8 N (without the constant force). This corresponds to 3.6% of the total actuation, which is still relatively large.

Although in this experiment only the x-direction is considered, the NXT-A7 is controlled in six DOFs. Of the remaining five directions, the R_y rotation is the most interesting. This rotation shows also shows a strong correlation with the servo error in x-direction; note the resemblance between the e_x (Figure 8.14) and R_y (Figure 8.18). It is interesting to see if this rotation can be prevented by using MIMO feedforward. This is addressed in the next chapter.

8.4 Pre- and Post-Actuation

In order to assess the effect of pre- and post-actuation, stable inversion has been applied with various values of T_{pre} and T_{post}. This is discussed in this section. Hereto, the second order reference trajectory of the previous section is used. Varying T_{pre} mainly affects the first part of the move in positive direction, while varying T_{post} mainly concerns u_{ff} and e_x after the move. Therefore, in the upcoming figures, only the relevant part of the interval is shown.

Pre-Actuation

As mentioned in Section 6.4.2, the feedforward signal u_{ff} is essentially cut off if insufficient pre-actuation is used. In Figure 8.19, the stable inversion feedforward signals for $T_{\text{pre}} = 0.0$ ms, $T_{\text{pre}} = 0.8$ ms and $T_{\text{pre}} = 1.6$ ms are shown. In all cases, $T_{\text{post}} = 0.24$ s. Using $T_{\text{pre}} = 0.8$ ms and $T_{\text{pre}} = 1.6$ ms, a large peak force of 198 N appears just before the system is supposed to move. Furthermore, the signal with $T_{\text{pre}} = 1.6$ ms shows a negative force of -50.0 N.

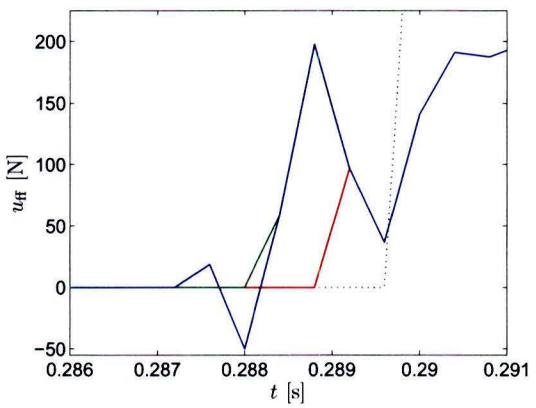


Figure 8.19: Scaled acceleration (dotted) and $u_{\text{ff}}(t)$ using $T_{\text{pre}} = 0.0$ ms (red), $T_{\text{pre}} = 0.8$ ms (green) and $T_{\text{pre}} = 1.6$ ms (blue). $T_{\text{post}} = 0.24$ s in all cases.

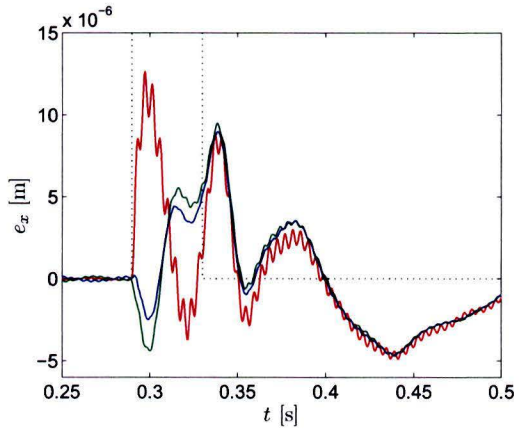


Figure 8.20: Servo errors in x-direction corresponding to the feedforward signals of Figure 8.19. Using $T_{\text{pre}} = 0.0$ ms (red), the servo error is largest and shows severe oscillations.

In Figure 8.20, the resulting servo errors are plotted. It appears that both the negative force of -50 N and the positive peak force of 198 N do not result in a servo error for $t < 0.29$ s, implying that u_{ff} operates in the zero-direction of the plant. However, the positive force turns the maximum transient error from 12.6 μm to -4.4 μm. This is further reduced by the negative force, leading to -2.5 μm. Furthermore, the 228 Hz oscillation is present if $T_{\text{pre}} = 0.0$ ms, so it can be concluded that pre-actuation is required to suppress oscillations at the main resonance frequency.

A further decrease of the servo error could not be obtained by increasing T_{pre}. It can therefore be concluded that the servo error using $T_{\text{pre}} = 1.6$ ms is dominated by a mismatch between plant and plant model and by the position dependent dynamics.

Post-Actuation

A similar exercise is performed with various values for T_{post}. In this case, $T_{\text{pre}} = 0.24$ s and $T_{\text{post}} = 0.0$ ms, $T_{\text{post}} = 4.0$ ms and $T_{\text{post}} = 8.0$ ms. This is plotted in Figure 8.21. Using $T_{\text{post}} = 0.0$ s, $u_{\text{ff}}(t) = 0$ for $t \geq 1.449$ s. The difference between $T_{\text{post}} = 4.0$ ms and $T_{\text{post}} = 8.0$ ms is much smaller and somewhat difficult to see. However, if $T_{\text{post}} = 4.0$ ms, $u_{\text{ff}}(t) = 0$ for $t \geq 1.453$ s while it keeps oscillating if $T_{\text{post}} = 8.0$ ms. As discussed, the oscillation have a frequency of 565 Hz and can hence be attributed to the first anti-resonance of the plant.

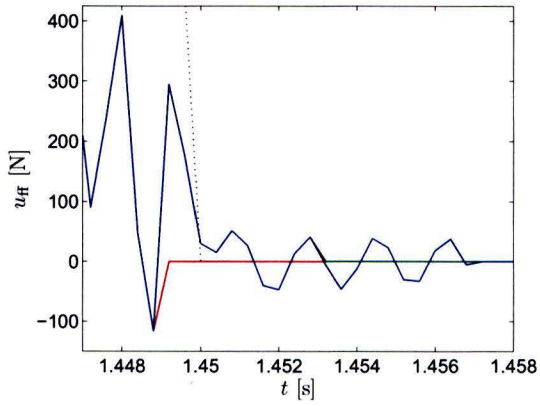


Figure 8.21: Scaled acceleration (dotted) and $u_{\text{ff}}(t)$ using $T_{\text{post}} = 0.0$ ms (red), $T_{\text{post}} = 4.0$ ms (green) and $T_{\text{post}} = 8.0$ ms (blue). $T_{\text{pre}} = 0.24$ s in all cases.

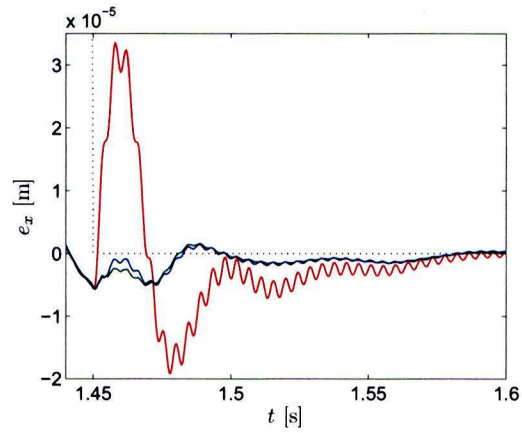


Figure 8.22: Servo errors in x-direction corresponding to the feedforward signals of Figure 8.21. Using $T_{\text{post}} = 0.0$ ms (red), the servo error is by far the largest and shows severe oscillations.

The errors in the x-direction resulting from these feedforward signals are shown in Figure 8.22. It appears that 4.0 ms post-actuation to a large extent attenuates the transient error after the move. If $T_{\text{post}} = 0.0$ ms, $|e_x|_{\max} = 34$ μm. If $T_{\text{post}} \geq 0.4$ ms, this is reduced to $|e_x|_{\max} = 5.7$ μm. In case of $T_{\text{post}} = 8.0$ ms, there is only little improvement. Besides suppressing the large transient error immediately after the move, post-actuation also reduces the oscillations at the resonance frequency.

8.5 Robust Feedforward

As mentioned before, in Quist (2010) it is shown that a robust feedback controller can effectively handle the position dependency of the NXT-A7 wafer stage. In this section, it is investigated whether this is also the case for stable inversion. Hereto, the multiplicative uncertainty weighting filter displayed in Figure 8.23 is used. Using (6.32) a 'robust' plant is computed. The Bode-diagrams of both systems are displayed in Figure 8.24. It appears that the robust plant is slightly larger at high frequencies. At low frequencies, there is hardly any difference visible. Nevertheless, zooming in on this plot it can be seen that at 1 Hz the magnitude of the nominal plant is 0.06 dB lower. As a result, the robust *inverse* system has a smaller magnitude and will thus also result in a feedforward signal with a smaller magnitude.

This resulting feedforward signal is shown in Figure 8.25. Indeed, during the constant acceleration part, the robust feedforward signal is smaller than the nominal one. However, in this figure it is difficult to distinguish both signals. Therefore, in Figures 8.26 and 8.27 enlargements of this figure are shown during pre-actuation and during the constant acceleration and velocity interval.

In the upper plot of Figure 8.26 it can be seen that, during pre-actuation, the feedforward signal oscillates at 208 Hz, which is close to the resonance frequency of the system. Fur-

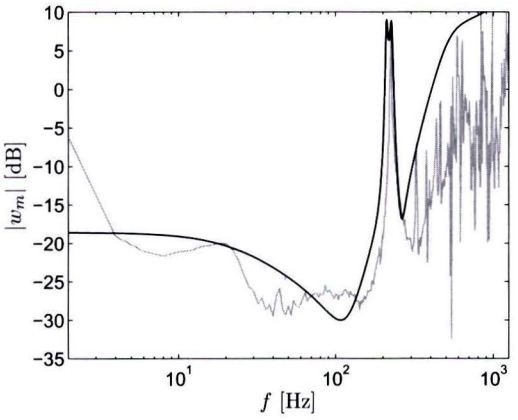


Figure 8.23: Frequency response functions of perturbed systems (gray) and multiplicative uncertainty weighting filter $w_m(z)$ (black).

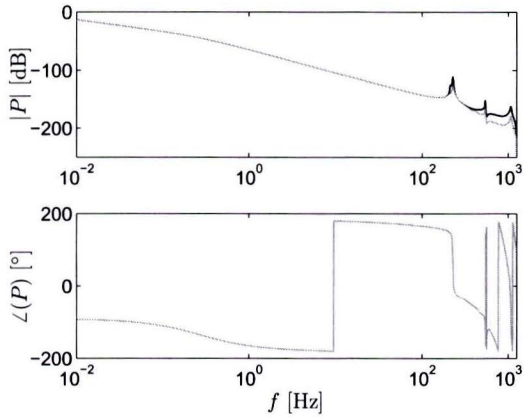


Figure 8.24: Bode-diagrams of the plant model $P(z)$ and the 'robust' plant model $P(z)\left(1+\frac{1}{2}w_m(z)w_m^*(z)\right)$.

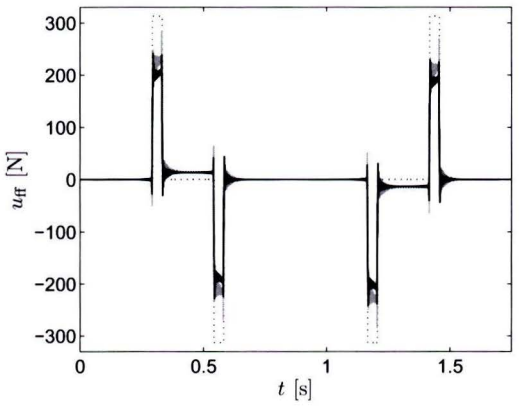


Figure 8.25: Scaled acceleration (dotted) and nominal (gray) and robust (black) feedforward signals (see Figures 8.26 and 8.27 for details).

thermore, in the lower plot it appears that the peaks during pre-actuation are smaller: the minimum at $t = 0.2904$ s is -28 N compared to -50 N of the nominal signal. Furthermore, the positive peak is 158 N instead of 198 N.

During the constant acceleration interval (upper plot of Figure 8.27), the magnitude of the nominal signal is larger than the robust one. The oscillations caused by the anti-resonance of the model at 558 Hz have a smaller amplitude. This also holds for the constant velocity part (lower plot).

The servo error resulting from this feedforward signal is plotted in Figures 8.28 and 8.29. Despite the oscillations in u_{ff} at 208 Hz, there is no significant error visible during pre-actuation. Nevertheless, the error using robust feedforward is much larger than using nominal feedforward, both at $x = 0$ cm and $x = 5$ cm.

This can be explained as follows: for robust stable inversion, the nominal model obtained at

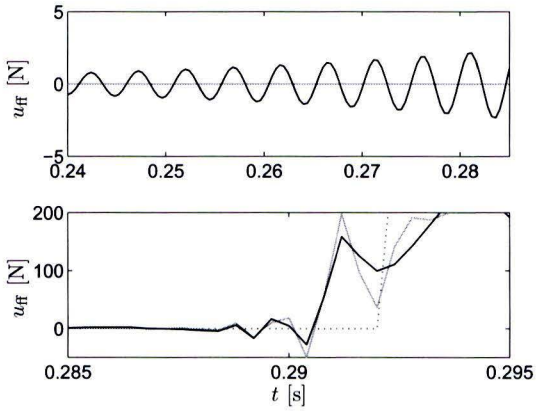


Figure 8.26: Enlargements of Figure 8.25 during pre-actuation.

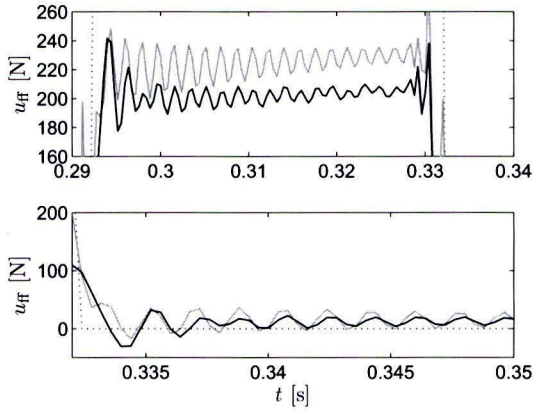


Figure 8.27: Enlargements of Figure 8.25 during constant acceleration (upper plot) and constant velocity (lower plot). As expected, robust feedforward has smaller magnitude than nominal feedforward.

$x = 0$ cm is used for $P_o(z)$ in (6.32) and an overbound is fitted such that $\mathcal{P}$ also contains the model obtained at $x = 5$ cm. This is graphically depicted by the empty circle in Figure 8.30. The smallest model set containing both measurements, on the other hand, is represented by the gray circle.

Since the empty circle is much larger than the gray circle, it can be concluded that using the model obtained from the measurement at $x = 0$ cm results in an unnecessarily conservative robust feedforward filter and hence in poor servo performance. Since the performance is optimized for the entire model set instead of the worst case present in this model set, it is even more important to minimize the model set $\mathcal{P}$ than using a common robust control solution. This result does raise the question whether a multiplicative uncertainty model is the most suited, or that a more advanced uncertainty model, *e.g.*, as in Wu and Zou (2009) is required since robust solutions can only decrease the gain.

Due to the poor choice of nominal model in this section, it cannot be concluded whether robust stable inversion is able to improve servo performance compared to nominal stable inversion in the presence of model uncertainty and possible position dependent dynamics. In order to assess the performance of robust stable inversion, the experiments have to be performed using a different nominal model $P_o(z)$, leading to an overbound $w_m(z)$ with smaller magnitude and a less conservative model set $\mathcal{P}$.

8.6 Summary

In this section, the most important conclusions of the experiments with the NXT-A7 wafer stage are summarized:

- Stable inversion feedforward results in a smaller transient error and less oscillatory behavior than velocity and acceleration feedforward, as shown in Section 8.3. Looking

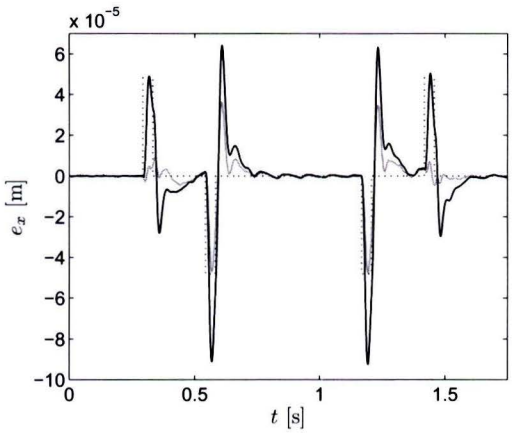


Figure 8.28: Scaled acceleration (dotted) and servo error in x-direction using nominal (gray) and robust (black) stable inversion feedforward.

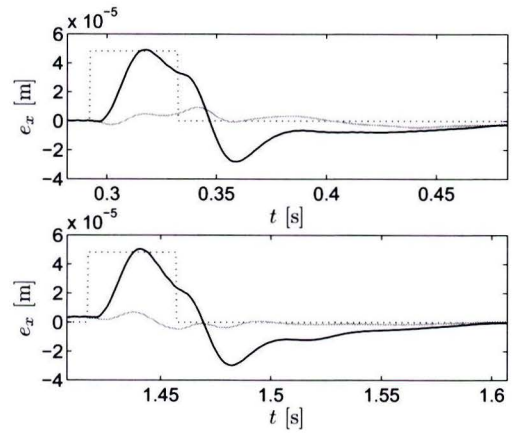


Figure 8.29: Enlargements of Figure 8.28

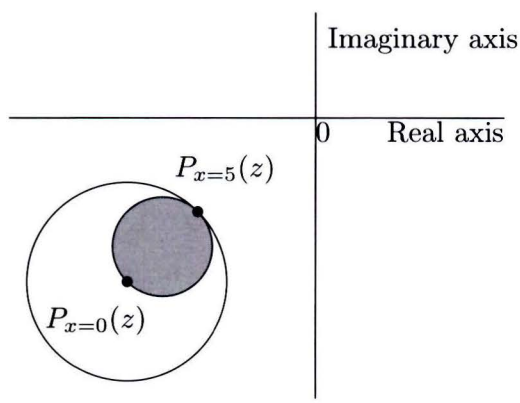


Figure 8.30: Graphical representation of the smallest possible model set $\mathcal{P}$ containing both measurements (gray circle) and the model set used in the experiments (empty circle). This shows why the used model set is unnecessary conservative.

at the PSD plot, it is concluded that stable inversion is able to reduce the high-frequent content of the servo error, above the bandwidth of the feedback controller. Nevertheless, in order to also get good low-frequent behavior, accurate models are essential.

- The models obtained by system identification techniques that are designed for feedback control do not meet the accuracy requirements at low frequencies imposed by feedforward control, as is shown in Section 8.2. Adding damping to the outside world proved to be an important improvement of the model quality, but further improvements are desired.

- In Section 8.4, it is confirmed that sufficient pre- and post-actuation is required if stable inversion feedforward is applied. Choosing T_{pre} too small leads to a transient error and oscillations at the start of the interval, while choosing T_{post} too small introduces similar effects immediately after the interval.

- The position dependent behavior of the NXT-A7 is a major issue for feedforward control.

This can be seen from the substantial difference in the servo errors during acceleration and deceleration at $x = 0$ cm and $x = 5$ cm. This shows that either a dedicated compensation algorithm or robust feedforward is required.

- The performance of robust stable inversion depends to a large extent on the nominal model that is selected from the model set. A poor choice results in unnecessarily conservatism. Further experiments should show whether robust stable inversion can lead to improved servo performance of the NXT-A7 wafer stage.

CHAPTER 9

Application to the NXT-A7 Wafer Stage: MIMO Simulations

9.1 Introduction

Due to the reduced mass reduction and therefore decreased stiffness of future wafer stages, the interaction between the various DOFs of a wafer stage is bound to increase. Mass and snap feedforward are typically based on the assumption that there is no interaction, while the current MIMO FIR filters are only able to compensate for high-frequent cross-talk. In theory, stable inversion is able to explicitly account for interaction. This is shown in this chapter using simulations with the 2×2 model of the NXT-A7 wafer stage introduced in Section 7.2.3. Even more than for SISO systems, it appears that a model that accurately describes the low-frequent dynamics is a necessity.

The purpose of this chapter is to demonstrate the difference between SISO and MIMO feedforward. Interaction between the DOFs is an issue even if a reference trajectory is defined for only one DOF while the other DOFs are supposed to stay at rest. Therefore, the reference trajectory of Chapter 8 is used. First, SISO feedforward is discussed, followed by MIMO feedforward and the results. All simulations are performed in closed-loop, with 25 Hz diagonal PID feedback controllers.

9.2 SISO Feedforward in the x-direction

There are two possibilities to compute SISO feedforward:

- Use stable inversion on $P_{xx}(z)$, *i.e.*, the plant in x-direction. This way, interaction is not taken into account.
- Use stable inversion on the equivalent plant (see Section 7.2.1) in the x-direction, *i.e.*:

$$P_{eq}(z) = P_{xx} - P_{xy}(z)\left(I + C_{yy}(z)P_{yy}(z)\right)^{-1} C_{yy}(z)P_{yx}(z) \tag{9.1}$$

  This still results in a SISO feedforward signal, but does take interaction into account. Nevertheless, it does not take into account that the output in the y-direction is supposed

to stay at zero. Note that the model used in Chapter 8 actually is the equivalent model of the x-direction.

The Bode-diagrams of both inverse systems are shown in Figure 9.1. Based on visual inspection of the Bode-diagrams, it appears that these systems are nearly the same. However, they do lead to significantly different results, as will be shown. Both inverse systems have a zero low-frequent slope. Although this is not an accurate description of the real system (see Chapter 8), it is still useful to demonstrate the use of MIMO feedforward. Both SISO

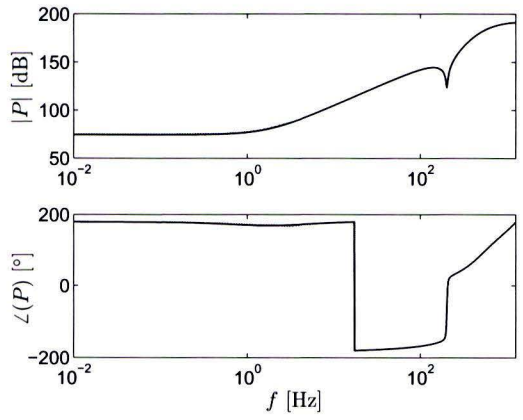


Figure 9.1: Bode-diagram of the inverse of the (1,1) entry of $P(z)$ (gray) and of the equivalent SISO plant in the x-direction (black). Both Bode-diagrams nearly coincide.

feedforward signals are plotted in Figure 9.3, where the black line denotes the feedforward signal resulting from inverting the equivalent plant. The stiffness[1] manifests itself in the fact that $u_{\text{ff}}(0.58 \text{ s} < t < 1.17 \text{ s}) \neq 0$ N. Although both SISO feedforward signals have more or less the same shape, the difference is clearly visible. Before looking at the resulting servo errors, MIMO feedforward is discussed.

9.3 MIMO Feedforward

In Figure 9.2, the Bode-diagram of the inverse 2×2 plant $P(z)$ is shown. The multivariable zeros of this system are located at $z = 0.18$, $z = 0.50 \pm 0.52j$ (MP) and $z = 1.82$ and $z = 1.14 \pm 1.86j$ (NMP). At steady state, the magnitude of the inverse system is given by:

$$P^{-1}(\omega = 0) = \begin{bmatrix} -0.528 \times 10^4 & -0.235 \times 10^4 \\ -0.206 \times 10^4 & 1.29 \times 10^4 \end{bmatrix} \tag{9.2}$$

This shows that the contribution of the off-diagonal entries is significant. A proper inverse plant is obtained by shifting according to the procedure described in Lunenburg (2009, Section 2.4). Next, MIMO stable inversion is applied, leading to the feedforward signals plotted

[1]Recall from Section 7.2.3 that the used model has low-frequent zero slopes in all entries of $P(z)$, while this is not likely in the true system.

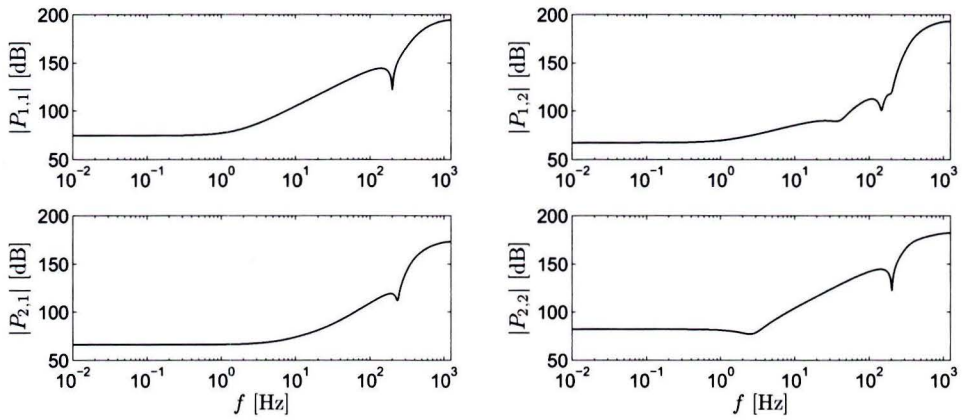


Figure 9.2: Bode-diagram of the inverse 2×2 MIMO plant in the x- and y-direction.

in Figure 9.3. In the x-direction, the feedforward signal nearly coincides with that of the SISO equivalent plant. The shape of the feedforward signal in the y-direction can be mainly attributed to the stiffness. However, apart from this contribution feedforward forces up to 20 N are observed.

9.4 Results

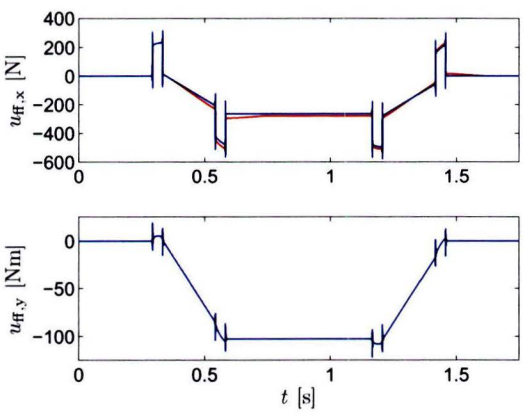


Figure 9.3: Feedforward in x-direction (upper plot) and y-direction (lower plot, only containing a blue line) using SISO feedforward with the (1,1) entry of the plant (red), SISO feedforward with the equivalent plant in x-direction (green) and MIMO feedforward (blue). The blue and green line are hard to distinguish by visual inspection.

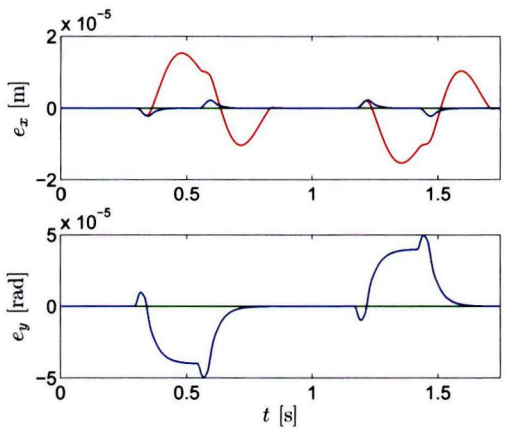


Figure 9.4: Servo errors in x-direction (upper plot) and y-direction (lower plot) using the feedforward signals of Figure 9.3. In the y-direction, the difference between the red and the blue line is maximally 4.1×10^{-8} m.

The servo errors resulting from the feedforward signals in Figure 9.3 are plotted in Figure 9.4. Inverting $P_{xx}(z)$ leads to the largest servo errors: $\max e_x = 15.3$ μm and $\max e_y = 49.6$ μm. If the equivalent plant in the x-direction is used instead, the servo error in x-direction is

significantly reduced: $\max e_x = 2.22\ \mu$m. In y-direction, however, the error is equally large. Finally, using MIMO feedforward nearly exact tracking is achieved in both DOFs: $\max e_x =$ 3.83 nm and $\max e_y = 1.00$ nm.

Concluding, it is remarked that:

- Stable inversion is able to successfully deal with the interaction that is present in MIMO systems.
- Nevertheless, the feedforward signals shown in this chapter are not realistic in practice. This is caused by the low-frequent behavior of the model, which has zero slopes up to 2 Hz. In the real system, the behavior around these frequencies will be different. In order to implement MIMO stable inversion in practice, the accuracy of the models at low frequencies should be enhanced.

CHAPTER 10

Conclusions & Recommendations

In the near future, lightweight wafer stages are required to avoid infeasible designs for actuators and amplifiers. Furthermore, motion profiles will become increasingly aggressive and the wafer size may increase from 300 mm to 450 mm to meet the throughput requirements. This asks for a feedforward method that effectively deals with the resonant dynamics of these stages. In this thesis, stable inversion feedforward is investigated and implemented on the NXT-A7 wafer stage. The conclusions of this research are discussed in the next section, followed by recommendations for future research.

10.1 Conclusions

- A non-minimum phase system can only achieve exact tracking of an arbitrary reference trajectory if both a suitable feedforward signal and corresponding initial conditions are used. Nevertheless, the states of a system cannot be brought from zero to arbitrary initial conditions using bounded pre-actuation on a bounded interval while keeping the output equal to zero. The resulting servo error depends on the amount of pre-actuation and the location of the NMP zeros, and can be made arbitrarily small by using a sufficiently long pre-actuation interval. This is shown in Chapter 5 using the lifted system representation.

- It is often argued that the inverse of a strictly proper system with relative degree ρ cannot be used as a feedforward controller, since this inverse system is non-proper. In case of discrete systems, however, ρ poles can be added to the inverse system at $z = 0$ to make it proper. As a consequence, the reference trajectory has to be delayed by ρ samples before being fed into the feedback loop. This way, the relative degree is no longer a restriction on using the inverse system for feedforward control.

- A system with non-minimum phase zeros results in an unstable inverse. If this is implemented straightforwardly as a feedforward filter, an unbounded feedforward signal is the result. Using stable inversion (Chapter 6), the inverse system is decomposed into a stable and unstable part. Subsequently, boundary conditions for the stable part are defined at the start of the interval, while the boundary conditions for the unstable part are defined at the end of the interval. The latter can be implemented by computing

the solution to the unstable part in reverse time. The resulting feedforward signal is bounded. The stable inversion procedure is graphically depicted in Figure 10.1.

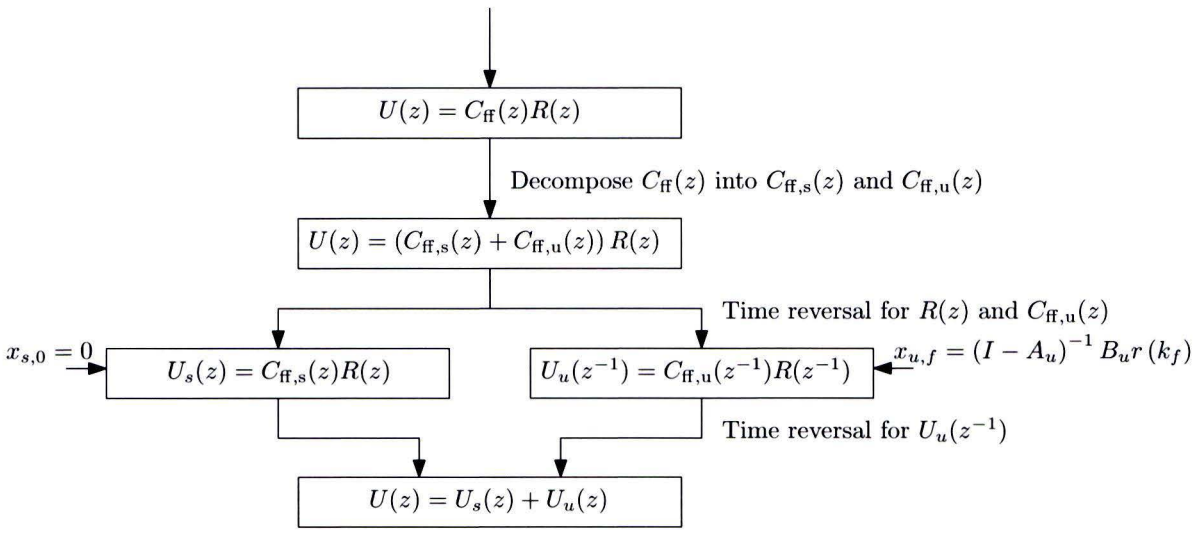


Figure 10.1: Flow-diagram of stable inversion.

- Stable inversion feedforward leads to smaller transient errors and reduces high-frequent oscillations compared to velocity and acceleration feedforward, see Chapter 8. In the frequency domain, it can be seen that stable inversion is able to reduce the high-frequent content of the servo error, above the bandwidth of the feedback controller. Nevertheless, sufficient pre- and post-actuation is hereto required, otherwise transient errors and oscillations are introduced at the start of the move and immediately after the move. Furthermore, stable inversion is able to explicitly deal with the interaction between the various DOFs of the plant, as shown in Chapter 9.

- The system identification procedure for feedback control is not sufficiently accurate for feedforward control, since the variance at low frequencies is too large. This manifests itself, *e.g.*, in the fact that the low-frequent damping is not present in the models obtained by system identification. (see Sections 7.2.2 and 8.2). In this research, the low-frequent damping has been experimentally determined at different constant velocities and added *a posteriori*. In case of MIMO systems, such a modification is much more difficult and it is even more important to have a suitable identification method.

- The position dependent behavior of the NXT-A7 wafer stage has a significant impact on performance of the wafer stage: there is a substantial difference between accelerating and decelerating at $x = 0$ cm and $x = 5$ cm. During the experiments, the compensations algorithms for these phenomena were disabled. From experiments with robust stable inversion to address the position dependency, it can be concluded that using a suitable nominal model is essential to avoid unnecessary conservatism, keeping in mind that a robust solution can only decrease the gain. Since the nominal model used in Section 8.5 proved to be a poor choice, no definitive conclusion on robust stable inversion can be drawn.

10.2 Recommendations for Future Research

Along with these conclusions, a number of recommendations for further research include:

- investigation of feedforward control relevant system identification. An accurate model of the system to be controlled is essential for inversion-based feedforward, especially since tuning of advanced feedforward is more difficult than tuning of restricted complexity feedforward, since more parameters are involved. Nevertheless, during the experiments on the NXT-A7 wafer stage (Chapter 8), it appeared that the models obtained by system identification techniques for feedback control (Oomen, 2010; Quist, 2010; Van Herpen, 2009b), were not sufficiently accurate at low frequencies, even though a low-bandwidth feedback controller had been used during the identification experiments. It seems that lower order reference trajectories (second or third order) in combination with a fixed structure model are more suited to measure the low-frequent characteristics. The higher order dynamics can subsequently be identified by methods that are similar to feedback control relevant identification. Alternatively, Machine-In-the-Loop tuning could be used to tune the feedforward parameters online. Both approaches, however, require further research.

- further investigation of feedforward control and model uncertainty. The performance of robust stable inversion on the NXT-A7 wafer stage was not satisfactory. This can partly be attributed to the fact that the uncertainty was relatively large due to the position dependency of the system. However, the robust feedforward signal had a smaller amplitude than the nominal signal, while, looking at the FRFs, a larger amplitude was required. Since robust feedforward solutions generally lead to a decrease of the magnitude of $C_{\text{ff}}(z)$, it should be investigated how to choose a suitable nominal model and how to define the corresponding uncertainty model. Position dependency might also be dealt with by other means than robustness, which has not been addressed in this thesis.

- online implementation of stable inversion. The feedforward signals throughout this thesis are all computed offline. In practice, however, it is desirable to be able to implement a feedforward filter online. How to implement stable inversion online remains subject to further research, in particular how to choose pre- and post-actuation intervals and how to discard the need for the batch-wise computation of the feedforward signal.

- a joint feedforward/input shaping approach. Throughout this thesis, it has been assumed that the reference trajectory is designed without taking the plant dynamics into account. In Section 5.2, it is shown that non-minimum phase systems cannot track arbitrary reference trajectories which possess components in the direction of any NMP zeros. However, by using knowledge of the plant dynamics, these components can be filtered out of the reference trajectory. The absence of these components in the reference trajectory can subsequently be taken into account in the design of the feedforward filter, potentially enabling the possibility to achieve exact tracking with a causal feedforward filter.

APPENDIX A

Numerical Integration

The aim of feedforward is to improve tracking of a certain reference signal. In this appendix, the design of a reference trajectory and, more specifically, digital implementation is addressed. First, it is shown how numerical integration is commonly performed (Section A.1) and why this method does not give satisfactory results. Thereafter, an alternative solution is presented in Section A.2.

A.1 Commonly Used Methods

In a trajectory planning algorithm (Lambrechts et al., 2005), switching times are calculated such that the highest bounded derivative of $r(t)$ is a block-shaped signal. This is subsequently integrated until $r(t)$ appears. In Lambrechts et al. (2005) the importance of implementation aspects is recognized. Therefore, the switching times that are originally calculated are modified so that they match sampling instants. Subsequently, numerical integration is carried out by putting a number of Euler integrators in series[1]. A forward (explicit) Euler integrator is given by the transfer function:

$$\frac{T_s}{z-1} \tag{A.1}$$

while a backward (implicit) Euler integrator can be written as:

$$\frac{zT_s}{z-1} \tag{A.2}$$

Putting discrete integrators in series results in signals that are not synchronized correctly. It is therefore proposed to *a posteriori* utilize (subsample) delays to synchronize $r(t)$ and its derivatives. A delay of $\frac{1}{2}T_s$ can be approximated by averaging the values of the current and previous sample (Van Donkelaar, 2003). The following example shows why subsample delays are required if backward Euler integrators are put in series:

Example A.1 *Consider a simple mass with continuous time transfer function $P_c(s) = \frac{1}{ms^2}$. Furthermore, a trajectory planning algorithm calculates a desired acceleration $r_a(t)$ which is integrated twice to obtain the desired position,* i.e., $r(t) = \iint r_a(t)dtdt$. *If feedforward*

[1]In (Lambrechts et al., 2005) forward Euler integrators are described while in the MATLAB toolbox the command cumsum is implemented, which implies a backward Euler integrator.

was implemented in continuous time, exact tracking could be obtained by using the desired acceleration as input for a simple gain m. In the Laplace domain, the output would then be given by:

$$Y(s) = \frac{1}{ms^2} mR(s)s^2 = R(s) \tag{A.3}$$

However, this is commonly implemented in discrete time, so what would this look like in that case? Including the ZOH and the sampler, the discrete time transfer function is given by $P_d(z) = \frac{T_s^2}{2m}\frac{z+1}{z^2-2z+1}$ where T_s denotes the sampling frequency. Furthermore, the trajectory planner now calculates a discrete desired acceleration $r_a(k)$ which has to be numerically *integrated twice to obtain $r(k)$. Using two backward Euler integrators in series, $r(k)$ becomes:*

$$r(k) = \left(\frac{zT_s}{z-1}\right)^2 r_a(k) = \frac{z^2T_s^2}{z^2-2z+1} r_a(k) \tag{A.4}$$

Using:

$$u_{\text{ff}}(k) = mr_a(k) \tag{A.5}$$

the output becomes:

$$Y(z) = P_d(z)mR_a(z) = \frac{T_s^2}{2m}\frac{z+1}{z^2-2z+1} m \frac{z^2-2z+1}{z^2T_s^2} R(z) = \frac{z+1}{2z^2} R(z) \tag{A.6}$$

which implies that output $y(k+2)$ equals $\frac{1}{2}(r(k+1)+r(k))$ which is the average of the past two samples.

This example shows that putting multiple Euler integrators in series does not lead to accurate results. Using slightly more advanced numerical integration algorithms such as the trapezoid rule (Heath, 2002) yields better results for the second integration step, but also fails for third or fourth order reference trajectories. An alternative is presented in the next section.

A.2 An Alternative Numerical Integration Method

The discrete system $P_d(z)$ in Example A.1 is obtained by viewing the continuous time system $P_c(s)$ in combination with a ZOH and a sampler, see Figure A.1.

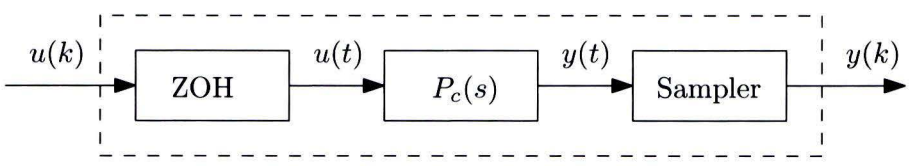


Figure A.1: Digital representation of a dynamical system.

Similarly, a backward Euler integrator can be viewed as a continuous time integrator with a ZOH and a sampler. In Lunenburg (2009) it is shown that a double continuous integrator including ZOH and sampler does *not* equal two separate continuous integrators including ZOH and sampler in series (see Figure A.2). Therefore, it is suggested to integrate the reference signals in a similar way, see Figure A.3. If, *e.g.*, a fourth order reference trajectory is designed,

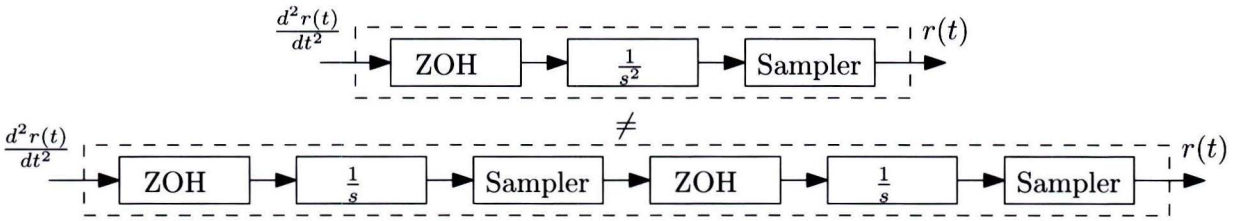


Figure A.2: A double continuous integrator including ZOH and sampler does not equal two separate continuous integrators including ZOH and sampler in series.

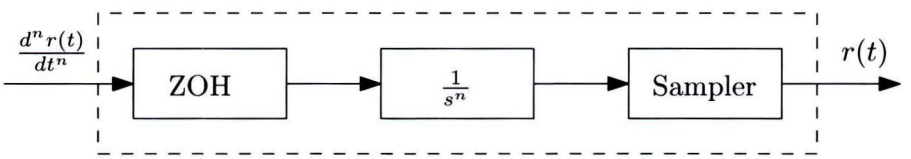


Figure A.3: Block representation of alternative numerical integration method.

$s(k)$ is a block shaped signal. Subsequently, $j(k)$, $a(k)$, $v(k)$ and $r(k)$ have to be computed directly from $s(k)$. Hereto, (A.7)–(A.10) can be used.

$$\frac{1}{s} \rightarrow T_s \frac{1}{z-1} \tag{A.7}$$

$$\frac{1}{s^2} \rightarrow \frac{T_s^2}{2} \frac{z+1}{z^2-2z+1} \tag{A.8}$$

$$\frac{1}{s^3} \rightarrow \frac{T_s^3}{6} \frac{z^2+4z+1}{z^3-3z^2+3z-1} \tag{A.9}$$

$$\frac{1}{s^4} \rightarrow \frac{T_s^4}{24} \frac{z^3+11z^2+11z+1}{z^4-4z^3+6z^3-4z+1} \tag{A.10}$$

Remark A.1 *Note that care has to be taken in implementing the transfer functions* (A.7)–(A.10)*, since very small scaling factors are involved,* e.g., $\frac{T_s^4}{24} = 6.7 \times 10^{-17}$ *with* $f_s = 5$ kHz. *The best results are obtained when unscaled trajectories are calculated which are* a posteriori *scaled. As an example, if a snap profile is available, the position can be computed by:* `x = Ts^4/24*lsim(tf([1 11 11 1],[1 -4 6 -4 1],Ts),snap)` *rather than putting* $\frac{T_s^4}{24}$ *inside the transfer function.*

A second possibility is to rewrite (A.7)–(A.10) *into state-space systems, resulting in the ma-*

trices:

$$A = \begin{bmatrix} 1 & T_s & \frac{T_s^2}{2} & \frac{T_s^3}{6} & \frac{T_s^4}{24} \\ 0 & 1 & T_s & \frac{T_s^2}{2} & \frac{T_s^3}{6} \\ 0 & 0 & 1 & T_s & \frac{T_s^2}{2} \\ 0 & 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{A.11a}$$

$$C = \begin{bmatrix} 1 & T_s & \frac{T_s^2}{2} & \frac{T_s^3}{6} & \frac{T_s^4}{24} \\ 0 & 1 & T_s & \frac{T_s^2}{2} & \frac{T_s^3}{6} \\ 0 & 0 & 1 & T_s & \frac{T_s^2}{2} \\ 0 & 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad D = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{A.11b}$$

The outputs of this system represent the fourth, third, second and first integral of the input (the final output is only direct feedthrough, hence equals the input). This has implications for the result in both the time and the frequency domain, which is discussed next.

A.2.1 Time Domain

To illustrate the difference between the original and the alternative integration methods, a typical second order reference trajectory is designed. It appears that the alternative integration method results in a velocity and position that exactly coincide with the analytical solution. In Figure A.4, it is shown that Euler and trapezoidal integration yield different results. The velocity profile resulting from Euler integration advances the analytical solution by one sample, but the shape coincides. The shape of the velocity using trapezoidal integration, on the other hand, is slightly different.

Looking at the position, it appears that all three shapes are approximately equal, but using Euler integration there is a 1.5 sample advance and using trapezoidal integration this is 0.5 sample. Had integration been performed using fourth order profiles, the differences would have been even larger.

A.2.2 Frequency Domain

Next to the time domain advantages described in the previous section integrating according to (A.7)–(A.10) has another benefit. If the ZOH and sampler introduce additional zeros (see Section 3.5), these are typically located at the negative real axis (Åström et al., 1984). If a zero at $z \approx -1$ is inverted, this may result in a feedforward signal u_{ff} which oscillates at the Nyquist frequency. However, looking at the Bode diagrams in Figure A.5, it becomes evident that this oscillation is excited much less by the trajectory computed by integrating according to (A.10).

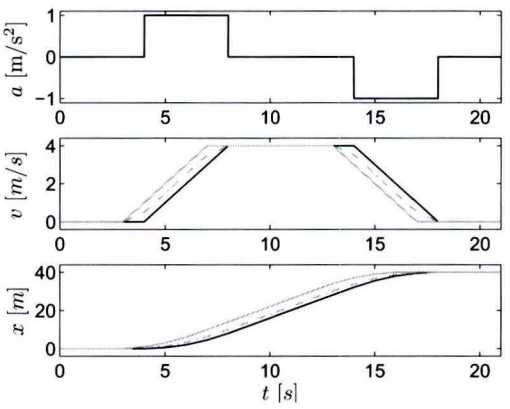


Figure A.4: Acceleration (upper), velocity (middle) and position (bottom) using Euler integration (gray, solid), trapezoidal integration (grey, dash-dotted) and alternative integration method (black).

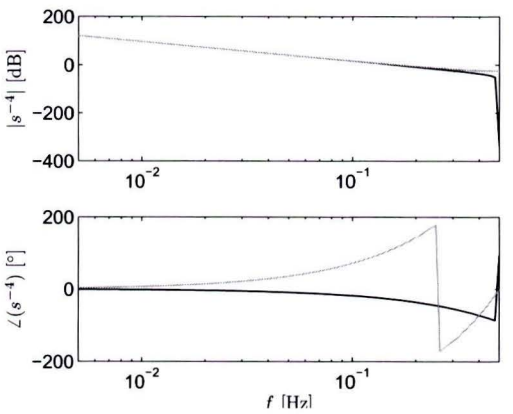


Figure A.5: Bode-diagram of 4 backward Euler integrators in series (gray) and 4 continuous integrators with one ZOH and sampler (black).

APPENDIX B

Interpreting the Singular Values and Vectors

Although the lifted system representation is increasingly used, there is in general little attention for the meaning of the singular values and singular vectors. However, when the generalized inverse (Ben-Israel and Greville, 1974) is calculated, these are important, since a number of singular directions is essentially discarded from the inverse. To gain more insight into what is actually removed, the interpretation of the singular values and singular vectors is investigated in this appendix. First, unstable poles and zeros are discussed, followed by the remaining singular vectors.

B.1 Unstable poles and zeros

In Hashemi and Hammond (1996); Lunenburg (2009), it is shown empirically that the singular values corresponding to unstable zeros are very small compared to the other 'clustered' singular values, but a mathematical explanation is lacking. In Dijkstra (2003) use is made of the notion that a zero implies that there exists some initial condition x_0 and a nonzero input signal $\boldsymbol{u}_0$ for which the output is identically zero, *i.e.*:

$$\boldsymbol{y} = J\boldsymbol{u}_0 + J_0 x_0 \tag{B.1}$$

The corresponding singular value is then given by:

$$\sigma_0(J) = \frac{\|J_0 x_0\|}{\|\boldsymbol{u}_0\|} \tag{B.2}$$

According to Skogestad and Postlethwaite (2005) $\boldsymbol{u}_0$ is increasing exponentially, which implies:

$$\lim_{N\to\infty} \|\boldsymbol{u}_0\| = \infty \tag{B.3}$$

and hence:

$$\lim_{N\to\infty} \sigma_0(J) = \lim_{N\to\infty} \frac{\|J_0 x_0\|}{\|\boldsymbol{u}_0\|} = 0 \tag{B.4}$$

This result is confirmed by Hashemi and Hammond (1996), where the converse is also shown: unstable poles in a system lead to singular values of the corresponding impulse response matrix

which are several orders of magnitude larger than the 'clustered' singular values. Again, this reference lacks a mathematical proof, but using similar reasoning it can be argued that:

$$\sigma_n(J) = \frac{\|J_0 x_0\|}{\|\boldsymbol{u}_n\|} \tag{B.5}$$

and

$$\lim_{N\to\infty} \|\boldsymbol{u}_n\| = 0 \tag{B.6}$$

which implies:

$$\lim_{N\to\infty} \sigma_n(J) = \lim_{N\to\infty} \frac{\|J_0 x_0\|}{\|\boldsymbol{u}_n\|} = \infty \tag{B.7}$$

B.2 Singular Vectors

In the previous section it is shown that the singular vectors corresponding to unstable poles and zeros are exponentially increasing or decreasing vectors. Still, it is not clear what the remaining singular values and vectors represent. First, the singular value decomposition is viewed in more detail. The matrices U, Σ and V resulting from the singular value decomposition $J = U\Sigma V^T$ can be computed by using:

$$\begin{aligned} U &= \text{eigenvectors of } JJ^T \\ \Sigma &= \sqrt{\text{diag}\left(\text{eig}\left(JJ^T\right)\right)} \\ V &= \text{eigenvectors of } J^T J \end{aligned} \tag{B.8}$$

Since J is Toeplitz, $J^T J$ equals JJ^T transposed along the anti-diagonal. This implies that the eigenvalues and eigenvectors of $J^T J$ and JJ^T are equal but in reverse time (Dijkstra, 2003):

$$u\left(\frac{1}{2}N - k\right) = \text{sign}(k) v\left(\frac{1}{2} + k\right) \tag{B.9}$$

A few input singular vectors from the convolution matrix of the NMP flexible cart system (Section 6.4) are plotted in Figure B.2. In this figure, it can be seen that the singular vectors represent harmonic oscillations with a specific frequency. To develop insight in the meaning of these singular vectors, the dominant frequency of the oscillations is determined for all singular vectors. Since every singular vector corresponds to a specific singular value, it is now possible to assign the singular values to a certain frequency. The result of this is plotted in Figure B.3: it appears that this resembles the Bode-magnitude diagram of the system. If multiple frequencies have a substantial contribution to one singular vector, this means that the magnitude of the transfer function is equal at these frequencies.

Remark B.1 *Note that the singular vector corresponding to the smallest singular value,* i.e., *the singular value related to the NMP zero, is not an oscillation but an exponentially increasing vector (see Figure B.1). Therefore, it is omitted from Figure B.3.*

A formal derivation of these observations can be found in Dijkstra (2003). In this reference, the derivation is based on the limit case with infinitely large J. Choosing a truncated J leads

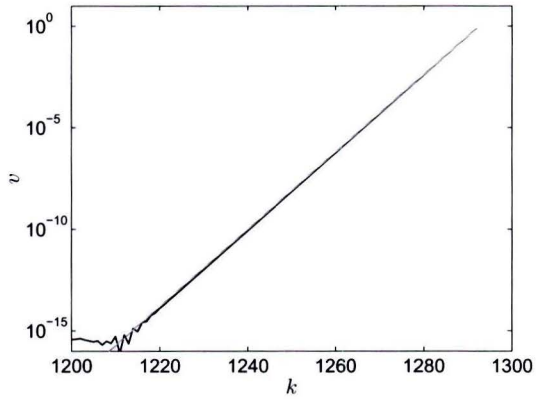


Figure B.1: Singular vector corresponding to the NMP zero of the flexible cart system plotted on a logarithmic scale.

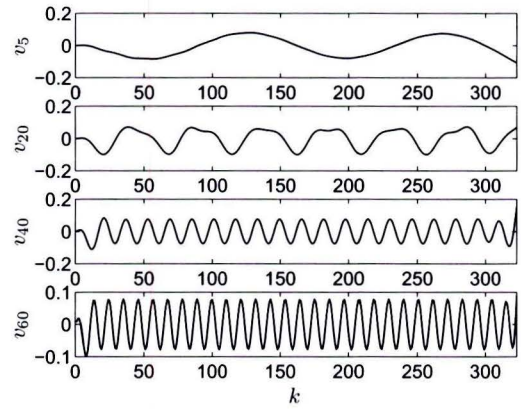


Figure B.2: Fifth, twentieth, fortieth and sixtieth column of V.

to a larger difference between the magnitude of the singular values and the magnitude of the frequency response function (FRF). This can be explained by the fact that the FRF is not properly defined for finite time. In case J is the impulse response matrix of a stable system, one could look at the length of the impulse response to obtain an estimate of the size of J necessary to give a reasonable estimate of the FRF.

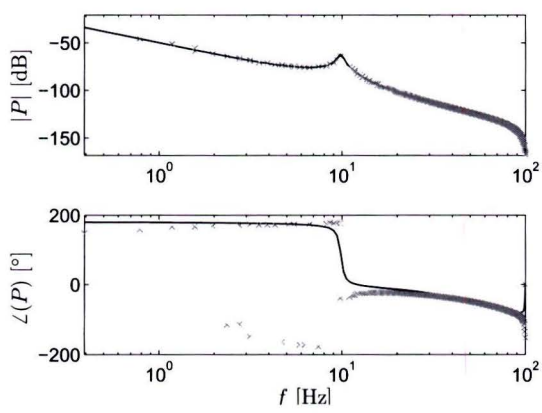


Figure B.3: Bode-diagram of the NMP system (black, solid) and singular values of the impulse response matrix (gray, crosses).

With this knowledge, it becomes clear what is discarded by using the pseudo-inverse. Removing the small singular values corresponding to the NMP zeros implies removing exponentially increasing (hence unstable) components from the feedforward signal u_{ff}. Besides the NMP behavior, discarding small singular values in calculating the pseudo-inverse implies that certain frequencies that may be present in the reference signal r, will not be present in the feedforward signal u_{ff}.

APPENDIX C

FIR Filters

C.1 Properties of FIR Filters

A Finite Impulse Response (FIR) filter is a discrete-time filter which owes its name to the fact that its impulse response settles to zero in a finite number of sample intervals. The number of samples before the output settles to zero depends on the order of the FIR filter. The transfer function of a FIR filter is given by:

$$H_{\text{FIR}} = \sum_{i=0}^{n} f_i z^{-i} \tag{C.1}$$

where n denotes the order of the FIR filter and f_i are the filter coefficients. Basically, it can be seen as a transfer function with all its poles located at $z = 0$. This means that the output of a FIR filter is given by:

$$y(k+n) = f_0 u(k+n) + f_1 u(k+n-1) + f_2 u(k+n-2) + \ldots + f_n u(k) \tag{C.2}$$

This shows that if the input of the FIR filter is bounded, the output is also bounded. A schematic representation of a third order FIR filter is shown in Figure C.1. The output of an Infinite Impulse Response (IIR) filter, on the other hand, not only depends on the past and present inputs, but also on the past outputs, meaning that not all poles are located at $z = 0$. Hence, if one or more poles are outside the unit disc, the IIR filter becomes unstable.

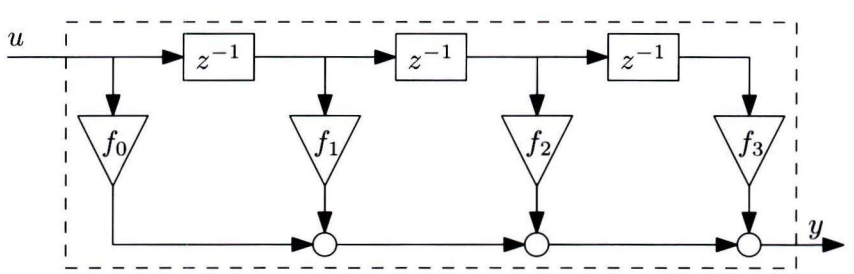


Figure C.1: Schematic representation of a third order FIR filter

C.2 A State-Space Representation

A n^{nth}-order FIR filter can also be written in a state-space representation:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned} \tag{C.3}$$

with:

$$A = \begin{bmatrix} 0 & 1 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

$$C = \begin{bmatrix} f_n & f_{n-1} & \cdots & f_2 & f_1 \end{bmatrix} \qquad D = \begin{bmatrix} f_0 \end{bmatrix}$$

This confirms that all poles are located in $z = 0$, since the eigenvalues of A are zero. Note that a state-space representation is not unique, so other representations are possible as well.

C.3 Interpretation of FIR Filter coefficients

In Hennekens (2009) it is shown how FIR filter coefficients can be represented as gains of time derivatives of the input signal. Take, *e.g.*, a feedforward filter:

$$C_{\text{ff}}(s) = k_0 + k_1 s + k_2 s^2 + k_3 s^3 + k_4 s^4 \tag{C.4}$$

Then, it can be shown that this can be represented as a FIR filter with coefficients:

$$f_0 = +\left(k_0 + \frac{k_1}{T_s} + \frac{k_2}{T_s^2} + \frac{k_3}{T_s^3} + \frac{k_4}{T_s^4}\right) \tag{C.5}$$

$$f_1 = -\left(\frac{k_1}{T_s} + \frac{k_2}{T_s^2} + \frac{k_3}{T_s^3} + \frac{k_4}{T_s^4}\right) \tag{C.6}$$

$$f_2 = +\left(\frac{k_2}{T_s^2} + \frac{k_3}{T_s^3} + \frac{k_4}{T_s^4}\right) \tag{C.7}$$

$$f_3 = -\left(\frac{k_3}{T_s^3} + \frac{k_4}{T_s^4}\right) \tag{C.8}$$

$$f_4 = +\left(\frac{k_4}{T_s^4}\right) \tag{C.9}$$

FIR filters are inherently stable, which an advantageous property. Nevertheless, (complex) poles are very hard to approximate using FIR filters. This is serious drawback, since it requires large order FIR filters which are difficult to implement in realtime.

Bibliography

Åström, K., Hagander, P., and Sternby, J. (1984). Zeros of sampled systems. *Automatica*, 20(1):31–38.

Baggen, M., Heertjes, M., and Kamidi, R. (2008). Data-based feed-forward control in MIMO motion systems. In *Proceedings of the 2008 American Control Conference*, pages 3011–3016, Seattle, Washington, United States.

Bamieh, B., Pearson, J., Francis, B., and Tannenbaum, A. (1991). A lifting technique for linear periodic-systems with applications to sampled-data control. *Systems & Control Letters*, 17(2):79–88.

Ben-Israel, A. and Greville, T. (1974). *Generalized inverses: theory and applications*. Pure and applied mathematics: a series of texts and monographs. Wiley-Interscience.

Boerlage, M. (2006). MIMO jerk derivative feedforward for motion systems. *Proceedings of the 2006 American Control Conference*, 2006:3892–3897. Minneapolis, Minnesota, United States.

Boerlage, M., Tousain, R., and Steinbuch, M. (2004). Jerk derivative feedforward control for motion systems. *Proceedings of the 2004 American Control Conference*, 5:4843–4848. Boston, Massachusetts, United States.

Bristow, D., Tharayil, M., and Alleyne, A. (2006). A survey of iterative learning control. *IEEE Control Systems Magazine*, 26(3):96–114.

Butterworth, J., Pao, L., and Abramovitch, D. (2008). The effect of nonminimum-phase zero locations on the performance of feedforward model-inverse control techniques in discrete-time systems. In *Proceedings of the 2008 American Control Conference*, pages 2696–2702, Seattle, Washington, United States.

Compter, J. (2004). Electro-dynamic planar motor. *Precision Engineering*, 28(2):171–180.

Devasia, S. (2002). Should model-based inverse inputs be used as feedforward under plant uncertainty? *IEEE Transactions on Automatic Control*, 47(11):1865–1871.

Devasia, S., Chen, D. G., and Paden, B. (1996). Nonlinear inversion-based output tracking. *IEEE Transactions on Automatic Control*, 41(7):930–942.

Dijkstra, B. (2003). *Iterative Learning Control with applications to a wafer stage*. PhD thesis, Delft University of Technology.

Franklin, G., Powell, J., and Emami-Naeini, A. (2002). *Feedback Control of Dynamic Systems.* Prentice Hall, fourth edition edition.

Hamers, M. (2006). Modelling of actuator-related servo errors sources in a permanent magnet synchronous planar motor. Master's thesis, Eindhoven University of Technology.

Hashemi, S. and Hammond, J. (1996). The interpretation of singular values in the inversion of minimum and non-minimum phase systems. *Mechanical Systems and Signal Processing*, 10(3):225–240.

Heath, M. T. (2002). *Scientific Computing, An Introductory Survey.* McGraw-Hill, second edition.

Heertjes, M. and Van de Molengraft, R. (2009). Set-point variation in learning schemes with applications to wafer scanners. *Control Engineering Practice*, 17(3):345–356.

Hennekens, D. (2009). *MIMO Wafer Stage Control using Iterative Feedback Tuning.* Msc thesis, Eindhoven University of Technology.

Hoagg, J. and Bernstein, D. (2007). Nonminimum-phase zeros: Much to do about nothing - classical control revisited. part II. *IEEE Control Syst Mag*, 27(3):45–57.

Hoyle, D., Hyde, R., and Limebeer, D. (1991). An $\mathcal{H}_\infty$ approach to two degree of freedom design. In *Proceedings of the 30th IEEE Conference on Decision and Control Part 1 (of 3)*, volume 2, pages 1581–1585, Brighton, England.

Isidori, A. (1995). *Nonlinear Control Systems.* Spinger.

Jemaa, L. and Davison, E. (2003). Performance limitations in the robust servomechanism problem for discrete time periodic systems. *Automatica*, 39(6):1053–1059.

Lambrechts, P., Boerlage, M., and Steinbuch, M. (2005). Trajectory planning and feedforward design for electromechanical motion systems. *Control Engineering Practice*, 13(2):145–157.

Lee, C. and Salapaka, S. (2009). Robust broadband nanopositioning: Fundamental trade-offs, analysis, and design in a two-degree-of-freedom control framework. *Nanotechnology*, 20(3):paper 035501.

Luenberger, D. G. (1977). Dynamic equations in descriptor form. *IEEE Transactions on Automatic Control*, AC-22(3):312–321.

Lunenburg, J. (2009). Inversion-based feedforward design for beyond rigid body systems: A literature survey. Literature survey, Philips Applied Technologies / Eindhoven University of Technology.

Middleton, R., Chen, J., and Freudenberg, J. (2001). Achievable performance and sensitivity integral constraints in preview control. In *40th IEEE Conference on Decision and Control*, volume 4, pages 3075–3080, Orlando, Florida, United States.

Middleton, R., Chen, J., and Freudenberg, J. (2004). Tracking sensitivity and achievable $\mathcal{H}_\infty$ performance in preview control. *Automatica*, 40(8):1297–1306.

Miu, D. (1991). Physical interpretation of transfer function zeros for simple control systems with mechanical flexibilities. *Journal of Dynamic Systems, Measurement and Control*, 113(3):419–424.

Ooi, C., Okajima, H., and Asai, T. (2006). Constraints in servo system design for non-minimum phase system. In *SICE-ICASE International Joint Conference*, pages 970–973, Busan.

Oomen, T. (2010). *System Identification for Robust and Inferential Control with Applications to ILC and Precision Motion Systems.* PhD thesis, Eindhoven University of Technology.

Oomen, T., van de Wijdeven, J., and Bosgra, O. (2009). Suppressing intersample behavior in iterative learning control. *Automatica*, 45(4):981–988.

Peeters, F., Wortelboer, P., and Daraei, M. (2000). Twincan levelling using advanced feed-forward on a twin stage. Technical Report CTB595-00-2131, Philips Centre for Industrial Technology.

Pintelon, R. and Schoukens, J. (2001). *System Identification: A Frequency Domain Aproach.* The Institute of Electrical and Electronics Engineers, Inc.

Prempain, E. and Postlethwaite, I. (2001). Feedforward control: A full-information approach. *Automatica*, 37(1):17–28.

Qiu, L. and Davison, E. J. (1993). Performance limitations of non-minimum phase systems in the servomechanism problem. *Automatica*, 29(2):337–349.

Quist, S. (2010). Improving wafer stage motion performance through robust-control-relevant model set identification and multivariable control. Msc thesis, Eindhoven University of Technology.

Rigney, B. P., Pao, L. Y., and Lawrence, D. A. (2006). Model inversion architectures for settle time applications with uncertainty. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 6518–6524. San Diego, California, United States.

Silverman, L. (1968). Properties and application of inverse systems. *IEEE Transactions on Automatic Control*, 13(4):436–437.

Silverman, L. (1969). Inversion of multivariable linear systems. *IEEE Transactions on Automatic Control*, 14(3):270–276.

Skogestad, S. and Postlethwaite, I. (2005). *Multivariable Feedback Control.* John Wiley and Sons Ltd, 2nd revised edition.

Sogo, T. (2002). Stable inversion for nonminimum phase sampled-data systems and its relation with the continuous-time counterpart. In *Proceedings of the 41st IEEE Conference on Decision and Control*, volume 4, pages 3730–3735, Las Vegas, Nevada, United States.

Sogo, T. (2010). On the equivalence between stable inversion for nonminimum phase systems and reciprocal transfer functions defined by the two-sided laplace transform. *Automatica*, 46(1):122–126.

Tomizuka, M. (1987). Zero phase error tracking algorithm for digital-control. *Journal of Dynamic Systems, Measurement, and Control*, 109(1):65–68.

Van de Wal, M. (2001). Model-based feedforward controller design for electromechanical motion systems: An exploratory study. Technical report, ASML Control Research, Philips CFT. CFT595-01-2055.

Van de Wal, M., Van de Vrande, B., and Hamers, M. (2007). Nxt-a7 basic servo tests. Technical Report APT536-07-2781, Philips Applied Technologies, Mechatronics, HTC-7.

Van de Wijdeven, J. (2008). *Iterative Learning Control design for uncertain and time-windowed systems.* PhD thesis, Eindhoven University of Technology.

Van der Meulen, S. (2005a). *Machine-In-the-Loop Control Optimization: A Literature Survey.* Literature survey, Eindhoven University of Technology.

Van der Meulen, S. (2005b). *Machine-In-the-Loop Control Optimization: Application to High-Precision Motion Systems.* MSc thesis, Eindhoven University of Technology.

Van der Meulen, S., Tousain, R., and Bosgra, O. (2008). Fixed structure feedforward controller design exploiting iterative trials: Application to a wafer stage and a desktop printer. *Journal of Dynamic Systems, Measurement, and Control*, 130(5):0510061–05100616.

Van Donkelaar, E. (2003). Some properties of the setpoint feedforward pir. Technical Report 50669/00, ASML.

Van Herpen, R. (2009a). Experimental modeling and validation for robust motion control of next-generation wafer stages. MSc thesis (paper), Eindhoven University of Technology.

Van Herpen, R. (2009b). Experimental modeling and validation for robust motion control of next-generation wafer stages. MSc thesis (report), Eindhoven University of Technology.

Van Herpen, R. (2010). Experimental evaluation of robust-control-relevance: A confrontation with a next-generation wafer stage. *Proceedings of the 2010 American Control Conference.*

Verhoeven, S. (2010). Improving inferential performance of flexible motion systems. MSc thesis, Eindhoven University of Technology.

Wortelboer, P. (1994). *Frequency-weighted Balanced Reduction of Closed-loop Mechanical Servo-systems: Theory and Tools.* PhD thesis, Delft University of Technology.

Wu, Y. and Zou, Q. (2009). Robust inversion-based 2-dof control design for output tracking: Piezoelectric-actuator example. *IEEE Transactions on Control Systems Technology*, 17(5):1069–1082.

Zhang, W., Allgower, F., and Liu, T. (2006). Controller parameterization for siso and mimo plants with time delay. *Systems & Control Letters*, 55(10):794–802.

Zou, Q. (2009). Optimal preview-based stable-inversion for output tracking of nonminimum-phase linear systems. *Automatica*, 45(1):230–237.

Zou, Q. and Devasia, S. (1999). Preview-based stable-inversion for output tracking of linear systems. *Journal of Dynamic Systems, Measurement, and Control*, 121(4):625–630.