

ISSN 0280-5316
ISRN LUTFD2/TFRT--5709--SE

Regulator Design for a Flexible Servo

Pontus Nordfeldt

Department of Automatic Control
Lund Institute of Technology
August 2003

Department of Automatic Control Lund Institute of Technology Box 118 SE-221 00 Lund Sweden		<i>Document name</i> MASTER THESIS	
		<i>Date of issue</i> August 2003	
		<i>Document Number</i> ISRN LUTFD2/TFR--5709--SE	
<i>Author(s)</i> Pontus Nordfeldt		<i>Superviso</i> Tore Hägglund and Rolf Johansson LTH, Lund Anders Sundberg TetraPak, Lund	
		<i>Sponsoring organization</i>	
<i>Title and subtitle</i> Regulator Design for a Flexible Servo. (Regulatordesign av ett flexibelt servo)			
<i>Abstract</i> The purpose of this thesis was to examine the possibilities to control a process containing a resonance frequency. System identification was made on a process containing a resonance frequency. The identification resulted in a model of the process. Different regulators that were capable of dealing with the resonance frequency were designed for the model. Finally one regulator was implemented and used to control the process.			
<i>Keywords</i>			
<i>Classification system and/or index terms (if any)</i>			
<i>Supplementary bibliographical information</i>			
<i>ISSN and key title</i> 0280-5316			<i>ISBN</i>
<i>Language</i> English	<i>Number of pages</i> 53	<i>Recipient's notes</i>	
<i>Security classification</i>			

The report may be ordered from the Department of Automatic Control or borrowed through:
University Library 2, Box 3, SE-221 00 Lund, Sweden
Fax +46 46 222 44 22 E-mail ub2@ub2.se

1 Introduction

The work for this thesis was done at Tetra Pak in Lund. Tetra Pak is one of the world's largest suppliers of packaging systems for milk, fruit juices and drinks, and many other products. The packaging machines that Tetra Pak develops contain a lot of mechanics that must move or rotate with high velocity and precision. Those movements are controlled by methods from the fields of automation and control theory. Improvements of control methods would mean improvement of the machines and is thus of interest.

When regulating processes that contain flexible mechanics, problem with resonance frequencies often occurs. Oscillations can decrease the working speed and shorten the life span of a machine. Finding a way to minimize oscillations would therefore be of interest.

The purpose of the work presented in this thesis was to investigate some different strategies of coping with resonance problems, to design and test different regulators for a flexible servo process at Tetra Pak in Lund and to implement one regulator for the process.

Model-based design methods were chosen and thus a model of the process was needed. System identification was used to obtain this model of the process. Some different regulators were designed for the model and tested in simulations.

Finally one regulator was designed and implemented for the process.

2 Theory

2.1 System identification

A rigid body mechanics model of the flexible servo process from torque to angular velocity has an order of three (see Åström & Wittenmark 1997 chap.4). Thus a model from torque to position would have an order of four (obtained by adding an integrator).

System identification is an area with a lot of different approaches and methods. The quality of the results may vary and therefore validation of obtained models is necessary.

Statistical analysis can be used for transfer function estimation (see Johansson 1993 chap.4).

For regulator design, a low order parametric model is often needed. One way of obtaining this is to use subspace methods (see Johansson 1993 chap.13). Singular values obtained in the calculations will give an idea of what order of the model that would be suitable.

Validation of a model obtained by system identification is necessary. This can be made with residual analysis and cross validation (see Johansson 1993 chap.9).

2.2 Regulator design

There are several ways of designing a regulator. Two model-based ways are pole placement design and linear quadratic design.

When a discrete model of a process is known, a regulator that makes pole placement can be designed. If the model is present as a pulse transfer operator this can be done using a polynomial approach. The possibility to cancel poles makes it possible to obtain a regulator that also acts as a notch filter. Integral action can be added to such regulators as well (see Åström & Wittenmark chap.5).

A regulator with notch filter action will not excite the resonant mode by its control signal but it will not damp this mode actively either. This means that if a load disturbance causes resonance oscillations the regulator will not stop them (see Åström & Wittenmark 1997 chap.5).

If a pole placement regulator without cancellation of any pole is designed, it is possible to obtain a regulator that makes active damping of the resonant mode. This means that any resonance oscillations will be damped (see Åström & Wittenmark 1997 chap.5).

Another type of model-based regulator design is LQ/LTR (linear quadratic regulator with loop transfer recovery) (see Åström & Wittenmark 1997 chap.5 and Glad & Ljung 1997 chap.9). Reference signal following can be obtained (see Åström & Wittenmark 1997 chap.4).

If the state gain L is chosen as a solution to a linear quadratic problem the regulated system will have some good characteristics. The phase margin will be at least 60 degrees, the amplitude margin will be infinite and the sensitivity function will be less than 1. (see Glad & Ljung 1997 chap.9). By choosing the observer gain carefully it is possible to come arbitrarily close to these good characteristics even when an observer is needed (see Glad & Ljung 1997 chap.9).

3 Material

The process was a flexible servo according to Figure 1.

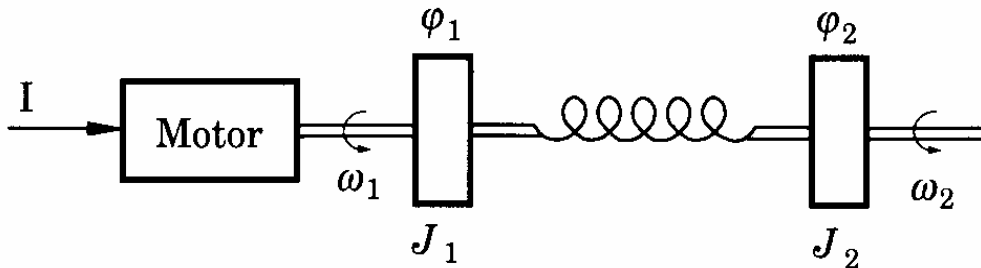


Figure 1 A schematic picture of the process

The flexibility of the process originated in a mechanical flexibility in a gear-box on the shaft. The input signal was a so called stator current quadrature component, and the output signal was the measured position of the motor. The torque produced by the motor is approximately proportional to the stator current quadrature component. A rigid body mechanics model of this process would give a system of fourth order.

4 Method

A model of the process was obtained by system identification. Then three different regulators were designed and tested in simulations.

4.1 System identification

The purpose of the system identification was to obtain a reliable low order parametric model of the process. There are different ways of achieving this and validation is crucial to determine whether the obtained model is good or not.

4.1.1 Data sets

The system was excited by an input signal and sampled at a frequency of 2000 Hz. System identification was made on one set of data (data1). Then cross validation was made with other data sets.

4.1.2 Friction compensation

Friction was measured by exciting the process with a constant input signal. The wheel stopped moving at an input signal of 0.2A. Before the identification friction was compensated for by decreasing the input of the data sets with 0.2A when the wheel was moving in positive direction, and increasing it with 0.2A when the wheel was moving in negative direction.

4.1.3 Statistical analysis

Matlab was used to do some spectral analysis of the data. Loading the data into Matlab as u and y (input and output signals respectively), and using the command line

```
>>specplot(spectrum(u,y),Ts)
```

where Ts is the sampling frequency (2000 Hz for all sets) produced functions and plots of the following:

- .Power spectral density and confidence of u
- .Power spectral density and confidence of y
- .Estimation of pulse transfer operator magnitude
- .Estimation of pulse transfer operator phase
- .Coherence function

The third and fourth quantity are extracted from the ratio between the power spectral density of the output and the input signals.

4.1.4 Low order parametric model estimation

For model estimation, subspace methods were chosen.

A rigid body mechanics model of the process (see Figure 1) would have an order of four. Thus, the identified model is expected to be of at least fourth order. This fact and the singular values (see Figure 5) suggested that a model order of four should be chosen.

The subspace identification provided a fourth order model estimation of the process.

4.1.5 Validation

A validation of data1 was made. The fourth order model estimated from data1 and the input signal and an estimation of the initial states of data1, were used to simulate an output signal. This signal was plotted in the same diagram as the true output signal of data1 in some different scales.

The residuals of the data were calculated as the difference between the real output of data1 and the output simulated by the estimated fourth order model with input from data1.

To be able to make a quantitative measure of the model's ability to simulate the output in an accurate way a relative standard deviation was introduced as

$$\text{rstd} = (\text{standard deviation of residuals}) / (\text{interval of output})$$

The variance of the residuals was estimated as

$$v = r' \times r / (\text{length}(r) - 1)$$

where v is the variance and r is a residual vector. The standard deviation, s , was calculated as the square root of the estimated variance.

One way of getting an idea of the accuracy of the model is to look at the auto correlation of the residuals and the cross correlation between the residuals and the input. The Matlab Control system toolbox command `resid`, with estimated model and data as input was used to calculate and plot these two functions. Ideally the residuals should be uncorrelated which means that the auto correlation function should vary around zero and be neither only positive nor only negative for more than a few units. Further the correlation function should be small. In the Matlab function plots, boundaries which should not be exceeded are indicated. The residuals and the input should also be uncorrelated. This imposes the same requirements on the cross correlation function as on the auto correlation function.

The power spectral density of the residuals with confidence interval was plotted using the Matlab Control system toolbox command `spectrum`, in order to see if some frequencies were overrepresented in the residuals.

Finally cross validation was made with three data sets (data2, data3 and data4).

4.2 Regulator design

Since a model of the process was identified a model-based design was chosen. One purpose was to try to avoid oscillations in the process. Three different regulators were designed and tested in simulations. Some Matlab programs for different regulator design were designed (see Appendix B).

4.2.1 Notch filter design by pole placement

The Matlab file `notchfilter.m` designs an RST regulator with notch filter and pole placement action according to the method below.

The sampled process contains four zeros and four poles. Two of the poles represent the oscillatory mode which is to be canceled and two are real. The process model's characteristic polynomial is factorized as

$$A = A^+ A^-$$

where A^+ contains a monic polynomial of degree two to be canceled and A^- is the rest of degree two. The desired characteristic polynomial is also of degree four and is chosen to be the sampled version (zero order hold) of

$$A_c = (s^2 + 2\xi_c \omega_c s + \omega_c^2)^2$$

The parameters can be chosen arbitrarily but in this case the following values were used

$$\xi_c = 1$$

$$\omega_c = 170$$

The observer polynomial A_o of degree three is chosen as

$$A_o = \bar{A}_o A^+$$

A_o -hat is thus of degree 1 and is chosen to have a zero in w_c

The Diophantine equation becomes

$$AR + BS = A^+ A_c \bar{A}_o$$

S is introduced as

$$S = A^+ \bar{S}$$

and the Diophantine equation is shortened to

$$A^- R + B \bar{S} = A_c \bar{A}_o$$

It is solved in the usual manner (the computer file uses the Matlab oldboxes command *dab*).

This gives the following transfer function from input to output

$$\frac{BT}{A^+ A_c \bar{A}_o}$$

If T is chosen as

$$T(z) = \frac{A_c(1)A^+(z)\bar{A}_o(z)}{B(1)}$$

the steady state gain is one and the resonant poles are canceled so that the command signal will not excite the resonant modes. Further, since A^+ is a factor of S the feedback will also not excite the resonant modes.

The sensitivity function was calculated to give an idea of how sensitive the regulator is to modeling errors.

Finally the regulator was tested in simulations. The regulator and process were implemented in Matlab Simulink (see Figure 18). The fourth order model from the system identification was used to simulate the process.

In the first simulation the step load disturbance was turned off (see Figure 18).

In the next simulation the step load disturbance was activated after 4.3s with an initial value of 0A and a final value of 30A. The reference signal was the same as before.

To avoid stationary errors integral action was added in the design. A new closed loop pole in the origin was introduced and an integrator was forced into the polynomial R.

4.2.2 Active damping by pole placement

The file *activedamping.m* designs an RST regulator with pole placement design for the model obtained in the system identification. This regulator does not cancel poles and thus it has no notch

filter action. The idea is to make an active damping of the oscillatory mode and not, as in the case with a notch filter, to make the regulator avoid to excite them. The process is the same as before and the desired characteristic polynomial A_c is also the same but the observer polynomial is chosen as

$$A_o = A_d$$

where A_d is the sampled versions of

$$A_d = (s^2 + 2\xi_c \omega_c s + \omega_c^2)(s + \omega_c)$$

The parameters are the same as those in A_c in the file but could be chosen differently.

The Diophantine equation becomes

$$AR + BS = A_c A_d$$

It is solved in the usual manner (Matlab command *dab* in the file).

Integral action is added in the same manner as in the case with the notch filter design.

The sensitivity function was calculated and plotted in order to give an idea of the regulator's sensitivity to modeling errors.

The regulator was implemented and simulated in Matlab Simulink. The Simulink model was the same as in the case with the notch filter design and the same reference signal and load disturbance was used.

4.2.3 LQG/LTR

The purpose of the LQG/LTR regulator design was to design a robust regulator.

An LQG/LTR regulator was designed for the process. The regulator was designed for the continuous equivalent of the model obtained in the system identification, and then sampled. To avoid computations with too great numbers the unit of the output signal was altered, divided by a factor 1000. For simplicity a small direct term was removed from the model.

An LQ regulator with robust closed loop performance was designed.

The open loop transfer function of the regulated system was plotted and the amplitude and the phase margins were investigated. The sensitivity function was also plotted.

Thereafter loop transfer recovery (LTR) was obtained by choosing observer gain K according to the theory. The open loop transfer function and the sensitivity function were plotted again to verify that the good characteristics of the LQ regulated system were recovered to a satisfactory extent. The calculations in the design were organized in the Matlab file LQGLTR.m (see Appendix B).

The regulator was simulated in Matlab Simulink (see Figure 38). The fourth order model from the system identification was used to simulate the process.

4.3 Implementation

A regulator for the process was implemented in the programming language C. The program was loaded to a PC computer. The PC computer communicates with a drive that contains the regulator. The control signal from the regulator is actually a reference signal for the current regulator in the three phase motor that drives the process. It is a reference signal for the current quadrature

component which is a virtual current. The torque on the shaft is approximately linear to the current quadrature component. The computer system had a real time scheduling from the beginning. Thus it was possible to implement the regulator without worrying too much about real time issues.

It was discovered that the model obtained from system identification had a shortcoming. Compared to the process it had a very large low frequency gain. A new data set (data4) was taken with an input signal that excited low frequencies more than the data sets used before. The new data set was used to identify a second order model in the same way as the identification of the fourth order model earlier. A pole placement regulator for this model was designed in the same way as the active damping regulator above. The desired closed loop characteristic polynomial had to be chosen with a lower order since the model order was lower. When this regulator was implemented and used to regulate the process it was noticed that there was a stationary error. The ratio between the reference signal and the position was constant but not one. This was compensated for by multiplying the reference signal with the mentioned ratio before feeding it into the regulator.

No resonance oscillations could be detected by the eye or by looking at plots of the output signal. In order to investigate the presence of resonance oscillations quantitatively the power spectral density of the output was plotted.

5 Results

In this section the results of the system identification and the regulator design is presented.

5.1 System identification

The results of the system identification is presented in order.

5.1.1 Statistical analysis

The statistical analysis resulted in an estimation of the transfer function of the process (see Figure 2, Figure 3) and a coherence spectrum for the estimation (see Figure 4)

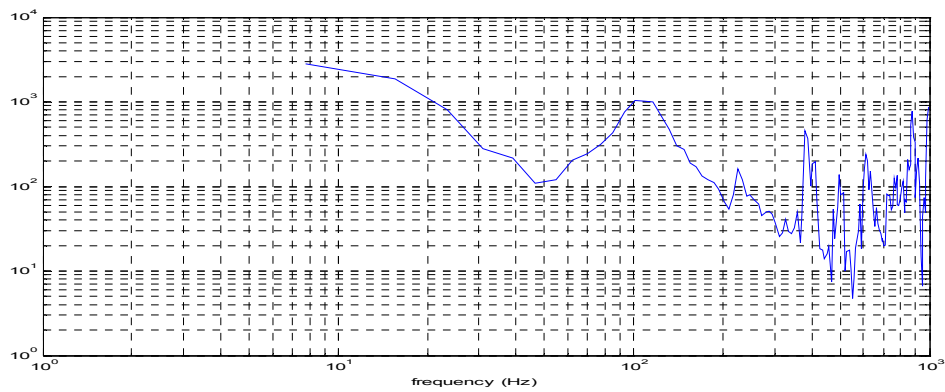


Figure 2 Estimated transfer function magnitude obtained by statistical analysis. The unit on the y axis is one.

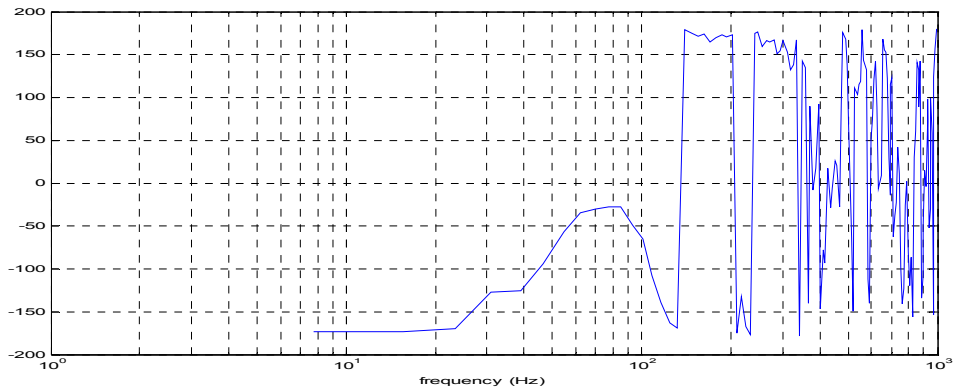


Figure 3 Estimated transfer function phase obtained by statistical analysis. The unit on the y axis is degrees.

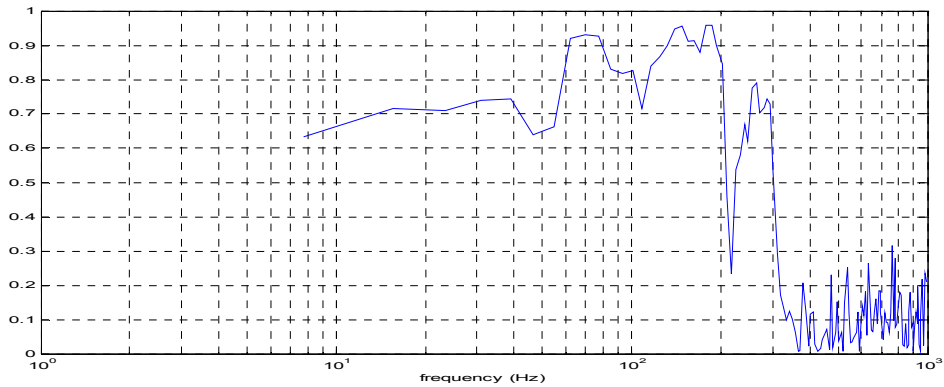


Figure 4 Coherence spectrum. The unit on the y axis is one.

The magnitude curve shows a main resonance frequency centered around 106 Hz with a half-value-width of approximately 40 Hz. Higher up the Bode plots show some additional resonance frequencies. The coherence is poor and the energy of the excitation signal is low at these frequencies.

5.1.2 Model estimation

Figure 5 shows the first twenty singular values from the subspace algorithm.

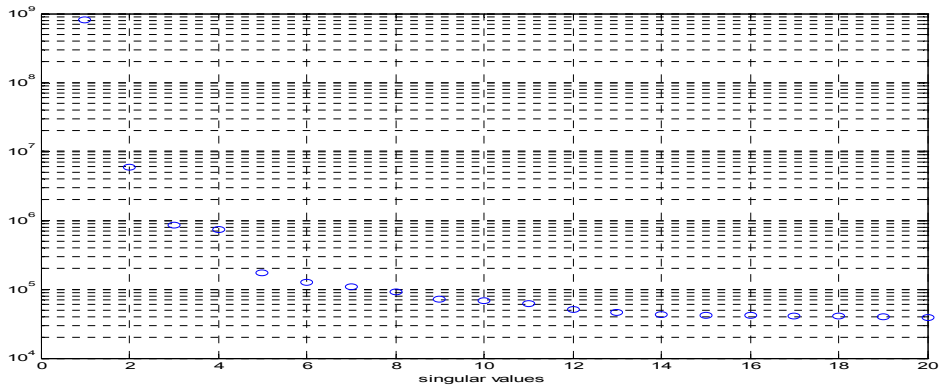


Figure 5 The first twenty singular values obtained in the subspace algorithm. The unit on the y axis is one.

A high singular value indicates that the corresponding state is important for model accuracy. Since the rigid body mechanics model says that the model should be of at least fourth order and the singular values from five and up are small (see Figure 5) in some sense, a model order of four is chosen.

Figure 6 shows a Bode plot of the estimated fourth order model.

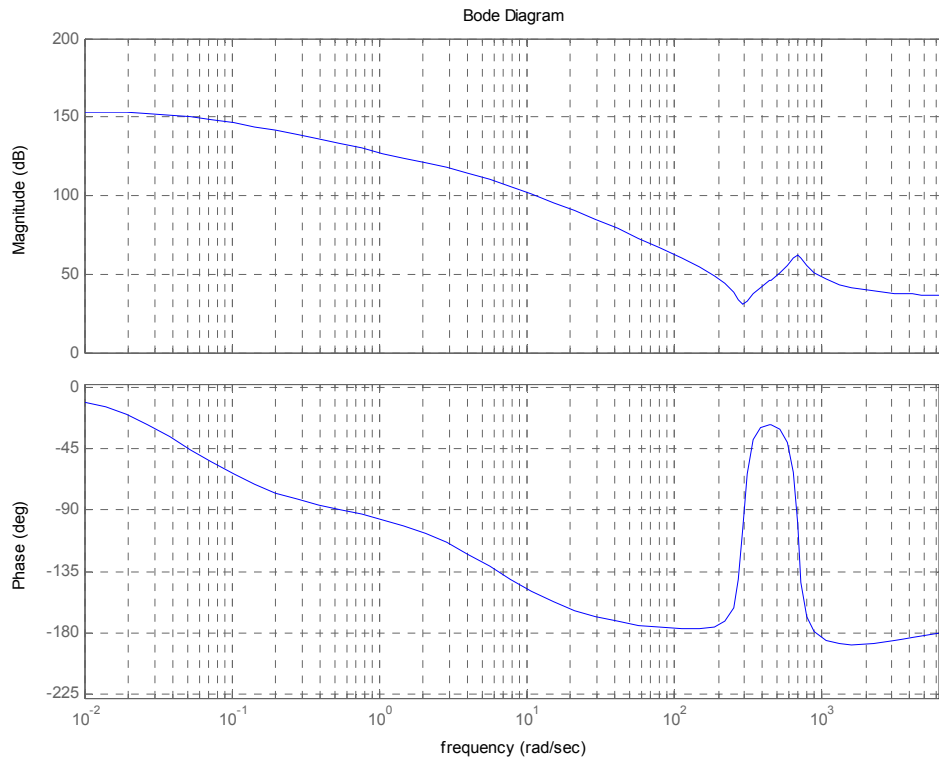


Figure 6 Bode diagram of the estimated fourth order model

Figure 6 shows a resonance frequency centered around 106 Hz. For frequencies under 200 Hz the shape of the diagram is about the same as in the diagrams obtained by statistical analysis (see Figure 2 and Figure 3).

5.1.3 Validation

Figure 7 and Figure 9 show the output and a model simulated output of data1. Figure 8 and 10 show residuals between the output and the simulated output.

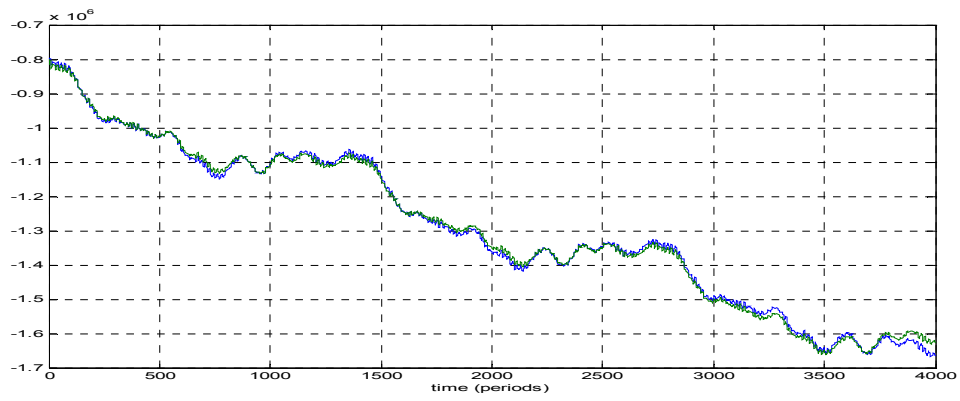


Figure 7 Output and simulated output of data 1. The unit on the y axis is the same as the units of the measured position.

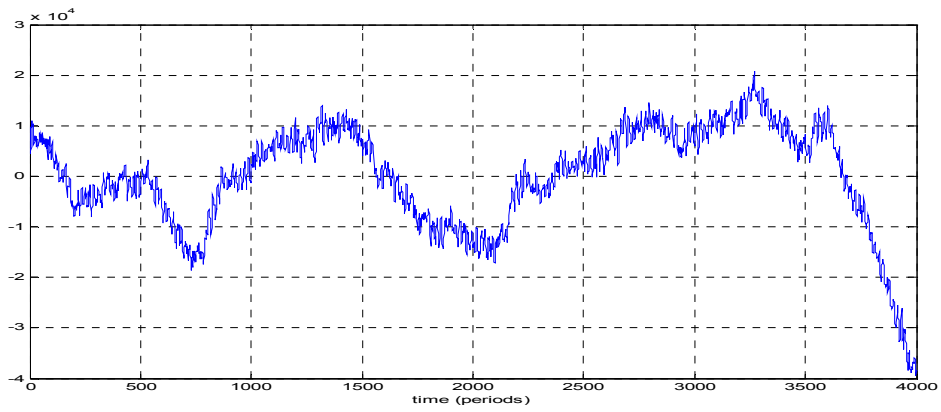


Figure 8 Residuals. The unit on the y axis is the same as the units of the measured position.

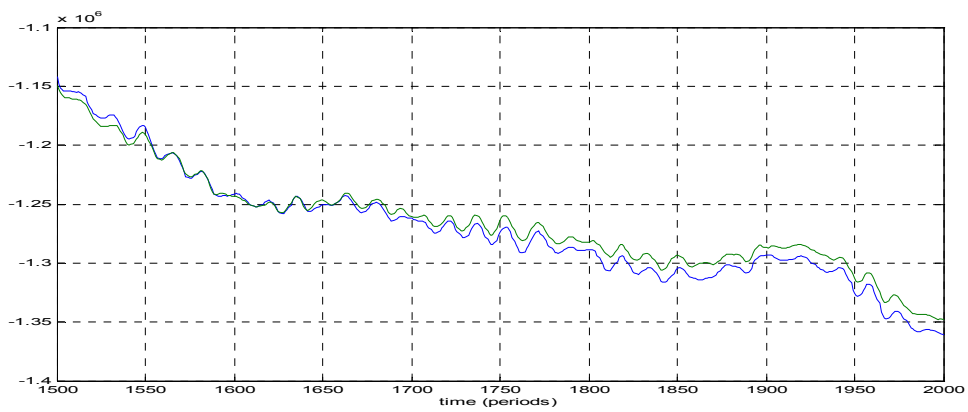


Figure 9 Figure 7 in a different scale.

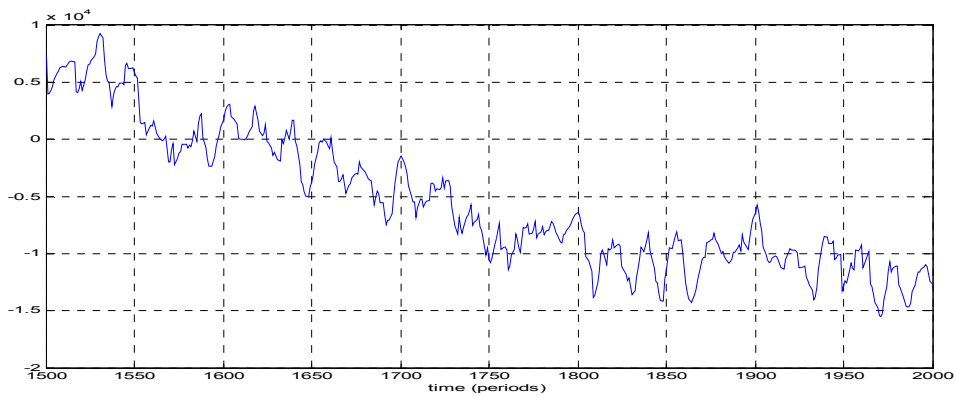


Figure 10 Figure 8 shown in a different scale.

The variance and the standard deviation of the residuals were calculated and were found to be

$$v = 1.022 \times 10^8$$

$$s = 1.011 \times 10^4$$

The data varies between approximately -8×10^5 and -17×10^5 units.

The relative standard deviation introduced above (section 4.1.4) was calculated.

$$\text{rstd} = s / ((-8 - (-17)) \times 10^5) = 1.1\%$$

The autocorrelation for the residuals and the cross correlation between the residuals and the input are shown in Figure 11 and Figure 12 in different scales.

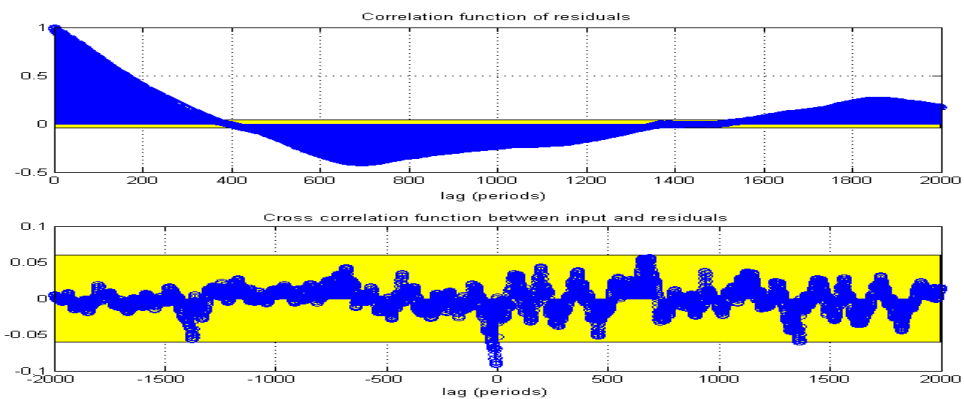


Figure 11 Auto correlation function of the residuals and cross correlation function between the residuals and the input.

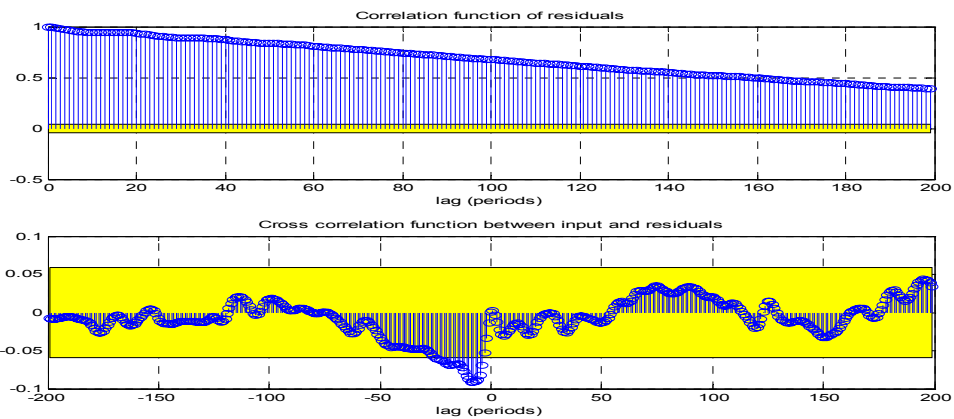


Figure 12 Figure 11 in a different scale

The cross correlation function shown in Figure 11 and Figure 12 is acceptable but the auto correlation function indicates that the residuals are correlated.

Figure 13 shows the power spectral density of the residuals.

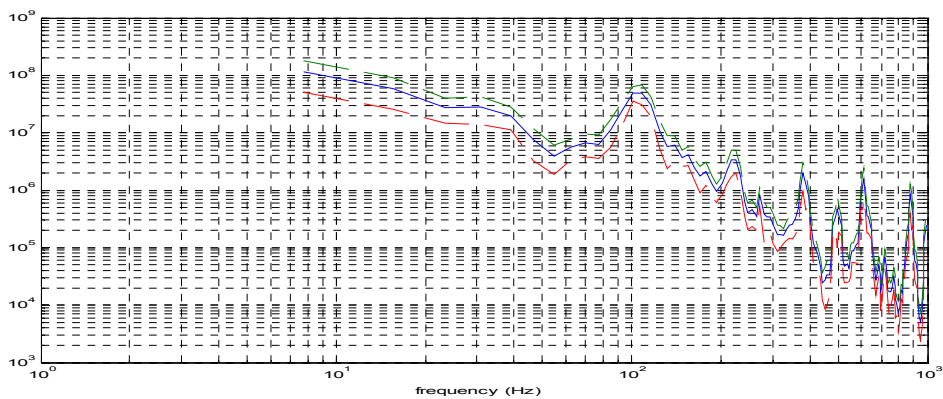


Figure 13 The power spectral density of the residuals with 95% confidence interval

Low frequencies and frequencies around the resonance frequency dominate the frequency content of the residuals.

Figure 14, Figure 16 show cross validation made with data2, data3. Figure 15 and Figure 17 show the corresponding residuals.

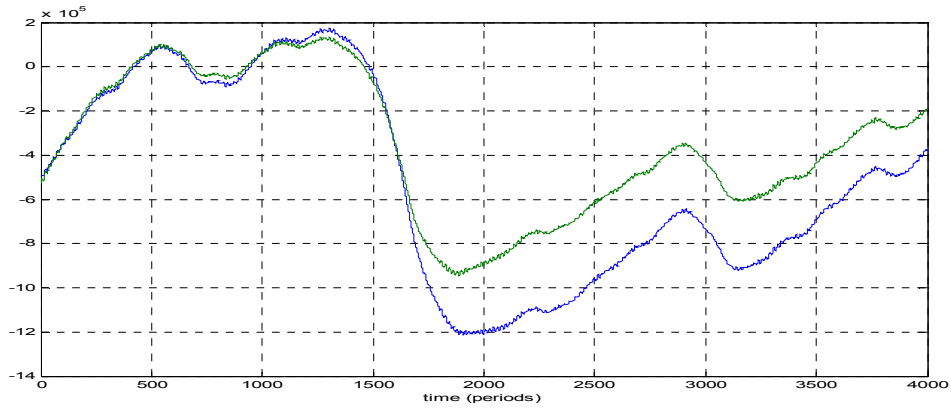


Figure 14 Cross validation with data2. The unit on the y axis is the same as in the measured position.

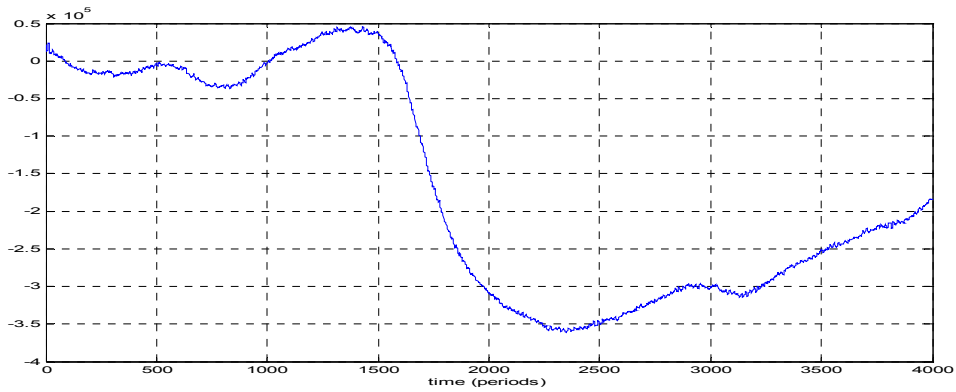


Figure 15 Residuals corresponding to the cross validation in Figure 14.

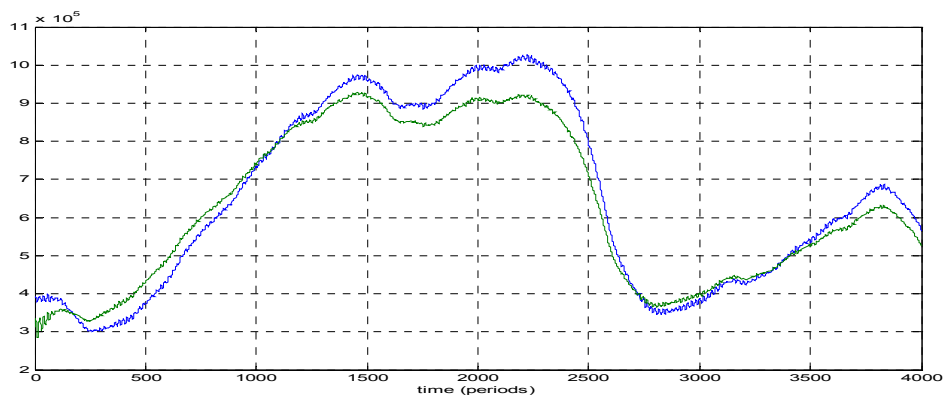


Figure 16 Cross validation with data3. The unit on the y axis is the same as in the measured position.

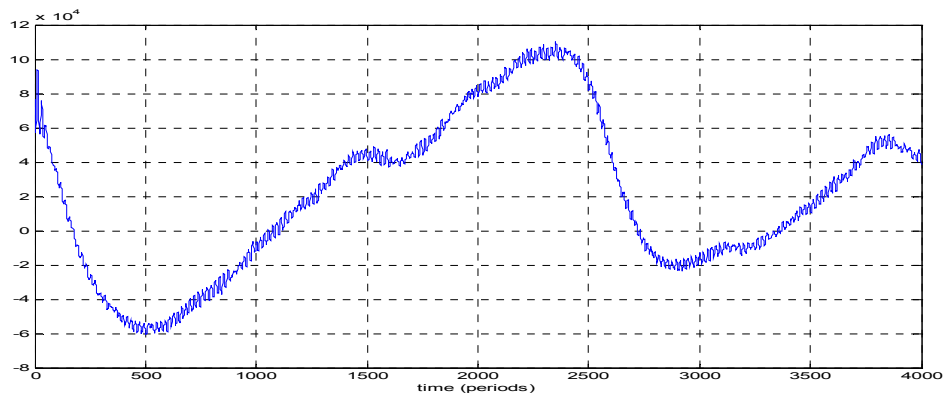


Figure 17 Residuals corresponding to the cross validation in Figure 16.

The high frequency behavior of the model is satisfactory but the low frequency behavior seems to be less good. However, the proportional part of a regulator could be able to compensate for this. Special attention has to be paid to the sensitivity function in the low frequency region of a regulator.

5.2 Regulator design

Three different regulators were designed and tested in simulations in Matlab Simulink.

5.2.1 Notch filter design

The notch filter regulator design resulted in three polynomials (R, S and T) to be used in the filters of the regulator.

The regulator was implemented and simulated in Matlab Simulink.

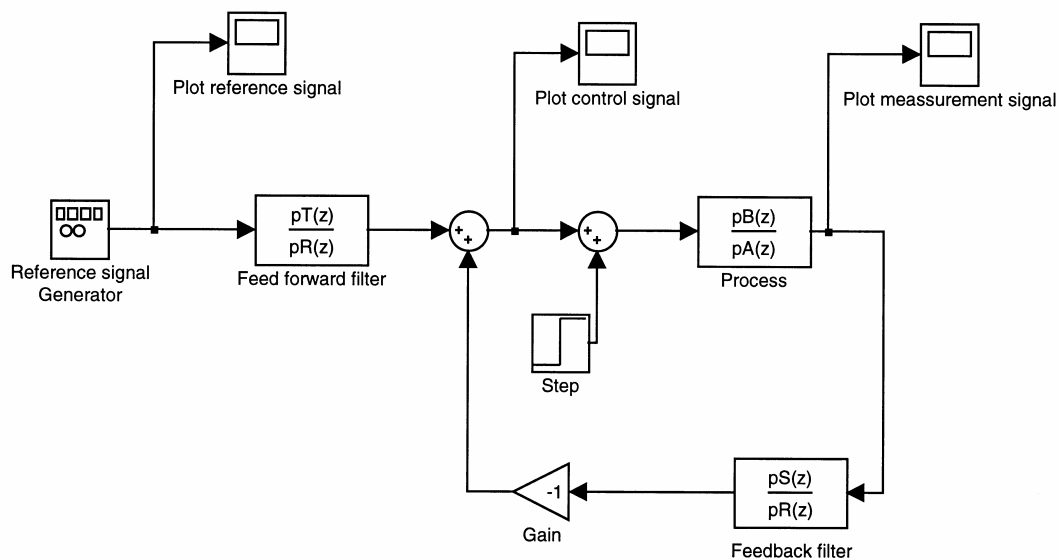


Figure 18 Implementation of regulator in Matlab Simulink

The fourth order model from the system identification was used to simulate the process.

5.2.1.1 Notch filter simulation example one

In this simulation example all poles of the closed loop system were placed in the discrete time equivalent of the real continuous pole in $p = -1.70$. No disturbances acted on the system. The fourth order model from the system identification was used to simulate the process. The implementation was done according to Figure 18. Figure 19 and Figure 20 show the reference signal and the output signal. Figure 21 shows the control signal. For the coefficients of the regulator and model polynomials see Appendix A.

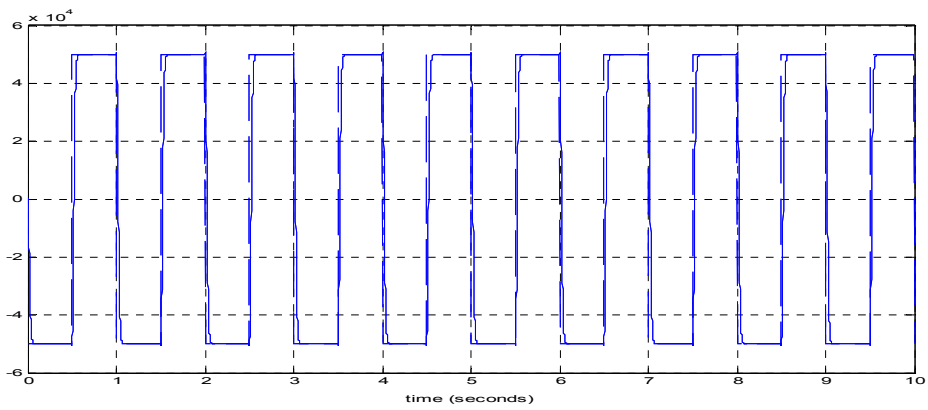


Figure 19 The reference signal (dashed) and the output signal (solid). The unit on the y axis is the same as the unit of the measured position.

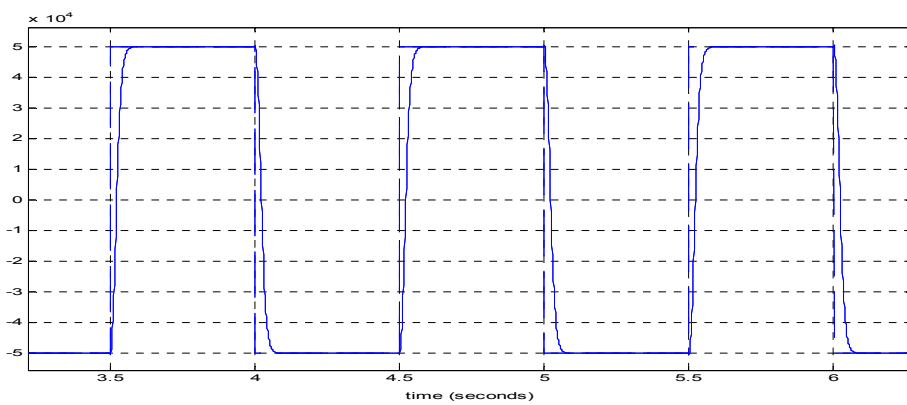


Figure 20 The same Figure as Figure 18 but in a shorter time interval.

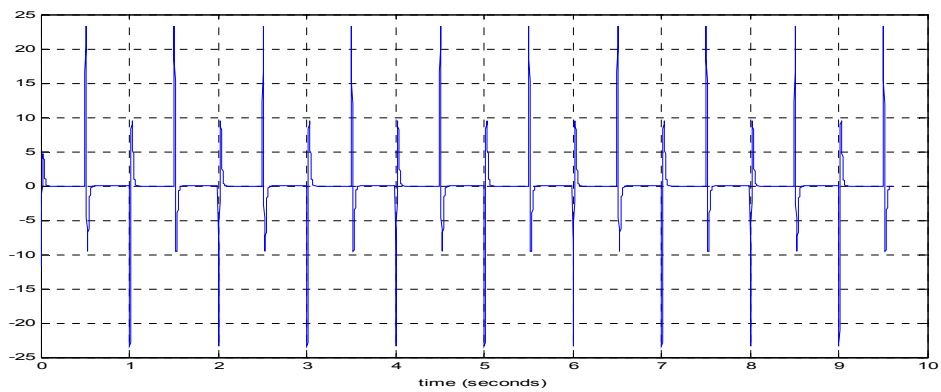


Figure 21 The control signal. The unit on the y axis is ampere.

The sensitivity function gives an idea of how sensitive the regulator is to modeling errors.

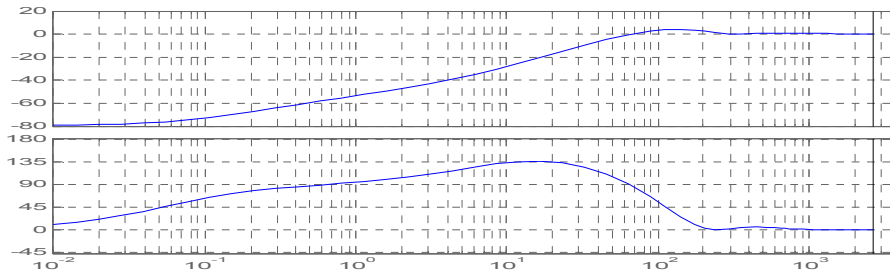


Figure 22 Sensitivity function for the notch filter regulator in the simulation. The unit on the x axis is radians per second. The first diagram shows the magnitude. It has the standard unit (dB) for Matlab Bode diagram of the magnitude on the y axis. The second diagram shows the phase. It has the unit degree on the y axis.

The sensitivity function has a peak between 80 and 300 rad/sec which means that the model has to be quite accurate in this region in order for the regulator to work as expected.

5.1.1.2 Notch filter simulation example two

In this simulation example all poles of the closed loop system were placed in the discrete time equivalent of the real continuous pole in $p = -170$. A load disturbance in the shape of a step acted on the system. The initial value of the disturbance was zero and the final value was 30 ampere. The step time was 4.3 seconds. The fourth order model from the system identification was used to simulate the process. The implementation was done according to Figure 18. Figure 23 shows the reference signal and the output signal. Figure 24 shows the control signal. For the coefficients of the regulator and model polynomials see Appendix A.

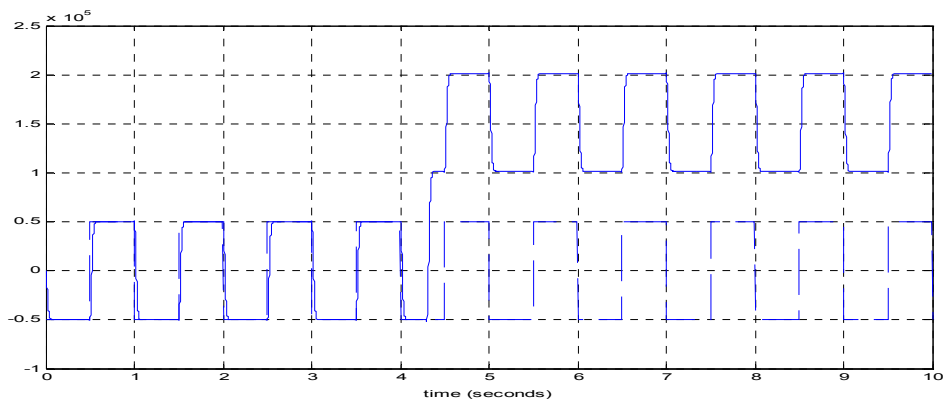


Figure 23 The reference signal (dashed) and the output signal (solid). The unit on the y axis is the same as the unit of the measured position.

A large stationary error can be seen in the position plot.

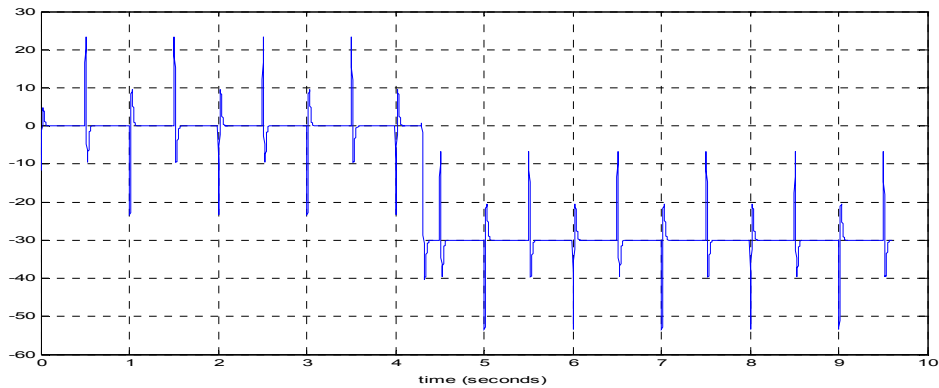


Figure 24 The control signal. The unit on the y axis is ampere.

5.1.1.3 Notch filter simulation example three

In this simulation example all poles of the closed loop system were placed in the discrete time equivalent of the real continuous pole in $p = -170$. A load disturbance in the shape of a step acted on the system. The initial value of the disturbance was zero and the final value was 30 ampere. The step time was 4.3 seconds. Integral action was added to cope with the load disturbance. This included adding of a new closed loop discrete time pole in the origin. The fourth order model from the system identification was used to simulate the process. The implementation was done according to Figure 18. Figure 25 shows the reference signal and the output signal. Figure 26 shows the control signal. For the coefficients of the regulator and model polynomials see Appendix A.

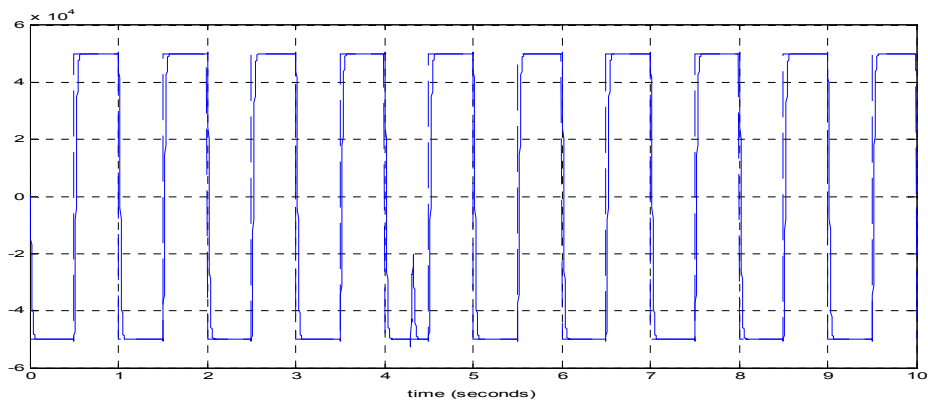


Figure 25 The reference signal (dashed) and the output signal (solid). The unit on the y axis is the same as the unit of the measured position.

The stationary error is effectively eliminated.

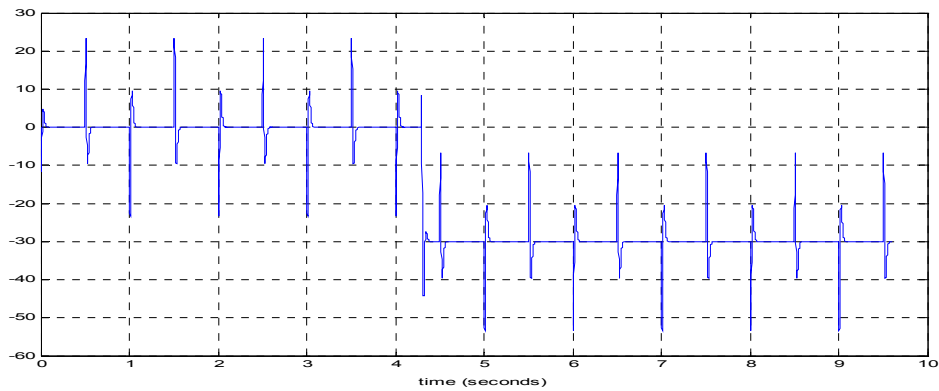


Figure 26 The control signal. The unit on the y axis is ampere.

A closer look on the output of the process when it is regulated with a notch filter regulator reveals that resonance oscillations can be excited by a load disturbance. Figure 26 and Figure 27 shows the same diagram as Figure 25 but in a different scale.

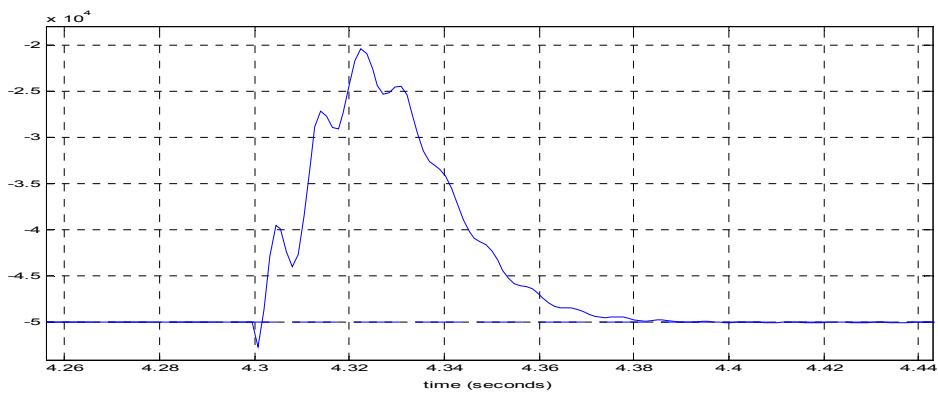


Figure 26 Resonance oscillations in the output signal

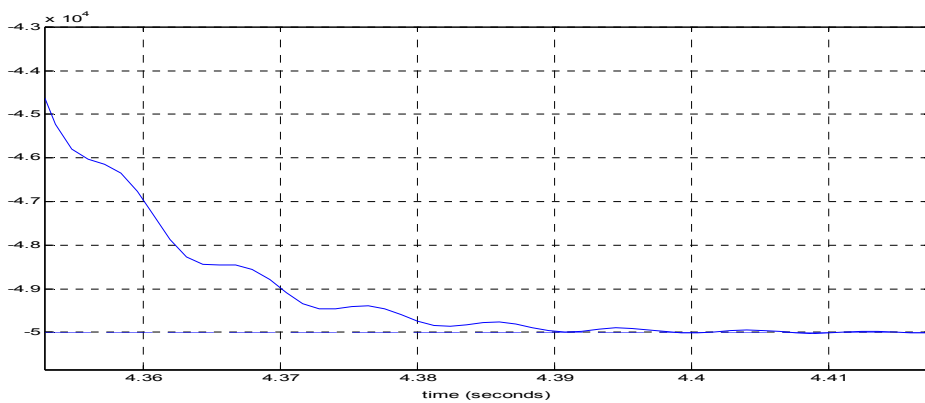


Figure 27 Resonance oscillations in the output signal

5.2.2 Active damping simulation example

In this simulation example all poles of the closed loop system were placed in the discrete time equivalent of the real continuous pole in $p = -170$. A load disturbance in the shape of a step acted on the system. The initial value of the disturbance was zero and the final value was 30 ampere. The step time was 4.3 seconds. Integral action was added to cope with the load disturbance. This included adding of a new closed loop discrete time pole in the origin. The fourth order model from the system identification was used to simulate the process. The implementation was done according to Figure 18. Figure 28 and Figure 29 shows the reference signal and the output signal. Figure 30 shows the control signal. Figure 31 show the sensitivity function. For the coefficients of the regulator and model polynomials see Appendix A.

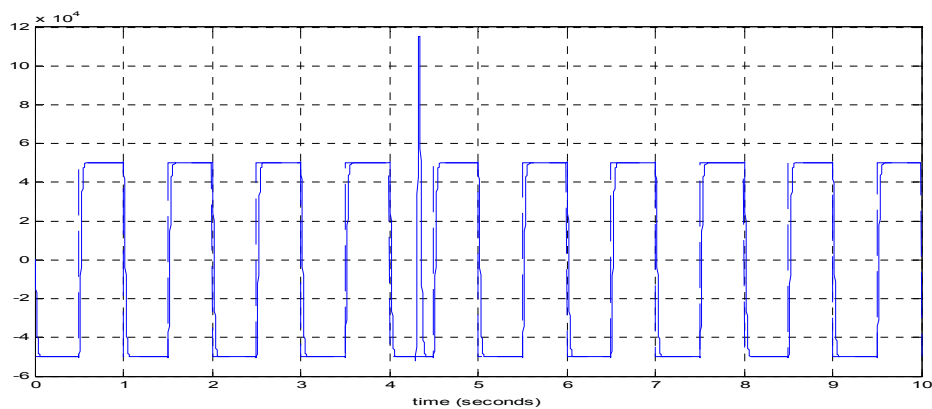


Figure 28 The reference signal (dashed) and the output signal (solid). The unit on the y axis is the same as the unit of the measured position.

The elimination of stationary errors is obviously slower for this regulator than for the notch filter regulator but a closer look reveals that it is much smoother (compare Figure 31 with Figure 26).

Figure 29 shows the same diagram as Figure 28 but in another scale. It shows that the load disturbance in this case does not excite any resonance oscillations.

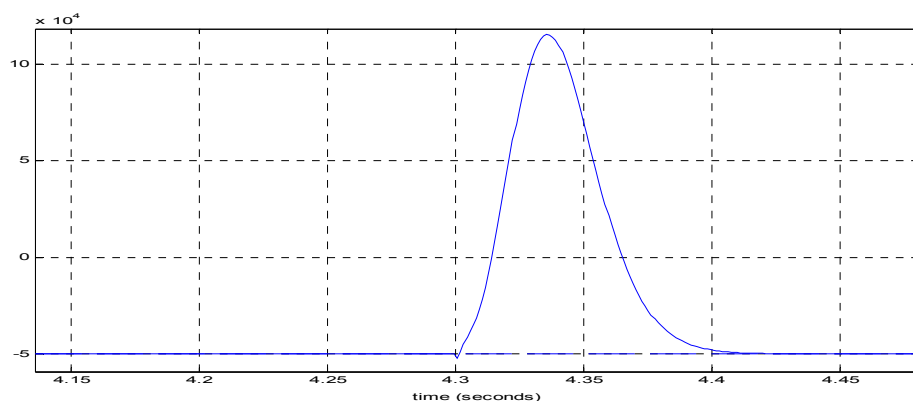


Figure 29 The same diagram as in Figure 28 but in a different scale

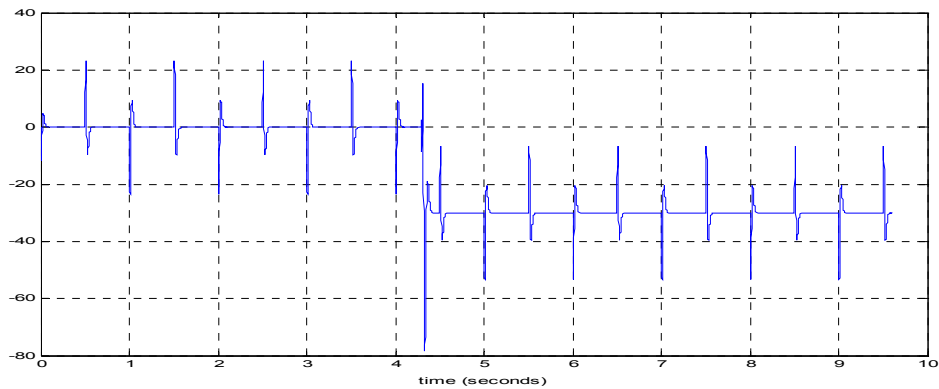


Figure 30 The control signal. The unit on the y axis is ampere.

Figure 31 shows the sensitivity function for the active damping regulator.

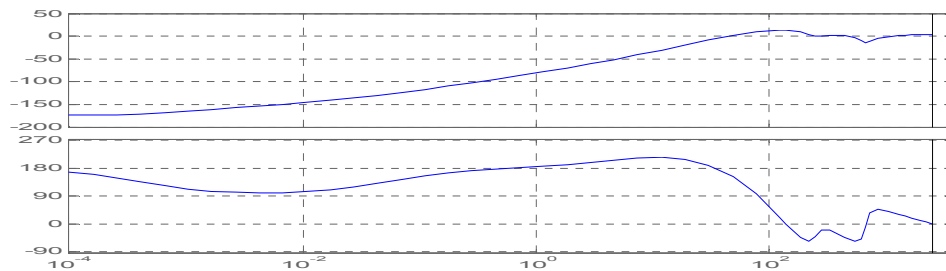


Figure 31 The sensitivity function for the active damping regulator in the simulation. The unit on the x axis is radians per second. The first diagram shows the magnitude. It has the standard unit (dB) for Matlab Bode diagram of the magnitude on the y axis. The second diagram shows the phase. It has the unit degree on the y axis.

The sensitivity function has a top at approximately the same place as the notch filter regulator. But it has a notch around the resonance frequency of the process.

5.2.3 LQG/LTR Simulation example

In the design of the regulator for this simulation a small direct term was removed from the model. The design was made in continuous time and then the regulator was sampled. In the loss function the Q matrix was chosen as the identity matrix of order four. R was chosen as one and N was chosen as a column vector of height four containing zeros.

The LTR method determined the covariance of the noises. RN was chosen as 1 and QN was chosen as 10000000000 times RN.

The names and definitions of the matrices in the loss function and the covariance constants were taken from Matlab Control System Toolbox commands Lqr and Kalman.

The fourth order model from the system identification was used to simulate the process. The simulation was implemented in Matlab Simulink according to Figure 38. For the model polynomials and the components of the Kalman filter gain and the state feedback gain see Appendix A.

Figure 32 and Figure 33 shows what impact the removal of a small direct term had on the process model.

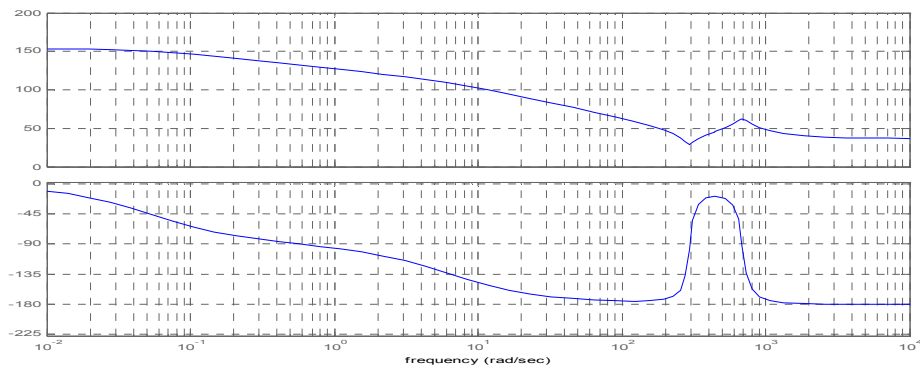


Figure 32 Model with direct term. The first diagram shows the magnitude. It has the standard unit (dB) for Matlab Bode diagram of the magnitude on the y axis. The second diagram shows the phase. It has the unit degree on the y axis.

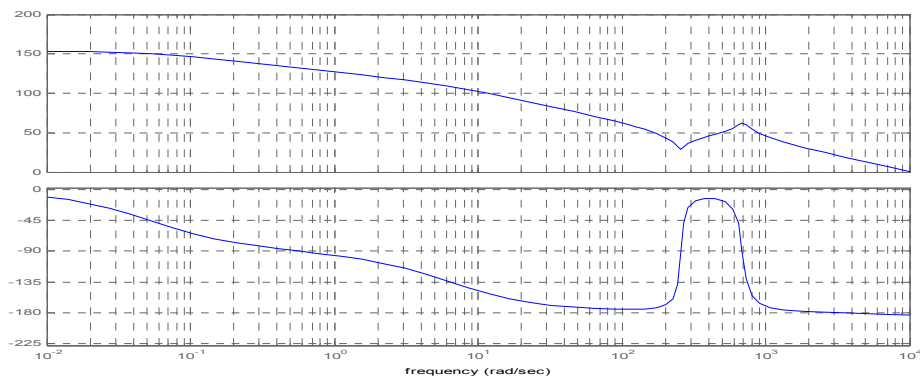


Figure 33 Model without direct term. The first diagram shows the magnitude. It has the standard unit (dB) for Matlab Bode diagram of the magnitude on the y axis. The second diagram shows the phase. It has the unit degree on the y axis.

Figure 34 shows the open loop transfer function of the LQ regulated system

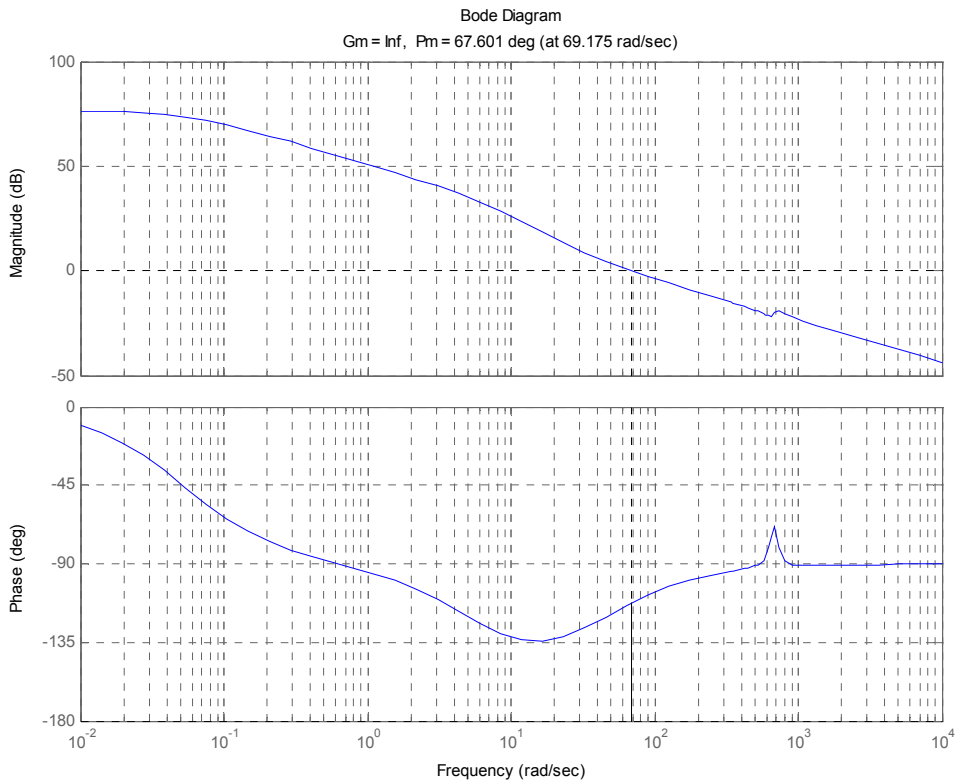


Figure 34 Open loop transfer function of the LQ regulated system. The amplitude and the phase margins are indicated at the top of the Figure.

The good margins promised by theory can be seen in practice in Figure 34

Figure 35 shows the sensitivity function of the LQ regulated system. The sensitivity is less than one.

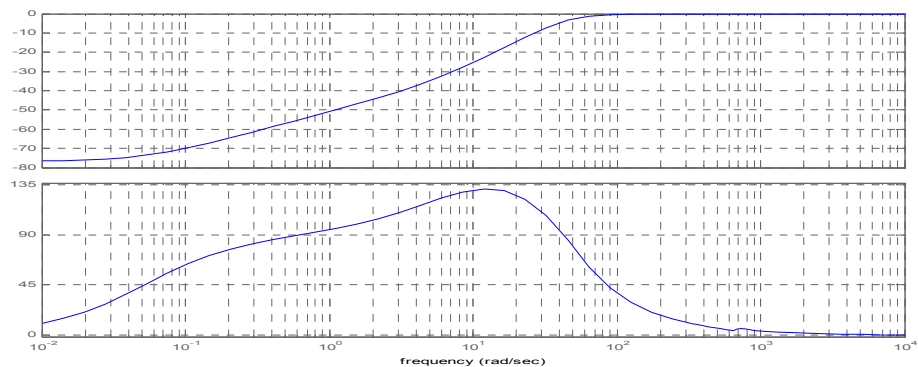


Figure 35 Sensitivity function for the LQ regulator. The first diagram shows the magnitude. It has the standard unit (dB) for Matlab Bode diagram of the magnitude on the y axis. The second diagram shows the phase. It has the unit degree on the y axis.

Thereafter loop transfer recovery (LTR) was obtained by choosing observer gain K according to the theory. Figure 36 shows the recovered open loop regulated system.

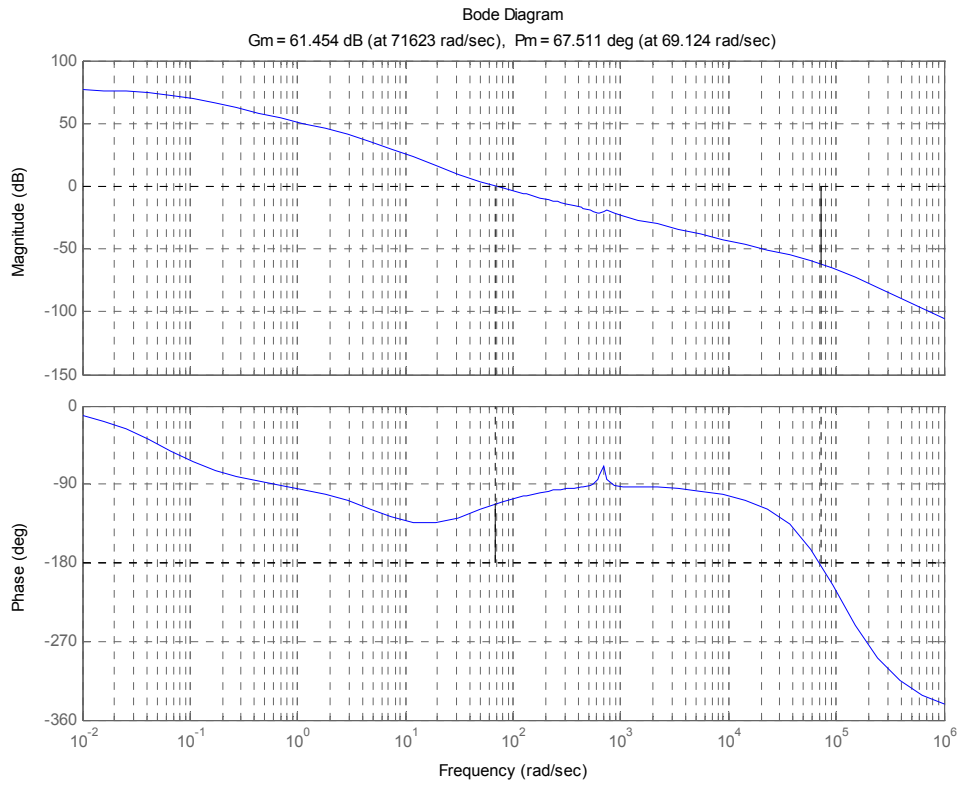


Figure 36 Recovered transfer function. Amplitude and phase margins are indicated at the top of the Figure.

Figure 37 shows the recovered sensitivity function.

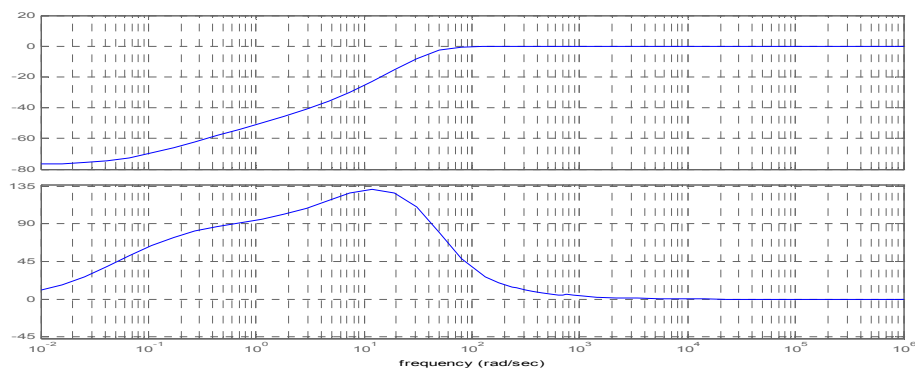


Figure 37 Recovered sensitivity function. The first diagram shows the magnitude. It has the standard unit (dB) for Matlab Bode diagram of the magnitude on the y axis. The second diagram shows the phase. It has the unit degree on the y axis.

The regulator was implemented and simulated in Matlab Simulink.

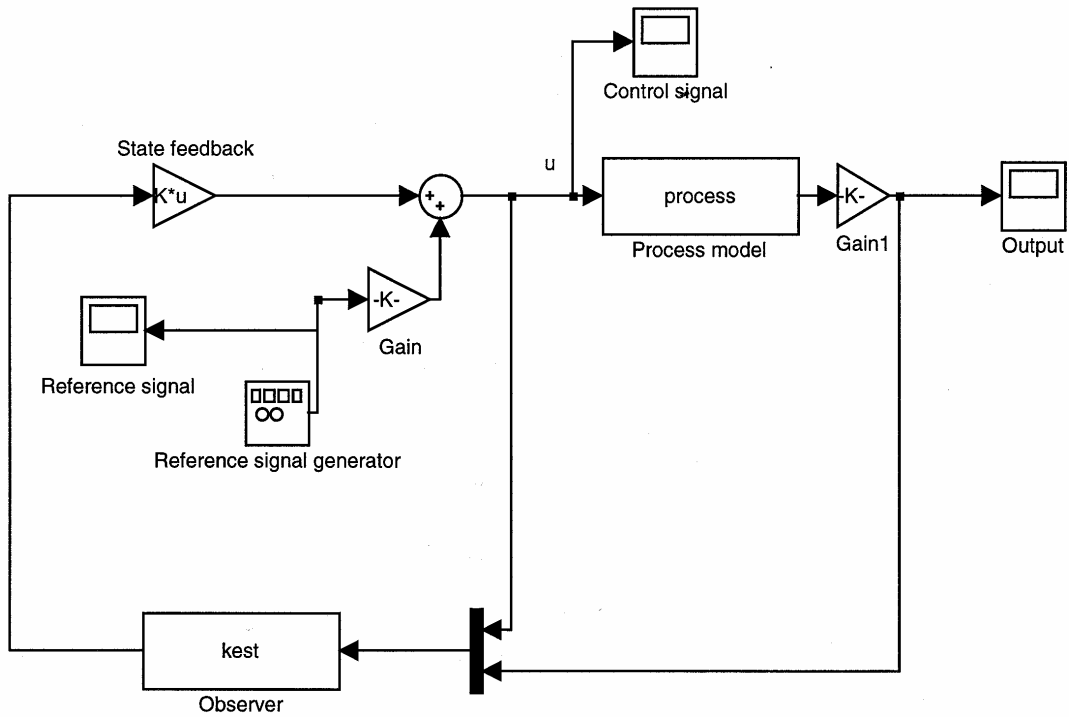


Figure 38 Matlab Simulink implementation of the LQG/LTR regulator

The fourth order model from the system identification was used to simulate the process.

Figure 39, Figure 40 and Figure 41 shows the reference signal used in the simulations and the output signal.

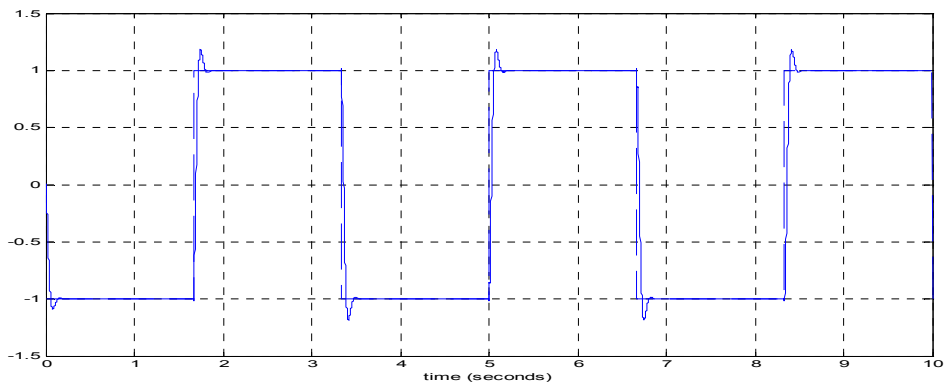


Figure 39 The reference signal (dashed) and the output signal (solid). The unit on the y axis is the same as the unit of the measured position divided by 1000.

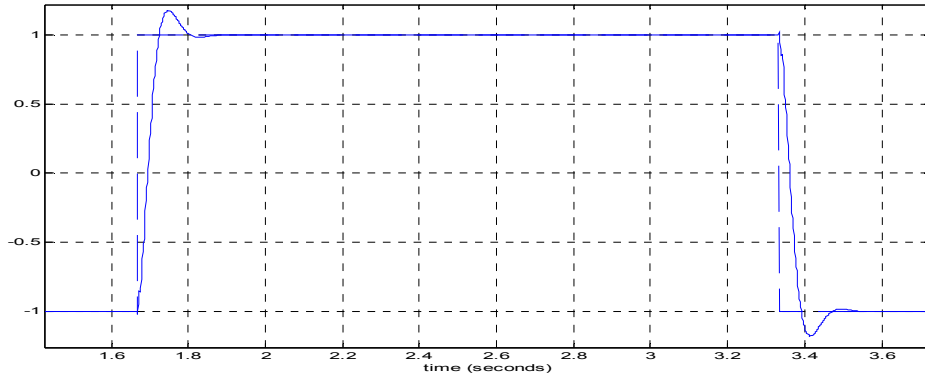


Figure 40 The same as Figure 39 but in a shorter time interval

Figure 42 shows the same diagram as Figure 41 but in a different scale.

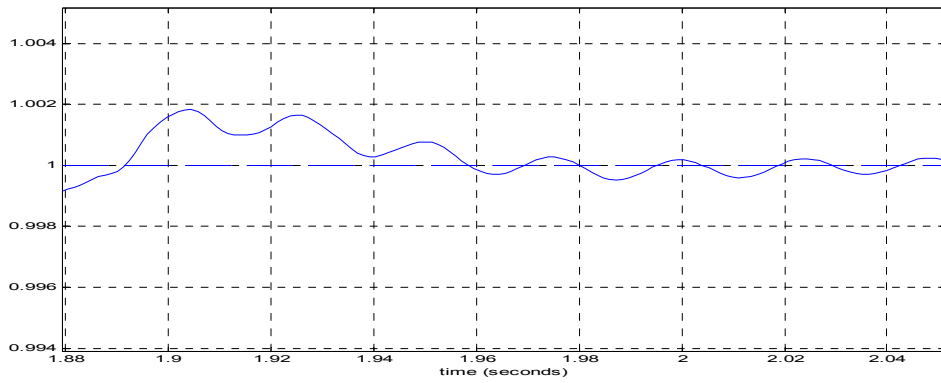


Figure 41 The same as Figure 39 but in a shorter time interval

Figure 42 shows the control signal.

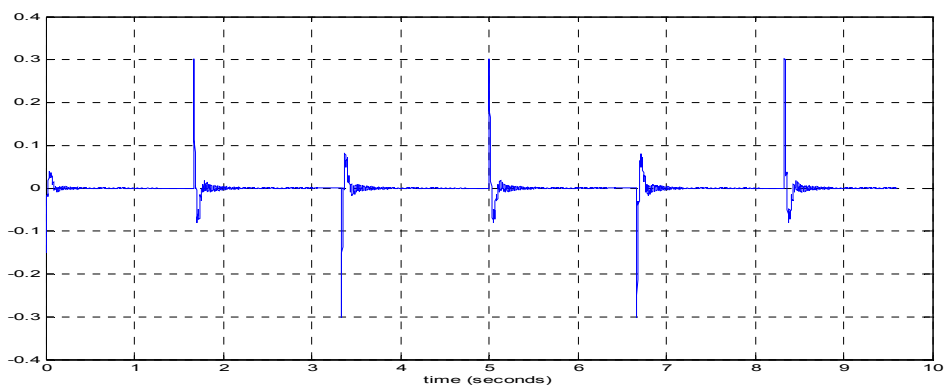


Figure 40 The control signal. The unit on the y axis is ampere.

5.3 Implementation

Figure 43 shows a bode diagram of the second order model used in the regulator design. For the coefficients of the regulator and model polynomials see Appendix A.

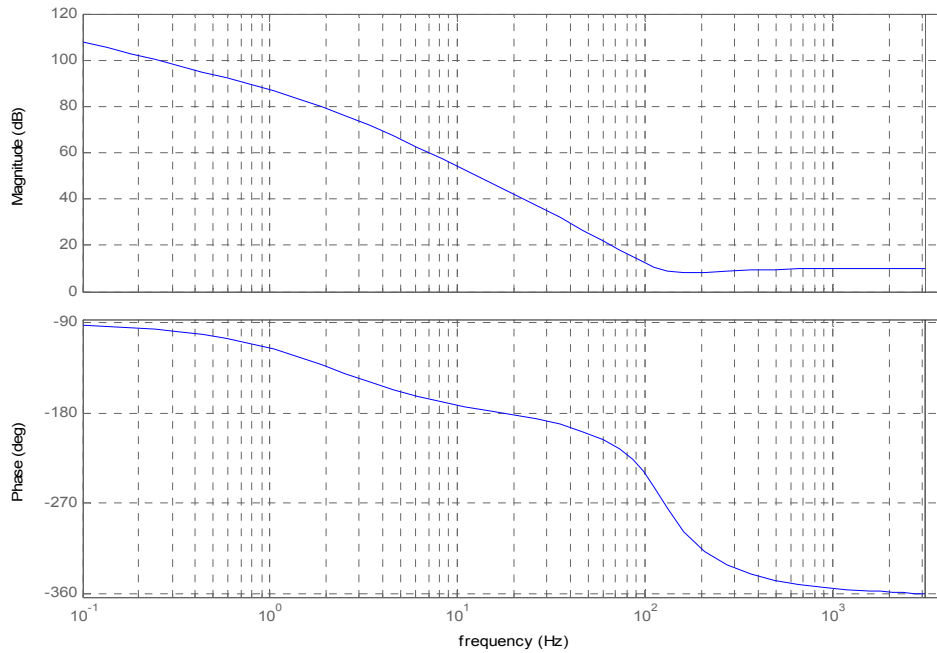


Figure 43 Bode diagram of the model used for regulator design.

5.4 Experimental results

A reference signal with steps with an interval of 0.72 seconds was used. Figure 44 shows a plot of the reference signal and the position when the steps had amplitude of 10000 units. Figure 45 and Figure 46 are the same as Figure 44 but in a different scale. Figure 47 shows the difference between the actual position and the reference signal. Figure 48 shows the control signal.

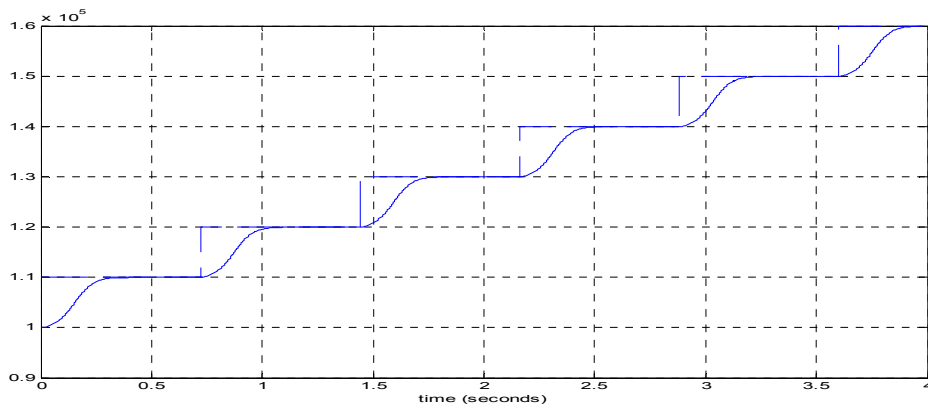


Figure 44 The reference signal (dashed line) and position (solid line). The unit on the y axis is the unit for the measured position.

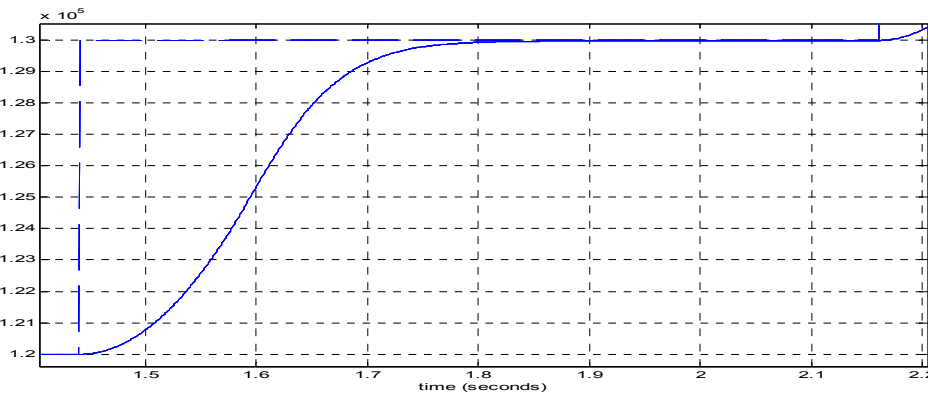


Figure 45 The same Figure as in Figure 43 but in a different scale.

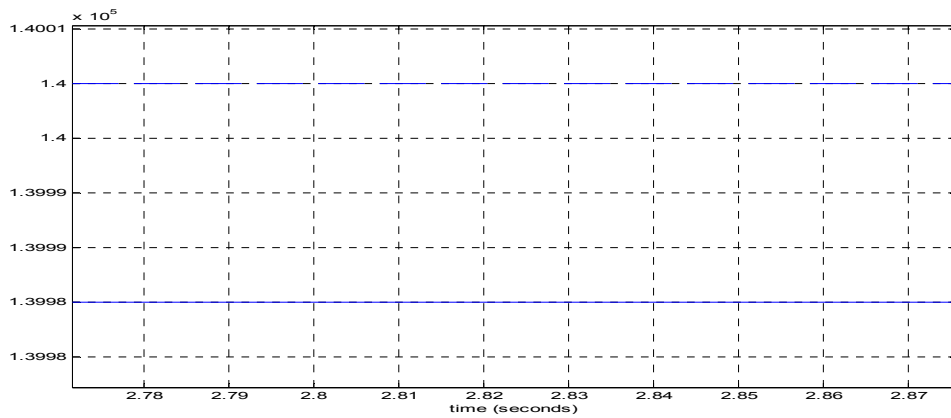


Figure 46 The same Figure as in Figure 43 but in a different scale. A stationary error about 20 units can be seen.

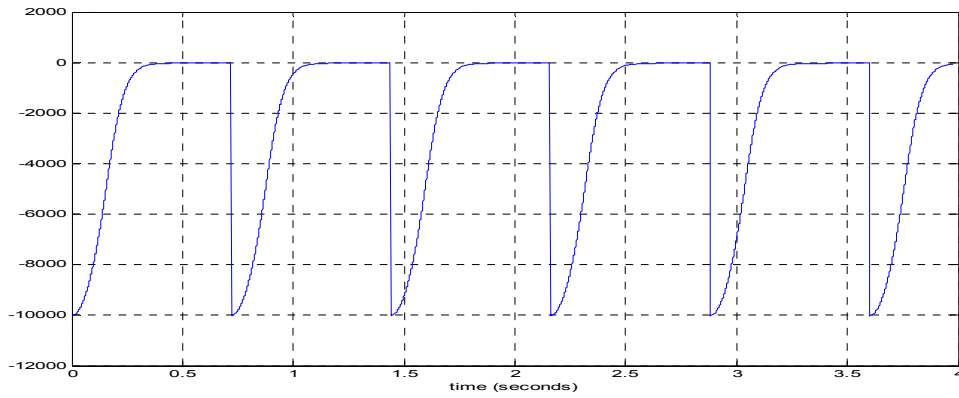


Figure 47 The difference between the actual position and the reference signal.

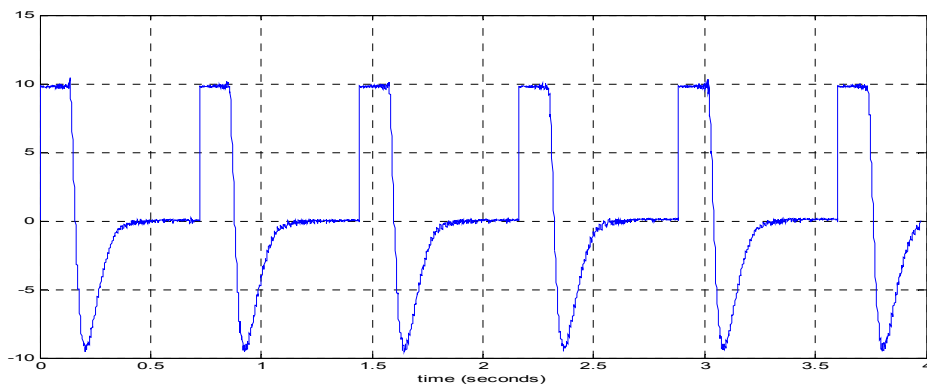


Figure 48 The control signal. The unit on the y axis is ampere. The control signal is saturated at 10 ampere.

Figure 49 shows the power spectral density of the output signal (position).

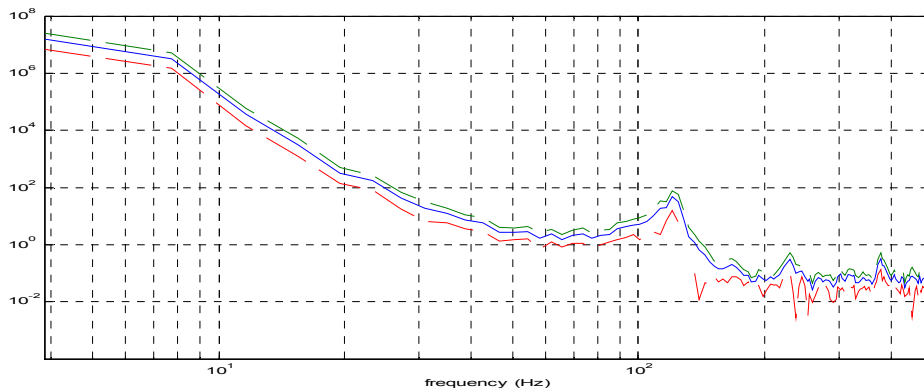


Figure 49 The power spectral density of the output signal with 95% confidence interval. Resonance oscillations are revealed in the diagram.

6 Discussion

The first part of the work was system identification. The model obtained was used to design and test different regulators in Matlab Simulink. The identified model was used to simulate the process. During this phase the difference in low frequency gain between the real process and the model was not discovered. The simulations are of general interest because the model looks like a typical process containing a resonance frequency and the regulators are designed to cope with this. Further the validation suggests that the model was quite accurate in the frequency region around the resonance frequency.

The sensitivity function that was plotted for the different regulators is a measure of the regulator's sensitivity to small model errors. Frequency regions where the sensitivity function is greater than one could be considered to be regions where the model must be very accurate. The simulations showed how the regulators react to reference signals and load disturbances. The different regulators tested have different advantages and shortcomings.

- The notch-filter design works well if the model accuracy is good and there are no load disturbances. If an integrator is added stationary error can be avoided, but the fact that load disturbances excite the oscillatory modes is a problem.
- The active damping approach is good when model accuracy is good. The regulator is capable of coping with load disturbances. If an integrator is added stationary error can be avoided.
- LQG/LTR design is robust and is therefore preferable when model accuracy is less good.

In every design case it has to be considered if it is necessary to damp resonance frequencies at all. As it turned out in this case, the power spectral density of the low frequencies in the output signal is greater than the power spectral density of the resonance frequency by a factor 10000. This means that resonance oscillations will cause small problems.

During the implementation phase the difference in low frequency behavior between the process and the model was discovered. A new identification of the model was made. An input signal that excited the low frequencies more than before was used. The first approach in the implementation was to use a second order model. The results of the implementation were good. Plotting the power spectral density revealed that resonance oscillations were present in the output signal but they were of low amplitude. This made it unnecessary to increase the model order to four in order to cope with resonance oscillations. When the regulator was tested it was discovered that saturation of the control signal did not cause big problems.

There are some advantages of the design of the implemented regulator as compared to the PID design that is used at present. The behavior of the closed loop system is design parameters of the regulator. This means that the regulator is designed to force the system to react in a specified way to reference signals. In this particular case it is crucial to have a well damped closed loop system and this is easy to achieve using the methods suggested in this thesis. The PID design uses a totally different approach and it can be hard to know how to tune the parameters to get a well damped closed loop system. Often it is necessary to use a reference signal that is not just a step for the PID-regulated process because the closed loop system is poorly damped. This is not always necessary if the process is regulated by a regulator that specifies the behavior of the closed loop system, and thus the damping of the system.

The regulator was implemented to demonstrate the possibility of using a model-based regulator for this process. It was not optimized. If a regulator of this type was to be used in an industrial application there are some issues that have to be considered.

- The regulated process had a stationary error. One possible way of coping with this is to add an integrator to the regulator. This might cause some stability problems and some kind of anti-windup might be necessary.
- It is possible to write a computer program that creates the regulator but an accurate model of the process has to be at hand. System identification can be made but is likely to be time consuming to an engineer that is not specialized in control theory. Therefore the system identification also has to be automatic to some extent. There are some problems with this. For example it is hard to know how to choose the input signal that gives a necessary excitation of the process. And it is hard to evaluate the model in an automatic way.
- The design assumes that the process is linear. In practice the region of linearity is limited. What happens outside the linearity region has to be considered.

Further work on this topic would be to develop software for optimal system identification and for regulator design that could be used in an industrial application. The problems mentioned above should be considered.

7 References

Johansson, Rolf. *System Modeling and Identification*, Prentice Hall, Englewood Cliffs, New Jersey, 1993.

Glad, Torkel och Ljung Lennart *Reglerteori, flervariabla och olinjära metoder*, studentlitteratur, Lund, Sweden, 1997.

Liljegren, Dan. *Tuning of Gain Parameters in Commercial Servo Controllers*, Master Thesis, ISSN 0280-5316, Dept. Automatic control, Lund university, Lund, Sweden, 1998.

Åström, Karl J. och Wittenmark, Björn. *Computer controlled systems*, Prentice Hall, New Jersey, 1997.

The MathWorks, Matlab V6.1, the System Identification Toolbox for Matlab and the Control System Toolbox for Matlab.

Appendix A - Constants of models and regulators

The fourth order model from the system identification is given as a transfer function. The numerator, B, and the denominator, A, are presented in the form

$$z^4 \times b_0 + z^3 \times b_1 + z^2 \times b_2 + z \times b_3 + b_4$$
$$z^4 \times a_0 + z^3 \times a_1 + z^2 \times a_2 + z \times a_3 + a_4.$$

The coefficients are the following:

a0 =1
a1 =-3.26169485236
a2 =4.40473220273
a3 =-3.01973756569
a4 =0.876700504779

b0 =-68.6810549537
b1 =295.714159865
b2 =-366.144180865
b3 =144.131579054
b4 =8.3061166878

The second order model used for regulator design of the regulator that was implemented is presented in the same way. The coefficients are the following:

a0 =1
a1 =-1.99752290806
a2 =0.997522907152

b0 =2.94809451537
b1 =-6.26019760955
b2 =3.38519886215

The R, S and T polynomials of the notch filter regulator without integral action are presented in the same way as the polynomials of the model. The coefficients are

r0 =1.02593787796
r1 =-2.16970438537
r2 =1.55380355748
r3 =-0.360314765171

s0 =0.000377656953301
s1 =-0.000840942628864
s2 =0.000792575892961
s3 =-0.000319429398435

t0 =0.00008702013171
t1 =-0.000181412248754
t2 =0.0001669399146
t3 =-0.0000626858985894

The R, S and T polynomials of the notch filter regulator with integral action are presented in the same way as the polynomials of the model. The coefficients are

r0 =1.28219032882

r1 =-3.27302870483
r2 =2.91990580759
r3 =-0.89807689481
r4 =-0.0309905367608

s0 =0.00410870696447
s1 =-0.0130104892442
s2 =0.0172268520272
s3 =-0.0115862212766
s4 =0.00327101342815

t0 =8.702013171e-05
t1 =-0.000181412248754
t2 =0.0001669399146
t3 =-0.0000626858985894

The R, S and T polynomials of the active damping regulator with integral action are presented in the same way as the polynomials of the model. The coefficients are

r0 =1.08423093284
r1 =-3.02359731536
r2 =2.8630365165
r3 =-0.888585119074
r4 =-0.0350850149029

s0 =0.00122640709134
s1 =-0.00712169064842
s2 =0.0142870865818
s3 =-0.0120944367783
s4 =0.00370318061155

t0 =0.00008702013171
t1 =-0.000212884928836
t2 =0.000173599648858
t3 =-0.0000471879937651

The R, S and T polynomials of the pole placement regulator that was implemented are presented in the same way as the polynomials of the model. The coefficients are

r0 =0.7957
r1 =-0.6519

s0 =0.0693
s1 =-0.06834

t0 =0.02314
t1 =-0.02219

The components of the LQG/LTR regulator Kalman filter gain are

k0 =1595961.70903
k1 =35973.9217414
k2 =101.277680112

$k_3 = 0.18396072437$

The components of the LQG/LTR regulator state feedback gain are

$k_0 = 4.11057563628$

$k_1 = 0.564387692912$

$k_2 = 7.11041651818$

$k_3 = 0.999852967082$

Appendix B - Computer programs

Matlab file estmod.m

```
clear all;
clf;

% Load data set to be used for estimation
data1;

% Do estimation and save the results
s=100;
[Sn,r]=dordpo(u,y,s);
n=4;
[A,B,C,D,K,ye,x0]=dmodpo(u,y,n,r,1);
sparA=A;
sparB=B;
sparC=C;
sparD=D;
sys=ss(A,B,C,D,Ts);
G=tf(sys);
M=idss(sys);

% Load data set to be used for validation
data2;

% Validation with some plots
[ye,x0,B,D] = dac2bdx(sparA,sparC,u,y);

e=zeros(size(u));
UE = [u e];
ye=idsim(UE,M,x0);

Figure(1);
plot([y,ye]);

Figure(2);
resid(M,z,1/Ts)

Figure(3)
bode(sys)

Figure(4)
d=y-ye;
plot(d)
```

Matlab file notchfilter.m

```
s=tf('s');
```

```

z=tf('z');

% New Sampling time
Ts=0.0012;

% Give the model the new sampling time
G=d2d(G,Ts);

%specifications
zi=1;    % desired relative damping in Ac
wc=170;  % desired velocity in Ac
zin=zi;  % in Ao
wcn=wc;  % in Ao

% transfer functions
Gd=1/(s+wcn);    % poles in Ao
sysm=ss(Gd);
Hd=tf(c2d(Gd,Ts));
Aohat=zpk([pole(Hd)],[],1,Ts);

Gm=1/(s^2+2*zi*wc*s+wc^2)^2; % desired poles
sysm=ss(Gm);
Hm=tf(c2d(Gm,Ts));
Ac=zpk([pole(Hm)],[],1,Ts);

AoAc=Ac*Aohat;

% determine Aminus and Aplus
p=pole(G);
k=1;
l=1;
for i=1:length(p)
    if imag(p(i))==0
        pminus(k)=p(i);
        k=k+1;
    else
        pplus(l)=p(i);
        l=l+1;
    end
end
Aminus=zpk([pminus], [], 1, Ts);
Aplus=zpk([pplus], [], 1, Ts);

% extract polynomial
[pAminus d]=tfdata(Aminus,'v');
[pAo d]=tfdata(Aohat,'v');
[pAoAc d]=tfdata(AoAc,'v');
[pAc d]=tfdata(Ac,'v');
[pB pA]=tfdata(G,'v');
R=0;
S=0;

```

```

T=0;

%R,S och T
[pR,pShat] = dab(pAminus,pB,pAoAc); % solving the Diophantine equation
S=tf(pShat,1,Ts)*Aplus;
[pS d]=tfdata(S,'v') ;
T=polyval(pAc,1)/polyval(pB,1)*AoAc*Aplus;
[pT d]=tfdata(T,'v');

% polynomial for plots
B=tf(pB,1,Ts);
A=tf(pA,1,Ts);
S=tf(pS,1,Ts);
R=tf(pR,1,Ts);
Hc=B*S/(R*A); % loop transfer function
karpol=AoAc*Aplus;
Hv=A*R/karpol; % sensitivity function

% integral action
x0=0;
y0=(1+x0)*polyval(pR,1)/polyval(pB,1)
R=(z+x0)*R-y0*B;
[pR,d]=tfdata(R,'v');
S=(z+x0)*S+y0*A;
[pS,d]=tfdata(S,'v');
zero(R)

% Plotting the sensitivity function
Figure(5);
bode(Hv);
title('sensitivity function for notchfilter')
xlabel('frequency (rAoAc/sec)')
grid on

```

Matlab file activedamping.m

```

s=tf('s');
z=tf('z',Ts);

Ts=0.0012; % New sampling time

% Give the model the new sampling time
G=d2d(G,Ts);

%specifications
zi=1; % desired relative damping in Ac
wc=170; % desired velocity in Ac
zin=zi; % in Ao
wcn=wc; % in Ao

% transfer functions

```

```

Gd=1/(s^2+2*zin*wc*n*s+wc^n^2)/(s+wcn); % poles in Ao
sysm=ss(Gd); %
Hd=tf(c2d(Gd,Ts)); %
Ao=zpk([pole(Hd)'],[],1,Ts); % Ao

Gm=1/(s^2+2*zi*wc*s+wc^2)^2; % desired poles
sysm=ss(Gm); %
Hm=tf(c2d(Gm,Ts)); %
Ac=zpk([pole(Hm)'],[],1,Ts); % Ac

AoAc=Ac*Ao;

% extract polynomial
[pAo d]=tfdata(Ao,'v');
[pAoAc d]=tfdata(AoAc,'v');
[pAc d]=tfdata(Ac,'v');
[pB pA]=tfdata(G,'v');
R=0;
S=0;
T=0;

%R,S och T
[pR,pS] = dab(pA,pB,pAoAc); % solving the Diophantine equation

T=polyval(pAc,1)/polyval(pB,1)*Ao;
[pT d]=tfdata(T,'v');

% polynomial for plots
B=tf(pB,1,Ts);
A=tf(pA,1,Ts);
S=tf(pS,1,Ts);
R=tf(pR,1,Ts);

% Integral action
x0=0;
y0=(1+x0)*polyval(pR,1)/polyval(pB,1)
R=(z+x0)*R-y0*B;
[pR,d]=tfdata(R,'v');
S=(z+x0)*S+y0*A;
[pS,d]=tfdata(S,'v');

Hc=B*S/(R*A); % loop transfer function
karpol=AoAc*z;
Hv=A*R/karpol; % sensitivity function

% plotting the sensitivity function
Figure(5);
bode(Hv);
title('sensitivity function for active damping')
grid on

```

Matlab file LQGLTR

```
% New sampling time
Ts=0.0012;

% Making the model continuous and giving it the new sampling time
sys=d2c(G);
G=d2d(G,Ts);

% Modifying the model a bit
syssim=ss(sys);
syssim.d=0;
syssim=tf(syssim);
syssim=syssim*0.001;
syssim=ss(syssim);

% Choosing R1 and R2 for LTR
alpha=10000000000;
R2=1;
R1=alpha*R2;
H=1;

% Determine estimator and regulator
[KEST,K,P]= kalman(syssim,R1,R2,0);
EST=ss(syssim.a-K*syssim.c,[syssim.b K],eye(4),zeros(4,2));
[L,S,E] = lqr(syssim.a,syssim.b,1*eye(4),1,0*ones(4,1));
contr = reg(syssim,L,K);

% Plotting some plots
Figure(1)
help=ss(syssim.a-syssim.b*L-K*syssim.c,K,L,0);
ols=tf(help)*tf(syssim);
margin(ols)

sens2=1/(1+ols)
Figure(3)
bode(sens2)
grid on;
title('sensitivity function')
xlabel('frequency (rad/sec)')

x0=[0,0,0,0];
s=tf('s');
lqsys=ss(syssim.a,syssim.b,L,0);
sens1=1/(1+lqsys);
Figure(2)
margin(lqsys)

Figure(4)
bode(sens1)
grid on;
```

```
title('sensitivity function')
xlabel('frequency (rad/sec)')
```

```
process=ss(G);
```

```
% Making the regulator and estimator discrete
contr=c2d(contr,Ts);
kest=c2d(EST,Ts);
```

Implementation in C

```
#include <bur/plctypes.h> /* IEC data types */

typedef struct ncCYCUSDAT_typ {
    USINT cnt;
    USINT reserve;
    USINT data[16];
} ncCYCUSDAT_typ;

_LOCAL BOOL          init_da, steg, stepsequense;
_LOCAL BOOL          ctrlon;
_GLOBAL BOOL        nw_ok;
_GLOBAL BOOL        alloc_ok;
_GLOBAL UDINT       todrv1_obj;
_GLOBAL UDINT       frdrv1_obj;
_LOCAL ncCYCUSDAT_typ* p_todrv1da;
_LOCAL ncCYCUSDAT_typ* p_frdrv1da;
_LOCAL REAL         current;
_LOCAL DINT         s_act1;
_GLOBAL BOOL        access_da;
_LOCAL REAL         r0, r1, s0, s1, t0, t1;
_LOCAL REAL         sr0, sr1, ss0, ss1, ss2, st0, st1;
_LOCAL REAL         position;
_LOCAL REAL         ref, safelimit, stepl, kvot;
_LOCAL INT          counter;

/*****init*****/
**/
void _INIT cyc_mon_init(void)
{
    r0 =0.7957;
    r1 =-0.6519;

    s0 =0.0693;
    s1 =-0.06834;

    t0 =0.02314;
    t1 =-0.02219;

    sr0=0;
    sr1=0;
}
```



```

    ss0=0;
    ss1=0;

    st0=0;
    st1=0;

    kvot=1.0105;
    position=0;
    init_da = 0;
    ctrlon=0;
    safelimit=5;
    steg=0;
    stepl=1000;
}

void _CYCLIC cyc_mon_cyclic(void)
{
    /*Cyclic User Data*/
    if ( init_da == 0 && access_da == 1 )
    {
        p_todrv1da = (ncCYCUSDAT_typ*)todrv1_obj;
        p_frdrv1da = (ncCYCUSDAT_typ*)frdrv1_obj;
        init_da = 1;
    }

    if ( (nw_ok == 1) && (alloc_ok == 1) )
    {
        /******reading
position******/
        s_act1 = *(DINT *) &p_frdrv1da->data[0];
        position=s_act1;

        /******taking a
step******/
        if (steg==1)
        {
            ref=ref+stepl;
            steg=0;
        }

        /******regulator*****
******/
        st0=ref*kvot;
        ss0=position;

        if(ctrlon==1)
        {
            current=(t0*st0+t1*st1-(s0*ss0+s1*ss1)-(r1*srl))/r0;
        }

        else
        {
            current=0;
        }
    }
}

```

```

        /*****saturate and send the
current*****/
        p_todrvlda->cnt++;
        if (current >= safelimit)
            current=safelimit;
        if (current <= -safelimit)
            current=-safelimit;
        memcpy(&p_todrvlda->data[0],&current,4);
    }

    /*****update regulator
state*****/
    srl=current;

    ssl=ss0;

    stl=st0;

    /*****step
sequense*****/
    if(stepsequense==1)
    {
        counter=counter+1;
        if(counter==600)
        {
            counter=0;
            steg=1;
        }
    }
    /*****
*****/

    /* End of Cyclic User Data*/
}

```

Appendix C - Data sets

Figure 49 to Figure 58 shows all data sets used in this thesis.

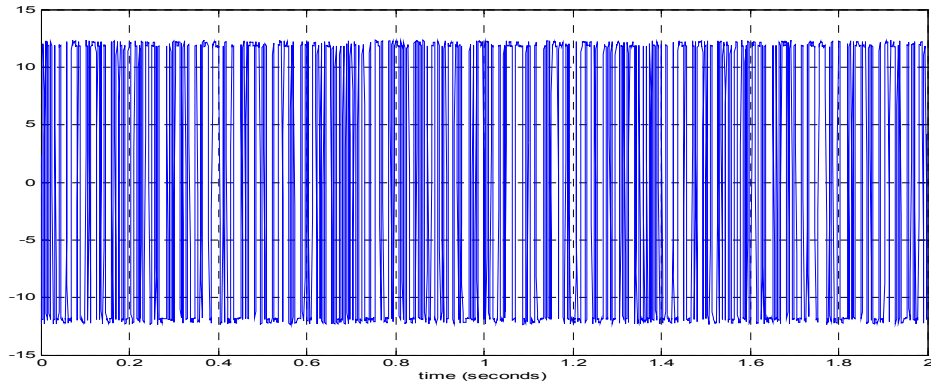


Figure 50 input signal of data1

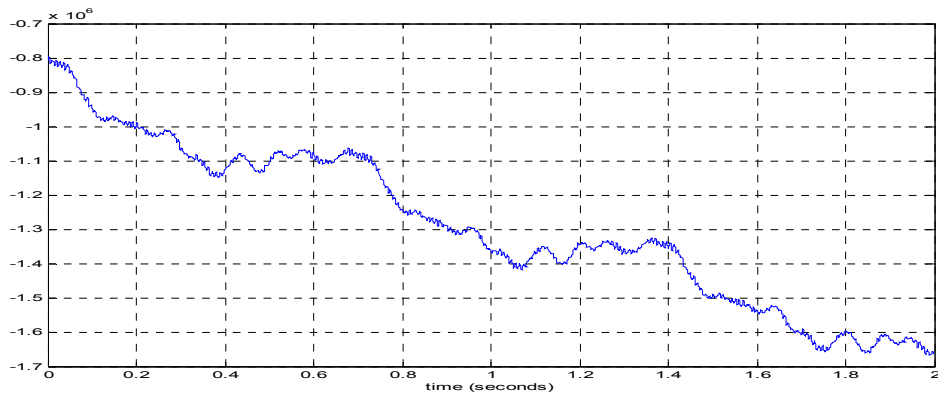


Figure 51 output signal of data1

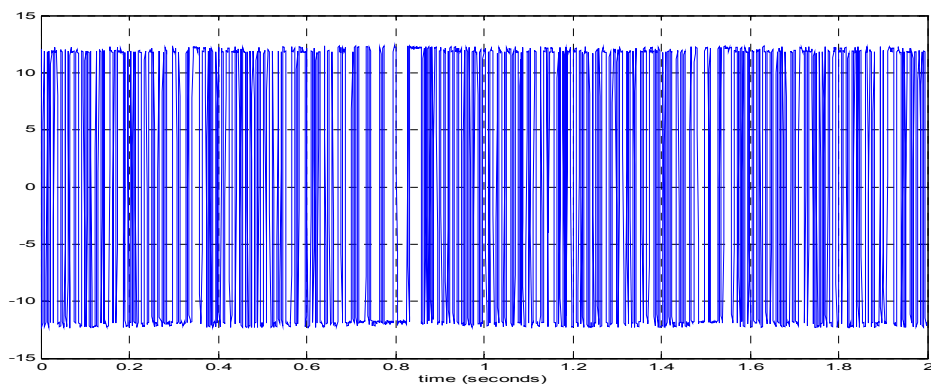


Figure 52 input signal of data2

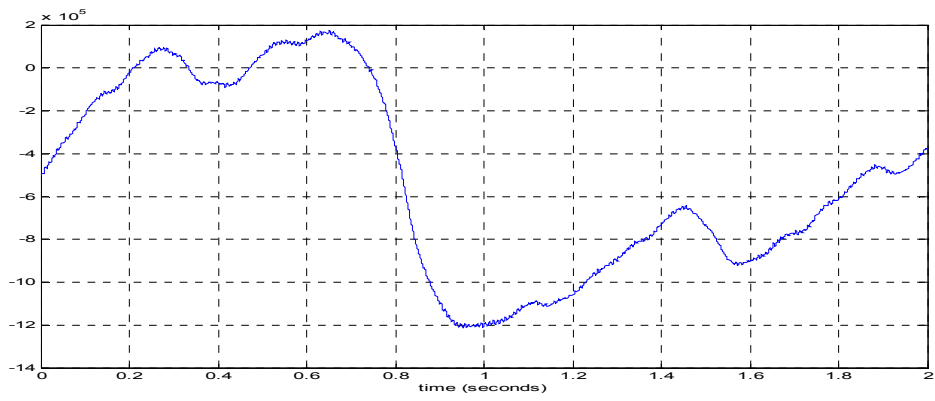


Figure 53 output signal of data2

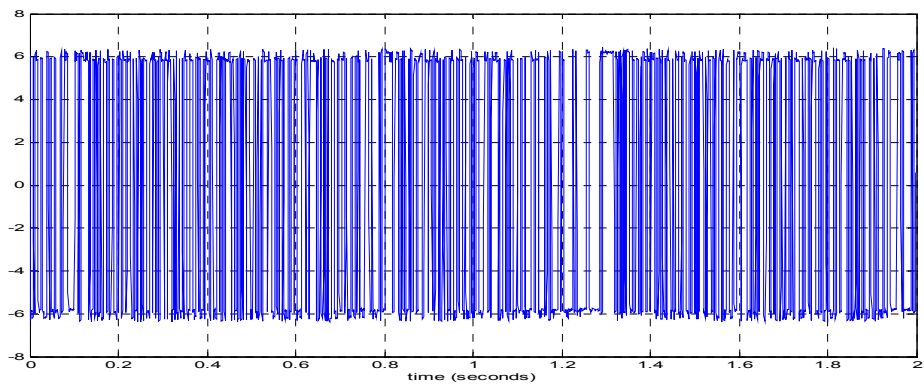


Figure 54 input signal of data3

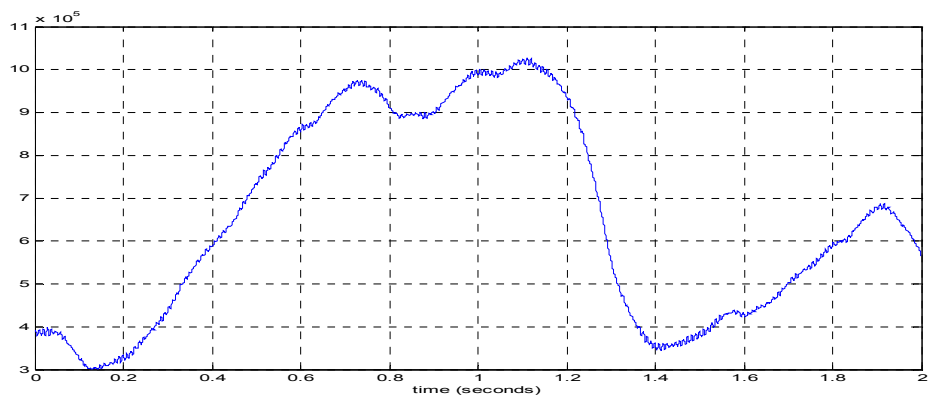


Figure 55 output signal of data3

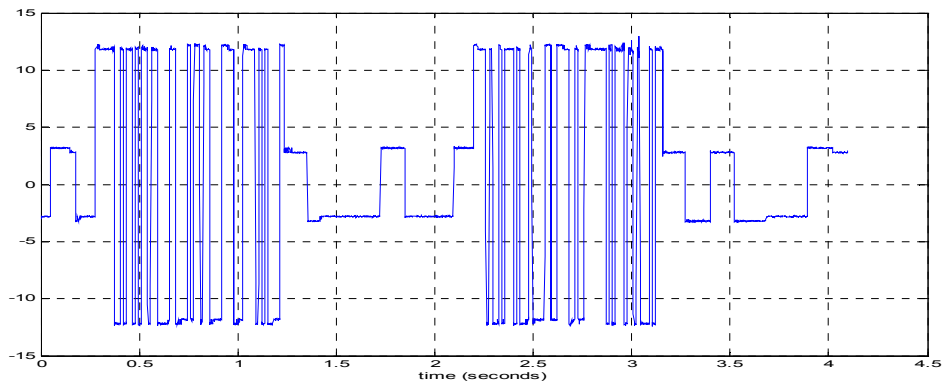


Figure 56 input signal of data4

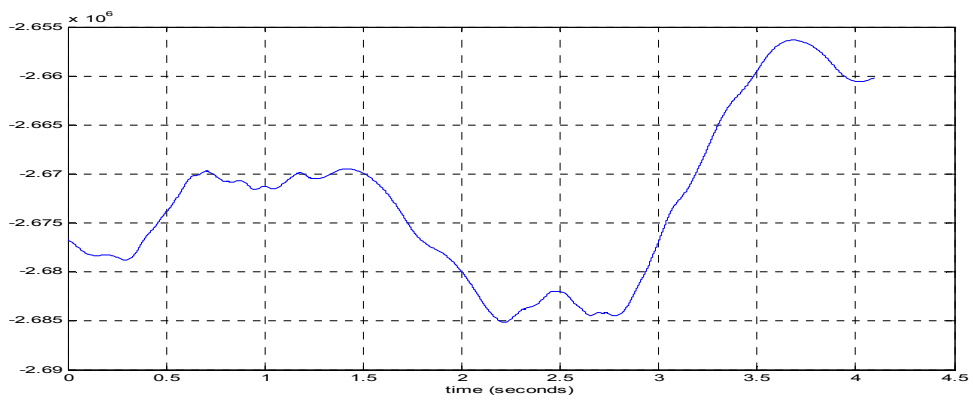


Figure 57 output signal of data4

