Development of a Motor Control Algorithm Used in a Shift-by-Wire System

Master's thesis
performed in **Vehicular Systems**
Performed for **DaimlerChrysler AG**

by

Daniel Gullberg

Reg nr: LiTH-ISY-EX-3305-2003

24th November 2003

Development of a Motor Control Algorithm Used in a Shift-by-Wire System

Master's thesis

performed in **Vehicular Systems**,
Dept. of Electrical Engineering
at **Linköpings universitet**

Performed for **DaimlerChrysler AG**
by **Daniel Gullberg**

Reg nr: LiTH-ISY-EX-3305-2003

Supervisor: **Michel Willems**
DaimlerChrysler AG, Stuttgart
Anders Fröberg
Fordonssystem, ISY, Linköpings Universitet

Examiner: **Professor Lars Nielsen/Anders Fröberg**
Linköpings Universitet

Linköping, 24th November 2003

Avdelning, Institution
Division, Department

Vehicular Systems,
Dept. of Electrical Engineering
581 83 Linköping

Datum
Date

24th November 2003

Språk
Language

☐ Svenska/Swedish
☒ Engelska/English
☐ __________

Rapporttyp
Report category

☐ Licentiatavhandling
☒ Examensarbete
☐ C-uppsats
☐ D-uppsats
☐ Övrig rapport
☐ __________

ISBN
—

ISRN
LITH-ISY-EX-3305-2003

Serietitel och serienummer
Title of series, numbering

ISSN
—

URL för elektronisk version
http://www.vehicular.isy.liu.se
http://www.ep.liu.se/exjobb/isy/2003/3305/

Titel Framtagning av en motorstyrningsalgoritm använd i ett Shift-by-Wire-system

Title: Development of a Motor Control Algorithm Used in a Shift-by-Wire System

Författare / Author: Daniel Gullberg

Sammanfattning
Abstract

This thesis was done at DaimlerChrysler AG in Stuttgart, Germany. The aim of the thesis is to develop an algorithm for controlling a motor used in a Shift-by-Wire System. The control algorithm is to be implemented in a prototype car for further testing.

The Shift-by-Wire System can be described as follows: An electrical actuator is mounted in an automatic gearbox to select gears instead of the gear stick. The actuator is controlled by a microcontroller, which runs a control algorithm. The position of the actuator is measured with a linear position sensor and sent to the controller.

Nyckelord
Keywords

Motorola HC12, PD controller, PLCD sensor, Shift-by-Wire, Butterworth filter

Abstract

This thesis was done at DaimlerChrysler AG in Stuttgart, Germany. The aim of the thesis is to develop an algorithm for controlling a motor used in a Shift-by-Wire System. The control algorithm is to be implemented in a prototype car for further testing.

The Shift-by-Wire System can be described as follows: An electrical actuator is mounted in an automatic gearbox to select gears instead of the gear stick. The actuator is controlled by a microcontroller, which runs a control algorithm. The position of the actuator is measured with a linear position sensor and sent to the controller.

Keywords: Motorola HC12, PD controller, PLCD sensor, Shift-by-Wire, Butterworth filter

Preface

This thesis was done at DaimlerChrysler AG in Stuttgart, Germany during the time 25 June - 21 December 2002. Additional work was done at Linköpings Universitet during 2003.

This report is written in LaTeX.

Acknowledgment

I would like to thank all the people who helped me with this thesis. First all the people at DaimlerChrysler AG, department REM/EP, for contributing to my very pleasant stay in Stuttgart, Germany. There are too many names to write here but I can assure you, I will not forget any one of you! Special thanks goes to my supervisor Michel Willems ("and he's Dutch...") who helped me with both small and big problems, as well as being a good friend.

Also I would like to thank the people at the Department of Vehicular Systems, Linköpings Universitet. My supervisor Anders Fröberg who always helped me with my report, and how about all those coffee breaks with free coffee!

And finally, thanks to my family and my girlfriend for always being there for me.

Contents

Chapter 1

Introduction

Background

Through the history of automobiles, the gear stick has almost always been placed between the front seats. This space is therefore occupied and cannot be used for other things. The reason for having the gear stick in this position, is because of the mechanical linkage between the gear stick and the gearbox. This linkage goes from the bottom end of the gear stick, under the floor, and to a connector on the gearbox. The gearbox (at least on rear wheel driven cars) is positioned right in front of the gear stick, under the floor. This makes the placing of the gear stick convenient, by mechanical means, as it makes the linkage between gear stick and gearbox short.

If a way was found to remove this mechanical link, there would be no need to have the gear stick positioned between the seats. And this is what Shift-by-Wire is all about. With Shift-by-Wire, the shifting of gears is independent of a mechanical link to a gear stick. Instead, electronics are used. Electrical wires are much more flexible than a mechanical link and the gear stick is totally freed from the area between the seats. A set of buttons on the steering wheel can be used for example. This may also be more ergonomic than the conventional gear stick.

New cars produced by DaimlerChrysler might have the option of being installed with Shift-by-Wire. Research is currently being made along with other projects like Steer-by-Wire and Brake-by-Wire.

This report will explain a way to control the actuator which is installed inside the gearbox.

Objectives

The objectives is to develop a motor control algorithm for the Shift-by-Wire-actuator in an automatic gearbox. This actuator will change between the four possible driving modes: P, R, N and D. The working system will be implemented in a prototype car. This prototype car is to be tested and Shift-by-Wire is planned to be introduced in future series of cars produced by Daimler Chrysler AG.

Methods

The methods used in this report for developing the motor control algorithm are as follows. First a Matlab/Simulink model of the actuator is put in a control loop in the computer. A controller is implemented in Simulink and the control parameters is chosen which gives the controller good behaviour. This controller is then implemented in C-code and programmed onto a Motorola microcontroller mounted in the SCU (Shift Control Unit).

The SCU controls the actuator. Measurements are made by sampling of the actual signals sent to and received from the SCU. The measured signals are then analyzed in Matlab and the control parameters are adjusted to give the regulator a stable and fast response.

Thesis outline

Chapter 2 will give a short introduction to Shift-by-Wire. In Chapter 3 the hardware used is described. Chapter 4 shows the Matlab models used, and Chapter 5 describes the development of the control algorithm. Finally, in Chapter 6 the results are shown and examples of further work are given.

Limitations

The SCU receives CAN-messages sent by a Man-Machine Interface (MMI). This report will not discuss the different kinds of MMIs.

In this work not much time is spent on optimization of program code and control parameters. The goal is to produce a *working* control algorithm to implement in a prototype car.

The implementation of the control algorithm; the program code written, the programming of the microcontroller in the SCU etc. will not be described deeply. This report focuses on the methods and results from the development of the control algorithm.

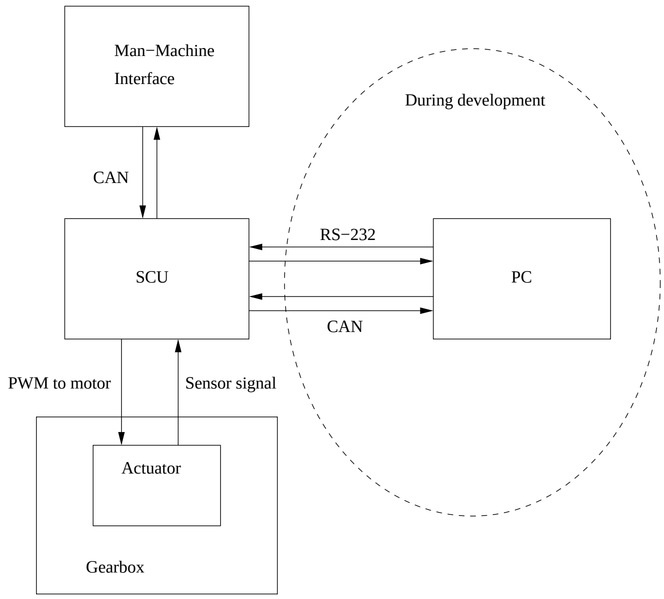

Figure 1.1: Overview of the Shift-by-Wire system

Overview

In Figure 1.1 the system is shown. The gearbox is a five speed Chrysler automatic transmission, usually fitted in the Grand Cherokee Jeep.

The Shift-by-Wire system consists of the SCU and the gearbox with actuator. The Man-Machine Interface (MMI) sends CAN-messages to select the driving position. The PC is only connected during the development of the control algorithm. With the PC, measurements can be made and also control parameters can be changed in the SCU.

Chapter 2

About Shift-by-Wire

Shift-by-Wire is the concept closely related to other concepts in this genre, like Drive-by-Wire, Steer-by-Wire and Brake-by-Wire. All of these concepts remove the mechanical link between two systems; in Shift-by-Wire the mechanical link between the gear shifter and the gearbox. In this work an automatic gearbox is used, with the driving modes P, R, N and D (Park, Reverse, Neutral and Drive).

Advantages with Shift-by-Wire are for example:

- Less noise in the coupé. The mechanical link between gear stick and gearbox is removed, and this link may transmit vibrations from the driveline into the coupé.
- More degrees of freedom for shifting gears because the shifting is made electrically. It may for example be possible to automatically put the transmission in driving mode P when the ignition key is removed, which prevents the driver to leave the car in transmission neutral.

To control the change of driving modes there is no longer a need for having the gear shifter between the driver- and passenger seat. Instead an electric actuator mounted inside the gearbox is used. The actuator can be controlled by for example buttons put anywhere in the coupé, because electrical wires are much easier to place than mechanical linkage.

When the driver wishes to change the driving mode, the system should respond sufficiently fast. This means that the control algorithm as well as the electric actuator should be fast.

This work is only about changing between the driving modes P, R, N and D. More specific, if for example a 5 speed automatic transmission is used, it could be possible not only to change driving modes, but also manually changing gear. A possible scheme would be P, R, N, D(auto), D(1), D(2), D(3), D(4) and D(5). But this type of shifting is beyond the scope of this work.

Chapter 3

Hardware

3.1 Overview

A Jeep Grand Cherokee is used in this project, see Figure 3.1. The original Chrysler gearbox mounted in the Jeep is replaced by a prototype gearbox. It is an automatic gearbox with the driving modes P, R, N and D. Originally, the driving modes are selected by means of the gear stick. On this prototype, the gear stick is removed and the driving modes are selected by an electric actuator, see Section 3.2.

The actuator is controlled by the SCU (Shift Control Unit), see Section 3.5. The SCU receives CAN-messages which changes the driving mode. While developing the control algorithm, also the control parameters were changed in real time via CAN and RS-232.

A MMI (Man-Machine Interface), in this case a hand held computer, is used to send the CAN-messages, and thus change the driving modes.

3.2 Actuator

The actuator, see Figure 3.2, is mounted inside the gearbox, at the position where the connection with the gear stick is located. The actuator moves two important parts of the gearbox: *the hydraulic valve* and *the parking lock mechanism.* The hydraulic valve changes the driving mode of the gearbox (i.e. P, R, N, D). (For more information about how automatic gearboxes work, see *A Short Course on Automatic Transmissions* [6]). When changing into mode P, the parking lock mechanism is responsible for locking the output shaft of the transmission. For more details about the parking lock, see Section 3.3.

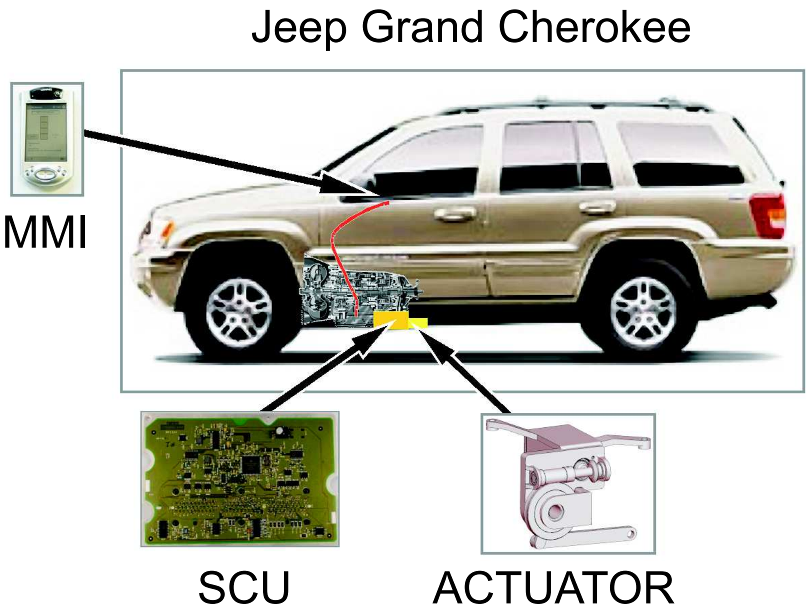

Figure 3.1: The Jeep Grand Cherokee

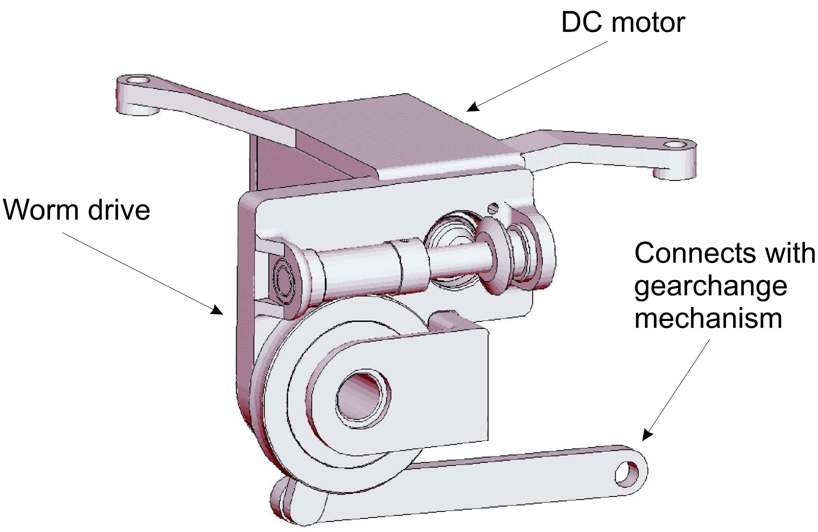

Figure 3.2: Actuator

3.2.1 Mechanical properties

The actuator consists of a DC-motor and a transmission. The transmission has a ratio of 1:200 which gives the actuator a high torque. The worm drive in the transmission makes the actuator keep it's position when the current to the DC-motor is turned off. Therefore, the actuator is self-holding which is both good and bad. Good, because if one driving mode is selected, the actuator will not need any control voltage to hold its position. Bad, because if an error in the system occurs, the gearbox may be stuck in one driving mode. If that driving mode is the P-position, the car will be impossible to tow if the tires are not lifted from the ground.

To work around the problem with the gearbox being stuck in the P-position during a electrical failure, a manual override is incorporated in the actuator. A ratchet is put in the worm wheel which makes it possible to go out of P-position even when the DC-motor will not run (for example, if the car battery accidently has been discharged). To use the manual override, a lever is mechanically connected to the gearbox.

3.3 Parking lock

The driving mode P is the parking lock. When in P, the output shaft from the gearbox is mechanically locked and cannot rotate. On the Grand Cherokee the parking lock is completely independent of the parking brake, which operates on the wheel brakes and not on the gearbox itself.

The selecting of parking lock position (driving mode P) exerts the most force on the actuator. If only the driving modes R, N and D were needed to be controlled by the actuator, it could be weaker dimensioned mechanically. Now, a greater force is needed. This is accomplished by a high gear ratio between the DC-motor and the output pin of the actuator. This gives a high torque, but also lower speed compared to a lower ratio.

If another construction were adapted in the gearbox for the parking lock, a construction using less force; a weaker and thus faster actuator could be used. But redesigning the parking lock is a costly procedure and is currently not available for the Shift-by-Wire project. Instead, the gearbox will be as much original as possible, to keep the costs low.

However, it may be possible to use a lower gear ratio in the actuator and still meet the demands of the force needed for the parking lock, without redesigning the gearbox. If research show that a lower ratio gives enough torque, the speed of the actuator could be increased, giving faster gear shifting.

Specifications	Detail	
Supply voltage		5V
Digital sampling	Resolution	10bit
	Response time	10ms

Table 3.1: PLCD sensor specifications

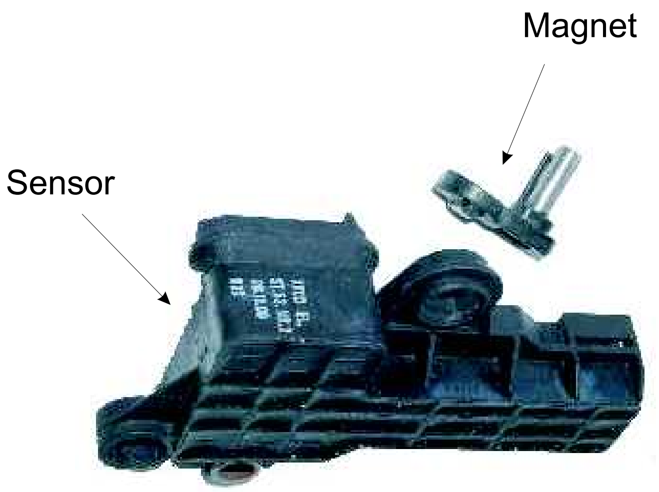

Figure 3.3: PLCD sensor

3.4 Position sensor

The position sensor monitors the position of the actuator in the gearbox. The sensor should correctly give feedback to the SCU of where the actuator is positioned, so correct control of the driving position can be accomplished.

The sensor used here is a sensor normally used in other automatic gearboxes. There it is used to verify that the hydraulic valve is in the correct PRND-position (it is in this case manually controlled by the gear stick). This sensor has been proven to work in the environment inside the gearbox. Some important data of the sensor is given in Table 3.1. For more information, see *Specification Driving Mode Sensor* [8].

The position sensor is of type PLCD-sensor *Permanent-Magnetic Linear Contactless Displacement sensor* and consists of two parts. A magnet, which mounts where the measurement is taking place, and a sensor over which the magnet slides, see Figure 3.3.

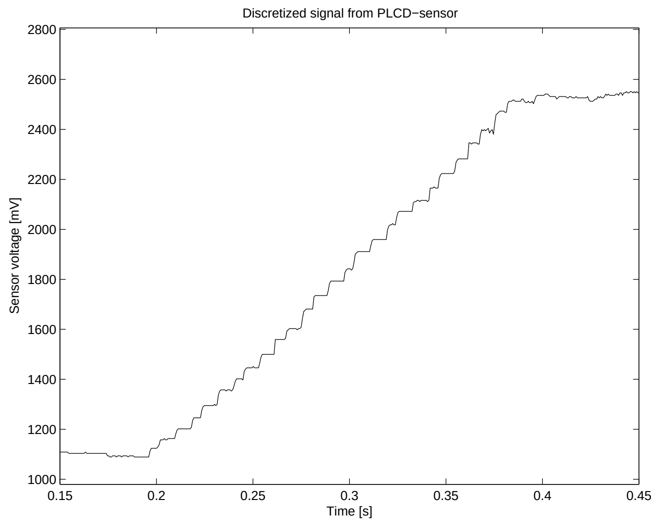

Figure 3.4: PLCD sensor sampling frequency examined

3.4.1 Difficulties with the sensor signal

The PLCD-sensor has analog output but internally samples the position digitally. The sampling frequency of the PLCD-sensor is slow compared to the sampling frequency of the SCU. This is shown in Figure 3.4. The step response should ideally be a smooth line as the position of the actuator changes linearly with time. The steps in the figure come from the sensor, as it is not capable of continuous measurement.

In Figure 3.5 each sample made from the SCU is shown as dots. It can be seen that approximately[1] 10 samples are made on every step from the PLCD-sensor. The SCU sampling time is approximately 1ms.

These steps in the signal sent from the PLCD-sensor will make it hard to implement a D-part in the control algorithm, as will be seen in Section 5.

3.5 SCU - Shift Control Unit

3.5.1 Overview

The SCU - Shift Control Unit controls the actuator by PWM voltage sent to the actuator DC-motor. The SCU receives position feedback

[1]The unevenness in the length of the steps are probably because of the OSEK-operating system running on the SCU, see Section 3.5.2. This operating system runs several simultaneous processes on the SCU and thus interrupts the sampling.

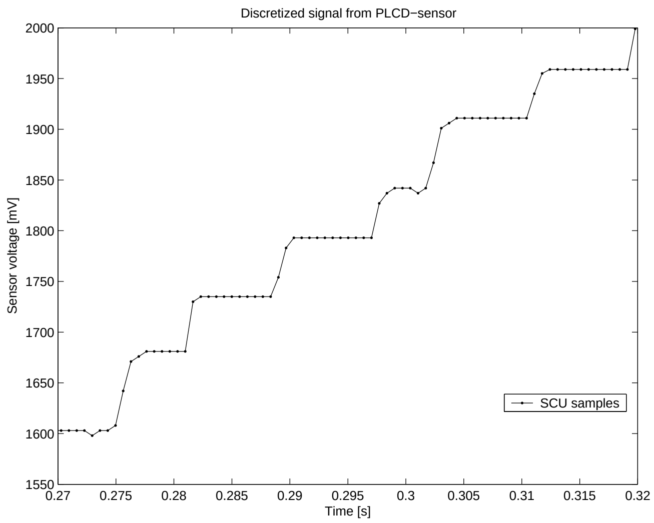

Figure 3.5: PLCD sensor sampling frequency examined. SCU sampling steps are shown as dots. (Detail from Figure 3.4)

Specifications	**Detail**	
Memory	RAM	12 kB
	EEPROM	4 kB
	Flash EEPROM	256 kB
A/D converter	16 channels	10 bit

Table 3.2: SCU specifications

from the PLCD-sensor. The position signal is sampled and used in the digital control algorithm running in the microcontroller.

Microcontroller

The microcontroller in the SCU is a Motorola HC12, 16 bit microcontroller. The specifications in Table 3.2 are taken from the *HC12 Manual* [5]. On the HC12, one channel of the A/D converter is used to acquire the PLCD sensor value. A total of 16 channels are available. The sampling time is depending on how fast the program runs in the HC12. Tests have shown that a sampling rate of approximately 1kHz (1ms/sample) is to be expected. (The PLCD-sensor is slower than this, see Figure 3.5. A faster sampling rate is not needed as it will produce no more information from the sensor).

3.5.2 OSEK

OSEK is the operating system running on the SCU. For more information about OSEK, see *Information about OSEK* [1]. OSEK uses multiple tasks that run simultaneously. A task with higher priority may interrupt a task with lower priority. Shown below are the tasks used in the software in order of priority. Task 1 has the highest priority:

- Task 1: Run control algorithm
- Task 2: Receive driver wish (P, R, N, D) via CAN

The following tasks are used only during development of the control algorithm:

- Task 3: Change control parameters via CAN
- Task 4: Communicate with PC

Task 1 need the highest priority to control the actuator as fast and accurate as possible. The other tasks are less time critical and are given lower priority.

Chapter 4

Modeling

4.1 Overview

A Matlab/Simulink model was used to make a good control algorithm. A model of the actuator, see Figure 4.2, was created by *N. Rehberg* [7]. Some changes was made to the original model from Rehberg: Added to the model was backlash (in the bottom right of Figure 4.2) and also the model was changed to run with both positive and negative control voltage. The actuator model was finally modeled together with a PID-regulator and the PLCD-sensor, see Figure 4.1, and simulations were made.

4.2 The Model

The block *Aktor* in Figure 4.2 is a three state model of a DC-motor together with a transmission. It takes three inputs: Control voltage, gearbox temperature and load. The control voltage is given from the controller but the temperature is here set as a constant (room temperature). Different temperatures are not tested in this work. The input signal *Load* is the force on the output rod. The greatest force is needed when selecting driving mode P as the parking lock mechanism needs more force than the hydraulic valve movement between R, N and D.

The block *Aktor* has many outputs. The only signal needed in this work is the movement of the output rod, which is connected to the hydraulic valve and the parking lock mechanism.

For more information of the model, see the report written by *N. Rehberg* [7]

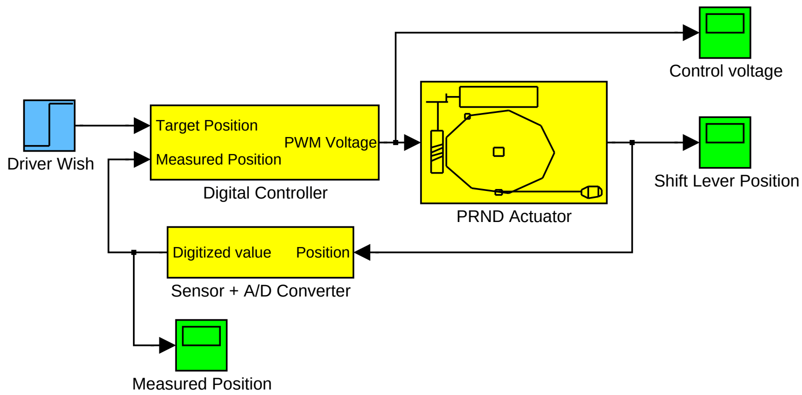

Figure 4.1: The Simulink model of the system

4.3 Simulation

When simulating this model, first a simple P-regulator was used. When the approximate K-value of the regulator was found, an I and a D-part was added. But the conclusion was that an I-part is unnecessary. When a sufficiently high K-value is used, the controller will not need an I-part. But, the D-part is needed for removing the overshoot that appears with only a P-part.

The conclusion is that a PD-controller should be used.

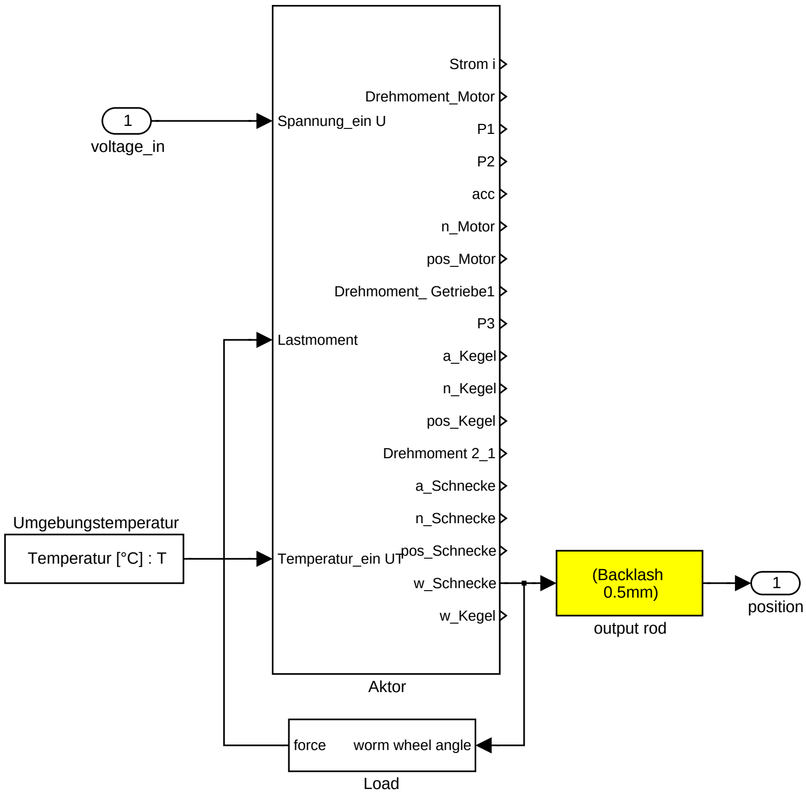

Figure 4.2: The PRND Actuator model

Chapter 5

Control algorithm

5.1 Overview

The goal is to control the actuator mounted in the gearbox. The four driving modes P, R, N and D are to be selected as fast and secure as possible. There are no mechanical stops related to the driving modes, so the controller must be able to freely position the actuator in the correct positions. Position feedback totally relies on the PLCD-sensor, see Section 3.4. The controller is implemented digitally in the SCU.

5.2 Driving modes

The four driving modes P, R, N and D are selected by moving the hydraulic valve in the gearbox. Also, for the P-position, the parking lock linkage needs to be moved. This is done by rotating the selector arm on the gearbox (the actuator rotates the selector arm). At different angles of the selector arm, the four different driving modes are located. The greatest distance is between P and R, and between the other driving modes, i.e. R-N and N-D, the distance is smaller.

The position sensor records the linear position of the hydraulic valve. The four driving modes can be found within the measuring range of the position sensor. As the sensor gives a signal approximately between 0-5V, the driving modes are distributed over this voltage span. As the signal is digitally encoded in the SCU with a 10 bit A/D-converter, the positions will be between 0-1023 ($2^{10} = 1024$).

When the position is within a specific limit, it will be "good enough" and the controller will stop, see Section 5.3. This limit can be changed in the control algorithm.

5.2.1 Play

Mechanically, the position will not be reached exactly. The reason for this is *play*. There will be a certain amount of play in the transmission between the DC-motor and the connecting rod, as well as between the connecting rod and the hydraulic valve. When the actuator runs in one direction, the play is taken up. If the direction of movement is changed, the play will be taken up in the other direction. This can cause instability, if for example the K-factor is set too high.

This is always the problem when using the same controller for the model as for the real system, as the play is difficult to model. If the model runs good without play and a stable and fast controller is developed, it will most certainly need to be changed when used in a real system. In this work, not too much work was spent on modeling. Instead, only a working controller was made with the model, and later the real system was used in making a stable controller.

5.3 When to stop controlling

To explain the need of stopping the controller, a simple model of a DC-servo will be used. This model looks like 5.1.

$$G(s) = \frac{k/\tau}{s^2 + \frac{1}{\tau}s} \tag{5.1}$$

$$Y(s) = G(s)U(s) + V(s) \tag{5.2}$$

(This model is derived from *Reglerteknik - Grundläggande teori* [3]). The sensor signal $Y(s)$ is the measurement of the servo position. $V(s)$ is white Gaussian noise introduced to model the measurement noise of the system.

When controlled by a P-regulator, the closed loop step response and the control signal for this system looks like in Figure 5.1. The dotted lines show the acceptable position error of the servo.

There is an overshoot and some oscillation in the step response. There is also noise both in the measured position and in the control signal. Because of the P-type regulator, the noise from the measured position is amplified. This means that even when the control error is within the limits (dotted lines in the plot), the controller continues to make small adjustments. This causes wear of the actuator, unnecessary current consumption and probably also mechanical noise. The control signal after $t \approx 1.7s$ should be 0, as it is unnecessary to reposition the actuator when the position already is within the limits.

Following are two examples how to prevent unnecessary movement of the actuator.

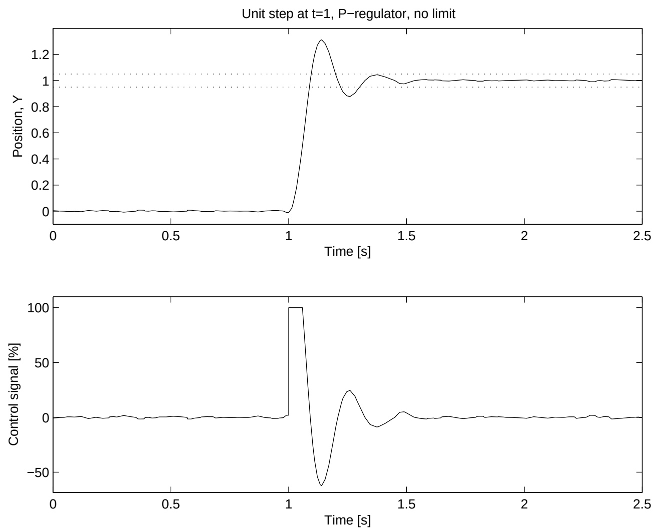

Figure 5.1: (top) P-regulator step response and limit (dotted lines) (bottom) Control signal

5.3.1 Stop when within limits

The position error of the controller is derived as

$$x_{error} = x_{target} - x_{sensor} \tag{5.3}$$

When the position error is small enough $|x_{error}| < \epsilon$ the controller should stop controlling the actuator. The result of this can be seen in Figure 5.2. The controller will continue to drive the motor until the position error is small enough. Then, when the position is within the limits, the control signal will be 0. But when looking closely at the figure, it can be seen that as soon as the position goes between the limits, the control signal will immediately be set to 0, regardless of the speed of the actuator. This means that the control signal will be clipped in an abrupt way as when the control voltage is turned off, the actuator may still be moving, making an overshoot and moving out of the limits.

5.3.2 Stop with time delay

A better way to cope with the problem of the actuator being unnecessary driven, is to incorporate a time delay before the controlling stops. The actuator stops only when the position has been between the limits for a certain length of time, see Figure 5.3. The diagram shows actual

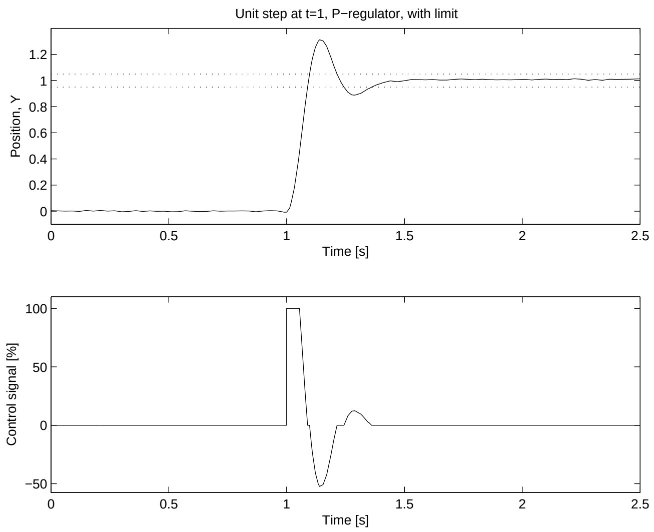

Figure 5.2: (top) P-regulator step response and limit (dotted line) (bottom) Control signal

measurements from the implemented controller. At a the position goes between the limits, but the controlling continues until b where the position has been between the limits for the predefined time. The control signal will be set to 0 after b, as seen in the figure.

With this time delay, the oscillations will have time to damp out before the control signal is set to 0, which occurs at $t \approx 0.47s$. Then, as long as the position is within the limits, the control signal will continue to be 0.

This method is implemented in the control algorithm.

5.4 Low Pass Filter

To remove noise from the measurement and to smooth out the steps created by the PLCD sensor, a low pass filter is used.

A low pass filter blocks high frequencies and lets through low frequencies. There are different kinds of low pass filters, like Chebychev and Butterworth filters. Here, a Butterworth filter of 2:nd order is used.

The discrete version of a Butterworth filter is taken from *Filter Functions & Coefficients* [4] The frequency characteristics of this filter

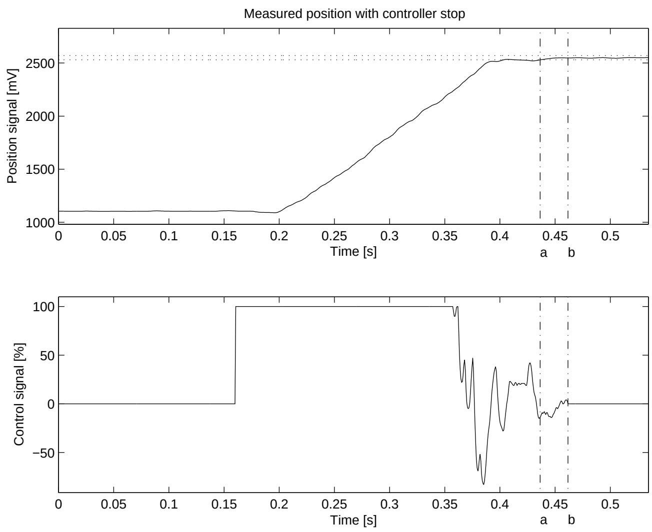

Figure 5.3: (top) P-regulator step response and limit (dotted line) (bottom) Control signal

can be seen in Figure 5.4. The transfer function of the filter is

$$H[z] = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}} \tag{5.4}$$

If this filter is operated on a signal, the output will be calculated as

$$y[n] = a_0 x[n] + a_1 x[n-1] + a_2 x[n-2] - b_1 y[n-1] - b_2 y[n-2] \tag{5.5}$$

When this filter is realized in the microcontroller, two old instances of each signal need to be saved: $x[n-1]$, $x[n-2]$, $y[n-1]$ and $y[n-2]$. These are saved in RAM in the microcontroller. The variables are of type `double` which make them 4 bytes each. That means a total of 16 bytes of RAM. Due to the limit amount of RAM in the microcontroller, see Section 3.5.1, care must be taken when designing the algorithm that not too much memory is used.

5.4.1 Filter model

To test the filter, a Simulink model is produced, see Figure 5.5. In the model, real data sampled from the SCU is used (Signal 1, Signal 2 and Signal 3 in the figure). The data is sampled in the SCU both before the filter (Signal 1) and after the filter (Signal 2). (Signal 3

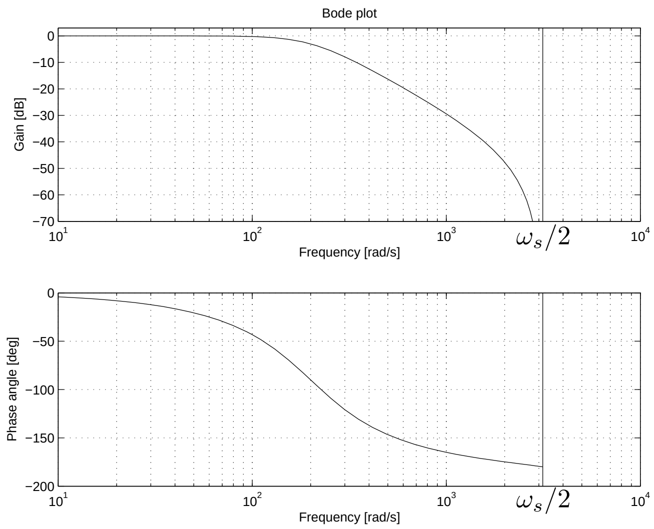

Figure 5.4: Bode diagram of time discrete Butterworth filter.

is not used here, but can be programmed to measure another signal). Then, the unfiltered data is fed through the Simulink model, and the output is matched with the filtered output from the SCU. This is used for validating the program code in the SCU. As seen in Figure 5.8 the data matches very well.

In Figure 5.6 and Figure 5.7 the effects of the filter is shown. Figure 5.6 shows the samples in the unfiltered data as dots.

A problem with the implementation of the Butterworth filter is the initialization of the internal variables. The first samples, when the SCU is turned on, will produce strange results from the filter, see Figure 5.9. The filter will need some samples to stabilize. To work around this problem, a timer is set to turn off the output to the actuator during the first number of samples after the SCU is turned on. This feeds the Butterworth filter with data, which stabilizes the signal.

5.4.2 Filter coefficients

The transfer function of the filter, Equation 5.4, has some coefficients to be defined. The calculations for defining these coefficients are taken from *Filter Functions & Coefficients* [4]

First, the desired cutoff-frequency f_c should be specified. The cutoff-frequency in the discrete domain is then defined as:

$$\Omega_c' = \tan(\frac{\pi}{f_r}) \tag{5.6}$$

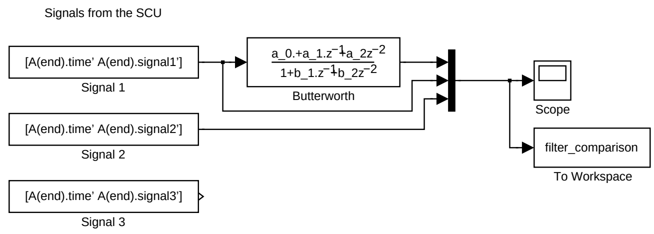

Figure 5.5: Simulink model of the Butterworth filter

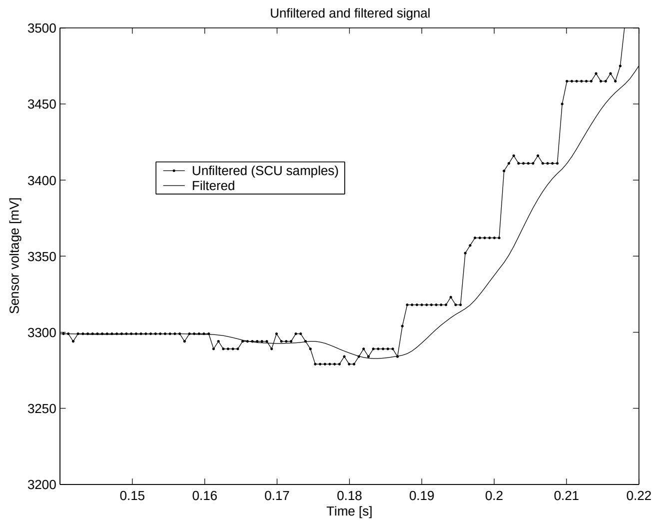

Figure 5.6: Butterworh filter: Unfiltered vs filtered signal (same data as in Figure 5.7)

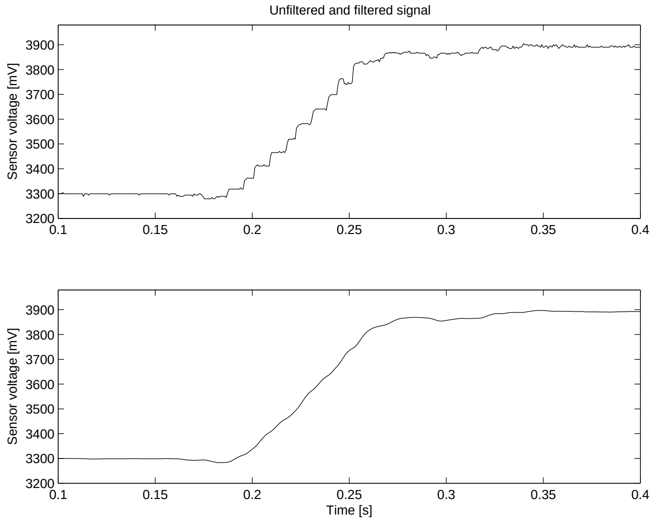

Figure 5.7: Butterworh filter: Unfiltered vs filtered signal (same data as Figure 5.6)

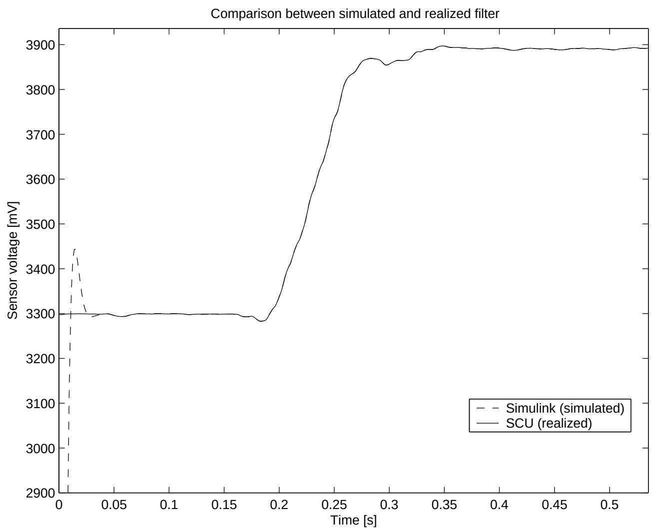

Figure 5.8: Butterworh filter: Comparison between C-code and Simulink model

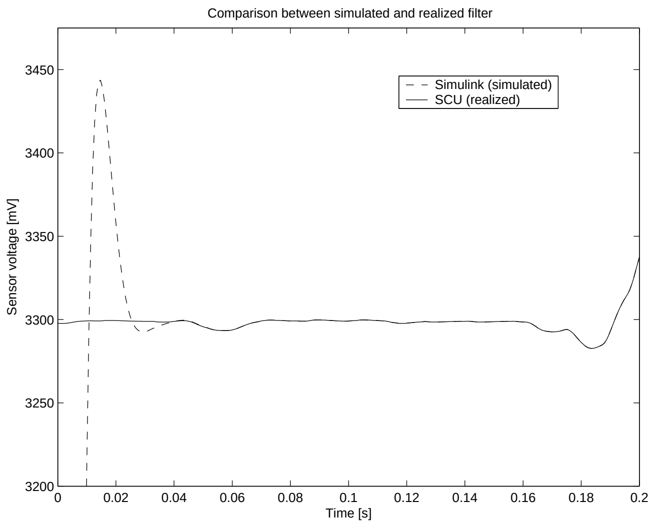

Figure 5.9: Butterworh filter: Comparison between C-code and Simulink model

where

$$f_r = \frac{f_s}{f_c} \tag{5.7}$$

f_s : sampling frequency (Hz)

In Equation 5.7 it is seen that f_r is the ratio between the sampling frequency and the cutoff frequency.

In the following calculations ω_c' is used, which is defined as *normalized* cutoff-frequency, which means the angular frequency multiplied by T:

$$\omega_c' = 2\pi f_c T = 2\pi f_c \frac{1}{f_s} = 2\pi \frac{f_c}{f_s} = 2\pi \frac{1}{f_r} \tag{5.8}$$

The filter coefficients are calculated as follows:

$$\Omega_c = \frac{2}{T}\tan(\frac{\omega_c'}{2}) \tag{5.9}$$

$$\Omega_c' = \tan(\frac{\omega_c'}{2}) \tag{5.10}$$

$$a_0 = a_2 = \frac{{\Omega_c'}^2}{c} \tag{5.11}$$

$$a_1 = 2a_0 \tag{5.12}$$

$$b_1 = \frac{2({\Omega_c'}^2 - 1)}{c} \tag{5.13}$$

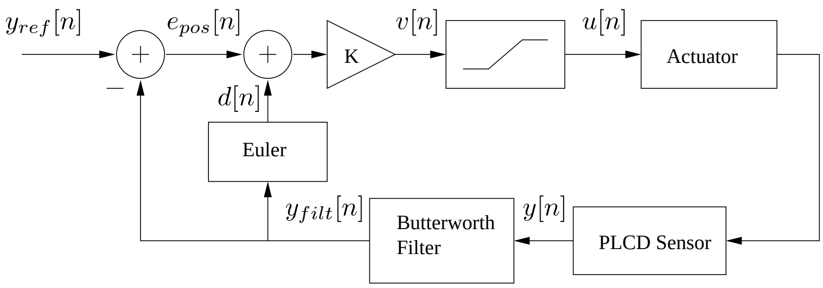

Figure 5.10: Overview of the control algorithm

$$b_2 = \frac{1 - 2\cos(\pi/4)\Omega_c' + {\Omega_c'}^2}{c} \tag{5.14}$$

$$c = 1 + 2\cos(\pi/4)\Omega_c' + {\Omega_c'}^2 \tag{5.15}$$

When the above equations are satisfied, the following will hold:

$$a_0 + a_1 + a_2 - b_1 - b_2 = 1.0 \tag{5.16}$$

5.5 Control Algorithm

A PD controller is used. The discrete version of a PD-regulator is taken from *Digital Styrning - Kurskompendium* [2]

$$v[n] = K(e_{pos}[n] + d[n]) \tag{5.17}$$

An overview of the final control algorithm is seen in Figure 5.10.

The K coefficient is the proportional, P-part, of the controller. The controller error e_{pos} is derived as follows

$$e_{pos}[n] = y_{target}[n] - y_{filt}[n] \tag{5.18}$$

where y_{filt} is the Butterworth filtered value of the position sensor signal and y_{ref} is the desired position.

5.5.1 Discrete differentiation

The discrete D-part is realized by the backward Euler mothod. Old values of the signal is saved and the difference between the new and old values, divided by the sampling time, gives an approximate value of the derivative of the signal in the specific point.

$$d_1[n] = \frac{y[n] - y[n-1]}{T_s} \tag{5.19}$$

If the signal two time-steps behind is used instead, division by two times the sampling time will give another approximation of the derivative. This approximation gives a smoother derivative, as a greater span of time between the points is used.

$$d_2[n] = \frac{y[n] - y[n-2]}{2T_s} \tag{5.20}$$

The coefficient T_d sets the amount of D-part being used in the controller. (If $T_d = 0$ the D-part is turned off and the controller will be purely proportional).

$$d[n] = -\frac{T_d}{T_s l_E}(y_{filt}[n] - y_{filt}[n - l_E]) \tag{5.21}$$

Here, the coefficient l_E (an integer value) selects the time difference between which the backward Euler is calculated. Equation 5.21 is implemented in the final control algorithm. A big l_E gives smooth values of the D-part, but with a phase lag as the calculation uses old values of the signal. Small values of l_E gives a D-part with a noisy but fast response. With a value of $l_E = 5$ the controller worked well, but further optimization of this value is possible.

5.6 Implementation

The control algorithm as seen in Figure 5.10 is finally implemented in C-code and flashed onto the Motorola HC12 microcontroller in the SCU. This process is not described closer in this work, as focus is put on the development of the control algorithm.

Chapter 6

Results and further work

6.1 Results

The working control algorithm has the following step responses from P-R R-N N-D.

These steps are not fast enough to comply with the demands from DaimlerChrysler AG. The time to shift between each driving mode has to be shorter. But this can not be accomplished only by changing the control algorithm, as the controller sends full control voltage to the motor during most of the controlling time. The DC-motor reaches full speed (this can be seen at time 0.2s in Figure 6.1 for example) and moves the rod as fast as possible. A solution to this would be to change the ratio in the gearbox, or to use a DC-motor with higher rpm.

The stability with this controller is satisfactory. With a gearbox in a test rig tests were made with different control parameters. A stable controller was found, but the control parameters can still be optimized.

The conclusion of this work is that a working control algorithm for the actuator is found.

6.2 Further work

At the end of this work, a working control algorithm was found. But there are still work to do to get a controller optimized for speed an stability. By further measuring step responses from the controller and changing the different control parameters it may be possible to get better results.

If a sensor with a higher sampling rate could be used, a faster response time of the controller is to be expected, as there would be less phase lag because of low pass filtering of the signal.

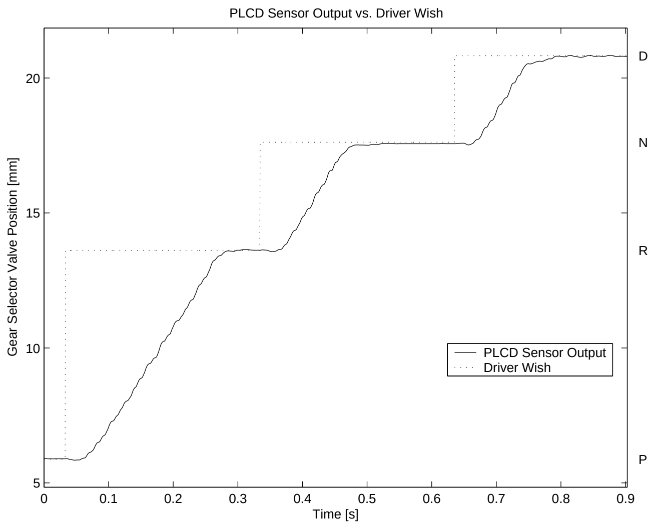

Figure 6.1: PRND steps

If the gear ratio of the actuator could be lowered, faster gear change would be possible.

Also, the mechanic coupling between the actuator and the moving parts in the gearbox should be examined. It is probably possible to reduce the play in the linkage, which will make controlling more accurate.

Chapter 7

Glossary

CAN Controller Area Network

DC-motor An electric motor driven by DC

Driving mode One of the modes P (Park), R (Reverse), N (Neutral) and D (Drive) selected in the automatic gearbox

OSEK German abbreviation for "Offene Systeme und deren Schnittstellen für die Elektronik im Kraftfahrzeug"(In English: "Open Systems and the Corresponding Interfaces for Automotive Electronics")

PWM Pulse Width Modulation

SCU Shift Control Unit (uSCU - universal Shift Control Unit)

References

[1] DaimlerChrysler AG et al. OSEC-VDX.org. internet, May 2003. Read 30 May 2003 from http://www.osek-vdx.org/.

[2] T. Glad, S. Gunnarsson, L. Ljung, T. McKelvey, A. Stenman, and J. Löfberg. *Digital Styrning - Kurskompendium.* Reglerteknik/Automatic Control, ISY, Linköping, Sweden, February 2002. In Swedish.

[3] T. Glad and L. Ljung. *Chapter 2 - Reglerteknik. Grundläggande teori.* Studentlitteratur, Lund, Sweden, 2nd edition, 1989. In Swedish.

[4] Young-Hoo Kwon. Filter functions & coefficients. internet, Nov 2002. Young-Hoo Kwon 1998, Read 20 November 2002 from http://kwon3d.com/theory/filtering/fil.html.

[5] Motorola Confidential Proprietary. *MCS912DP256 Advance Information*, 2000. Rev. 0.1, Read 26 nov 2002.

[6] Charles Ofria. A short course on automatic transmissions. internet, 2000. Read 6 November 2003 from http://www.familycar.com/transmission.htm.

[7] N. Rehberg. Modellierung einer elektromechanischen integrierten Shift-by-Wire-Aktorik für den Einsatz im Automatikgetriebe. Diplomarbeit, Fachhochschule Esslingen, Esslingen, Germany, June 2002. In German.

[8] Schikora. Specification driving mode sensor. Siemens, Transmission Control Units, Regensburg. Datasheet., July 2002.