Modelling and analysis of digital control systems

Citation for published version (APA):

Document status and date:
Published: 01/01/2003

Publisher Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.

Download date: 25. Jul. 2020
Modelling and analysis of digital control systems

Sander Noijen
DCT 2003.83

Supervisors:

Ir. J. Starreveld, Mecal Applied Mechanics BV
Prof. dr. H. Nijmeijer
Dr. Ir. P.F. Lambrechts

9th September 2003
Abstract

Control systems are mostly modelled in continuous time. Implementation however happens in general by digital computers (mostly microprocessors), which influences the behavior of the control system. This makes a well-defined method for modelling of digital control systems desirable. In the past Mecal Applied Mechanics BV simulated digital control systems as an analog system with a time delay in its feedback loop. This report investigates whether this method sufficiently reflects digital control properties or another method should be used. For this the Mecal-method is compared with several other methods in time and frequency domain.

The sampled-data method, that uses a model with a controller in discrete time and system in continuous time, gives the best representation of the properties of digital control systems in time domain. Fourier analysis shows that both the sampled-data method and the adjusted Mecal-method, that is the Mecal-method with time delay in the control loop instead of the feedback loop, can be used to give a good representation of the properties of digital control.

Time simulations of a sampled-data model show that A/D-conversion of the feedback output signal has little effect on the digital response. It shows that the extra delay in the controller’s hardware may destabilize a control system. And it also shows that the response at the Nyquist frequency is equivalent to the step response.

Frequency domain analysis show that the analog method can be used up to 5% of the sampling frequency and that sampling induces a frequency dependent phase lag with respect to the analog phase response. Furthermore it is shown that digital control systems are stable at a sampling frequency above about five times the bandwidth depending on the phase margin and that the damping factor decreases as the sampling frequency decreases.
# Table of Contents

1 Introduction 2

2 Preliminaries 3
   2.1 Stability of systems 3
   2.2 Implementation aspects 4
   2.3 Sampling 5

3 A frequency spectrum analysis method 8
   3.1 A brief discussion of the Fourier theory 8
   3.2 Implementation of the Fourier routine 10

4 Modelling the digital system 11
   4.1 Analog method 11
   4.2 Digitized method 12
   4.3 Sampled-data method 12
   4.4 Mecal-method 13
   4.5 Adjusted Mecal-method 13
   4.6 Extra properties 14
     4.6.1 Controller delay 14
     4.6.2 A/D-conversion 14

5 Simulations 16
   5.1 Comparison of the models 16
     5.1.1 Simulations of time domain models 16
     5.1.2 Simulations of frequency domain models 18
     5.1.3 Results 21
   5.2 Properties of digital control 22
     5.2.1 Simulations in time domain 22
     5.2.2 Simulations in frequency domain 25
   5.3 Implementation of digital control in the Mecal-tools 26

6 Multi degree of freedom (mdof) systems 28
   6.1 Mdof-control 28
   6.2 Digitization of mdof-system 30

7 Conclusion and recommendations 31

A Frequency Responses 33
   A.1 Open loop bode plots of system 2 33
   A.2 Nyquist plots 34

B Simulink-models 35
   B.1 Sdof-Simulink-files 35
     B.1.1 Dofourier.mdl 35
     B.1.2 Fouriercheck.mdl 35
     B.1.3 Timesystem.mdl 36
   B.2 Mdof-Simulink-files 37
     B.2.1 Analog Simulink model of example 37
     B.2.2 Digital Simulink model of example 37
Chapter 1

Introduction

This is a report of my external traineeship at Mecal Applied Mechanics BV in Veldhoven. Mecal is specialized in mechanical, dynamical and thermal analysis with the finite element method (FEM) and contributes in this way to the development of technological products. The areas of work are wind turbines, precision machines, machine support frames, applied mechanics and performing turn key projects within these areas.

My traineeship took place within the precision machines group. This group performs mostly dynamical and thermal FEM-analysis with ASM Lithography as their main customer. One of the problems faced with during dynamical analysis is the difference between the theoretical control and the real implemented control of the machines. The main difference is that in theory the whole control system is modelled in continuous time, while in reality most control systems are implemented by a digital computer resulting in a discrete time controller and a continuous time plant.

The purpose of this traineeship is to find a method to model and analyse digital control systems with as well analog as digital behavior as implemented in real life and to compare this method with the method currently used at Mecal. To do this first some aspects of the theoretical background of dynamics and control like discretization, stability and Fourier analysis are handled with. Then several methods for modelling a digital control system, one of which is the method currently used at Mecal, are introduced. These methods are evaluated in the time and frequency domain. The best method is chosen and two simple single-degree-of-freedom-systems are analysed. Most systems however have more than one degree of freedom. In some cases these multi-degree-of-freedom-systems can be controlled by decoupling the input/output relations as done at Mecal. In this report also a begin is made for the modelling method of these mdof-systems.

This report is organized as follows: first some aspects of fundamental control theory and sampling are discussed. In chapter 3 Fourier analysis needed to evaluate the modelling methods is discussed. In chapter 4 several methods are introduced in the time and the frequency domain. In chapter 5 the methods are compared by means of Fourier analysis, and properties of the final digital control model are analysed in the time and the frequency domain. Finally the digital control model is implemented in the mecal-tools. In chapter 6 the analysis is expanded to a specific multi-degree-of-freedom case. Chapter 7 concludes this report.
Chapter 2

Preliminaries

Below some control concepts are briefly recalled and some sampling aspects are introduced.

2.1 Stability of systems

We introduce the control system of figure 2.1. If we define \( C \) to be the transfer function of the controller and \( H \) of the plant the resulting open and closed loop transfer functions are given by:

\[
H_{cl} = C H \\
H_{ol} = \frac{C H}{1 + C H}.
\] (2.1)

In general a system is said to be stable if the output and all of the internal variables never become unbounded and go to zero as time goes to infinity (see [4, pp. 214]), which will happen if all the poles of the system are in the open left half plane. This means for the Nyquist plot\(^1\) that the line of increasing frequency must pass the point \((-1,0)\) at the right hand side for the (minimum phase) system to be stable.

Measures for the robustness of stability of a system are defined by:

- The gain margin (GM): The GM is the factor by which the gain is less than the neutral stability value (0 dB), measured by the vertical distance between the open loop gain and the zero dB gain at the frequency where the phase is \(-180^\circ\).
- The phase margin (PM): The PM is the amount by which the phase of the open loop system is less than \(-180^\circ\) at the frequency where the open loop gain is zero dB.

One is referred for further readings to [4].

\(^1\)real against imaginary part of the open loop frequency response
Control systems are mostly analysed in continuous time. This means controller and plant are being simulated as if their in- and output are continuous functions of time (see figure 2.2). In reality, however, most control systems use digital computers to implement controllers. The main difference between digital and analog systems is that the digital controller operates on samples of the plant output, rather than on the continuous time output signal. This is done in reality by sampling the analog plant output signal, mostly a voltage, and converting it with an A/D-converter into a binary number. Consequently, the output of the digital controller is defined every instant period of time (sample period). By inserting a D/A-converter between the controller and the plant, the output is changed from a binary number to an analog voltage again and a zero order hold.
maintains that voltage throughout the sample period. The foregoing can be modelled as in figure 2.3 and is called a sampled-data control system. In the past Mecal modelled the digital controller as a time delay of the sample time $T_s$ divided by two in the feedback loop, followed by the analog controller (see figure 2.4). This is done because the minimum delay and the maximum delay of

![Figure 2.5: Sample delay of a sinusoid-signal due to an A/D-converter](image)

the A/D-converter are 0 and $T_s$ respectively, resulting in a mean delay of half the sample time, as can be seen in figure 2.5.

### 2.3 Sampling

![Figure 2.6: Sample and hold of a random signal](image)

The sampled error signal (see figure 2.6) of a sampled-data control system can be expressed as in [3, pp. 91].

$$E(t) = E(0)[u(t) - u(t - T_s)] + E(T_s)[u(t - T_s) - u(t - 2T_s)] + E(2T_s)[u(t - 2T_s) - u(t - 3T_s)] + \ldots$$
where \( u(t) \) is the unit-step function. This causes the input of the controller to be a sequence of step functions, which means that the plant may be affected by frequencies exceeding the sampling frequency. This effect however is not accounted for in this report and may be subject for further research.

The Laplace transform of time signal \( E(t) \) is given by

\[
E(s) = E(0) \left[ \frac{1}{s} - \frac{e^{-T_s s}}{s} \right] + E(T_s) \left[ \frac{e^{-T_s s}}{s} - \frac{e^{-2T_s s}}{s} \right] + E(2T_s) \left[ \frac{e^{-2T_s s}}{s} - \frac{e^{-3T_s s}}{s} \right] + ...
\]

and can be written as

\[
E(s) = \sum_{n=0}^{\infty} E(nT_s)e^{-nT_s s} \left[ \frac{1 - e^{-T_s s}}{s} \right].
\] (2.2)

Since the first factor is a function of the input \( E(t) \) and the sampling period \( T_s \) and the second factor is independent of \( E(t) \), the last factor can be seen as the transfer function for the sampler and hold system. However for small \( T_s \) it holds that the first term is equal to \( 1/T_s \), which means (see [3, pp. 457]) that for frequency response comparison purposes the gain of the sampler and hold should be multiplied by \( 1/T_s \). Thus we can conclude the transfer function for a sampler with zero-order hold to be:

\[
Q(s) = \frac{1}{T_s} \left( 1 - \frac{e^{-T_s s}}{s} \right).
\] (2.3)

Because the Laplace transform is a bit unwieldy in discrete time signals and systems, as can be seen from (2.2) and (2.3), it is useful to introduce the z-transform.

Whereas the Laplace transform for signals is defined by

\[
\mathcal{L} \{ f(t) \} = F(s) = \int_0^{\infty} f(t)e^{-st}dt,
\] (2.4)

the z-transform is defined by

\[
\mathcal{Z} \{ f(k) \} = F(z) = T_s \sum_{k=0}^{\infty} f(k)z^{-k},
\] (2.5)

where \( f(k) \) is \( f(t) \) at time \( t = kT_s \), with \( T_s \) still the sample time. If we compare the Laplace transform and the z-transform of continuous time signals, as done for instance in [5], the relationship between these two transforms appears to be

\[
z = e^{sT_s}.
\] (2.6)

Using this relationship a transformation table can be created for signals and systems as for the Laplace transform. If we use this z-transform as a transfer function for the sample and hold function we obtain

\[
\mathcal{Z} \{ Q(s) \} = \frac{1}{T_s} \left( 1 - z^{-1} \right) \mathcal{Z} \{ \frac{1}{s} \}.
\]

This transfer function can be used to convert a continuous time system, in our case a controller, into a discrete time system with a sampler and zero-order hold at the input by:

\[
\mathcal{Z} \{ C(s) \} = \frac{1}{T_s} \left( 1 - z^{-1} \right) \mathcal{Z} \{ \frac{C(s)}{s} \}.
\] (2.7)
In Matlab this can be done by using the commands c2d.m or c2dm.m and using the default method zoh (zero-order hold).

A continuous frequency response cannot be calculated from the z-transform, but we can determine the frequency response at several frequencies by substitution of $z = e^{j\omega T_s}$. The complex transfer function of the plant $H = H(s)$ can be written as $H = X_p + Y_p j$ and of the controller $C = C(z) = C(e^s)$ as $C = e^{x_c + y_c j}$. Then magnitude and phase are determined by the absolute value and angle with the real axis respectively of the complex transfer function. So the magnitude ($M_p$) and the phase ($P_p$) of the plant can be written as:

$$
M_p(s) = \sqrt{X_p(s)^2 + Y_p(s)^2}
$$

$$
P_p(s) = \arctan \frac{Y_p(s)}{X_p(s)}
$$

(2.8)

and, recalling $e^{x+yj} = e^x (\cos y + j \sin y)$, the magnitude ($M_c$) and phase ($P_c$) of the controller are:

$$
M_c(s) = e^{x_c(s)}
$$

$$
P_c(s) = Y_c(s).
$$

(2.9)

If the magnitude and phase are calculated for many frequencies, a line can be drawn through the corresponding points resulting in a kind of frequency response. This frequency response is only valid up to $f = \frac{1}{2T_s}$ due to the aliasing effect as described in chapter 3.
Chapter 3

A frequency spectrum analysis method

In this chapter a routine will be derived to find the frequency response of a system through a simulated experiment.

3.1 A brief discussion of the Fourier theory

In chapter 4 a range of methods is described to derive frequency response plots of the digital control system. To check which model is most realistic, Fourier analysis will be used.

The Fourier transform of a continuous function $x(t)$ is its frequency spectrum and is defined by

$$X(f) = \mathcal{F}[x(t)] = \int_{-\infty}^{\infty} x(t)e^{-2\pi j ft}dt$$  \hspace{1cm} (3.1)

We assume however that the only information of the time signal we have is not in the interval $-\infty < t < \infty$ but in the interval $0 \leq t \leq T$ and the signal is digitized using equidistant time intervals. This is done by sampling the time signal with fixed time steps $\Delta T$ over a total measuring time $T$ as shown in figure 3.1. As can be seen the number of intervals $N$ is equal to $\frac{T}{\Delta T}$. Therefore we approximate (3.1) by the commonly used Discrete Fourier Transformation:

![Figure 3.1: Sampling of the signal $x(t)$](image-url)

```plaintext

**Figure 3.1:** Sampling of the signal $x(t)$
```
Because the signal has finite length $T$, the lowest frequency, called the fundamental frequency, that can be found is equal to $f_0 = \frac{1}{T}$. Therefore (3.2) will only be evaluated for the discrete frequencies $f_n = n f_0 = n/T, n = 0, 1, 2, \ldots \infty$. Consequently (3.2) can be written as

$$X[f_n] = \frac{\Delta T}{T} \sum_{k=0}^{N-1} x[k \Delta T] e^{-2\pi j n k / N}$$

(3.3)

For each integer value $r$ we see that:

$$X[f_n + rN] = \frac{1}{N} \sum_{k=0}^{N-1} x[k \Delta T] e^{-2\pi j k (n+rN)/N} = \frac{1}{N} \sum_{k=0}^{N-1} e^{-2\pi j k r} \left[ x[k \Delta T] e^{-2\pi j kn / N} \right] = X[f_n]$$

This means the DFT is periodic with frequency period $f_s = N/T$. Therefore it only makes sense to evaluate this DFT for the discrete frequencies $f_n = n f_0 = n/T, n = 0, 1, 2, \ldots N - 1$. Furthermore it holds for a real time signal that the DFT is “folded” around the frequency line $f_N = f_s/2$ (Nyquist frequency). This can be visualized by looking at figure 3.2. If for instance an 8 Hz signal is sampled with 10 Hz ($f_N = 5$), this signal will also appear as a 2 Hz signal, because this signal passes the same sample points as the 8 Hz signal. This is called aliasing and can be avoided by filtering the signal before sampling. This means only the first half of the column $X = [X[f_0], X[f_1], \ldots, X[f_{N-2}], X[f_{N-1}]]^T$ of DFT values contains essential information.
3.2 Implementation of the Fourier routine

Knowing this we are able to implement a Fourier routine and compare the outcomes of this Fourier analysis with the earlier defined methods. The following has been implemented as a sequence for many frequencies, in the Matlab-file Fourier.m and the Simulink-files of Appendix B.1.1 and B.1.2:

1. Determine the frequencies to be evaluated.
2. Choose $\Delta T < T_s/20$ to make sure $T_s$ has little influence on the DFT process.
3. Choose $f_{\text{sim}} = n f_0$ with $n \in \mathbb{N}$.
5. Evaluate input frequency using Simulink-system
6. Compute DFT (Discrete Fourier Transform) of input and output
7. Compute amplitude and phase by division and subtraction respectively of the DFT of the output over the input.

Example:
To check whether the routine works properly a simulation is performed on a system in the Laplace domain and the results of this simulation are compared with the real frequency response. The system used is the second order system: $\frac{1}{s^2+s+1}$. From figure 3.3 it can be concluded that the implemented Fourier routine results in exactly the same frequency response as the theoretical response. So the routine works well for this system and probably also for other systems, because Fourier analysis in general is very reliable.

![Figure 3.3: Fourier analysis performs well](image-url)
Chapter 4

Modelling the digital system

There are several methods to model digital systems as discussed in chapter 2. In this chapter five different methods will be introduced, both in the time domain and in the frequency domain.

4.1 Analog method

Differences between digital and analog control will be found by comparison of the analog model with the digital one. The analog method is a continuous time method, this means the controller and the plant receive and handle information continuously. Consequently, the transfer functions of controller and plant are modelled in the Laplace domain. This results in the following open and closed loop transfer functions:

\[
H_D(s) = \frac{C(s)H(s)}{1+C(s)H(s)}
\]

In simulations this can be done by inserting one continuous transfer function block containing \( H_D(s) \) or \( H_{cl}(s) \) or with one transfer function block for the controller and one for the plant (see figure 4.1).

\[\text{Figure 4.1: Analog simulation model}\]
4.2 Digitized method

A first estimate of the sampled-data control system can be made by digitizing the whole system. This is done as in (2.7) with $C(s)$ not only the controller but the entire system, in other words:

$$H_{cl}(z) = \frac{1}{T_s}(1 - z^{-1})Z\{C(s)H(s)\}$$

$$H_{cl}(z) = \frac{1}{T_s}(1 - z^{-1})Z\{\frac{C(s)H(s)}{s+K(s)}\}$$  \hspace{1cm} (4.2)

In simulations this can be done by inserting one discrete transfer function block containing $H_{cl}(z)$ or $H_{cl}(s)$ (see figure 4.2).

![Figure 4.2: Digital simulation model](image)

4.3 Sampled-data method

The sampled-data method should represent an exact copy of figure 2.3. This means the controller is simulated in the discrete time domain, and the system is simulated in the continuous time domain. The major problem is that these two cannot directly be combined to form one frequency response. This difficulty can be avoided by calculating the complex transfer functions from the magnitude and phase for several frequencies of the digitized controller and continuous plant as described in section 2.3.

From (2.8) it can be concluded that:

$$X_p(s) = \sqrt{\frac{M_p(s)^2}{1 + \tan^2(Y_p(s))}}$$

$$Y_p(s) = \sqrt{M_p(s)^2 - X_p(s)^2}$$  \hspace{1cm} (4.3)

and from (2.9)

$$X_c(s) = \ln M_c(s)$$

$$Y_c(s) = P_c(s)$$  \hspace{1cm} (4.4)

by which $H(s) = X_p(s) + Y_p(s)j$ and $C(s) = e^{X_c(s)+Y_c(s)j}$ can be found. In Matlab this is done by transjm and dtransjm respectively. The open and closed loop transfer functions then become:

$$H_{ol}(s) = e^{(X_c(s)+Y_c(s)j)}(X_p(s) + Y_p(s)j)$$

$$H_{ol}(s) = e^{(X_c(s)+Y_c(s)j)}(X_p(s) + Y_p(s)j)$$  \hspace{1cm} (4.5)

The simulation of this method can be done in two ways. Firstly the controller can be simulated by a discrete transfer function block and the plant by a continuous transfer function block. Also the controller and plant can be simulated in continuous time with a preceding zero-order hold block as sampler (see figure 4.3).
4.4 Mecal-method

The Mecal-method is the method currently used by Mecal as described in section 2.2. The only difference with the analog method is the time delay in the feedback loop. This time delay in the frequency domain is simulated by a tenth order-Padé-approximation. If we call $P(s)$ the transfer function of the Padé-approximation, Mecal assumes the resulting open loop transfer function to be equal to the analog method and the closed loop transfer function as

$$H_{cl}(s) = \frac{C(s)H(s)}{1 + C(s)H(s)P(s)}. \quad (4.6)$$

The simulation can be done by implementing a time delay of $Ts/2$ in the feedback loop (see figure 4.4).

4.5 Adjusted Mecal-method

The time delay in the Mecal-method is implemented in the feedback loop. This causes that Mecal did not model the time delay at all in open loop. The effect of the delay however affects the phase of the forward loop (it is caused by the measurement of the states at the end of this line) and should be modelled there. The total open and closed loop transfer functions then become

$$H_{cl}(s) = C(s)P(s)H(s)$$
$$H_{cl}(s) = \frac{C(s)P(s)H(s)}{1+C(s)H(s)P(s)}. \quad (4.7)$$
Logically the time delay in the time domain should be placed before the controller (see figure 4.5).

4.6 Extra properties

4.6.1 Controller delay

Implementation of digital controller units, consisting of a measurement-sensor, the D/A- and A/D-converter, and the discrete time controller, is done via electrical circuits. This means measurement-time of the sensor, the conversion-time of the converter and the calculation time of the controller can never be zero, and this consequently introduces a delay in the system. Because the delay of the controller is the largest we will introduce $T_c$ as the additional delay in the system. In figure 4.6 the total delay of the control unit is plotted. A measurement of the output is made at time $t_0$, due to the extra delays the corresponding output of the controller is sent at $t = t_0 + T_c$. The next sample is taken at $t = T_s T_c$, etc., from this it follows that the total mean delay is $T_s/2 + T_c$. However in general $T_c$ is much smaller than the sample time. In subsection 5.2.1 it will be discussed how much faster the controller has to be than the A/D-converter to justify neglecting $T_c$. This extra delay can be seen in the Laplace domain as $e^{-T_c s}$ (see [3, pp. 144]) and can be inserted by adding the option 'outputdelay' to the function by which the controller transfer function is created.

4.6.2 A/D-conversion

In chapter 2 the A/D-converter is referred to as a device that converts a voltage into a binary number. This happens by division of the dynamic range of the analog signal into discrete levels, of which to each discrete level a digital number is assigned. For instance, if we have a signal with a dynamic range of $[0 : 1]$ Volts and a 3-bits converter, then the dynamic range is divided into $2^3 = 8$ equidistant quantization levels of $\frac{1}{2^3} = 0.125$ volts. When the analog voltage is for instance 0.17 the voltage is in the second level and receives the corresponding digital number 001. This conversion has negligible influence on the frequency response of a system (as will be shown
in subsection 5.2.1) and will be simulated in the time domain as a saturation block followed by a quantizer with quantization interval \( R_{\text{max}} - R_{\text{min}} \) with \( R_{\text{max}} \) and \( R_{\text{min}} \) as the maximum and minimum value of the dynamic range and bits the number of bits of the system.
Chapter 5

Simulations

The methods described in chapter 4 are simulated in this chapter and compared with the results of the Fourier analysis of chapter 3. From these simulations it is concluded which method can be used for what purposes. Thereafter, the method that fulfills all purposes is used to perform simulations to inspect the influence of digital control on the time- and frequency-response of a dynamical system. All the simulations are done on two different systems, namely system 1 with a

<table>
<thead>
<tr>
<th></th>
<th>System 1</th>
<th>System 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controller</td>
<td>( \frac{2\pi^2}{s} )</td>
<td>( \frac{10}{s^2+s} )</td>
</tr>
<tr>
<td>Plant</td>
<td>( \frac{1}{s+5} )</td>
<td>( \frac{1}{(s+2)(s+5)} )</td>
</tr>
<tr>
<td>( \omega_n ) [rad/s]</td>
<td>( 2\pi )</td>
<td>( \frac{0.77}{2\pi} )</td>
</tr>
</tbody>
</table>

first order controller and first order plant and system 2 with a second order controller and second order plant. The specifications of these systems are given in table 5.1.

5.1 Comparison of the models

The models are compared by simulation of all methods in the time- and frequency-domain. In the time domain the performance of the methods is compared with the sampled-data method of section 4.3, since this method is the realistic modelling method of a digital controller (see for instance [8, pp. 908]). The frequency responses of the methods are compared with the Fourier analysis of the sampled-data model simulation.

5.1.1 Simulations of time domain models

The simulations are performed on the closed loop control systems with the Matlabfile timesim.m and the Simulink-file of Appendix B.1.3. Timesim.m compares the different
Figure 5.1: Simulations of system 1 with $T_s = 0.05$ (up) and system 2 with $T_s = 0.4$ (down)
methods with the sampled-data method by the root mean squared error defined as

$$RMSE = \sqrt{\frac{1}{T} \int_0^T \left| x_1(s) - x_2(s) \right|^2 ds},$$

(5.1)

with \( T \) the total simulation time, \( x_1(s) \) the sampled-data response and \( x_2(s) \) the compared method response, both as a function of time \( s \). The results are shown in table 5.2. The error percentage (E%) of the last two columns can be found by comparing the root mean squared error with the root mean squared response of the sampled-data method \( \times 100\% \). From table 5.2 and figure 5.1 it follows that all the models, especially the digital and adjusted Mecal model, follow the trend of the sampled-data method, but in detail (E% is never less than 30%) they differ too much from the sampled-data model to form an acceptable alternative.

### 5.1.2 Simulations of frequency domain models

The frequency domain simulations have been performed using the Matlabfile freqsim.m and Simulinkfile dofourier.mdl (Appendix B.1.1) for frequencies \( \left[ \frac{f_n}{10} : 10f_n \right] \). The mean squared error of the models is calculated as in (5.1) with \( T \) the frequency range, \( x_1(s) \) the logarithmic Fourier response and \( x_2(s) \) the logarithmic compared method response, both as a function of logarithmic frequency \( s \). From figure 5.2 it is clear that for low frequencies all the models behave the same as the Fourier response. From figure 5.2 and table 5.3 the following can be said about the methods for higher frequency response:

<table>
<thead>
<tr>
<th>Model</th>
<th>RMSE System 1 ( \cdot 10^{-2} )</th>
<th>RMSE System 2 ( \cdot 10^{-1} )</th>
<th>E% System 1</th>
<th>E% System 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analog</td>
<td>6.7</td>
<td>2.9</td>
<td>65.6</td>
<td>76.4</td>
</tr>
<tr>
<td>Digital</td>
<td>3.2</td>
<td>2.0</td>
<td>31.4</td>
<td>31.5</td>
</tr>
<tr>
<td>Mecal</td>
<td>6.3</td>
<td>2.3</td>
<td>62.3</td>
<td>61.5</td>
</tr>
<tr>
<td>Adjusted Mecal</td>
<td>3.6</td>
<td>1.7</td>
<td>35.3</td>
<td>44.0</td>
</tr>
</tbody>
</table>

Table 5.2: Numerical comparison of the models

<table>
<thead>
<tr>
<th>Model</th>
<th>RMSE Magnitude System 1</th>
<th>RMSE Magnitude System 2</th>
<th>RMSE Phase System 1 °</th>
<th>RMSE Phase System 2 °</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analog</td>
<td>0.44</td>
<td>1.50</td>
<td>14.1</td>
<td>40.4</td>
</tr>
<tr>
<td>Digital</td>
<td>1.56</td>
<td>1.20</td>
<td>6.2</td>
<td>10.3</td>
</tr>
<tr>
<td>Sampled Data</td>
<td>0.64</td>
<td>3.78</td>
<td>0.2</td>
<td>17.1</td>
</tr>
<tr>
<td>Sampled Data</td>
<td>0.41</td>
<td>1.48</td>
<td>0.2</td>
<td>1.8</td>
</tr>
<tr>
<td>Mecal</td>
<td>0.35</td>
<td>1.18</td>
<td>13.5</td>
<td>39.1</td>
</tr>
<tr>
<td>Adjusted Mecal</td>
<td>0.35</td>
<td>1.18</td>
<td>0.2</td>
<td>1.7</td>
</tr>
</tbody>
</table>

Table 5.3: Numerical comparison of the models
Figure 5.2: Frequency response simulations of system 1 with $T_s = 0.05$ (up) and system 2 with $T_s = 0.4$ (down)
Analog The magnitude is almost the same, up until the eigenfrequency, where the magnitude of the analog model is significantly lower than the Fourier response. The analog phase leads more and more as the frequency increases.

Digitized This method compares well both for the magnitude and for the phase. Around the natural frequency however, there are substantial differences. For higher frequencies the phase of system 1 approaches the phase of the analog method.

Sampled-data There is a good correspondence at all frequencies, however the high frequency amplitude differs a little and for system 2 the high frequency phase is leading with respect to the Fourier analysis.

Mecal The amplitude is almost exactly the same as with the Fourier analysis. However the phase is similar to the results of the analog method.

Adjusted Mecal There is a good correspondence at all frequencies resulting in the lowest RMSE.

Figure 5.3: Fourier sampling at half Nyquist frequency

Furthermore, it must be said that the Fourier results around \( f = f_N \) depend on the input, even if the absolute magnitude of the sampled signal is constant. At the Nyquist frequency the input may be sampled as it crosses zero. This means the magnitude is zero and the phase is undetermined. If the input sinusoid has a phase shift, the magnitude and phase of the resulting signal changes with respect to the magnitude and phase of the input (see figure 5.3). As a consequence from this nothing can be said about the response of the systems on \( f_N \) on basis of the Fourier analysis.

However it is very plausible that the phase at the Nyquist frequency runs 90° behind. The A/D-converter samples twice every period, with an average delay of \( T_s/2 \), which is \( \frac{1}{2} \) of the period time of the input and thus a \(-90°\) phase shift with respect to the analog response.

Figure 5.2 shows that the adjusted Mecal-method performs best on the two examples. The sampled-data method is nevertheless also a realistic method. This is shown if the control system is digitized in Laplace transform using (2.3). The results are shown in figure 5.4 and table 5.3 (sampled data 2), it is clear that the RMSE is very low and that the error around the Nyquist frequency for system 2 is gone. This can be explained by the fact that the sampled-data method
Figure 5.4: Sampled-data model works best in the Laplace domain

consists of a discretized controller, which is translated back into the Laplace domain by $z = e^{STs}$, in order to create a frequency response. Since the $z$-transform is only defined at discrete points $kT_s$ with $k \in \mathbb{N}$, information between these points is lost and not known by the Laplace transform, resulting in the deviation around the Nyquist frequency.

5.1.3 Results

From the two foregoing subsections it can be concluded that the sampled-data method is the method with the best performance to be used in the time domain.

For frequency simulations more methods can be used. If we are interested in the whole frequency range up to $f = f_N$, the sampled-data method can be used in the Laplace domain. It is however advisable to use the Adjusted Mecal-method since this method is very easy to implement and very accurate.

In practice the interesting area lies in the low to middle frequency region (up to the bandwidth / eigenfrequency). So for this purpose also the sampled-data method with controller in the $z$-domain can be used. If we are only interested in low frequencies (below $f = fs/20$) all the methods perform well, since there the digital system is identical to the analog system. This subsection is summarized in table 5.4.

<table>
<thead>
<tr>
<th>Low frequencies</th>
<th>Simulation in time domain</th>
<th>Simulation in frequency domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low and middle frequencies</td>
<td>Sampled-data method</td>
<td>Sampled-data method</td>
</tr>
<tr>
<td>Up to $f_N/2$</td>
<td>Sampled-data method</td>
<td>All methods</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sampled-data method</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Adjusted Mecal-method</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sampled-data method 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Adjusted Mecal-method</td>
</tr>
</tbody>
</table>

Table 5.4: When to use which method?
5.2 Properties of digital control

Next the digital control system is compared with its analog counterpart as to find out what the differences are between them. In accordance with the previous section we use the adjusted Mecalmethod for frequency simulations and the sampled-data method for time simulations, both performed with propsim.m.

5.2.1 Simulations in time domain

Three aspects are investigated in this subsection: the influence of controller delay (as discussed in subsection 4.6.1 and the influence of A/D-conversion (as discussed in subsection 4.6.2).

The influence of controller delay

Figure 5.5: Time responses of system 1 with $T_s = 0.05$ (l) and system 2 with $T_s = 0.42$ (r) with additional controller delay $T_c = T_s/2$ and $f_{\text{sim}} = \frac{1}{20}f_s$.

The time responses of both systems (Figure 5.5) shows that the additional delay $T_c$ introduces an extra delay of $T_c$ in the response of the systems and that this extra delay causes a larger amplitude in the closed loop system. Extra controller delay may even cause instability since the phase lags with respect to the system without controller delay, resulting in a decrease of the gain and phase margin as can be seen in figure 5.6. This decrease of the phase margin also exhibits a decrease in the damping factor (see subsection 5.2.2), which explains the larger amplitude of the time simulation performed at the bandwidth of the systems.

In general it is recommended to have $T_c < T_s/10$ (see for instance [4, p.621]) to prevent the controller delay of having significantly influence on the response of the system. As it can be seen in figure 5.7 the amplitude of the response and the delay then indeed become smaller. It depends however on the designer’s purpose what an acceptable criterion is for the controller delay.
Figure 5.6: Open loop responses of system 1 (up) with $T_s = 0.05$ and system 2 (down) with $T_s = 0.42$, additional controller delay $T_c = T_s/2$ and $f_{sim} = \frac{1}{20}f_s$. 23
The influence of A/D-conversion

In order to reflect the influence of A/D-conversion clearly, low-performance A/D-converters are inserted into the systems (see figure 2.3). The specifications of the converters is not interesting if the quantization interval is known. In the simulations a quantization interval of one is used, this means that the output feedback is rounded to the nearest integer. The consequences of this are visualized in figure 5.8: it is clear that even very low performance A/D-conversions have little influence on the system. Only when the dynamic range of the system is greater than the range of the A/D-converter, the feedback output is bounded and the influence of the conversion is noticable. Note however that this is true for simple systems with a sine as input.

Figure 5.7: Responses of system 1 with $T_s = 0.05$ (l) and system 2 with $T_s = 0.42$ (r) with controller delay $T_c = T_s/10$ and $f_{sim} = \frac{1}{20} f_s$.

Figure 5.8: Responses of analog system 1 (l) and system 2 (r) with quantization interval of one.
5.2.2 Simulations in frequency domain

Simulations in frequency domain can result in different plots. First the stability of the systems is discussed using open loop frequency responses and Nyquist plots. Also the closed loop responses are discussed.

Intuitively, one expects the frequency response of digital systems for low frequencies to be the same as for analog systems. In figure 5.9 the open loop frequency responses for increasing sampling frequency of system 1 (for system 2 see Appendix A.1) are plotted. It can be seen that the analog magnitude can be used until \( f_N/2 \) since it resembles the digital one. The analog phase response can be used until \( f = 2f_N \); above this frequency the phase responses differ significantly. In general stable analog systems become unstable, as a consequence from frequency dependent phase difference, if sampled by a frequency slower than approximate \( 5f_N \) (see phase and gain margins and see Appendix A.2 for Nyquist plots). The exact destabilizing sampling frequency can be calculated by reviewing subsections 4.6.1 and 5.2.1 that have shown the total
delay of the system to be approximated by $T_s/2$. This results in a frequency dependent phase lag

$$PL(f) = \frac{180f}{f_s}$$

(5.2)

by which the digital system becomes unstable if sampled by a frequency slower than

$$f_s \approx \frac{180f_n}{PM}$$

(5.3)

where $f_n$ is the natural frequency of the system. E.g. a system with a phase margin of 60° results in an unstable digital system if $f_s < 3f_n$.

Furthermore it can be said that the amount of damping decreases as the sampling frequency does, since the phase margin which forms a degree for damping, decreases. This is confirmed by the closed loop frequency responses of figure 5.10. The amount of which the digital phase lags depends on the sample time and the dynamics of the system. Consequently a rule as easy as for the open loop response cannot be derived. Also it is shown that for low sampling frequencies the closed loop dynamics become unstable.

5.3 Implementation of digital control in the Mecal-tools

Now that the properties of digital control are known, the digital models are implemented in the Mecal-tools. While the implementation and results of simulations already have been discussed, the implementation and results of digital control in the Mecal-tools are handled in a separate report [9].
Figure 5.10: Frequency responses of closed loop systems 1 (up) and 2 (down) with increasing sampling frequency.
Chapter 6

Multi degree of freedom (mdof) systems

Until now only single degree of freedom (sdof) systems have been dealt with. In reality however most systems consist of more than one degree of freedom. The digitization of these systems works parallel to the sdof case, the results may differ however. In this chapter one sort of mdof-control is explained and digitized, thereafter the properties are discussed.

6.1 Mdof-control

A mdof-system is a system with more than one input and output. Often a single input influences more than one output. This can be explained by the example of figure 6.1.

A mass \( m \) without friction has three degrees of freedom: horizontal \((x)\) displacement, vertical \((y)\) displacement and rotation \((\Theta)\) around its axis. The mass is forced in x-direction by \( F_{x1} \) and \( F_{x2} \) and in y-direction by \( F_y \). As can be seen the input \( F_{x1} \) influences outputs \( x, y \) and \( \Theta \). This means there exist non-zero transfer functions from every input to every output. In this case this
results in nine transfer functions as depicted in figure 6.2. It is not hard to imagine that it is a difficult task to find some controller that stabilizes this system.

At Mecal this is done by decoupling the system, which means that every single input uniquely influences one output. The first step is to relate the forces acting on the system to the center of gravity (cog), this is called gain balancing (GB). In the example this means that the three inputs are translated to three inputs $F_{x\text{cog}}$, $F_{y\text{cog}}$ and $T_{Rz\text{cog}}$ independently acting on the three outputs at the center of gravity by

$$
\begin{bmatrix}
F_{x\text{cog}} \\
F_{y\text{cog}} \\
T_{Rz\text{cog}}
\end{bmatrix} =
\begin{bmatrix}
1 & 1 & 0 \\
0 & 0 & 1 \\
-\frac{b}{2} & \frac{b}{2} & 0
\end{bmatrix}
\begin{bmatrix}
F_{x1} \\
F_{x2} \\
F_{y}
\end{bmatrix}
$$

resulting in three independent transfer functions.

The inputs of these transfer functions control the center of gravity, while in reality the states will probably be measured somewhere else. To compensate for this, another matrix is defined, which at Mecal is referred to as the gain scheduling (GS) matrix (for derivation see [10]). For simplicity in this example $x, y$ and $\Theta$ are measured at the center of gravity so GS is defined by the identity matrix:

$$
\begin{bmatrix}
x_{\text{cog}} \\
y_{\text{cog}} \\
\Theta_{\text{cog}}
\end{bmatrix} =
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
\Theta
\end{bmatrix}
$$

For the resulting mdof closed loop system (see Appendix B.2.1) a stabilizing controller can be found for the entire system by finding a stabilizing controller for every decoupled input-output pair. A lead-filter is chosen as stabilizing controller for every input-output pair $118.4\frac{s+\frac{3}{2}f_n}{s+\frac{1}{2}f_n}$ for the first two and $0.8\frac{s+\frac{3}{2}f_n}{s+6f_n}$ for the last, resulting in the open loop frequency responses of figure 6.3.
Figure 6.3: Open loop magnitude (up) and phase (down) frequency response of example system

6.2 Digitization of m dof-system

The implementation of controllers in m dof-systems is done in the same manner as for s dof-systems. So in the time domain simulation every controller is implemented using the sampled-data method (see Appendix B.2.2). In the frequency domain the adjusted Mecal-method can be implemented for each controller (see m dof.m).
Chapter 7

Conclusion and recommendations

In this research five digital control modelling methods are compared for use at the precision machines group at Mecal. These methods are compared by simulations, stability analysis and Fourier techniques. The sampled-data method appears to be preferable for simulation in the time domain and both the sampled-data method and the adjusted Mecal-method can be used for simulations in the frequency domain. Because of its easy implementation the adjusted Mecal-method is preferred. The original Mecal-method poorly resembles the digital control responses.

Furthermore time domain simulations with the model using the sampled-data method have shown that:

- The quantization effect of A/D-conversion of the feedback output signal has little effect on the digital response of simple systems during sine-input-simulations.
- Extra delay in the controller's hardware may destabilize a control system.

Frequency domain simulations with the model using the Adjusted Mecal-method have shown that:

- The analog response resembles the digital response if \( f \leq \frac{f_s}{20} \).
- Sampling induces a frequency dependent phase lag with respect to the analog phase response of \( PL(f) = \frac{180f}{f_s} \).
- Digital control systems are stable at a sampling frequency of \( f_s > \frac{180f_n}{PM} \) with \( f_n \) the natural frequency.
- The damping factor decreases as the sampling frequency decreases.

From these findings it may be concluded that Mecal should replace the original method by the sampled-data method in the time domain and by the Adjusted Mecal-model in the frequency domain. Furthermore it must be advised to use a sampling frequency of at least twenty times the bandwidth and a controller delay that is substantial, say ten times, smaller than the sampling time.

The modelling of digital control for mdof-control systems at Mecal is performed similar to the sdof case. When dealing with mdof-systems Mecal uses decoupling to reduce the problem to a
number of sdof-systems, for which the modelling methods described in this report can be applied. Note that the validity of this approach depends heavily on the achieved decoupling performance.

Recommendations:
For future research it is advised to simulate more complex systems. One reason for this is that these systems could have high frequency dynamics above the bandwidth, that could be affected by the step inputs created by inter sample behavior.
Furthermore it is recommended for future research to find properties of digital control by simulations with another input than a sine, since this information is also given by the frequency response.
Also it is advisable to investigate further the properties of mdof-control and to analyse the effect of digitization on mdof-systems.
Appendix A

Frequency Responses

A.1 Open loop bode plots of system 2

Bode diagram open loop models, \( f_s = 2f_n \) Hz

Bode diagram open loop models, \( f_s = 5f_n \) Hz

Bode diagram open loop models, \( f_s = 10f_n \) Hz

Bode diagram open loop models, \( f_s = 20f_n \) Hz
Nyquist plots of system 1 (l) and system 2 (r) with increasing sampling frequency.
Appendix B

Simulink-models

B.1 Sdof-Simulink-files

Simulink-models used in Fourier.m.

B.1.1 Dofourier.mdl

B.1.2 Fouriercheck.mdl
B.1.3 Timesystem.mdl

Simulink-model used in timesim.m.
B.2 Mdof-Simulink-files

B.2.1 Analog Simulink model of example

B.2.2 Digital Simulink model of example
Bibliography


