ISSN 0280–5316
ISRN LUTFD2/TFRT--5638--SE

Adaptive Realtime Control of a Nonlinear Throttle Unit

Johan Gagner
Rickard Bondesson

Department of Automatic Control
Lund Institute of Technology
February 2000

Department of Automatic Control
Lund Institute of Technology
Box 118
SE-221 00 Lund Sweden

Document name
MASTER THESIS

Date of issue
February 2000

Document Number
ISRN LUTFD2/TFRT--5638--SE

Author(s)
Johan Gagner and Rickard Bondesson

Supervisor
Bo Bernhardsson

Sponsoring organisation

Title and subtitle
Adaptive real-time control of nonlinear throttle unit

Abstract

This master thesis has been done as a joint project between the Department of Automatic Control in Lund and Volvo Technological Development in Gothenburg. The thesis treats methods to handle nonlinearities in a throttle unit. The approach has been to first design a linear controller based on the results from system identification, and then to develop an adaptive updating law estimating uncertain parameters of the throttle. The implementation has been done on a PC using Matlab.

The underlaying robust linear controller design was done with two different approaches. A first attempt was pole placement on polynomial form, i.e. an RST-controller. This controller was not easy to tune, and the design of a well behaved closed loop system failed. A second attempt was made using the method of optimal control on state space form. The use of weight functions in the design made it easier to tune the controller. The closed-loop performance of the derived linear controller meets all specifications except the rise time for small steps.

For the adaptive update law three methods were tested based on gradient theory, stability theory and a state observer. The task has been to estimate the characteristics of the main nonlinearity of the throttle, the dead-zone. The first two methods can not estimate all of the required parameters, so they were rejected. By introducing a disturbance driven by white noise as an extra state in the state observer it is possible to estimate all the needed parameters. The closed loop performance through the dead-zone is very fast and meets almost every demand in the specification. The control law fails to meet the specified rise time for small steps and the overshoot is sometimes too large. There are also some oscillations under certain circumstances when controlling close to the dead-zone.

Key words

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280–5316

ISBN

Language
English

Number of pages
75

Recipient's notes

Security classification

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.lu.se

Summary

This master thesis has been done as a joint project between the Department of Automatic Control in Lund and Volvo Technological Development in Gothenburg. The thesis treats methods to handle nonlinearities in a throttle unit. The approach has been to first design a linear controller based on the results from system identification, and then to develop an adaptive updating law estimating uncertain parameters of the throttle. The implementation has been done on a PC using Matlab.

The underlaying robust linear controller design was done with two different approaches. A first attempt was pole placement on polynomial form, i.e. an RST-controller. This controller was not easy to tune, and the design of a well behaved closed loop system failed. A second attempt was made using the method of optimal control on state space form. The use of weight functions in the design made it easier to tune the controller. The closed-loop performance of the derived controller meets all specifications except the rise time for small steps.

For the adaptive update law three methods were tested based on gradient theory, stability theory and a state observer. The task has been to estimate the characteristics of the main nonlinearity of the throttle, the dead-zone. The first two methods can not estimate all of the required parameters, so they were rejected. By introducing a disturbance driven by white noise as an extra state in the state observer it is possible to estimate all the needed parameters. The closed loop performance through the dead-zone is very fast and meets almost every demand in the specification. The control law fails to meet the specified rise time for small steps and the overshoot is sometimes too large. There are also some oscillations under certain circumstances when controlling close to the dead-zone.

Table of contents

1 Preface

This master thesis has been performed at the Department of Automatic Control, Lund Institute of Technology and Volvo Technical Development in Gothenburg. Supervisors for the thesis have been Ph.D. Mats Andersson at Volvo and at LTH, Professor Bo Bernhardsson, who also is our examiner.

We would like to thank the following persons. Without their commitment this thesis would not have been possible to fulfil.

Our supervisor Professor Bo Bernhardsson and the two co-supervisors Ph.D. students Anton Cervin and Magnus Gäfvert for support and hints during the project. They have also read our report and given us very good feedback. We would also like to thank Mats Andersson at Volvo for providing us with an interesting real world control problem.

All personnel, staff and Ph.D. students at the Department have been a great support. If someone deserves some extra attention it is Anders Robertson. He has helped us both in good and hard times and shared with us some of his knowledge in the field of both control theory and practical aspects.

Our good friends Pär and Henrik have helped us boost morale during hard times of the project. Last but not least we would thank Pernilla and Sofie, our girlfriends, for their understanding and support during hard times of the project.

2 Introduction

2.1 Background

In the Volvo car engine, mounted on for example Volvo S80, the accelerator pedal is not directly connected to the throttle as in elder car models. This means that when the driver wants to make a change in speed (s)he moves the accelerator, called Accelerator Pedal Module (APM) and the new position is sent as a signal over the Controller Area Network (CAN) to the Engine Control Module (ECM). The module will then calculate the new airflow, fuel flow and different exhaust gas parameters needed in the engine according to the new accelerator position. After the airflow calculation is done it is transformed into an angular position, which is then sent over the CAN bus to the Electronic Throttle Module (ETM). The ETM is in this report called simply, the throttle. The throttle plate moves to the new position corresponding to the angular reference value. The actual throttle angle is sent back through CAN to the ECM, which compares it with the set-point. If the set-point and actual position of the throttle differ, a new calculation is done. The control loops are parts of the Engine Management System (EMS) and are supervised by this system.

The ETM and the ECM are produced by different manufacturers which both sell their products to Volvo. The products are then assembled into a complete engine control system. Volvo designs a product specification and the manufacturers are then responsible for the implementation of the design.

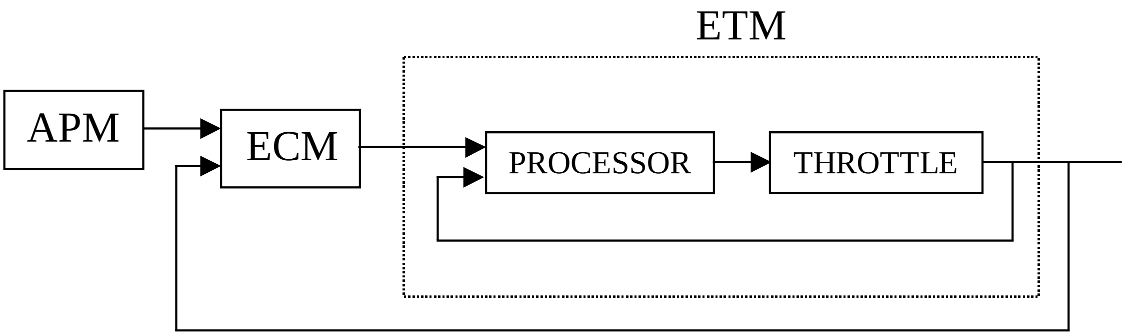


Figure 1. Schematic view of the airflow control system. The Accelerator Pedal Module sends a velocity reference to the Engine Control Module, which transforms the reference to a plate position and sends it to the Electronic Throttle Module. The inner control loop then moves the plate accordingly. The outer control loop is also used for safety and monitoring of the throttle.

2.2 Problem Description

This report focuses on the control of the plate position in the throttle. In the construction stage some nonlinearities were built into the throttle. The most significant nonlinearity is a dead-zone. It has its origin in a security position of the throttle called the “limp home” position. This is the position that the throttle will reach if the electric power to it, for some reason, would be cut. The dead-zone width differs between throttle units due to the manufacturing process. There are other nonlinearties e.g. Coulomb friction and nonlinearities related to tear and wear. The vendor of the throttle, Magneti Marelli, has approached the control problem with a linear cascade controller. This control method works well, except close to the dead-zone. In an earlier master thesis, [1], an attempt to make a better linear controller based on QFT design was made. The resulting controller is good at operating points strictly above the dead-zone. No attempt was made in [1] to move through the dead-zone.

To achieve acceptable control responses the manufacturer of the S80, Volvo Car Corporation, has made a list of requirements that have to be met. In the specifications, Volvo choose to denote the plate angle in %angle, where 0 %angle corresponds to a closed throttle and 100 %angle to a fully open. The requirements are as follows,

- The steady state error should be less than 0.10 %angle.
- Maximum time before steady state is reached, (T_s), should not be more than 0.2 s.
- 5 % overshoot (M) of the step response is allowed.
- 10 % of the step response should be reached within 20 ms, (T_{delay}).
- The step time (T_{step}), 10 % to 90% should not be greater than,
  - 20 ms, when the angular change is less then 3 %angle.
  - 40 ms, in the range of 3 %angle to 10 %angle change.
  - 60 ms, if the change is larger than 10 %angle.

These criteria can be represented as in the following figure, all according to [1].

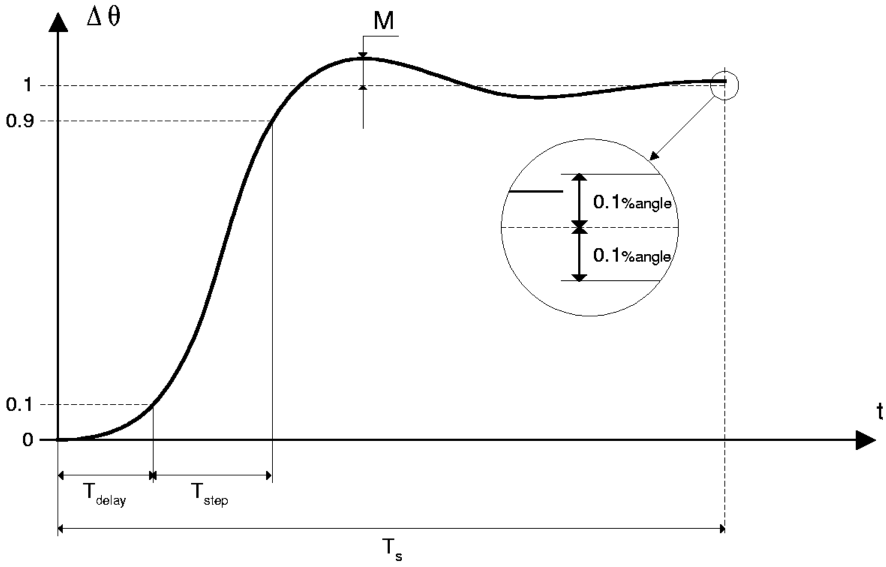


Figure 2. The performance specifications for the throttle. T_{step} is the rise time,T_{delay} is the time delay, T_s is the time to steady state and M is the overshoot. 0.1%angle in steady state corresponds to 0.09°. The 20 ms step time requirement shows to be close to impossible to meet while the 40 ms condition is easier to fulfil.

2.3 Goal

In this master thesis different approaches to deal with the nonlinearties will be discussed. The primary aim is to find a robust control law that is better than the one used today. The controller should also be able to take care of most of the uncertainties in the nonlinearities e.g. compensate for the spring torques. Outside the dead-zone, two linear controllers will be used, one above the dead-zone and another one below.

2.4 Limitations

In a Volvo S80 the communication between sub systems is performed over a CAN-bus. In this thesis work, the loop will be closed outside the throttle. For this purpose a new interface needed to be developed in hardware and software.

Since only one complete rebuilt throttle was available it was impossible to do any tests of the robustness due to manufacturing process variations. All experiments have been done indoors at room temperature. In reality the engine temperature, and the throttle temperature, can vary between -40 °C and 120 °C. The performance of the throttle under these temperature variations has not been simulated, tested or validated. Throughout this work it is assumed that a good estimation of the throttle transfer function is known.

2.5 Contributions

This master thesis will improve the throttle closed loop performance, trying to meet all VTD demands. In an earlier Master thesis a robust controller was developed. This controller was only valid in an area around the linearized point. Outside this area there were problems with instability.

First a robust linear controller will be designed taking care of the problems outside the dead-zone area. The aim is to design two controllers, which can be used in the whole area above respectively below the dead-zone respectively. Then an adaptive update law and a compensator will be developed taking care of the nonlinearities close to and inside the dead–zone. The same demands are used for the controllers when the throttle plate moves through the dead-zone as when controlling it outside the dead-zone.

It will be hard to make any comparisons of results between this master thesis and the former work done at Volvo, the previous master thesis and at Magneti Marelli. The reason is that the process dynamics are not the same, the electrical parts differ. When implemented, the QFT-controller from the previous master thesis became unstable and can therefore not be evaluated. The performance and structure of the implemented controller in the throttle mounted on an S80 is not known at all, due to secrecy issues at Volvo. The evaluation of the control law of this thesis will be done by comparison with the performance specifications on the closed loop system received from Volvo and using plots from the previous master thesis.

3 Hardware

3.1 The Throttle

The throttle is a component of the Engine Management System and has been developed by the Italian corporation, Magneti Marelli, according to specifications from Volvo. When the driver wants to change speed or gear, changes in the airflow to the engine are needed. The throttle module provides the engine with the new airflow as described in the introduction chapter.

An overview of the throttle is seen in Figure 3 below. The controllable plate is mounted in the middle of the air outlet. A movement of the plate results in a change of the airflow through the throttle. The control of the plate position is performed using a DC motor and a potentiometer, which are situated on the left side of the throttle in the picture. Located to the right is a spring package consisting of two torsion springs and a second potentiometer. The motor torque will rotate the plate, and the spring package acts as a counter torque. The two potentiometers are used to measure the plate axis position. One as a primary output for measurements and the second for safety reasons.

A microprocessor is placed under the outlet together with necessary electric circuits for the communication between the throttle and the supervisor control system. The microprocessor is embedded and closes the local control loop. The motor is fed with a pulse-width modulated current at a frequency of 8 kHz from the microprocessor. Using this technique the motor input will be a mean voltage instead of a current.

During the work on this master thesis the microprocessor was removed from the throttle. Instead an external power supply was used, and the potentiometers were connected to an AD converter. The control signal to the motor was fed through a DA converter to an external pulse-width modulator connected to the throttle. The loop was then closed outside the throttle unit via the AD/DA converter.

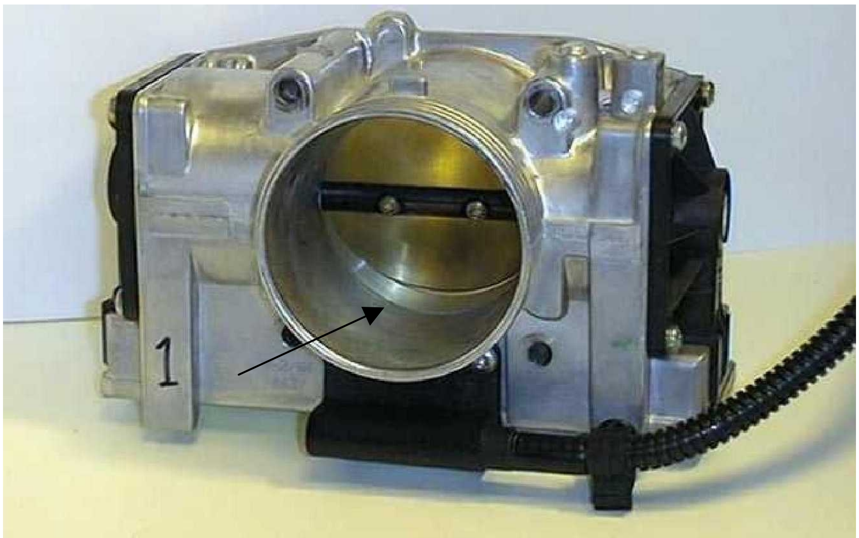


Figure 3. The inactive throttle unit. Notice that the throttle is not fully closed. The gap makes it possible to run the engine even if the throttle is offline for some reason. The plate position when the throttle is inactive is called the "limp home" position.

As can be viewed in Figure 3 above, the plate is not fully closed when the throttle is inactive. This is called the limp home position. With this feature it is possible to drive the car at low speed even if the power supply is broken. The construction uses counter-directed springs in the spring package. To be able to move the plate, the motor torque first has to overcome the counteracting spring torque, and after this is done it is possible to control the plate position. This is the origin of the specified, but difficult to handle, dead-zone.

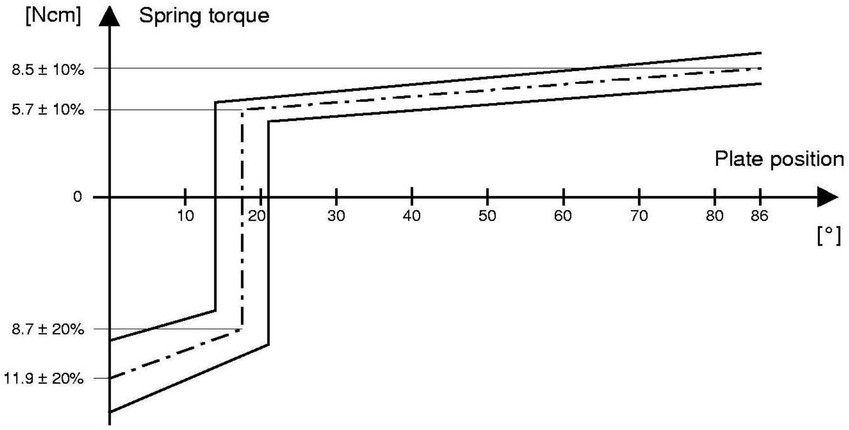


Figure 4. Static characteristics of the spring package, according to [10]. The spring torque differs highly between the working areas above and below the dead-zone but also between throttle units. The lack of exact parameter values makes a static spring torque compensation difficult to implement.

3.2 The experiment equipment

All experiments were carried out on a PC, 450 MHz Pentium III, with Linux as operating system. To be able to control the motor on the throttle and to read measurement values the throttle was connected to an external power supply. The power supply can feed the system with 5 A maximum current during a short period of time before it is shut down due to safety measures. The computer was used as a Graphical User Interface (GUI) to the experiments. A 12 bit AD/DA converter was mounted on the computer for measurement input reading and output control signal setting.

The throttle system, including the power supply, was examined with an oscilloscope and some noise was found. The noise was mainly a sinusoid with a frequency of 1.67 MHz. Therefore a low pass filter at 1 MHz was connected to the AD converter. After the filtering of the input signal, much of the noise problems disappeared.

The process communication was built in the Matlab simulation environment, Simulink. Software to realise communication between a computer and a process has been developed at the Department of Automatic Control. In Figure 5, function blocks for handling the A/D and D/A conversion in Simulink are shown. The two blocks are connected with a feedback link to determine which block will be calculated first in the control loop. The blocks take a port number on the AD/DA card and a sample time as parameters.

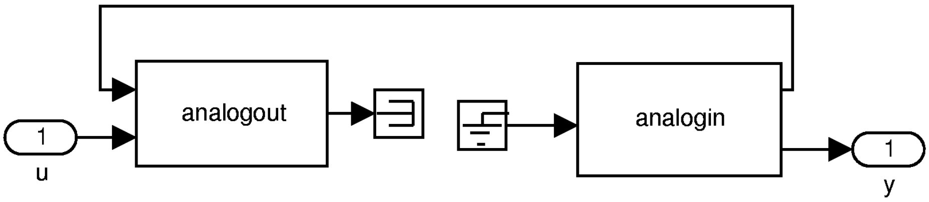


Figure 5. The AD/DA converter blocks used in the Simulink model. This way of communicating with the process is extremely easy and done without any real-time programming.

4 System Identification

4.1 About system identification

When performing modeling and system identification, it is necessary to do it in a systematic manner. If the identified model is close to the real process, a linear controller can be designed quite easily. The performance of the closed-loop system can then be tested in simulations before it is implemented to avoid problems.

The system identification process can be divided into stages. First a physical model is derived from a theoretical point of view. A model structure has to be selected and experiments suggested that estimates the model. The next step is to carry out the experiments and perform data examination. The last step is the validation of the estimated model. The validation gives an estimate of the accuracy of the model. An accurate model can be used to design a controller. The identification process is iterative, this means that, if one of the steps gives a result that can not be used further, it is necessary to back up in the analysis scheme. The design procedure is in this aspect much like trial and error, see Figure 6 below.

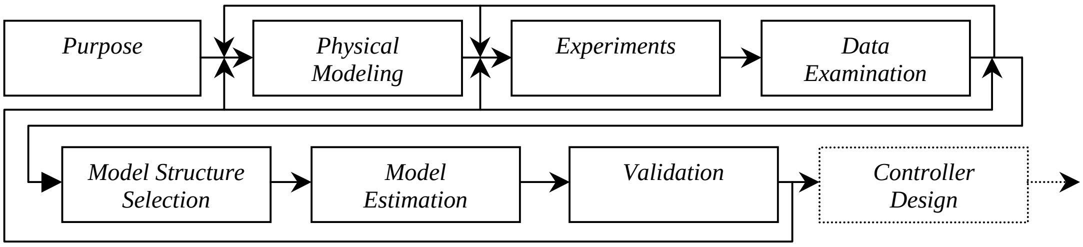


Figure 6. Algorithm for modeling and system identification copied from [6]. Some iterations between the steps in the algorithm are often necessary for obtaining an accurate result. The algorithm is divided into three parts, physical modeling, experiments and validation.

In Matlab there is a toolbox called the System Identification toolbox. The toolbox gives the user access to a great number of identification functions. It is possible to use these functions and produce Matlab scripts to execute, but one can also do the main work in a program called Ident. This program has a graphical user interface based on a drag and drop feature and is therefore preferred to script coding.

4.2 Physical modeling

The physical modeling gives information on the model structure. The gathered information could be model order, dominating poles, static gain, delays and nonlinearities, among many others. The throttle with equipment can be decomposed into two parts. The first is the motor and the second is the spring package. The obtained model in this thesis is largely the same as in [1].

The motor is a two-pole brush-less DC motor. It is permanently magnetised and fed with a current from the power circuits. To smooth out current peaks and to reduce power consumption, a pulse width modulator is used. Therefore the input signal to the motor is a voltage reference to the modulator.

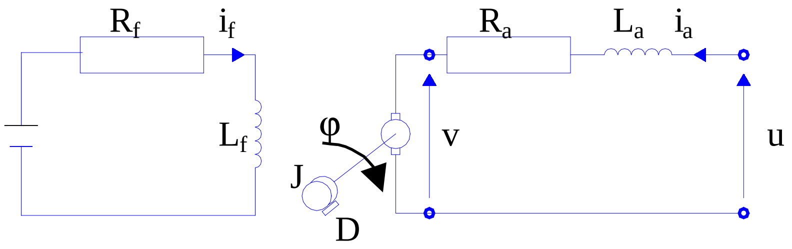


Figure 7. The anchor current controlled DC motor. The left most part is the field fed circuit with a resistance, R_f and an inductance L_f. The right most circuit is the anchor circuit with a resistance, R_a, and an inductance L_a. By changing the anchor voltage, *u*, the angular velocity, φ, will be affected. The load consists of the inertia, *J*, and the damping , *D*, and spring constants, *K*, of the throttle.

The resistance and the inductance in the anchor windings and the feeder is denoted R_a respectively L_a. If the input voltage, *u*, to the motor is varied, the output motor axis angle φ will change and also the torque transferred to the spring package. It is assumed that the spring package consists of a total inertia, *J*, including the motors own inertia. There is also a dynamic damping, *D*, and a spring constant, *K*. Exactly how the two springs interact is not known but should probably not be modelled according to [10].

Ohm's law for the anchor circuit is then,

$$u(t) = R_a \cdot i(t) + L_a \frac{di(t)}{dt} + v(t) \tag{4.1}$$

Or it can also be expressed in current after Laplace transforming,

$$I(s) = \frac{U(s)\text{-}V(s)}{R_a + sL_a} \tag{4.2}$$

A counter-directed emf, *v*, is induced when the motor anchor rotates in the magnetic field. Assume that the field current, i_f, is constant. Then the air gap magnetising will be constant. The counter-directed field then depends only on motor axis velocity and can be expressed as follows:

$$v(t) = K_u \cdot \frac{d\varphi(t)}{dt}, \tag{4.3}$$

where K_u is the motor voltage constant and φ is the position of the motor axis. After Laplace transforming the equation becomes,

$$V(s) = K_u \cdot s\Phi(s) \tag{4.4}$$

Since it is assumed that the magnetisation is constant, the anchor torque is proportional to the anchor current,

$$M_d(s) = K_m \cdot I(s) \tag{4.5}$$

K_m is the motor torque constant. If the energy transfer is approximately made without any losses, K_u is equal to K_m. Introducing the mechanical torque equation acting on the spring package results in,

$$J \cdot \frac{d^2\varphi(t)}{d^2t} = M_d(t) - D \cdot \frac{d\varphi(t)}{dt} \tag{4.6}$$

Laplace transformation gives

$$\left(J \cdot s^2 + D \cdot s + K\right) \cdot \Phi(t) = M_d(t) \tag{4.7}$$

By combining equations (4.2),(4.4),(4.5) and (4.7) the a block diagram for the throttle could look like in Figure 8.

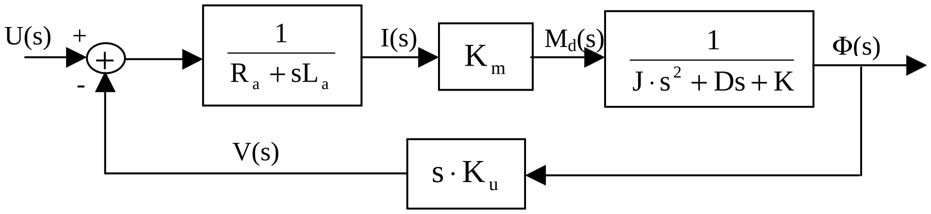


Figure 8. The block scheme of the theoretical model, including motor and spring package.

The transfer function from *U(s)* to *Φ(s)* is then as follows.

$$\frac{\Phi(s)}{U(s)} = \frac{K_m}{\left(J \cdot s^2 + D \cdot s + K\right) \cdot \left(R_a + L_a \cdot s\right) + K_m \cdot K_u \cdot s} \tag{4.8}$$

The obtained model is linear and only valid inside the linearised area. It has three poles and no zeros. Some nonlinearities are already known, e.g. a dead-zone caused by the limp home mode, static and dynamic friction both in the DC motor and in the spring package. Other nonlinearities could be found in the spring package, in the spring damping, *D*, and the spring constant, *K*, and some not perfect parts in the electrical devices. The properties of the system will also change with temperature.

None of these nonlinearities have been modelled yet, but they will turn up during the identification and will be compensated for when needed. Later in the thesis the dead-zone position and the spring torque will be estimated and used as a part of the controller structure.

4.3 Experiments

Physical modeling is often not enough to determine the process dynamics. More information can be found by performing experiments on the real process. First some fast and well-chosen tests should be carried out to get better information about the process and to plan the main identification experiments. The experiments performed were the following:

- Location of the dead-zone.
- Friction experiments and compensation. This to avoid effects of friction in further experiments.
- Examination of the process linearity properties.
- Plotting of an approximated transfer function.

- Determining the main time constants, the time delay and the process gain.
- Planning and executing of the main identification experiments to achieve a parametric model.

All identification experiments have been performed in open-loop. The sample time during the identification was chosen to 1 ms, the same speed as the control loop of the real process.

4.4 Dead-zone

The first thing that was examined was the width and position of the dead-zone. It is important to know the dead-zone characteristics so that no identification experiments are performed in this area. If the identification is carried out close to or inside the dead-zone the obtained information is useless.

The throttle plate was fully closed with a negative step, and a ramp was applied as command signal. After the throttle was fully open, the ramp changed direction and closed the throttle again. The total sweep time was 120 seconds, and a few different magnitudes of the slope on the ramp were tested. A typical measurement reading can be seen in Figure 9 below.

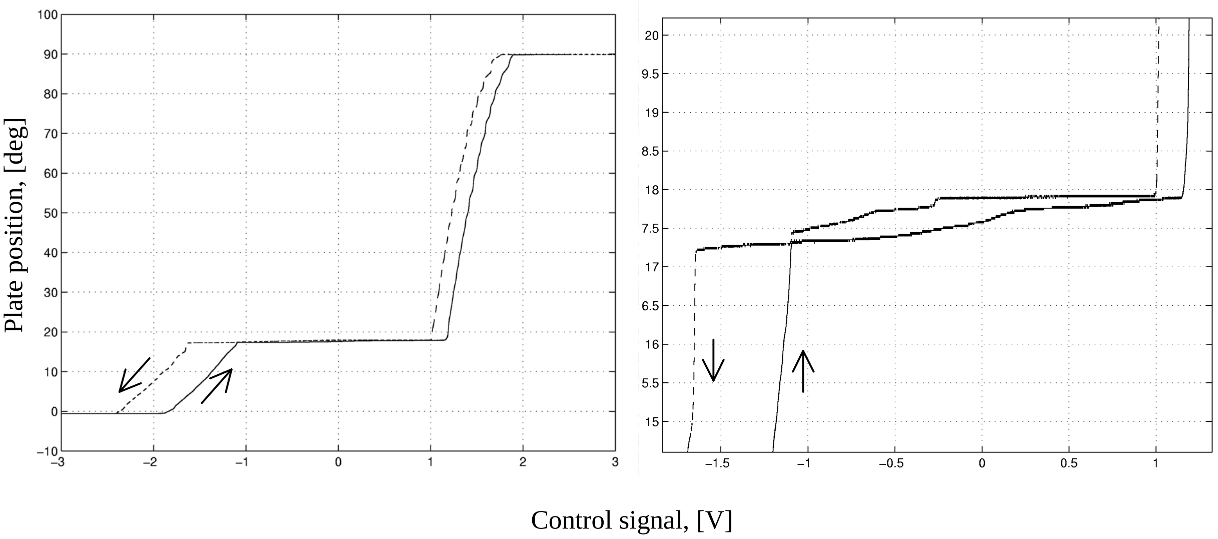


Figure 9. To the left is a typical sweep of the throttle from closed to open and back again. To the right is a close-up around the dead-zone position. The gains of the system are different above and below the dead-zone, but it can also be seen that the gains are not constant within these areas. This implies the use of gain scheduling in the control law. The close up shows the complex behaviour of the dead-zone, which is impossible to fully model. The friction in the springs etc. creates a gap between the measurements of increasing and decreasing control signal.

It can be seen in Figure 9 that the dead-zone spans from -1.10 V to 1.15 V on the way up and from 1.0 V to -1.65 V on the way down. The plot also shows that the dead-zone has a width in the angular direction. This shows as a little slope from the starting point at 17.2° to the ending point at 18.0°. The scans in upward and downward directions are separated and different on both sides of the dead-zone. This is due to the friction in the system, which is the next thing to be examined.

4.5 Friction

The friction can be divided in two components. The first is the static friction, which has to be overcome before the metal plate can be moved. Once it has started to move the static friction can be neglected. There is also a dynamic friction part depending on the velocity of the plate. To determine the friction, the system is fed with a triangular wave as reference signal. In the Figure 10 below, it can easily be seen when the plate movement is started and stopped. The estimated friction contains both static and dynamic friction but the static contribution is the main part.

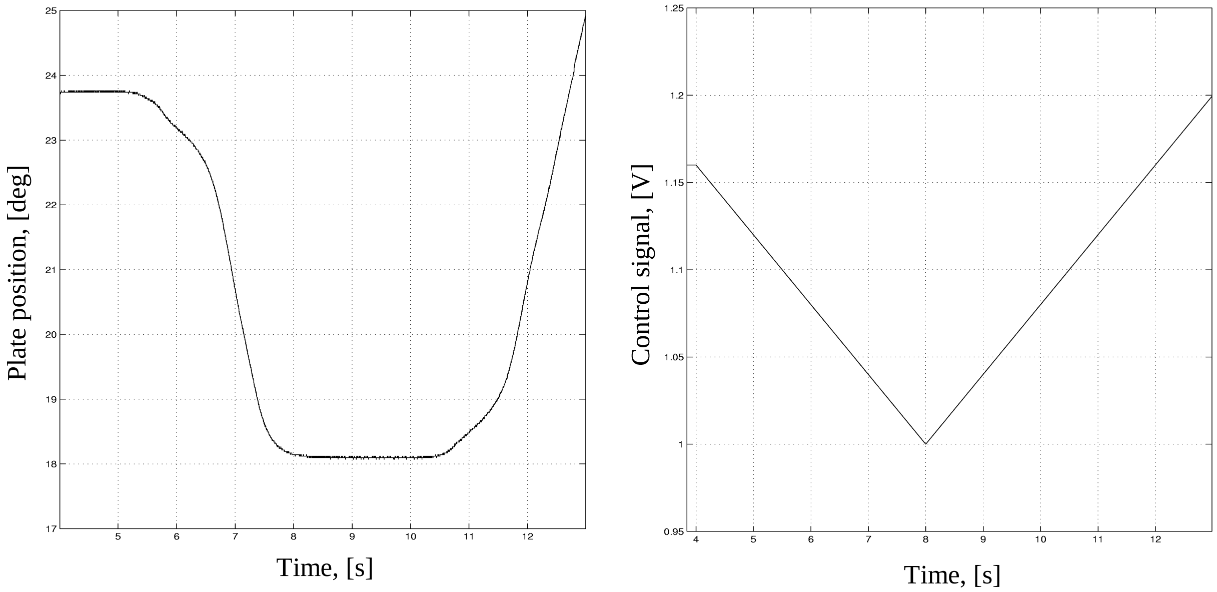


Figure 10. One measurement over the friction, above and close to, the dead-zone. The stick-slip motion due to friction is obvious. The friction will be a problem in laboratory experiments but has less importance in the real world due to the vibrations of the engine.

During the experiments some conclusions were drawn. The friction below the dead-zone position is much larger than above. The required voltage to make the plate start moving is larger than the applied voltage when it stops. After performing many experiments with different slopes of the triangular wave the control signal needed to overcome the friction was determined, see Table 1

	Above dead-zone	Below dead-zone
Min voltage to move plate, i.e. static friction.	0.046 V	0.31 V
Max voltage to fasten plate i.e. kinetic friction.	0.034 V	0.22 V

Table 1. Results from the friction experiments. There is an order of magnitude larger friction below the dead-zone than above.

To avoid any friction problems during the remaining experiment the command signal to the DC motor has to be at least 0.31 V when below the dead-zone and 0.046 V if above. Another way to reduce effects of friction is to add a dithering signal during the experiments. By introduction of this feature the plate will always be in motion and the static friction will be negligible.

4.6 Transient response analysis

The analysis of transient responses is a simple but efficient test to determine the main properties of the system, i.e. the dominating time constant, the stationary gain and the time delay. The test can also indicate whether the process is non-minimum phase or not. By performing several steps it is possible to investigate if the process is linear. From the physical modeling it is known that the process has three poles, one from the electrical parts and two from the spring package. The electrical pole is probably fast compared to the other two.

The first thing to investigate is the electrical time constant. It is not easy to obtain a good estimate since the time constant is so fast. The time constant can be found by blocking the throttle plate at a given angle so that it can not move from this position. A large control signal reference step is then sent through the pulse width modulator. The time it takes for the current in the motor to change its value is measured with an oscilloscope. The time is equal the electrical time constant and the time delay of the electrical parts. It could roughly be approximated as a time constant of 3 samples i.e. 3 ms or having a break-point at 333 rad/s.

The dominating time constants give a good estimation of a system's break frequencies. It is for a first order system defined as the time it takes until 63% of the steady state value is reached. Another definition is the time it takes between 10% and 90% of the steady state value, which is used in this thesis. The definitions are taken from [5].

The response to a step in the control signal was investigated to determine the main time constant. The procedure was repeated to get more prior information about the system for the main identification. Three working points were chosen on each side of the dead-zone. One was close to, the second in the middle and the third far away from the dead-zone.

Linearisation point	Main time constant	Static gain
-1.9 V, 5°	79 ms	20.1
-1.8 V, 8°	78 ms	22.2
-1.55 V, 14°	75 ms	16.1
1.2 V, 30°	108 ms	141.2
1.4 V, 60°	103 ms	163.4
1.5 V, 80°	91 ms	151.1

Table 2. Step response results at different angles. Both the dominating time constants and the gain of the system changes with the angular position. This creates difficulties when controlling the throttle, i.e. it might be necessary to use gain scheduling.

As can be seen in Table 2, the process is nonlinear as was suggested in the physical modeling. There are big differences between the two regions, above and below the dead-zone. Inside the region there are minor differences which a robust and well-dimensioned controller, or perhaps controllers, can handle. From the step response the conclusion can be drawn that the system is minimum phase since the plate does not move initially in the wrong direction during a step response. The time delay is also very small, less than one sample and will be neglected. This delay can easily be added later to the model during the parametric estimation. The electrical discrete time pole can be found at 0.717 and the main pole at 0.987 respectively 0.989 in the complex plane. Due to the fast sampling, the poles will be situated close to the unit circle, see equation 4.9.

$$Discrete\ pole = e^{(continuous\ pole \cdot h)} \tag{4.9}$$

There will also be two zeros from sampling of a continuous-time model. According to [3] the extra zeros could be found as follow:

$$z_d = z^2 + 4z + 1 \;\; when \;\; h \to 0 \tag{4.10}$$

From this equation the conclusion can be drawn that there could be zeros found outside the unit circle in the parametric model even if it is a continuous time minimum phase system. It will also later explain the high degree of the process model numerator.

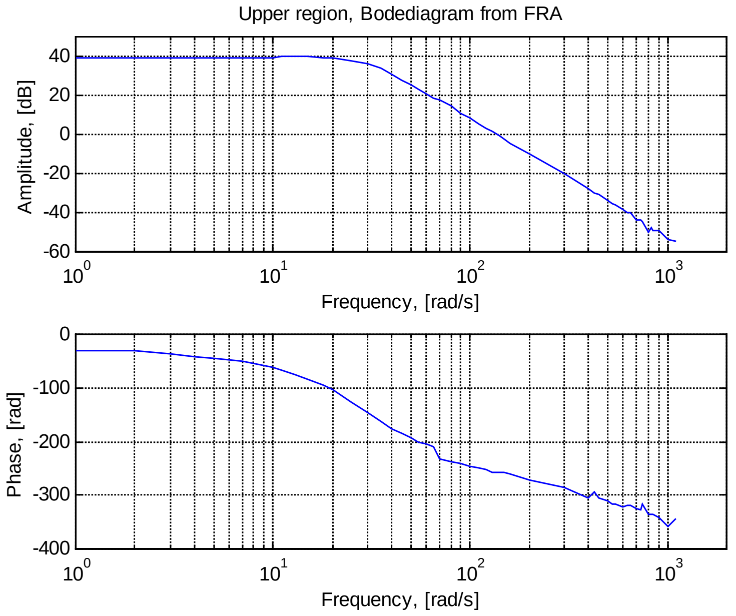
4.7 Frequency response analysis

To get more information about the process dynamics a Bode diagram can be constructed. This is done point by point using a method called frequency response analysis, FRA, described in [2]. The method is very time demanding and is not easy to perform at very low and high frequencies. The FRA is based on sending sinusoidal inputs, $u(t) = u_1 \sin \omega t$, with different frequencies to the system and using the obtained output to construct a Bode diagram. The algorithm multiplies the output with a sine and a cosine signal to reduce any disturbances. The estimated amplitude and phase of the process transfer function is then calculated from

$$\left|\hat{G}(iw)\right| = \frac{2}{Tu_1}\sqrt{s_T{}^2(w) + c_T{}^2(w)} \; , \; \arg\left(\hat{G}(iw)\right) = \arctan\left(\frac{s_T}{c_T}\right) \tag{4.11}$$

$$s_T(w) = \int_0^T y(t)\sin wtdt \, , \; c_T\,(w) = \int_0^T y(t)\cos wtdt \, , \; T = kh = \frac{2\pi}{w}k$$

Here u₁ is the input amplitude and w is the input frequency. The constructed diagrams will later be used to validate the results of parametric modeling and model reduction. There were some problems during the FRA experiments that reduced the accuracy of the method. The responses to control signals with low frequencies were disturbed by friction. Instead of a sinusoidal the measurement signal resembled a square wave. This had a large impact on both the amplitude and frequency estimates. To compensate for friction a dithering signal was used. The dithering makes sure that the system never reaches stationarity. This was not a problem at low frequencies since the frequency of the dithering signal was rather high. Another problem became visible at high frequency control signals. Now the amplitude of the measurement signal was low due to the fast changing control signal. When the signal amplitude was in the same order of magnitude as the resolution of the AD/DA converter, the measurements naturally became useless. Taking these two problems into consideration, it is reasonable to think that the Bode diagram from the FRA is only accurate in the frequency interval of 5 to 700 rad/s.



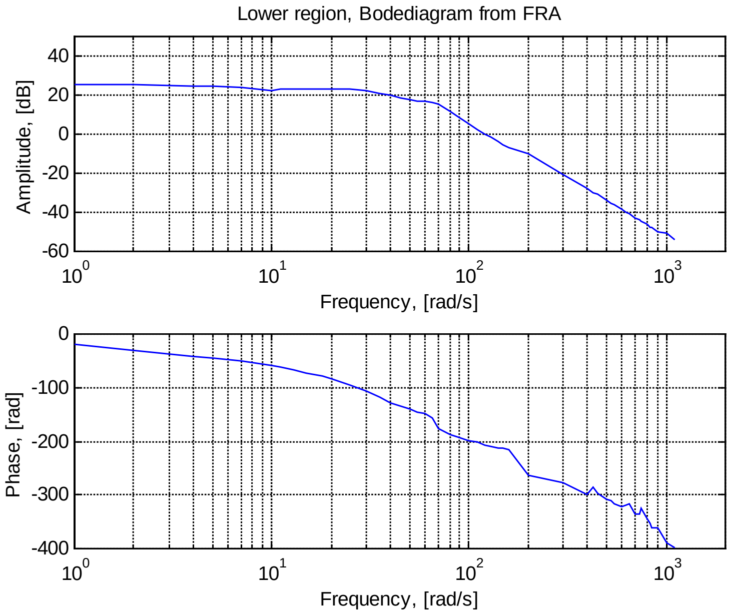


Figure 11. The resulting Bode diagram from a frequency response analysis. The first plot is from above, static gain of 40 dB and break frequency 130 rad/s, and the second plot is from below the dead-zone, static gain of 25 dB and break frequency 110 rad/s.

From the plots, the static gain can be estimated to 23 dB below dead-zone and 40 dB above. The break frequencies of the open-loop system are approximately 110 rad/s in the lower area and 140 rad/s in the upper. The Bode diagram in Figure 11 can be later be compared with the ones obtained from the parametric models.

4.8 Parametric model

So far only experiments giving an approximate model of the system and its transfer function has been performed. It is mostly graphical estimations and the major characteristics of the system that have been found. These estimations are enough to make a controller, but if a robust controller is needed like in our case, a parametric estimation based on numerical methods has to be done. There are lots of models and ways to do the calculations. The common factor is the procedure how it is done. One of the available parametric models is the Box Jenkins model. It has the following expression,

$$A(q)y(k)=\frac{B(q)}{F(q)}u(k-n_k)+\frac{C(q)}{D(q)}e(k) \tag{4.12}$$

Here u is the input, y the output and e is the noise. The constant n_k represents the delay between the input and output signal. A simpler model structure is the ARMAX model, which has the following format,

$$A(q)y(k)=B(q)u(k-n_k)+C(q)e(k) \tag{4.13}$$

and the state-space model using both noise and disturbances.

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + v(k) \\ y(k) &= Cx(k) + Du(k) + e(k) \end{aligned} \tag{4.14}$$

The estimations were performed using the system identification toolbox in Matlab. In the program it is easy and fast to treat data and fit it to a model with a given order. The data can e.g. be split in two halves, one for estimation and the second for validation. It is also possible to do some filtering and remove trends and biases easily. The software also proposes a model order that gives good result.

Identification was done on both sides of the dead-zone at the same angles as in the transient response analysis. The input signal has to be chosen very carefully. If the signal has the same frequency spectrum as white noise, it is possible to estimate parameters at all frequencies up to the Nyquist frequency. The reason not to use white noise directly is that it has the same energy at all frequencies, but in a normal estimation the low and high frequencies are not so interesting. Instead a pseudo random binary signal, or PRBS, was generated as input signal to the system. The PRBS signal has almost the same properties as white noise but is frequency restricted. Two PRBS-signals were chosen to give good excitation up to a certain frequency, one up to 300 rad/s and the second up to 400 rad/s. The signals were chosen based on the desired bandwidth of the system and the coherence spectrum. The amplitude of the input signal was 0.3 V in the lower area and 0.1 V in the upper region.

For a low order model the Box Jenkins gave best result. The optimal model order was 3. The State-space and ARMAX models gave the same result, just a little less accurate. The models were compared in a Bode diagram with the result from the frequency response analysis, see Figure 12 below.

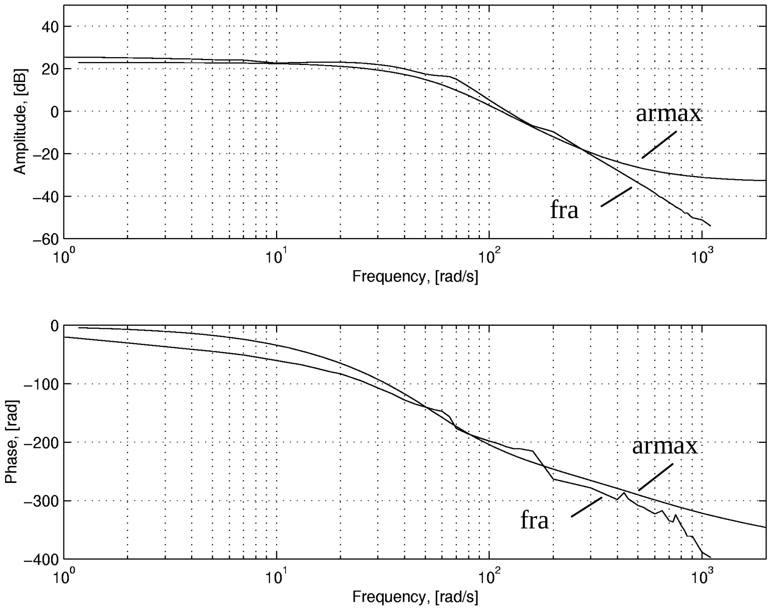


Figure 12. Example of Bode diagram of a estimate third order ARMAX model and frequency response analysis. The estimation has be done below the dead-zone with a PRBS signal of 400 rad/s and an amplitude of 0.3 V. The model fits well with the FRA up to about 300 rad/s, after that the amplitude is different.

The plot shows that there is no good fit over 300 rad/s and that the crossover frequency is below the desired bandwidth, about 400 rad/s. A new attempt was done in the same manner as before, but this time the model was overestimated to a tenth order model. Best result was achieved with the ARMAX model structure. The model was then reduced using a discrete model reduction, based on balanced realisation, to a third order model see Figure 13. Since the direct term of the model was very small it was for convenience set to 0.

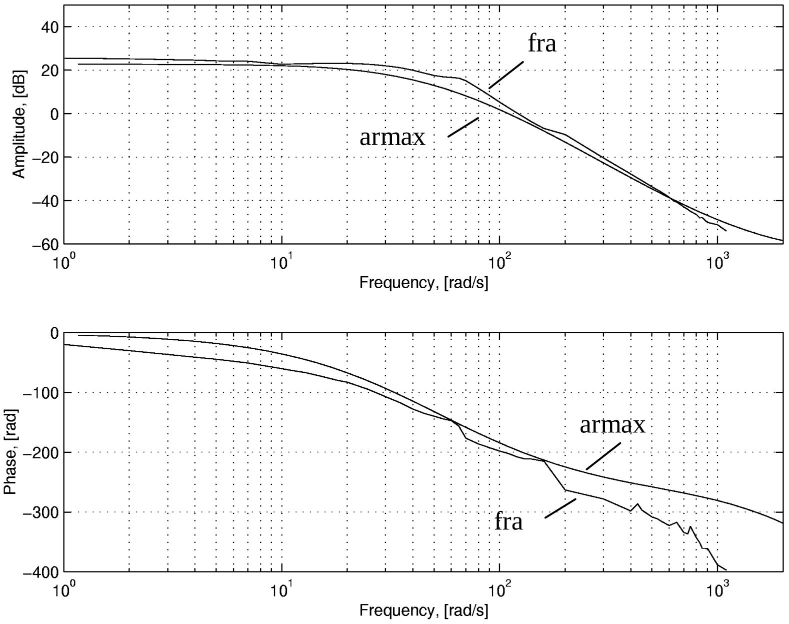


Figure 13. Example of Bode diagram of an estimated tenth order ARMAX model reduced to a third order and the results from frequency response analysis. The estimation has been done below the dead-zone with a PRBS signal of 400 rad/s and amplitude of 0.3 V. The model seems to fits the FRA well at all frequencies. The phase curve has the same characteristics as the FRA.

This model fits very accurately up to 700 rad/s, which is far higher than the desired bandwidth. The best model was estimated using the 400 rad/s PRBS signal. The estimated models at each angle can be seen in Table 3.

Estimation position	ARMAX model polynomials
-1.9 V, 5°	$A = q^3 - 2.8704q^2 + 2.7485q - 0.8779$ $B = -0.0370q^2 + 0.0108q - 0.0055$ $C = 0.4315q^2 - 0.6964q + 0.0133$
-1.8 V, 8°	$A = q^3 - 2.8651q^2 + 2.7379q - 0.8725$ $B = -0.0050q^2 + 0.0133q - 0.0066$ $C = 0.4544q^2 - 0.7264q + 0.2927$
-1.55 V, 14°	$A = q^3 - 2.8309q^2 + 2.6704q - 0.1839$ $B = -0.0009q^2 + 0.0042q - 0.0015$ $C = 1.9204q^3 - 1.9826q^2 - 1.7029q + 2.2219$
1.2 V, 30°	$A = q^3 - 2.9284q^2 + 2.8597q - 0.9313$ $B = -0.0034q^2 + 0.0091q - 0.0036$ $C = 0.4157q^2 - 0.7456q + 0.3403$
1.4 V, 60°	$A = q^3 - 2.9102q^2 + 2.8236q - 0.9134$ $B = 0.0005q^2 + 0.0005q - 0.0013$ $C = 0.4212q^2 - 0.7581q + 0.3490$
1.5 V, 80°	$A = q^3 - 2.9091q^2 + 2.8223q - 0.9132$ $B = -0.0019q^2 + 0.0055q - 0.0015$ $C = 0.4576q^2 - 0.8274q + 0.3490$

Table 3. Parameters of third order ARMAX model reduced from a tenth order model. The system poles are about the same as in the direct third order estimation but the zeros are different.

4.9 Validation

When a parametric model has been estimated, it is not known if the model is a sufficiently accurate description of the process so that a good controller can be calculated. There are many tests that can be applied on the estimated model to determine if it is suitable for controller design. These tests are used to e.g. check if the model order is sufficient, if there is any covariance between input signal and output signal, and if it can predict the output for a given input signal.

A selected model order is usually checked with a method based on the least square of a modified loss function. The loss function is used together with residual analysis, where a low order model is preferable. The model order was estimated to be order 2 or 3, depending on different measurements and noise components. Using the Bode plots as earlier described when reducing a higher order model to a lower order the result is that a third order model should be used.

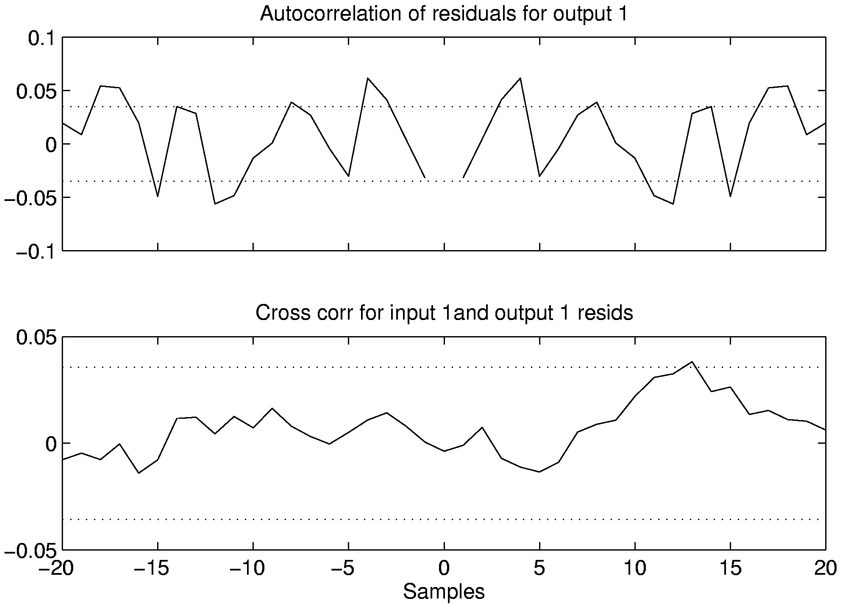


Figure 14. Example of the auto correlation respectively the cross correlation of the reduced ARMAX model below the dead -zone. The auto correlation plot is not perfect. This implies that the remaining unmodelled noise is not purely white noise, i.e. some dynamics in the system are neglected. The cross correlation plot indicates that the inputs do not affect the residuals.

Figure 14 above shows that the cross correlation between the input signal and the residuals are inside the 99 % confident interval. This indicates that different inputs do not affect the residuals. However, if the model is tested by examining the auto correlation of the residuals, it shows that there is some dependence between residuals at different sample times. This implies that the remaining unmodelled noise component is not white noise. A higher order model would give a better result but requires more parameters to tune in the controller design and is therefore not wanted.

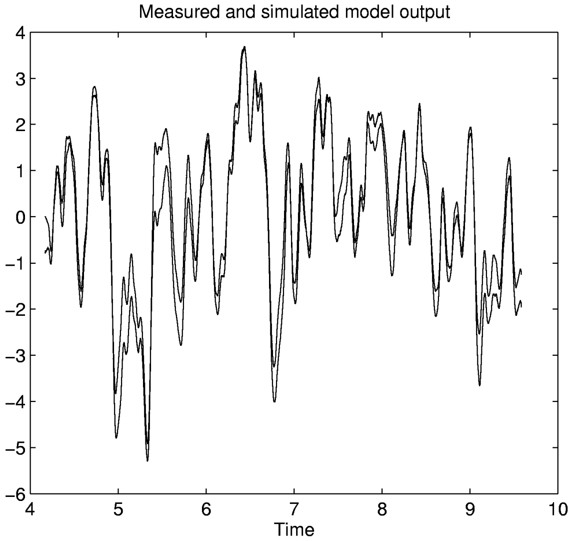


Figure 15. Example of validation of the reduced third order ARMAX model. The model predicts the measured signal with an accurate result. The validation was performed below the dead-zone.

The validation of the reduced third order model shows that it can predict and follow an output signal without any large problems. Since all tests, except the auto correlation, give satisfactory results, the model is probably a sufficiently good description of the real process. Some correlation between the residuals can be accepted from the controller design point of view.

5 Start-up experiments

5.1 Introduction

Simulations show that accurate dead-zone parameter estimations are crucial for the control requirements to be fulfilled. The manufacturer, Magneti Marelli, includes an estimate of these parameters in their data sheet of the throttle. The estimates have been found by comparing the characteristics of different throttle units. The parameter estimates in the data sheet are then represented in intervals rather than exact values. This results in unnecessary slow control. Therefore it is preferred to use an individual estimate for every throttle unit.

A good estimate of parameters for compensation of a nonlinearity can be found by performing simple experiments. If a process is constant, i.e. the parameters never change, there is no need to update the estimates when the throttle is used and the parameter estimation can be done once and for all. If the assumption of a constant throttle process is made, the dead-zone could be determined by start-up experiments. According to [10] the available time for experiments at the start-up sequence of the engine is 300 ms. This implies that it should be possible to get accurate estimates of the dead-zone compensation parameters before control of the throttle is needed. Different ways to achieve this will be examined next.

5.2 Calibration

Before any experiments can be carried out some calibration of the throttle has to be done. First is the scaling factor from voltage to degrees determined by practical measurements of the lower and upper limits. In the beginning of all experiments the dead-zone position is measured at the first sample. When the plate saturates in the lower area the lower limit is measured. The lower saturation position corresponds to -0.5°, according to [10]. At 90° the throttle is fully open which is the same as the upper saturation point. The calibration parameters are then:

- The conversion factor between volts and degrees:	4.92 °/V
- The upper saturation point	9.97 V or 90°
- The lower saturation point:	-8.48 V or –0.5°
- The dead-zone position at start-up:	-4.72 V or 17.55°

5.3 Control signal ramping

The first thing that comes to mind when determining the edges of the dead-zone compensation is to ramp the control signal in open-loop and observe when the plate starts to move. The corresponding control signal would then be the edge of the dead-zone. Simulations show that this procedure is both inaccurate and time consuming. The main problem is that the signals need to be filtered before they can be used. This filter introduces a time delay and therefore it is difficult to determine the exact time when the throttle starts to move. Another drawback of this method is that once the system starts to move and the first dead-zone edge is found the system needs to relax to stationarity before the second edge can be found.

5.4 Step responses in open-loop

A second approach to the problem is to perform sequential step responses in open-loop and then calculate the gain and dead-zone edges. Before the experiment begins, the dead-zone position is measured and the plate is moved from the dead-zone to reduce friction effects in the estimates. The first two steps of the responses should be in the same direction, e.g. downwards, and is used to calculate the gain of the process and the dead-zone edge, *b*. Next the lower saturation point is measured to calibrate the potentiometer. Now the plate is moved to the other side of the dead-zone. This is done with a ramp control signal to prevent oscillations. The following steps, directed upwards, give the gain above the dead-zone together with the positive dead-zone edge. A real experiment on the throttle is used as an example of how the dead-zone parameters could be estimated when there is no time limit for performing experiments, see Figure 16.

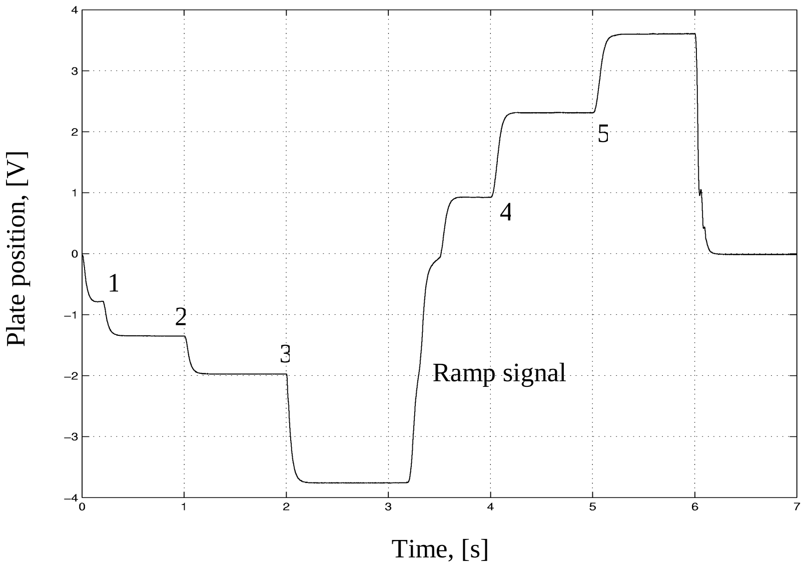


Figure 16. Open-loop experiments on the real throttle when the available time for experiments is long. The step responses are used to calculate the gain of the process and to give an estimate of the spring torque.

The relation at stationarity between input and output of the linear process is as follows,

$$y = K \cdot (u - b) \tag{5.1}$$

where *K* is the gain of the process and *b* is the dead-zone edge. The step response method has the following format.

The relation between step 2 and step 3 respective step 4 and step 5 is

$$K_l = \frac{(y_3 - y_2)}{(u_3 - u_2)}, \quad K_u = \frac{(y_5 - y_4)}{(u_5 - u_4)} \tag{5.2}$$

giving the gains K_l and K_u. The negative dead-zone edge, b_l, can now be extracted from step 1.

$$b_l = u_1 - \frac{y_1}{K_l} \tag{5.3}$$

The positive edge, b_u, is calculated from step 5.

$$b_u = u_5 - \frac{y_5}{K_u} \tag{5.4}$$

The variable y in the method is the output at stationarity. Since the system is affected by noise, a true stationary value will never be reached. Therefore *y* is calculated as the mean of the output at the latter part of the step response. This way the influence of noise is discarded. The method gives good estimates if the total time at hand is approximately 2 s to make estimation on both sides. If only one side e.g. below the dead-zone is wanted, the time needed to carry out the experiments is about 550 ms, see Figure 17. Since only 300 ms is available for measurements the method needs to be redesigned.

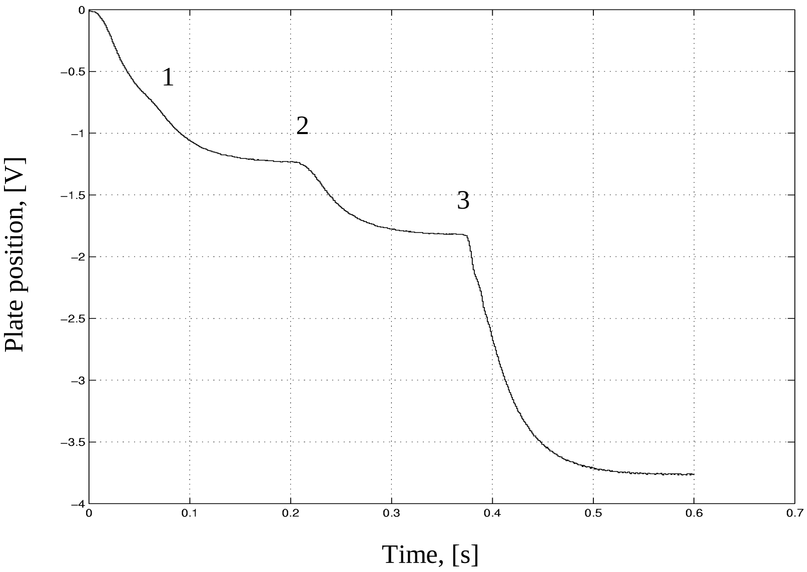


Figure 17. An open-loop experiments on the real throttle when short time is available. From this plot the gain of the process is calculated and an estimate of the spring torque is given by the relation in (5.1).

5.5 Step responses in closed-loop

To speed up the estimation procedure the step responses could be performed in closed-loop. The introduced proportional controller has unit gain and the output is fed back scaled with a constant factor γ, see Figure 18. The feedback constant, γ, is chosen to make the step responses fast and non oscillative.

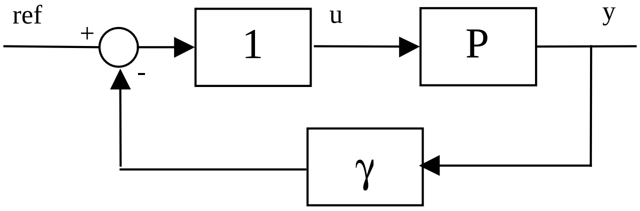


Figure 18. The closed-loop experiment presented as block diagram. The feedback gain is introduced to speed up the step responses settling time.

The input-output relation now becomes,

$$y = (u - \gamma \cdot y - b) \cdot K \tag{5.5}$$

The experiment is performed in the same manner as in the open-loop case but the different step responses are faster, compare with Figure 16. The closed-loop method is the following.

The relation between step 1 and step 2 respective step 4 and step 5 in closed-loop is,

$$K_l = \frac{(y_2 - y_1)}{(u_2 - u_1 - \gamma_l (y_2 - y_1))}, \quad K_u = \frac{(y_5 - y_4)}{(u_5 - u_4 - \gamma_u (y_5 - y_4))} \tag{5.6}$$

which gives the closed-loop gain K_l an K_u. From step 2 the lower edge of the dead-zone can be estimated.

$$b_l = u_2 - \gamma_l \cdot y_2 - \frac{y_2}{K_l} \tag{5.7}$$

In the same way the upper edge can be found from step 5.

$$b_u = u_5 - \gamma_u \cdot y_5 - \frac{y_5}{K_u} \tag{5.8}$$

The redesigned step response method is faster than the previous method but still can not perform the estimation within the 300 ms. The one sided experiment takes about 400 ms.

5.6 Performance and results

The experiments gave the following results.

	Open-loop	Closed-loop	Real process
K_u	128,2	171,6	100
K_l	18,0	27,0	15
B_u	1,09	1,12	1,05
B_l	-1,29	-1,36	-1,20

Table 4. Estimated parameter values during the start-up experiments. The "real process" values are obtained by adaptive parameter estimation, and identification experiments.

The step response method gives very good estimates of the real edges of the dead-zone. The closed-loop experiment is faster but not as accurate in the estimations as the open-loop experiment. The estimated values can be used as initial values to online estimators. The factor, γ, is found by trial and error. The signals must not saturate otherwise the estimation is ruined.

The time available for start-up experiments is not enough to estimate parameters on both sides with the closed loop system above. Instead one side of the dead-zone could be estimated during the start-up phase and the other side while shutting down the engine. By saving the last estimated parameters and choosing the estimation side of the test randomly, there would be enough time to estimate both parameters. A better way would be to use the linear controller for the system. The time for the system to reach stationarity would then decrease significantly.

The estimation procedure could also be performed at e.g. recurrent services or repairs at authorised workshops. These types of estimations should naturally also be performed during the manufacturing of the car.

6 Linear controller design

6.1 Controller design

After the process has been identified it is possible to derive a suitable controller. The controller can either be designed in discrete or continuous-time using numerical and/or graphical methods. Since the identified process is on discrete-time parametrical form, the controller will be designed using a numerical method. Graphical methods, e.g. designs from Bode or Nyquist plots will be used to confirm the performance of the controller. The criteria could be the bandwidth, phase or amplitude margin and the behaviour at low respectively high frequencies. The identification showed that the process parameters are different in the areas above and below the dead-zone. There are differences in both the static gain and the main time constants. This implies that the controller structure should be robust and consist of at least two, perhaps more, linear controllers. Bumpless transfer will be used together with antiwindup schemes. This prevents problems when switching between controllers and integrator wind up when the command signal saturates. The problems can be handled by introduction of tracking of the output signal, *u* and *v*, see Figure 19.

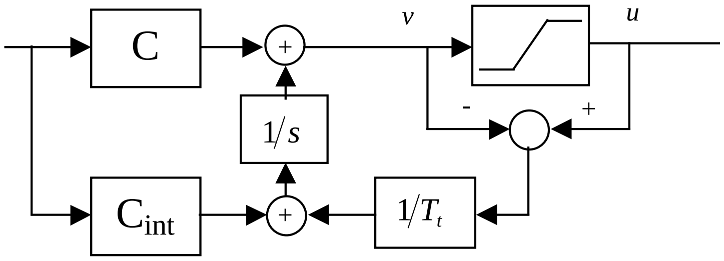


Figure 19. Block scheme that shows tracking of the signals *u* and *v*. The tracking signal is added to the integral action part of the controller, C_{int}. The scheme is used to implement integrator windup. Bumpless transfer between controllers is performed by exchanging the signal, u, for the control signal of the active controller.

After the controller had been calculated and evaluated it was tested in a simulation model. The model was built in Matlab according to the results from the identification. Some measurement noise and other disturbances was also be added to test the robustness of the controllers.

For simplicity only the calculations corresponding to the controller just above the dead-zone will be shown. Other controllers needed to achieve good performance would be calculated in the same manner. The results will be presented in the end of the sections for each design method.

6.2 A polynomial approach

One way to calculate the controller is to do it on polynomial form using pole placement. In the literature many such design methods can be found. Since the process is discrete, a method based on discrete pole placement will be used in this thesis. One example of this is the RST-controller which is a discrete, two-degree of freedom controller. A brief description of the design procedure will be given below. For the interested reader the design method along with proofs can be found in [3]. The controller has the following format,

$$R(q)\cdot u(k) = T(q)\cdot u_C(k) - S(q)\cdot y(k) \tag{6.1}$$

where *R*, *S* and *T* are polynomials in the shift operator, *q*. The signal, *u*, is command signal, u_c is the reference signal and *y* the output. The controller can also be viewed in a block diagram as,

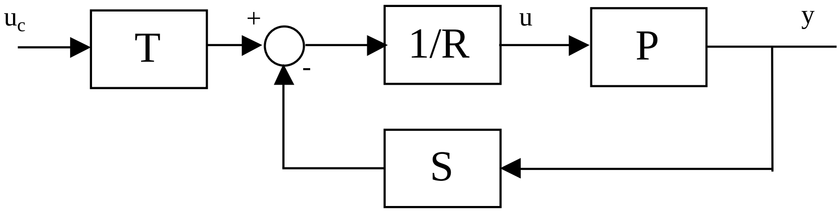


Figure 20. Block diagram for a RST-controller. The controller consists of a feedforward filter, T/R, and a feedback filter, -S/R.

By choosing a characteristic polynomial it is possible to find a controller that gives desired closed-loop dynamics. The design polynomial consists of two parts, the faster observer polynomial, $A_o(z)$, and slower controller polynomial, $A_m(z)$. The characteristic polynomial of the closed-loop system is then,

$$A_{cl}(z) = A_o(z) \cdot A_m(z) \tag{6.2}$$

but it can also be written as,

$$A_{cl}(z) = A(z) \cdot R(z) + B(z) \cdot S(z) \tag{6.3}$$

Where *B(z)* and *A(z)* is the process numerator respective denominator. The equation 6.3 is a so called Diophantine equation and can be solved inserting equation 6.2 giving,

$$A(z) \cdot R(z) + B(z) \cdot S(z) = A_o(z) \cdot A_m(z) \tag{6.4}$$

The equation above is a linear equation system and the polynomial coefficients of *R* and *S* can easily be found. The next step is to determine the polynomial *T*, which shapes the step response of the closed-loop system. It is often chosen to cancel the observer polynomial $A_o(z)$. This gives the following relation,

$$T(z) = t_0 \cdot A_o(z) \tag{6.5}$$

The t_0 parameter decides the static gain of the closed-loop system. Normally it is chosen so that the system has unit gain,

$$t_0 = A_c(1)/B(1) \tag{6.6}$$

An integrator can easily be added to the controller by including the factor *(z-1)* in the *R* polynomial. This increases the degree of the Diophantine equation. In a similar fashion, a notch filter can be added to the *S* polynomial.

In the Matlab local toolbox ppbox there is a function called rstd, which calculates a controller by solving the Diophantine equation. Input parameters to the function are, *$A_o(z)$, $A_m(z)$,* the process and whether the controller should contain additional terms in *R(z)* and/or *S(z)*.

In the throttle case the controller finally after many iterations consists of a rather slow closed-loop dynamics in the controller polynomial with continuous-time poles placed at 27 rad/s and 100±60j rad/s. The real pole is chosen to be the slowest one. The observer polynomial is must faster, so it can detect fast changes in the process. Also here is the real pole the slowest one. This is because the real pole in the observer polynomial can be interpreted as the integrator part of the controller. The integrator has been added to eliminate static control errors. The poles of the observer are located at 70 rad/s and 730±210j rad/s respectively. The designed controller has a bandwidth of about 200 rad/s and is fast enough according to the desired step time, see Figure 21.

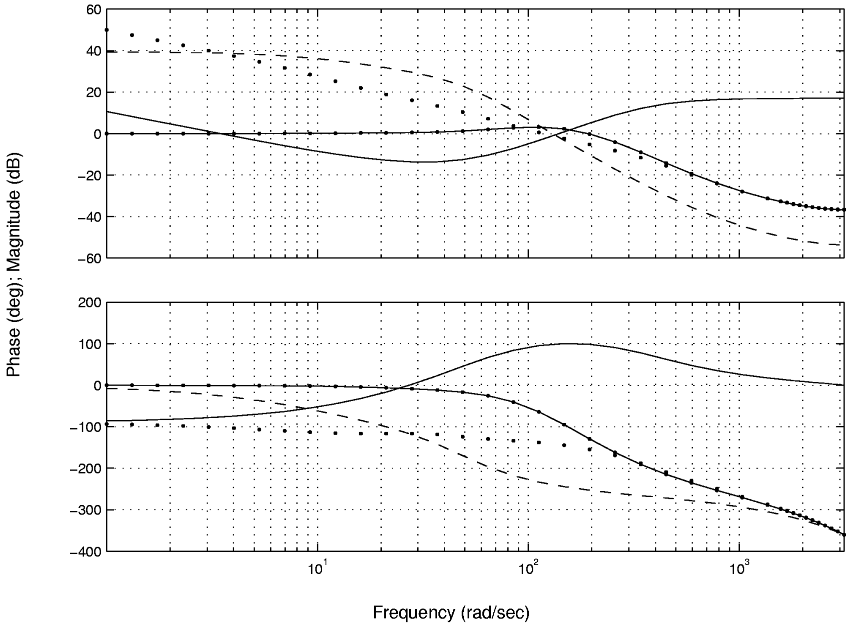


Figure 21. Example of Bode diagram showing the upper controller (solid), the process (--), open-loop (..) and close loop behaviour(.-) below the dead-zone. The controller has a large phase addition around 150 rad/s and a large gain at high frequencies, which makes it sensitive to high frequency noise.

The Bode diagram clearly shows that there is need for a phase advance at high frequency. As a result of the increased phase, the gain is automatically also increased at high frequencies. The high gain results in amplification of measurement noise, which therefore probably will be a problem.

When testing the controller in simulations it behaves as expected with good step responses. After implementing it on the real process it was noticed that the controller was sensitive to noise and could not follow the reference signal fast enough. For some reason the plate stopped before it had reached the desired angular position and the integrator had to eliminate the static error. At first the step responses were too slow when comparing with the specifications. After a considerable amount of fine tuning the responses were satisfactory, see Figure 22.

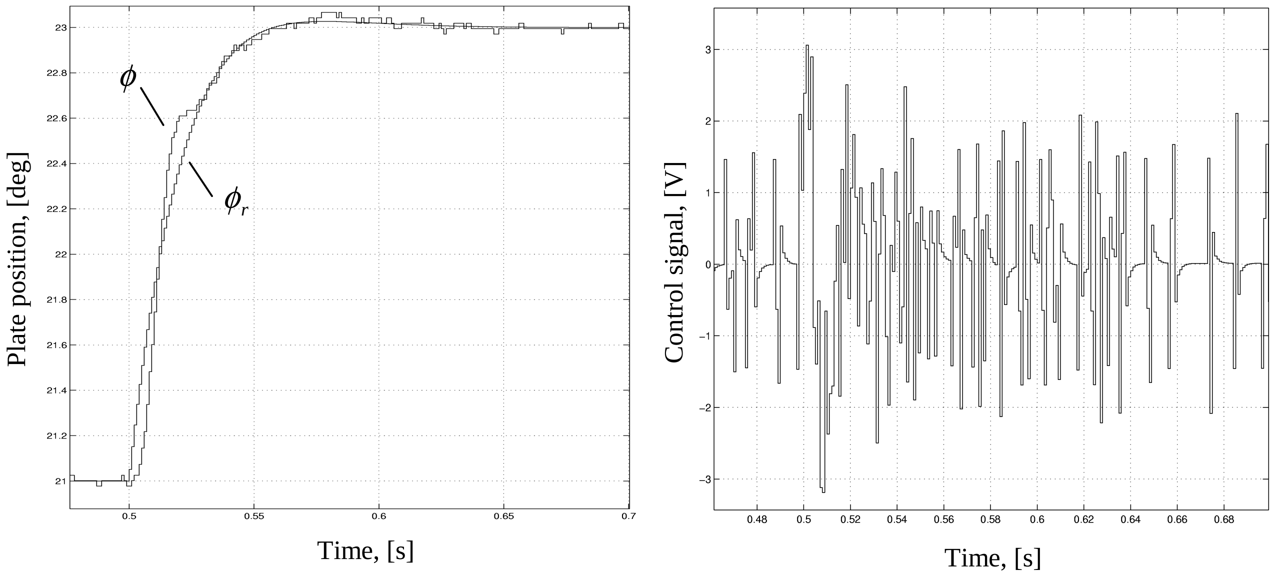


Figure 22. A 2 degree step response above the dead-zone at 21° and the corresponding uncompensated control signal. The step response meets the specifications from Volvo but the control signal is too noisy.

The following controllers gave the best result,

Working range	RST controller polynomials
Above the dead-zone	$R = q^3 - 1.4149q^2 + 0.3784q + 0.0366$ $S = 61.167q^3 - 173.72q^2 + 164.55q - 51.974$ $T = 0.3788q^3 - 0.7179q^2 + 0.4330q + 0.0849$
Below the dead-zone	$R = q^3 - 1.6872q^2 + 0.7067q - 0.0195$ $S = 32.324q^3 - 92.027q^2 + 87.317q - 27.612$ $T = 0.1249q^3 - 0.2481q^2 + 0.1592q + 0.03349$

Table 5. Polynomials of the best RST controller above and below the dead-zone.

During the controller design some problems occurred. The first task was to place the poles in a good way. It is not easy to guess where the poles should be located. The use of trial and error became harder when the controller and observer polynomials grew in size. It was also discovered that it was not possible to place the poles arbitrarily. Sometimes the integrator in the *R(z)* polynomial was cancelled by a pole very close to *(z-1)* in the *S(z)* polynomial. After some investigation it was realised that it had to do with a numerical calculation problem in Matlab. Small changes in for example the seventh decimal in the process or in the design polynomials had large impact on the controller design result due to the fast sampling. As mentioned before, the control signal is very nervous and it is possible to hear and feel the amplified noise in the throttle plate. This is not good performance in the long term. As an effect of this an alternative controller design method was investigated.

6.3 State-space approach

Another way to design a linear controller is to use optimal control theory described in [3]. The process to control could be on either polynomial or state-space form. In this section the state-space method is chosen because of better calculation precision. The process is converted to state-space form using the Matlab function, `ssbal`, which gives a balanced realisation of the state-space system. A drawback with this function is that the parameters of the model will be calculated from a numerical point of view. It means that the obtained state-space form has no physical meaning at all. To use the state-space representation for controller design, all states have to be measurable or observable. Since only the angle is measurable a Kalman observer will be used. The observer design is treated in chapter 9.

The Linear Quadratic controller (LQ), uses a quadratic function of the states and the control signals to minimize a loss function. Since the real process is affected by unmodelled coloured noise, this method is suitable for deriving a robust linear controller. A short description of the method will be discussed here. The complete derivation of the method could be found in [3].

The process model can, after the conversion from polynomial to state-space form using `ssbal`, be written as,

$$\begin{aligned} x(kh+h) &= \Phi x(kh) + \Gamma u(kh) + v(kh) \\ y(kh) &= Cx(kh) + e(kh) \end{aligned} \tag{6.7}$$

The specific system matrices can be found in the Appendix A. A discrete loss function on the following format is used. It is assumed that the input is constant during one sample,

$$J(k) = x^T(kh)Q_1x(kh) + 2x^T(kh)Q_{12}u(kh) + u^T(kh)Q_2u(kh) \tag{6.8}$$

It is possible to introduce the following equation if *u(k)* is allowed to be a function of *x(k), x(k-1) …*

$$S(k) = \Phi^T S(k+1)\Phi + Q_1 - \left(\Phi^T S(k+1)\Gamma + Q_{12}\right)\cdot\left(\Gamma^T S(k+1)\Gamma + Q_2\right)^{-1}\left(\Gamma^T S(k+1)\Phi + Q_{12}^T\right) \tag{6.9}$$

with $S(N)=Q_0$. If Q_0 is positive semi definite and the inverse above is positive definite. Then the unique optimal controller can be found as a state feedback law,

$$u(k) = -L(k)x(k) \tag{6.10}$$

where the feedback matrix, *L*, that minimizes the loss function in equation 6.8., has the format,

$$L(k) = \left(Q_2 + \Gamma^T S(k+1)\Gamma\right)^{-1}\left(\Gamma^T S(k+1)\Phi + Q_{12}^T\right) \tag{6.11}$$

The equation for *S(k)* can be solved using backwards iteration to the initial time when *k=0* or by solving the algebraic time invariant Riccati equation. The Riccati equation has the following solution,

$$\overline{S} = \Phi^T\overline{S}\Phi + Q_1 - \left(\Phi^T\overline{S}\Gamma + Q_{12}\right)\left(\Gamma^T\overline{S}\Gamma + Q_2\right)^{-1}\left(\Gamma^T\overline{S}\Phi + Q_{12}^T\right). \tag{6.12}$$

A simple way to solve the equations above is to use the Matlab function `lqrd`. The design parameters in the method are the weighting matrices Q_1, Q_2 and Q_{12}. The linear controller is preferred to have integral action. In [3] a method is described for adding integral action to a state feedback controller. The following is the result of that method.

First an integrator is introduced as an extra state to the state-space system, see Figure 23.

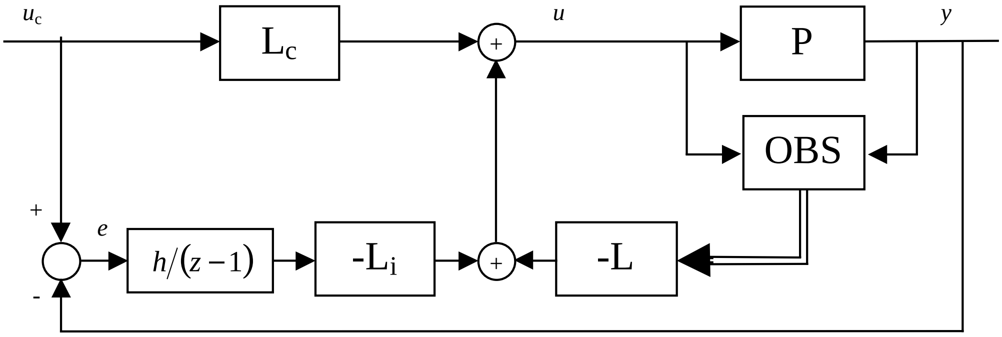


Figure 23. Block scheme describing how the integrator is introduced as an extra state. The standard state feedback matrices are extended with a state corresponding to the difference between the reference and the measured plate position.

The integrator extended system matrices are then,

$$\Phi_{ie} = \begin{bmatrix} \Phi & 0 \\ -h \cdot C & 1 \end{bmatrix}, \quad \Gamma_{ie} = \begin{bmatrix} \Gamma \\ 0 \end{bmatrix}, \quad C_{ie} = \begin{bmatrix} C & 0 \end{bmatrix} \tag{6.13}$$

Using the Matlab function `lqrd` and the extended system above, the state feedback controller with integral action can be calculated. To start with, the diagonal elements in Q_1 are set to one. Then the control signal is weighted by varying the Q_2 matrix. The cross term matrix Q_{12} is zero. Once good performance on the control signal has been found, the Q_2 is matrix is "fixed" at this value during the rest of the design procedure. Next the Q_1 matrix is varied with beginning on element $Q_{1(4,4)}$, the one corresponding to the integrator. The design continues in the same manner with all the elements on the diagonal until a satisfactory behaviour on the close loop system is obtained. In the end of the controller design smaller adjustment on the Q_2 and Q_1 could be done to improve the performance. Perhaps the cross term matrix Q_{12} could also be used.

A useful way to move the poles of individual states is to use orthogonality of eigenvectors. To for example move the forth state in the state vector above, find a vector, q, that is orthogonal to the eigenvectors of the other states. Once such a vector is found, modify the weight matrices as follows,

$$Q_{1tot} = Q_1 + \alpha \cdot qq^T \tag{6.14}$$

This way weight is put on the specified state on not on others. The factor, α, determines the weight added in the specified direction. For example, the following weighting matrices resulted in a controller meeting the design demands. The weighting matrices for both controllers can be found in the Appendix A.

$$Q_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad Q_{12} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad Q_2 = [100], \quad q = \begin{bmatrix} 0.0027 \\ -0.0117 \\ 0.0090 \\ 1.0 \end{bmatrix}, \quad \alpha = 10e4 \tag{6.15}$$

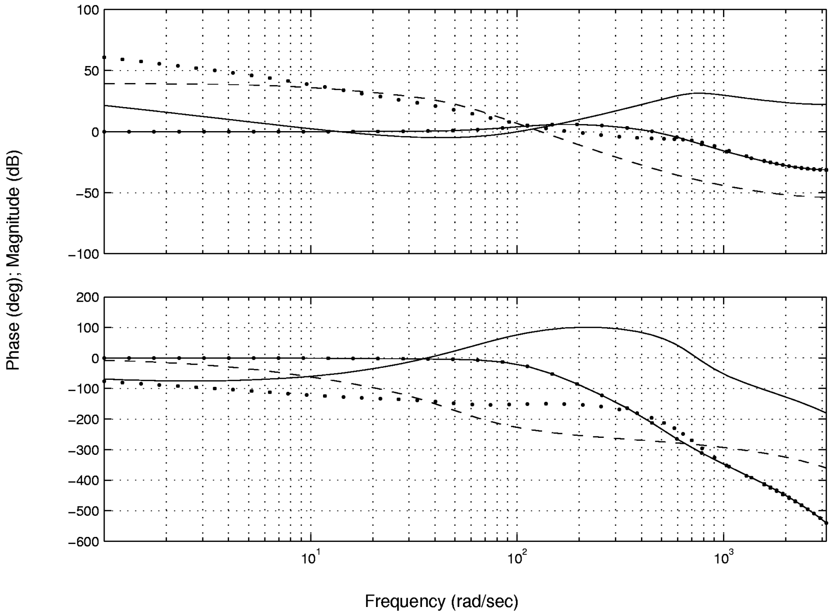


Figure 24. Example of Bode diagram showing the upper controller above and close to, the dead-zone. The process (--), open-loop (..) and close loop behaviour(.-). The controller has approximately the same features as the RST-controller, compare with Figure 21.

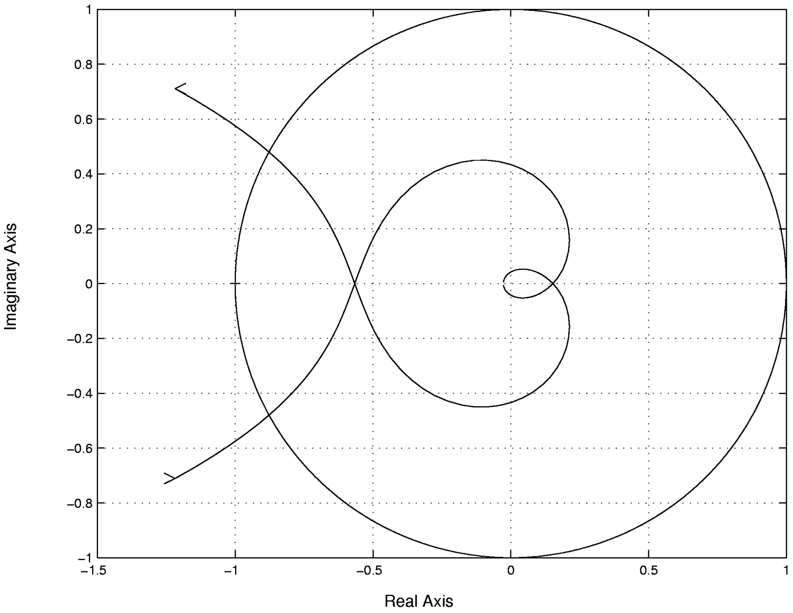


Figure 25. The Nyquist plot of the open-loop and controller, compare to the Bode diagram in Figure 24. The amplitude margin in the open-loop controlled system is about 4.4 dB.

A closed-loop bandwidth of 300rad/s, which is very fast, is the result of the design. This is needed to meet the demands on step responses in [10]. The LQG controller needs a quite large phase addition at high frequencies. Unfortunately the gain will also increase and the controller becomes noise sensitive. During the controller design the validation was made in a simulator, the same used in the polynomial design. When it worked fine it was finally implemented on the real process. Going from simulations to the real throttle did not cause any problems. There was no need the make any changes on the weighting matrices.

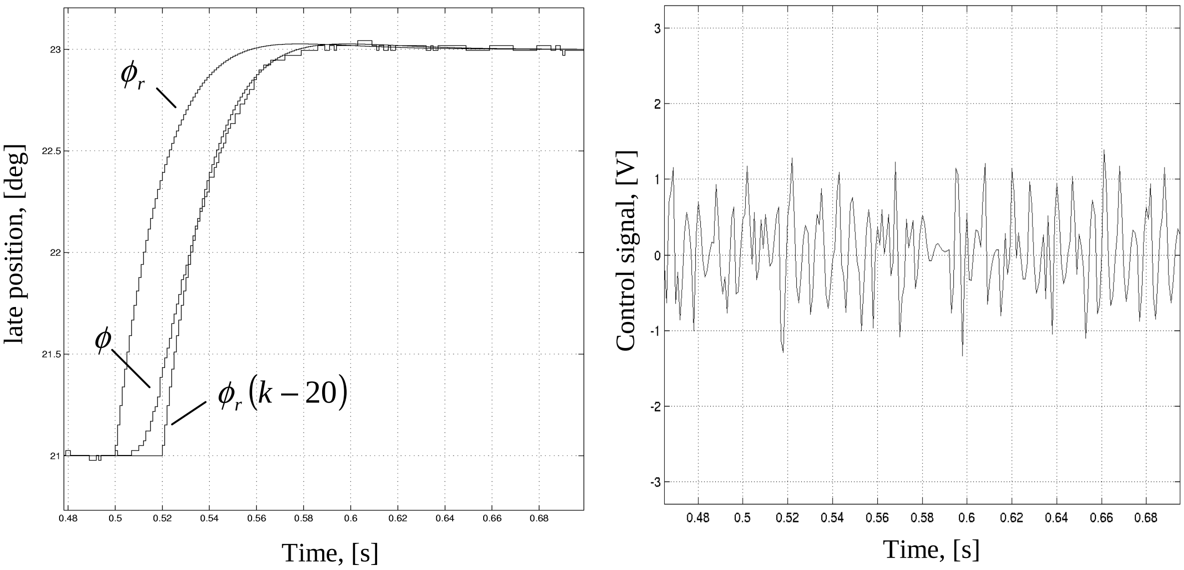


Figure 26. A 2 degree step response above the dead-zone at 20° and the corresponding uncompensated control signal. The step response meets the demands and the control signal is less noisy than for the RST-controller, compare with Figure 23. There are two reference signals in the plot. The second signal is delayed for 20ms and therefore represents the lower edge of the rise-time, according to the system specifications.

When investigating the command signal it shows that it is still shaky, but not as much as in the RST case.

The following controllers were obtained and gave good results.

Working range	LQG controller matrixes
Above the dead-zone	$L_c = [1]$ $L_i = [-0.1457 \quad 2.2668 \quad -1.8970 \quad -27.7395]$
Below the dead-zone	$L_c = [1]$ $L_i = [-0.0491 \quad 1.7510 \quad -1.4972 \quad -34.6661]$

Table 6. The feedback gain matrices for the best LQG controller, above and below the dead-zone.

The state feedback controllers above can be interpreted on polynomial form. The Matlab function `lqgd` was used to calculate the corresponding RST-controllers. Note the increase in degree compared with RST design. The extended matrices result in a polynomial representation with degree 4 instead of the previous 3. The two extra states are the integrator state in the controller and, as will later be seen, the torque estimation in the observer. For more information about the observer, see chapter 9 in this thesis.

Working range	Corresponding RST controller polynomials
Above the dead-zone	$R(q) = q^4 - 2.3279q^3 + 2.3693q^2 - 1.3526q + 0.3112$ $S(q) = 35.5616q^3 + -101.1234q^2 + 95.8813q - 30.3139$ $T(q) = q^4 - 2.5880q^3 + 2.6317q^2 - 1.2938q + 0.2501$
Below the dead-zone	$R(q) = q^4 - 2.2214q^3 + 2.0288q^2 - 10.9987q + 0.1913$ $S(q) = 29.6045q^3 + -82.2013q^2 + 76.0486q - 23.4427$ $T(q) = 1.1q^4 - 2.6323q^3 + 2.4505q^2 - 1.1145q + 0.1964$

Table 7. The RST polynomials of the converted LQG controller, above and below the dead-zone.

6.4 The noise

Since the obtained controllers are sensitive to noise there is need for some more investigation of the noise characteristics. A step was applied on the process and after a few seconds, when steady state has been reached, the noise affecting the plate was monitored. It was found to have an amplitude of approximately 0.05°, or two resolution levels in the AD/AD converter. An oscilloscope was also used to see if there was any noise in the electrical circuits, but none was found. The Bode diagrams shows that at high frequencies the gain in the controllers is approximately 20-40 dB, i.e. a gain of 10-20. This gives a large impact on the control signal.

A way to reduce problems due to noise, a better analogue low pass filter could be introduced, say at 1000 rad/s. This filter would have a large impact on the process dynamics. Since this implies that the system identification has to be performed once more, this option is not discussed further. It is probably the best way to reduce measurement noise however. For now the nervous behaviour of the controller is accepted.

6.5 RST Vs LQG

In this thesis two different approaches in linear controller design has been used, the polynomial and state-space. There are many controller structures and design methods based on the two approaches. The two methods chosen was pole placement, resulting in an RST controller, respectively an optimisation method giving the state feedback LQG controller.

Since both above mentioned design methods are available as toolboxes in Matlab, it is quite easy to do the calculations. The tricky part is to choose the controller and the observer polynomials in the RST controller. In the LQG controller the problem lies in choosing the weighting matrixes. It is more intuitive to use weighting matrices than observer and controller polynomials as design parameters. In the polynomial case the design parameters are often stated in continuous time because it is easier to place poles in continuous-time than in discrete-time. All the poles in the polynomials have to be chosen simultaneosly and it is hard to place e.g. 8 poles in a clever way.

The introduction of an integrator in the controllers can be done without any large problems. This is easier to do in the polynomial controller than in the state feedback controller. In the LQG controller the integrator has to be implemented as an extra state corresponding to the control error. If not all states are available for state feedback, an observer has to be designed and implemented. There is always the possibility to rewrite the LQG controller on RST form and then implement it as an ordinary polynomial controller.

7 Nonlinearities

7.1 About nonlinearities

A system is non-linear if it contains at least one non-linear component. All physical systems are nonlinear to some extent, e.g. systems effected by friction or a mechanical backlash. Nonlinearities are classified in terms of their origin as natural, which comes from the mechanical conditions and motion of the system, or as artificial which means that a control law constructor introduces them when controlling the system. Examples of natural nonlinearities are Coulomb friction between surfaces and air drag when an aeroplane is flying through the atmosphere. Nonlinear compensator methods such as sliding mode control and adaptive control laws are examples of artificial nonlinearities.

The continuous or discontinuous mathematical properties are another way of classifying nonlinearities. A discontinuous or "hard" nonlinearity can not be locally approximated with a linear function. Examples are backlash, dead-zones and relays. These can often be found in control systems and affect the system both in small and large range operation. Whether to consider a system as nonlinear or not depends on the magnitude of the hard nonlinearity and the effect it has on control performance of the system. Two good examples of continuous or "soft" nonlinearities are air drag and effects caused by the inertia of the system.

A third way of dividing nonlinearities is to examine if their behaviour is static or dynamic. By static means the behaviour does not depend on earlier states of the system, i.e. it is "memoryless". A dynamic nonlinearity on the other hand changes its response for different states and operating points of the system. An example of a static nonlinearity is a dead-zone, and of a dynamic, a backlash.
The definitions above are according to [7] and [8]. The three main nonlinearities in the throttle are a dead-zone, friction and inaccurate compensation. In the upcoming sections these effects will be examined more closely.

7.2 Dead-zone

A dead-zone is a static nonlinearity that describes a component insensitivity to small input signals. If a signal is below a certain "break-point" value, the output will not be affected. Above the break-point the output depends linearly on the input. This behaviour results in difficulties controlling the system and may be the reason for oscillations. The characteristic of a dead-zone model is shown in Figure 27.

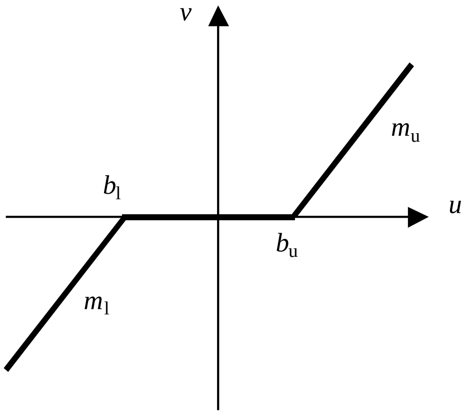


Figure 27. The characteristics of a dead-zone from input signal, *u*, to output signal, *v*. In an interval, [b_l, b_u], of the input signal, *u*, the output signal, *v*, is 0.

Here $b_u \geq 0$ and $b_l \leq 0$ represents the right and left "break-points", $m_u > 0$ and $m_l > 0$ are the "slopes". v is input and u is output signal. The slope represents a possible linear gain above and below the dead-zone. In general, none of the constants above are equal, but in this thesis m_u and m_l are taken to be equal to 1. A different gain than one is considered to be part of the dynamics of the throttle.

As previously described, the Magneti Marelli throttle has a built in "limp home" mode, realised by two opposite working torsion springs and a physical stop dividing the work ranges of the springs. In the system model this mode can be seen as a dead-zone active around the physical stop position of the throttle. The control signal has to exceed, in magnitude, the torque asserted by the springs in order to move the throttle. These minimum control signals are different above and below the "limp home" position, this since the two springs are not equally strong.

7.3 Friction

Friction forces appear between surfaces in contact. It is a dynamic nonlinearity that can be modelled as two different effects, static and kinetic friction. The static friction describes the force needed to start moving the surfaces against each other. Once the object is moving some of the static friction disappears. Kinetic friction on the other hand describes the forces between surfaces in motion. When the objects stops there is no more kinetic friction. The friction forces are also dependent on the direction of the motion. The magnitude of the static friction can be different in e.g. the forward and backward direction. Likewise the kinetic friction force, *F*, is opposite in direction and proportional to the velocity of the object in motion. In [8], friction is modelled as below, compare with Figure 28,

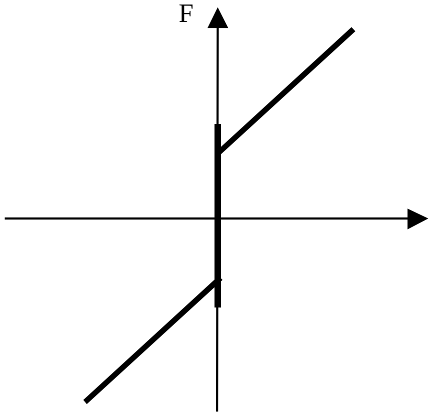


Figure 28. The characteristics of static and kinetic friction. The model includes stick-slip motion and constant kinetic friction.

$$F(t) = \begin{cases} F_c \, sign\, v(t) + F_v \, v(t) \,, & v(t) \neq 0 \\ F_s \, sign\, F_e, \quad |F_e| > F_s \,, & v(t) = 0 \\ F_e \,, \quad |F_e| \leq F_s \,, & v(t) = 0 \end{cases} \tag{7.1}$$

$$F_e(t) = external\ applied\ \ force, F_c, F_v, F_s\ \ are\, const.$$

The throttle can only move in two directions, i.e. positive or negative angular velocity. This implies the use of the definitions of friction in the equations above.

When the angular velocity of the throttle changes sign, the direction and magnitude of the kinetic friction also changes. It is the dynamics of the spring package that determines the behaviour of the kinetic friction. This implies that the kinetic friction has different influence on the system above and below the "limp home" position since there are two opposite working springs in the spring package.

After consideration and consulting of [10], the static and the kinetic friction are discarded in the system model. When mounted in a car, the vibrations from the engine and the road, keeps the throttle unit from obtaining a stationary position. It is therefore a reasonable approximation that no static friction is present in the system.

7.4 Compensation errors

An artificial nonlinearity is by definition created by the control loop designer. When the parameters of a nonlinear compensator are not known exactly, the compensator will make an error in the cancellation of the nonlinearity. The resulting nonlinearity could have the same properties as the first one but be smaller in magnitude, but it could also have different characteristics much worse to handle. The way to deal with these compensation errors is to make them as small as possible, so that they can be ignored.

Successful control of the throttle requires among other things, dead-zone compensation. Not only is it necessary to compensate for the springs with an accurate amount of control signal, but more important to know the exact moment when to change the direction of the compensation when passing through the dead-zone. This is because of the large control signals involved and the high bandwidth of the closed-loop system. The compensation assumes that the position of the throttle is known at every time instant. If there is a time delay in the system, an additional difficulty lies in predicting the angle of the throttle with good accuracy.

8 Describing function analysis

8.1 About describing functions

The design procedure of a control law for a system can consist of long and complicated mathematical expressions. The lack of physical meaning is a problem for the understanding of the process. To build intuitive knowledge, one can study the system in the frequency domain instead of the time domain. Examples of this are Bode-diagrams and Nyquist-plots, which facilitate the design for linear systems.

For the same reason as for linear systems, it would be a good idea to study systems with nonlinearities in the frequency domain. This is called describing function analysis and is used mainly to predict existence and stability of periodic solutions for a closed-loop system. The periodic solutions can reduce the efficiency of an otherwise perfectly adequate control law. A stable oscillation is called "limit cycle" and is a common problem in nonlinear systems.

There are many spectacular and expensive examples of system failures due to limit cycles. One particular expensive example is the Swedish military aircraft, JAS 39 Gripen, that crashed in Stockholm in 1993. The following investigation pointed out the reason for the crash, so called pilot induced oscillations. This is the same thing as having a limit cycle appearing when the pilot acts as a relay, i.e. banking hard right, hard left, hard right, etc. This example shows the importance of predicting and compensating for periodic solutions in processes with nonlinearities.

8.2 Theory

The key idea in describing function analysis is to approximate a nonlinearity with a complex function. The analysis searches for periodic solutions to the closed-loop system. This is done by Fourier expansion of the output from the nonlinearity, see Figure 29. The predictions derived from this analysis are frequency and amplitude of oscillations on the input to the nonlinearity.

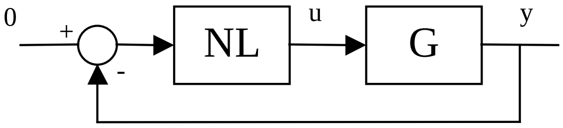


Figure 29. Nonlinearity, NL, on the input to a system, G, in closed-loop. Note that the input to the nonlinearity in this case is the same as the process output. A first order Fourier expansion is calculated on the output of NL. The obtained function is used to predict oscillations on the input to the non linearity.

In [8], the theory of describing function analysis is derived. The following is a short extract of the most important parts. A non frequency dependent, i.e. static, describing function has the following expression,

$$N(A) = \frac{b_1 + ia_1}{A} \tag{8.1}$$

A is the limit cycle amplitude. The constants a_1 and b_1 are the first order constants of a Fourier expansion of the output from the nonlinearity when the input is sinusoidal. The search for limit cycles is done by examination of the Nyquist-plot of the process and the describing function. The expression for a periodic solution to the closed-loop system in Figure 29 is,

$$y = G(i\omega)u = -G(i\omega)N(A)y \Leftrightarrow G(i\omega) = -\frac{1}{N(A)} \tag{8.2}$$

The intersection(s) of *G(iω)* and *-1/N(A)* in the Nyquist-diagram gives frequency, *ω*, and amplitude, *A*, for possible limit cycles. An example is shown in Figure 30.

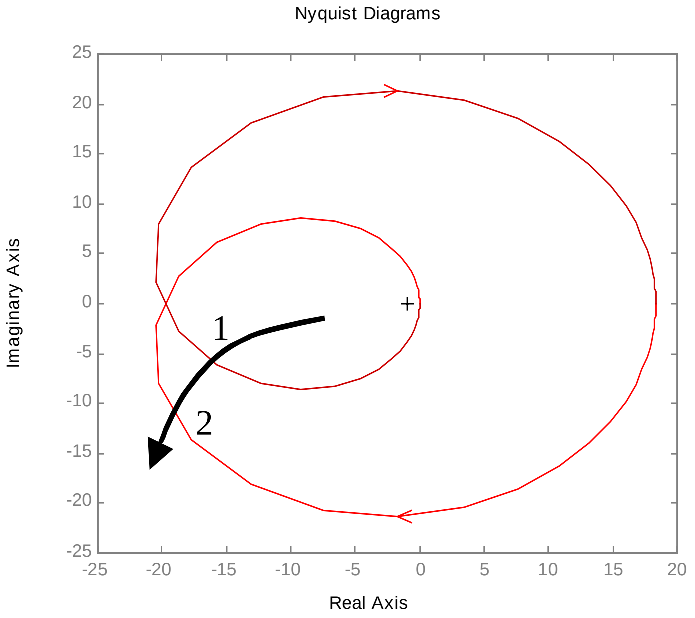


Figure 30. An example of a Nyquist plot and *–1/N(A)*. There is an unstable limit cycle at 1 and a stable at 2, according to the statements below.

The arrow on *-1/N(A)* in the plot indicates increasing amplitude. The stability of a limit cycle is predicted by use of the statements,

Assume G(s) stable. For a given $A=A_0$:
- *A* increases if the point $-1/N(A_0)$ is encircled by *G(iω)*
- *A* decreases otherwise

One must keep in mind that the predicted amplitudes and frequencies are only approximations and can be far from the true values. The analysis may also predict a limit cycle, even if it does not exist. Just as well can a limit cycle exist, even if the analysis does not predict it. The accuracy of the describing function analysis depends on how sine-like the input to the nonlinearity is. The more sine-like input signal, the more accurate analysis. Note that the input to the nonlinearity in this case is the same as the output from the process.

8.3 Hard dead-zone compensation with offset

The springs creating the dead-zone can be cancelled with a good compensator. The price paid is the introduction of an artificial nonlinearity depending on faulty estimation of the dead-zone position and time delay of the system. The effects are gathered into a single parameter, the offset. In Figure 31, the characteristic of an offset in a hard compensator is shown. The spring torque below the dead-zone position, *B_l*, is positive and the spring torque above, *B_u*, is negative.

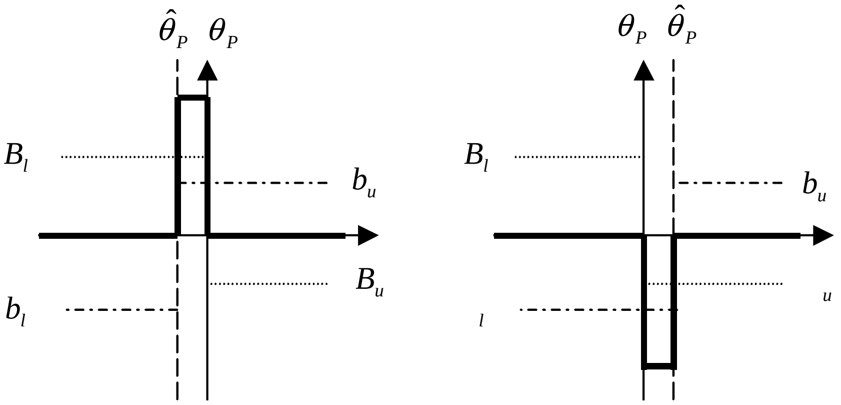


Figure 31. The characteristics of the nonlinearity created by hard dead-zone compensation with an offset. The nonlinearity consists of the sum of the spring torque and the compensation signal. Note: In the describing function analysis method the nonlinearity is fed back with a minus sign why the plots above must be inverted to fit the method.

Here B_u and B_l are the dead-zone edges, θ_P is the true dead-zone position and $\hat{\theta}_P$ is the estimate.

Figure 32 shows how the dead-zone nonlinearity affects the closed-loop system. The approximation that the electric motor constant is fast enough to be neglected is made. The closed-loop system needs to be transformed to fit the theoretical scheme described earlier. It is easily shown that the transform results in,

$$G = \frac{P}{1 + PC} \tag{8.3}$$

The problem is now expressed on the right form for describing function analysis. In the following sections describing function analysis will be applied on the identified throttle system controlled by different compensators.

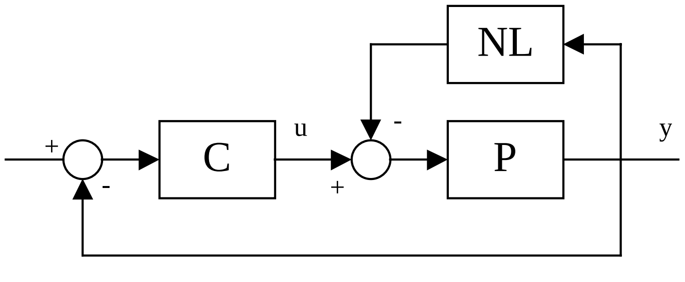


Figure 32. The effects of the nonlinearity on the throttle. If the electric motor constant is neglected, the compensation signal and the spring torque can be added to the nonlinearity as in Figure 31.

The offset is defined as $OF = \hat{\theta}_P - \theta_P$. To examine the existence of limit cycles, the describing function for the offset is derived. In the describing function method the nonlinearity is fed back with a minus sign. Therefore the resulting nonlinearity shown in Figure 31 must be inverted. The relation between a sinusoidal input and output of the nonlinearity is shown in Figure 33.

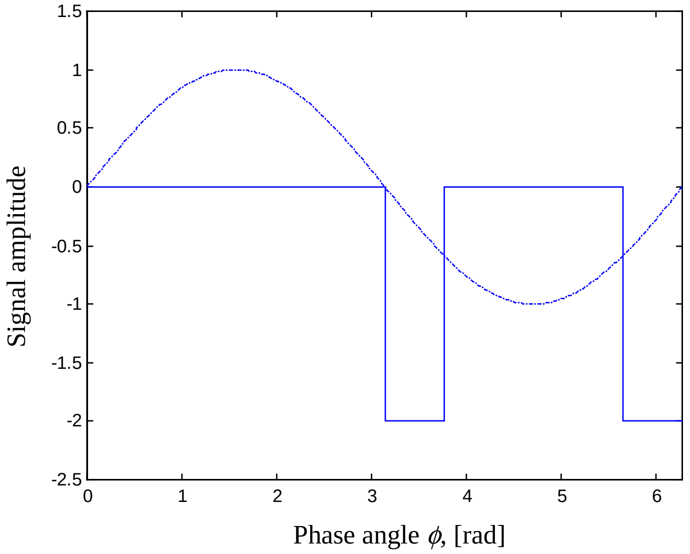


Figure 33. The sinusoidal input, dashed, and the output, solid, of the offset nonlinearity when *A*>|*OF*|. The output is *C*=(*b*$_u$-*b*$_l$) in the intervals [$0 < \phi < \phi_F$] and [$\pi - \phi_F < \phi < \pi$].

In the case of negative offset, *OF*, the resulting nonlinearity has the following expression,

$$u(\phi) = \begin{cases} C, & -\phi_F < \phi < 0, \quad \pi < \phi < \pi + \phi_F \\ 0 & , \quad otherwise \end{cases} \quad , \quad \phi_F = \left| \arcsin\left(\frac{OF}{A}\right) \right| \tag{8.4}$$

As can be seen in Figure 31 and Figure 33, the offset nonlinearity is odd. This implies that a$_{1,}$ in the Fourier expansion, is equal to 0. The nonlinear effects on the sinusoidal can be divided into two cases, *A*>|*OF*| and *A*≤|*OF*|.

A>|*OF*|

$$b_1 = \frac{1}{\pi}\int_{-\phi_F}^{0} C\sin(\phi)\,d\phi + \frac{1}{\pi}\int_{\pi}^{\pi+\phi_F} C\sin(\phi)\,d\phi = \frac{2C}{\pi}\left(1-\cos(\phi_F)\right) \tag{8.5}$$

A≤[OF]

$$b_1 = \frac{1}{\pi}\int_{\pi}^{0} C\sin(\phi)\,d\phi = \frac{2C}{\pi} \tag{8.6}$$

In the case of positive offset with good estimates of the spring torque, $C = (-B_u - b_l) \approx (b_u - b_l)$. The derivation for a positive offset is done in the same way as for negative, see Appendix B. The describing functions are real and the same for the two cases. They have the following expression,

$$\begin{cases} N(A) = \dfrac{2C}{A\pi}\left(1-\cos(\phi_F)\right) & , A > |OF| \\ N(A) = \dfrac{2C}{A\pi} & , A \le |OF| \end{cases} \tag{8.7}$$

The describing function derived above is plotted in Figure 34.

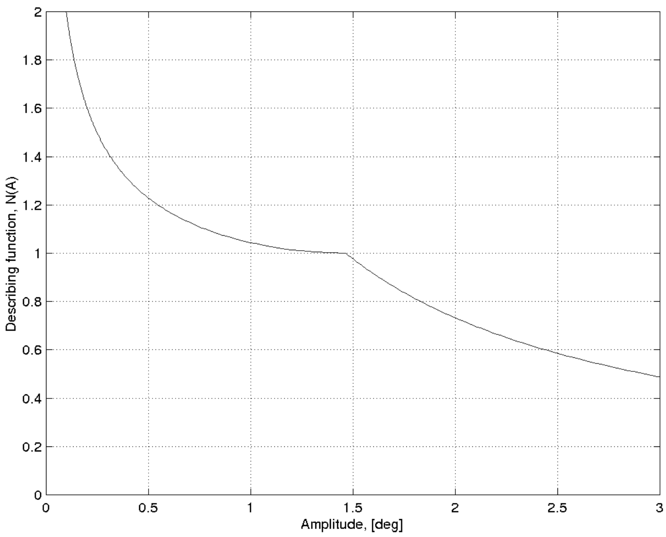


Figure 34. The describing function for hard dead-zone compensation with an offset. *N(A)* is large when the amplitude is small and 0 when the amplitude goes to infinity.

8.4 Soft dead-zone compensation with offset

Instead of the hard nonlinear compensator described in the previous section, a less discontinuous can be used. A linear interpolation between the compensation values around the dead-zone is a way of making a soft compensation. The reason for using a soft nonlinear compensator is not that it cancels the nonlinearity more exact, but the fact that it makes the nonlinearity more continuous like. A linear controller compensates more easily for a continuous than a discontinuous nonlinearity. The use of linear interpolation in the dead-zone compensation reduces the effects of an existing offset, see Figure 35.

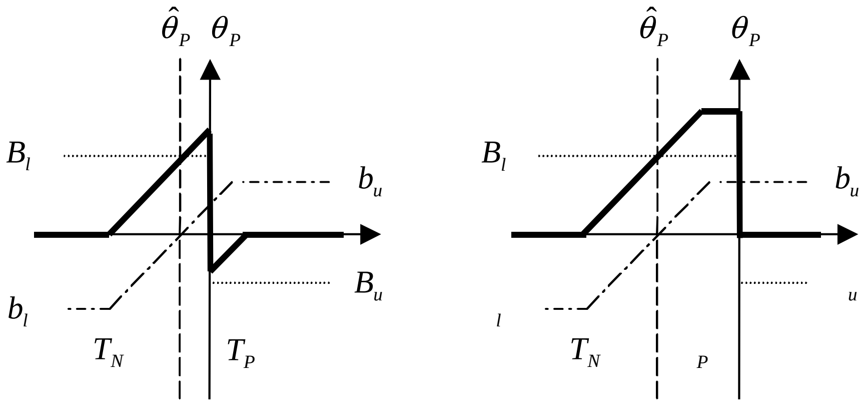


Figure 35. Resulting nonlinearity from the soft compensated dead-zone with an offset. The soft compensation decreases the magnitude of the nonlinearity but increases the interval in which the nonlinearity affects the system. Note that the compensation is only effective when the offset is smaller than the interpolation interval, [T_N, T_P].

With the same notation as in the previous section, the describing function of the soft compensated offset can be derived. For convenience, only the case where the offset is smaller than the interpolation area will be discussed. If the offset is larger the soft compensation will only make the nonlinearity wider, and therefore worse than hard compensation. It is a justified presumption that a good estimate, within a few degrees, of the dead-zone position is known.

The expressions are a bit more complex and the calculations a bit longer for soft compensation considering the simple result of the hard compensation. The characteristics are divided into two cases, positive and negative offset. In each of these cases the limit cycle amplitude, *A*, could be, compare with Figure 35.

- Smaller than both compensation edges. $A<|T_l|$ and $A<|T_u|$.
- Larger than one edge and smaller than the other. $|T_l|<A<|T_u|$ or $|T_u|<A<|T_l|$.
- Larger than both edges. $A>|T_l|$ and $A>|T_u|$.

The input to output relation with positive offset and $A>\max(|T_l|,|T_u|)$, is shown in Figure 36. Note that the resulting nonlinearity is inverted for the same reasons as for the hard compensation.

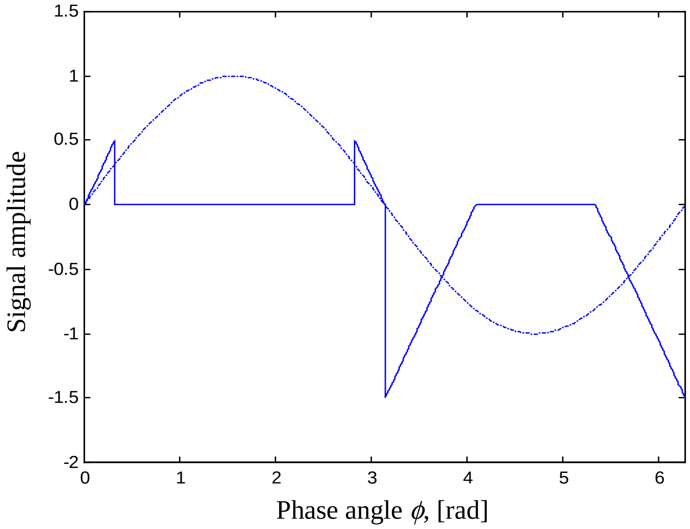


Figure 36. The sinusoidal input, dashed, and the output, solid, of the offset nonlinearity when $A>(|T_l|$ *and* $|Tu|)$. This is an example where the nonlinearity is reduced in magnitude by the soft compensation.

As can be seen in Figure 35, the nonlinearity is odd and the describing function will be a real function. Unfortunately, the describing function will consist of many long expressions. A plot of *N(A)* after tedious calculations, which can be found in Appendix B, are shown in Figure 37.

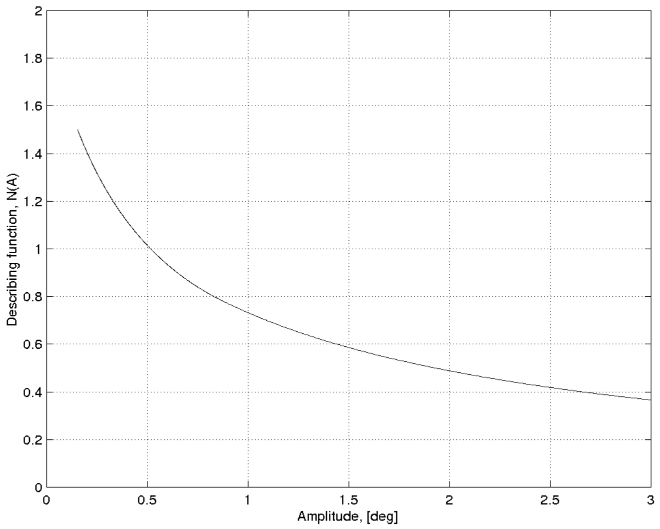


Figure 37. The describing function *N(A)* of soft dead-zone compensation with an offset. The function is real and goes from infinity to 0 with increasing *A*. This implies that *–1/N(A)* lies in the interval [-∞, 0]. Note that the curve lies below the *N(A)* curve in Figure 34.

8.5 Predictions and results

The describing function analysis gives knowledge and intuition about a nonlinear system. It predicts amplitude and frequency of possible limit cycle. It is not always that the frequency or the amplitude of the oscillations that are the most important pieces of information obtained. In the throttle case it is more important to know if there are ways to totally get rid of the oscillations than to know their exact amplitude. These types of conclusions will be discussed next.

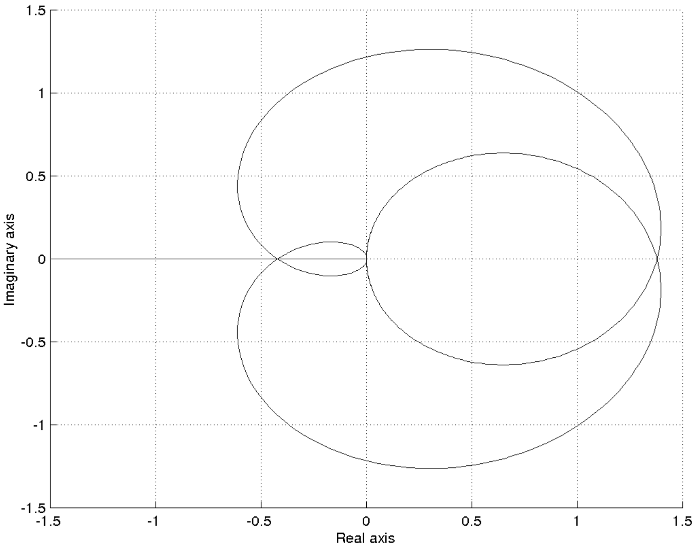


Figure 38. The Nyquist plot and *–1/N(A)* for dead-zone compensation and the open-loop controlled system. The Nyquist plot will always intersect the describing function resulting in a stable limit cycle.

The describing functions, *N(A),* for the two compensation approaches are both real and lies in the interval [-∞,0]. A glance at Figure 38 gives that the open-loop system always intersects the negative real axis. With the chosen controller structure and bandwidth, there will always be limit cycles when controlling near the dead-zone position. The way to reduce the amplitude of the oscillations is to obtain large amplitude margin for the system, which gives an intersection between the describing function and the Nyquist plot as close to 0 as possible. The soft compensation scheme only reduces the amplitude of the oscillations if the offset is smaller than the interpolation area. Simulations and real world testing confirms the theory that there will be oscillations in the system when controlling close to the dead-zone position. The analysis predicts the amplitude and frequency of the oscillations that appear during simulations correctly. Since there are different dynamics above and below the dead-zone, the results from the analysis are not directly applicable to the real throttle. The analysis shows the order of magnitude of the oscillations that can be expected though.

The only way to avoid limit cycles is to forbid the setting of reference signals close to the dead-zone. From manufacturing specifications it should be possible to find an interval in which the dead-zone position is situated for all throttles. It is very important to get rid of any oscillations in the system. The tear on the plate axis and the electric motor on the throttle would be very large during the life length of the car if oscillations would be permitted.

9 Adaptive parameter estimation

9.1 About adaptive estimation

The characteristics of a system will not be the same for all time. External factors will affect parts of the system and change its response to control signals. For example, wear and tear will change the characteristics of the spring package on the throttle. The strength of the springs will decrease during the life length of the car. Another external disturbance is the temperature of the motor driving the throttle. The electrical resistance of the motor will increase with higher temperatures giving a varying gain of the system. Considering that the temperature difference between a cold and a hot engine can be as large as $-40°C$ to $120°C = \Delta 160°C$, the impact on the gain and the time constants of the spring package will be important for control performance. The difference in characteristics between throttle units due to the manufacturing process will also create difficulties in the control law.

For high performance control with changing parameters, the problem is complex. In this case the characteristics of the system change during operation. There are two solutions to this problem, a robust controller or online estimation of important parameters. A combination of the two solutions above is probably the most successful approach to the problem.

It has been shown earlier that the three most important parameters of the compensator, are the offset and the dead-zone edges. If these parameters are not known exactly at every time instant, the result of compensation is a new nonlinearity. Since the control performance is largely dependent on these parameters it makes sense to estimate them during operation. This way the compensator will only make small errors. In the following sections different ways to estimate the spring torque will be examined.

9.2 Least-squares

The least-squares method is an intuitive technique for parameter estimation. The principle was formulated by K.F. Gauss and used to determine the orbits of planets and asteroids. The key idea is to minimize the sum of the squares of the errors between measured values and calculated values. A thorough investigation of the theory is done in [7]. The results are briefly discussed here. The following results are based on the assumption that the system to be estimated can be expressed linearly in its parameters,

$$y(t) = \varphi_1(t)\theta_1{}^0 + \varphi_2(t)\theta_2{}^0 + \ldots + \varphi_n(t)\theta_n{}^0 = \Phi(t)\theta^0(t) \tag{9.1}$$

The variable, *Φ*, are called the regressor and *θ* are the parameters to be estimated. The parameter estimate from signals up to time t is

$$\hat{\theta}(t) = \left(\Phi^T\Phi\right)^{-1}\Phi^T Y \tag{9.2}$$

Here *Y(t)* is a vector of the measured values up to time *t*. To speed up online estimation calculations the result above can be formulated on recursive form, i.e. a new estimate is calculated at every time instant, *t*, with information gathered up to time *t-1*. In this way there is no need to consider signals of all time instants in the calculations. The modified algorithm is, in its simplest form, as follows,

$$\begin{aligned}\hat{\theta}(t) &= \hat{\theta}(t-1) + K(t)\left(y(t) - \varphi^T(t)\hat{\theta}(t-1)\right)\\ K(t) &= P(t-1)\varphi^T\left(\lambda I + \varphi^T(t)P(t-1)\varphi^T(t)\right)^{-1}\\ P(t) &= \left(I - K(t)\varphi^T(t)\right)P(t-1)/\lambda\end{aligned} \tag{9.3}$$

Here λ is the forgetting factor, which indicates how fast the update law should be. As an example, the dead-zone compensation problem is discussed. When compensating for the dead-zone, the resulting input signal to the system is,

$$u = u_d + \hat{\theta} + \theta \tag{9.4}$$

Where $\hat{\theta}$ is the spring torque estimate and θ the true torque. The control signal from the linear controller is u_d and the actual input signal to the system is u. In perfect compensation, $\hat{\theta}$ and θ cancel each other and $u = u_d$, if not there is an error in the compensation. The relation between the input signal u_d and the output y in Figure 39 is,

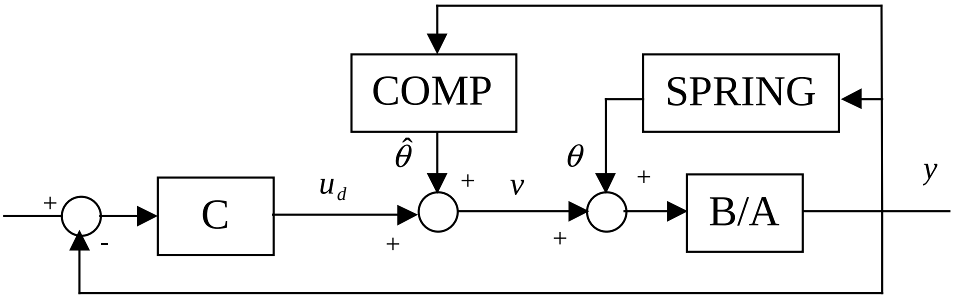


Figure 39. Control signal u_d, compensated control signal, v, and the spring torque, θ, affecting the system.

$$y = \frac{B}{A}u = \frac{B}{A}\left(u_d + \hat{\theta} + \theta\right) \tag{9.5}$$

The equation above can be reformulated to fit the least squares method,

$$y_{tot} = B\theta = Ay - B\left(u_d + \hat{\theta}\right) \tag{9.6}$$

The B polynomial becomes the regressor, φ, and the estimated parameters, θ, at time instants according to the time shifts of the polynomial. With these matrices an estimate of the dead-zone edge can be calculated with the recursive least squares method.

The drawback with this method is the absence of offset estimation. There is no obvious way to derive regressor matrices for the dead-zone position. It has been shown earlier that this is the most important parameter, why the least-squares method will not be sufficient as adaptation law for the throttle.

9.3 Stability theory

There are many types of methods based on the minimization of loss functions. These are divided into gradient methods, e.g. the least-squares method, and stability theory methods. The most famous of the latter methods is the Lyaponov stability theory, [7, 8].

The key idea in stability theory is to find a positive definite loss function, *V(x),* with time derivative negative or zero. This shows that the loss function, which can be interpreted as the energy of the system in some sense, always will decrease. A function has to fulfil four requirements to be a Lyaponov candidate, *V*. The definitions are taken from [8].

1 $V(X) > 0, X \neq X_0$	; Loss function always > 0 except in one point
2 $V(X_0) = 0$	; The loss function is only 0 in one point, X_0
3 $V(X) \to \infty, X \to \infty$	; Gives global stability
4 $\dot{V}(X) < 0, X \neq X_0$	; The time derivative always negative

With a cleverly chosen loss function a parameter update law can be found. The error, *e*, when compensating for the dead-zone can be described linearly in the estimates as:

$$e = y - y_m = \frac{B}{A}(u + \theta) - \frac{B}{A}\left(u_d + \hat{\theta}\right) = \frac{B}{A} d\theta \tag{9.7}$$

Here $d\theta$ is the difference between estimated and true parameters. A simple loss function is chosen,

$$V = \frac{e^2}{2} \tag{9.8}$$

The first three statements are fulfilled and,

$$\dot{V} = e \cdot \dot{e} \approx e \cdot k \cdot \dot{\theta} \;\; ; \quad \dot{e} \approx \frac{B(0)}{A(0)} \cdot \dot{\theta} \tag{9.9}$$

The approximation that the change in θ will be low frequent is used, hence the expression for the time derivative of *e*. With this intuitive parameter update law below, the forth requirement is fulfilled.

$$\dot{\theta} = -\gamma \cdot e \quad \Rightarrow \quad \dot{V} = -k \cdot \gamma \cdot e^2 \tag{9.10}$$

A better but probably more complex update law could be found with a different choice of Lyaponov candidate and more strict approximations. The parameter γ should be chosen to make the adaptation slow enough so that the parameters will not be affected by the integral action of the controller. The Lyaponov theory approach is appealing because it is easy to show stability of the parameter estimates.

Just as for the recursive least squares method, the Lyaponov-method fails in estimating the dead-zone position. Neither of these methods described so far will be sufficient as adaptation laws for the throttle.

9.4 State Observer

As an alternative to the polynomial based methods above, a state-space approach to estimate the compensation error in the system will be investigated. In [5] it is described how the internal states of a given system can be estimated with a state observer. A method to add integral action to a linear state-space feedback compensator is derived in [3]. An extra state, driven by a constant disturbance, is added to the state equations. The new state is then fed back in the same way as the other states. A somewhat modified such method can be used in online estimation of parameters. Figure 40, shows the throttle system with a disturbance added at the input.

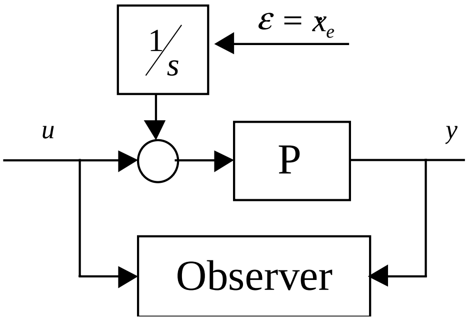


Figure 40. Block diagram of the observer and the extra state on the input to the system. The disturbance is modelled as being driven by white noise. The disturbance is then the torque asserted by the springs.

The original system process, P, is described on state-space form, i.e.

$$\begin{cases} x_{k+1} = \phi x_k + \Gamma u_k \\ y_k = C x_k \end{cases} \tag{9.11}$$

Since the torque of the springs only changes slowly, the extra state of the observer should track a constant disturbance. In [3] the derivation of the extended matrices is performed. The extended state observer gets the following expression,

$$\begin{bmatrix} \hat{x}_{k+1} \\ \hat{x}_{e\,k+1} \end{bmatrix} = \begin{bmatrix} \phi & \Gamma \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{x}_k \\ \hat{x}_{e\,k} \end{bmatrix} + \begin{bmatrix} \Gamma \\ 0 \end{bmatrix} u_k + K\left(y_k - \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_k \\ \hat{x}_{e\,k} \end{bmatrix} \right) \tag{9.12}$$

The extended matrices are referred to as ϕ_e, Γ_e and C_e.

The design of a state observer is done by placement of the eigenvalues, λ, in the matrix below,

$$eig(\phi_e - KC_e) = \det(\lambda I - (\phi_e - KC_e)) \tag{9.13}$$

The design can easily be done in Matlab with the commands `place` or `lqed`. The `place` command calculates the *K*-matrix to place the eigenvalues at given positions. This is a simple and intuitive design procedure, but it has drawbacks. The designed eigenvalues have to be chosen ad hoc, and it can be difficult to make the observer both noise insensitive and fast. The design problem also gets worse with increasing number of states in the systems state representation. There are as many design parameters as there are states in the representation.

An alternative design procedure, based on weight matrices, is used in the `lqed` command. The command produces an optimal Kalman observer by solving a Riccati equation, see Theorem 11.5 in [3]. The design is the equivalent of the LQG design described in chapter 6. The Kalman predictor is optimal in the sense that the variance of the reconstruction error is minimized. This is an easier to use design procedure since it is more intuitive to give estimates of disturbances than it is to choose eigenvalues.

9.5 Parameter update laws for the state observer

There are two different measurable signals that can be sent as input to the estimator. The first is the control signal from the controller, u_d . The second available signal is the compensated control signal, v, where $v = u_d + \hat{\theta}$ compare with Figure 41.

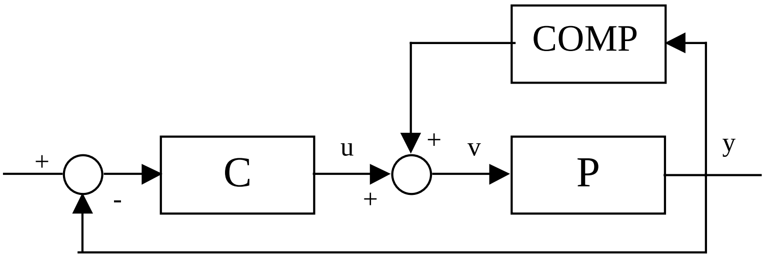


Figure 41. The closed-loop system with the controller, C, the dead-zone compensator, COMP and the process, P including the springs.

The use of *v* as input signal to the observer results in an estimate of the compensation error, i.e. the difference between the estimated spring torque $\hat{\theta}$ and the true torque θ. When the compensation is perfect, the error is 0, see Figure 42. , The offset is defined as the true dead zone position minus the estimate, $offset = \theta - \hat{\theta}$.

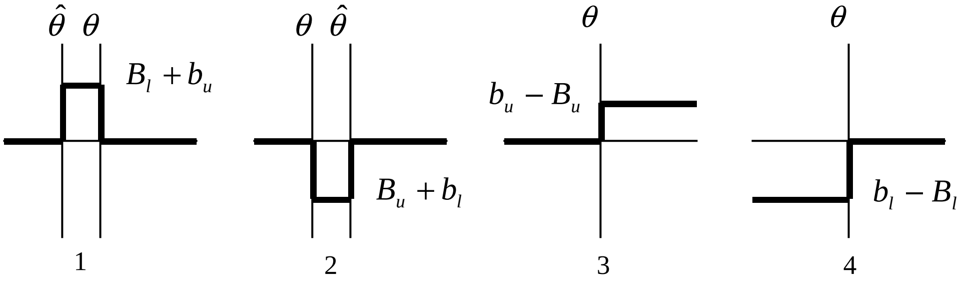


Figure 42. Examples of compensation errors, corresponding to the observed extra state. Cases 1 and 2 are the types of signals the observer would track with input signal *u*, and cases 3 and 4 if the observer is fed with the signal *v*.

The work range of the estimator is divided into three areas, above, around and below the estimated dead-zone position. Above and below, the dead-zone edges b_u and b_l are estimated, whilst the offset estimate is updated around the dead-zone position. The parameter update laws are simple and intuitive,

$$\begin{cases} b_u = b_u - kx_e \\ b_l = b_l - kx_e \\ offset = offset + kx_e \end{cases} \tag{9.14}$$

Here k is a constant that determines in how large steps the estimated parameters can change from one sample to another. If the observed compensation error is positive, the estimated torque, *b*, is too large and therefore decreased. The offset is increased if the compensation error is positive, compare with Figure 42.

These are very simple and reasonable update laws, but there are some drawbacks with this approach to the estimation. Since the signals to be estimated are either small or appear during short time intervals, the observer needs to be fast and noise insensitive. This is as always a hard condition to fulfil. If the observer is fast and the signal to noise ratio is small the useful information will drown in the noise and the spring torque adaptation will be slow and unreliable. On the other hand if the observer is slow and noise insensitive, it will not be able to track the short pulses of information when crossing through the dead-zone. This indicates an unsatisfactory estimate of the dead-zone position.

Another approach to the problem is to use the compensated control signal as input to the observer. In this case the estimator will track only the spring torque acting on the system in every time instant, see Figure 43.

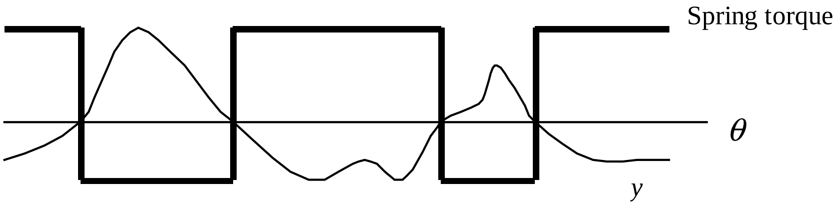


Figure 43. The spring torque affecting the system as a function of throttle angle and time. The observed disturbance is ideally the same as in the figure.

These signals are always large in magnitude and have the same characteristics at all time. The dead-zone edge estimation is similar to the one in the previous case. The big difference is in the offset adaptation. The offset will now be estimated as the throttle position when the extra state of the observer changes signs. The time delay of the system and the speed of the observer will of course affect the offset estimates. These effects can be neglected by calculating the mean value of the n latest offset estimations, half of them while moving up through the dead-zone and the other half while moving down. The approximation that the time delay is the same in the two cases is reasonable.

$$\begin{cases} b_l = b_l + k(b_l - x_e) \\ b_u = b_u + k(b_u - x_e) \\ dzpos = \frac{1}{n}\sum_{1}^{n}\left(y_{k_i}\right) ; \quad k_i = \{sign(x_e(k)) \neq sign(x_e(k-1))\} \end{cases} \tag{9.15}$$

The approximation above is only justified if the angular velocity of the throttle is constant while moving through the dead-zone. The estimation error on account of the approximation can be minimized by calculating the time delay, t_d, of the system. The alternative offset update law would then be,

$$dzpos = \frac{1}{n}\sum_{1}^{n}\left(y_{k_i - t_d}\right) \; ; \; k_i = \{sign(x_e(k)) \neq sign(x_e(k-1))\} \tag{9.16}$$

The state observer estimates the required parameters for successful control of the throttle. The simplicity of the method also makes this approach the most appealing of the three adaptive schemes investigated. In the following sections issues on simulation and "real world" application will be discussed.

Another observation must be made. Since there are different dynamics above and below the dead-zone, two observers are used. The two observers need to be equally fast in the disturbance observations to result in an accurate estimate of the dead-zone position.

9.6 Simulations

To evaluate the estimator derived in earlier sections a simulation model was built in Simulink. The process model was taken from the results of the identification chapter. All three estimators show good results in simulations. The RLS and Lyaponov methods estimate the spring torque only, while the Kalman observer also estimates the dead-zone position. As an example, simulations with the first two methods are shown in the figure below.

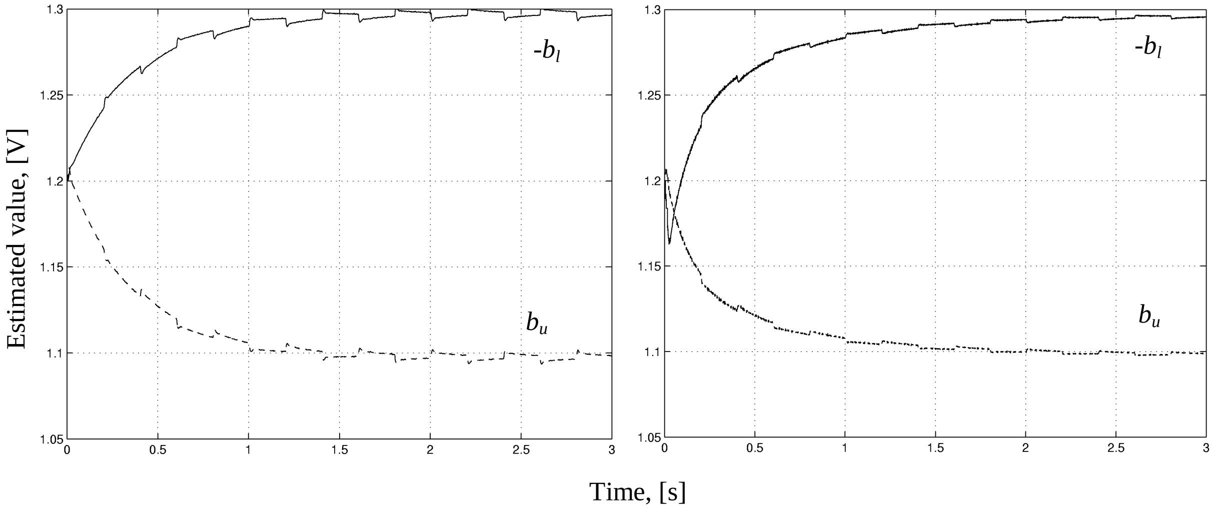


Figure 44. Convergence of the dead-zone parameters b_u=1.1 and $-b_l$=1.3, for the Lyaponov method to the left and the recursive least squares method to the right. The reference signal used was a square wave with a frequency of 2.5 Hz. Each plot is the result of two simulations, above and below the dead-zone.

The parameter estimates of the RLS and Lyaponov method converges to their true values, see Figure 44. The excitation signal used to produce the plots was a square wave with a frequency of 1 Hz. Each of the plots are the result of two separate simulations. The convergence times in the examples are about 3 s. The convergence rate is a design parameter in both methods. In the RLS this is controlled by forgetting factor, λ, and in the Lyaponov method there is the gain, γ, in the parameter update law. By tuning these parameters faster convergence can be realised. There is of course as always, a drawback to be considered. The faster the convergence of the estimator the more sensitive it is to noise and other disturbances. Since the changes in the characteristics of the throttle are very slow, it is not necessary to have a fast estimator and therefore the noise should not be a problem.

After the Kalman observer had been implemented in the simulator it could be seen that both the spring forces and the dead-zone position estimates are converging fast. During the design of the observer it is possible to vary the weighting matrices to get a good estimation of all parameters within a reasonable time.

9.7 Performance and results

The Kalman observer was implemented and used on the real process. After some trouble dealing with the different characteristics above and below the dead-zone, the estimator worked fine. The design was performed with weight matrices, using orthogonality to move individual states in the observer. The values of the weight matrices and the process model matrices can be found in Appendix A. The design resulted in the following observer matrices, K_l and K_u.

$$K_l = \begin{bmatrix} -11.7182 \\ 46.5102 \\ 35.0284 \\ 10.7837 \end{bmatrix}, \quad K_u = \begin{bmatrix} -11.4934 \\ 42.4049 \\ 32.8217 \\ 8.4208 \end{bmatrix} \tag{9.17}$$

The results of the simulation were reproduced on the real throttle. The speed constants in the adaptation laws where chosen so that the time needed for all estimated parameters to converge was about 15 to 20 s with a reference signal around the dead-zone with a frequency of 1 Hz.

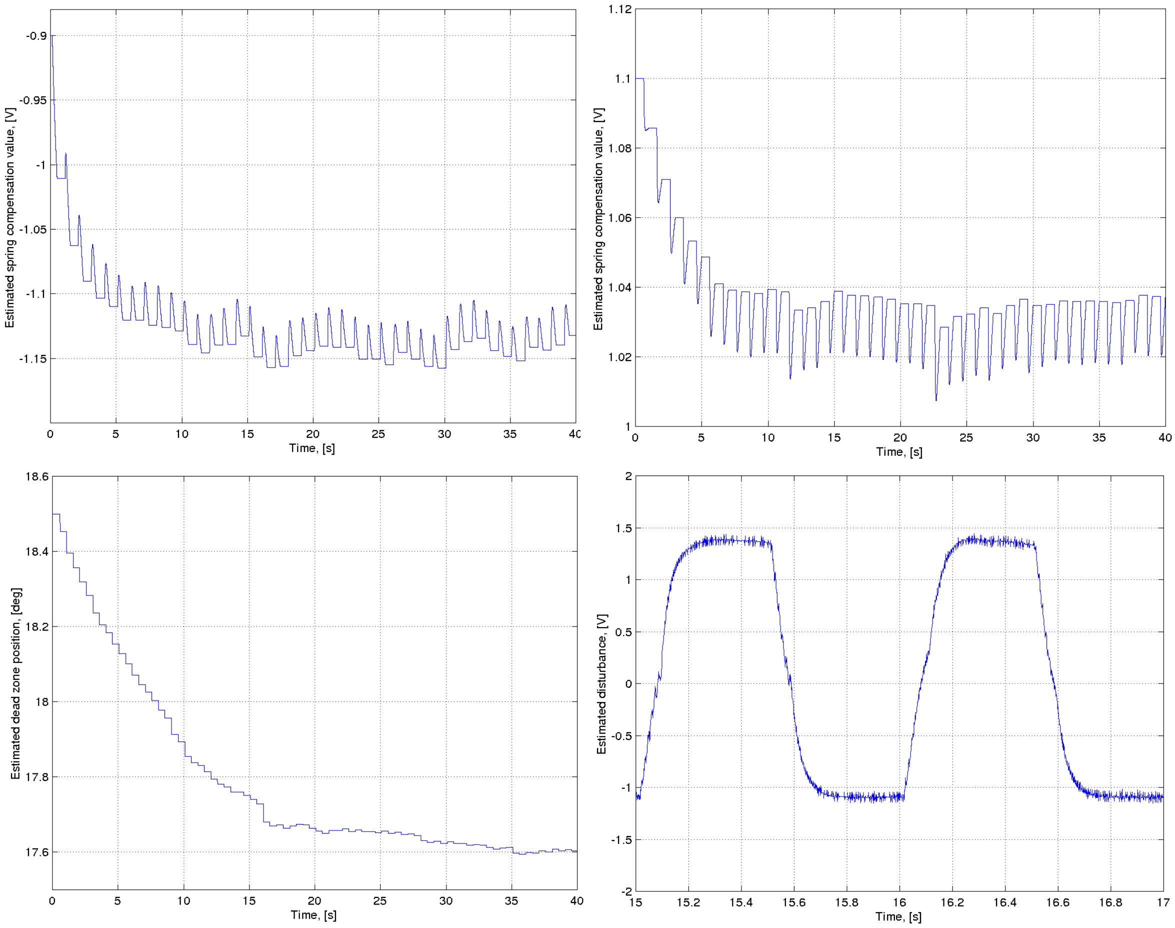


Figure 45. Estimated parameters with Kalman observer. Top left is b_l, top right is b_u, bottom left is dead-zone position and bottom right is the estimated disturbance during a step through the dead-zone. The peaks in the estimations on the top row occur because the estimated disturbance is not a true square wave, bottom right plot.

The parameters converge to some value, but are the values the true ones? That question is not easy to answer. The estimates are close to the values that have been calculated and obtained through e.g. identification and start-up experiments. It was also shown in the simulations that the estimates converged to the true values. It could be assumed the real process estimation is correct. It was discovered that the estimated values, especially for the spring torque, were noise sensitive, but this did not affect the estimation in a crucial manner. When choosing the weighting matrices, the designer must make sure that the spring force estimation is slow enough not to affect the process states in the observer. If the estimate is too fast it will “steal” information about the process from the states that are used in the linear controller. On the other hand, if the extra state, corresponding to the force estimation is updated too slowly there will be problems in the adaptive update laws.

10 Implementation issues and fixes

10.1 Simulation and control environment

The control environment has throughout this master thesis been Matlab and Simulink. The real-time communication between the Simulink model and the throttle was handled by A/D-D/A blocks, so called “Blomdelleri” blocks. These blocks have been developed by Anders Blomdell at the Department of Automatic Control at Lund Institute of Technology.

The real-time environment was running on a 450 MHz Pentium III PC with the operating system, Linux, installed. The control loop has a sample period of 1 ms. Fast sampling requires control algorithms with short computation times. If the processor can not do all calculations in one sample there will be jitter, i.e. lag in the control loop. The use of S-function blocks is a way of speeding up calculations in Simulink. These blocks can be written in different programming languages, e.g. Matlab code, C, Fortran.

As a first attempt, S-functions written in Matlab code were used. The computation capacity of the processor then only allowed a short S-function to be present in the model. If more computation steps were added, the jitter in the process was too big. Apparently the Matlab code is compiled at runtime and not before the model is started. This makes even the simplest block very slow.

The S-function blocks can be speeded up by implementing them in C code. The functions are now compiled separately and then used in the Simulink model. It has been found that S-functions compiled before the model is started are up to **20 times** faster than functions implemented in Matlab code. This makes it possible to sample faster and still have a good amount of calculations done in every sample. It is, for example, possible to run the control model of throttle with a sample time of 0.5 ms.

Directions on how to build S-functions in C code can be found in the full documentation to Matlab.

10.2 Compensation method

Hard feedback compensation of the dead-zone has been used throughout the work on the master thesis. The aim has been to use the same specifications for the closed loop behaviour through the dead-zone as in the linear areas, hence a soft compensation is discarded.

The use of the same state observer for both linear control and adaptive updating creates some difficulties. The adaptation mechanism needs the compensated control signal as input to the observer. This is necessary to be able to estimate the dead-zone position. Unfortunately this has bad effects on the linear controller performance when passing through the dead-zone. Since the input signal changes its value abruptly, the internal oscillating modes in the observer are excited. This results in an oscillating control signal and a decrease in performance when moving through the dead-zone.

There are several ways to get rid of such unnecessary oscillations. One is to use separate observers for the linear controller and the adaptation, and feed the controller observer with the previous calculated linear control signal only. Another is to soften the compensation during a few samples. The first solution results in more calculations to be performed each sample. The second way to reduce the oscillations slows the compensation and thus also decreases the performance through the dead-zone.

The compensator used in this master has been softened, but the use of separate observers could just as well be used.

10.3 Overlapping springs

During the work on controlling the throttle, it has been noticed that the characteristic of the spring package is more complex than was modelled in earlier sections. The fact that the dead-zone is operative in an angular interval instead of only at one specific angle has not been modelled. Figure 46 shows the angular position during a step change through the dead-zone. At first the measured value follows the reference quite well. Then, as it is closing in to the dead-zone, there appears to be an opposite directed force affecting the throttle and the measured value divers from the reference. When this extra force finally is overcome, and the throttle passes through the dead-zone, the force disappears. This results in a large overshoot.

A model that will explain this behaviour is to assume that there is an overlap in the work ranges for the springs. If the work ranges overlap, there will exist a region were both springs affect the system and the dead-zone will change its characteristics, see Figure 46.

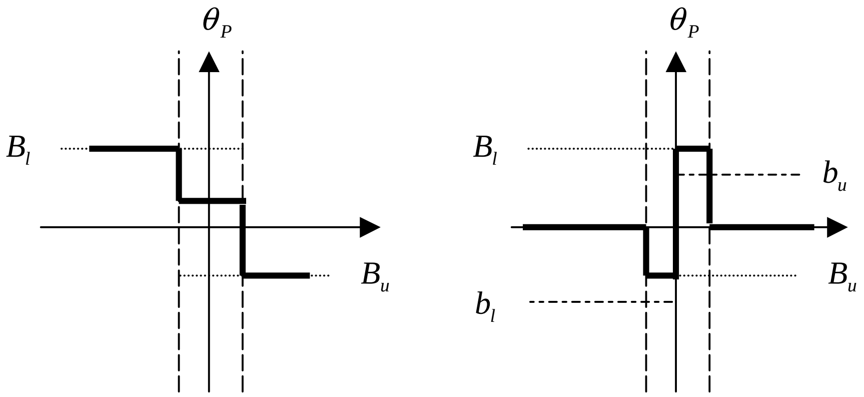


Figure 46. To the left are the characteristics of overlapping springs. To the right is the resulting nonlinearity after hard compensation for the offset. The resulting nonlinearity predicts the behaviour in Figure 47.

As can be seen in the figure above, the new characteristics of the dead-zone will affect the performance of the throttle when passing the dead-zone. The result of hard compensation is a nonlinearity with a large magnitude and affecting the system at a sensitive time for the control performance.

According to [10] there is no specification from Magneti Marelli confirming that the overlap would be deliberately added in the spring package. It is most likely to depend on manufacturing precision. Since the overlap is not a designed part of the throttle it can not be included in the dead-zone model.

There could of course be other reasons for the described behaviour of the springs. However, the symptom that the springs loose torque amplitude close to the dead-zone remains. The overlap is then an easy way of modeling a complex behaviour. Since these characteristics cannot be modelled, a fix for the problem is suggested.

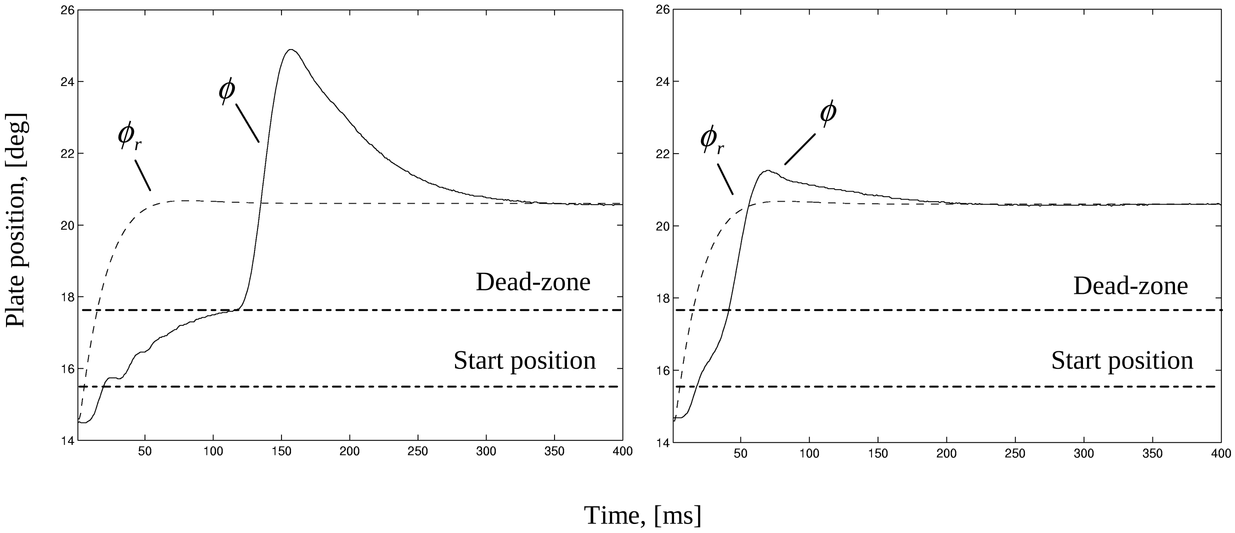


Figure 47. A step through the dead-zone, (-.-), without the fix to the left and with the fix to the right. The nonlinearity in Figure 46 acts as a retarding force before the dead-zone is passed and as an extra push afterwards. The fix reduces the effects of the problem. The fix is active in the region between the start position and the dead-zone position. In this interval the derivative of the plate position is tracked. If the acceleration changes signs, i.e. the angular velocity decreases, the direction of the dead-zone compensation is changed.

To get rid of the time consuming behaviour in Figure 47 one can track the acceleration, $\ddot{\phi}$, of the plate position. The fix then works as follows,

During a step up through the dead-zone:

- If $\ddot{\phi} < 0$ or $\phi > \hat{\phi}_0$ then the compensation changes signs.

During a step down through the dead-zone:

- If $\ddot{\phi} > 0$ or $\phi < \hat{\phi}_0$ then the compensation changes signs.

The result is a reduction of the retarding torque and an increase of the accelerating torque, compare with Figure 46. The fix decreases the rise time but increases the overshoot. Since the reference value during operation in a car engine is changed very often, it is more important to have a short rise time than a small overshoot. The acceleration of the plate is calculated simply by differentiating the plate position twice. This is not a very good method in noisy systems. A better way to determine the acceleration would be preferable.

In the control law used today in the Volvo cars the dead-zone compensation is implemented as feedforward compensation. This means that the compensation changes signs when the reference value passes the dead-zone position estimate. The two methods are very much alike. Both changes the direction of the compensation earlier than it should, if considering the dead-zone position estimate only.

11 Results

After the design of the two robust linear controllers, one on each side of the dead-zone, and the adaptive Kalman filters, the implementation was made in the real-time environment Matlab/Simulink. The model was extended with the blocks used in the adaptive respectively linear controller design. Switching between the two different controllers was performed using gain scheduling. The change between controllers is then done at the estimated dead-zone position. To avoid transients during the switching phase, a tracking signal was introduced, this to ensure bumpless transfer and to avoid integrator windup. The aim in this chapter is to test and evaluate the derived controller strategy.

First the controller was tested without adaptation of the parameters, or compensation for, the dead-zone. As expected the step response was slow and it did not meet the closed loop specifications from [10], see Figure 48. The integrator part of the controller compensates for the springs, but not fast enough to meet the specifications.

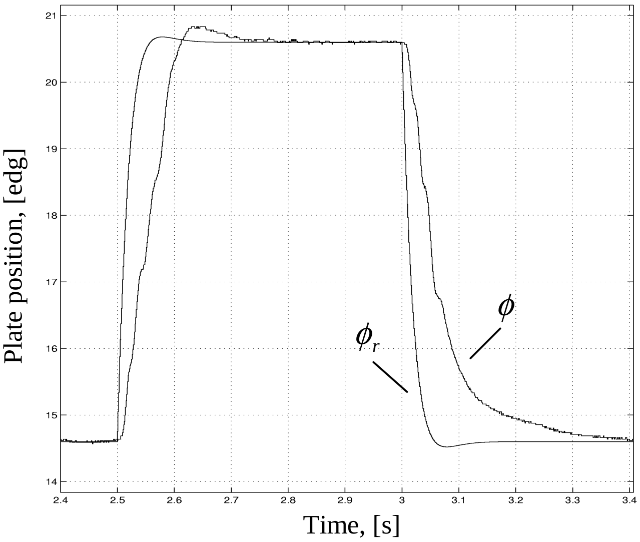


Figure 48. Two 6° steps through the dead-zone using the implemented controller without dead-zone compensation. The plate moves slowly towards the dead-zone and passes through it when the integrator part of the controller has overcome the spring torque. The result is a slow behaviour through the dead-zone.

Next the adaptation and the dead-zone compensation is switched on. In the beginning, the step responses through the dead-zone are bad due to compensation errors resulting from inaccurate estimations of the parameters. After some steps back and forth through the dead-zone, the controller has estimated all the necessary information about the dead-zone, the position and the spring torque. When the estimated parameters has reached stationarity, about 40 steps with bad initial values and slow updating, the steps through the dead-zone is fast and almost as if there was no dead-zone.

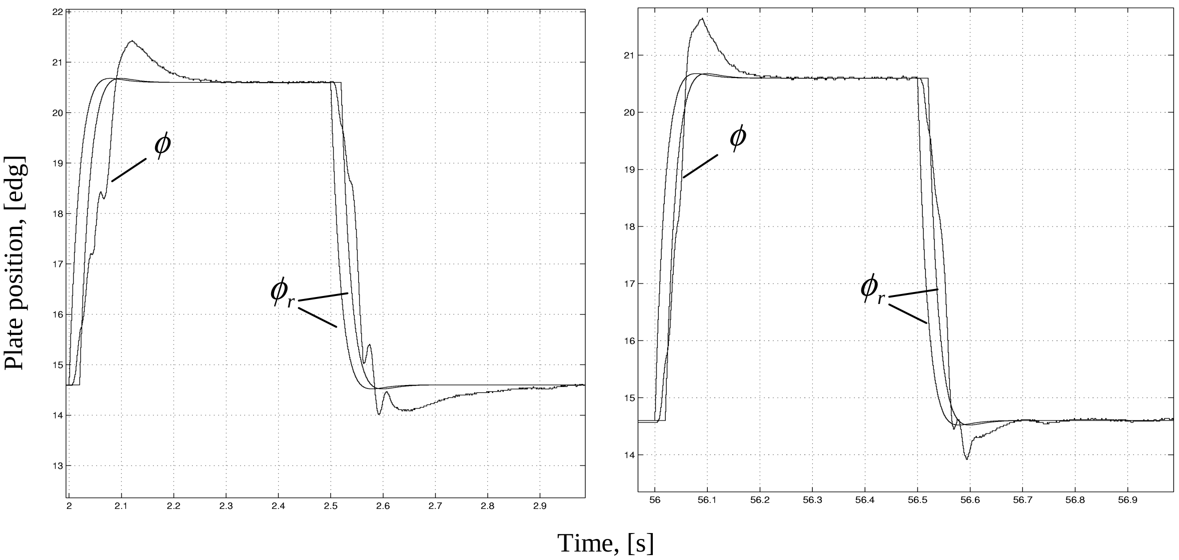


Figure 49. The left figure shows a 6° step through the dead-zone when the adaptation has just started. The initial values are badly chosen, (b_u=1.5, b_l=-1.5 and *dzpos*=18.5), and are updated slowly. After a while the estimated parameters have become stationary, (b_u=1.03, b_l=-1.06 and *dzpos*=17.7), see the plot to the right. The compensation improves the rise time, but also creates overshoots.

The specifications are still not met however, especially for small steps through the dead-zone. The problem of the remaining unmodelled characteristics of the dead-zone results in unnecessary slow performance. These remaining characteristics could, as been shown before, be seen as an overlap between the springs creating the dead-zone. Since there is nothing about a spring overlap in the data sheet of the throttle, the feature can not be modelled into the compensation. Instead, a fix for the problem is suggested, which basically monitors the acceleration of the plate and compensates for the retarding force of the overlap nonlinearity. The fix functions in a similar way as a hard dead-zone compensation acting according to the reference signal instead of the measurement signal. A small step through the dead-zone with the fix applied is shown in Figure 50.

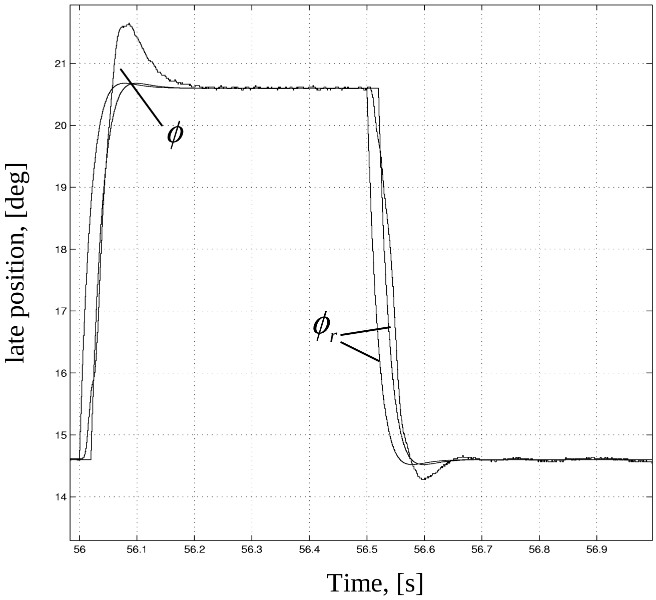


Figure 50. A 6° step through the dead-zone with the fix. The fix is active around the dead-zone position, *dzpos* ± 2°. The nonlinearity in Figure 46 acts as a retarding force when the dead-zone is passed and an extra push afterwards. The fix reduces the negative effects of the springs loosing torque close to the dead-zone.

Now all of the closed loop specifications will be meet except the overshoot for small steps around the dead-zone. A large step has time to build up momentum in the plate and motor before crossing the dead-zone. The nonlinearities of the dead-zone are then hardly noticeable. During a small step, little momentum builds up before the plate crosses the dead-zone, and the plate becomes more sensitive to the nonlinearities in the dead-zone. The step responses in the figures above indicate that there are still unmodelled nonlinearities around the dead-zone. The remaining nonlinearities are probably due to changing spring dynamics and friction. It can be hard to see the improvement on performance in the plots above. Therefore different specification parameter values are listed in the table below.

	Side	Delay Time (T_d), ms	Step Time (T_s), ms	Over shoot (M), %	Steady State ms
Without adaptation	Above	15	76	5	195
	Below	16	125	-	300
With adaptation	Above	16	42	18	175
	Below	16	47	12	170
With adaptation and fix	Above	16	36	16	170
	Below	16	44	5,5	135

Table 8. A summary of the results using the linear controllers only, the controllers together with adaptation and finally, the controllers with adaptation and a fix. The best closed-loop performance was achieved using adaptation of the dead-zone with a fix together with well tuned linear controllers.

When approaching the dead-zone area and trying to control the plate position close to the dead-zone, the result was sometimes oscillations in the plate position. A limit cycle can be excited when a large step is applied towards the dead-zone area and the final desired position is within the dead-zone or about 1.5 ° from it. An example can be seen in Figure 51 below. The amplitude is constant at 0.2 ° with a frequency of 380 rad/s.

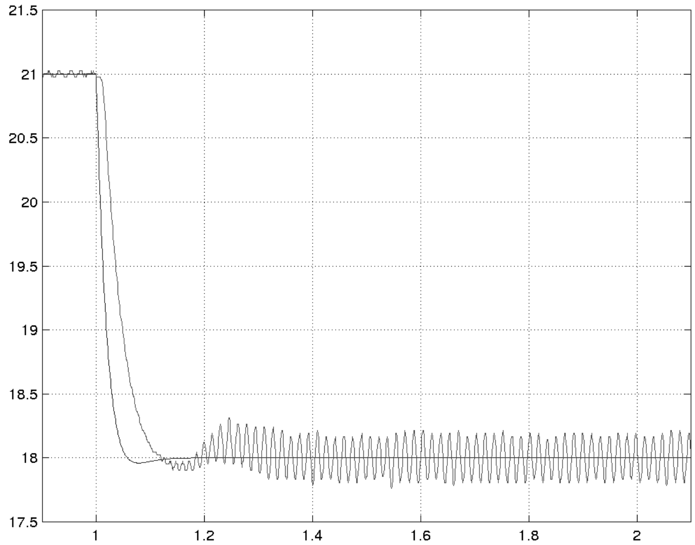


Figure 51. A limit cycle generated just above the dead-zone. Changing the plate position from 21° to 18° with an error of 0.5° between the true dead-zone position and the estimate. The limit cycle has an amplitude of 0.2° and a frequency of 380 rad/s. The true dead-zone position is approximately 17.5°.

The simplest way to avoid limit cycles is to avoid controlling within a safety area around the dead-zone, approximately 2°. This is not acceptable from the closed-loop performance point of view. A better way is to make the plate approach to the dead-zone a little smoother with first a large step close to the critical area and then a final smaller step. The controller could also be designed to be slower, or having a lower gain. The obvious way is to apply a perfect dead-zone compensation at every sample. This is not realistic though since there is a time delay in the system and the parameters used are only estimates of the true values.

The described methods will not only reduce the risk of excitation of a limit cycle, but also reduce the amplitude and frequency if a limit cycle do appear.

12 Conclusions

After finishing this master thesis there are some main conclusions that can be drawn.

- It is possible to implement fast real-time control loops in the Simulink environment.

The control loop of throttle was closed using Matlab/Simulink running on a PC with Linux as operative system. A graphical user interface was easily built using the different blocks in Simulink together with blocks developed at the Dept. of Automatic Control in Lund, simply called "Blomdelleri". In the beginning there were no problems to meet the real-time demands on the system using a sample rate of 1 ms. During the work on thesis the model complexity grew, especially when the adaptive estimators and compensators were added, and the jitter became to large. To solve the problem all calculations had to be written in C-code and compiled using a normal compiler for C-code. After this it was possible to run the control law with a minimum sample time at 0.25 ms.

- More than one linear controller is required for accurate control of the throttle.

During the identification process it was discovered that there were different dynamics above and below the dead-zone. The characteristics also changes with angular position within each area. This implies that at least two different controllers should be used. This presumption was later confirmed during the linear controller design. To meet the closed loop specification in every single operating point, gain scheduling with at least two linear controllers must be used.

- LQG is a robust and easy to use linear controller design method.

The linear controllers were designed based on the models achieved from the system identification. First an attempted was made using pole placement on polynomial form, the RST design method. Different poles from a fast observer polynomial and a slower controller polynomial were placed so that the closed loop system would meet the demands. This turned out to be far from trivial since there were eight poles to match. A new attempt was made using the LQG controller design method based on state space representation. The design method uses weight matrices that was easily tuned to get the desired closed loop performance. The latter method is much nicer to work with since the weighting matrices has some physical representation and only one matrix is varied at a time, beginning with the command signal. By introducing a vector orthogonal to the eigenvectors of the rest of the states, the closed-loop pole of a certain state could be moved arbitrarily.

- The parameters of the dead-zone can be estimated online with a state observer.

An adaptive scheme has also been added as a part of the controller structure. The purpose is to track and compensate for the dead-zone. Different adaptive methods were tested, gradient, stability theory and a state observer. The first two approaches could not estimate the dead-zone position so they were rejected. With a state observer it is possible to estimate both the dead-zone position and the spring torque. The state observer was designed using optimal control theory, i.e. a Kalman filter.

- Oscillations are inevitable but can be reduced when compensating for the dead-zone.

If the reference value is close or inside the dead-zone, the result will be oscillations in the plate position, so called limit cycles. The way to reduce this problem is to always have good estimates of the dead-zone parameters. It was also found that there was an overlap between the springs creating the dead-zone. The result will be that it is not advisable to set any reference values closer than ±0.5° around an estimate of the dead-zone position.

13 APPENDIX A: DATA SHEET

13.1 Above the dead-zone

Identified system close to the dead-zone on state-space format,

$$\Phi = \begin{bmatrix} 0.9906 & -0.1838 & 0.0333 \\ 0 & 1.9378 & -0.9401 \\ 0 & 1 & 0 \end{bmatrix}, \ \Gamma = \begin{bmatrix} 0.0292 \\ 0.1166 \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} -0.1166 & 0 & 0 \end{bmatrix}, \ D = 0$$

or on ZPK-format

$$ZPK\left(\begin{bmatrix} 2.1806 & 0.4922 \end{bmatrix}, \ \begin{bmatrix} 0.9906 & 0.9689 \pm 0.0371i \end{bmatrix}, \ -0.0034 \ , \ 0.001 \right)$$

Controller design using the LQG method:
Extended system with an integrator state

$$\Phi_e = \begin{bmatrix} \Phi & 0 \\ C & 1 \end{bmatrix}, \ \Gamma_e = \begin{bmatrix} \Gamma \\ 0 \end{bmatrix}$$

$$C_e = \begin{bmatrix} C & 0 \end{bmatrix}, \ D_e = 0$$

Controller weighting matrices

$$Q_1 = \text{diag}\begin{bmatrix} 1 & 3 & 3 & 1 \end{bmatrix}, \ Q_2 = 100 \ , \ Q_{12} = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T$$

$$q = \begin{bmatrix} 0.0027 & -0.0117 & 0.009 & 1{,}0 \end{bmatrix}^T \ , \ \alpha = 10e4$$

Controller feedback gain matrix

$$Le = \begin{bmatrix} -0.1457 & 2.2668 & -1.8970 & -27.7395 \end{bmatrix}$$

State space Kalman observer design:
Extended state-space system with an estimation of a constant disturbance on the input to the process.

$$\Phi_o = \begin{bmatrix} \Phi & \Gamma \\ 0 & 1 \end{bmatrix}, \ \Gamma_o = \begin{bmatrix} \Gamma \\ 0 \end{bmatrix}$$

$$C_o = \begin{bmatrix} C & 0 \end{bmatrix}, \ D_o = 0$$

Observer weighting matrices

$$R_1 = \text{diag}\begin{bmatrix} 1 & 12 & 12 & 0{,}1 \end{bmatrix}, \ R_2 = 0.1 \ , \ \mathrm{R}_{12} = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T$$

$$q = \begin{bmatrix} 0.0009 & -0.0662 & 0.0627 & 1.0 \end{bmatrix}^T \ , \ \alpha = 30$$

Observer gain matrix

$$K = [-11.4934 \quad 42.4049 \quad 32.8217 \quad 8.4208]$$

LQG controller converted to RST format

$$\begin{aligned} R(q) &= q^4 - 2.3279q^3 + 2.3693q^2 - 1.3526q + 0.3112 \\ S(q) &= 35.5616q^3 + -101.1234q^2 + 95.8813q - 30.3139 \\ T(q) &= q^4 - 2.5880q^3 + 2.6317q^2 - 1.2938q + 0.2501 \end{aligned}$$

13.2 Below the dead-zone

Identified system close to the dead-zone on state-space format,

$$\Phi = \begin{bmatrix} 0.9098 & -0.2002 & 0.0502 \\ 0 & 1.9211 & -0.9226 \\ 0 & 1 & 0 \end{bmatrix}, \ \Gamma = \begin{bmatrix} 0.0077 \\ 0.0982 \\ 0 \end{bmatrix}$$

$$C = [-0.1227 \quad 0 \quad 0\,], \ D = 0$$

or on ZPK-format

$$ZPK([4.1036 \quad 0.3815], \ [0.9697 \quad 0.9514 \quad 0.9098], \ -0.000941, \ 0.001)$$

Controller design using the LQG method:
Extended system with an integrator state

$$\Phi_e = \begin{bmatrix} \Phi & 0 \\ C & 1 \end{bmatrix}, \ \Gamma_e = \begin{bmatrix} \Gamma \\ 0 \end{bmatrix}$$

$$C_e = [C \quad 0], \ D_e = 0$$

Controller weighting matrices

$$Q_1 = \text{diag}\,[0.3 \quad 4.2 \quad 2.1 \quad 1], \ Q_2 = 140 \ , \ Q_{12} = [0 \quad 0 \quad 0 \quad 0]^T$$

$$q = [0.0013 \quad -0.0101 \quad 0.0082 \quad 1{,}0]^T \ , \ \alpha = 20e4$$

Controller feedback gain matrix

$$Le = [-0.0491 \quad 1.7510 \quad -1.4972 \quad -34.6661]$$

State space Kalman observer design:
Extended state-space system with an estimation of a constant disturbance on the input to the process.

$$\Phi_o = \begin{bmatrix} \Phi & \Gamma \\ 0 & 1 \end{bmatrix},\ \Gamma_o = \begin{bmatrix} \Gamma \\ 0 \end{bmatrix}$$
$$C_o = \begin{bmatrix} C & 0 \end{bmatrix},\ D_o = 0$$

Observer weighting matrices

$$R_1 = \text{diag}\begin{bmatrix} 1 & 15 & 15 & 0{,}01 \end{bmatrix},\ R_2 = 0.05\ ,\ \mathrm{R}_{12} = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T$$
$$q = \begin{bmatrix} 0.0002 & -0.0190 & 0.0175 & 1{,}0 \end{bmatrix}^T\ ,\ \alpha = 30$$

Observer gain matrix

$$K = \begin{bmatrix} -11.7182 & 46.5102 & 35.0284 & 10.7837 \end{bmatrix}$$

LQG controller converted to RST format

$$R(q) = q^4 - 2.2214q^3 + 2.0288q^2 - 10.9987q + 0.1913$$
$$S(q) = 29.6045q^3 + -82.2013q^2 + 76.0486q - 23.4427$$
$$T(q) = 1.1q^4 - 2.6323q^3 + 2.4505q^2 - 1.1145q + 0.1964$$

14 APPENDIX B: DFA CALCULATIONS

14.1 Hard dead-zone compensation

The describing function analysis is in this case divided into two areas.

$$A \geq \left|\hat{\theta}_p - \theta_p\right| \quad ; c = \left(b_l - b_u\right)$$

$$u(\phi) = \begin{cases} -c & 0 < \phi < \pi \\ 0 & otherwise \end{cases}$$

$$a_1 = \frac{1}{\pi}\int_0^{\pi} -c \cdot \cos\phi \; d\phi = 0$$

$$b_1 = \frac{1}{\pi}\int_0^{\pi} -c \cdot \sin\phi \; d\phi = \frac{c}{\pi}\left[\cos\phi\right]_0^{\pi} = -\frac{2c}{\pi}$$

$$N(A) = -\frac{2c}{\pi A}$$

$$A \geq \left|\hat{\theta}_p - \theta_p\right| \quad ; c = \left(b_l - b_u\right)$$

$$u(\phi) = \begin{cases} -c & 0 < \phi < \phi_F \;, \pi - \phi_F < \phi < \pi \\ 0 & otherwise \end{cases} \qquad ; \phi_F = \arcsin\left(\frac{\left|\hat{\theta}_p - \theta_P\right|}{A}\right)$$

$$a_1 = \frac{1}{\pi}\int_{0,\pi-\phi_F}^{\phi_F,\pi} -c \cdot \cos\phi \; d\phi = -\frac{c}{\pi}\left(\left[\sin\phi\right]_0^{\phi_F} + \left[\sin\phi\right]_{\pi-\phi_F}^{\pi}\right) = 0$$

$$b_1 = \frac{1}{\pi}\int_{0,\pi-\phi_F}^{\phi_F,\pi} -c \cdot \sin\phi \; d\phi = -\frac{c}{\pi}\left(\left[-\cos\phi\right]_0^{\phi_F} + \left[-\cos\phi\right]_{\pi-\phi_F}^{\pi}\right) = -\frac{2c}{\pi}\left(1 - \cos\phi_F\right)$$

$$N(A) = -\frac{2c}{\pi A}\left(1 - \cos\phi_F\right)$$

14.2 Soft dead-zone compensation

The describing function analysis is in this case divided into three areas. The calculations are different for positive and negative $\left(\hat{\theta}_p - \theta_p\right)$, but the resulting describing function, *N(A)*, is the same.

$\left(\hat{\theta}_p - \theta_p\right) < 0$

$A < \left|\left(F_l - \theta_p\right)\right|$; $c = \left(b_l - b_u\right)$; *k* is the slope of the interpolation, [V/°].

$$u(\phi) = \begin{cases} -c & 0 < \phi < \pi \\ -k \cdot \left(A\sin\phi - F_l\right) & 0 < \phi < 2\pi \end{cases}$$

$$a_1 = \frac{1}{\pi}\int_0^{\pi} -c \cdot \cos\phi \; d\phi + \frac{1}{\pi}\int_0^{2\pi} -k \cdot \left(A\sin\phi - F_l\right) \cdot \cos\phi \; d\phi = 0$$

$$b_1 = \frac{1}{\pi}\int_0^{\pi} -c\cdot\sin\phi\ d\phi - \frac{1}{\pi}\int_0^{2\pi} kA\sin\phi - kF_l\ d\phi = -\frac{2c}{\pi} - \frac{kA}{\pi}\int_0^{2\pi}\sin^2\phi\ d\phi + \frac{kF_l}{\pi}\int_0^{2\pi}\sin\phi\ d\phi =$$

$$= -\frac{2c}{\pi} - \frac{kA}{2\pi}\left[\phi - \frac{\sin 2\phi}{2}\right]_0^{2\pi} + \frac{kF_l}{\pi}\left[-\cos\phi\right]_0^{2\pi} = -\frac{2c}{\pi} - kA$$

$$N(A) = -\frac{2c}{\pi A} - k$$

$\left|\left(F_l - \theta_p\right)\right| < A < \left|\left(F_u - \theta_p\right)\right|$; $c = \left(b_l - b_u\right)$; k is the slope of the interpolation, [V/°].

$$u(\phi) = \begin{cases} -c & 0 < \phi < \pi \\ -k\cdot\left(A\sin\phi - F_l\right) & -\phi_l < \phi < \pi - \phi_l \end{cases} \qquad ;\phi_l = \arcsin\left(\frac{\left|F_l - \theta_P\right|}{A}\right)$$

$$a_1 = \frac{1}{\pi}\int_0^{\pi} -c\cdot\cos\phi\ d\phi + \frac{1}{\pi}\int_{-\phi_l}^{\pi-\phi_l} -k\cdot\left(A\sin\phi - F_l\right)\cdot\cos\phi\ d\phi = 0$$

$$b_1 = \frac{1}{\pi}\int_0^{\pi} -c\cdot\sin\phi\ d\phi - \frac{1}{\pi}\int_{-\phi_l}^{\pi-\phi_l} kA\sin\phi - kF_l\ d\phi = -\frac{2c}{\pi} - \frac{kA}{\pi}\int_{-\phi_l}^{\pi-\phi_l}\sin^2\phi\ d\phi + \frac{kF_l}{\pi}\int_{-\phi_l\,0}^{\pi-\phi_l}\sin\phi\ d\phi =$$

$$= -\frac{2c}{\pi} - \frac{kA}{2\pi}\left[\phi - \frac{\sin 2\phi}{2}\right]_{\phi_l}^{\pi-\phi_l} + \frac{kF_l}{\pi}\left[-\cos\phi\right]_{-\phi_l}^{\pi-\phi_l} = -\frac{2c}{\pi} - \frac{kA}{2}\phi_l + \frac{kA}{2\pi}\sin 2\phi_l$$

$$N(A) = -\frac{2c}{\pi A} - \frac{k}{2}\phi_l + \frac{k}{2\pi}\sin 2\phi_l$$

$\left|\left(F_u - \theta_p\right)\right| < A$; $c = \left(b_l - b_u\right)$; k is the slope of the interpolation, [V/°].

$$u(\phi) = \begin{cases} -c & 0 < \phi < \phi_u\ ,\pi - \phi_u < \phi < \pi \\ -k\cdot\left(A\sin\phi - F_l\right) & -\phi_l < \phi < \phi_u\ ,\pi - \phi_u < \phi < \pi + \phi_l \end{cases} \qquad ;\phi_{l,u} = \arcsin\left(\frac{\left|F_{l,u} - \theta_P\right|}{A}\right)$$

$$a_1 = \frac{1}{\pi}\int_{0,\pi-\phi_u}^{\phi_u,\pi} -c\cdot\cos\phi\ d\phi + \frac{1}{\pi}\int_{-\phi_l,\pi-\phi_u}^{\phi_u,\pi+\phi_l} -k\cdot\left(A\sin\phi - F_l\right)\cdot\cos\phi\ d\phi = 0$$

$$b_1 = \frac{1}{\pi}\int_{0,\pi-\phi_u}^{\phi_u,\pi} -c\cdot\sin\phi\ d\phi - \frac{1}{\pi}\int_{-\phi_l,\pi-\phi_u}^{\phi_u,\pi+\phi_l} kA\sin\phi - kF_l\ d\phi =$$

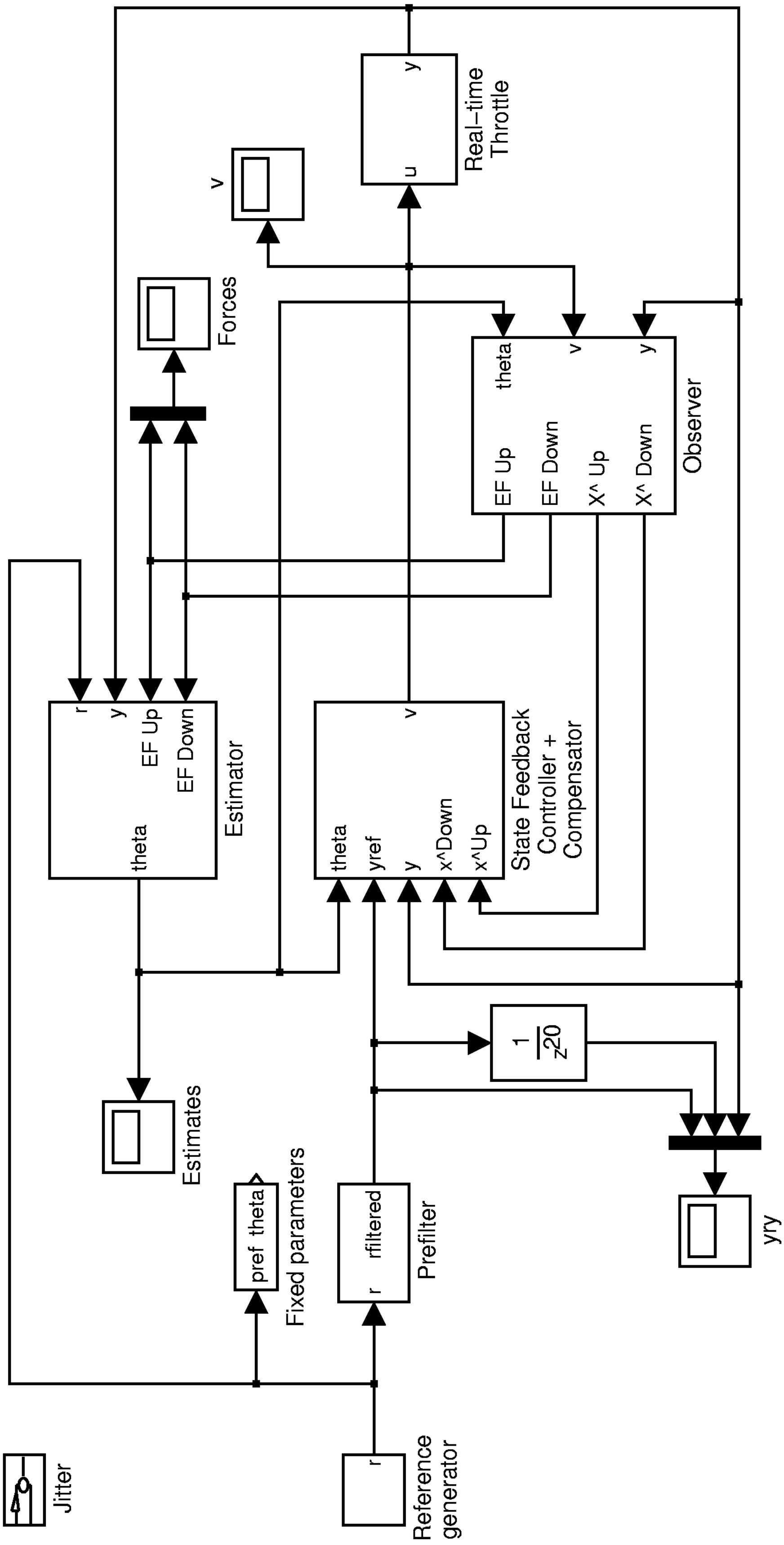
$$= -\frac{2c}{\pi}\left(1 - \cos\phi_u\right) - \frac{kA}{\pi}\int_{-\phi_l,\pi-\phi_u}^{\phi_u,\pi+\phi_l}\sin^2\phi\ d\phi + \frac{kF_l}{\pi}\int_{-\phi_l,\pi-\phi_u}^{\phi_u,\pi+\phi_l}\sin\phi\ d\phi =$$

$$= -\frac{2c}{\pi}\left(1 - \cos\phi_u\right) - \frac{kA}{2\pi}\left[\phi - \frac{\sin 2\phi}{2}\right]_{-\phi_l,\pi-\phi_u}^{\phi_u,\pi+\phi_l} + \frac{kF_l}{\pi}\left[-\cos\phi\right]_{-\phi_l,\pi-\phi_u}^{\phi_u,\pi+\phi_l} =$$

$$= -\frac{2c}{\pi} - \frac{kA}{\pi}\left(\phi_u - \phi_l\right) + \frac{kA}{2\pi}\left(\sin 2\phi_u + \sin 2\phi_l\right) - \frac{2kF_l}{\pi}\left(\cos\phi_u - \cos\phi_l\right)$$

$$N(A) = -\frac{2c}{\pi A} - \frac{k}{\pi}\left(\phi_u - \phi_l\right) + \frac{k}{2\pi}\left(\sin 2\phi_u + \sin 2\phi_l\right) - \frac{2kF_l}{\pi A}\left(\cos\phi_u - \cos\phi_l\right)$$

15 APPENDIX C: Simulink model structure



Bibliography

[1] Svensson, M., 1999. Master of Science Thesis: Robust design of an electronic throttle controller. Volvo Technological Development. Göteborg: Chalmers University of Technology

[2] Johansson, R., 1999. System Modeling and Identification. Englewood Cliffs, New York.: Prentice-Hall

[3] Åström, K.J., and B. Wittenmark, 1997. Computer Controlled Systems-Theory and Design, 3rd edition. Englewood Cliffs, New York.: Prentice-Hall

[4] Åström, K.J., and B. Wittenmark, 1995. Adaptive Control, 2nd edition. Englewood Cliffs, New York.: Prentice-Hall

[5] Glad, T., L.Ljung, 1989. Reglerteknik Grundläggande teori. Lund.: Studentlitteratur

[6] Andersson, L., U. Jönsson, K-H. Johansson, 1994. A Manual for System Identification. Department of Automatic Control, Lund.: Lund Institute of Technology

[7] Tao, G., P.V. Koktović, 1996. Adaptive Control of Systems with Actuator and Sensor Nonlinearities. New York.: John Wiley & sons Inc

[8] Slotine, J-J. E. and Li, Weiping, 1991. Applied Nonlinear Control. Englewood Cliffs, New Jersey.: Prentice-Hall

[9] Schmidtbauer, B., 1995. Analog och digital reglerteknik, Andra upplagan.: Studentlitteratur

[10] Volvo Technical Development, internal reports and correspondence.

[11] Andersson, B D.O. and J.B. Moore, 1989. Optimal Control, Linear Quadratic Methods. Englewood Cliffs, N.Y.: Prentice-Hall