Master-slave motor control

van Zijl, Aron

Award date:
1997

Eindhoven University of Technology
Department of Electrical Engineering
Group MBS, Measurement and Control Section

Master-slave motor control

Aron van Zijl

Master of Science Thesis
Carried out from January to August 1997
Commissioned by Prof.Dr.Ir. P.P.J. van den Bosch
Under supervision of Ir. W.P.M.H. Heemels
Ir. M.R. Vonder
Ir. R.J.A. Gorter

Abstract

This report was written as part of a project that was carried out to obtain the Master of Science degree from the faculty of Electrical Engineering at the Eindhoven University of technology. This project was carried out at the Measurement and Control section of the group MBS of this faculty. The company Buhrs-Zaandam was a participant in this project.
Buhrs-Zaandam makes so-called mailing-machines, these are machines that can automatically gather and package all kinds of printed material so that these packages can be put to post. These machines are driven by electrical motors. The aim of the project and the subject of this thesis is to develop a means of synchronising the different motors of these mailing-machines.
An important aspect is the cost. As several of these synchronisation means are needed in one machine, the cost should be low. Conventional synchronisation systems are too expensive. The main expenses for such a synchronisation system lie in the cost of the sensor that measures the position of a motor. These sensors are usually of high resolution, allowing typically 1000 position measurements per revolution of the motor axis.
In this thesis it is investigated, whether a control scheme can be developed wherefore a position sensor with lower resolution is sufficient. The aim is to use only a few (1-10) position measurements per revolution of the master motor.
First it is investigated, that for conventional control schemes, this resolution is not sufficient. The performance for a standard PI and a H_∞-controller are investigated.
Next two different control schemes are developed, which are especially tailored for low resolutions. These controllers rely on the fact that they work asynchronous in time instead of having a fixed sample rate. The difference between the two controllers is that one measures asynchronously in time and updates its control action with a fixed sample rate (synchronous). The other is fully asynchronous.
These two controllers are compared via simulations and measurements on a test set-up. Both controllers answer to the wanted specifications, even for very low resolutions. Typically 1 or 2 measurements per revolution of the motor axis are sufficient.
Both controllers have their own (dis)advantages. The final choice will depend on the ease of implementation of both controllers.

Acknowledgements

During the past six months I worked on this Master of Science Thesis at the Measurement and Control Group of the faculty of Electrical Engineering at the Eindhoven University of technology. During my work I was supported and guided by numerous people of which I shall name only a few.

First my thanks go out to my professional coaches: Ir. Maurice Heemels for sharing his ideas and opinions with me. Next to this, he gave me an idea of what a mathematician can contribute to the art (?) of control engineering. Ir. Robert Jan Gorter for his many practical tips and all the time he put into the project. Prof.Dr.Ir. P.P.J. van den Bosch for giving me the opportunity and for supervising the overall project. My gratitude goes out to the people of Buhrs-Zaandam bv, especially Ir. Mathijs Vonder for providing the equipment and posing the challenge.

There are a few people I would like to thank especially for their personal support. First my mother for her concern and encouraging words. Next, my father for being proud and giving me the genes.

My final thanks go out to Judith Belo for being patient every time I missed the train.

Aron van Zijl

Contents

1 CHAPTER ONE : INTRODUCTION 9

1.1 SCOPE 9
1.2 MAILING MACHINES 9
1.3 MASTER-SLAVE MOTOR SYNCHRONISATION 11
1.4 THE ASYNCHRONOUS CONTROL SCHEME 12
1.5 REFERENCES 13

2 CHAPTER TWO: PROBLEM ANALYSIS 15

2.1 PROJECT OBJECTIVES 15
2.2 TEST SET-UP 16

3 CHAPTER THREE: SYSTEM MODELLING 19

3.1 SYSTEM DESCRIPTION 19
3.2 FREQUENCY CONVERTER 19
3.3 ASYNCHRONOUS MOTOR 20
3.4 SHEET FEEDER 23
3.5 LUG CHAIN 25
3.6 ENCODER 25
3.7 MODEL VALIDATION 25
3.8 REFERENCES 27

4 CHAPTER FOUR: PID CONTROL 29

4.1 OPEN LOOP SYSTEM 29
4.2 PID CONTROLLER DESIGN 31
4.3 HIGH ENCODER RESOLUTION 34
4.4 LOW ENCODER RESOLUTION 36
4.5 ALTERNATIVE POSITION ERROR CALCULATION 40
4.6 REFERENCES 42

5 CHAPTER FIVE: H_∞-CONTROL 43

5.1 DESIGN 43
5.2 SIMULATIONS ON THE LINEAR SYSTEM 47
5.3 DESIGN FOR NON-LINEAR ACTUATOR 48
5.4 EVALUATION OF H_∞-CONTROLLER 50
5.5 REFERENCES 51

6 CHAPTER SIX: ASYNCHRONOUS CONTROL 53

6.1 SYSTEM MODELLING 53
6.2 CONTROLLER DESIGN 60
6.3 PERFORMANCE EVALUATION 65
6.4 INFLUENCE OF ENCODER RESOLUTION 67
6.5 EXTRA NOTE 69
6.6 REFERENCES 72

7 CHAPTER SEVEN: TESTS 73

7.1 START UP 73
7.2 SHUT DOWN 76
7.3 PHASE CHANGE 78
7.4 STEP ON FEED-FORWARD SIGNAL 80
7.5 LOW FREQUENCY DISTURBANCE 82

8 CHAPTER EIGHT: CONCLUSIONS AND RECOMMENDATIONS 85

8.1 CONCLUSIONS 85
8.2 RECOMMENDATIONS 87
8.3 REFERENCES 87

A APPE NDIX A: MATLAB LISTING FOR H_∞-CONTROLLER 89

B APPENDIX B: MATLAB LISTING FOR BODE PLOTS 91

1 Chapter one : Introduction

In industry motor synchronisation is often used. This chapter describes what motor synchronisation is and what its main purposes are. To elucidate this, some applications are introduced.
After this, a specific application of motor synchronisation is described in more detail. This application is a so-called mailing machine, a machine which automatically gathers and packages printed papers and labels them with an address, so that they can be mailed.
In this thesis a special control scheme is developed to solve the problem of motor synchronisation for this particular problem. The principle on which this control scheme is based, is described at the end of this chapter.

1.1 Scope

Nowadays the electric motor is one of the main sources of mechanical energy. In industry, it is used in all kinds of machinery.
One such piece of machinery often contains several electric motors working together on a single product, for instance CNC-machines, robots and product handling machines.
Several applications ask for some kind of synchronisation between the different motors. A robot for instance, has to move several joints simultaneously to pick up a product. Another example is a printing press in which one motor feeds the paper into the press and another spins the ink rollers. The speeds of these two motors have to be tightly co-ordinated in order to get a correct print to paper.
In this thesis, motor synchronisation for a so-called mailing machine is described. For this mailing machine, a controller is developed that guarantees tight synchronisation at low cost.
Although the described control scheme is specifically applied to this special purpose, it should be clear that the taken considerations and developed techniques can also be used in a vast number of other applications.

1.2 Mailing Machines

The company Buhrs-Zaandam bv in Zaandam, the Netherlands builds after-press machines. These machines are used to handle all kinds of printed material such as leaflets, books and magazines. The printed materials can be gathered to form a package, which is wrapped in plastic foil and labelled with an address. These machines are used to make high volume mailings.
With a staff of around 160 people, Buhrs-Zaandam is one of the three main companies in the world selling these machines.
To clarify the functioning of such a mailing machine in Figure 1-1 an overview is given.

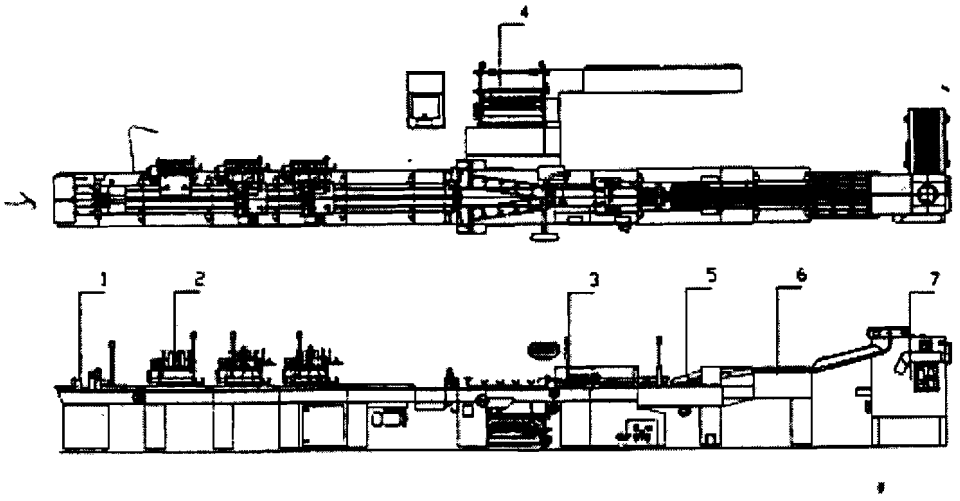


Figure 1-1 : The Buhrs-Zaandam mailing machine

The machine is built up around a transport means transporting the printed materials through different stages. Moving from left to right the main material is first put between the lug chain by a main feeder (1). Next several supplements are added to the main product by a number of other feeders (2). The main product together with the supplements form a package that is wrapped in plastic foil by a packaging module (3). The foil is being supplied by a reel stand (4). At (5) the product is released after which an address is printed on it, using a label or directly, via an inkjet printer (6). After this the product is put onto a stack (7). Now the product is ready to be sent away.
The machines Buhrs-Zaandam makes, have a very high capacity, producing 18.000 products per hour. Buhrs-Zaandam is already working on machines with an even higher capacity of about 24.000 products.
This high product throughput asks for a careful timing of the different operations. Because the lug chain moves at a constant, high speed all the operations have to be performed at exactly the right time. Otherwise the supplements are not correctly placed into the existing package or the wrapping machine would incorrectly wrap the package.
To ensure tight synchronisation between the different production stages, the modules of the gathering section are powered by one electric motor. This is done via a long mechanical axis running along the full length of the machine.
Now, for example, imagine there is a phone book between the lug chain. In general, this forms a heavier load then a leaflet and it will cause the lug chain to run somewhat slower. Because all the modules along the chain are mechanically coupled to the belt by the mechanical axis, they will all run a bit slower. Thus, synchronisation between the different modules is maintained.
To increase flexibility Buhrs-Zaandam now wants to offer the customer a modular system with which he can compose his own production line. This should enable the customer to add or remove modules to the necessity of a specific task. The problem with the mechanical coupling is that it is not easy to add or remove a module to or from the production line. In order to do this, a mechanic has to mount the module on the production line and couple it to the mechanical axis. This can take quite a long time and calls for trained staff, making this very costly.

The idea of Buhrs-Zaandam was to replace the single motor driving all the different modules, by several electric motors each driving a single module. The problem now is, how to synchronise the different electric motors. To this end an 'electric axis' is introduced. This is a system which lets the motors communicate with each other via electrical signals. The motors can share information about their respective speeds and positions to maintain synchronisation.

1.3 Master-Slave motor synchronisation

The working of an electrical axis can generally be exlained by the following figure.

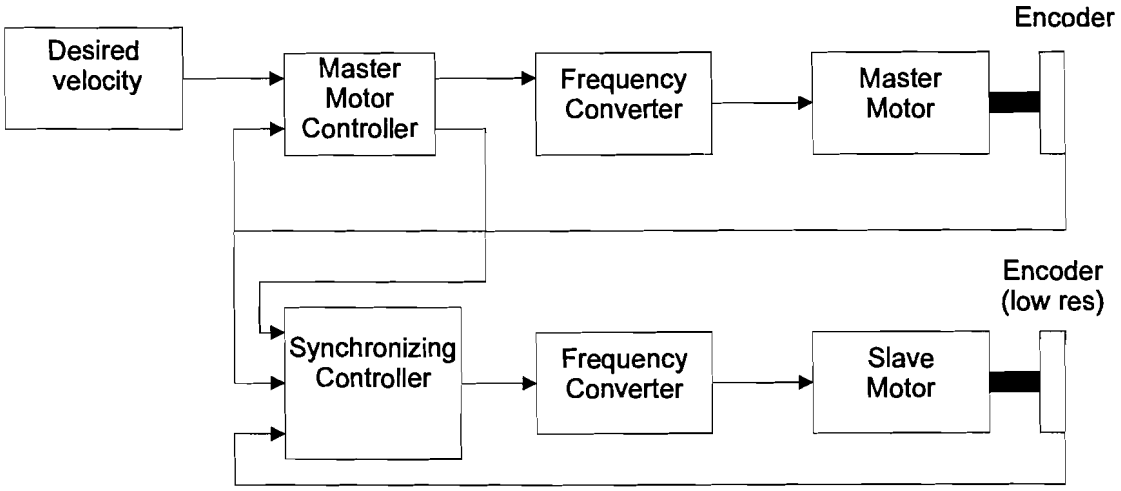


Figure 1-2 : Possible implementation of electric axis

In this figure two motors, the master and the slave are shown. These motors are powered by a frequency converter. The master motor is the reference motor to which one or more slave motors are synchronised. One can, for example, think of the master motor driving the conveyor belt and the slave motors each driving a different machine along this conveyor belt.
The user enters a desired velocity for the master axis and the master motor control unit controls the master motor to reach the desired velocity. To measure the position of the master motor an encoder is used. This is a device which gives pulses proportional to the relative position of the master motor axis. Typical encoders produce about 500 to 2000 pulses per revolution of the motor axis. By counting the pulses produced by the encoder in a certain time interval the velocity of the motor can be estimated.
The slave motor has to follow the exact position of the master motor, to maintain synchronisation. To do this a synchronisation device is used. This device uses the information coming from the master encoder as a reference and compares this with the position information from the slave motor.
Because of the high resolution of conventional encoders, they are rather costly. For Buhrs-Zaandam this makes them too expensive to use them in all the slave motors. So another approach is used. On the slave motor axis a small number of notches is fixed. If the slave axis turns, the notches also turn. These turning notches pass a switch which is fixed to earth. If a notch passes the switch, it is triggered. At that time, the exact position of the motor axis is known. Because only a small number of notches is

fixed to the axis, the resolution of the position estimation is very low compared to that of a conventional encoder.
In conventional synchronous time control schemes this resolution would be too low to ensure tight motor synchronisation. Therefore a new control scheme which works asynchronous in time is developed. The basic idea behind this scheme is described in section 1.4.
To reduce the control effort of the synchronisation device, a feed-forward signal from the master motor is directly fed to the slave motor. In this case, the control signal coming from the master controller is also fed to the slave motor. This gives a smaller reaction time of the slave motor and less control effort for the slave motor controller.

1.4 The asynchronous control scheme

In discrete control theory all control actions and calculations are made synchronously in time. This means that at equidistant discrete time points a sample of the controller inputs is taken. Based on these inputs the control action is updated. The time between two updates is constant.
At the moment of sampling an input signal, the last measured value is taken. At the instant of sampling the real value of the input signal may be changed from its last updated value. Through this a measurement error is introduced.
For example, if an encoder with only one measurement per axis revolution is used, the axis might have turned almost 360 degrees from its last updated value. This means that for the calculation of the error signal we assume that the axis has made a turn through 0 degrees, whereas in reality it might have turned as much as 350 degrees. It is clear that from this large measurement error, it is impossible to determine a correct control signal.
To circumvent this problem, one can take a large sampling time that allows the motor axis to make several turns before a new measurement value is sampled. This way the relative measurement error is smaller. However, this introduces a long dead-time in which no control actions are taken. This makes it very difficult to provide tight synchronisation.
Another way of solving this problem is not to update the measurement and control signals at discrete time intervals. Instead one should update these signals at the instant of measurement. At the instant that the switch of the slave motor is triggered by a notch, we have a position measurement with virtually no measurement error.
This control scheme is called asynchronous because updates are made asynchronous in time. A.M.Phillips (see [1]) used this method to update the estimates of a state observer asynchronous in time. The application of asynchronous control is however not restricted to LQG control. It can be used for any type of controller, whether PID, LQG or robust control is applied. The principle on which this control scheme is based, is illustrated in Figure 1-3 .

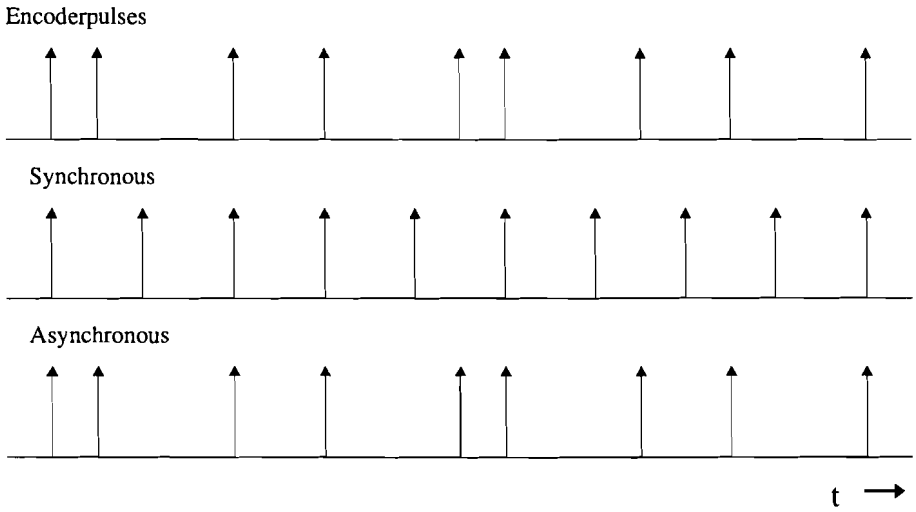


Figure 1-3 : Timing of synchronous and asynchronous control scheme

On the top axis the instants at which, of a new position measurement becomes available are drawn. In the synchronous case, shown in the middle, the control actions are updated at fixed time intervals. As the exact motor position is not known at every instant, its latest measured value or an estimate based on the measured values, is taken as the current position. In the asynchronous case, however control signals are calculated as soon as new measurement information becomes available. Thus the measurement error is zero for every update of the controller.
In the next chapter, the objectives of this project are set. Because it is not practical to work on the mailing machine as a whole, a test bed of two motors is considered. This set-up is also introduced in the next chapter.

1.5 References

[1]
Phillips, A.M. and Tomizuka, M.
MULTIRATE ESTIMATION AND CONTROL UNDER TIME-VARYING DATA SAMPLING WITH APPLICATIONS TO INFORMATION STORAGE DEVICES
In : Proceedings of the 1995 American Control Conference, Seattle, WA, USA, 21-23 June 1995
Evanston, Ill. : American Autom. Control Council, 1995, Vol. 6, p. 4151-5

2 Chapter two: Problem analysis

In the previous chapter, the concept of motor synchronisation was introduced and a specific application was given. In this chapter the Buhrs-Zaandam application and the objectives of this project are investigated. After this the test set-up on which experiments are carried out is described. For this test set-up the control problem is formulated. In later chapters, controllers are designed to solve the control problem. But first in chapter three the test set-up is modelled.

2.1 Project objectives

As described in chapter one, the output of all the modules of the gathering section is put onto a lug chain. Along this lug chain are a number of sheet feeders and a label printing machine. At the end of this chain, the mailings are packaged by a packaging section. The aim of this project is to synchronise the motion of all the different modules to that of the lug chain. Therefore an electrical axis is introduced. Each of the modules is powered by its own electrical motor and a control system is used to maintain synchronisation.
At each point in time the position of all the different motor axis should be the same, inside a given error bound. This error bound is the most significant performance measure. The larger the error bound, the worse the motor synchronisation will be. Synchronisation should be maintained not only during continuous production but also when starting or stopping the machine.
Another performance measure is the start-up time of the machine. The faster the start-up sequence is, the higher the average production will be.
Of significant importance is also the cost. The controller must be implemented on simple low cost hardware and the sensor should be as cheap as possible. As encoders usually have a high resolution, they are rather costly. Therefore the encoder is replaced by a system of notches on the motor axis. These notches are detected by an approximation switch. The resolution of this position sensor is equal to the number of notches. The notches should be mounted under equal angles. Less notches result in a simpler and therefore cheaper sensor. Ideally only one notch is used.
Summarising the target of this project is to develop a system that is able to synchronise the position of a series of induction motors within a certain bound at low cost.
Partial problems that should be solved and questions to be answered during this project are the following:

- What performance is to be obtained? How can this be measured?
- How is this performance reached? What type of controller is needed and what resolution should the position measurement have?
- How can the functioning of the obtained controller be explained? What theory can be found, that explains and predicts the controllers behaviour ?
- What are possible implementations of the controller?
- What is the cost of the developed electrical axis as opposed to that of a mechanical axis? What are the other benefits?

Because it is not practical to consider the mailing machine as a whole, a test set-up is used during this project. The test set-up configuration is described in section 2.2.

2.2 Test set-up

In Figure 2-1 the used test set-up is drawn.

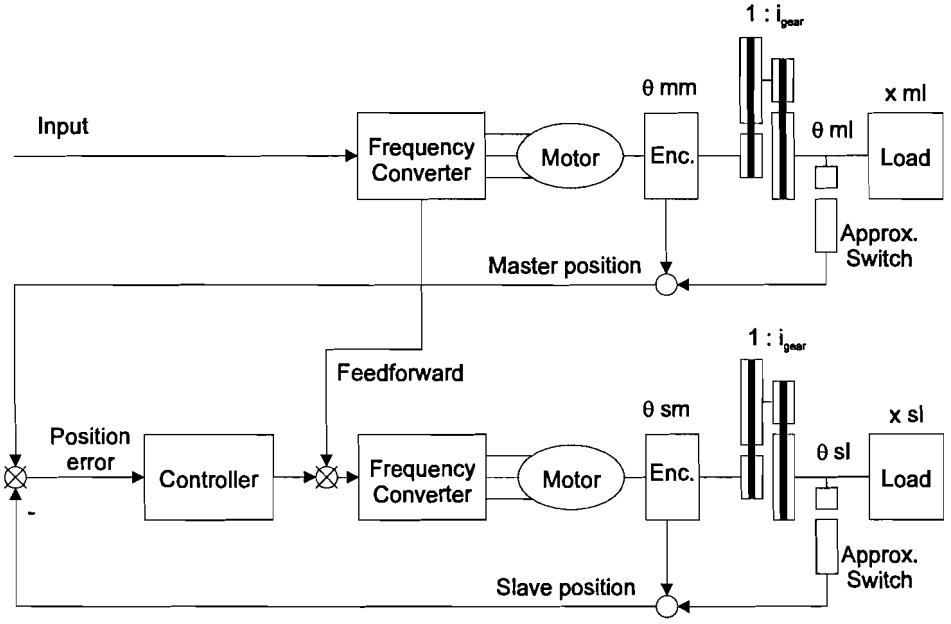


Figure 2-1 : The test set-up

The test set-up consists of two motors each driving a load. These loads are connected to the motors via a gear box. The master motor drives the lug chain and the slave drives a sheet feeder.
In the test set-up the master speed is not controlled. As the master speed is not our main concern it is left running free. A fixed voltage is applied to the master frequency converter.
The position of the motor is measured by using an encoder. This device has a resolution of 1024 measurement per revolution of the motor axis.
The aim of the controller is to control the slave motor, so that the slave load is synchronised to the master load. To aid the controller with this, a feed-forward signal coming from the frequency converter is fed to the slave as well. In the ideal case that both motors are identical and that they drive the same load, no additional controller action will be necessary.
As stated earlier, first it has to be found out what performance measures are important. If the gearboxes of the master and slave motor have equal gear ratios i_{gear}, the positions of the load axis are given by:

$$\theta_{m,l} = \frac{1}{i_{gear}} \cdot \theta_{m,m}$$
$$\theta_{s,l} = \frac{1}{i_{gear}} \cdot \theta_{s,m} \qquad (2.1)$$

Where $\theta_{m,l}$ and $\theta_{s,l}$, are the positions of the master and slave load axis respectively. $\theta_{m,m}$ and $\theta_{s,m}$ are the positions of the respective motor axis (both in radians). In general

the master motor will drive the lug chain. In the used test set-up the slave motor is driving a sheet feeder. These two loads will 'transform' the rotary positions of the load axis $\theta_{m,l}$ and $\theta_{s,l}$ (in radians) into the translational positions $x_{m,l}$ and $x_{s,l}$ (in meters).

$$x_{m,l} = C_m \cdot \theta_{m,l}$$
$$x_{s,l} = C_s \cdot \theta_{s,l} \tag{2.2}$$

Where C_i is a constant conversion factor that is determined by the dimensions of the lug chain resp. the sheet feeder. In general, for good machine synchronisation, the following must be true:

$$\left| x_{m,l} - x_{s,l} \right| \leq \varepsilon \tag{2.3}$$

Where ε is some desired error bound. As the positions of the motor axis are measured, (2.3) is rewritten as:

$$\left| C_m \cdot \theta_{m,m} - C_s \cdot \theta_{s,m} \right| \leq \varepsilon \cdot i_{gear} \tag{2.4}$$

In the used test set-up, the gear ratio i_{gear}=12.5. The conversion constant C_s is determined by the dimensions of the sheet feeder. The sheet feeder consists of a large metal drum that transports the sheets. The radius of this drum r = 0.10 m. For every radian that the load axis turns, the sheet is transported by 0.10 m. In other words C_s = 0.10 m/rad. It then follows that equation (2.4) is

$$\left| \theta_{m,m} - \theta_{s,m} \right| \leq 125 \cdot \varepsilon \tag{2.5}$$

for this set-up. As the conversion factor of the lug chain are not known, it is simply assumed that C_m is equal to C_s.

If we assume that the sheets are allowed to shift by ± 1 centimeters, the maximum error in the motor axis position is:

$$\left| \theta_{m,m} - \theta_{s,m} \right| \leq 1.25\ rad \tag{2.6}$$

In the next chapter a model will be derived, to develop a controller. Using this model, the behaviour of the system can be investigated without carrying out tests on the real system.

3 Chapter three: System modelling

In this chapter a mathematical model is derived for the asynchronous motor and its load. This model is used in later chapters to design and evaluate a controller for motor synchronisation.

3.1 System description

Reconsider Figure 2-1. In this figure the test set-up was drawn. In this chapter only the slave motor is considered. Then the following block scheme can be drawn.

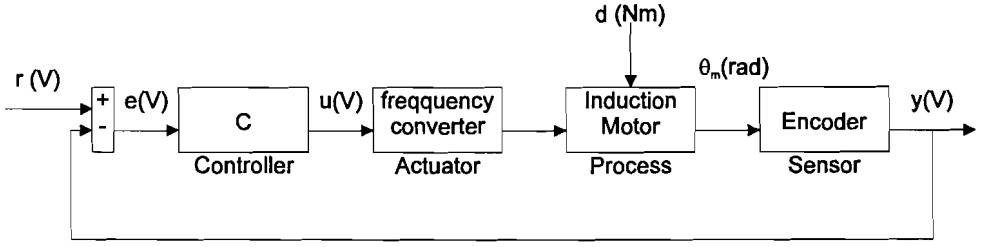


Figure 3-1 : The slave motor configuration

In this figure the feed-forward signal coming from the master frequency converter is omitted. Perturbations from the ideal case, where both motors have the same response and drive the same load, are considered. These perturbations usually result from the different loads that the master and slave motor have to drive.
The position signal of the master motor is denoted by the reference signal r. From this signal the position error is calculated. The position error is fed into the controller, which generates an output voltage u. The frequency converter, converts this to a voltage of desired frequency and amplitude, which drives the induction motor. The position of the slave motor is determined by the encoder. This gives the signal y. The torque needed to drive the sheet feeder is denoted as the distortion signal d and depends on the motor position θ_m.
In the next sections, the different parts mentioned above are modelled. First the frequency converter is covered, then the asynchronous motor. After this the sheet feeder is modelled, followed by the encoder.
To validate the model, its responses are compared to those of the real system.

3.2 Frequency converter

As mentioned before, the frequency converter converts the input voltage into three sine waves, driving the motor. In our set-up the KEB-COMBIVERT F0 is used. This frequency converter has a linear relation between the input voltage and the frequency of the output signals. However, in order not to produce too much slip, i.e. the difference between the stator frequency and the mechanical frequency, a rate limiter is included in the frequency converter.
Due to this rate limiter, the actual output frequency can be different from the commanded. Therefore the frequency converter has an extra output. The voltage of this output is directly related to the actual frequency of the output signals.
This gives the following block scheme for the frequency converter.

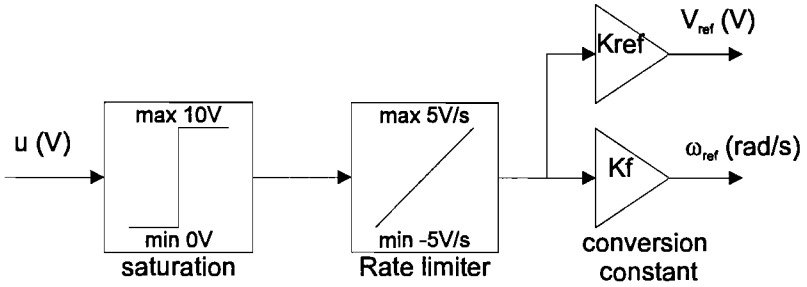


Figure 3-2 : Block scheme of frequency converter

From the manual of the KEB-COMBIVERT (see [1]) it is obtained that the standard settings are: input voltage u=0-10 V, maximum rate of limiter is 5V/s. To accurately obtain the values of K_{ref} and K_f some measurements were carried out. A series of input voltages was imposed on the frequency converter, which was driving an electric motor with zero load. The value of K_{ref} was found by taking the mean of V_{ref}/U. This was found to be 0.757.
The value of K_f was found by measuring the speed of the motor using the position encoder and a velocity observer (see [2]). It was assumed that the load torque d and the motor damping factor B both could be neglected. From Equation (3.2), it can be seen that for steady state $\omega_m=\omega_{ref}$. K_f is then determined by taking the mean of ω_m/u. This was found to be 46.3 rad/Vs.

3.3 Asynchronous motor

In this section a simple model for an asynchronous motor is derived. The used motor is of the type ODF 912-E. The following data was obtained from the motor nameplate and is used to model the motor.

Table 3-I : Motor data

	symbol	#	unit
synchronous speed	ω_{net}	$100\cdot\pi$	rad/s
rated speed	$\omega_{m,nom}$	$95.3\cdot\pi$	rad/s
power factor	$\cos\phi$	0.88	-
rated torque	T_{nom}	5.1	Nm
moment of inertia	J_{rot}	0.00137	kgm^2

A simple second order model is used (see [4]), the model parameters are derived from the name plate data. The block-scheme of this model is given in Figure 3-3.

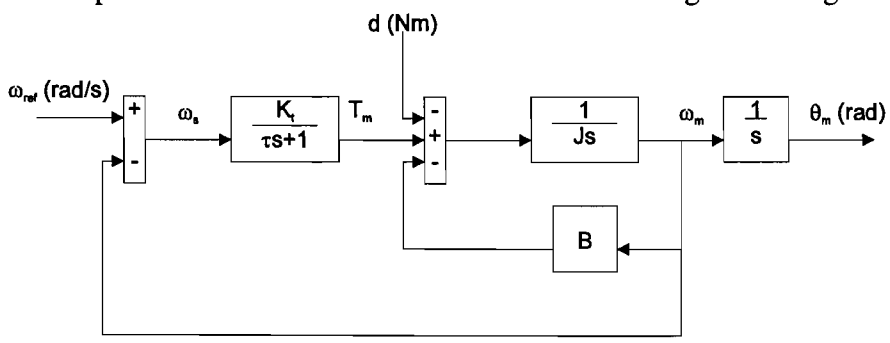


Figure 3-3 : Block-scheme of the asynchronous motor

In this circuit ω_{ref} is the output frequency of the frequency converter, d is the torque of the motor load and y is the slave motor position. For this model, the following transfer functions can be derived.

$$\theta_m = \frac{K_t}{J \cdot \tau \cdot s^3 + (J + B \cdot \tau) \cdot s^2 + (K_t + B) \cdot s} \omega_{ref} - \frac{\tau \cdot s + 1}{J \cdot \tau \cdot s^3 + (J + B \cdot \tau) \cdot s^2 + (K_t + B) \cdot s} d \quad (3.1)$$

Now the model parameters will be estimated. First the parameters associated with the electrical part of the machine will be derived. After this the mechanical parameters are considered.

3.3.1 Electrical parameters

The operation of a frequency converter controlled induction motor is based on the following. The frequency converter generates three sinusoidal signals that are out of phase with each other by 120 degrees. These signals are fed towards the stator of the induction motor. This produces a rotary magnetic field in the airgap of the motor. This magnetic field produces a flux in the squirrel cage rotor, which causes the rotor to turn. As the rotor lags behind on the stator field a mechanical torque is produced, that tries to rotate the rotor and the connected load as fast as the stator magnetic field. The amount by which the rotor lags can be described by the slip frequency ω_s, given by:

$$\omega_s = \omega_{ref} - \omega_m \quad (3.2)$$

Where ω_{ref} is the frequency of the stator flux in rad/s and ω_m is the motor (rotor) speed. The static relation between the slip frequency and the torque delivered by the motor is typically as shown in Figure 3-4.

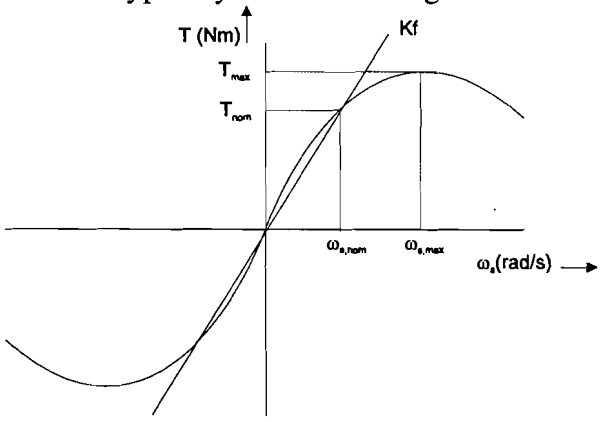


Figure 3-4 : Torque - slip angle curve

For $\omega_s > 0$, the motor is in motor operation (as opposed to generator operation). In this mode of operation, the torque delivered by the motor increases with increasing slip frequency up to a certain point. This point is known as the pull out point, this point corresponds with $\omega_{s,max}$ in Figure 3-4. Another characteristic point is the nominal operating point $\omega_{s,nom}$. For this point the produced torque and the motor frequency are given in the motor nameplate data (see Table 3-I). The torque - slip characteristic can be approximated by a constant gain in the region $[-\omega_{nom},\omega_{nom}]$. This factor can be found by drawing a straight line from the origin to the nominal operating point.
In the motor nameplate data, the speed at which the motor produces its nominal torque is given by $w_{m,nom}$ (see Table 3- I). For this it is assumed that the frequency ω_{net} of the

European public electricity net is applied to the motor. T ıis frequency is equal to ω_{net}= $100 \cdot \pi$ rad/s. From this the nominal slip $\omega_{s,nom}$ and the torque-slip factor K_t can be calculated as follows:

$$\omega_{s,nom} = (100 - 95.3) \cdot \pi$$
$$T_{nom} = 5.1\ Nm \tag{3.3}$$
$$Kt = T_{nom} / \omega_{s,nom} = 0.35\ Nms / rad$$

Figure 3-4 shows the steady-state relation between the slip frequency ω_s and the torque T. The dynamic relation is rather complex and here a simplified model from [4] will be used to describe the dynamic relation between the slip frequency and torque.

$$T = \frac{K_t}{\tau \cdot s + 1} \cdot \omega_s \tag{3.4}$$

The time constant τ is caused by the fact that, it takes some time before the current in the machine follows a change in the voltage. τ is given by:

$$\tau = \frac{L_l}{L_l + L_m} \cdot \frac{L_m}{R_r} = \sigma \cdot \tau_r \tag{3.5}$$

Here $\sigma = \frac{L_l}{L_l + L_m}$ is a ratio between the stator induction L_l and the mutual induction L_m between the stator and the rotor. $\tau_r = \frac{L_m}{R_r}$ is the relation between the mutual induction and the rotor resistance R_r.

It can be shown that for the rated operating point (see [3]):

$$\cos\varphi = \frac{1-\sigma}{1+\sigma} \tag{3.6}$$

Where cos φ is the power factor of the motor. Another equation to arrive at the nominal slip frequency is:

$$\omega_{s,nom} = \frac{1}{\sqrt{\sigma} \cdot \tau_r} \tag{3.7}$$

From equations (3.5), (3.6) and (3.7) and the nameplate data, τ can be calculated for our motor, $\tau = 0.02\ s^{-1}$.

3.3.2 Mechanical parameters

Now there are two parameters left to be determined. These are the inertia J and the mechanical damping B. For the motor with no load connected to the axis it is assumed that:

$$J = J_{rotor} = 0.00137\ kgm^2$$
$$B = 0\ Nms / rad \tag{3.8}$$

If a load is applied to the motor, the following equations hold:

$$J = J_{rotor} + \frac{1}{i_{gear}^{\ 2}} J_{load}$$

$$B = \frac{1}{i_{gear}^{\ 2}} \cdot B_{load} \tag{3.9}$$

$$d = \frac{1}{i_{gear}} \cdot d_{load}$$

where i_{gear} is the gear ratio of the gear box between the motor and the load axis. J_{load} and d_{load} are the inertia of the load and the torque needed to drive the load respectively.
The values of B_{load} and J_{load} for the sheet feeder and the lug chain, are derived below.

3.4 Sheet feeder

In the test set-up a sheet feeder is used as the load to the slave motor. On the sheet feeder a large stack of paper can be loaded. The sheets are then put between the lug chain one by one.
The sheet feeder consists of a large metal drum, that is rotated by the motor. On this drum a set of grippers, that grab the paper, is connected. The paper is transported over the metal drum to the lug chain. The grippers then release the paper. The grippers are pressed to the drum by a set of springs. These springs are tensioned and released via a cam shaft that is attached to the metal drum.
To grip the paper accurately a set of vacuum cups is used. A swing arm is used to hold the stack of paper on top of the sheet feeder. This arm swings back whenever a sheet is to be grabbed. The vacuum cups and the swing arm are kept in place by a set of tensioning springs. These springs are also pressed and released by a cam shaft, thus moving the vacuum cups and the swing arm.
The pressing and releasing of the various springs results in a torque being applied to the motor. As all the operations are cyclic with the rotation of the drum, the amount of torque that is applied to the motor is dependent on the position of the load axis (the drum). In Figure 3-5 the approximated torque as a function of the load axis position is given.

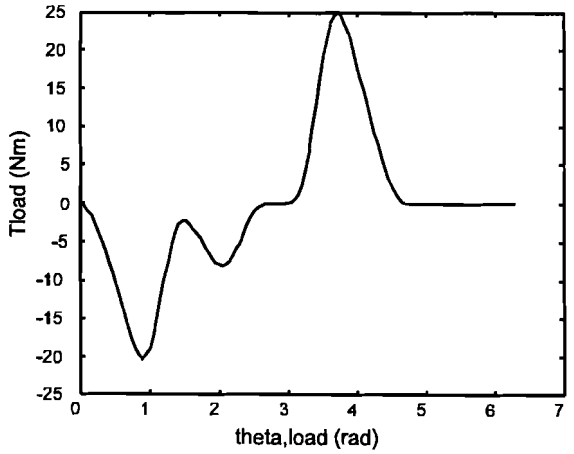


Figure 3-5 : Sheet feeder torque

This torque function was calculated from the spring constants. The drawn torque is the torque on the load axis. To obtain the torque that is applied on the motor axis it has to

be devided by the gear ratio i_{gear}. In this case i_{gear}=12.5. Next to this fluctuating torque a constant torque has to be generated by the motor to overcome the Coulomb friction. To determine the value of this torque and the value of the mechanical damping B a set of measurements was carried out.

The average motor speed was measured for a set of constant input voltages applied to the frequency converter. The average torque $\overline{T}_m$ generated by the motor is then equal to (see Figure 3-3):

$$\overline{T}_m = K_t \cdot \left(\omega_{ref} - \overline{\omega}_m\right) = K_t \cdot \left(K_f \cdot u - \overline{\omega}_m\right) \tag{3.10}$$

where $\overline{\omega}_m$ is the average motor velocity.

The average torque was thus determined for a whole set of input voltages (and speeds). This gives the following figure.

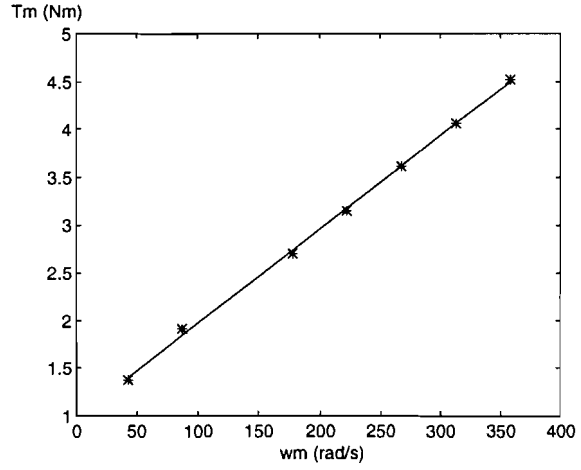


Figure 3-6: Torque-speed curve

The measurements, indicated by a *, can be approximated by a first order function :

$$\overline{T}_m = 9.8 \cdot 10^{-3} \cdot \omega_m + 0.99 \tag{3.11}$$

This function is also drawn in Figure 3-6. From Figure 3-3 it can be seen that for a constant motor speed the following is true :

$$\overline{T}_m - \overline{d} - T_f - B \cdot \overline{\omega}_m = 0 \tag{3.12}$$

Where T_f is the torque that is needed to overcome the friction. The mean value of the varying sheet feeder torque was determined to be $\overline{d} = -11.2 \cdot 10^{-3}\, Nm$. From this and equations (3.11) and (3.12) follows that $B = 9.8 \cdot 10^{-3}\ Nms/rad$ and $T_f = 1.0\, Nm$.

The only parameter that is yet to be determined is the inertia of the sheet feeder. To get a rough estimate of this inertia the metal drum is considered. Although the sheet feeder consists of a lot of rotating elements, it is expected that this drum forms the largest portion of the inertia. To allow for easy calculations the drum is considered to be a solid steel cylinder. The inertia of a cylinder is given by:

$$J_{cyl} = \tfrac{1}{2} \cdot \pi \cdot \rho \cdot l \cdot r^4 \tag{3.13}$$

Where ρ is the mass volume density of the material the cylinder is made of. For iron this is $\rho = 7.9 \cdot 10^3\ kg/m^3$. The dimensions of the cylinder are, length $l = 0.36\, m$ and radius $r = 0.10\, m$. This results in $J_{cyl} = 4.5\, kg \cdot m^2$. Using equation (3.9), the total inertia of the motor rotor plus the load was found to be $J = 30 \cdot 10^{-3}\ kg \cdot m^2$. As the drum actually is not solid but has some slots, the real inertia will be less. This is shown in section 3.7.

3.5 Lug chain

As stated earlier on, generally the master motor will drive the lug chain of the mailing machine. The average load on the lug chain is assumed constant. Therefore the torque needed to turn the lug chain can be assumed constant. As this torque is unknown, it is assumed that it is zero.
The inertia and mechanical damping of the lug chain depend on the dimensions of the lug chain. Again the values of these quantities is unknown. Therefore it is assumed that they are the same as for the sheet feeder.

3.6 Encoder

For position measurement an incremental encoder is used. The encoder used in the test set-up has a resolution of 1024 measurements per revolution. This is a rather high resolution, therefore the position measurement is considered to be 'perfect'. From this position measurement the motor speed can be obtained by using a speed observer (see [2]).
A smaller encoder resolution can be simulated by skipping a number of encoder measurements. In simulation this can be emulated by driving the position signal through a quantization function.

3.7 Model validation

The described model was incorporated in the software simulation package Simulink. The Simulink block scheme of the slave motor, driving the sheet feeder is given in Figure 3-7.

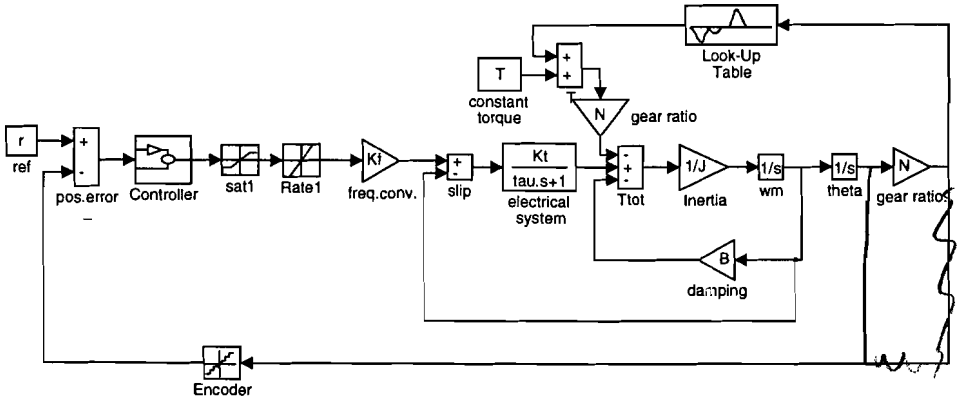


Figure 3-7: Simulink model of slave motor

The model was now validated by comparing the simulated speed signal with that of the real process. Important here is that the maximum deviation of the speed caused by the alternating sheet feeder torque is the same as the real-life system. It is assumed that the influence of this torque is our main problem in the synchronisation. The aim of the synchronisation is to make the maximum deviation between the master and slave position as small as possible. Therefore it is especially relevant to know the maximum influence of the torque on the motor.

The response of the motor to this torque depends on the value of the inertia J and the electrical time constant τ. The values of J and τ calculated earlier on gave non-satisfactory results. Therefore J and τ where iterated.

An increase in the electrical time constant τ, provides for less oscillation in the response. An increase of J gives a system with a slower response. These two influences on the response can not be separated easily, so it was a matter of trial and error before the final J and τ were found. The following values were found: $J = 8.5 \cdot 10^{-3}\ kg \cdot m^2$ and $\tau = 0.05\ s^{-1}$.

In the Figure 3-8 real and simulated speed signals are compared, for several motor speeds.

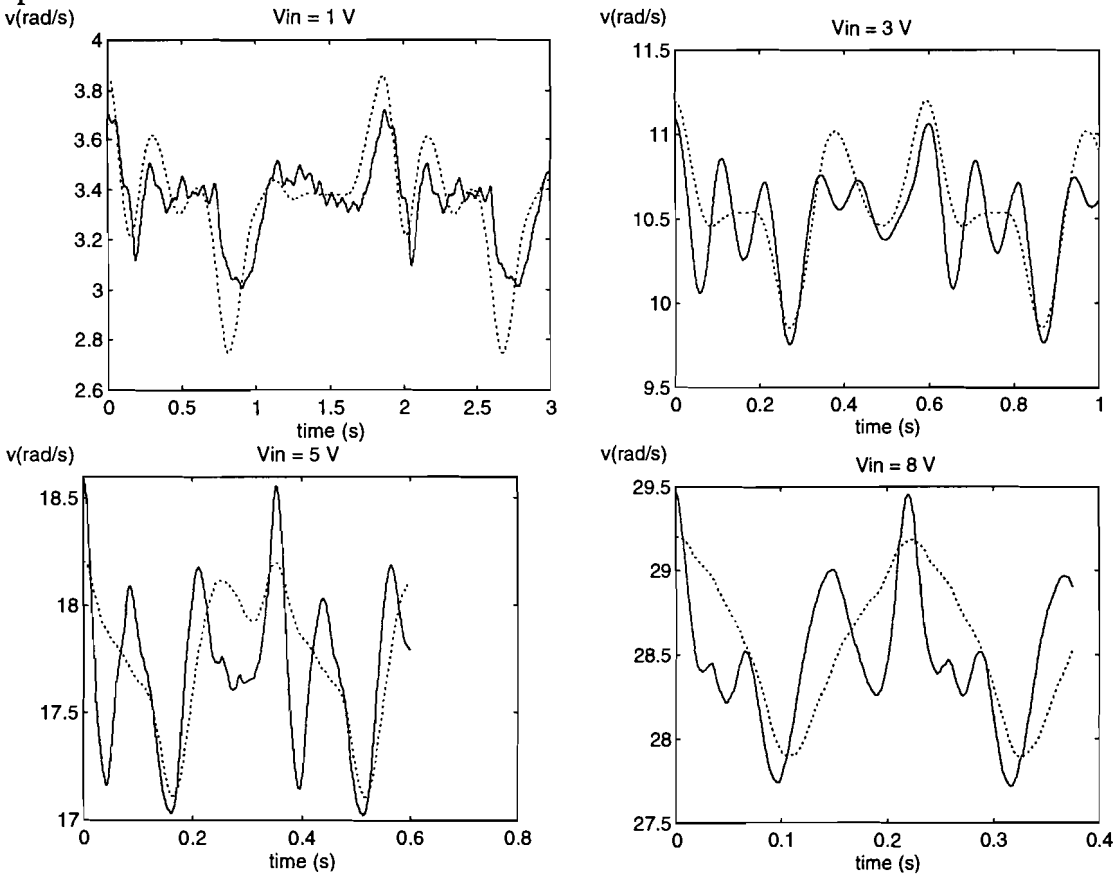


Figure 3-8 : Real and simulated speed signal

In this figure the solid lines are the real speed signals. As can be seen the simulated speed starts to deviate quite a bit for higher speeds. This can probably become somewhat better by fine tuning J and τ. However, overall the real speed fluctuates somewhat more than the simulated speed. This can be explained by the fact that the second order model derived here is only an approximation. A higher order model, would result in more high frequency components in the simulated motor speed. Another possibility is that the real torque deviates somewhat from the calculated one (drawn in Figure 3-5). The real torque was measured using a torque wrench. Indeed it could be measured that the real torque was somewhat different from the calculated one. However because of the high friction and the low precision of the torque wrench, it was not possible to measure the torque accurately.

Using the dSPACE real time interface to Simulink (see [5]), a developed controller can easily be tested for the real system. Therefore further refinement of the simulation model was omitted. Simulation results are validated on the real life system.

Substituting the found parameter values in equation (3.1) results in the following transfer functions:

$$y = 823 \cdot \frac{1}{s(s+10-27i)(s+10+27i)} \cdot \omega_{ref} + 117 \cdot \frac{(s+20)}{s(s+10-27i)(s+10+27i)} \cdot d \quad (3.14)$$

In the next chapter a conventional, synchronous controller will be developed based on this model. First a classical PID-controller is developed, after this the H_∞ method will be used to find a controller with higher performance.

3.8 References

[1]

BETRIEBSANLEITUNG KEB COMBIVERT F0 V1.2
Available from: Karl E. Brinkmann GmbH, Postfach 1109, D-32677 Barntrup, Germany.

[2]

Kamphuis, P.E.
THE DSPACE SYSTEM; A DEVELOPMENT SYSTEM FOR FAST CONTROLLER IMPLEMENTATION
M, Sc. Thesis, Eindhoven University of Technology; Department of Electrical Engineering, Measurement and control group, 1996.

[3]

Gorter, R.J.
IDENTIFICATION OF PHYSICAL PARAMETERS IN AN INDUCTION MACHINE MODEL
Eindhoven, The Netherlands; Eindhoven University of Technology, 1997.
Doctoral Dissertation
To be published by: CIP-Data Koninklijke bibliotheek, Den Haag, The Netherlands. Appendix A

[4]

Leonhard, W.
CONTROL OF ELECTRICAL DRIVES
2 nd. Edition, Springer Verlag Berlin, Heidelberg; 1990, pg. 204-237

[5]

RTI 30 / RTI 40, THE DSPACE REAL-TIME INTERFACE TO SIMULINK, DOKUMENT VERSION 2.4
Available from : dSpace GmbH, Technologiepark 25, D-33100 Paderborn, Germany

4 Chapter four: PID control

In this chapter a classical PID-controller is designed. The design is based on the model that was derived in chapter 3. The aim of the PID-control design is to get a first impression of what could be achieved using a simple controller. To get a better understanding of the systems behaviour, the open loop system is briefly investigated in section 4.1.
After this, the PID-controller is designed using the root locus method. The performance of the closed loop system is than evaluated by simulation and by tests on the real system.As it is our aim to decrease the resolution needed for the position measurement, this is also investigated. It will be shown, that decreasing the encoder resolution reduces the systems performance as expected.
To cut the effect of lower encoder resolution down, the position error that is fed to the controller is calculated in an alternative manner. This is described in section 4.5.
In this chapter it will be shown, that the PID controller is of too low an order, to place all the poles of the closed loop system satisfactorily. Therefore, in the next chapter a higher order controller is designed. For that the H_∞ design method is used.

4.1 Open loop system

The following Simulink model is used to simulate the master-slave motor combination.

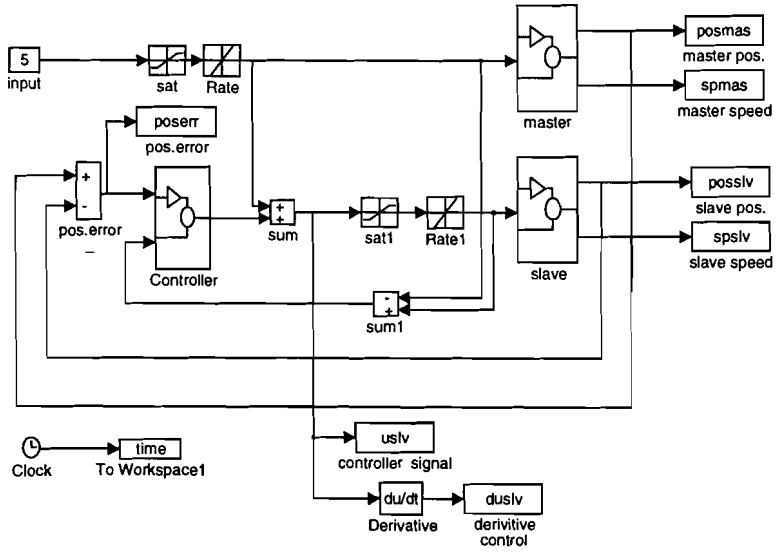


Figure 4-1: Simulink block diagram of master and slave motor

In this figure, two subsystems representing the master and the slave motor are drawn. These subsystems both contain a motor model as given earlier in Chapter 3. The only difference is that the slave motor drives the alternating plus constant torque of the sheet feeder. The master motor drives the lug chain. As stated in chapter 3, the torque needed to drive the lug chain is assumed to be constant.
If only the feed-forward signal is fed to the slave motor and no further control action is used, the 'open loop' behaviour is obtained. For a constant input to the master

frequency converter, the simulated difference between the master and slave position is given in Figure 4-2.

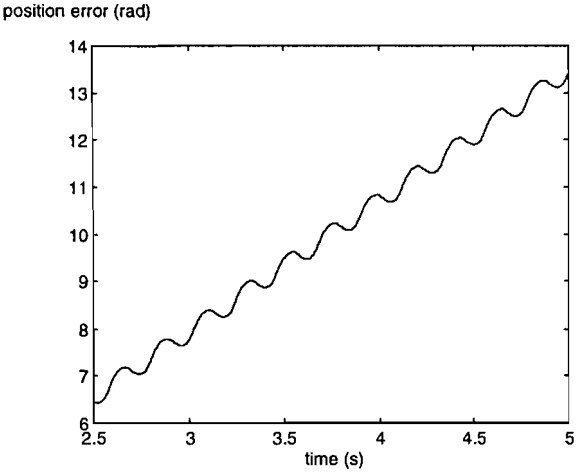


Figure 4-2 : The 'open-loop' position error

Because it takes the motors some time to arrive at a constant speed, the first 2.5 seconds of the simulation are omitted. From Figure 4-2, it is seen that the position error contains a 'steady state' factor that continually grows larger as well as an alternating part. To understand why this happens, the motor transfer function is repeated here :

$$y = 3.81 \cdot 10^4 \cdot \frac{1}{s \cdot (s+10-27i)(s+10+27i)} \cdot u - 117 \cdot \frac{(s+20)}{s \cdot (s+10-27i)(s+10+27i)} \cdot d$$
$$= H_p(s) \cdot u + H_d(s) \cdot d \qquad (4.1)$$

In this formula, u is the input to the frequency converter and d is the distortion torque. It is assumed that the master and slave motor have the same dynamics, so the inertia and damping of the loads applied to the master and the slave are considered to be the same. Therefore equation (4.1) holds for the master as well as for the slave, except for the disturbance. It is assumed that for the master d is zero, as mentioned earlier. For the slave, d consists of the alternating torque of the sheet feeder and a constant part corresponding to the friction.

As only the feed-forward signal is fed to the slave, u is the same for the master and the slave. Since d=0 for the master, the behaviour seen in Figure 4-2 can be ascribed to the response of $H_d(s)$ on the sheet feeder torque. In Figure 4-3 the characteristic pole locations and the Bode plot of $H_d(s)$ are drawn.

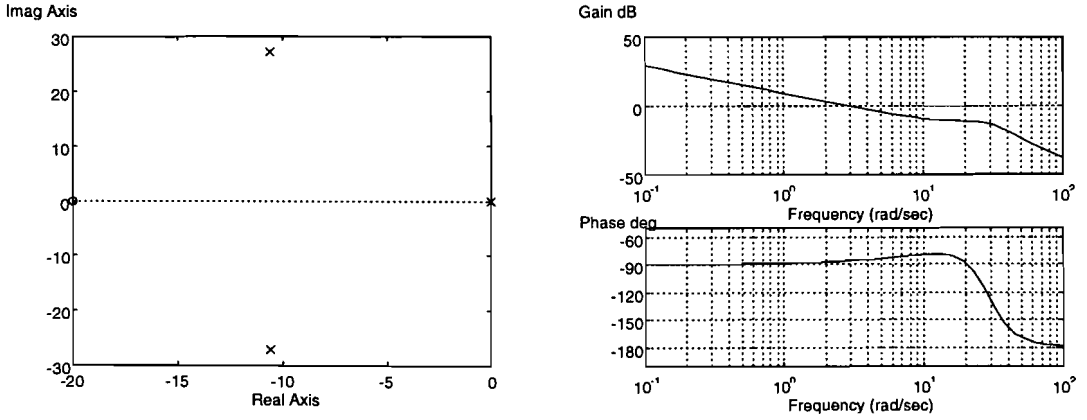


Figure 4-3 : Characteristic poles and Bode plot of $H_d(s)$

In these figures, it can be seen that $H_d(s)$ has a pole in the origin of the s-plane. For low frequencies the system behaves like an integrator. This explains the continually growing part of the position error. The integrator integrates the constant part of the distortion d. Next to this, the distortion d contains a varying part. The attenuation of this part can be read from the Bode-amplitude plot. As the torque needed to drive the sheet feeder depends on the position of the load axis (see Figure 3-5), the frequency of the distortion depends on the motor speed.

4.2 PID controller design

The control objective can be stated as to obtain good suppression of the distortion d for all frequencies where this distortion occurs.
There are now two questions that have to be answered :

- At what frequencies does the distortion occur ?
- What is good suppression ?

The distortion contains a constant part at frequency $\omega=0$ and an alternating part with frequencies dependent on the load axis speed.
In general the load axis speed will deviate from 0 up to the maximum production rate of 20.000 products per hour. This corresponds to 0 to 35 rad per second. To get a good suppression of the distortion, the higher harmonics of this distortion must also be suppressed. The frequency spectrum of the calculated torque (see Figure 3.5) is drawn in Figure 4-4.

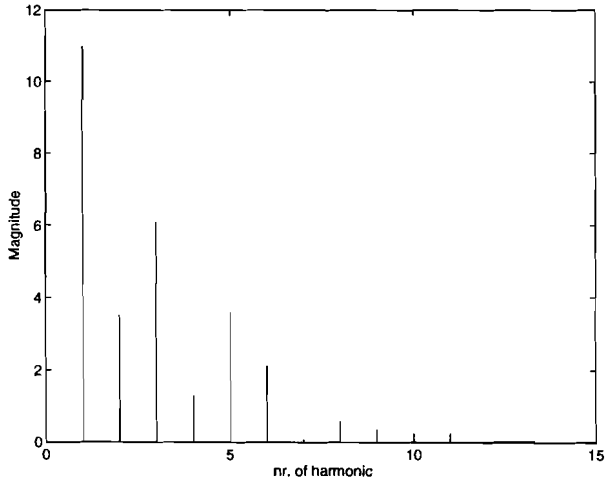


Figure 4-4: Frequency spectrum of calculated torque signal

On the x-axis the number of the harmonic is given. On the y-axis its amplitude is given. From this figure is concluded, that the up to the tenth harmonic the amplitude is relevant. Then, the distortion ranges over the frequencies from 0 to 350 rad/s.
The question of good suppression is somewhat more difficult to answer. As stated in Chapter 2, the position error between the master and the slave motor axis should be kept below ± 1.25 rad. The translation of this constraint in the time domain to a constraint in the frequency domain is far from straightforward. Therefore it is tried to make the attenuation of the distortion in the frequency domain as big as possible. Through simulations and measurements on the real system, it is then investigated whether this gives satisfactory results.
The input signal to the process, u^* consists of the controller signal plus the feed-forward signal coming from the master frequency converter. As only the deviations

between the master and slave motor position are considered, the feed-forward signal can be left out of consideration. The signal u coming from the controller, can then be expressed as: (see Figure 3-2)

$$u = H_c(s) \cdot e = H_c(s) \cdot (r - y) \tag{4.2}$$

where $H_c(s)$ is the controller transfer function.
The closed loop transfer is then :

$$y = \frac{H_p(s) \cdot H_c(s)}{1 + H_p(s) \cdot H_c(s)} \cdot r + \frac{H_d(s)}{1 + H_p(s) \cdot H_c(s)} \cdot d \tag{4.3}$$

To get an impression of what performance can be obtained using a simple, low-order controller, the PID-controller is investigated. By placing the poles of the closed loop system, we try to obtain a good attenuation of the distortion.
In general, an ideal PID-controller will have the following transfer function:

$$H_c(s) = K \cdot (P + \frac{I}{s} + D \cdot s) = K \cdot \frac{D \cdot s^2 + P \cdot s + I}{s} \tag{4.4}$$

As we want to use the root locus method, it is important to determine what a good location for the closed loop poles will be. A system with poles with larger absolute damping will have a faster response on any signal. This will result in a better suppression of the distortion. Poles with better absolute damping have a larger (negative) real part and are thus found more to the left in the left half plane.
Stated otherwise, the attenuation of a signal at a certain frequency ω is dependent on the distance of the poles and zeroes to a point on the imaginary axis corresponding to this frequency ω. In formula :

$$|H(j\omega)| = K \frac{\prod |j\omega + z|}{\prod |j\omega + p|} \tag{4.5}$$

Where $|j\omega + z|$ and $|j\omega + p|$ are the distances of the zeros and the poles to the point jω on the imaginary axis. A small value of $|H(j\omega)|$ over the frequency range of the distortion can be obtained if the closed loop poles are far away and the zeros nearby the imaginary axis for all distortion frequencies. This can only be obtained if the poles have a large negative, real part.
The PID-structure enables us to add an open loop pole in the origin and two arbitrarily placed zeros to the characteristic poles of the feeder drawn in Figure 4-3. The following characteristic loci can be found:

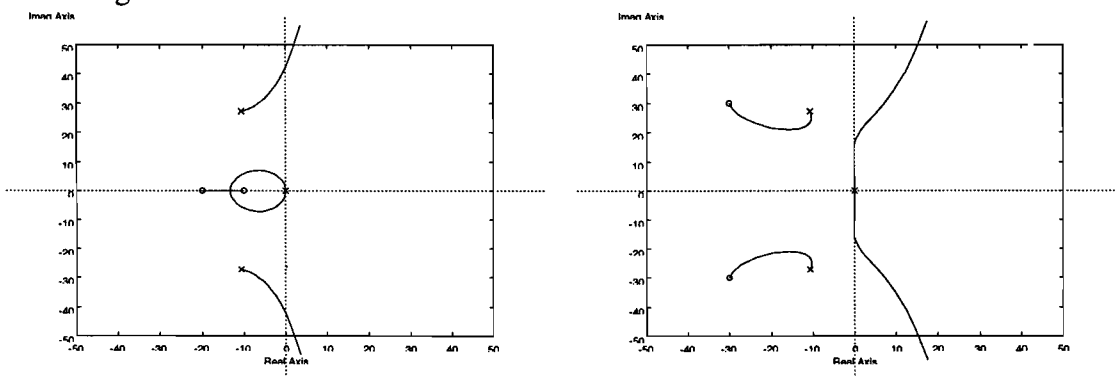


Figure 4-5 : Characteristic Root-loci of the closed loop process

With two zeroes in the controller it is only possible to attract two process poles. The behaviour of the other two results from this. Therefore it is impossible to pull all the

poles to arbitrary places. For large values of the overall controller gain K, the system becomes unstable. This limits the possibilities even further.
It was found that compensation of the poles in the origin, as drawn in the left figure, results in the best overall disturbance reduction. The overall shape of the locus does not change significantly if only one zero is used. The derivative action will not result in a better performance.
After a few iterations, the following controller was found :

$$H_c(s) = 0.21 \cdot \frac{(s+14)}{s} \tag{4.6}$$

The closed loop transfer from the distortion d to the output y, is now given by (see Equation (4.3)):

$$H_d(s) = 117 \cdot \frac{s(s+20)}{(s+4.9+22i)(s+8.1-22i)(s+5.7+14i)(s+5.7-14i)} \tag{4.7}$$

The Bode magnitude plot of this transfer is drawn as the solid line in Figure 4-6. The dashed line represents the open loop (process) transfer.

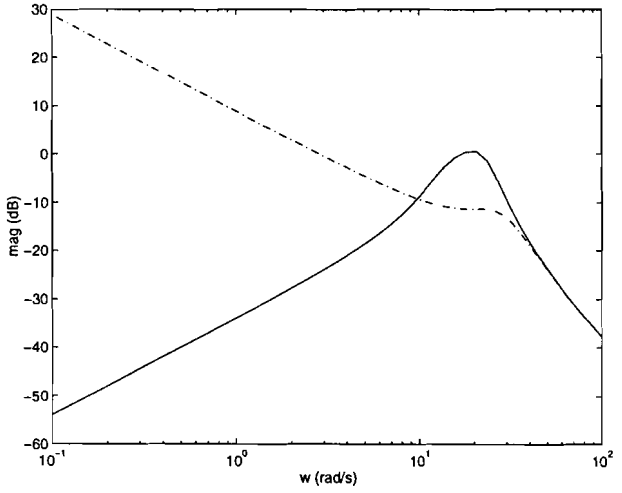


Figure 4-6 : Bode Magnitude plot closed- and open-loop

From this figure can be seen that the closed loop system has high attenuation for low frequencies. Between 10 and 50 rad/s the attenuation is less than that of the original process. This rather poor performance can be ascribed to the fact that it is impossible to attract all the poles using a low (PID) order controller. If a better attenuation is needed a higher order controller should be used. This controller will be developed in the next Chapter.

4.3 High encoder resolution

Now the performance of the found controller will be evaluated by means of simulations and measurements on the real system. To test the controller on the real system, it was implemented using a digital signal processor. The continuous controller was converted to discrete time with a sample frequency of 2 kHz. Using a zero order hold approximation, the following discrete transfer function is found for the controller:

$$H_c(z) = 0.21 + \frac{15 \cdot 10^{-4}}{z-1} \tag{4.8}$$

As the sample frequency of 2 kHz is much higher than the highest process frequency, the influence of the discretisation may be neglected. To avoid wind-up of the integrative action, the conditioning technique described in [1] was used. This results in the following Simulink block diagram for the discrete time controller.

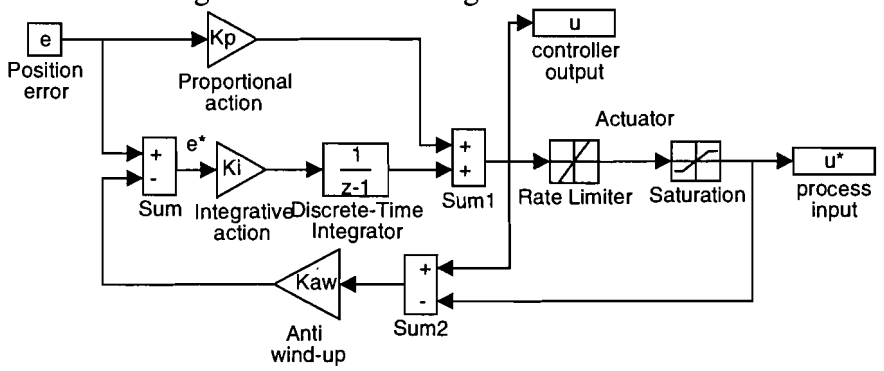


Figure 4-7 : Simulink model of PI controller

In this figure the proportional and integrative action of the controller are drawn. Also drawn are the non-linearities of the actuator. The wind-up works as follows: A step change in the position error e causes a jump in the controller output u, due to the proportional gain K_p. The rate limiter causes the process input u^* to only slowly follow this step and the response of the process will be slower than in the unlimited case. Due to the slower system response, the position error e will decrease slowly. As a result, the integral term increases much more than in the unlimited case and it becomes large. If the sign of the position error changes, the integral term still will be large. The sign of the integral term, will be opposite to the sign of the position error. The actuator still remains saturated. This will lead to a slow system response and thus to a large overshoot.

The anti wind-up tries to circumvent this problem by adjusting the value of the integrator output such that u becomes equal to u^* in case of actuator saturation. If the actuator saturates, the input to the process u^* will differ from the controller output u. The difference between these two signals is multiplied by the anti-windup constant K_{aw} and subtracted from the position error e. This yields the input to the integrative term e^*. If the anti wind-up term is equal to the position error e, e^* will be zero and the integral term will not increase any further. In [1] it was argued, that K_{aw} should be large enough to give a correct working of the anti-wind-up and small enough not to give too much noise feedback. It was argued there, that for the conditioning technique, K_{aw} should be equal to the inverse of the proportional gain K_p.

In Figure 4-8, the real and simulated position errors are compared. The solid lines are the real signals. The dashed lines are the simulated signals.

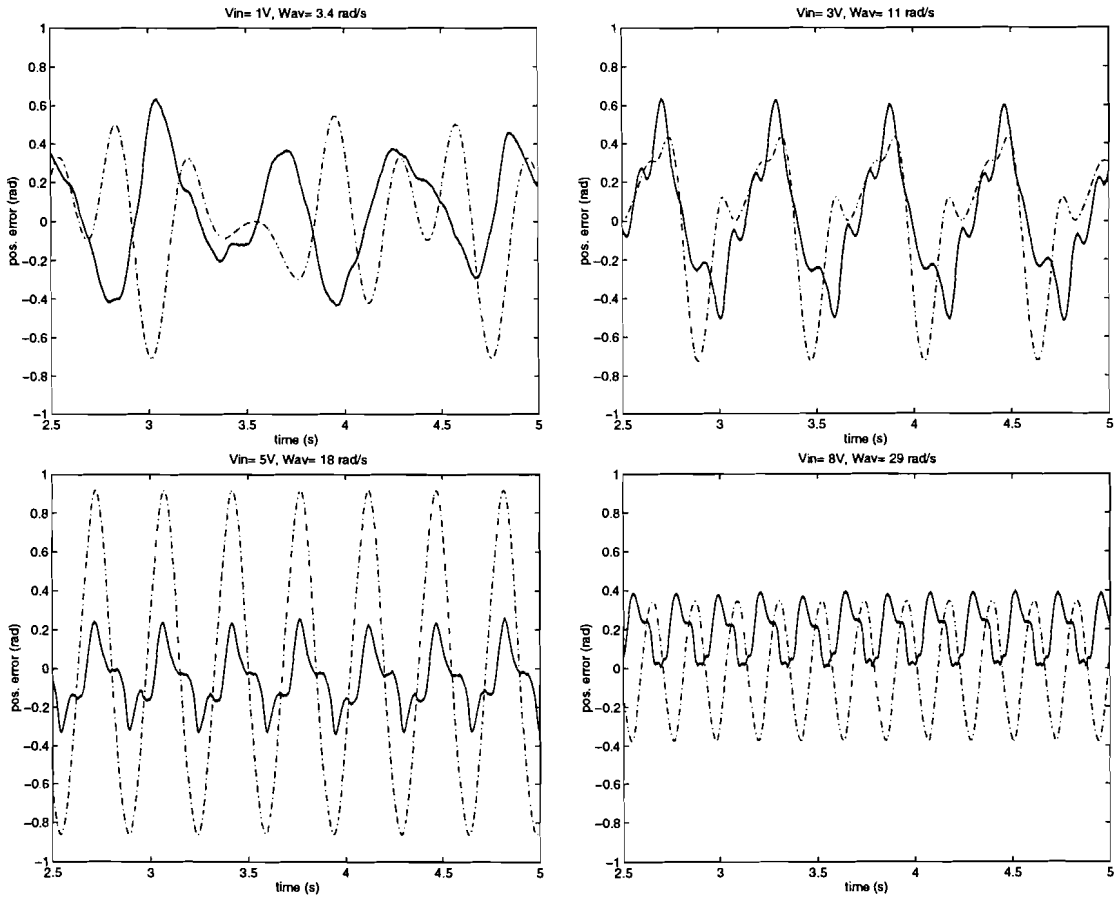


Figure 4-8: Position error for different motor speeds

In this figure, the position error in radians on the motor axis is given for different input voltages corresponding to different motor speeds. The average load axis speed is also given. As can be seen from this figure, the position error lies within the previously given error bound of ±1.25 rad.
For V_{in}=5 Volts there is quite a significant difference between the maxima of the real and the simulated position error. This can probably be explained by inaccuracies of the model. However, in this set-up the position measurement was 'near perfect'. As it is aimed to decrease the resolution of the used encoder, the influence of a lower encoder resolution is investigated.

4.4 Low encoder resolution

First the encoder resolution was lowered to one pulse per revolution on the motor axis. For a gear ratio of 12.5, this corresponds with 12.5 measurements per revolution of the load axis. The same controller as before was used, applying a sample frequency of 2kHz. In Figure 4-9 the real and simulated position errors are given.

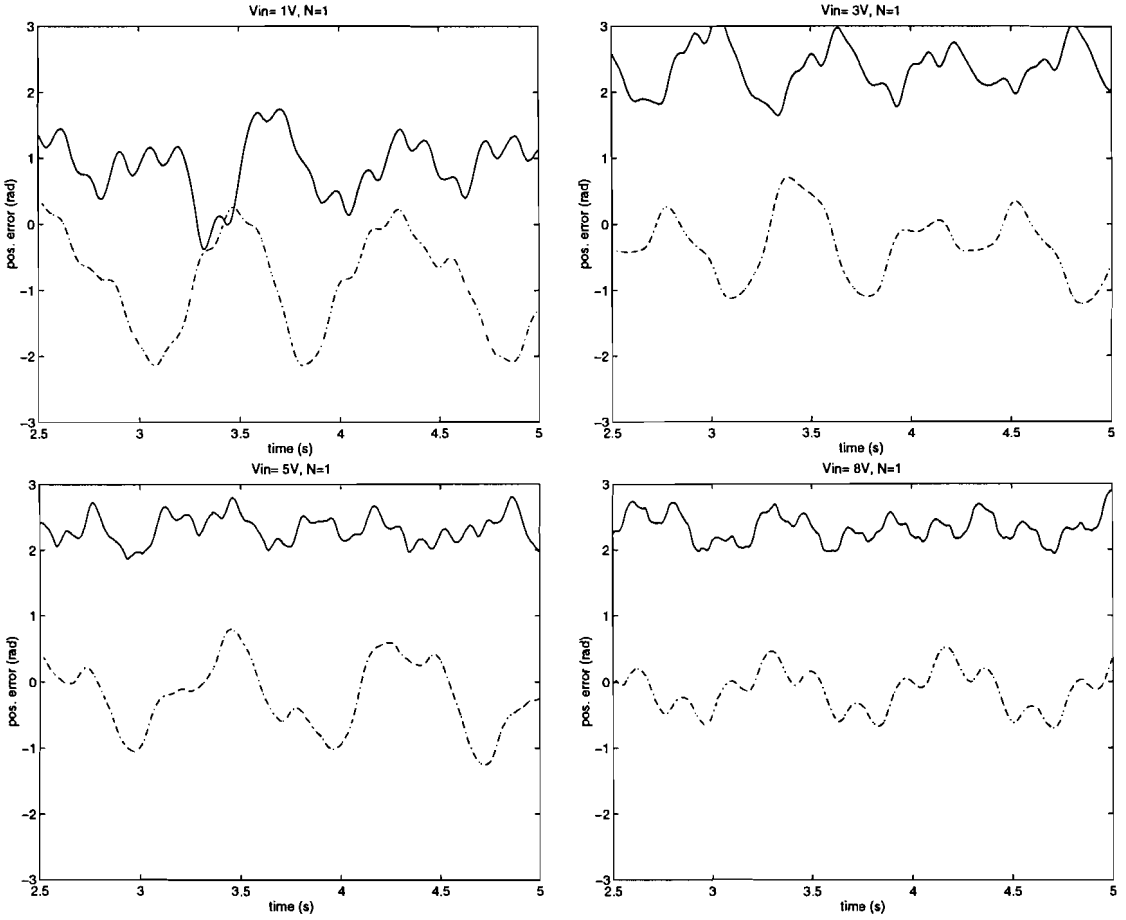


Figure 4-9 : Position error using one measurement per revolution

Most strikingly is that the average value of the position error deviates from zero. This is best seen on the real position error, but it is also present in the simulation case. The reason for this is best explained by the following figure.

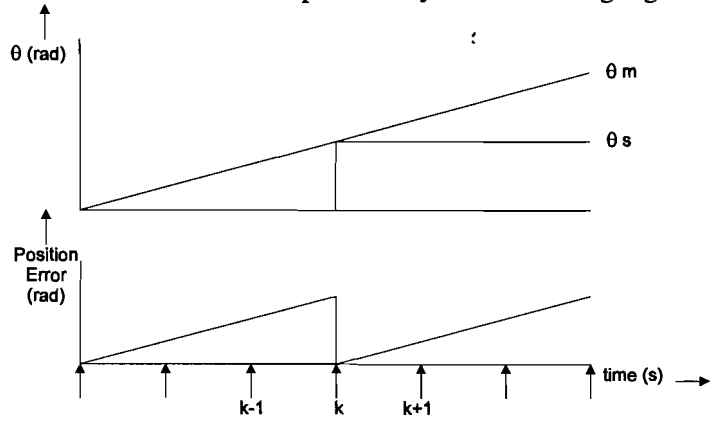


Figure 4-10 : Position measurement using low encoder resolution

In this figure, the measured master ($\hat{\theta}_m$) and slave ($\hat{\theta}_s$) positions are drawn. It is assumed that there is ideal tracking and the real position error is zero ($\theta_m=\theta_s$).
As a result of the low resolution of the slave encoder, the measured slave position will change in discrete steps, as drawn. As the master encoder has a very high resolution, its position measurement can be regarded to be continuous. This introduces a measurement error on the position error, which is also drawn in Figure 4-10. The maximum value of this measurement error is equal to $2\cdot\pi/n$, where n is the number of measurements per revolution of the slave motor axis. As the frequency of the error signal is equal to the motor axis speed, the error will be a quickly varying signal.
Due to the proportional action of the controller (see Figure 4-7), the controller output will also vary quickly over time. The process input u^* will try to follow this signal, but will lag behind due to the rate limiter. This introduces a constant difference between u and u^*. To verify this, u and u^* were measured on the real system.

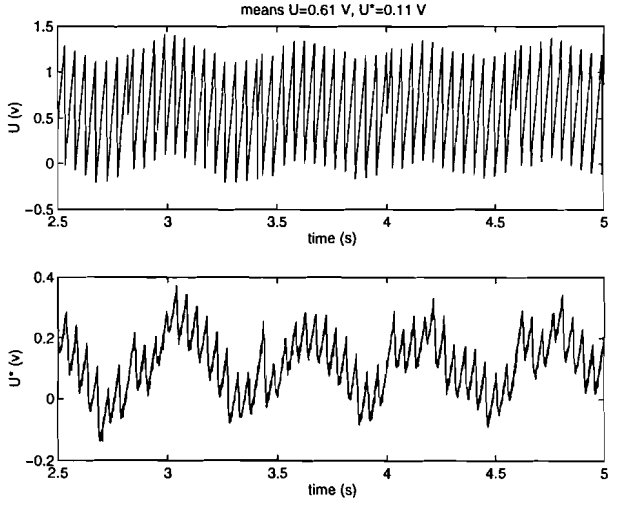


Figure 4-11: Controller output u and process input u* using low res. Encoder

From the measured data the mean values of u and u^* were obtained. It was found that u=0.61 Volts on average and u^* is 0.11 V. This is quite a significant difference. It was assumed that this difference is the effect of the rate limiter. This was investigated by simulation in Simulink. A saw tooth signal with a mean value of zero was applied to a rate limiter. The output of this rate limiter was investigated and compared to its input. The following Simulink block scheme was used.

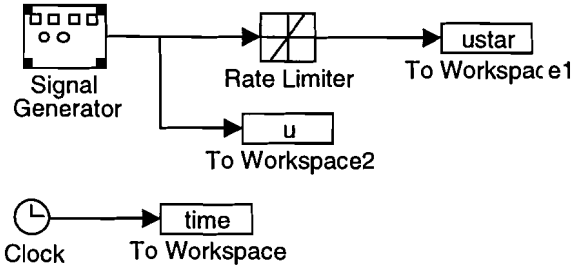


Figure 4-12: Simulink block diagram of rate limiter test

The rate limits were set to + and - 1 V/s maximum. The saw tooth signal had a mean value of 0V and an amplitude of 1 V. The frequency was chosen 5 rad/s. In the following figure the simulated output of the rate limiter as well as the original saw-tooth signal is given.

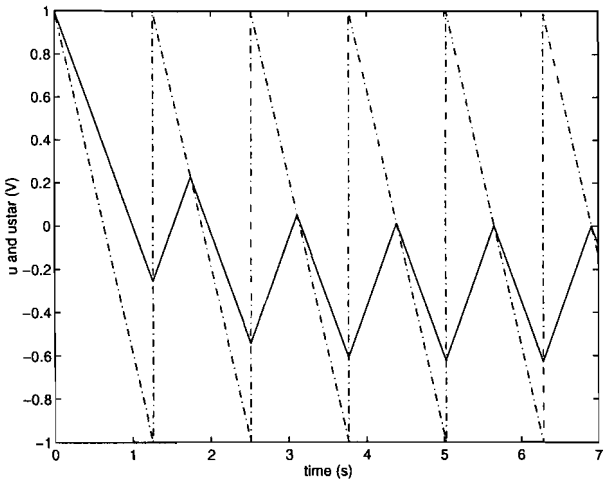


Figure 4-13: Simulation of saw-tooth through rate limiter

In this figure, the dashed line is the original saw tooth signal, the solid line is the output of the rate limiter. Although the saw toot has a mean value of 0 Volts, the output of the rate limiter has a mean which is not 0. From this can be concluded that the difference in the means of the signals u and u^*, is caused by the influence of the rate limiter.
Through the anti wind-up, this difference is fed back to the integrative term of the controller. The output of the integrative term will not change anymore when e^* is equal to zero. As the anti wind-up term has a certain mean value, the mean value of the position error will not be zero.
Now the influence of changing the encoder resolution is investigated by varying the number of pulses n between 1,2,4 and 8 per revolution. This is done for a fixed motor speed equivalent with 1 Volt input voltage (u). In Figure 4-14, the results are given both for the simulation and the real system.

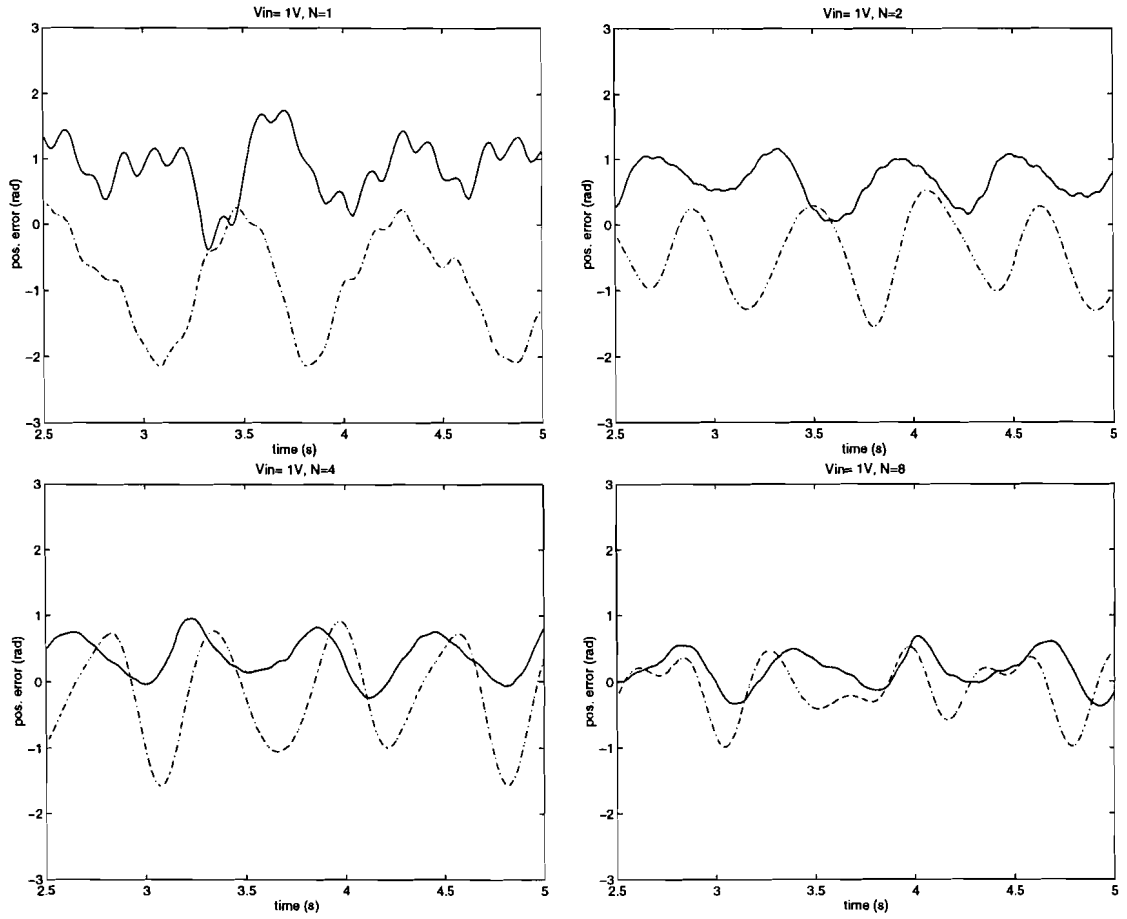


Figure 4-14: Position error for varying encoder resolution

As expected, a higher resolution of the slave encoder results in a smaller quantization error. The position measurement is more accurate and thus a smaller position error is obtained.

For low encoder resolutions, there is a large difference between the mean of the real and simulated position error. Earlier on, it was stated that this non-zero mean arises from the rate limiter of the frequency converter. Therefore the difference between the real and simulated means can be explained from the fact, that the simulated rate limiter, behaves somewhat different than the real rate limiter. This was not thoroughly investigated.

If a longer sample time is taken for the controller, the relative error in the position measurement will decrease. This will however introduce long dead times, in which no control action is taken. This makes it difficult to control the motor position accurately. A possible solution to the problem of low encoder resolution, is investigated in section 4.5. There the measured position error is kept constant between two measurements on the slave motor. This gives less high frequency components in the measurement error, which could possibly lead to a better performance.

4.5 Alternative position error calculation

As was shown in section 4.4, the quickly varying measurement error poses a severe problem when using a PID-controller with anti wind-up. A possible solution could be to keep the measurement error constant in between two slave motor position measurements. That is, implement a zero order hold circuit, whose value is updated whenever a new slave position measurement becomes available. This can be seen as a very simple state observer, where one state of the process, namely the position is kept constant. As such this is comparable to [2], except then all the states were estimated by using a model of the process, whereas here the system is modelled by keeping the position constant. In Figure 4-15, the output of such a circuit on some signal is shown.

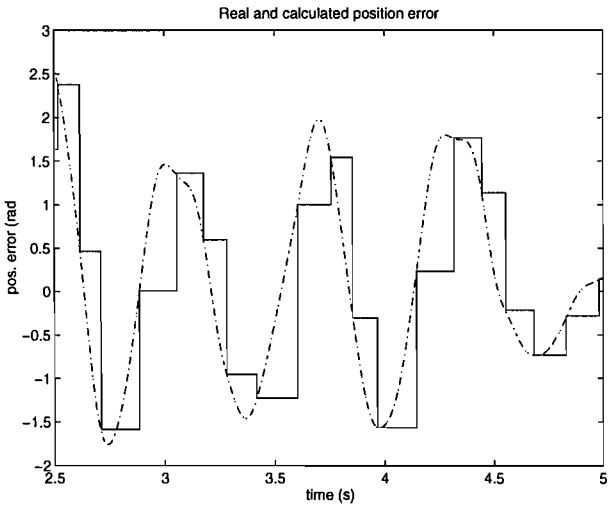
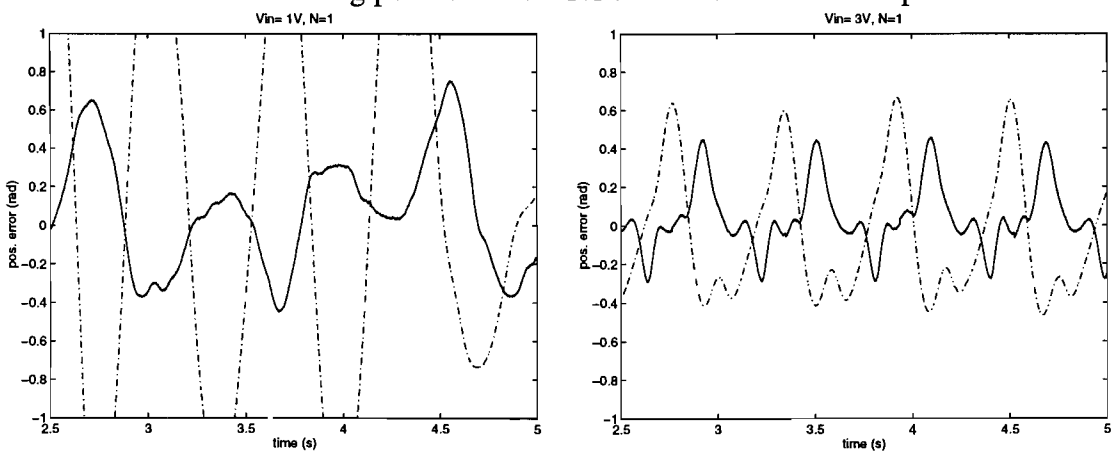


Figure 4-15: Real and calculated position error

In Figure 4-15, the dashed line is the input to the circuit and the solid line is its output. The calculation of the position error is now done asynchronous in time. The control action however, is still updated at the much higher synchronous sample rate, which is still 2kHz. This poses a problem for the integrative action. As the measured position error is constant, the value of the integrator continually grows larger between measurement updates. This results in very large values of the integrative action, which cause instability.

Therefore the value of the integrative action K_i was decreased. After some iterations on the real process, the following value for K_i was found : K_i=0.00015.

This results in the following position errors for the various reference speeds.



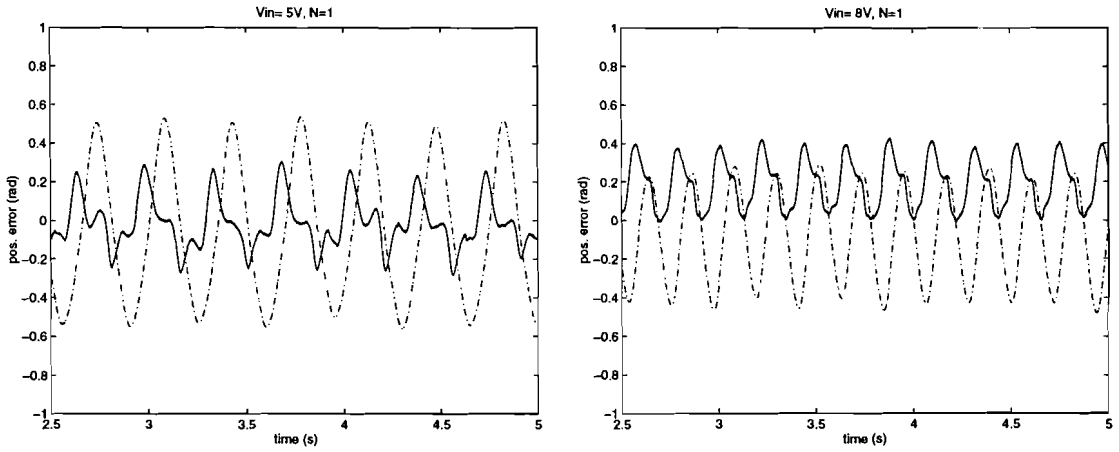


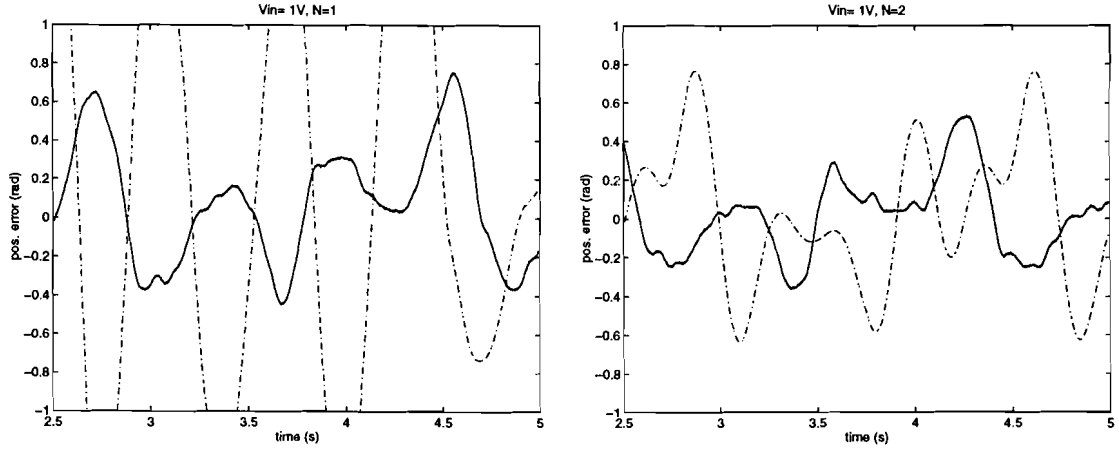
Figure 4-16: Position error with asynchronous position error calculation

If this figure is compared to Figure 4-8, where the full encoder resolution is used, it can be seen that the lower encoder resolution does not result in larger position errors ! This seems quite remarkable, but is explainable by the fact that the PI-controller only works on the low frequency components of the position error (see Figure 4-6). The asynchronous position error calculation provides a good fit to the real position error for these low frequencies. This explains the good results using a lower encoder resolution.

A problem is that the update of the position error only happens if the slave motor is turning. It was found out that for low speeds, the controller output sometimes grows larger (negative), than the feed forward signal. Now a negative reference frequency is applied to the motor. This causes the motor to stop and as the position error is not updated anymore, the control action stays the same and the motor is stuck. This problem will be further investigated in Chapter 6.

If a controller is developed that also controls the higher frequencies of the position error, a bigger difference between low and high encoder resolutions will be observed. This is investigated in Chapter 5, where a H_∞ based controller is developed.

For reference some plots of the position error for various encoder resolutions are also included.



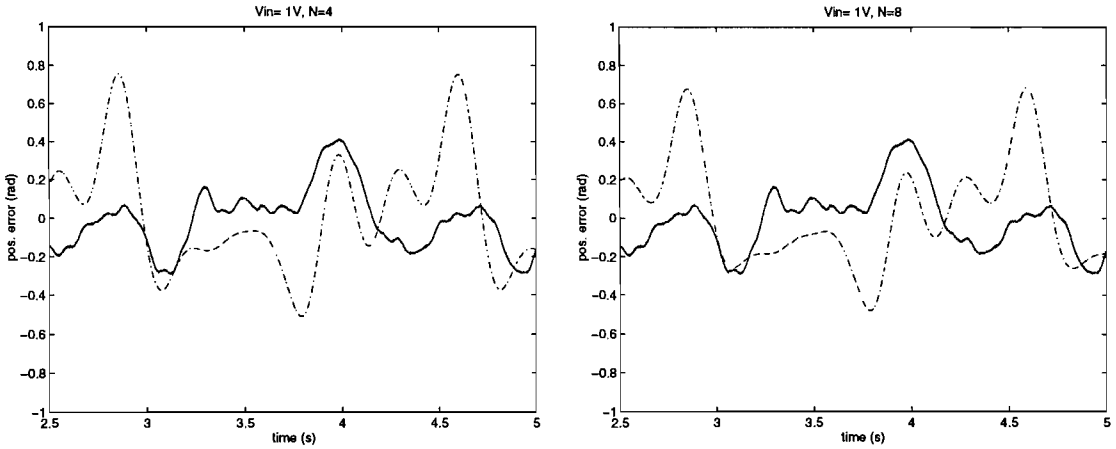


Figure 4-17: Asynchronous position errors for various encoder resolutions

For V_{in}= 1V, increasing the encoder resolution beyond two pulses per revolution does not give a significant change in the position error. In Chapter 6, the effect of lower encoder resolutions will be investigated further. But first the H_∞ design method is used to obtain a higher order controller.

4.6 References

[1]

Peng, Y.;D. Vrancic and R. Hanus
ANTI-WINDUP, BUMPLESS, AND CONDITIONED TRANSFER TECHNIQUES FOR PID CONTROLLERS
In : Control Systems, IEEE Control Systems Society
Volume 16, Nr. 4, August 1996

[2]

Philips, A.M. and Tomizuka, M.
MULTIRATE ESTIMATION AND CONTROL UNDER TIME-VARYING DATA SAMPLING WITH APPLICATIONS TO INFORMATION STORAGE DEVICES
In: Proceedings of the 1995 American Control Conference, Seattle, WA, USA, 21-23 June 1995
Evanston, Ill.; American Autom. Control Council, 1995, Vol.6, p. 4151-5

5 Chapter five: H_∞-control

In this chapter, a higher order controller is designed using the H_∞-method. With a simple PI-controller it is not possible to suppress the disturbance over the whole frequency range (see Figure 4-5). Therefore it is investigated, what can be achieved using a higher order controller.
The H_∞ design method is a well-known method to arrive at such a controller. Over the years it has been thoroughly investigated. Several books and numerous articles have been written covering both the theory and applications of H_∞ (see [1]). Therefore the H_∞ design method is not explained here.
First the performance using full encoder resolution is investigated.

5.1 Design

For the H_∞ design, the standard control loop drawn in Figure 3-2, was transformed and extended with weighting filters, which results in the following block scheme.

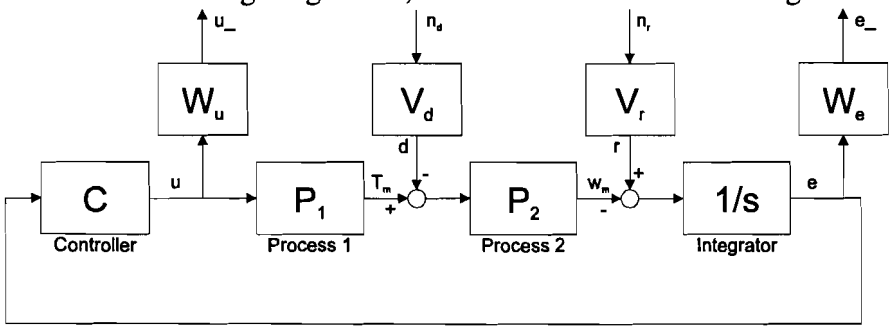


Figure 5-1: Block scheme extended with weighting filters

In this figure C is the controller transfer. P_1 is the transfer from the actuator input signal u (in Volts) to the torque generated by the motor T_m (in Newton meters). P_2 is the transfer from the summation of T_m and the disturbance torque d (in Nm) to the motor speed ω_m (in radians per second).
Note that the reference signal r is the speed of the master motor (in rad/s) and not the master motor position. From r and ω_m the speed error is calculated, this is integrated to obtain the position error e (in rad). This is done to make the weighting filter selection more straightforward. It has no impact on the found controller.
Furthermore, in this picture the in- and output weighting filters are drawn. The inputs n_d and n_r are weighted by V_d and V_r respectively, to obtain the actual inputs d and r. The actual outputs u and e are weighted by W_u and W_e, to obtain the penalised outputs $\tilde{u}$ and $\tilde{e}$ (denoted in the figure by u_ and e_).
If the process is regarded as a MIMO system with inputs r and d and outputs u and e, the following state space representation can be found (see also Chapter 3).

$$\begin{bmatrix} \dot{e} \\ \dot{\omega}_m \\ \dot{T}_m \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 0 & -\frac{B}{J} & \frac{1}{J} \\ 0 & -\frac{K_t}{\tau} & -\frac{1}{\tau} \end{bmatrix} \cdot \begin{bmatrix} e \\ \omega_m \\ T_m \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & -\frac{1}{J} & 0 \\ 0 & 0 & \frac{K_t \cdot K_f}{\tau} \end{bmatrix} \cdot \begin{bmatrix} r \\ d \\ u \end{bmatrix} \tag{5.1}$$

$$\begin{bmatrix} u \\ e \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} e \\ \omega_m \\ T_m \end{bmatrix} + \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} r \\ d \\ u \end{bmatrix}$$

The controller is now given by :

$$u = \mathbf{C} \cdot e \tag{5.2}$$

Where **C** represents the transfer of the controller.

Now the weighting filters are chosen as follows:

V_r Represents the master motor speed

As mentioned in Chapter 3, the maximum motor speed is limited by the frequency that the frequency converter can generate. The maximum frequency is about 450 rad/s. This corresponds to a production rate of about 20.000 products per hour. 450 is therefore taken as the amplitude of the weighting filter for $\omega = 0$. The feed-forward signal coming from the master motor largely covers the reference tracking. Therefore, it is assumed that the reference tracking is less important than the disturbance rejection. A low pass filter, with a very low cut-off frequency is chosen. The cut off frequency is chosen at 0.2 rad/s.

This results in the following weighting filter:

$$V_r(s) = \frac{450}{\frac{1}{0.2} \cdot s + 1} \tag{5.3}$$

V_d Represents the distortion torque.

As seen from Figure 3-5 the amplitude of the alternating sheet feeder torque is about 25 Nm on the load axis. This corresponds with a maximum torque of $25/i_{gear} = 2$ Nm on the motor axis, where $i_{gear}=12.5$ is the gear ratio. The frequency of the alternating torque is dependent on the load axis speed. The speed of the load axis ranges from zero to the maximum speed = $450/i_{gear} = 36$ rad/s. This is also the maximum frequency of the base component of the distortion torque. In the previous chapter, it was found that the torque signal has 9 higher harmonics that have a significant amplitude. The maximum frequency of the distortion signal thus is 360 rad/s. This leads to the following weighting filter :

$$V_d(s) = \frac{2}{\frac{1}{360} \cdot s + 1} \tag{5.4}$$

W_u Represents the limitations on the actuator.
First the non-linearities of the frequency converter are left out of consideration. This is done to investigate the possible system performance without the constraining non-linearities.
The weighting filter is taken constant.
The maximum amplitude of the actuator signal is 10 volts. This corresponds to the following weighting filter :

$$W_u(s) = \frac{1}{10} \tag{5.5}$$

W_e Represents the penalty on the position error.
The position error should be as small as possible for all frequencies. Thus it is taken constant. The constant value W_e is iterated until a controller can be found that complies with all the performance measures. In H_∞ design this means that the value of γ should be close to one. γ is the upper bound of the H_∞ norm of the closed loop system. After some iterations it was found that the following value of W_e produces a γ close to one :

$$W_e(s) = 1000 \tag{5.6}$$

The γ iteration was carried out using the LMI toolbox, which is available for Matlab. The listing of the Matlab file is given in Appendix A. Eventually the value of γ was found to be 1.00.
A fourth order controller was found. The Bode plot and pole zero map of this controller are given in Figure 5-2.

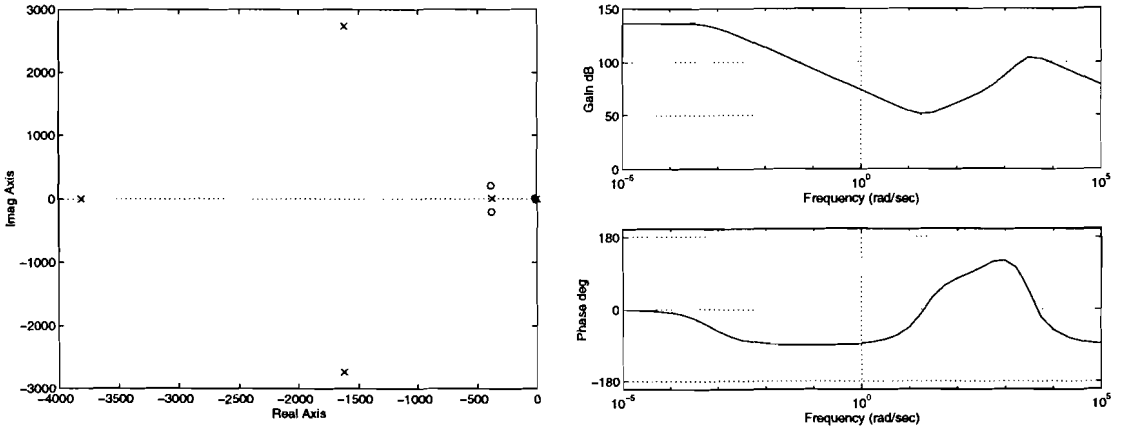


Figure 5-2: Pole zero map and Bode plot of H_∞ controller

In Figure 5-3, a part of the root locus of the closed loop system is drawn.

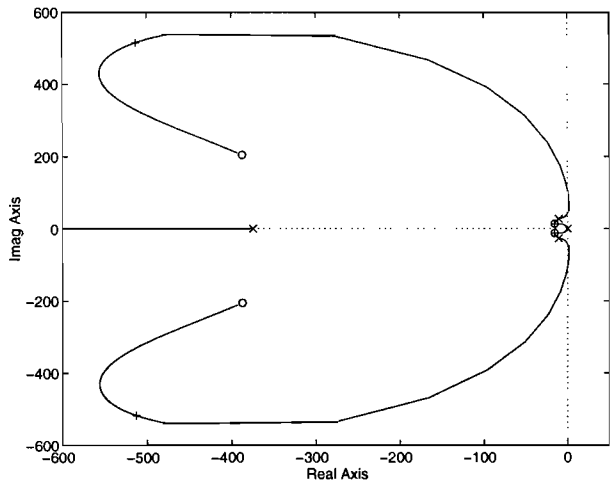


Figure 5-3: Root locus of characteristic equation

It was found that the controller has a pole very close to the origin. This leads to a small steady state error. This pole and the pole in zero of the original system are pulled to the left by two zeros. By that, a good suppression of the low frequency distortion is achieved.
The two complex conjugate poles of the original process are pulled away by two zeros that lie far away in the left half plane. This gives two poles with a very large real damping. These are denoted by the two '+' signs. As argued in Chapter 4, these poles with large (negative) real parts give a large attenuation of the disturbance.
To investigate what disturbance suppression is achieved, the Bode magnitude plot of the transfer from the disturbance to the position error is drawn below.

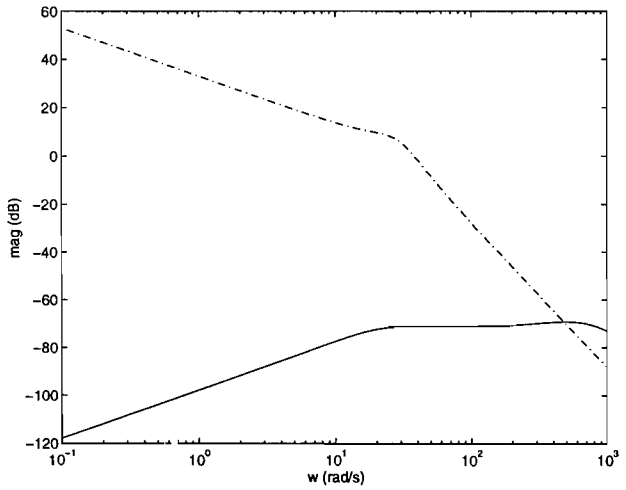


Figure 5-4: Bode Magnitude plot of disturbance attenuation

The suppression of the open loop process is again drawn as the dashed line. As can be seen from this picture the controller gives a large suppression of the disturbance for the whole frequency range of the disturbance (0 -360 rad/s).
In the next section, the performance of this controller is evaluated by means of simulations.

5.2 Simulations on the linear system

For the simulations, again the Simulink model, as drawn in Figure 4-1 is used. As the controller is designed, disregarding the rate limiter of the actuator, it is omitted from the simulation model. In Figure 5-5, the simulated position errors are plotted for four different reference speeds.

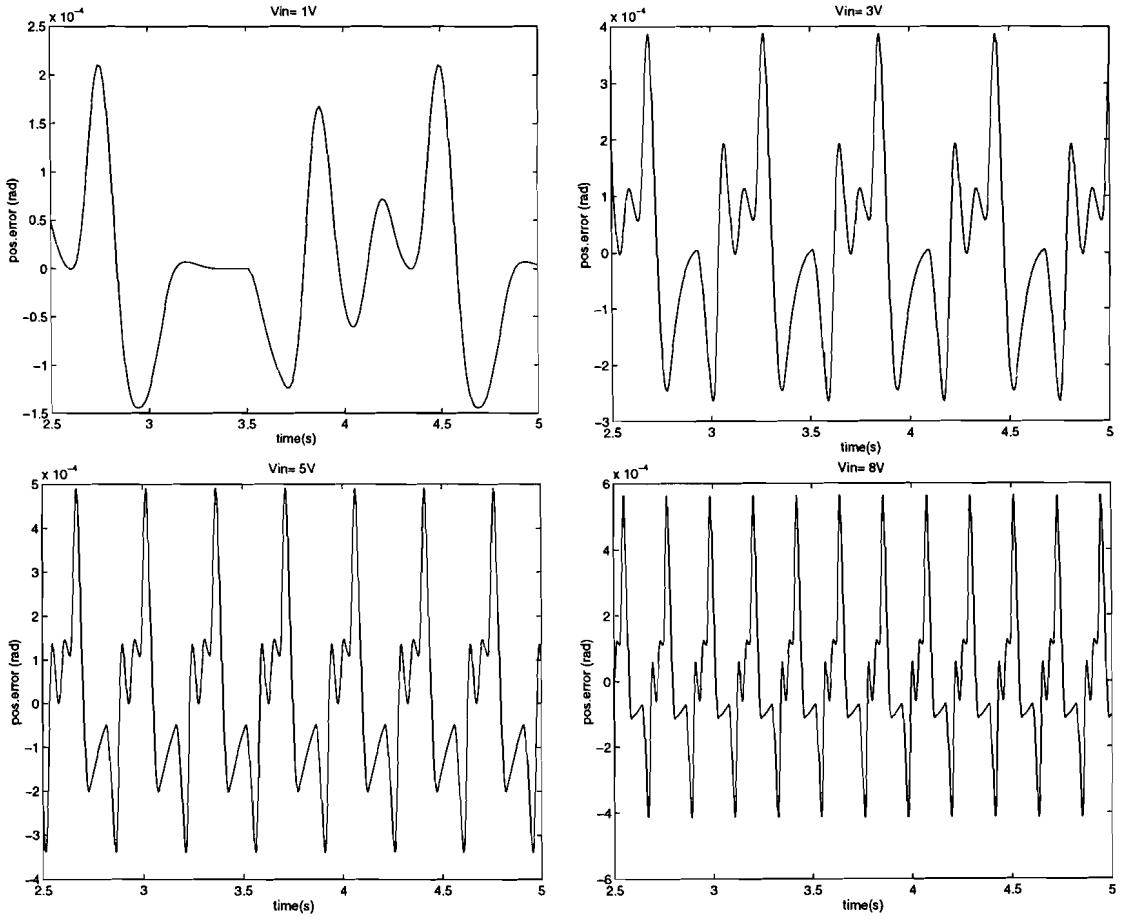


Figure 5-5: Simulated position errors using H_∞ controller

The maximum position error is less than $1*10^{-3}$ rad for all reference speeds. This is a great improvement over the performance of the PID-controller (see Figure 4-7). However when the control signal was investigated, it was found that it varies as much as 80 Volts per second for a reference voltage of 8 Volts. This largely exceeds the allowed rate of 5 Volts per second by far, so that this controller can not be implemented on the real system. Therefore the controller was altered.

5.3 Design for non-linear actuator

By putting more penalty on the higher frequencies of the actuator signal, it was tried to find a controller with an output signal, that has a maximum rate of plus or minus 5 Volts per second.
The weighting filter W_u, that penalises the actuator signal is changed to a high pass filter. The allowed constant voltage is still 10 Volts. Arbitrarily, the pole of the weighting filter transfer was placed at a very high $1*10^4$ rad/s.
The frequency of the zero was altered until it was shown that the maximum rate of 5 Volts per second was not exceeded. This was done by carrying out simulations and investigating the derivative of the controller signal. As the position error changes faster for higher motor speed, it is expected that the control signal changes more quickly in these cases. Therefore it is sufficient to investigate the controller signal for a high reference speed.
Finally it was found that the zero should be placed at 1 rad/s. The maximum fluctuation of the controller signal was now about 5 Volts per second for a reference voltage of 8 Volts. From simulations, it followed that for lower reference speeds, the rate of change was indeed lower. This gives the following weighting filter:

$$W_u(s) = \frac{1}{10} \cdot \frac{s+1}{\frac{1}{1 \cdot 10^4} \cdot s + 1} \tag{5.7}$$

The weighting filter on the position error W_e, also had to be changed. Under these severe actuator limitations, it is not possible to keep the position error below $1*10^{-3}$ rad. The constant value of W_e was again iterated until a γ close to one was found. It was found that W_e should be 1 for a γ of 0.98.
A sixth order controller was found. The pole zero map and Bode diagram of the found controller are drawn in Figure 5-6.

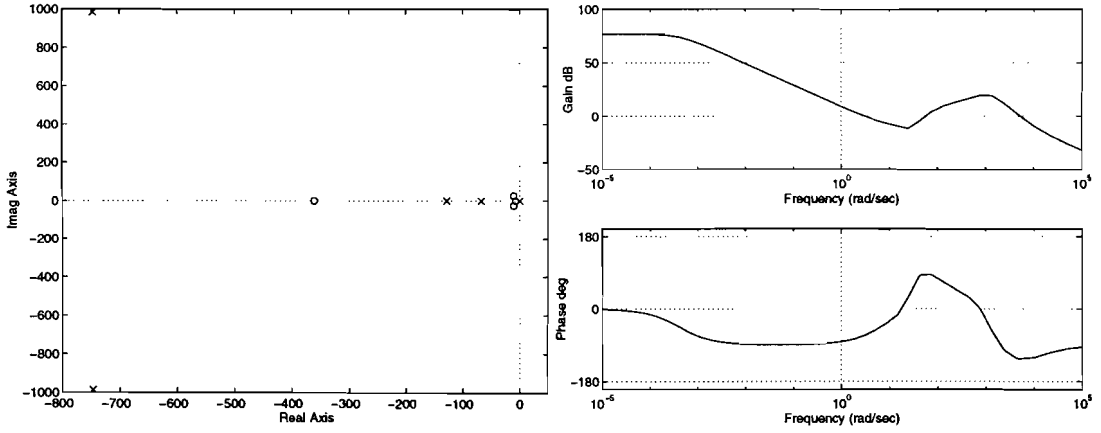


Figure 5-6: Pole zero map and Bode plot of controller

Again, the root locus of the closed loop system is investigated.

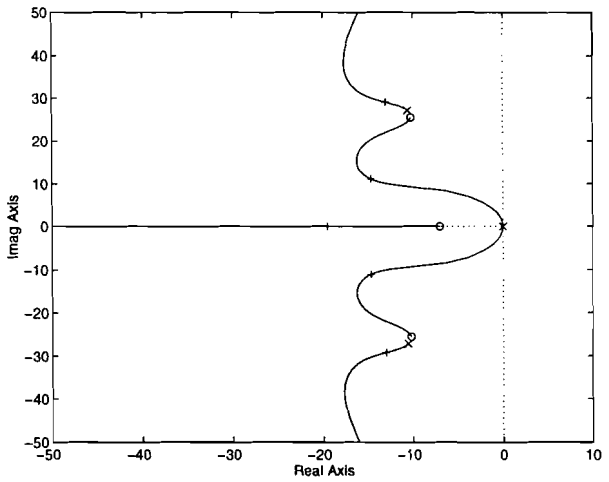


Figure 5-7: Root locus of closed loop system

The closed loop poles are again denoted by the '+' marks. The controller inserts an extra pole near the origin to give a small steady state error. This pole and the process pole in zero are attracted to the left by two zeros. The two complex conjugate process poles are again pulled towards the left half of the plane. The poles are however less well damped than in Figure 5-3, which will result in less attenuation of the distortion. Again, this is investigated by plotting the Bode magnitude plot of the closed loop transfer function from the disturbance input to the error output.

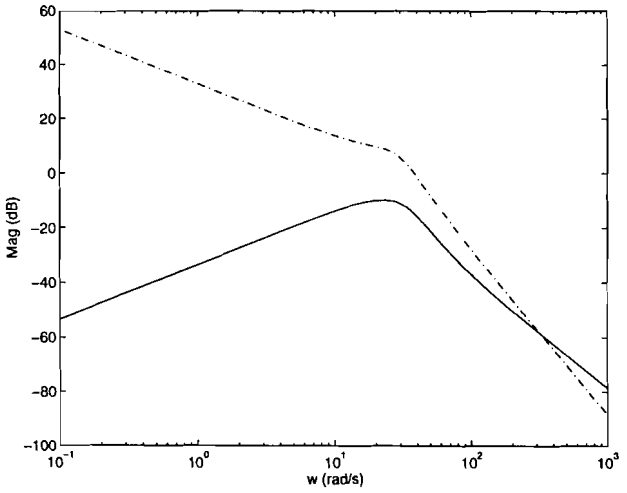


Figure 5-8: Bode Magnitude plot of closed loop system

The open loop process transfer is drawn as the dashed line. If this figure is compared to Figure 5-4, it is seen that a lot of attenuation had to be traded in. If this figure is compared to Figure 4-5 however, it is seen that a little attenuation is gained.
This is tested by applying the controller in a simulation as well as on the real system.

5.4 Evaluation of H_∞-controller

As this controller takes into account the actuator limitations it can not only be applied in simulation, but also on the real system. This is again done for various reference speeds. Pleas note that this controller uses the full encoder resolution and a sample frequency of 2 kHz. The results are drawn in Figure 5-9.

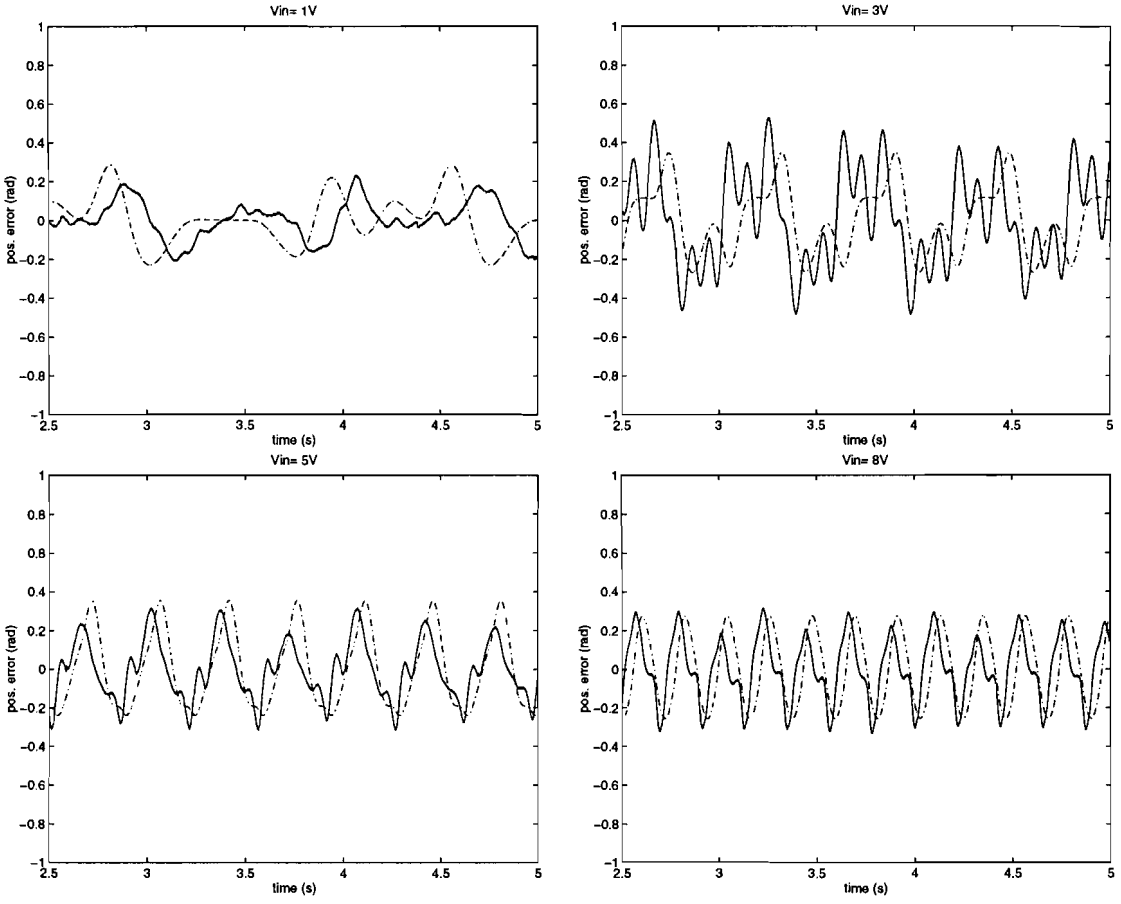


Figure 5-9: Real and simulated position error for H_∞ controller

If the real position errors of Figure 5-9 and Figure 4-8 are compared, it is seen that the H_∞ controller has a smaller position error, especially for low speeds. The H_∞ controller however is sixth order as opposed to the one order of the PI-controller. The question is whether the extra performance justifies the five extra controller states.
As the difference in performance is only small the influence of lower encoder resolutions was not investigated. Furthermore, a lower encoder resolution results in high frequency components on the position error signal. The H_∞ controller is more sensitive to this high frequency noise, which will probably lead to less damped behaviour.
If the alternative position error calculation of section 4.5 is used, this problem can be circumvented. However the position error signal then consists of a series of steps. These steps also have high frequency components, which will again lead to less damped behaviour.
Instead a controller based on a different control technique was designed. This is described in the next chapter.

One of the main goals of this chapter was to determine the maximum performance a controller can achieve given the practical limitations of the process. Although every design can always be improved it is save to say that it is not to be expected, that a controller using a lower encoder resolution will achieve smaller position errors, taking into account the actuator limitations..

5.5 References

[1]
Maciejowski, J.M.
MULTIVARIABLE FEEDBACK DESIGN
1989, Addison Wesley Publishing Co., Wokingham, England

6 Chapter six: Asynchronous control

The asynchronous control scheme described earlier in Chapter 1, is the subject of this chapter. This scheme is especially useful for low encoder resolutions as it makes full use of every position measurement. In section 4.5 already an asynchronous position error calculation algorithm was described. The used controller was still updated at a fixed sample rate.
The asynchronous controller that is to be designed in this chapter does not only calculate the position error asynchronous in time. It also updates its output value, whenever a position measurement becomes available. Now both measurement and control are asynchronous in time. As described in Chapter 1, this means that the error in the measurement of the position is zero at the instants that a new controller output has to be calculated.
The difficulty of designing an asynchronous controller is that all discrete control theory assumes that a fixed sample rate is used. This makes it impossible to use the well-known controller design techniques. To circumvent this problem the system description is transformed to a synchronous form. This is described in section 6.1. After this conversion, the 'standard' theory can be applied to find a controller. This is the subject of section 6.2.
The performance of this controller is then evaluated in section 6.3 and 6.4, by simulations and tests on the real system. This is done analogously to the tests carried out in Chapters 4 and 5. As these tests only cover the case were the reference frequency is constant and no load is applied to the master motor, some further tests will be carried out in Chapter 7. These tests concern start up, shut down and low frequency disturbance rejection.

6.1 System modelling

In section 4.5 an asynchronous method was used to update the calculated position error. The control signal was however updated at fixed time intervals. If the position of the motor is regarded as a state of the motor, the measurement algorithm can be regarded as an extremely simple observer of that state. The 'observer' predicts the value of this state, between measurements by keeping it constant. It can be said that the observer works asynchronously in time, but the controller works synchronously. In fact every control system, can be thought of as being divided in an observer and a controller part. The observer measures and estimates some states of the process. The controller calculates its output based on these states. The observer often consists of nothing more than a sensor. If it is assumed that the observer as well as the controller part can be either operate synchronously or asynchronously in time, four observer-controller configurations can be distinguished. In a matrix, this can be denoted as follows:

		Controller	
		synchronous	**asynchronous**
Observer	**synchronous**	Chapter 4+5 except 4.5	Not investigated
	asynchronous	Paragraph 4.5	Chapter 6

Figure 6-1: Classification of various observer-controller configurations

It can be said that a set-up with a synchronous controller and observer is the subject of 'classical control theory. The synchronous observer with asynchronous controller is not considered here. Due to the asynchronous nature of the position measurement, the configurations using an asynchronous observer are of special interest. An example of an asynchronous observer with a synchronous controller was already given in section 4.5. The controller that is to be developed in this chapter is of the fourth and final type. The controller as well as the observer part is asynchronous.
As classical controller design, only covers synchronous observers and controllers, it can not be used here without modification. A transformation is needed to describe the system as being synchronous.
The key idea behind this technique is that the sampling of the position error and the update of the control action is not synchronous in time, it is however synchronous in the position of the motor axis. The sampling takes place at fixed motor positions. The model of the system, which is a function of the independent variable, time (t) is therefore transformed into a function where the motor position (θ) is the independent variable. The model of the system is transformed from a description in the time domain to a description in the position domain.
The block scheme used to model the motor in Chapter 3, can be rewritten in the following state space description:

$$\begin{aligned}
\frac{d\theta(t)}{dt} &= \omega(t) \\
\frac{d\omega(t)}{dt} &= \frac{1}{J}\cdot T(t) - \frac{1}{J}\cdot d(t) - \frac{B}{J}\cdot\omega(t) \\
\frac{dT(t)}{dt} &= \frac{K_t \cdot K_f}{\tau}\cdot u(t) - \frac{K_t}{\tau}\cdot\omega(t) - \frac{1}{\tau}\cdot T(t)
\end{aligned} \tag{6.1}$$

Where θ(t), ω(t), T(t), u(t) and d(t) are all functions of the time t.
The systems output is the motor position θ:

$$y(t) = \theta(t) \tag{6.2}$$

The transformation from the time to the position domain can be made by noting that:

$$\frac{d\theta(t)}{dt} = \omega(t) \tag{6.3}$$

This is no more than the simple relationship between the angular speed and position of a mechanical body. As long as $\omega \neq 0$ it holds that:

$$\frac{dt(\theta)}{d\theta} = \frac{1}{\omega(\theta)} \tag{6.4}$$

if t is cons.dered a function of θ, instead of θ as a function of t. That is, there exists a 1-1 correspondence between the position of the motor θ and the time t as long as ω>0 or ω<0 rad/s. This correspondence is shown in Figure 6-2.

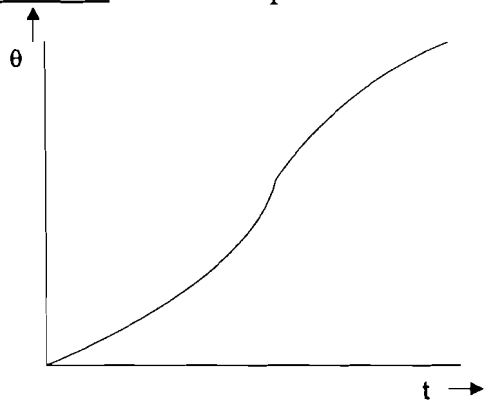


Figure 6-2: 1-1 Correspondence of the motor position θ and the time t

Now the linear differential equations given in Equation (6.1) can be transformed from the time to the position domain using the following:

$$\frac{df(\theta)}{d\theta} = \frac{df(t)}{dt} \cdot \frac{dt}{d\theta} = \frac{1}{\omega} \cdot \frac{df(t)}{dt} \tag{6.5}$$

The system description then becomes:.

$$\frac{dt(\theta)}{d\theta} = \frac{1}{\omega(\theta)}$$

$$\frac{d\omega(\theta)}{d\theta} = -\frac{B}{J} + \frac{1}{J \cdot \omega(\theta)} \cdot \left(T(\theta) - d(\theta)\right) \tag{6.6}$$

$$\frac{dT(\theta)}{d\theta} = \frac{1}{\tau \cdot \omega(\theta)} \cdot \left(K_t \cdot K_f \cdot u(\theta) - T(\theta)\right) - \frac{K_t}{\tau}$$

Note that the time t is now a state of the model, which is dependent on θ. The output of the model is the time.

$$y(\theta) = t(\theta) \tag{6.7}$$

Hence t(θ) is the time instant at which the motor is at position θ.

The position error between the master and the slave motor in the time domain can be expressed as a time error in the position domain. This is clarified by Figure 6-3.

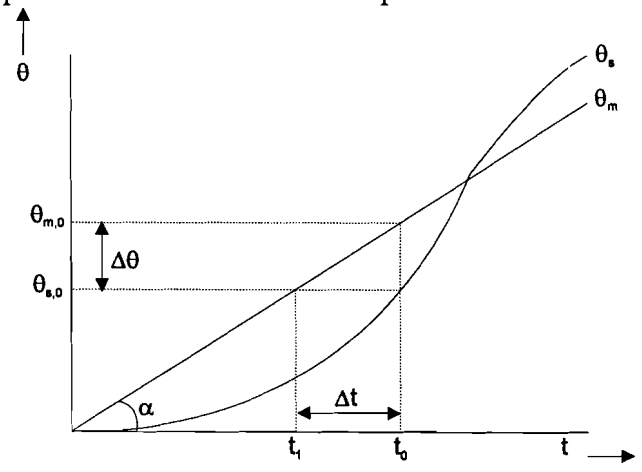


Figure 6-3: Master and slave position

In this figure, a position characteristic for the master (θ_m) and slave motor (θ_s) is drawn. Suppose that at time point $t=t_0$ the position error is measured. If the position of the master motor at that point in time is denoted by $\theta_{m,o}$ and that of the slave motor by $\theta_{s,0}$, the position error is given by:

$$\Delta\theta = \theta_{m,0} - \theta_{s,0} \tag{6.8}$$

This position error in the time domain can be translated into a time error Δt in the θ domain, by noting that the master motor arrived at the position $\theta_{s,0}$ at the time point t_1. To obtain a position error of zero, it has to be true that $\Delta\theta=0$. In the position domain this corresponds to $t_1=t_0$. It is demanded that the master and slave motor reach a given position at the same instant. Instead of minimizing the position difference between master and slave, the difference between the times that the master and slave reach a certain position is minimized. The equivalent of the position error in the time domain is now given by:

$$\Delta t = t_0 - t_1 \tag{6.9}$$

The objective of the control action is to minimize Δt. The controller transfer in the position domain is then given by:

$$u(\theta) = H_c(\theta) \cdot \Delta t(\theta) \tag{6.10}$$

If the master is ahead of the slave ($\Delta t>0$), the error can be measured. However, if the slave is ahead of the master ($\Delta t<0$) the time error is not known. Then the slave reaches a given position $\theta_{s,0}$ at a certain point in time t_0 and the master will reach this position at a later time point t_1. At what point in time the master is going to reach this position is not known at the instant t_0.

However, the time error can in both cases be calculated, if the velocity of the master motor is constant. Assume that it is equal to ω_r, then a position error in the time domain can be converted to a time error in the position domain using the following equation.

$$\frac{\Delta\theta}{\Delta t} = \omega_r \tag{6.11}$$

This property can easily be found from Figure 6-3, by noting that $\tan\alpha=\omega_r$ and looking at the triangle, with $\Delta\theta$ and Δt as its sides. From the next figure it should be clear, that the equation also holds for the case where the slave is ahead of the master.

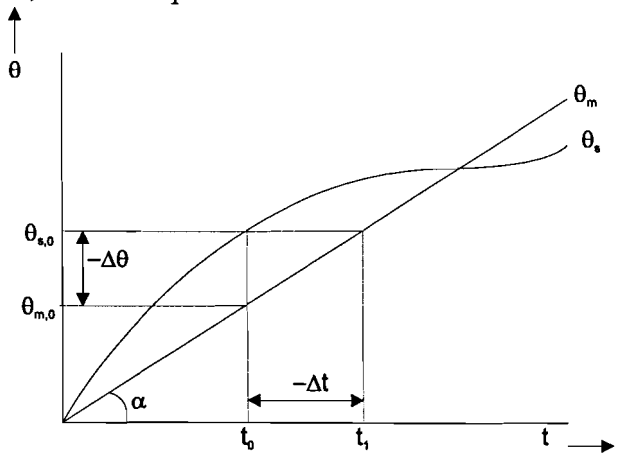


Figure 6-4: Master and slave motor position, slave ahead

From the above and Equation (6.11) it follows that the time error Δt can always be calculated if ω_r is constant and not zero. If ω_r is not constant, Equation (6.11) is an approximation.
In steady state operation the master motor speed will be constant and Equation (6.11) holds. The controller that is developed in the discrete position domain $H_c(z^*)$, can now easily be implemented in the time domain by using the following transformation. Note that z^* is used instead of z, to denote that $H_c(z^*)$ is the transfer function of the controller in the position domain instead of in the time domain.

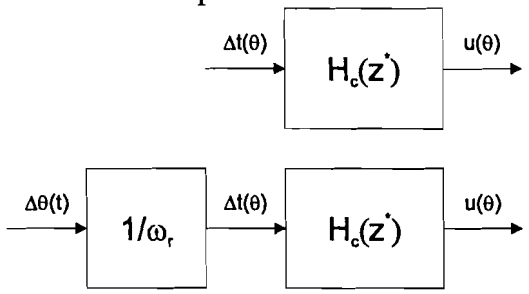


Figure 6-5: Controller transformation

As all the process transfers in this chapter are given in the position domain, the * is omitted from now on.
The necessary time error in the position domain is obtained by converting the measured position error using Equation (6.11). The value of ω_r can be measured by using the information coming from the master encoder. This calls for a speed observer, which can estimate the speed based on the position measurements of the encoder. Another and rather crude method is to derive ω_r, from the feed-forward signal coming from the master motor (see Figure 2-1). This method is crude, because the velocity of the master motor is not only dependent on the voltage applied to the master motor, but also on the load that the master motor has to drive.
The control problem that was asynchronous in the time domain is now converted into a synchronous control problem in the position domain. As the states of the model given in Equation (6.6) depend on $1/\omega(\theta)$, the system description in the position domain is nonlinear. The control of a nonlinear system is not as straightforward as the control of a linear system. When trying to control a nonlinear system there are a number of applicable solutions. These all have their own advantages and drawbacks as will be discussed below (see[1]).

1. Linearisation by compensation.
   Hereby the nonlinearity is compensated by its complementary function. This is however only possible in case of an essentially static nonlinearity. The nonlinearity found here is however essentially dynamic, as it depends on dynamic parameters. Therefore application of this technique is not possible.
2. Adaptive control.
   In the case that the plant dynamics are linear over limited time horizon but that they change over the long run, adaptive control can be applied. During the time span that the dynamics can be supposed to be linear and time invariant, the transfer function is estimated and a proper controller tuned for this transfer. This estimation and tuning is then continued on-line so that a slow change in the plant over time can be followed.

As the speed of the motor ω, will continually change between two measurement instants, the process that is under study can not be regarded as slowly varying.

3. Linearisation by feedback

   Under certain complex conditions, that can be verified if the plant dynamics are not too complicated, the dynamics can be made linear by means of nonlinear feedback. It can only be applied if the model is very good and if the sensor noise and disturbances can be neglected to a certain level.
   Although this method was not studied thoroughly, it is expected that to use this technique, quite a few states of the system have to be measured or estimated. In this control problem only the position (time) error is measured. All the other states of the process are unknown. Therefor a rather complex observer would have to be used to estimate the other states. The complexity of this observer is contradictory to the demand for a simple solution.

4. Nonlinear Control

   One can attack the nonlinear plants in their own nonlinear domain by nonlinear controllers. Soon, if the plant is somewhat complex, this very complex computations become necessary.
   Again, this method has not been extensively studied. The question is whether, these complex computations can be carried out at all whether this controller can be easily implemented.

5. 'Intelligent control'

   This method is based on finding a nonlinear dynamic controller by optimisation within a set of nonlinear, dynamic transfers. This set can be a neural network or a fuzzy logic based controller. The main problem here is the choice of the controller set and the optimisation: One never knows whether the set is big enough and the optimisation can stop in a local extremum.
   As the synthesis of an 'intelligent controller is rather tedious and the found controller can be difficult to implement, it is omitted here.

6. Linearisation

   If the signals just show small excursions about some average value, the obvious solution is to linearise the plant in the proper equilibrium or working point. The higher order, nonlinear effects are then supposed to remain relatively small and can be treated as model perturbations or considered as disturbances. This way the linearised plant can be controlled by classical PID-controllers or, if better quantisised, by robust control techniques as the H_∞ design method described in Chapter 5.
   As the signals of the motor model vary over a large range of values, linearisation does not seem applicable. However, the signals largely depend on the average motor speed. If the average motor speed is constant, the signals can be regarded as small excursions about this average motor speed value.
   The total control range is partitioned into subranges, each corresponding with a certain interval of the reference speed. For each of these subranges, a single linearisation can be developed. This is known as multiple linearisation. Different controllers can be found for different subranges. If the system switches from one subrange to the other, the controller should also switch. Therefore this can be regarded as a kind of adaptive control.

The process model given in Equation (6.6) is linearised around an equilibrium point. This equilibrium point and the first order excursions are given in Equation (6.12).

$$\omega = \omega_r + \tilde{\omega}$$

$$t = \frac{1}{\omega_r} \cdot \theta + \tilde{t}$$

$$d = \bar{d} + \tilde{d} \tag{6.12}$$

$$T = B \cdot \omega_r + \bar{d} + \tilde{T}$$

$$u = \frac{1}{K_f} \cdot \omega_r + \frac{B}{K_t \cdot K_f} \cdot \omega_r + \frac{1}{K_t \cdot K_f} \cdot \bar{d} + \tilde{u}$$

In Equation (6.12), setting $\tilde{\omega}$, $\tilde{t}$, $\tilde{d}$, $\tilde{T}$ and $\tilde{u}$ to zero, gives the equilibrium point. In this equilibrium point, the average motor speed is ω_r. The average disturbance torque is taken $\bar{d}$. The rest of the equilibrium points are found from the constraint that in the equilibrium point the right hand side of Equation (6.6) should be zero. For this equilibrium point, the following first order approximation can be derived.

$$\frac{d\tilde{t}}{d\theta} = -\frac{1}{\omega_r^2} \cdot \tilde{\omega}$$

$$\frac{d\tilde{\omega}}{d\theta} = -\frac{B}{J \cdot \omega_r} \cdot \tilde{\omega} + \frac{1}{J \cdot \omega_r} \cdot \tilde{T} - \frac{1}{J \cdot \omega_r} \cdot \tilde{d}$$

$$\frac{d\tilde{T}}{d\theta} = -\frac{K_t}{\tau \cdot \omega_r} \cdot \tilde{\omega} - \frac{1}{\tau \cdot \omega_r} \cdot \tilde{T} + \frac{K_t \cdot K_f}{\tau \cdot \omega_r} \cdot \tilde{u} \tag{6.13}$$

$$\tilde{y} = \tilde{t}$$

As the controller is designed using the root locus method, the characteristic poles of this system are investigated. The following poles were found:

$$p_1 = 0$$

$$p_{2,3} = -\frac{1}{2 \cdot \tau \cdot J} \cdot \frac{1}{\omega_r} \cdot \left(1 \pm i \cdot \sqrt{-(J - B \cdot \tau)^2 + 4 \cdot J \cdot \tau \cdot K_t}\right) \tag{6.14}$$

The location of two of the characteristic process poles are now dependent on the value of the reference speed ω_r. In the s-plane the characteristic poles all lie on the same line for varying ω_r. That is, they all have the same relative damping. The distance from the origin is however inverse proportional to ω_r. In Figure 6-6 the poles of the system are drawn as a function of ω_r.

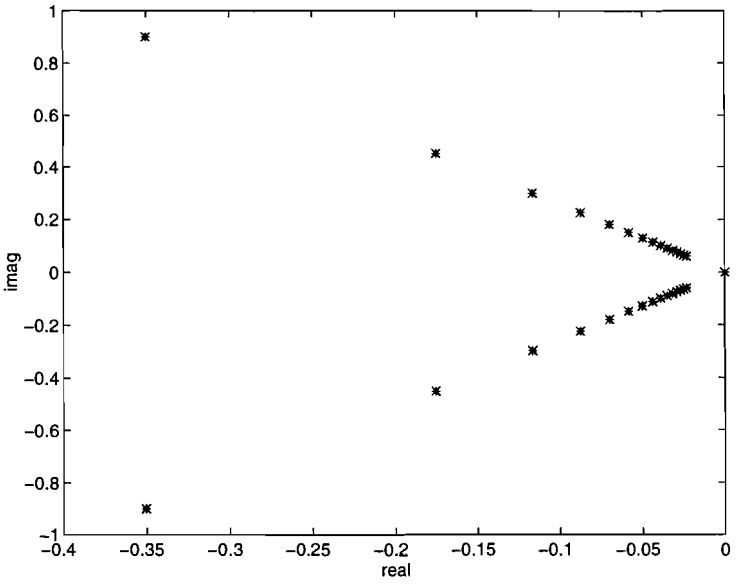


Figure 6-6: Poles of the process for varying ω_r

In this figure ω_r is varied from 30 to 460 rad/s, which is the maximum motor speed. The poles move from left to right for increasing ω_r. This is done in incremental steps of 30 rad/s. The distance of the two complex conjugate poles becomes smaller with increasing ω_r. The amount by which the poles move for a fixed absolute change of ω_r becomes smaller with increasing ω_r.
As the process is now synchronous and lineair and the poles of the process are known, a controller can be developed that places the poles in desirable locations.

6.2 Controller design

As the control action of the asynchronous controller takes place whenever the slave motor position reaches a certain value, it can be said that the controller is a discrete controller in the position domain. As the sample rate of the controller is not much larger than the highest process frequency, the controller has to be designed directly in the discrete domain.
Therefore the process described by Equation (6.13) is converted to its discrete equivalent. As the output of the controller is kept constant between measurement updates, the process can be thought of as being proceeded by a zero order hold function. In [2] it was argued that the discrete equivalent of the process and the zero order hold circuit can then be found by:

$$H_p(z) = (1 - z^{-1}) \cdot Z\left\{\frac{H_p(s)}{s}\right\} \tag{6.15}$$

where Z{H(s)} is the z-transform of H(s) and $H_p(s)$ is the process transfer in the position domain and $H_p(z)$ its discrete equivalent in the position domain.
The block scheme of the closed loop process plus controller in the position domain, is drawn in Figure 6-7.

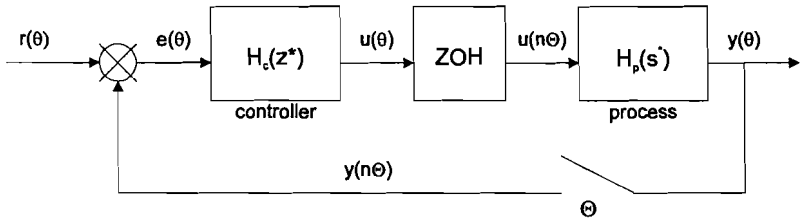


Figure 6-7: Block scheme of closed loop system in position domain

To keep the figure as simple as possible, the disturbance is omitted. Here the zero order hold is explicitly drawn, whereas in reality it will be incorporated in the controller. The output of the process y(θ) is sampled at discrete position intervals with fixed period Θ. This yields the discrete signal y(nΘ). This signal is compared to the reference r(θ), which comes from the master motor. Subtracting y(nΘ) from r(θ) gives the error signal e(θ), which was earlier described as the time error Δt.
Now the influence of varying the sampling period on the position on the discrete process poles is investigated. The open loop poles of the discrete process are drawn for $\Theta=2\cdot\pi$, π, $\pi/2$ and $\pi/4$ which corresponds to N=1, 2, 4 and 8 measurements per revolution of the motor axis respectively. The poles are drawn for various values of ω_r. This time the value of ω_r is iterated in very small steps, from values close to zero up to the maximum motor speed ω_r= 460 rad/s. Solid lines are drawn through the poles that were found. The zeros that are introduced by the discretisation are also drawn as the dashed lines.

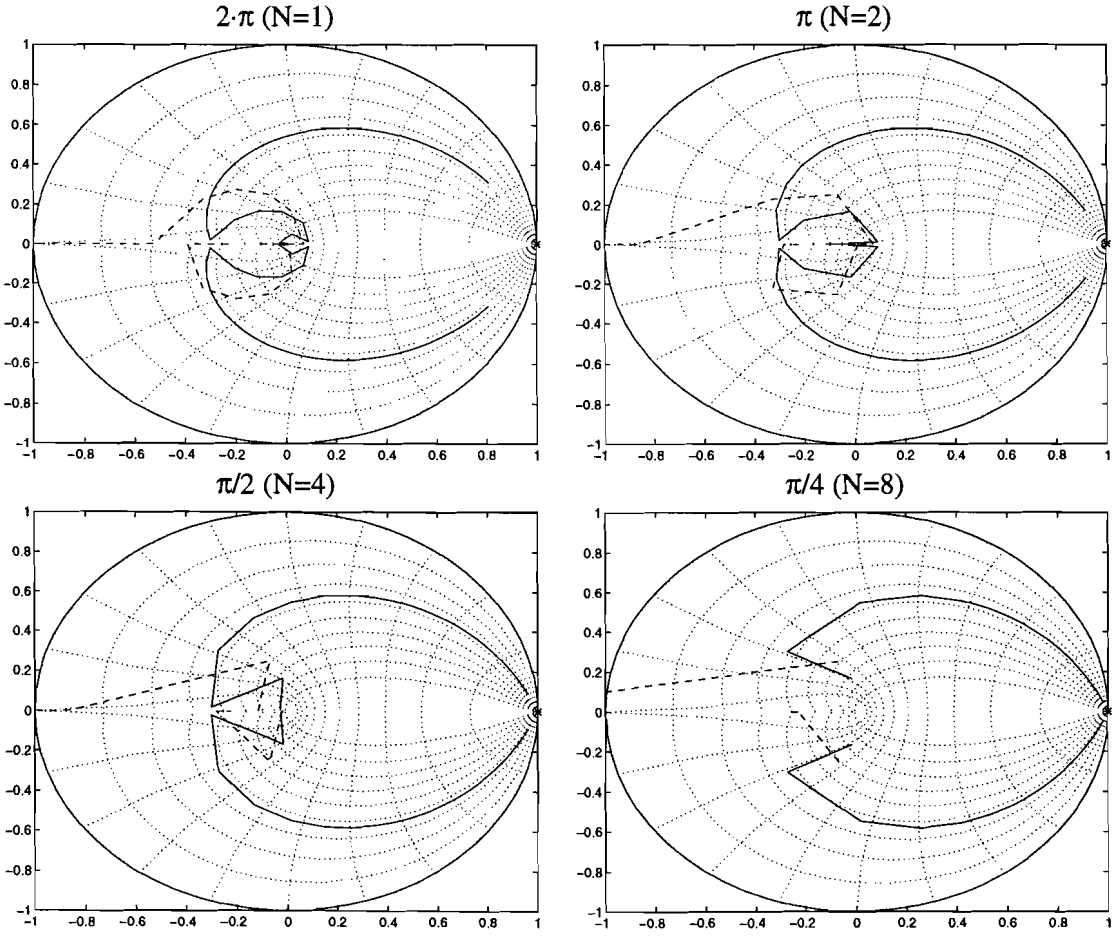


Figure 6-8: Process poles and zeros for varying sample-rate (N) and ω_r

The system consists of a pole i: z=1 and two complex conjugate poles, which are moving more and more towards the point z=1 with increasing ω_r. For larger values of ω_r the relative damping of the poles is constant. For very small values of ω_r, these poles are close to the origin of the z-plane. This corresponds to the fact that for ω_r=0, the poles go towards minus infinity in the s-plane. The two zeros, which are introduced by the discretisation, move from the origin towards the point z=-0.2 and z=-3 respectively.
If the sampling period Θ is decreased, the poles move more towards z=1 for a constant ω_r. In the discrete domain, this means that the poles become slower. The response of the system on an arbitrary input signal will take more samples. As Θ decreases the sample period also decreases. In absolute time the effect of a larger number of samples with a smaller sampling time is zero. This can be understood from the fact that physically the system does not change. Only the rate at which the output is sampled changes.
Now it is tried to find a discrete controller using the pole placement method. This is done for a sample period $\Theta=2\cdot\pi$ (n=1) and it is investigated, what the effect of altering the sampling period is. A problem in finding a controller for this system is the large dependence of the process poles on ω_r. This makes it almost impossible to find a fixed controller which gives satisfactory results for all ω_r. A possible solution is to make the controller dependent on ω_r also, thus neutralising the dependence of the process on ω_r. If the process poles and zeros are known it is possible to add some controller poles and zeros so that the poles of the closed loop system are placed at arbitrary locations. This is known as arbitrary pole placement. In [2] it was argued that this can also be done online. The position of the process poles is estimated based by using some measurements. The controller poles are then calculated online so that the closed loop poles stay at the same, desirable position. This is a sort of an adaptive control scheme. As the poles of this process largely depend on ω_r, it should be possible to estimate the location of the poles by measuring ω_r. As ω_r is the feed-forward signal of the master motor, this can easily be implemented. As the process model is of third order, in [2] it was argued that the controller should then be at least of second order. The controller should then be biproper, e.g. have two poles and two zeros. These two zeros and two poles have to be placed depending on the value of ω_r. For this, four equations with four unknowns have to be calculated online. This calls for quite a fast processor, preferably a DSP. This would however be too expensive
To get some feeling for the systems dynamics, the performance of a fixed controller is inv estigated. In Chapter 4 it was argued that a simple PI-controller gives quite satisfactory results in the continuous case. As the system described here, has essentially the same pole configuration, again the PI controller is investigated.
If the PI-controller is made discrete using an Euler approximation, the following transfer function can be found.

$$H_c(z) = K_p + \frac{K_i}{z-1} = \frac{K_p \cdot z + (K_i - K_p)}{z-1} = K_c \cdot \frac{z-a}{z-1} \qquad (6.16)$$

Note that the controller is rewritten in a form where, the location of the zero can be chosen by choosing a and the position on the root locus is determined by K_c. The discrete time PI controller has a pole in z=1 and a zero in z=a, where a is some real number. The zero of the PI-controller can be arbitrarily placed as in the continuous case. The zero is used to attract the process + controller pole in z=1. At first ω_r is set to 460 rad/s and a controller is designed. For this maximal value of ω_r the complex

conjugate process poles are closest to the point z=1 and will have the most influence on the loci of the poles in z=1. The zero of the controller was arbitrarily placed in z=0.9. A value for K_c was determined for which the system is still stable, that is for which all the closed loop poles are within the unity disk. After some iterations a value of K_c= 1.1 was found.
It was then investigated, where the poles of the closed loop system are located for other values of ω_r. This yields the following figure.

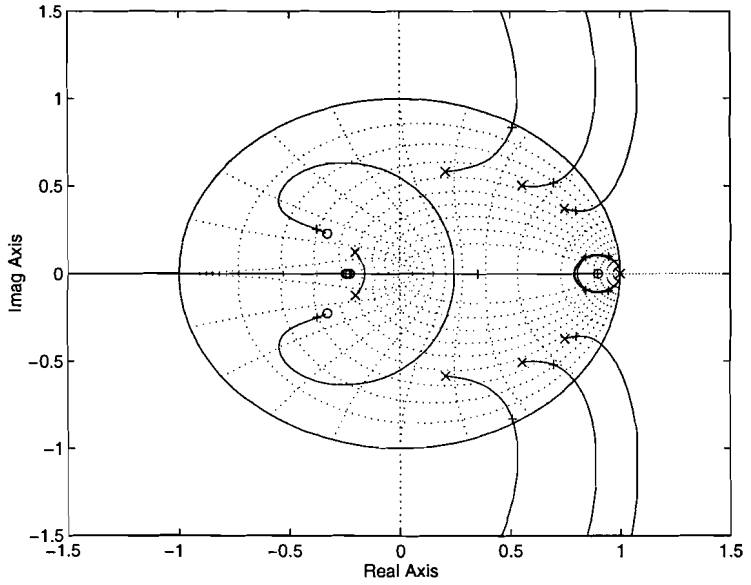


Figure 6-9: Root loci for varying ω_r and constant controller

In this figure the root loci are drawn for four values of ω_r. These four values correspond with a voltage of 1,3,5 and 8 volts on the input of the master frequency converter. These voltages were used, because they correspond to the investigations in Chapter 4 and 5, thus allowing for easy comparison. The loci that are more to the right in this picture correspond to higher reference speeds. The closed loop poles corresponding to K_c=1.1 are denoted by '+'-marks. Now regard the complex conjugate poles of the original process. With decreasing ω_r the amount by which the closed loop roots are shifted from their original value increases. The relative damping of these poles decreases with decreasing ω_r. From further investigations, it was seen that for some intermediate values of ω_r, the poles are outside the unit disk. Then the system is no longer stable. As the value of the overall controller gain K_c determines the amount by which the poles travel over the loci, instability of the system can be circumvented by choosing K_c sufficiently small. It can however not be prevented, that the relative damping changes.
It is tried whether a dependency of K_c on ω_r can reduce this effect. K_c should be smaller for smaller values of ω_r. Therefore K_c was made linearly dependent on ω_r. To make a fair comparison, the value of K_c was kept equal to 1.1 for ω_r=460 rad/s. This yields the following equation for K_c:

$$K_c = 1.1 \cdot \frac{\omega_r}{460} \tag{6.17}$$

Now the same figure as above was drawn for this set-up.

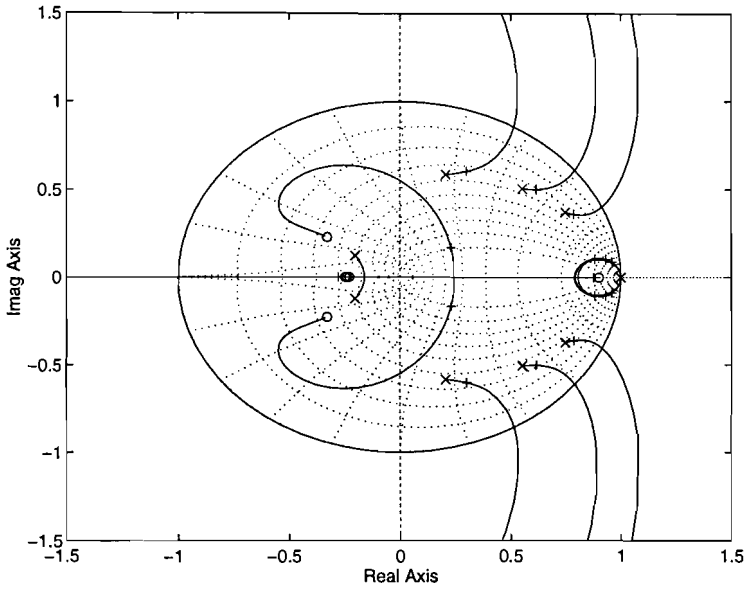


Figure 6-10: Root loci for varying ω_r with varying controller

The damping of the complex conjugate poles (denoted by a '+' in the figure) is now almost constant and the system does not become unstable for small ω_r anymore. Thus stability is ensured for all values of ω_r. The results presented here where all obtained using numeric calculation. To find out what the exact position of the poles is for varying ω_r, it was tried to find the location of the poles using an algebraic calculation package Maple. However, Maple was not able to find the roots of a fourth order polynomial. The location of the poles could not be determined algebraically. Therefore this phenomenon was studied using numerical calculations. In order to get some idea of the 'robustness' of the controller, the location of the zero and the value of K_c was varied. Although the location of the closed loop poles changes, the relative damping of the complex conjugate poles stays more or less the same for varying ω_r. Thus a 'robust' controller, which is stable for all ω_r can be designed.
Another important advantage lies in the implementation of the ω_r varying controller. The fact that the controller is dependent on ω_r, seems to make it somewhat harder to implement. As ω_r is proportional to the feed-forward signal coming from the master motor, ω_r is readily available. Disturbances on ω_r, however will also have effect on the controller performance. Now reconsider Figure 6-5. There it was argued that the controller designed in the position domain can be applied in the real system by converting the position error in the time domain to a time error in the position domain. This was done by multiplying the position error with the inverse of the reference speed $1/\omega_r$. The controller that is found here, depends linearly on ω_r.
In formula:

$$H_c(z) = \omega_r \cdot H_c^*(z) \tag{6.18}$$

If this controller is substituted in Figure 6-5, ω_r cancels in the equations and the following block scheme is found.

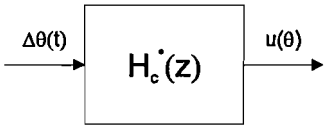


Figure 6-11: Implementation of controller

In the practical implementation, the controller is not dependent on ω_r anymore. This ensures a very simple implementation. Using this property and combining Equations (6.16) ,(6.17) and the fact that the pole of the controller was placed in z=0.9 leads to the following description of the controller:

$$H_c^*(z) = \frac{1.1}{460} \cdot \frac{z-0.9}{z-1} \tag{6.19}$$

The performance of this controller is evaluated in the next section.

6.3 Performance evaluation

Now the controller performance is evaluated by drawing the Bode amplitude plots of the transfer function from the disturbance d to the systems output y. The controller is discrete and contains a zero order hold and a sampler (see Figure 6-7). Therefore drawing the Bode diagram of this set-up is not as straightforward as when using a continuous controller.

In [3] a technique is described, to obtain the frequency response of a continuous system, which is controlled by a discrete controller. The transfer of the sampler is given by:

$$H_{sam}(s) = \frac{1}{h} \tag{6.20}$$

Where h is the sample time. The sample time was defined as Θ earlier on. The transfer of the zero order hold is:

$$H_{zoh}(s) = \frac{1-\exp(-s \cdot h)}{s} \tag{6.21}$$

Note that the following holds for the discrete time controller:

$$H_c(z) = H_c(e^{sh}) \quad \text{as } z = e^{sh} \tag{6.22}$$

Using these transfers and adding the disturbance input to the block scheme of Figure 6-7 yields the following.

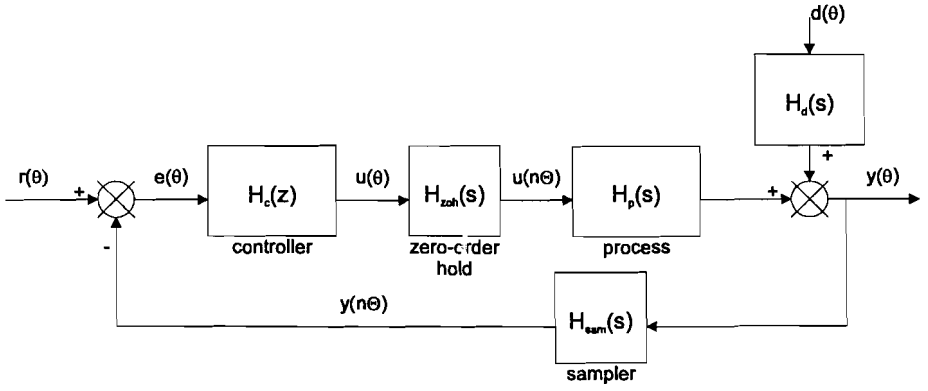


Figure 6-12: Closed loop system with added sample and hold circuits

$H_d(s)$ is the open loop transfer from the disturbance to the process output. Note that all signals and transfers are still given in the position domain.

The closed loop transfer function from the disturbance d to the output y is given by:

$$H_{dy}(s) = \frac{H_d(s)}{1 + H_{sam}(s) \cdot H_c(e^{sh}) \cdot H_{zoh}(s) \cdot H_p(s)} \tag{6.23}$$

This transfer function is given in the position domain. Therefore, the disturbance suppression can be plotted for frequencies relative to the motor velocity. The Bode magnitude plots are standardized, such that one revolution of the motor axis agrees with a value of one (10^0) on the x-axis. The torque of the sheet-feeder has a frequency relative to the motor axis speed given by $1/i_{gear}$, where i_{gear} is the gear ratio. In the test set-up i_{gear}=12.5, so that the first harmonic of the sheet feeder torque lies at the point on the x-axis $1/12.5=8\cdot10^{-2}$. As argued earlier, the tenth harmonic of the sheet feeder torque signal is considered relevant. The frequencies over which the torque ranges are denoted in Figure 6-13 by two dashed, vertical lines.
The Bode plot of the controller that was derived in the previous section is drawn. For this the same values of ω_r are taken as those in the root -locus of Figure 6-10. The dashed lines in this figure are the Bode plots of the open loop system for various values of ω_r. The Matlab .m file that generates these plots can be found in Appendix B.

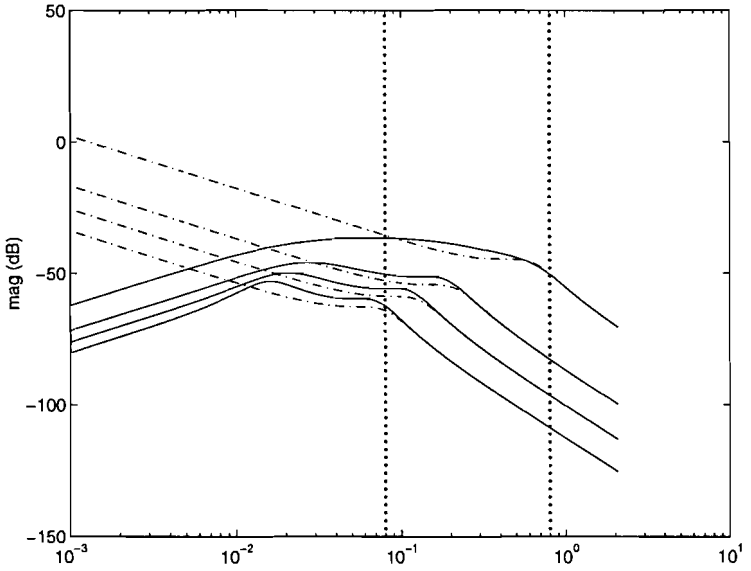


Figure 6-13: Bode plot of closed and open-loop system, for various values of ω_r

Drawn from top to bottom are the closed (solid lines) and open loop (dashed lines) magnitude plots complying with the reference voltages V_{in}=1,3, 5 and 8 respectively. Again the controller has the most influence for low frequencies. In general, the smaller the reference speed, the smaller the disturbance rejection.
Now simulations and measurements on the real system were carried out. Again the reference voltages of V_{in}=1V, 3V, 5V and 8V were used. The results are given in Figure 6-14.

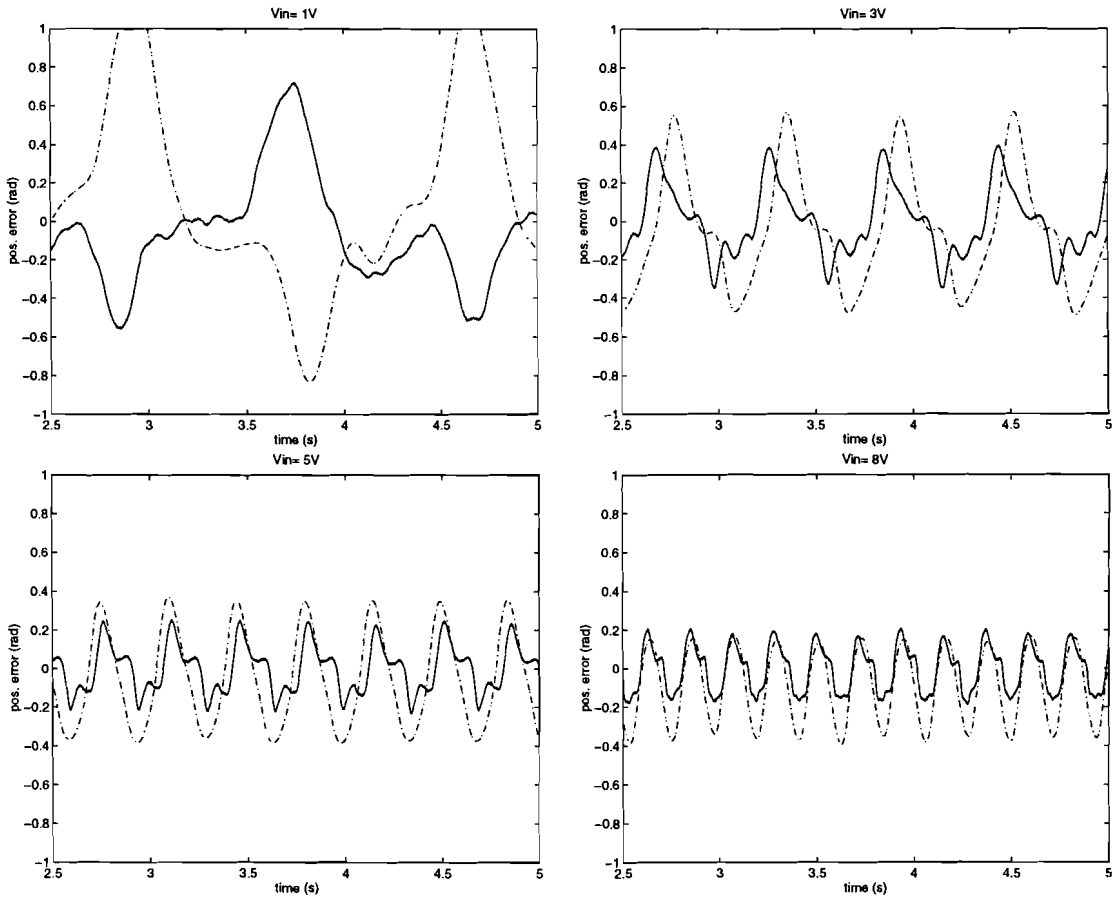


Figure 6-14: Position error for asynchronous control using 1 pulse per revolution

The solid lines are the position errors measured on the real system and the dashed lines are the simulation results. As expected from the Bode plots the disturbance rejection, is better for higher speeds. Then, the maximum position error is smaller.

6.4 Influence of encoder resolution

Now, the influence of changing the encoder resolution is investigated. As may be clear from the previous section, the disturbance rejection is the worst for low motor velocities. Therefore the influence of altering the number of pulses is investigated for a low motor speed. The reference voltage to the master motor is kept constant at 1V. To determine only the influence of a different encoder resolution, the controller is kept constant. First the root loci are investigated.
In Figure 6-15, the root loci of the closed loop system are drawn for sample periods $\Theta=2\cdot\pi$, π, $\pi/2$ and $\pi/4$. This corresponds to encoder resolutions of respectively N=1, 2, 4 and 8 measurements per revolution of the motor axis.

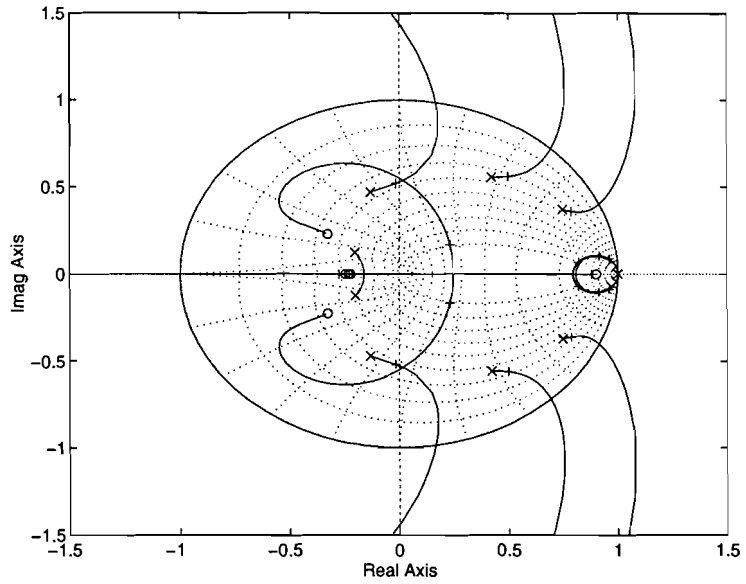


Figure 6-15: Root loci for various encoder resolutions

Moving from left to right in this figure, the loci for higher encoder resolutions are found. The effect of increasing the encoder resolution is the same as increasing the reference speed when looking at the tendency of the root loci(see Figure 6-10). For higher resolutions, the loci move towards z=1. This can be explained from the fact that the starting points of the loci, e.g. the poles of the open loop process, move towards z=1 with increasing encoder resolution. This was already seen in section 6.2. There, it was argued that the net effect of these movements in the continuous domain is zero, as the sample period increases also. Now the net effect of increasing the encoder resolution on the disturbance rejection is investigated. This is done by drawing the Bode amplitude diagram for the various values of the encoder resolution.

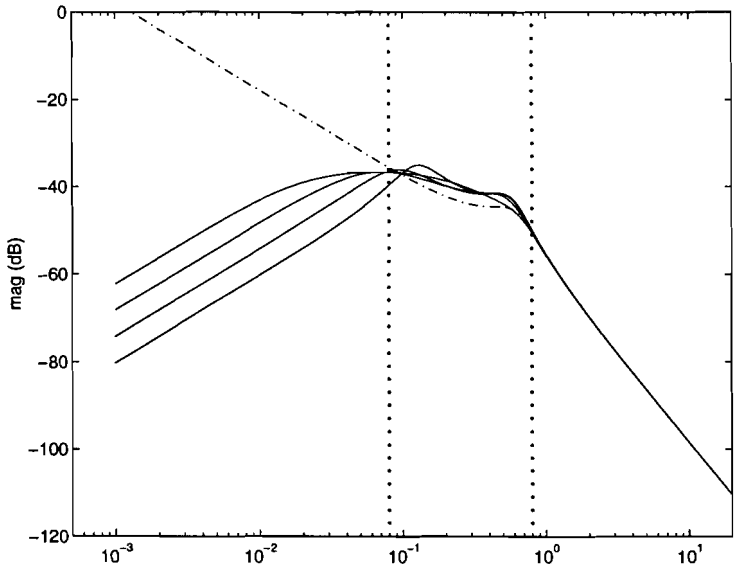


Figure 6-16: Bode diagram of disturbance rejection, various encoder resolutions

From this figure can be seen that with increasing resolution, the rejection of low frequency disturbances increases. As the PI-controller only influences the behaviour for low frequencies, for high frequencies the open loop transfer is applicable for all

encoder resolutions. This open loop transfer is drawn as the dashed line. The frequencies of the sheet feeder disturbance, which ranges from $8 \cdot 10^{-3}$ to $8 \cdot 10^{-2}$ rad/s are denoted by the two vertical lines. For these frequencies there is little difference is seen between the different encoder resolutions and the open loop transfer.
The disturbance rejection over time is now investigated both in simulation and on the real process. The results are given in Figure 6-17.

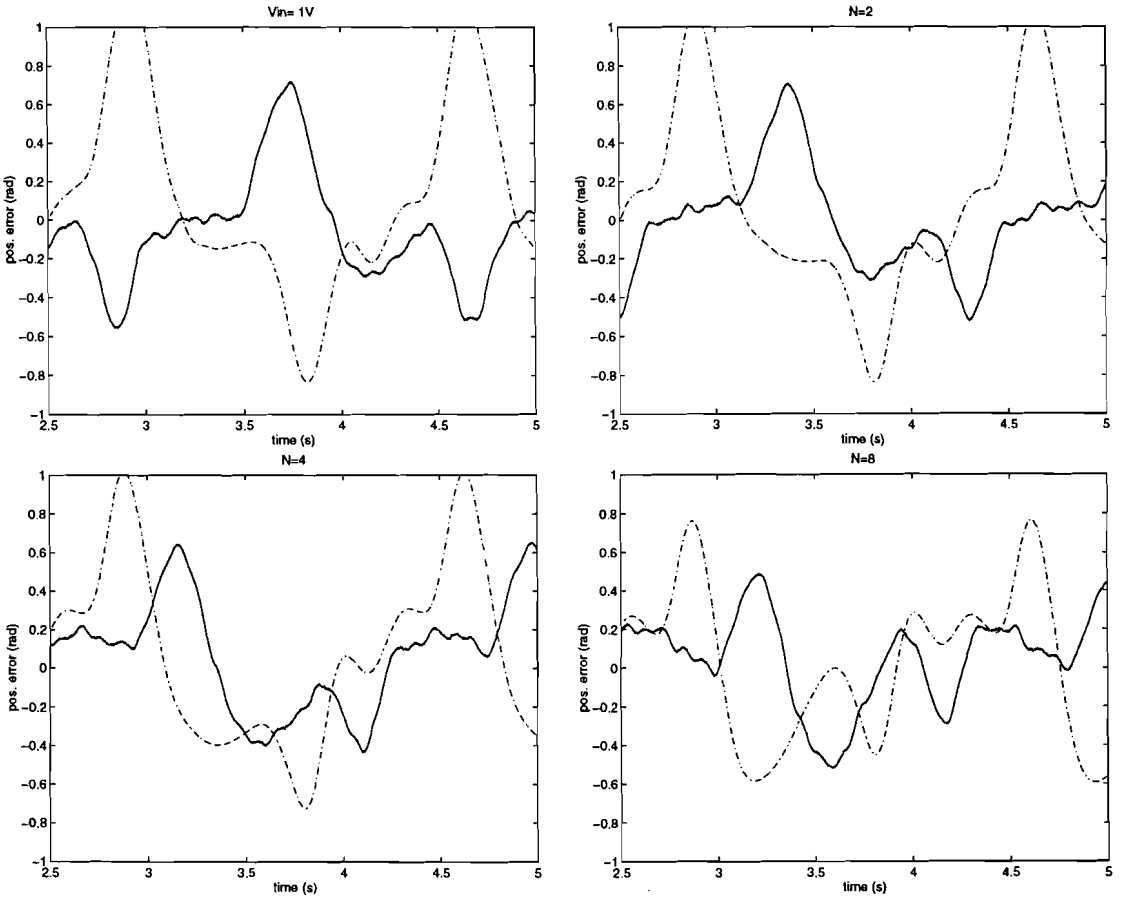


Figure 6-17: Real and simulated position error for various encoder resolutions

The simulation results are given by the dashed lines. The real results are drawn in solid lines. It is seen that the maximum position error is getting lower for higher encoder resolutions. This influence is however very small.
Up till now, the controller performance has only been investigated for the case where the master motor speed is constant. In the next chapter a series of measurements is carried out on the real system, to evaluate the performance of the controller under varying conditions.

6.5 Extra note

Whilst carrying out the experiments in Chapter 7, it was found out that for higher encoder resolutions the developed controller becomes unstable at high speeds. This lead to the investigation described in this section. Due to the fact that the test set-up was no longer available, the controller that is to be developed in this section, could not be tested on the real system.

The fact that the system b(comes unstable at high motor speeds for higher encoder resolutions can be explained as follows. In Figure 6-15, it was shown that the complex conjugate poles of the original process move towards z=1 for increasing encoder resolutions. In Figure 6-10, it was shown that the same effect occurs for increasing values of the reference speed ω_r. If a high encoder resolution is used and the reference speed is high, the effect is doubled. The complex conjugate poles can then come very close to the point z=1. It can now occur that the complex conjugate poles are attracted by the zero of the controller. The two poles in z=1 are then no longer attracted into the unity disc and become unstable. This is illustrated by Figure 6-18.

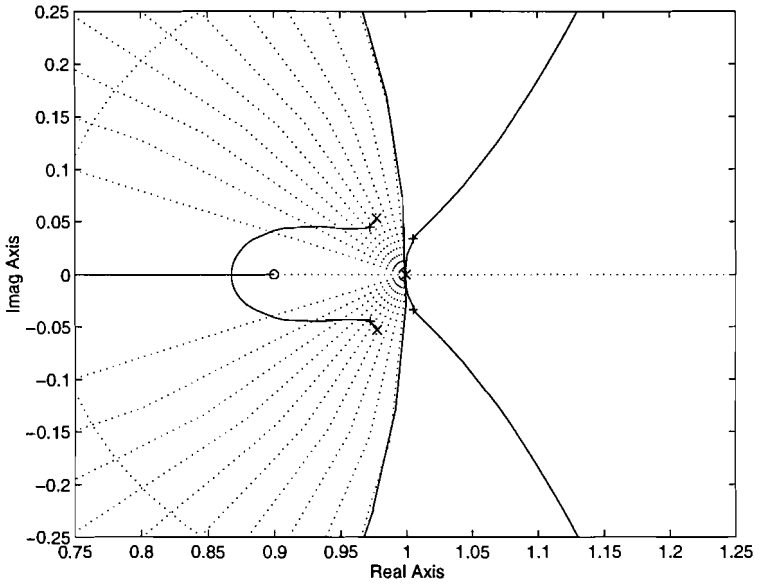


Figure 6-18: Root locus for N=8 and V_{in}=10 Volts

In this figure, the root locus is drawn for a system using an encoder resolution of 8 measurements per revolution of the motor axis. For V_{in} a value of 8 Volts is chosen, which corresponds to the highest possible motor speed. In this figure, it is clearly seen that the two poles in z=1 are no longer attracted by the zero and become unstable.
To solve this problem, the zero of the controller has to be moved more towards z=1. Then the zero is closer to the poles in z=1 then it is to the complex conjugate poles and the poles in z=1 will be attracted by the zero.
By varying the location of the zero with varying encoder resolution it is tried to find a controller that is stable for all resolutions. The location of the zero is varied with the encoder resolution using the following equation:

$$a = 1 - 0.1 \cdot \frac{\Theta}{2 \cdot \pi} \tag{6.23}$$

where a is the location of the zero and Θ is again the value if the sample frequency in the position domain (see Figure 6-7). Then for an encoder resolution of one pulse per revolution ($\Theta=2\cdot\pi$), the zero still lies at z=0.9. For higher encoder resolutions (smaller values of Θ) the zero moves more towards z=1. Then the following root-loci can be found for V_{in}=10 V and encoder resolutions of 1, 2, 4 and 8 pulses per revolution of the slave axis.

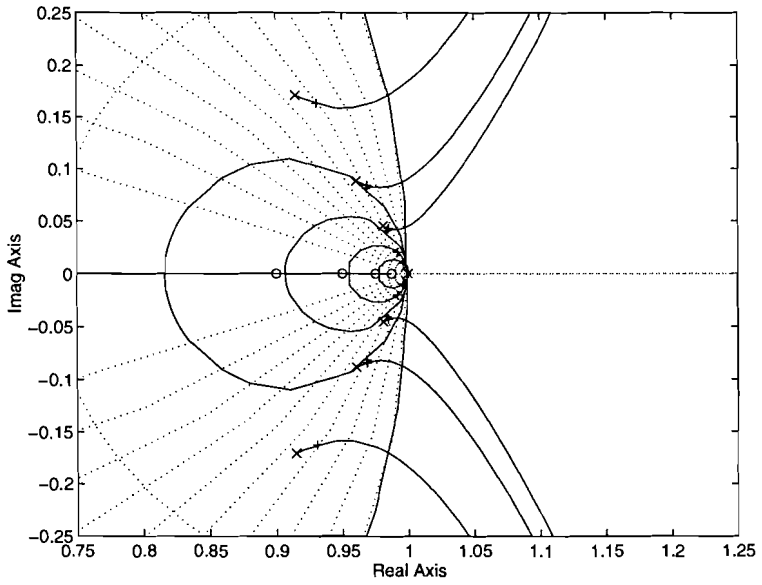


Figure 6-19: Root loci for varying encoder resolution with varying zero location

The loci with smaller circles correspond to higher encoder resolutions. It is seen that for all encoder resolutions the system remains stable. The Bode amplitude plot of the disturbance to output transfer is plotted for the same cases as above.

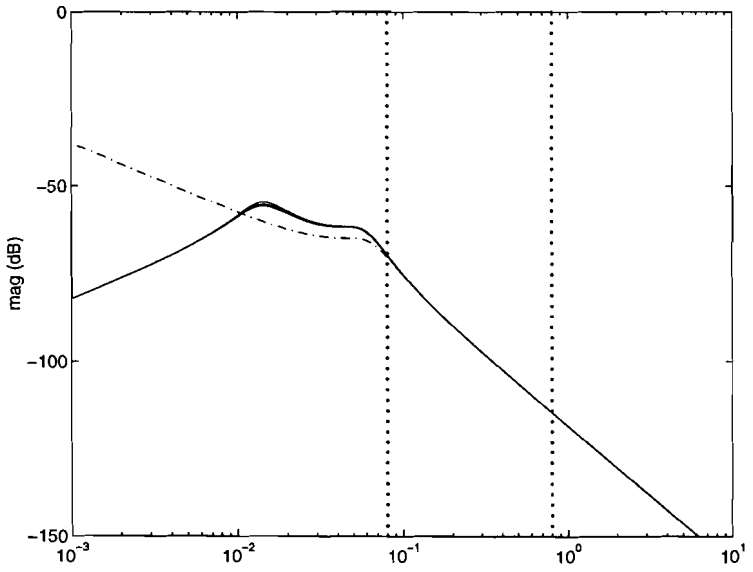


Figure 6-20: Bode plot of disturbance attenuation for varying zero location

It is seen, that for this speed the transfers from the distortion to the output is (almost) similar for all encoder resolutions. For lower speeds, there is more variation in the location of the process poles and thus more variation in the transfer. This is verified by plotting the Bode plots for V_{in}=1V and varying encoder resolutions.

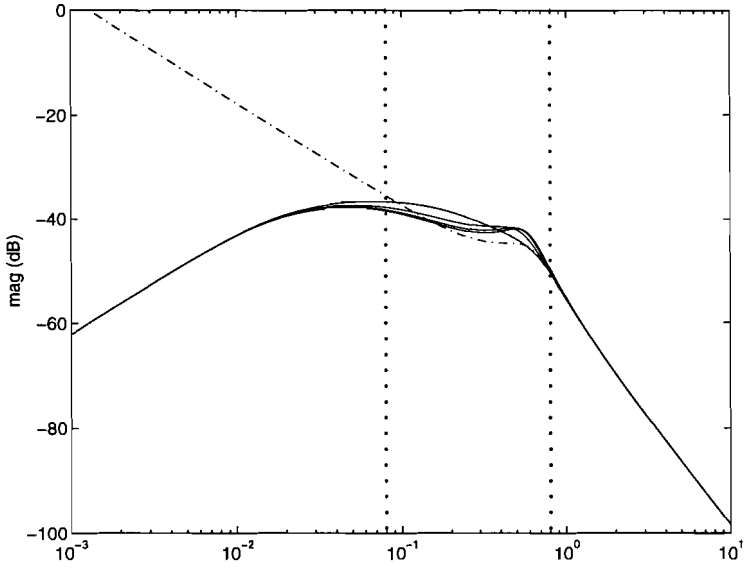


Figure 6-21: Bode plot of disturbance rejection for V_{in}=1 V, varying zero location

As said earlier this controller could not be tested on the real system, because the test set-up was no longer available.

6.6 References

[1]

Damen, A.A.H.
INTRODUCTIE REGELAARS ONTWERPEN
Dictaatnr.: 5790, Technische Universiteit Eindhoven, November 1995

[2]

Franklin, G.F.; J.D. Powell; M.L. Workman
DIGITAL CONTROL OF DYNAMIC SYSTEMS
Addison-Wesley publishing company, 1990

[3]

Åström, K.J.; B. Wittenmark
COMPUTER CONTROLLED SYSTEMS: THEORY AND DESIGN
Prentice-Hall International, Inc., Eaglewood Cliffs, N.J., 1984

7 Chapter seven: Tests

In this chapter some tests are carried out on the real system. These tests involve starting up and shutting down the machine, altering the phase between the master and slave motor and application of a load to the slave motor.
In this chapter, two possible controllers are used and the results are compared. The first controller, is the asynchronous controller developed in Chapter 6. The other is the hybrid controller, described in section 4.5. This controller uses an asynchronous method for calculating the position error . The control action, however is updated synchronous in time, with a frequency of 2 kHz.
In the next chapter, the conclusions that can be derived from this and earlier chapters are described and some recommendations are given.

7.1 Start up

First the master and slave motor are started up from stand-still, to the full production rate. For the Buhrs-Zaandam mailing machine, the maximal production rate is currently 18,000 products per hour. This corresponds to 18,000 revolutions per hour of the load axis. The reference voltage needed to obtain this speed can easily be calculated as follows.

$$V_{ref} = \frac{18{,}000}{3600} \cdot 2 \cdot \pi \cdot \frac{i_{gear}}{K_f} \tag{7.1}$$

The load axis speed in revolutions per hour is devided by 3600 to find the load axis speed in revolutions per second. This speed can then be converted into radians per second. As the load axis is connected to the motor via a gear box, the motor axis should be i_{gear} times as fast as the load axis speed. Here i_{gear}, is the gear ratio. The motor speed is converted to the voltage that has to be applied to the frequency converter. Therefore the motor speed is divided by the frequency converter constant K_f. For the test set up, a value of 8.5 Volts was found for V_{ref}.
In chapter 3, it was stated that the frequency converter limits the maximum and minimum rate at which its output frequency changes. It was stated that this maximum (minimum) corresponds to 5 V/s (-5 V/s) on the reference voltage applied to the frequency converter. If the slave motor gets behind on the master, the slave motor should be able to accelerate in order to catch up with the master. If the master accelerates at the maximum rate, this is not possible. Therefore, the reference voltage applied to the master frequency converter, is limited to a rate of change of 2.5 V/s (-2.5 V/s). This leaves some room to increase the acceleration of the slave so that it can catch up with the master.
In Figure 7-1, the results are given for starting up the test system. Here the asynchronous controller, developed in chapter 6 was used. The encoder resolution was varied from 1, 2, 4 and 8 pulses per revolution of the motor axis.

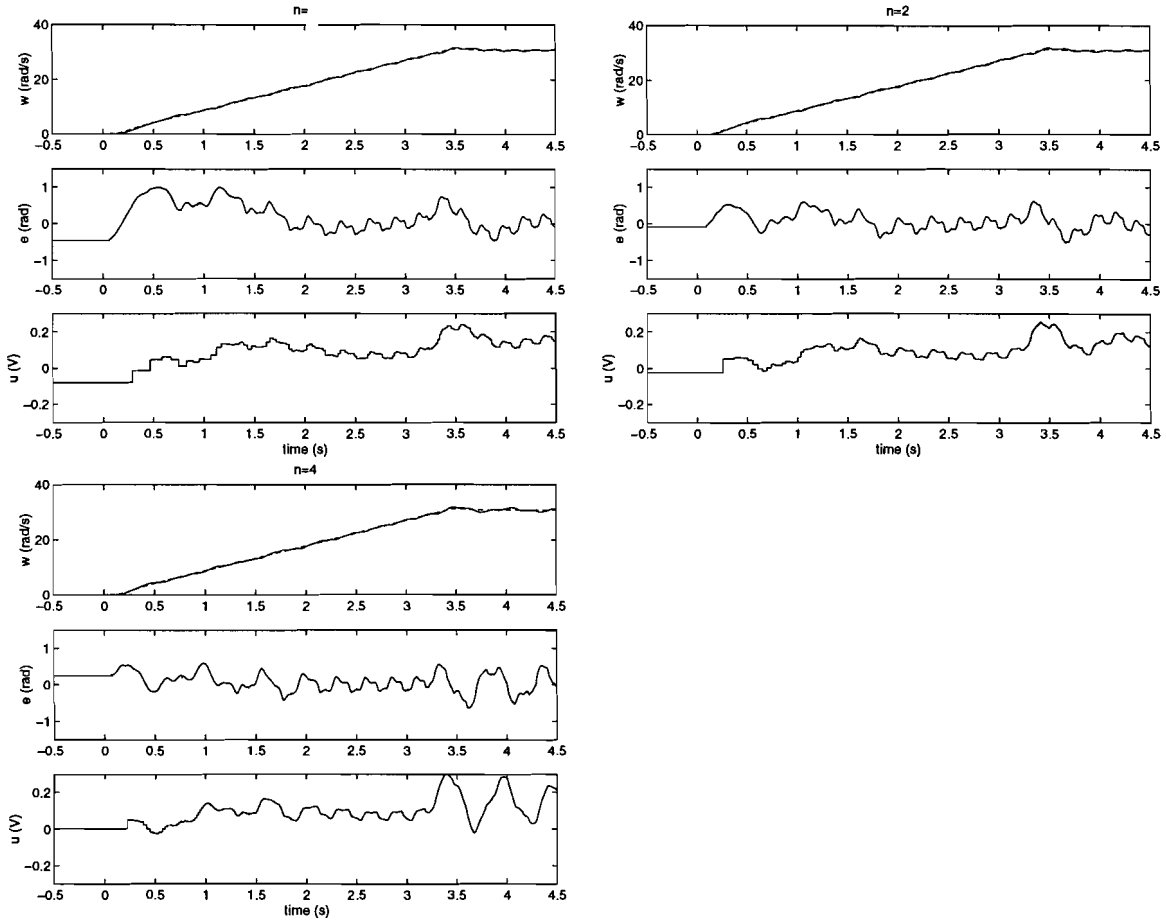


Figure 7-1: Start up for asynchronous controller

On the top axis, the master and slave motor speed are given. On the middle axis, the position error (e) is given in radians. Below is the control action in Volts coming from the slave motor controller.

It was found that for an encoder resolution of 8 pulses per revolution, the system becomes unstable at high speeds. Why this happens and a possible remedy were already given in section 6.5. Due to the fact that the test set-up was no longer available, it was not possible to test this remedy. Therefore no further tests are carried out using the asynchronous controller with this resolution.

Now the same tests were carried out, using the hybrid controller derived in section 4.5.

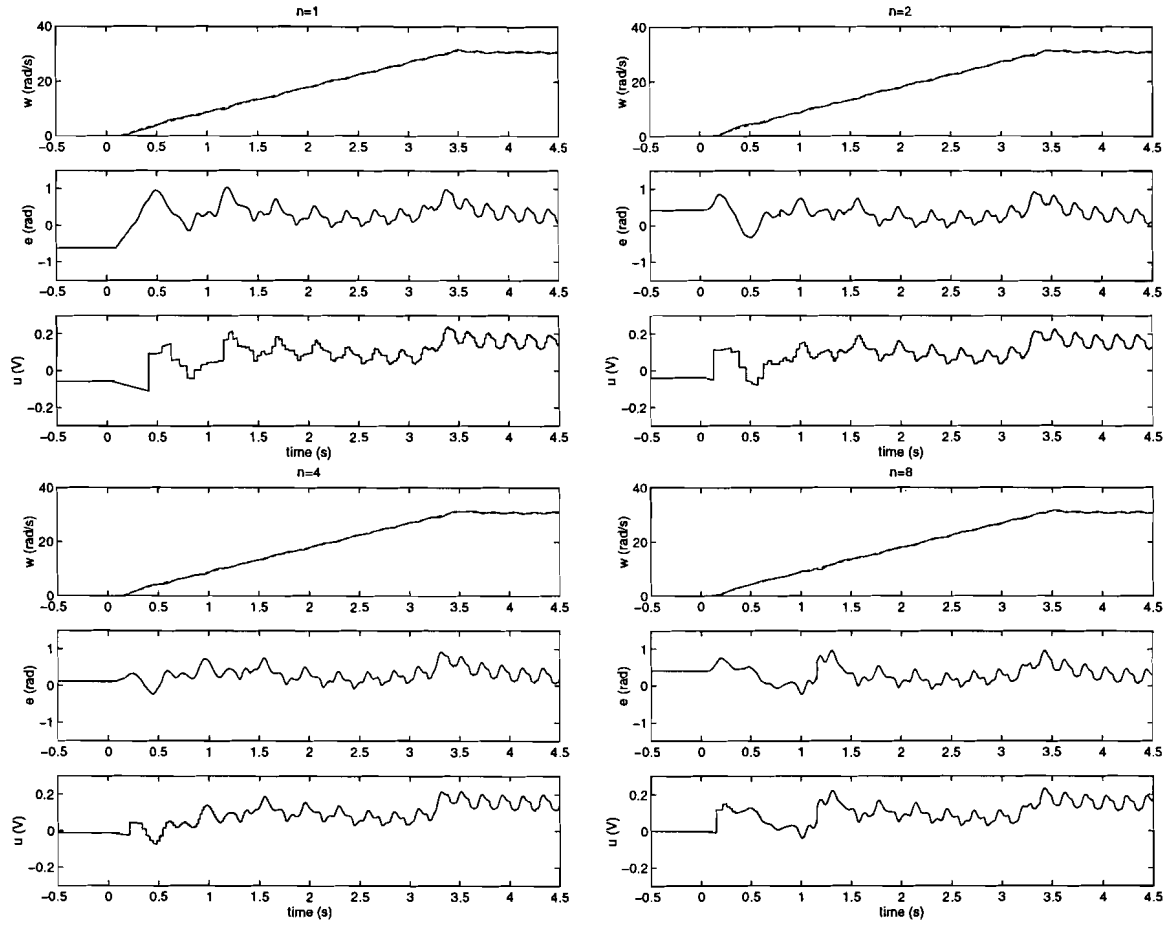


Figure 7-2: Start up for hybrid controller

The hybrid controller does not become unstable for higher encoder resolutions.

7.2 Shut down

Now the reverse of the above was done, the system was brought from its nominal production rate to stand still. Again the maximum rate of change of the reference voltage was limited.
These are the results for the asynchronous controller.

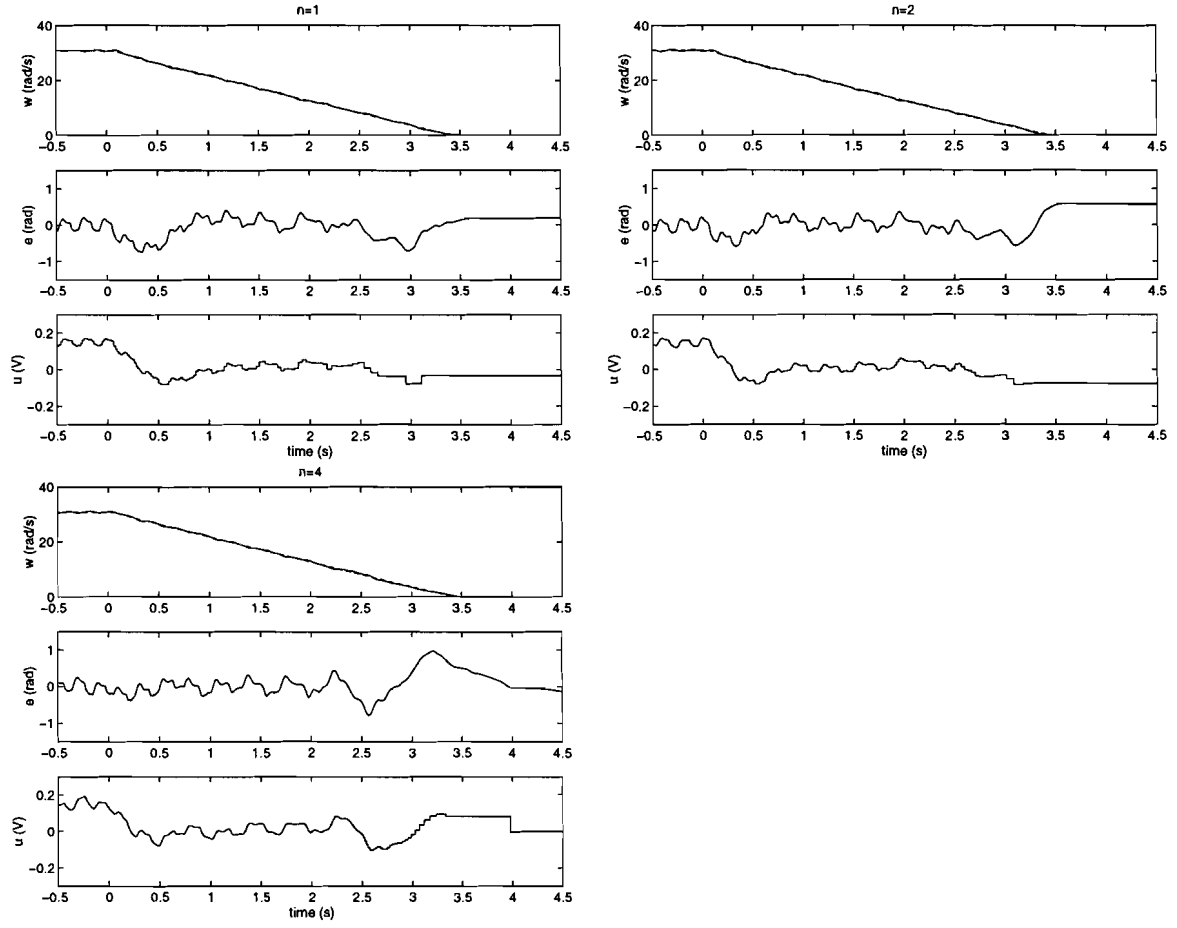


Figure 7-3: Shut down using asynchronous controller

The hybrid controller derived in section 4.5 leads to the following results.

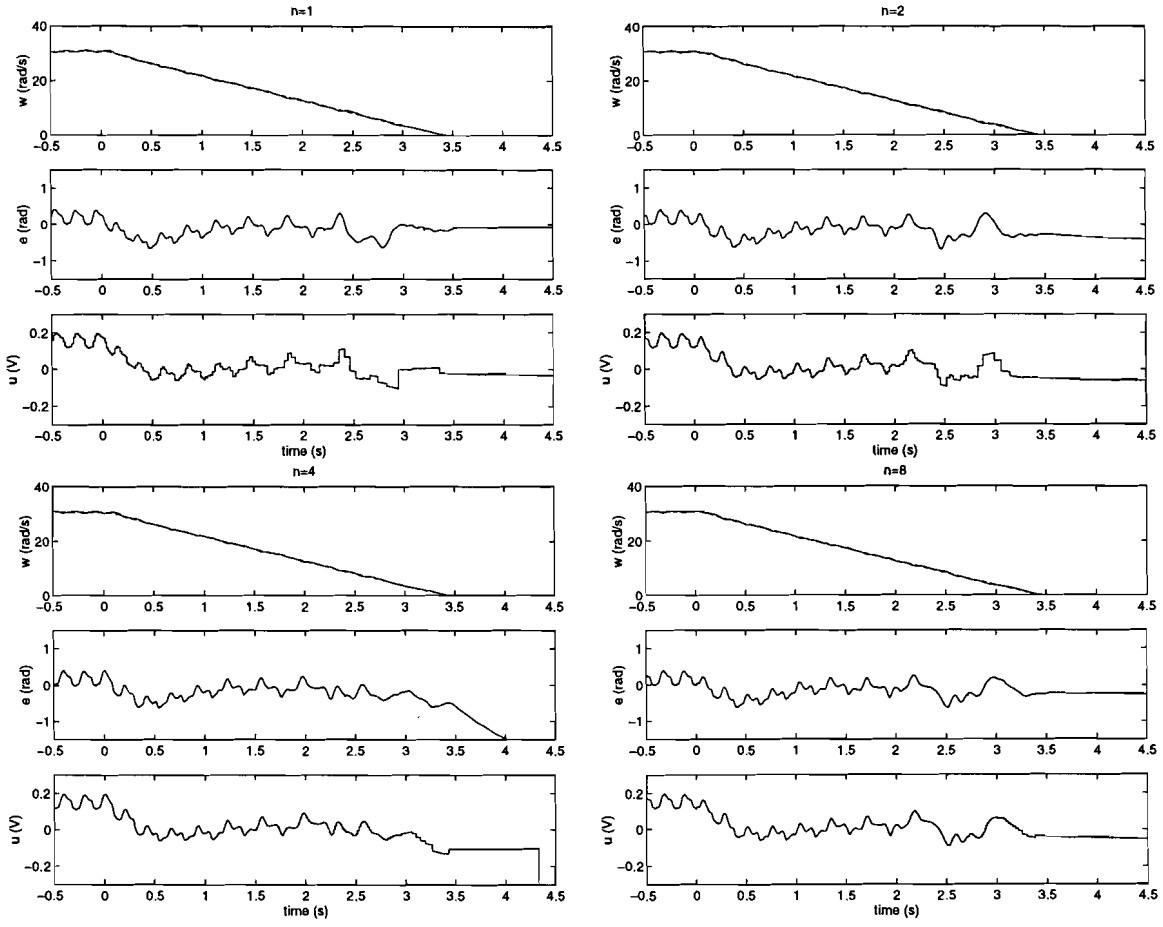


Figure 7-4: Shut down using hybrid controller

An interesting phenomena occurs for an encoder resolution of 4 pulses per revolution. In this figure it is seen that the position error becomes larger after t=3.5 seconds. At t=3.5 seconds the master motor is (almost) standing still. Although the feed-forward signal from the master motor is zero, the slave motor is still turning. This is due to the fact that the controller still keeps its last applied output voltage. Due too the asynchronous algorithm, the position error and thus the controller output is not updated anymore until a new position measurement becomes available. Therefore the position of the slave motor gradually changes. This imposes a growing position error. This problem can probably be solved, by switching off the controller for very low values of the feed-forward signal.

The occasions that this effect occurs are more or less random. It depends on the value of the controller output just before the master motor halts. If the controller output is smaller than some value, the static friction is higher than the driving force of the frequency converter. If the controller output is larger, the friction will be overcome and the described effect will occur.

7.3 Phase change

Now the effect of changing the phase in the position between the master and slave motor is investigated. A change of the phase between the master and slave motor, can be needed to fine tune the position of the printed material on the lug chain. It can also be needed to change the phase between the master and slave motor if the size of the printed material is altered.
Now it is investigated, what the effect of a step change in the phase is. Therefore a step change of 3 rad on the motor position is applied. This corresponds to a step of ¼ rad on the load axis for the used gear ratio. The motors are again running at the nominal production speed.
These are the results for the asynchronous control scheme.

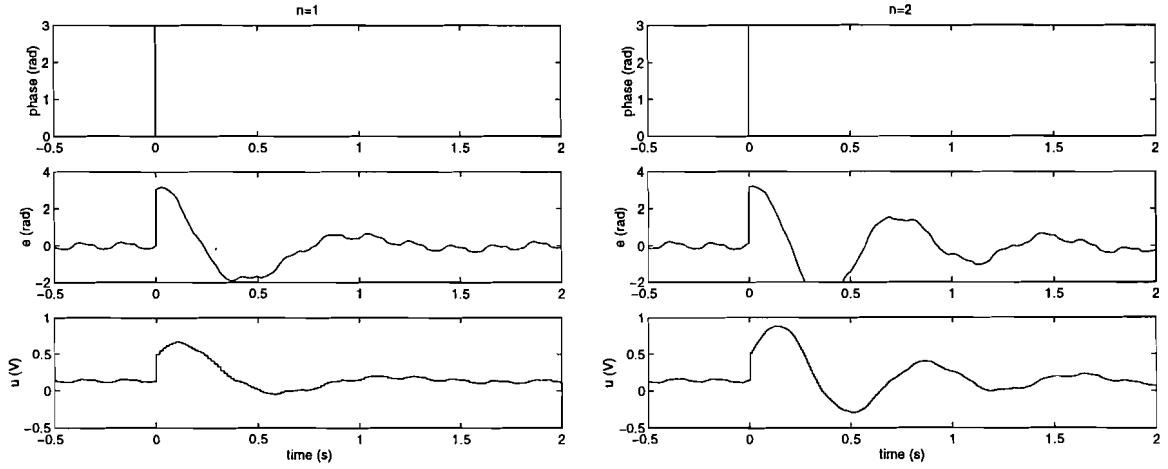


Figure 7-5: Step on phase, using asynchronous controller

On the top axis, the step in the phase signal is drawn. In the middle, the position error is drawn in radians. Below is the controller signal in volts.
It was found that for an encoder resolution of 4 pulses per revolution, the response becomes unstable. This problem probably again be solved, by tuning the controller parameters for this encoder resolution as described in section 6.5.
The results for the hybrid controller, of section 4.5 are given below.

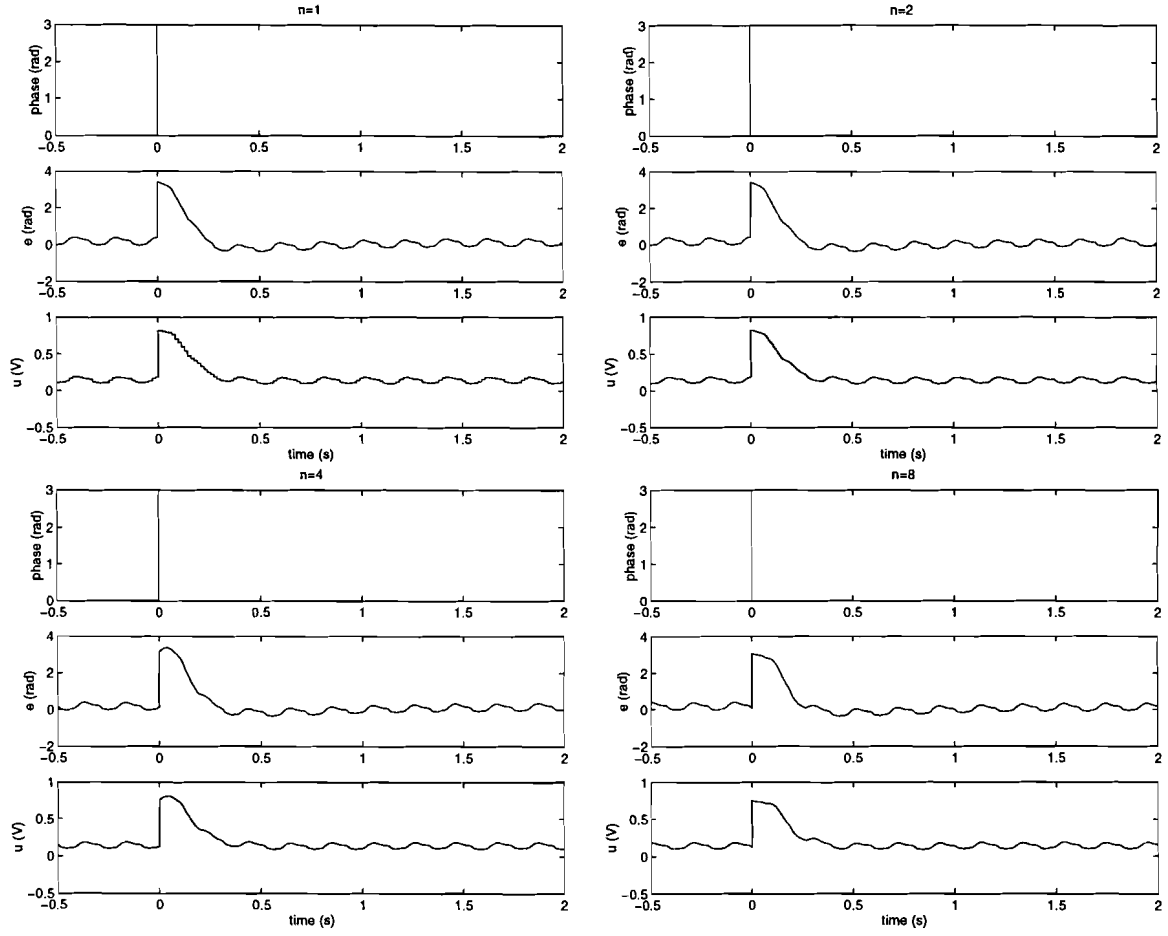


Figure 7-6: Step on phase, using hybrid controller

If the results of the hybrid controller are compared to those of the asynchronous controller, it can be seen that the hybrid controller has less overshoot in its response. This can probably be ascribed to the settings of the controller.

7.4 Step on feed-forward signal

The effect of applying a load to the master motor is investigated. This can be simulated by distorting the feed-forward signal coming from the master motor. If a load is applied to the master motor, the master motor speed will be smaller than the frequency coming from the master frequency converter. As this reference is also applied to the slave motor, the slave motor will run faster if all other variables are kept constant. This will lead to a position error between the master and the slave.
To investigate this, a step of 0.8 Volts is applied to the feed forward signal. This is done, while both motors where running at the nominal speed, which corresponds to a reference voltage of 8.5 Volts.
Here are the results for the asynchronous controller.

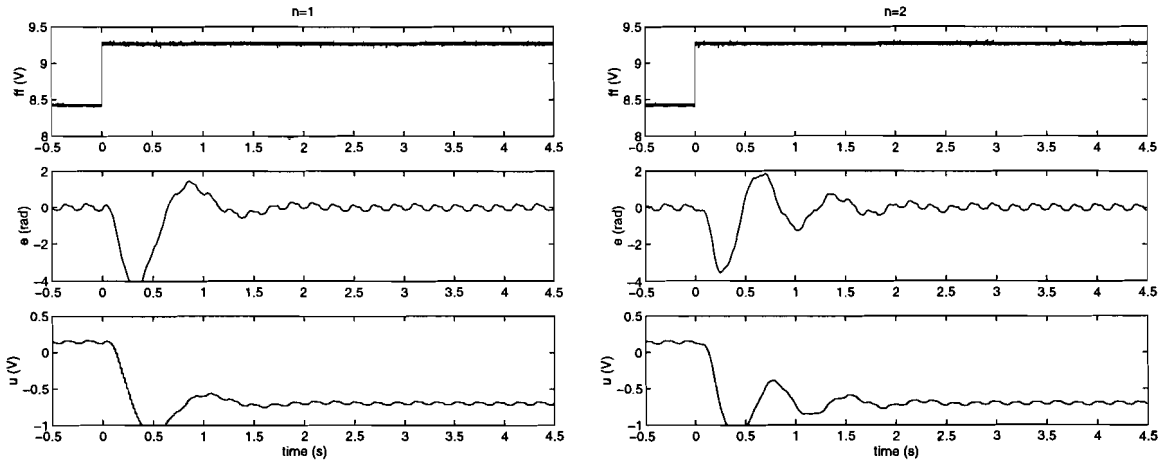


Figure 7-7: Step on feed-forward, using asynchronous controller

On the top axis the feed-forward signal is drawn. Before the step, this signal is equal to the reference voltage of 8.5 Volts. After the step, this signal is 9.3 Volts. In the middle, again the position error is drawn. Below is the controller signal.
The system was unstable for an encoder resolution of 4 pulses per revolution.
The results for the controller form section 4.5 are given below.

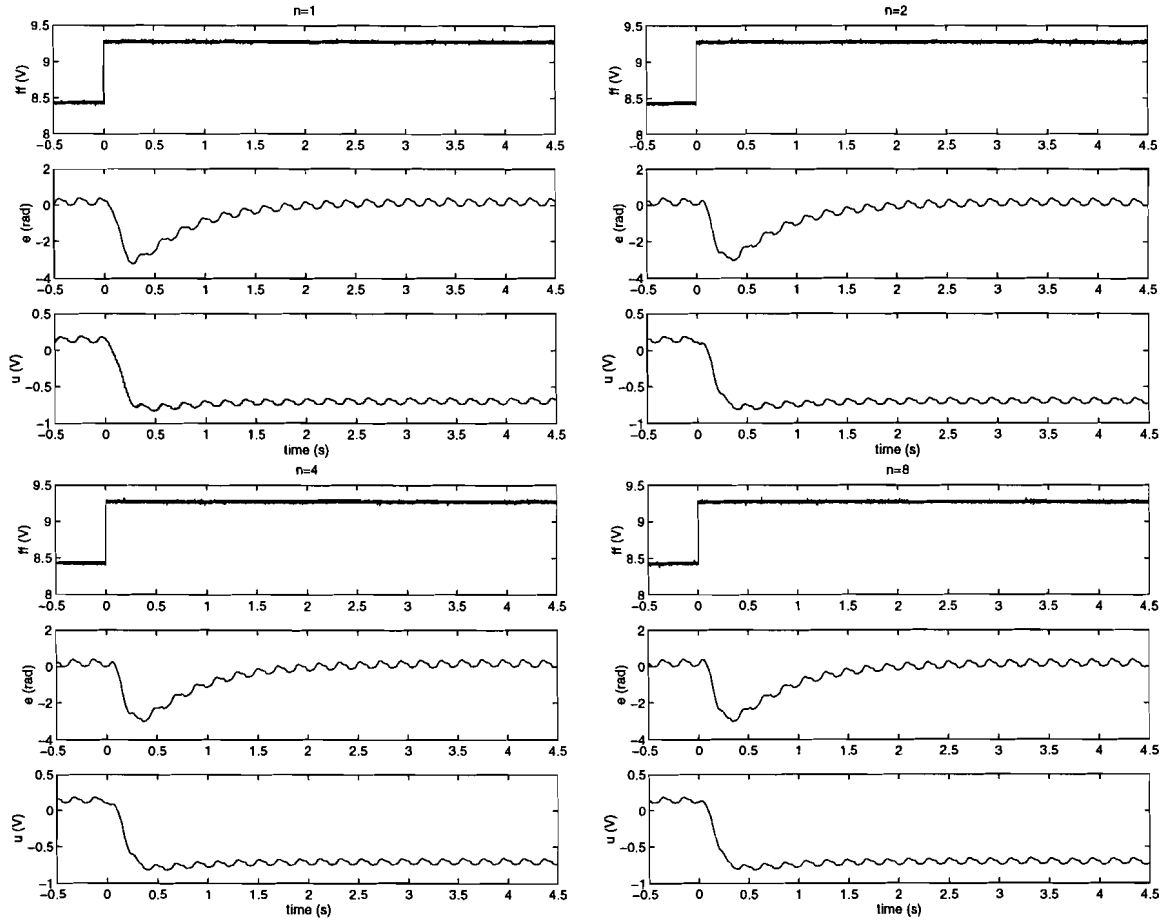


Figure 7-8: Step on feed-forward signal, using hybrid controller

There were hardly any differences to be seen between the different encoder resolutions. The response of the hybrid controller has less overshoot than the asynchronous.

7.5 Low frequency disturbance

Here the rejection of low frequency disturbances is investigated. Therefore at the nominal production speed a sine wave is applied to the feed-forward signal. This sine wave has a fixed amplitude of 0.5 V and a varying frequency.
The effect of this distortion on the position error was investigated. To eliminate the effect of this distortion, from the effect of the sheet feeder torque, the Fourrier transform of the distortion signal was considered. The magnitude of this Fourrier transform at the frequency of the applied feed-forward was compared to the peak of the applied disturbance for that frequency. The amplitude of the transfer for that frequency is then given by:

$$\left|H_{ff}(\omega_0)\right| = \frac{\left|e(\omega_0)\right|}{\left|V_{ff}(\omega_0)\right|} \tag{7.2}$$

Here H_{ff} is the transfer function describing the transfer from the feed-forward signal to the position error. $|V_{ff}(\omega_0)|$ is the magnitude of the feed-forward signal for its base frequency ω_0. In this case the feed-forward signal is a sine wave. Therefore, the magnitude will be equal to the amplitude of the sine wave. $|e(\omega_0)|$ is then the magnitude of the position error for the frequency ω_0.
For the nominal production rate, the load axis speed is about 30 rad/s (see above). This means that the base component of the sheet feeder torque lies at this frequency. For higher frequencies the effect of the applied distortion can not be discriminated from the effect of the sheet feeder torque. Therefore only lower frequencies are considered.
In the next table, the results are given for an encoder resolution, N=1. In the first column the frequency of the applied distortion is given. In the second, the magnitude of the distortion is given for this frequency. In the third the magnitude of the position error is given for this frequency. The magnitude of the transfer from the distortion to the position error is then calculated using Equation 7.2 and is given in the fourth column.

Table 7-1: Disturbance attenuation, asynchronous controller

ω(rad/s)	ff(V)	|e(rad)|	|H(dB)|
0.1	0.5056	0.0691	-17.2
0.3	0.5065	0.2070	-7.8
0.5	0.5136	0.3406	-3.6
0.8	0.5065	0.5545	0.8
1.0	0.5065	0.6993	2.4
3.0	0.5063	2.45	14
5.0	0.5092	5.10	20

For higher frequencies the system becomes unstable. If this will prove to be a problem in the practical implementation, this can probably be solved by tuning the controller parameters.
The following results were found for the synchronous controller with asynchronous measurement update.

Table 7-2: Disturbance rejection, hybrid controller

w(rad/s)	|ff(V)|	|e(rad)|	|H(dB)|
0.1	0.5084	0.1669	-9.7
0.3	0.5076	0.4981	-0.16
0.5	0.5120	0.7907	3.8
0.8	0.5037	1.2110	7.7
1.0	0.4714	1.4448	9.7
3.0	0.5087	2.50	14
5.0	0.4992	2.5489	14
8.0	0.5051	2.4538	14
10	0.5142	2.3087	13

In the Figure 7-9, for the two control schemes, the disturbance rejection is drawn as a function of ω (rad/s).

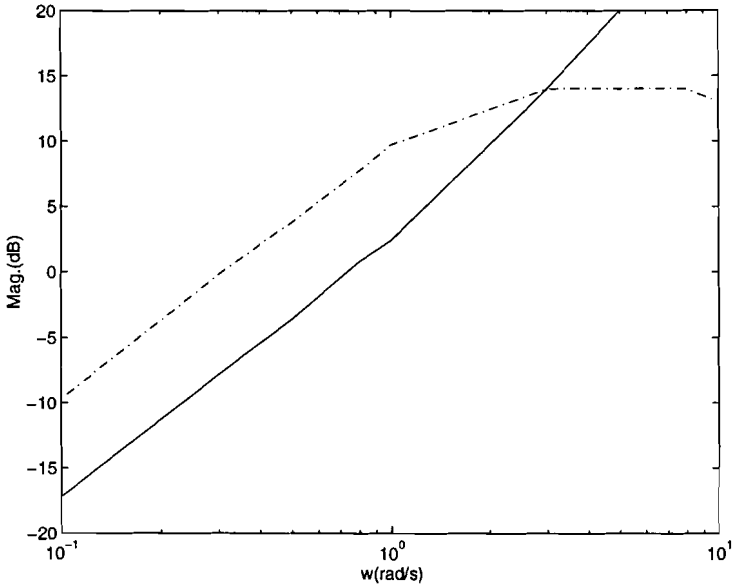


Figure 7-9: Disturbance rejection for low frequencies

The gain for the asynchronous controller, continually grows larger and ultimately leads to instability. As stated earlier the frequencies of the sheet feeder torque start at 30 rad/s if the machine is running at its nominal production speed. For these higher frequencies, the system is stable. So there is a peak in the transfer between 5 and 30 rad/s for which the system in unstable.
The hybrid controller of section 4.5 has a somewhat higher gain for frequencies lower than 1 rad/s. For frequencies from 3 to 10 rad/s, the gain stays the same and no instabilities are found.
Although from the previous could easily be concluded that the hybrid controller has a better performance than the asynchronous controller, one has to be careful. In these tests only two controllers are compared, not the two classes of controllers. If different values of the controller parameters were used, the results could be different. These tests are only meant to illustrate what can be achieved using a first implementation of both controller types.

8 Chapter eight: Conclusions and recommendations

In the previous chapters several controllers have been designed and evaluated. All these controllers had one objective: to keep the position error between the master and slave motor as small as possible. An important constraint was that the cost of the controller should be low, because several controllers are needed for one mailing machine. The cost of a given controller is largely determined by the resolution of the encoder. Therefore, it was tried to keep the encoder resolution as small as possible. Another important cost factor, is the simplicity of the implementation.
The developed controllers will now be evaluated using these three points:

- Position error bound
- Encoder resolution
- Simplicity of implementation

Where needed, some additional conclusions will be drawn

8.1 Conclusions

8.1.1 Synchronous PI-controller

The following conclusions can be drawn, for the synchronous PI-controller developed in Chapter 4 (except section 4.5).

- For high encoder resolutions, the measured position error lies within the maximum allowed error bound. The PI-controller is very easy to implement as there are a lot of standard PI-controllers on the market.
- For lower encoder resolutions, a large measurement error is introduced on the measurement of the position error. The real position error is perturbed by this measurement error. Due to this measurement error, a steady state error is introduced on the real position error. To solve this problem, an alternative position measurement algorithm is developed. Using this algorithm, we arrive at the so called hybrid controller described in section 8.1.3.

8.1.2 Synchronous H_∞-controller

The following conclusions can be drawn for the H_∞-controller, developed in Chapter 5:

- Using a high order H_∞-controller, it is possible to obtain very small position errors, provided that the limitations of the actuator are not present. The fact that the frequency converter limits the rate by which its output changes, greatly reduces the possible controller performance.
- Taking the actuator limitations into account, leads to another H_∞-controller. The maximum position error obtained using this controller is slightly smaller then was obtained using the PI-controller. As the H_∞-controller is fifth order instead of the first order controller, it is harder to implement. A trade-off has to be made between performance and ease of implementation. It is decided that the improved performance, does not counterbalance the more difficult implementation. Therefore, the effect of lowering the encoder resolution is not investigated for H_∞-controllers. The focus will be on PI-controllers instead.

8.1.3 Hybrid cc ıtroller

The following conclusions can be derived for the hybrid controller of section 4.5 (see also Chapter 7):

- Using an alternative position error calculation, the performance of a synchronous PI-controller can be enhanced for low encoder resolutions. This is done by keeping the measured position error constant in between measurement instants. This control structure is called hybrid, because it uses an asynchronous measurement update and a synchronous control action
- The controller is a simple first order system and is therefore easy to implement. The measurement error, however should be updated asynchronous in time. This calls for some interrupt driven logic circuits.
- The performance of this controller is robust for all disturbances that were investigated. During the tests that were carried out, it did not become unstable.
- In our tests, during start-up and shut-down, the measured position error complies with the demanded specifications, for all encoder resolutions. So even one pulse per revolution is sufficient in the test set-up

8.1.4 Asynchronous controller

The following is concluded for the asynchronous controller found in Chapter 6 (see Also Chapter 7):

- For one measurement per revolution of the motor axis, the asynchronous controller satisfies the maximum error bound. It was found that for some distortion signals that lie in a particular frequency range, the controlled system becomes unstable. This can probably be solved by adjusting the controller parameters.
- For higher encoder resolutions, the controlled system can become unstable at high motor speeds. This can probably be solved by tuning the controller parameters for a given encoder resolution.
- The asynchronous controller is very easy to implement, it is a basic first order system. To obtain the asynchronous behaviour however, some interrupt driven logic circuits are needed.
- Using the transformation described in Chapter 6, the asynchronous control problem in the time domain can be converted into a synchronous problem in the position domain. Thus the asynchronous problem, which is hard to tackle using standard control techniques is converted into a 'standard problem'. The only drawback is that the systems description then becomes nonlinear.

8.1.5 Final

Summarizing, the following pros and contras can be given for the various controller types:

- Synchronous PI-controller
    - Simple design
    - Simple implementation
    - High encoder resolution necessary
- Synchronous H_∞-controller
    - More complex design
    - More complex implementation
    - Due to limitations of the actuator little is gained from the PI-controller

    - High encoder resolution necessary
- Hybrid controller
    - Low encoder resolution is sufficient
    - Asynchronous position error calculation
    - High controller update frequency needed
- Asynchronous controller
    - Low encoder resolution is sufficient
    - Asynchronous position error calculation and control update

8.2 Recommendations

The following recommendations for further actions and investigations can be given:

- The limitation on the rate of change of the frequency converter is the biggest bottleneck for the possible performance of any controller. If the performance should be enhanced, this limitation should be made less severe.
- With asynchronous as well as the hybrid controller, it is possible to obtain the performance desired for this application. The resolution needed for the encoder can be as low as one pulse per revolution. The resolution that is needed on the final machine however depends on practical implementations as gear ratio's and disturbances that were not investigated here. The asynchronous controller sometimes became unstable during the tests. This can probably be solved by fine tuning the controller parameters. The choice for an asynchronous or hybrid controller largely depends on the ease of implementation.
- The possibilities for a cheap and simple implementation have yet to be investigated. The ease of implementation largely depends on what standard components can be used. After this has been investigated a choice between the asynchronous or the hybrid controller can be made.
- In Chapter 6, the asynchronous control problem was rewritten into a synchronous problem. The system description then became non-linear. A number of possible techniques to handle asynchronous systems was described. Only one, the linearisation technique was fully investigated. It could be interesting to investigate the other solutions, in part non-linear control.
- The position error presented to the hybrid controller is now kept constant between measurement updates. The performance can probably be enhanced by estimating the real position error in between measurement updates. This can be done in an observer type approach based on a model of the system. This was earlier described in [1].

8.3 References

[1] Phillips, A.M. and Tomizuka, M.
MULTIRATE ESTIMATION AND CONTROL UNDER TIME-VARYING DATA SAMPLING WITH APPLICATIONS TO INFORMATION STORAGE DEVICES
In : Proceedings of the 1995 American Control Conference, Seattle, WA, USA, 21-23 June 1995
Evanston, Ill. : American Autom. Control Council, 1995, Vol. 6, p. 4151-5

A Appendix A: Matlab listing for H_∞-controller

```
% HINFMOT.M
%       calculates H/inf controller for
%       linear model of asynchronous motor

% motor model parameters
        kt=0.35;% torque-slip angle factor
        tau=0.05;       % electric time constant
        j=8.5e-3;       % inertia of motor + load
        b=9.8e-3;       % damping
% frequency converter parameters
        kf=46.3;        % conversion factor volts=>rad/s

% calculate process state space equations
        % d/dt x=Ap*[e wm tm] + Bp*[r d u]
        Ap=[    0       -1      0;
                0       -b/j    1/j;
                0       -kt/tau -1/tau];
        Bp=[    1       0       0;
                0       -1/j    0;
                0       0       kt*kf/tau];
        % [u e]=Cp*d/dt x + Dp*[r d u]
        Cp=[    0       0       0;
                1       0       0
                1       0       0];
        Dp=[    0       0       1;
                0       0       0
                0       0       0];

% weighting filters
        % inputs
        rnum=450*0.2;
        rden=[1 0.2];
        dnum=2;
        dden=[1/360 1];
        % outputs
        enum=1000;
        eden=1;
        unum=1/10;
        uden=1;

% calculate augmented plant
        proces  = ltisys(Ap,Bp,Cp,Dp);          % process
        wr      = ltisys('tf',rnum,rden);  % reference weight
        wd      = ltisys('tf',dnum,dden);  % distortion weight
        we      = ltisys('tf',enum,eden);  % error weight
        wu      = ltisys('tf',unum,uden);  % actuator weight
        aug     = smult(sdiag(wr,wd,1),proces,sdiag(wu,we,1));

% calculate controller
        [gopt,control]  = hinflmi(aug,[1 1]);
        [Ac,Bc,Cc,Dc]   = ltiss(control);
        [nc,dc]         = ltitf(control);

% evaluation
        % open loop
        [npr,dpr]=ss2tf(Ap,Bp,Cp,Dp,1);
```

```
[npd,dpd]=ss2tf(Ap,Bp,Cp,Dp,2);
[npu,dpu]=ss2tf(Ap,Bp,Cp,Dp,3);
[numcar,dencar]=series(nc,dc,1,dpu);

% closed loop
[Acl,Bcl,Ccl,Dcl]=feedback(Ap,Bp,Cp,Dp,Ac,Bc,Cc,Dc,[3],[3]);
[nclr,dclr]=ss2tf(Acl,Bcl,Ccl,Dcl,1);
[ncld,dcld]=ss2tf(Acl,Bcl,Ccl,Dcl,2);
[nclu,dclu]=ss2tf(Acl,Bcl,Ccl,Dcl,3);
```

B Appendix B: Matlab listing for Bode plots

```
% BODEDIS
        % Matlab .m file to calculate Bode plots of discrete controller +
        % process in the position domain

% VARIABLE DECLARATION
vin=1;                          % input voltage                 (V)
for sample=[2*pi pi pi/2 pi/4]; % sample rate                   (rad)
wr=46.3*vin;                    % master reference speed        (rad/s)
B=9.8e-3;                       % dynamci damping               (Nm*s/rad)
J=8.45e-3;                      % inertia                       (kg*m^2)
kt=0.35;                        % torque slip-angle constant    (Nm*s/rad)
kf=46.3;                        % frequency converter constant  (rad/V*s)
tau=0.05;                       % electrical time constant      (s)


% SYSTEM DESCRIPTION (THETA DOMAIN)
        % output : tracking error e       (rad/s)
        % input 1: input voltage  u       (v)
        % input 2: distortion     d       (Nm)

Ap=[    0                       -1/(wr)^2                       0;
        0                       -B/(J*wr)                       1/(J*wr);
        0                       -kt/(tau*wr)                    -1/(tau*wr)];

Bp=[    0                       0;
        0                       -1/(J*wr);
        (kt*kf)/(tau*wr)  0];

Cp=[    1                       0                               0];

Dp=[    0                       0];

        % transfer functions
[npu,dpu]=ss2tf(Ap,Bp,Cp,Dp,1);
[npd,dpd]=ss2tf(Ap,Bp,Cp,Dp,2);
% DISCRETIZATION
[Ad,Bd,Cd,Dd]=c2dm(Ap,Bp,Cp,Dp,sample,'zoh');

        % transfer functions
[ndu,ddu]=ss2tf(Ad,Bd,Cd,Dd,1);
[ndd,ddd]=ss2tf(Ad,Bd,Cd,Dd,2);

% CONTROLLER DESIGN

numc=-5*vin*poly(1-0.1*(sample/(2*pi)));
denc=poly(1);

j=sqrt(-1);
w=logspace(-3,2*inv(sample),250);
freq=w/2/pi; %

[magc,phac]=bode(npu,dpu,w);
Gu=magc.*exp(j*phac/180*pi);

[magd,phad]=bode(npd,dpd,w);
Gd=magd.*exp(j*phad/180*pi);
```

```
[magr,phar]=dbode(numc,denc,sample,w);

Gr=magr.*exp(j*phar/180*pi);

% Sample and hold

Gs=1./(sample*j*w').*(1-exp(-j*w'*sample));

%  transfer from u to y with discrete controller

Gyu=Gu./(1+Gu.*Gs.*Gr);

Gyd=Gd./(1+Gu.*Gs.*Gr);

%  output

semilogx(w,20*log10(abs(Gyd)),'-w');
ylabel('mag (dB)');
hold on;
pause;
end;
semilogx(w,20*log10(abs(Gd)),'-.w');
axis([1e-3 1e1 -150 50]);
for i=-150:3:50; plot(8e-2,i,'.w');end;
for i=-150:3:50; plot(8e-1,i,'.w');end;
hold off;
```