

Institutionen för systemteknik

Department of Electrical Engineering

Examensarbete

Model-based Air and Fuel Path Control of a VCR Engine

Master's thesis
performed in **Vehicular Systems**

Tobias Lindell

LiTH-ISY-EX--09/4274--SE

Linköping 2009



Linköpings universitet
TEKNISKA HÖGSKOLAN

Model-based Air and Fuel Path Control of a VCR Engine

Master's thesis
performed in **Vehicular Systems**
Dept. of Electrical Engineering
at **Linköping University**


Tobias Lindell

LiTH-ISY-EX--09/4274--SE

Supervisor: **Ph.D. Student Oskar Leufven**
isy, Linköpings universitet

Examiner: **Associate Professor Lars Eriksson**
isy, Linköpings universitet

Linköping, 26 November, 2009

	Avdelning, Institution Division, Department Division of Vehicular Systems Department of Electrical Engineering Linköpings universitet SE-581 83 Linköping, Sweden		Datum Date 2009-11-26										
	Språk Language <input type="checkbox"/> Svenska/Swedish <input checked="" type="checkbox"/> Engelska/English <input type="checkbox"/> _____	Rapporttyp Report category <input type="checkbox"/> Licentiatavhandling <input checked="" type="checkbox"/> Examensarbete <input type="checkbox"/> C-uppsats <input type="checkbox"/> D-uppsats <input type="checkbox"/> Övrig rapport <input type="checkbox"/> _____	ISBN _____ ISRN LiTH-ISY-EX--09/4274--SE Serietitel och serienummer ISSN Title of series, numbering _____										
URL för elektronisk version http://www.fs.isy.liu.se/ http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-ZZZZ													
<table border="0"> <tr> <td style="vertical-align: top;">Titel</td> <td>Modellbaserad luft- och bränslereglering av en VCR-motor</td> </tr> <tr> <td style="vertical-align: top;">Title</td> <td>Model-based Air and Fuel Path Control of a VCR Engine</td> </tr> <tr> <td colspan="2" style="height: 40px;"></td> </tr> <tr> <td style="vertical-align: top;">Författare</td> <td>Tobias Lindell</td> </tr> <tr> <td style="vertical-align: top;">Author</td> <td></td> </tr> </table>				Titel	Modellbaserad luft- och bränslereglering av en VCR-motor	Title	Model-based Air and Fuel Path Control of a VCR Engine			Författare	Tobias Lindell	Author	
Titel	Modellbaserad luft- och bränslereglering av en VCR-motor												
Title	Model-based Air and Fuel Path Control of a VCR Engine												
Författare	Tobias Lindell												
Author													
Sammanfattning Abstract <p>The objective of the work was to develop a basic control system for an advanced experimental engine from scratch. The engine this work revolves around is a Saab variable compression engine.</p> <p>A new control system is developed based on the naked engine, stripped of the original control system. Experiments form the basis that the control system is built upon. Controllers for throttles, intake manifold pressure for pressures less than ambient pressure and exhaust gas oxygen ratio are developed and validated. They were found to be satisfactory. The lambda controller is tested with several parameter sets, and the best set is picked to be implemented in the engine. Models necessary for the development and validation of the controllers are developed. These models include models for the volumetric efficiency, the pressure dynamics of the intake manifold, the fuel injectors and wall wetting.</p>													
Nyckelord Keywords Engine control, volumetric efficiency modeling, throttle control, lambda control													

Abstract

The objective of the work was to develop a basic control system for an advanced experimental engine from scratch. The engine this work revolves around is a Saab variable compression engine.

A new control system is developed based on the naked engine, stripped of the original control system. Experiments form the basis that the control system is built upon. Controllers for throttles, intake manifold pressure for pressures less than ambient pressure and exhaust gas oxygen ratio are developed and validated. They were found to be satisfactory. The lambda controller is tested with several parameter sets, and the best set is picked to be implemented in the engine. Models necessary for the development and validation of the controllers are developed. These models include models for the volumetric efficiency, the pressure dynamics of the intake manifold, the fuel injectors and wall wetting.

Acknowledgments

First and foremost I would like to thank my supervisor Oskar Leufven, who is gifted with a patience above human reckoning and a mind always open for different ideas. His help and guiding during this work have been vital. I would also, of course, like to thank my examiner Lars Eriksson for giving me the opportunity to conduct this master thesis to begin with, and for allowing me to tinker with an expensive experimental engine. I would also like to thank Andreas Thomasson for invaluable help with the throttles as well as inputs on almost everything imaginable I could think to ask about. Per Öberg also needs mentioning for inputs and help with the measuring system.

Contents

1	Introduction and outline	1
1.1	Background	1
1.2	Limitations	1
1.3	Outline	2
2	The Engine	3
2.1	Variable Compression Engines	3
2.2	The SVC Engine Concept	4
2.3	The Engine Control System	6
2.3.1	Hardware	6
2.3.2	Sensors	6
2.3.3	Software	8
3	Induction System	9
3.1	Volumetric Efficiency	9
3.1.1	Theory	9
3.1.2	Experiments	10
3.1.3	Results	11
3.1.4	Adaptation to Transients	12
3.2	Throttle Controllers	15
3.2.1	Theory	16
3.2.2	Experiments	20
3.2.3	Identification	23
3.2.4	Implementation	26
3.2.5	Validation	29
3.3	Super-charger	35
3.3.1	Theory	35
3.3.2	Experiments	36
3.3.3	Identification	36
3.4	Intake Manifold Pressure Controller for $p_{im} \leq p_{amb}$	38
3.4.1	Theory	38
3.4.2	Experiments, Modeling and Identification	40
3.4.3	Resulting Controller	47

4 Fuel	51
4.1 Fuel Injectors	51
4.1.1 Theory	51
4.1.2 Experiments	53
4.1.3 Results Assuming $p_{fuel} - p_{im} = \text{constant}$	53
4.1.4 Results Assuming $p_{fuel} - p_{im} \neq \text{constant}$	54
4.2 Open Loop Part of the Fuel Controller	60
4.2.1 Theory	60
4.2.2 Results	60
4.3 Wall Wetting	61
4.3.1 Theory	61
4.3.2 Experiments	62
4.3.3 Results	64
4.4 Closed Loop Part of the Fuel Controller	66
4.4.1 Theory	66
4.4.2 Identification	70
4.4.3 Implementation	70
4.4.4 Verification	72
5 Summary of Models and Controllers	77
6 Conclusions	83
7 Future Work	85
A Wall Wetting Plots	89
B Source Code for Lambda Controller Parameter Finding	96

Chapter 1

Introduction and outline

1.1 Background

This work was done at the division of Vehicular System at the Department of Electrical Engineering at Linköping University. The objective of the work was to develop a basic control system for an advanced experimental engine from scratch. The engine this work revolves around is a Saab variable compression, SVC, engine.

The engine came supplied with a control system, which is non-free and not very transparent or easy to manipulate by a user so that different experiments can be made. The main purpose of this work is to lay the foundations for an open control system that will substitute the control system supplied by Saab, based around very configurable hardware in the form of a MicroAutobox from the supplier dSpace. This will give much more freedom to add, change or in other ways manipulate what the control system outputs and/or takes as input.

The new control system will be developed based on the naked engine, stripped of the original control system. Experiments will form the basis that the control system is built upon (Ljung and Glad, 1991). Model and controller parameters are identified from these experiments and then the models and controllers are validated in their respective validity region. After that conclusions are drawn regarding model structure and/or controller performance.

1.2 Limitations

There have been two major limitations that have affected this work, and those are the reliability of the engine in the engine cell and the available time. The first has influenced the latter in that there has been a lot of time set aside for fault searching and repairing.

The limited time available has meant that the scope of this work has had to be re-

stricted, and one such restriction is that this work will only take into consideration the normally aspirated mode of operation of the SVC engine. The experimentation has also been limited due to time restraints and the fact that the engine has been standing still due to fault searching and repairs of the leaking intercooler and the oil-cooler found to be faulty.

1.3 Outline

The engine is presented in chapter 2. A general explanation of the basic fundamentals and purpose of a variable compression engine is presented in section 2.1, and a more specific description of the SVC engine is presented in section 2.2. The control system with its hardware and software, together with an explanation about the different sensors the engine is equipped with, is presented in section 2.3.

The induction system is examined in chapter 3. Here a model of the volumetric efficiency is presented in section 3.1. Section 3.2 handles the throttle controllers. A brief examination of properties of the super-charger is also conducted in section 3.3 of this chapter. An intake manifold pressure controller for intake manifold pressures below ambient pressure are developed in section 3.4.

The fuel system is examined in chapter 4. Here a model of the fuel injectors is presented in section 4.1. Open loop control is discussed in section 4.2. Section 4.3 presents a model of the wall wetting properties, and a closed loop fuel controller is developed in section 4.4.

Conclusions are drawn in chapter 6. Future work proposals are given in chapter 7.

Chapter 2

The Engine

2.1 Variable Compression Engines

Spark-ignited (SI) four stroke combustion cycles are often thought to be comparable to the ideal Otto cycle, which consists of isentropic compression, heat addition at constant volume, isentropic expansion and heat loss (to the environment) at constant volume. This is of course not true due to various real world problems like heat transfer. The efficiency of the ideal Otto cycle is

$$\eta = 1 - \frac{1}{r_c^{1-\gamma}} \quad (2.1)$$

where r_c is the compression ratio and γ is the ratio of specific heats. According to equation 2.1 the efficiency increases with increasing r_c and would peak at $\eta = 1$ when r_c reaches infinity. But in a real engine the rate of heat transfer also increases with higher compression ratio so the optimum efficiency occurs with a limited r_c . Caris and Nelson (1959), cited in Nilsson et al. (2006), found that increasing the compression ratio above 16:1 or 17:1 caused a decrease in engine efficiency. Another problem with a high compression ratio is that the temperature and pressure peaks are higher, which leads to higher risk of harmful auto-ignition, also known as knock, inside the cylinder (Heywood, 1988).

The ability to change the compression ratio of an engine online is an interesting way to improve the overall efficiency of combustion engines, and by doing that lowering the fuel consumption (Nilsson, 2007). This is achieved by allowing the engine to run with higher compression ratio at low loads when there is a low risk of harmful knock due to low intake manifold pressure and lowering the compression ratio at higher loads, when knock would have been inevitable with the higher compression ratio. An ordinary engine, with a fixed compression ratio, would have to either increase the amount of injected fuel or retard the ignition timing from the optimum ignition angle, consuming more fuel or producing less power, to achieve the lower end gas temperature that cures the knocking. The problems gets worse when the engine is running super-charged, due to the higher intake

manifold pressure which produces a higher end gas pressure with a higher knock tendency. One way to limit these problems is to find a compromise compression ratio, where the engine runs reasonable well at both low loads and high loads. This compromise compression ratio of an ordinary normally aspirated engine and, to an even larger degree, that of a supercharged engine is avoided with the variable compression engine.

2.2 The SVC Engine Concept

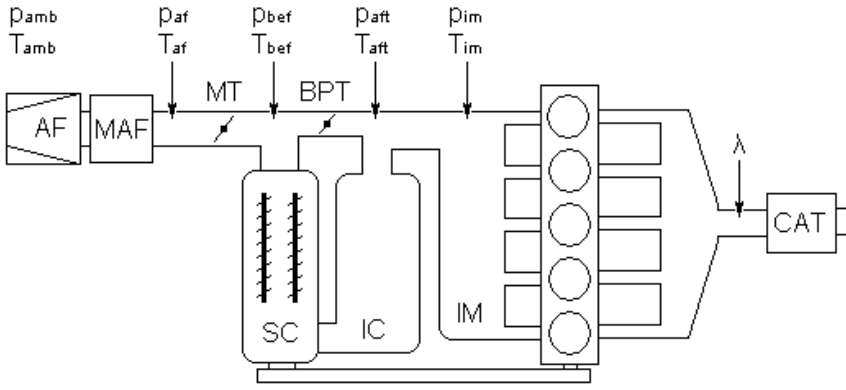


Figure 2.1. An overview of the SVC engine. The air filter (AF), the air mass flow meter (MAF), the main throttle (MT), the super-charger (SC), the by-pass throttle (BPT), the inter-cooler (IC), the intake manifold (IM) and the catalyst (CAT) can be seen. The different temperatures and pressures that are significant for this work have been added in their places as well as the lambda value (λ).

The SAAB Variable Compression (SVC) engine this work is based around is a straight inline five cylinder 1.6 liter engine, where the cylinders and cylinder head are forged as one unit called a mono-head (SAAB Automobile AB, 2000). This mono-head is tilt-able in relation to the engine block with the crank shaft. This means that the distance from crank shaft to cylinder head is made larger when the mono-head tilts from its original straight position. This in turn changes the size of the clearance volume inside the cylinder head, since the pistons don't travel as far up into the cylinders when the mono-head is tilted compared to when it is straight. And when the clearance volume is changed so is the compression ratio, as

$$r_c = \frac{V_{displacement} + V_{clearance}}{V_{clearance}} \quad (2.2)$$

The tilting of the mono-head on the SVC engine is done with hydraulics. There is a pressure sensor in the hydraulic system, that the control system uses with the help of a hydraulic pump to make sure the pressure is kept inside certain preset

values so that the tilting is predictable fast.

The air intake of the engine consists of an air filter (AF), an air mass meter (MAF), a main throttle body (MT), a mechanical super-charger (SC), a by-pass throttle body (BPT), a water-cooled intercooler (IC) and lastly an intake manifold (IM). The MT and BPT are controlled electronically. Both throttles have limp home functionality, where the MT is kept partially open when there is no drive voltage connected to it and the BPT is kept fully open at the same condition. The SC is belt-driven with an electronically actuated hydraulic clutch. The BPT is placed in parallel with the SC and controls how big a portion of the air that has passed the MT that goes through the SC. This means that there are two possible ways to achieve a controlled forced induction - either open the MT in full and regulate the intake manifold pressure (p_{im}) by actuating the BPT or close the BPT in full and actuate the MT to control p_{im} . After the air has passed through either the BPT or the SC it is collected, and then passed through the IC on it's way to the IM.

The fuel system consists of a fuel rail where fuel is held at certain unknown pressure with the help of a mechanical pressure regulator, a fuel feeder pump that feeds the fuel rail from a fuel tank and five fuel-injectors. The fuel injection is a multi-point sequential port injection system, meaning that each injector injects fuel into the intake port just before the cylinder's intake valve. Each injection is usually timed to occur just after the closing of the intake valves of the respective cylinder, independently of the other injectors. This timing of the injection is done to cool the intake valves during the combustion and allow for the best possible vaporisation of the fuel.

The ignition system consists of five independent ignition coils (one for each cylinder) that can be individually controlled by the control system. Each ignition coil has two circuits, one primary that charges the coil and a secondary that the energy stored in the primary coil is released into and that sends energy to the spark plug and creates the ignition pulse (spark). The actuator signals are dwell angle and ignition angle, where ignition angle is the angle before top dead center (TDC) of the respective cylinder that the ignition pulse is 'released' to the spark plug. Dwell angle is the measure of time that the primary circuit is charged before the ignition angle. The circuit will be shortened for the fraction of a revolution that the dwell angle represents.

With the help of the SC, and the variable compression, the power of the SVC's 1.6 liter is designed to be comparable to that of a 3.0 liter big NA engine. The engaging and disengaging of the SC has the potential of introducing sudden leaps in p_{im} , which is something that a p_{im} controller needs to take into account so that a good driveability is delivered to a potential driver. There is also a considerable amount of friction introduced by the SC, which also needs to be compensated for so that there are no unnecessary leaps in power output when engaging and disengaging the SC.

2.3 The Engine Control System

The engine control system consists of a prototyping real-time system that collects, assesses and makes decisions based on the values of the sensors that are placed on the engine and then sends out signals to the actuators on the engine. This is the same function as an ordinary engine's electronic control unit (ECU) has, but the system used for this work is much more flexible.

2.3.1 Hardware

The engine is controlled by a fast function prototyping real-time system that is called dSpace MicroAutobox (MABx), which is a aluminum box that contains an IBM PowerPC processor and comprehensible I/O capabilities. Additional signal conditioning and power stages are supplied by a connected RapidPro unit, which can be modularly expanded to suit the needs of the current engine/car. The RapidPro unit used has one RapidPro Control Unit, one RapidPro Signal Conditioning unit and two RapidPro Power units. Each unit is in itself modularly constructed, with slots for different predefined modules. The program of the MABx is constructed by building a model in Matlab/Simulink by using pre-defined control blocks, which is automatically generated into running code by compiling in Matlab and then merged with any added pre-compiled code.

2.3.2 Sensors

There are a number of sensors on the engine that are used by the engine control system. There are combined pressure and temperature sensors in the intake manifold and before and after the super-charger. There is a position sensor for the mono-head, which gives the current compression ratio. There are two engine position sensors that together gives the correct crank angle and current stroke of the engine, so that the control system knows for example which cylinder to ignite at a certain time. There is an air mass meter located before the main throttle. There are throttle position sensors on both the main and on the by-pass throttle, to measure the throttle plate angle. There are a hydraulic pressure sensor each for the mono-head and super-charger actuator pressure. There is a coolant water temperature sensor. And there is a continuous exhaust gas oxygen, lambda, sensor.

Lambda Sensor

The lambda sensor that the engine control system uses is a universal exhaust gas oxygen, UEGO, sensor. This is a wideband continuous oxygen sensor, and it is placed before the catalyst. It detects the oxygen ratio, the lambda value, in the exhaust gas, and gives a corresponding linear continuous voltage, between 0 and 5 V, that the RapidPro box measures at 80 Hz. The lambda value is used in the fuel feedback controller.

The engine is also equipped with two EGO sensors, which is a discrete type of

oxygen ratio sensor. The output of this type of sensor makes a jump in voltage output when a certain lambda threshold is passed, and is thus not quite as useful as a UEGO sensor. One sensor is placed before the catalyst and one after. These two sensors were not connected to the engine control system during this thesis.

Air Pressure and Temperature Sensors

The engine is equipped with several air pressure and temperature sensors. The sensors used by the engine control in this work are the "production" sensors the engine came equipped with from Saab. They are of a combined type, where the pressure and temperature sensors are paired together in one sensor body. There are three pairs of sensors placed on the engine, and they are located after the throttle, between the super-charger and the inter-cooler and at the intake manifold. The sensors gives a continuous linear voltage depending on the pressure/temperature, and the engine control system samples this voltage at 80 Hz.

There are several places in the induction system where it is possible to attach additional air pressure or temperature sensors, of a more accurate nature. They will not be used by the engine control system, but may be used to verify the "production" sensors or making faster and/or more accurate measurements if necessary.

Hydraulic Pressure Sensors

The tilting of the mono-head is done with hydraulic pressure, and the engine control system must ensure that there is enough pressure in the hydraulic system to be able to do that tilting. Therefore the pressure in the system needs to be controlled, and to able to do that the engine control system needs to measure the pressure. This is done with a hydraulic pressure sensor.

The sensor gives a continuous linear voltage depending on the pressure, and the engine control system samples this voltage at 80 Hz.

Mono-head Position Sensor

The position of the mono-head is directly related to the compression ratio of the engine. The engine control system therefore needs to measure this position to be able to control the current compression ratio. This is done with a position sensor mounted on the engine block. It is sampled at 80 Hz by the control system, and mapped to compression ratio.

Hall Effect Sensors

For the engine control system to be able to pinpoint the position of the crank shaft and the current stroke of the engine there is a Hall effect sensor mounted so that it can measure the intake cam. Such a sensor measure the change in magnetic field and varies its output voltage accordingly. On the intake cam there are tips that acts as magnets, and a set times per revolution a tip passes near the Hall sensor

and there is a spike in voltage output from the sensor and the position of the cam can be determined.

Air Mass Flow Meter

The air mass flow meter measures the air mass flow and outputs a frequency to the engine control system. This frequency is translated with a look-up table to the current air mass flow.

Throttle-plate Angle Sensors

The throttle are equipped with two throttle plate position sensors each. They are basically potentiometers that changes their resistance as the throttle plate moves. Only one sensor per throttle is used by the engine management. They are sampled at 320 Hz by the control system.

2.3.3 Software

To use the hardware in the engine control system there is some software needed to build the program, controllers etc. The software used by the control system is Matlab/Simulink, dSpace Real-time Interface (RTI) and dSpace ControlDesk.

Matlab/Simulink is an environment for model-based design and simulation for dynamic and embedded systems. It provides a graphical interface and customizable block libraries to design, simulate, implement and test models of time-varying systems. It uses a drag-and-drop interface, where the user selects blocks that will be used from the libraries, place them on a workspace and connects them by pulling connections between them. There is also the ability to directly embed Matlab algorithms as well as pre-compiled C code into the Simulink model. Matlab/Simulink then compiles the model into runnable code that is uploaded to the RapidPro by using dSpace ControlDesk.

Real-time Interface is a way of automatically implement Matlab/Simulink models on dSpace hardware. It is basically an add-on package that adds blocks to Simulink for communication with the hardware. There are several blocks that are pre-constructed for use by the automotive industry, like for example blocks that handles the fuel injection and ignition timing of an engine.

dSpace ControlDesk is used to access and manipulate the internal signals and variables of the MABx. It is also used to upload the program code that Matlab/Simulink and Real-time Interface delivers to the MABx. It provides a graphical interface where the user can build and use instrumentation layouts, where the internal signals and variables of the MABx are presented. It can be used to measure selected variables and export them to Matlab format for later use.

Chapter 3

Induction System

In this chapter the induction system of the engine and its parts are discussed and a reliable way to predict the mass of air that is fed to the cylinders is developed. The induction system consists of, in order, an air filter (AF), an air mass flow meter (MAF), a main throttle (MT), a by-pass throttle (BPT) in parallel with a super-charger (SC), an intercooler (IC), an intake manifold (IM) and finally the cylinders. For a graphical presentation see figure 2.1. Some parts of the induction system have been assumed to have little or no impact on the behavior of the air flow, they are the AF, the MAF and the IC. The IC cools the air, but the effect of the volume of the IC is paired with the volume of the IM and seen as one volume.

3.1 Volumetric Efficiency

The volumetric efficiency, η_{vol} , will be used to predict the amount of air going into the cylinders. This amount of air flow in turn will be used to calculate the feed forward part of the amount of fuel to be injected.

3.1.1 Theory

η_{vol} is the measurement of how effective the induction process of an engine is (Heywood, 1988). It is defined as the flow rate of air into the intake system, \dot{V}_a , divided by the swept flow rate of the pistons, \dot{V}_d .

$$\eta_{vol} = \frac{\dot{V}_a}{\dot{V}_d} = \frac{\dot{m}_a n_r}{\rho_{ai} V_{d,cyl} n_{cyl} N} \quad (3.1)$$

where ρ_{ai} is the intake manifold air density, \dot{m}_a is the air mass flow, $V_{d,cyl}$ is the swept volume of one cylinder, n_{cyl} is the number of cylinders, n_r is the number of revolutions per engine cycle ($= 2$ in this case, since this is a four stroke engine) and N is the engine speed.

ρ_{ai} can be calculated using the ideal gas law

$$\rho_{ai} = \frac{p_{im}}{RT_{im}} \quad (3.2)$$

where p_{im} is the intake manifold pressure, R is the specific gas constant for dry air ($= 287.05 \text{ J/kgK}$) and T_{im} is the temperature of the air in the intake manifold.

The air mass flow \dot{m}_a is measured as the air mass flow directly after the AF, \dot{m}_{af} . Since the air mass flow and the pressure and temperature are measured at different locations in the induction system steady-state condition, by which means that all engine actuators and sensor values are constant, is necessary. If that is achieved the assumption that the air mass flow at the air filter is the same as through the intake manifold can be drawn, and $\dot{V}_{a,im}$ can be calculated. And with steady-state condition it can also be assumed that the air flow rate into the cylinders, $\dot{V}_{a,cyl}$ is the same as the air flow rate through the intake manifold.

With these assumptions the volumetric efficiency is

$$\eta_{vol} = \frac{\dot{m}_a n_r R T_{im}}{p_{im} V_d n_{cyl} N} \quad (3.3)$$

which can be used to calculate η_{vol} at steady-state load points.

According to Eriksson and Nielsen (2007) one widespread way to model the volumetric efficiency is simple black box models which is used here. The following are introduced here and subsequently tested:

$$\begin{aligned} \eta_{vol,mod1} &= k_0 + k_1 N + k_2 N^2 + k_3 p_{im} && \text{Hendricks and Sorensen (1990)} \\ \eta_{vol,mod2} &= k_0 + k_1 N p_{im} + k_2 N p_{im}^2 + k_3 N^2 p_{im} && \text{Crossley and Cook (1991)} \\ \eta_{vol,mod3} &= k_0 + k_1 N + k_2 p_{im} + k_3 N p_{im} \end{aligned}$$

$\eta_{vol,mod3}$ is just a combination of the most basic parts of the two other models.

3.1.2 Experiments

A series of steady-state measurements at different p_{im} and different N were done to get a simple map of the engine, using the ‘production’ sensors the engine are equipped with. These sensors were used, even though they are not as good as can be, because they are the sensors the control system will use to control the engine, and they are already in place and easy to incorporate into the control system. The first map was of the engine running normally aspirated, and the different p_{im} were 50, 75 and 100 kPa and the different N were 1000, 1500 and 2000 revolutions per minute (rpm).

There are a limited amount of data points, and this is due to the fact that the engine is running almost without any control system at all. The SC is not connected, and this makes running with higher p_{im} than the ambient atmosphere pressure impossible. The variations of p_{im} at 75 kPa is due to interference with

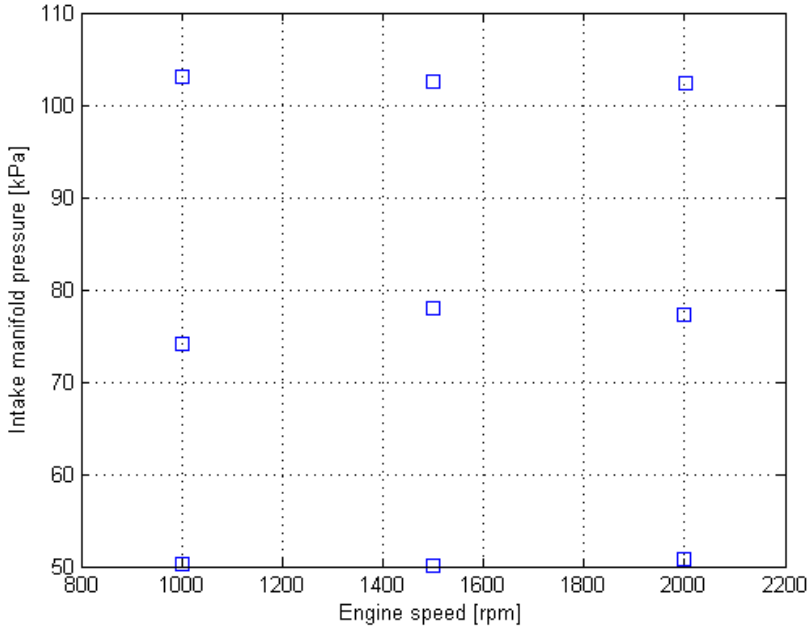


Figure 3.1. Engine speed and load points used to parameterize the initial η_{vol} black box models. The engine is running naturally aspirated, and the inaccuracies seen at 100kPa is due to sensor errors (both throttles are fully open and the pressure is approximately the same as the ambient pressure). The inaccuracies seen at 75kPa is due to lack of a proper intake manifold pressure controller at this stage.

the limp home functionality of the throttle and the fact that there is no p_{im} controller in place. The limp home functionality is what is making it hard to get the throttle plate angle to the exact right angle for achieving the right pressure in the intake manifold, but a p_{im} controller and a better MT controllers would solve that problem.

3.1.3 Results

First the map data was used to calculate the volumetric efficiency using equation (3.3). Then the parameters of the different η_{vol} -models, described above, was calculated using the least square method. Using the different inputs to the models the respective value of the volumetric efficiency for each map point was calculated and then compared to the calculated η_{vol} in table 3.1 below.

To be able to pinpoint the best η_{vol} -model the mean absolute value as well as the variance of the models relative errors were calculated. They can also be seen in

table 3.1.

According to table 3.1 $\eta_{vol,mod3}$ is the best, and is selected to be used in the control system. This is due to that model having both the smallest absolute mean relative error and the smallest variance of the relative errors. The volumetric efficiency model will be used to estimate the air mass flow into the cylinders based on the current values of the indata to the η_{vol} -model, namely the engine speed and the intake manifold pressure.

3.1.4 Adaptation to Transients

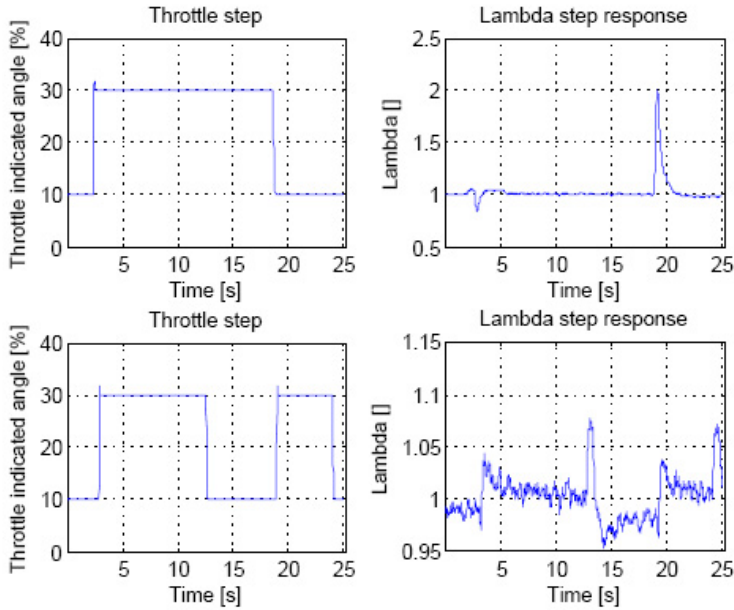


Figure 3.2. This plots show two steps in throttle position and their respective step responses in lambda. The two upper plots shows a step response where the injected fuel is based entirely on the open loop part of the fuel controller running with $\eta_{vol,mod3}$. It can be seen that there is a rather curious behavior where the lambda step response goes up very high when the throttle is closed. Some experiments were done to see what could cause this, and the results from these experiments were that the likely reason was the η_{vol} -model (based on p_{im}). One guess was that the p_{im} -sensor was filtered, or slow by other means, so that the resulting η_{vol} was slow in responding to changes in engine load point. Trying to solve this switching η_{vol} from one pre-calculated value to another at the same time as the change in throttle plate position occurred was tried. The two lower plots show such an experiment. As can be seen the lambda step response when changing throttle position is much better than the result of the result based on $\eta_{vol,mod3}$. This result is what $\eta_{vol,mod4}$ is based around.

When running the engine only on the open loop part of the fuel controller it was found that the η_{vol} -model chosen in previous section, $\eta_{vol,mod3}$, proved to lack a certain quickness in transient response. That meant that it was necessary to find a model better adapted to suit the needs of the control system. Some experiments were done where variables were changed at the same time as the change in throttle position occurred, starting with changing the actual fuel injected and going backwards in the fuel control chain to find the actual reason for this behavior. It was found that the likely reason was the η_{vol} -model used, see figure 3.2 for a comparison between the original open loop part of the fuel controller with $\eta_{vol,mod3}$ and the experiment where η_{vol} was changed at the same time (from one pre-calculated value to another) as the change in throttle position occurred. One guess to why is that the p_{im} -sensor is most likely filtered in some way, and therefore slow in responding to changes in p_{im} . One way around this is to base the η_{vol} -model on throttle position instead of p_{im} .

The following η_{vol} -model is introduced and tested in the same way as the other models previously. mt stands for main throttle in this model.

$$\eta_{vol,mod4} = k_0 + k_1 N + k_2 \theta_{mt} + k_3 N \theta_{mt}$$

The mean absolute value of $\eta_{vol,mod4}$ is 2.43 % and the variance of the relative error is 9.56 %. This model is also included in table 3.1 for comparison. This model is worse than the originally chosen model $\eta_{vol,mod3}$, but it should perform better in transients between different load points. The transient performance have, however, not been tested on the engine, due to the work limitations.

η_{vol} Calculated [%]	$\eta_{vol,mod1}$ Rel. error [%]	$\eta_{vol,mod2}$ Rel. error [%]	$\eta_{vol,mod3}$ Rel. error [%]	$\eta_{vol,mod4}$ Rel. error [%]
54.44	5.70	4.97	0.03	4.57
61.14	-2.03	0.66	-2.04	2.61
65.21	-4.31	-5.40	0.77	1.66
65.96	1.55	0.85	1.27	-5.15
70.32	0.99	5.22	0.90	-1.93
72.92	-0.03	-0.78	-0.05	-3.17
80.09	3.52	-0.03	-0.48	0.19
80.28	0.70	3.59	0.56	2.39
82.67	-5.00	-7.38	-0.93	-0.20
Mean abs error [%]	2.65	3.21	0.78	2.43
Variance [%]	12.04	18.74	1.07	9.56

Table 3.1. Comparing the results of the different η_{vol} -models at different engine speeds and p_{im} steady-state points. It can be seen that $\eta_{vol,mod3}$ gives the smallest absolute mean error and the smallest variance of relative errors. But, as can be seen in figure 3.2, the transient behavior of this η_{vol} -model is not that good, and $\eta_{vol,mod4}$ has been proposed as a possible solution to that problem. It can be seen that $\eta_{vol,mod4}$ performs worse at steady-state than $\eta_{vol,mod3}$ but better than the other two proposed models.

3.2 Throttle Controllers

There are two throttles in the SVC engine, a main (MT) and a by-pass throttle (BPT). The MT is placed before the super-charger (SC) in the induction system, and its function, just like throttles in any other SI engine, is to starve the engine of air so that the desired power output is achieved. The BPT is placed in parallel with the SC. The main function of the BPT, when the SC is not compressing the air, is to short circuit the SC. When the SC is running the BPT can be used either by blocking the by-pass route completely, thus allowing the MT to influence the degree of air compression, or the BPT will be used to control the pressure difference before and after the SC with a fully open MT. This will be discussed in more detail in chapter 3.3.

The throttles are of butterfly model, by which means they both respectively consist of a throttle body and a throttle plate used to restrict the air flow in the body tube. They are both electronically actuated. They have two position sensors each, of which one is used by the engine control system in this work. They also have an electric DC servo motor each that are used to change the position of the respective throttle plate in the throttle body. The DC motor is actuated by a PWM signal ranging from 0 to 1, where 0 is maximum torque in one direction, 1 is maximum in the other direction and 0.5 is "neutral" meaning that no motor torque is acting upon the throttle plate. The actuator signal in the controller is called u , and it ranges from -1 to 1. This signal will be transformed to a PWM signal in the controller. Both throttles have limp home functionality. The BPT will be fully open if the power to it is cut, which will make sure that there is air going to the cylinders but no risk of too high intake manifold pressure due to super-charging. The MT will go to a partially open state instead, which will let the engine run in a low load mode.

The limp-home of the BPT is a return spring pulling the throttle plate to a fully open state. The limp-home of the MT on the other hand is slightly more complex in that there are two springs that both pulls, from different directions, the throttle plate to a partially open limp-home position.

There is considerable friction that influence the behavior of the throttle plate as well, and this is not, unlike the limp-home effect, introduced into the throttle on purpose. According to Vařak et al. (2006) there is substantial friction in the large-series mechanical components, like bearings and gearbox, in the throttle. There is stiction friction, which occurs when the throttle plate is not moving, sliding friction, which occurs when the throttle plate is moving and something called pre-sliding friction, which is the friction that occurs in the transition between stiction and sliding friction.

3.2.1 Theory

Here a simplified model of an electronic throttle will be presented based on the findings in the papers by Eriksson and Nielsen (2000) and Thomasson and Eriksson (2009). A controller that uses the model will also be presented. There is a PID controller included in the complete controller, and the basic principles of such a controller will be presented and explained. A simplification for a throttle with a simpler limp-home functionality (without a limp-home nonlinearity) will also be presented.

There are five main parts of the torque acting on the throttle plate (Thomasson and Eriksson, 2009). They are the driving torque from the DC motor, T_{qu} , the return spring torque, T_{qs} , the static friction torque, T_{qfs} , the dynamic friction torque, T_{fv} and the back electromotive force, T_{emf} . The total equations of movement for the throttle plate are given by Newtons second law

$$\begin{aligned}\dot{\theta} &= \omega \\ J\dot{\omega} &= T_{qu} - T_{qs} - T_{qfs} - T_{fv} - T_{emf}\end{aligned}\tag{3.4}$$

where J is the moment of inertia of the the throttle plate, the gearbox and the DC motor together.

Since the models for both T_{fv} and T_{emf} are linear functions of angular velocity they are modeled together in a single torque model

$$T_{fv} + T_{emf} = K_{fv}\omega\tag{3.5}$$

where K_{fv} has the unit Nm s/rad.

To model the static friction, T_{qfs} , an ordinary Coulomb friction model have been used

$$T_{qfs}(Tq, \omega) = \begin{cases} Tq & \text{if } \omega = 0 \text{ and } |Tq| < T_{qc} \\ T_{qc} \text{sgn}(\omega) & \text{otherwise} \end{cases}\tag{3.6}$$

where the friction torque is equal to the applied torque Tq when there is no throttle plate movement and when $|Tq|$ is below the Coulomb friction T_{qc} . The friction can be clearly seen in figure 3.9 and compared to the real measurement in figure 3.7.

The friction compensation uses equation 3.6 to make a feed-forward circuit that tries to eliminate the impact of the friction on the throttle plate. Thomasson and Eriksson (2009) has chosen to base the friction compensator on the current tracking error, e , where the compensation is made in the direction the controller wants to move the throttle plate since this direction is the same direction as the throttle plate is moving. To avoid oscillations around $e = 0$ an ideal relay, which would have been what the equation above suggests, is not used, instead the friction compensation is ramped up to the maximum compensation.

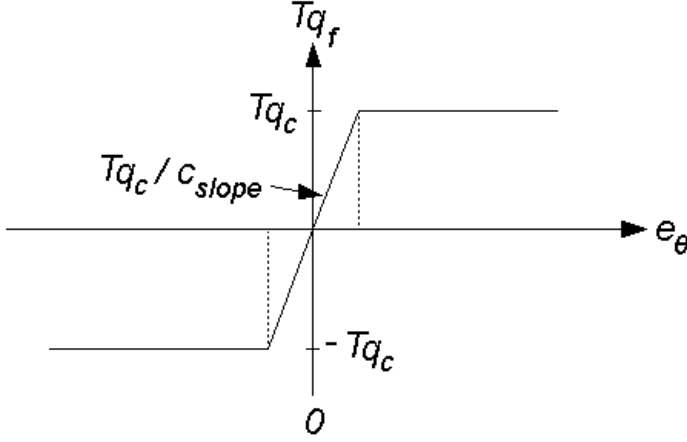


Figure 3.3. A sketch over the friction compensation of the main throttle plate. Tq_c is the identified Coulomb friction constant measured in controller output signal u , and the slope that the friction is ramped with between the minimum and maximum, or vice versa, friction with the slope constant $1/C_{slope}$ can be seen. This is the basis of the friction compensation model.

$$Tq_f(e_\theta) = \begin{cases} \frac{\tilde{T}q_c}{c_{slope}} e_\theta & \text{if } |Tq_f(e_\theta)| < \tilde{T}q_c \\ \tilde{T}q_c \operatorname{sgn}(e_\theta) & \text{otherwise} \end{cases} \quad (3.7a)$$

$$\tilde{T}q_c = Tq_c \cdot k \quad (3.7b)$$

where $\tilde{T}q_c$ is an adjusted Coulomb friction with an adjustment factor k slightly larger than 1 to make sure the initial friction is overcome so that the throttle plate starts to move straight away. c_{slope} is a manually calibrated slope constant according to figure 3.3. This friction compensator is what is used in this work.

The spring torque from the return springs in the throttle is handled as a piecewise linear model, illustrated in figure 3.4, where the spring constant differs according to equation 3.8 in regards to the limp-home nonlinearity.

$$Tq_s(\omega) = \begin{cases} m_{lh}^+ + k^+(\theta - \theta_{lh}^+) & \text{if } \theta > \theta_{lh}^+ \\ m_{lh}^+(\theta - \theta_{lh}) / (\theta_{lh}^+ - \theta_{lh}) & \text{if } \theta_{lh} < \theta \leq \theta_{lh}^+ \\ m_{lh}^-(\theta_{lh} - \theta) / (\theta_{lh} - \theta_{lh}^-) & \text{if } \theta_{lh}^- < \theta \leq \theta_{lh} \\ m_{lh}^- + k^-(\theta_{lh}^- - \theta) & \text{if } \theta \leq \theta_{lh}^- \end{cases} \quad (3.8)$$

The limp-home compensation uses this equation with the desired throttle angle, θ_{ref} , to make a feed-forward circuit that tries to eliminate the impact of the limp-home nonlinearity.

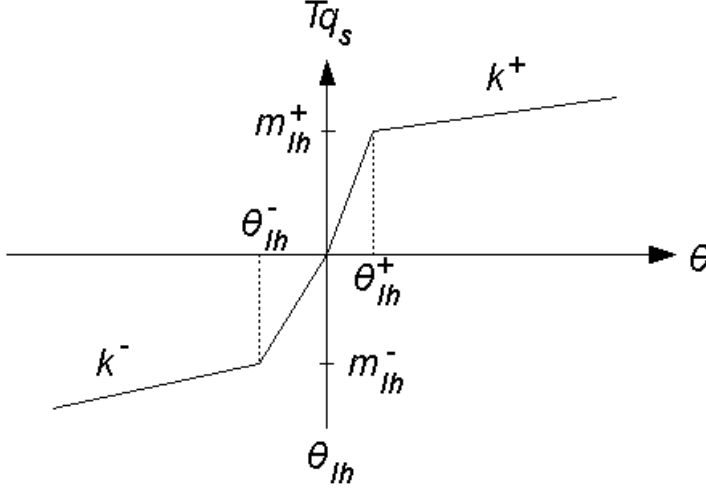


Figure 3.4. A sketch over the spring resistance acting on the main throttle plate. Tq_s is the spring torque measured in controller output signal u . m_{lh}^- , θ_{lh}^- and m_{lh}^+ , θ_{lh}^+ are the end-points of the limp-home nonlinearity, where m is the spring torque and θ is the throttle plate angle at the respective point. k^- and k^+ are the different spring constants on either side of the limp-home. This is the basis of the spring torque model.

Combining equations 3.5, 3.6 and 3.8 gives the complete torque model

$$J \frac{d\omega}{dt} = -K_{fv}\omega - Tq_s(\theta) - Tq_{fs}(T, \omega) + Ku \quad (3.9)$$

which will be used by the controller.

The BPT does not have the limp-home nonlinearity of the MT, but it has a simple limp-home functionality where a spring torque pulls the throttle plate in one direction (fully open) and it also has friction.

Included in the controller is a PID controller. The basic principle of a such a controller is output of the controller, in this case the indicated throttle plate angle, is compared to a desired set point, in this case the desired throttle plate angle. The error between actual and desired output is used to influence the process so that the right output value is reached. A PID controller consists of a proportional, a integrating and a derivative part, where the proportional part uses the current error, the integrating part uses the sum of recent errors and the derivative part uses the current rate of change of the error. See figure 3.5. Each part of the PID controller has its own tuning parameter, K_p , K_i and K_d respectively, and for the PID controller in the throttle controller they have been manually tuned to achieve the behavior of the total controller that was desired. The result of each parts contribution are then added together and sent to the process, which in this case

is the actual throttle. More about the basics of PID controllers can be found in for example Glad and Ljung (1989).

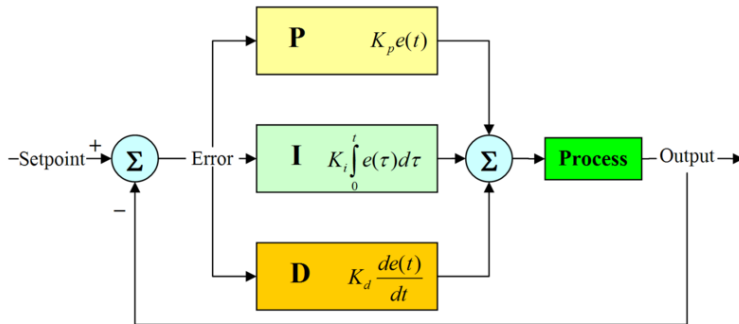


Figure 3.5. A PID controller (picture courtesy of Wikipedia.org) with its three parts, the proportional, the integrating and the derivative part.

3.2.2 Experiments

To find the spring and friction parameters necessary to parameterize the model a ramp response from the throttle is used. The throttle is fed a ramp signal, seen in figure 3.6, where the u signal is first ramped up and then ramped down. The ramp starts a bit over -1 and does not go all the way up to 1, this is because that the first and last part of the u signal have no influence on the throttle plate since the plate have already reached the corresponding end position. The extra bit of u in each direction can, and will be, used by the final controller to pull the throttle plate faster to the desired position, but it is not necessary when doing this slow ramp.

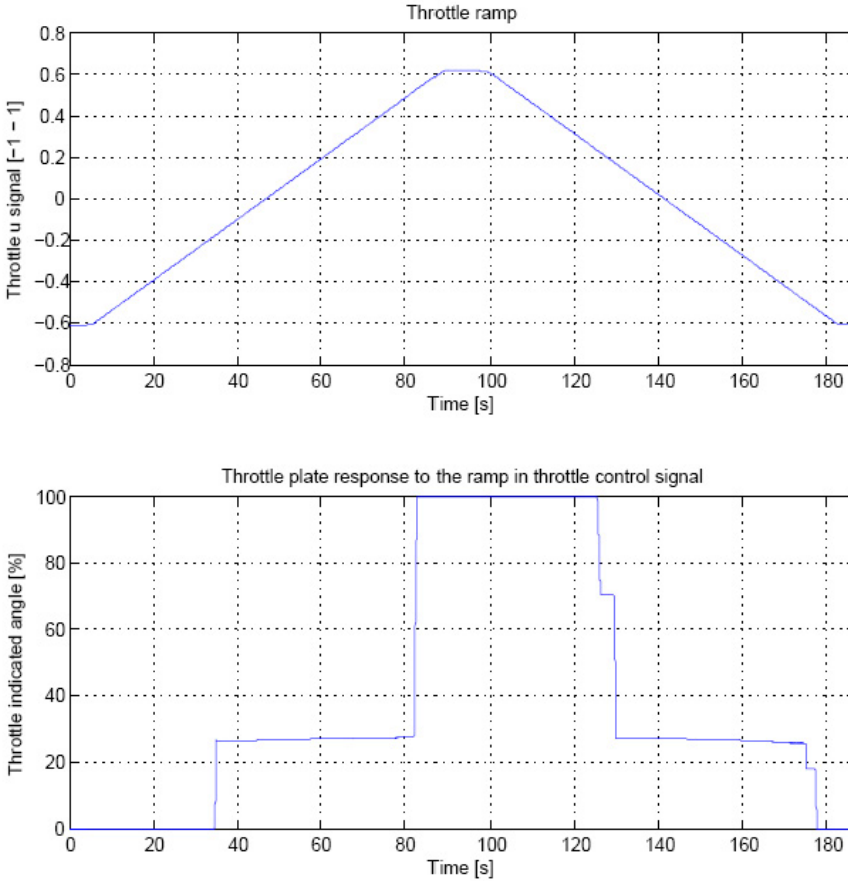


Figure 3.6. Throttle input, and controller output, u signal in throttle spring and friction experiment and also the main throttle response to this ramp. It can be seen that the throttle is actuated by a relatively slow ramp up and down. Since the plotted ramp response is for the MT the limp-home nonlinearity can be clearly seen (at around 27 % open throttle).

From this data the Coulomb friction can be seen, if we instead plot the throttle angle response in relation to the u signal. This can be seen in figure 3.7. In this figure there have been two responses to the same ramp plotted, so that it can be seen that there are individual difference from where the throttle plate "sticks" by friction.

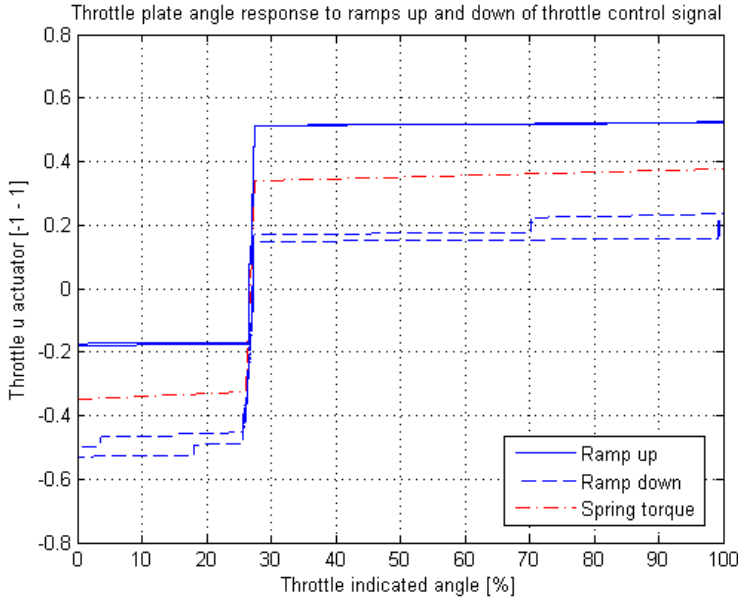


Figure 3.7. Friction acting on the main throttle plate. The limp-home non-linearity is clearly seen at around 27 % indicated open throttle. Two ramp responses have been plotted in this figure, and it can be seen that there are individual differences between the two ramps where the throttle plate "sticks" by friction. The calculated spring resistance is also plotted in this figure, so that the fit to the friction curves can be seen.

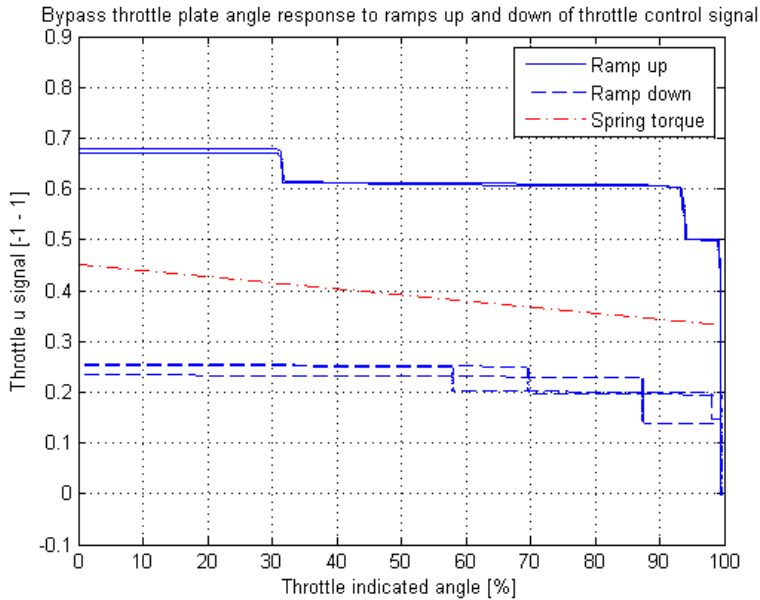


Figure 3.8. Friction acting on the by-pass throttle plate. The calculated spring resistance is also plotted in this figure, so that the fit to the friction curves can be seen. It can be seen that the throttle plate sticks to the throttle body in a worse way compared to the main throttle (compare with figure 3.7). The sticking is most severe near the maximum open throttle, and the approximated spring resistance suffers some from this. It is, however, not noticed when the controlling the throttle plate position (see figure 3.19).

3.2.3 Identification

To be able to parameterize the model described in 3.2.1 an indicated throttle angle ramp response experiment of throttle servo u signal is performed. The result of this experiment is illustrated in figure 3.9 where the important points of the curves are marked. These points will be used to determine the parameters of equations 3.6 and 3.8.

The Coulomb friction Tq_c , which is assumed to be equal in magnitude below and above the limp-home nonlinearity, is identified according to equation 3.10. It is basically the mean value of the difference of the curve of the up and down ramp response, with the exception of the limp-home.

$$Tq_{c,local,x} = \frac{u(Ax) - u(Bx)}{2} \quad (3.10a)$$

$$Tq_c = \frac{Tq_{c,local,1} + Tq_{c,local,2} + Tq_{c,local,3} + Tq_{c,local,4}}{4} \quad (3.10b)$$

Where $u(Yx)$ is the u signal at the point Yx, seen on the y-axe in the figure 3.9.

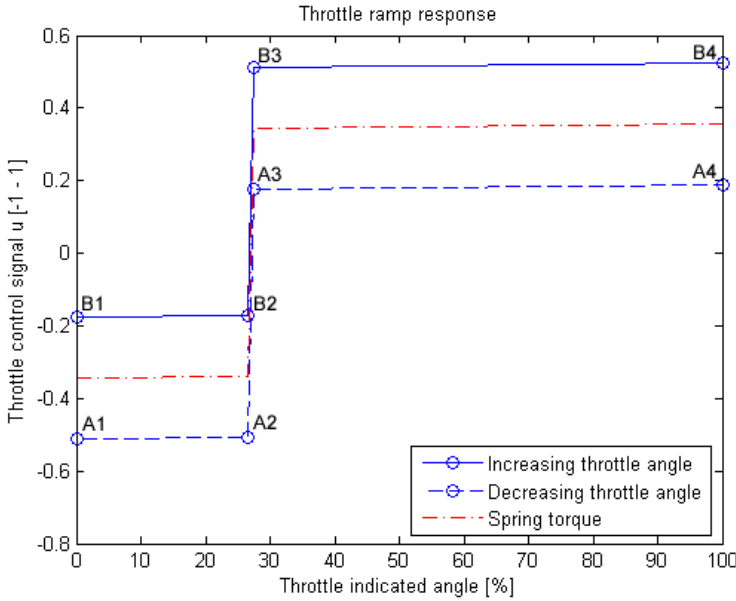


Figure 3.9. A sketch over the friction and the calculated spring resistance acting on the main throttle plate. The curves of a slow ramp up and down in the controller output u , which gives a increasing and decreasing throttle angle, can be seen. There is also a sketch over the spring torque resistance plotted, which is calculated from the other two curves. The points A1-A4 and B1-B4 are the points of interest for the different models in this work, and are used to identify the different model parameters. This is the experiment measurement that both the friction and the spring model are based upon.

The BPT consists of just an elongated rectangle A1-A2-B2-B1, with the appropriate friction and spring torque. There will be a spring torque slope k , a spring torque constant (the mean value of A1 and B1) and a Coulomb friction Tq_c associated with this throttle.

The limp-home position, θ_{lh} , is identified with

$$\theta_{lh} = \frac{\theta(A2) + \theta(A3) + \theta(B2) + \theta(B3)}{4} \quad (3.11)$$

where $\theta(Yx)$ is the indicated throttle angle at the point Yx, seen on the x-axe in the figure 3.9.

To parameterize the spring torque model the left-hand side of the limp-home non-linearity, θ_{lh}^- , is identified by

$$\theta_{lh}^- = \frac{\theta(A2) + \theta(B2)}{2} \quad (3.12)$$

and

$$m_{lh}^- = \frac{u(A2) + u(B2)}{2} \quad (3.13)$$

Finally the slope constant, k^- on the left-hand side of the limp-home is identified using

$$k^- = \frac{u(A2) - u(A1)}{\theta(A2) - \theta(A1)} \quad (3.14)$$

The right-hand side equations are not shown here, but they are determined in an equivalent manner to the equations for the left-hand side given here.

If we use the rules of identification set by figure 3.9 and equations 3.10 - 3.14 on the data in figure 3.7 the parameters of the MT can be identified and they are presented in table 3.2. Figure 3.8 is used, together with the "left" part of the rules of identification, to identify the parameters of the BPT which is presented in table 3.3. The respective PID parameters were identified by manual tuning, starting out from the PID parameters found in earlier work Ahlberg et al. (2008).

Parameter	Value	Parameter	Value
T_c	0.1590	K_p	1.2
θ_{lh}	26.75	K_i	50
θ_{lh}^-	26.09	K_d	0.08
θ_{lh}^+	27.36	k^-	0.0954
m_{lh}^-	-0.3230	k^+	0.0523
m_{lh}^+	0.3370		

Table 3.2. Parameters of the MT identified according to the identification rules stated in section 3.2.3. T_c is the Coulomb friction constant acting on the MT plate. θ_{lh} is the throttle angle where the limp-home non-linearity is placed, and θ_{lh}^- and θ_{lh}^+ are the end points of that nonlinearity. k^- and k^+ are the spring constants of the springs on the respective side of the limp-home pulling the throttle plate to the limp-home position. See figure 3.3 and 3.4. Also in this table are the tuned PID parameters (K_p , K_i and K_d) of the MT controller.

Parameter	Value	Parameter	Value
T_c	0.1993	K_p	1.8
θ_{lh}^-	100	K_i	60
m_{lh}^-	0.3300	K_d	0.08
k^-	-0.0013		

Table 3.3. Parameters of the BPT identified according to the identification rules stated in section 3.2.3. T_c is the Coulomb friction constant acting on the BPT plate. θ_{lh}^- in this case is the both the limp-home position and the end point of the limp-home "non-linearity" (there is no non-linearity for the BPT, but this parameter is used here so that the MT controller don't had to be changed). k^- is the spring constant of the spring pulling the BPT plate to the limp-home position, which in this case is negative (compare figure 3.8 to figure 3.7). Also in this table are the tuned PID parameters (K_p , K_i and K_d) of the BPT controller.

3.2.4 Implementation

The controller is implemented in Simulink to be compiled and transferred to the MABx control unit as part of the complete engine control system. There is a built

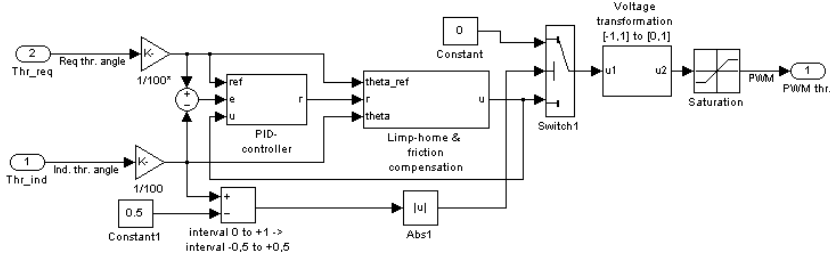


Figure 3.10. The complete throttle controller. The inputs are desired and indicated throttle plate angle in % and the output is u . In this figure the transformation to the PWM signal sent to the output block in dSpace is also seen, this transforms u , ranging from -1 to 1, to a PWM signal ranging from 0 to 1. The controller consists of a PID controller and a compensation block where the nonlinearities of limp-home and friction are linearised. The error between desired and indicated angle, e is calculated and sent to the PID controller as input. If the indicated throttle plate angle is smaller than 0 or larger than 100 u is set to 0. This is done to protect the throttle house, so that the throttle plate does not hit the extreme positions with force, and the gearbox inbetween the DC motor and the throttle plate.

in security in the controller, where the controller detects if the desired throttle angle is smaller or larger than 0 and 100 %. If such a signal is detected the PWM signal to the servo motor in the throttle is set to deliver no torque, in other words going to the limp-home position. This is done to protect the throttle, to assure that a faulty position sensor does not break anything mechanical.

The PID controller looks like figure 3.11. The internal value of the integrator is reset when there is a large step in desired reference throttle angle. There is an anti-windup circuit for the integrator part as well, which looks like figure 3.12. It works by allowing the integrator to work only when the absolute value of e , the error between desired and indicated throttle angle, is less than a set value. This value is set to 10 % in the controller. The integrator is also allowed to work when the controller output u is not saturated (in other words when u is between the output minimum and maximum -1 and 1). When those criteria are not fulfilled the integrator is sent a 0 so that the internal value of it is not changed.

The limp-home compensation is implemented by a look-up table, with the desired reference throttle angle as input and a feed forward spring torque compensation as output. The friction compensation, in the lower part of figure 3.13, is implemented as a gain which takes e and uses that to ramp up or down to the maximum or minimum Coulomb friction. The saturation block after the gain is making sure

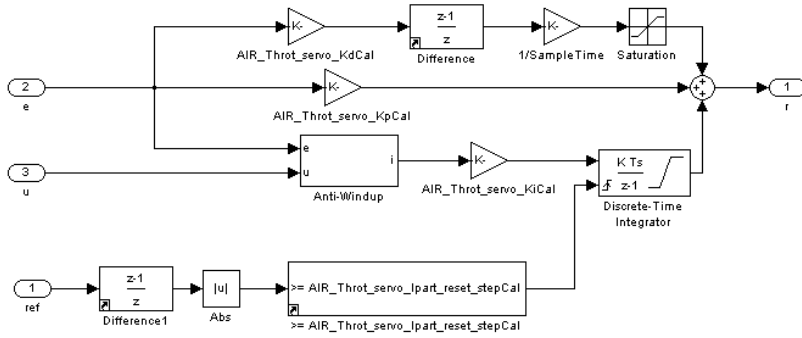


Figure 3.11. The PID controller in the controller. It consists of three parts, namely a proportional, a integrating and a derivative part. The proportional is just the error between desired and indicated throttle angle, e , multiplied by a constant, K_p . The integrating part integrates e multiplied with a constant, K_p . There is a anti-windup circuit before the integrator which is described at figure 3.12. There is also a reset circuit for the integrator, which resets the integrator if there is a large step in desired throttle plate angle. The derivative part uses the definition of the discrete derivation to derivate e multiplied with a constant, K_d . Finally the contributions of the three part are added together to the signal r .

that the compensation is kept inside the setpoints. The c_{slope} parameters decide the slope of the ramp and is set so that the ramp is as steep as possible without there being any oscillations due to the friction compensation switching too fast between maximum and minimum compensation.

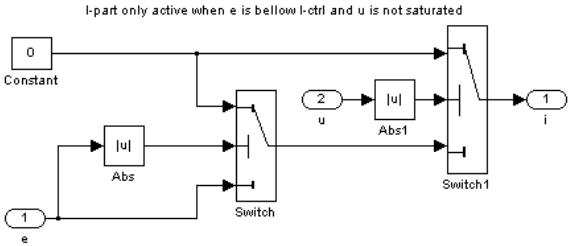


Figure 3.12. The integrator anti-windup part of the PID controller. The constant 0 is sent to the integrator in figure 3.11 if the absolute value of the error e is larger than a certain value, which is set to 10 % in the controller. A 0 is also sent to the integrator if the controller output u is saturated. When a 0 is sent to the integrator the internal value of the integrator does not change.

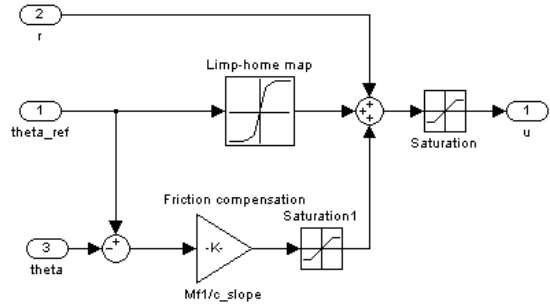


Figure 3.13. The friction compensation, in the lower part of this figure, is implemented as a gain which takes the error between desired and indicated throttle angle and ramps that up or down to the maximum or minimum Coulomb friction. The saturation block after the gain is making sure that the compensation is kept inside the setpoints. The limp-home compensation is implemented as a look-up table with the identified spring torque compensation.

3.2.5 Validation

To validate the controller below, through and above the limp-home position at around 27 % indicated open throttle step responses have been used, where the controller tries to follow a step in the desired reference throttle position.

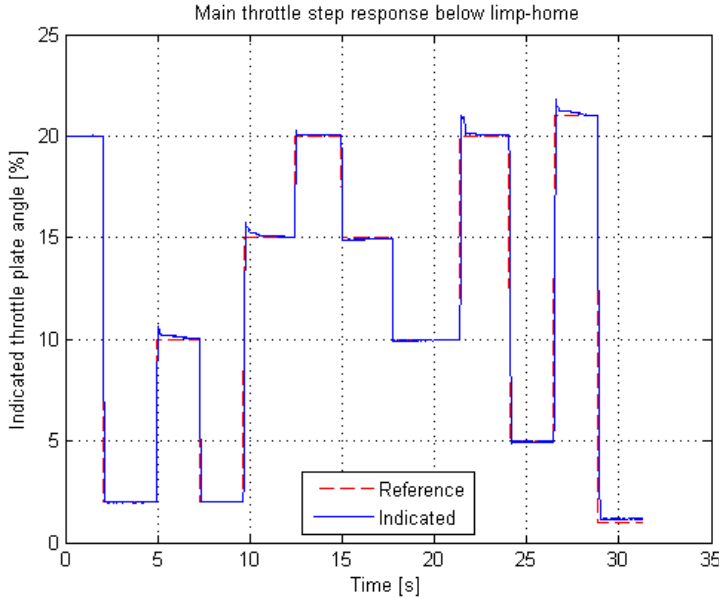


Figure 3.14. Step responses in main throttle plate indicated based on steps in the main throttle reference value. The steps are done below the limp-home so the accuracy of the friction and spring models together with the PID controller in that part can be seen. It can be seen that there are small over-shoots on the steps up.

It can be seen that there are small over-shoots in figure 3.14 when there is a positive step in the reference angle. This may be because the friction is different in the lower part of the throttle angle register than in the upper part, and the friction compensator uses the same magnitude of Coulomb friction in both parts. This could possibly be corrected by adding some complexity to the friction compensation like having the added torque request being different below and above the limp-home position. Or changing the PID parameters so that they are more optimized for stability in the lower region of indicated open throttle. The over-shoots are so small though, so it has not been a priority to get rid of them.

There does not seem to be any problems with the by-pass throttle. If one looks at the friction and spring torque in figure 3.8 there is a lot of nonlinearities due to the throttle plate sticking due to friction. It does not seem as that is an issue in the controller if one looks at the step responses in figure 3.19, in other words there is no sudden over-shoots due to wrong friction compensation being predicted. It may be that the PID controller is able to compensate for those implied errors or

it may be that the predicted friction is not as important on the smaller by-pass throttle.

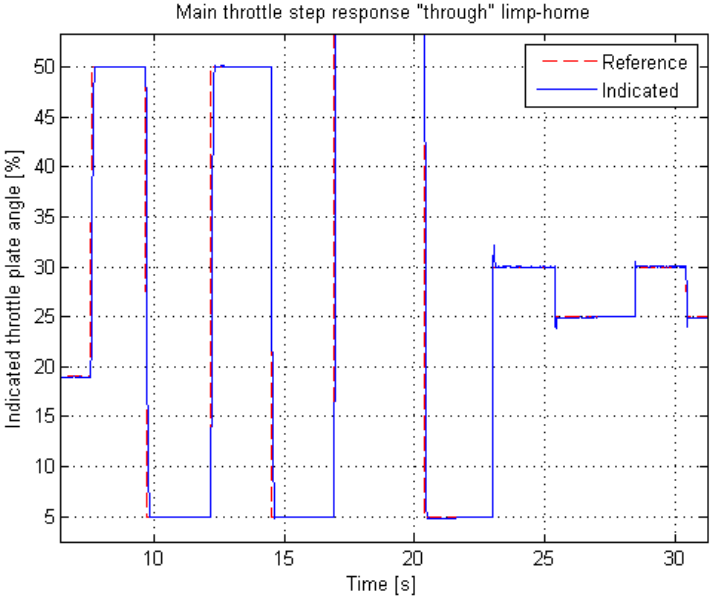


Figure 3.15. Step responses in main throttle plate indicated based on steps in the main throttle reference value. The steps are done through the limp-home so the accuracy of the friction and spring models together with the PID controller in that part can be seen. One of the steps in this figure has been zoomed in figure 3.16 so that the dynamics of the step response can be seen better.

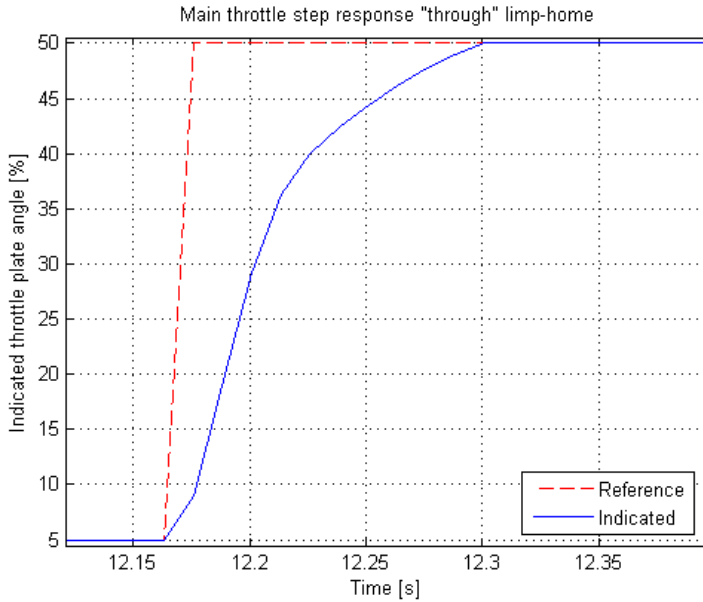


Figure 3.16. One of the steps of figure 3.15 zoomed, so that the dynamics of the step response can be seen better. It is seen in the figure that the rise time, the time it takes for the throttle plate to reach 63 % of its final value, in this step is around 48 ms. The delay time is very small, the throttle plate starts to move almost at the same time that the reference value changes. The reference value changes from 5 % to 50 % directly, the perceived slope of the plotted reference value is down to the 80 Hz that the variables are sampled with.

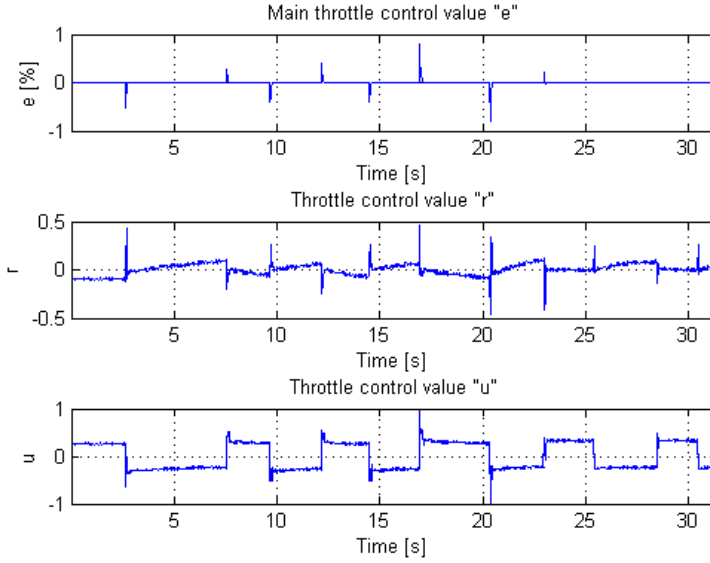


Figure 3.17. Influence of individual parts of the main throttle controller during the steps in figure 3.15. The first plot is the error between indicated throttle angle and reference angle, this is the variable that drives the controller. The second plot is the influence of the PID controller based on the error in the first plot. The third plot is u , which is signal r with the added friction and limp-home compensation, and this is the signal that the final PWM signal is based upon. The value of u can be between -1 and 1, and it can be seen that the signal is saturated in short intervals. The different internal signals of the by-pass throttle looks almost identical to these main throttle signals, the only difference are slightly different PID parameters, slightly different friction compensation and no limp-home nonlinearity.

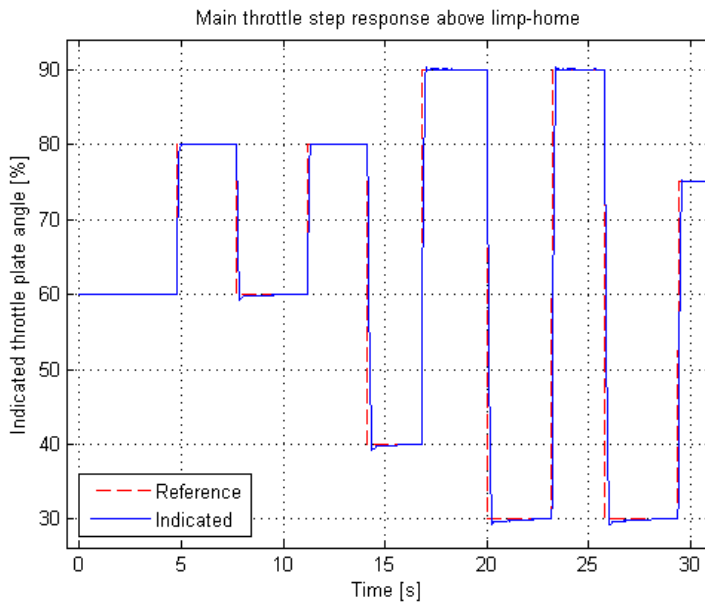


Figure 3.18. Step responses in main throttle plate indicated angle based on steps in the main throttle reference value. The steps are done above the limp-home so that the accuracy of the friction and spring models together with the PID controller in that part can be seen.

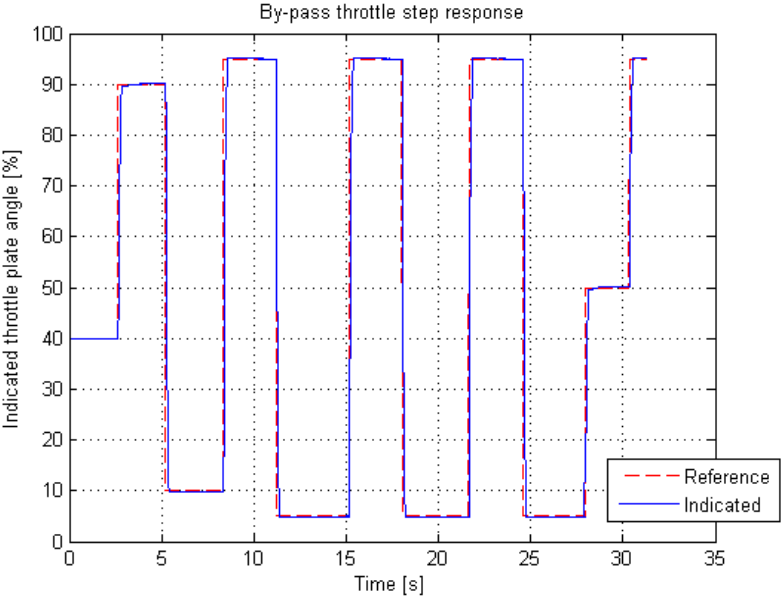


Figure 3.19. Step responses in by-pass throttle plate indicated angle based on steps in the by-pass throttle reference value. The steps are done so that the accuracy of the friction and spring models together with the PID controller can be seen.

3.3 Super-charger

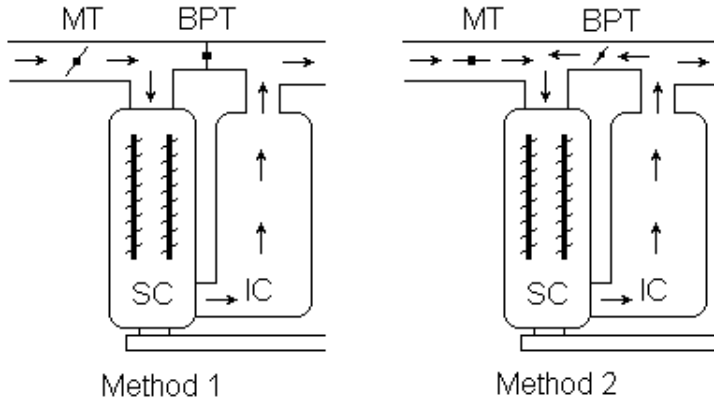


Figure 3.20. An overview of the two different methods for controlling p_{im} when $p_{im} > p_{amb}$. Method 1 uses the MT, with a fully closed BPT, to control the pressure before the SC so that the pressure after is correct while method 2 uses the BPT to control the pressure after the SC to the right value with a fully open MT. The arrows represents the air flow.

The SVC engine is equipped with a mechanical super-charger. The super-charger in this case is a positive displacement pump of Lysholm screw model, where two screws, which partially overlap leaving a small air pocket open in between, rotates and thus drives the air pocket forward axially. The space allowed for the air pocket grows smaller the further forward the pocket is pushed, which increases the pressure over the super-charger by a fixed ratio. The screws are driven by a belt from the crankshaft. The super-charger can be turned on or off by an electrically controlled hydraulic clutch.

3.3.1 Theory

When running the engine normally aspirated the control of the intake manifold pressure (p_{im}) is rather straightforward, it is controlled by restricting the air flow into the IM volume. This is done by using the main throttle (MT) while keeping the by-pass throttle (BPT) fully open. The super-charger (SC) can be either running or standing still for this to happen. This is a valid control method up to around the ambient pressure (there is a slight pressure drop over for example the air filter).

Control of p_{im} over ambient pressure (p_{amb}), however, is slightly more complex. First of all the fixed ratio SC must be running. That the SC is a fixed ratio SC means that the pressure directly after the SC will always be the pressure before the SC multiplied with this ratio. To control p_{im} the control system can either restrict the pressure before the SC to an appropriate pressure to achieve the right pressure

after the SC. This is done by using the MT to restrict the air flow before the SC, and keeping the BPT closed. This directs all the air flow into the SC, where it is compressed into the right pressure. This way is called method 1 in figure 3.20. The other way, method 2 in figure 3.20, is letting the pressure before the SC be p_{amb} , or as close as possible with the pressure restrictions from for example the air filter and the MT. Then the BPT, and the pressure differential over the BPT, is used to equalize the pressure after the SC to the right value.

There are different potential drawbacks from each of these different ways to control the level of super-charging. There could be that the air temperature after the compression could be high. Or it could be that running the SC under full load, where the SC compresses the air to maximum pressure possible (in other words when the pressure before the SC is as high as possible) consumes more engine torque than the other way.

3.3.2 Experiments

Two measurements were done where the p_{im} was set to 125 kPa. The measurements were done with the two different methods of controlling the p_{im} described in previous chapter. The measurement where the BPT is fully closed and the MT is used as actuator is called "method 1" and where the MT is fully open and the BPT is used as actuator is thus called "method 2". The lack of further measurements is due to the limitations.

3.3.3 Identification

The first thing to identify is the ratio of super-charging that the SC has. Measurement 1 is used. The pressure after the SC is 125 kPa and the pressure before the SC is 60 kPa. The ratio is identified as 2.08.

The second thing to identify is the subjectively best way to control p_{im} , based on the available data. p_{im} is the same, of course, but other things change. The first thing noticed is that the temperature of the air is different.

Method 1 gives the following temperatures: Before SC: 45 °C, After SC: 122 °C, IM - 44 °C.

Method 2 gives the following temperatures: Before SC: 43 °C, After SC - 74 °C, IM - 41 °C.

The net torque output, $T_{q,net}$, of the engine when running with the two methods described is the same, since p_{im} is the same. There is however a noticeable gross torque output difference, where method 2 gives around 4 % more output torque than method 1. This is down to the driving shaft power, P_{sc} , of the SC

$$P_{sc} = c_p \dot{m} \Delta T \quad (3.15)$$

where \dot{m} is the air mass flow through the SC and ΔT is the temperature change over the SC (Eriksson et al., 2002). In this case ΔT for method 1 ($122 - 45^\circ\text{C}$) is larger than for method 2 ($74 - 43^\circ\text{C}$), and this is the explanation of the difference in gross output torque.

This together with the higher temperature of method 1 gives that method 2 is best. That is, to use the BPT as actuator and keeping the MT fully open is considered the best way to control the p_{im} (given the limited amount of measurements to base a decision on).

3.4 Intake Manifold Pressure Controller for $p_{im} \leq p_{amb}$

The output power of an engine is related to the intake manifold pressure, so to control output power the pressure needs to be controlled. The intake manifold can be seen as a control volume, where the inlet is the air flow through the throttle and the outlet is the air flow that the engine pumps. The rate of change of the air mass inside the intake manifold is the difference between those two flows. Then the ideal gas law gives the pressure. One way to control the pressure in the intake manifold is by estimating the amount of air the engine pumps out of the volume and then actuate the throttle to let in as much air as is needed to achieve the right pressure. The throttle changes the air flow through it by restricting the flow area. This is done by changing the angle of the throttle plate.

3.4.1 Theory

The intake manifold, IM, can be seen as a control volume where the air mass flow in is restricted by the throttle and the air mass flow out is given by the engine state (engine speed, current intake manifold pressure and intake manifold temperature). An adiabatic model of the control volume has been used, by which means that the heat transfer is set to zero and that the temperature in the control volume is allowed to change. The model is based upon the mean value engine modeling in Eriksson (2007) and the thermodynamics in Kittel and Kroemer (1980). That the temperature is allowed to change means that the ideal gas law needs to be differentiated, which gives

$$V_{im} \frac{dp}{dt} = RT_{im} \frac{dm}{dt} + mR \frac{T_{im}}{dt} \quad (3.16)$$

where V_{im} is the volume of the IM, R is the specific gas constant for dry air, T_{im} is the current temperature of the air in the IM and m is the mass of the gas in the IM.

The mass is eliminated in equation 3.16 by using the fact that the rate of change of the mass is the same as the difference between the mass flow in and the mass flow out of the control volume, see equation 3.17, and the ideal gas law. Doing that and rewriting the equation gives equation 3.18 where p_{im} is the current IM pressure.

$$\frac{dm}{dt} = \dot{m}_{in} - \dot{m}_{out} \quad (3.17)$$

$$\frac{dp_{im}}{dt} = \frac{RT_{im}}{V_{im}}(\dot{m}_{in} - \dot{m}_{out}) + \frac{p_{im}}{T_{im}} \frac{dT_{im}}{dt} \quad (3.18)$$

Energy is conserved and stored in the system given by the control volume. In the modeling it has been assumed that the gas is ideal and that c_p and c_v are constant.

This gives that the temperature in the IM can be determined from the internal energy, as seen in equation 3.19.

$$U = mu(T_{im}) = mc_v T_{im} \quad (3.19)$$

The rate of change of the internal energy is given by the first law of thermodynamics, the energy is always conserved. There are flows of enthalpy coming in and going out of the control volume, but there is no mechanical work being done by or to the control volume (so there is no work transfer). This gives that the rate of change of the internal energy is given by

$$\frac{dU}{dt} = \dot{H}_{in} - \dot{H}_{out} - \dot{Q} \quad (3.20)$$

where \dot{H} is enthalpy flow and \dot{Q} is heat transferred from the air to the surroundings. The system is so fast that the heat transfer can be assumed to be zero. The enthalpy flows are given by

$$\dot{H}_{in} = \dot{m}_{in} c_p T_{in} \quad \text{and} \quad \dot{H}_{out} = \dot{m}_{out} c_p T_{out} \quad (3.21)$$

where the temperature of the outgoing mass flow, T_{out} , is the same as the temperature of the air inside the IM, T_{im} .

If we differentiate the internal energy, given in equation 3.19, we get

$$\frac{dU}{dt} = u(T) \frac{dm}{dt} + m \frac{du(T)}{dt} \quad (3.22)$$

If equations 3.21 and 3.22, together with the assumption that $\dot{Q} = 0$, are inserted into equation 3.20 the result is

$$mc_v \frac{dT_{im}}{dt} = \dot{m}_{in} (h(T_{in}) - u(T_{im})) - \dot{m}_{out} (h(T_{im}) - u(T_{im})) \quad (3.23)$$

The definition of enthalpy is

$$h(T) = u(T) + RT \quad (3.24)$$

Inserting equation 3.24 into equation 3.23, and rearranging the result somewhat together with equation 3.18, the result is

$$\begin{cases} \frac{dT_{im}}{dt} = \frac{RT_{im}}{p_{im} V_{im} c_v} (\dot{m}_{in} c_v (T_{in} - T_{im}) + R(T_{in} \dot{m}_{in} - T_{im} \dot{m}_{out})) \\ \frac{dp_{im}}{dt} = \frac{RT_{im}}{V_{im}} (\dot{m}_{in} - \dot{m}_{out}) + \frac{p_{im}}{T_{im}} \frac{dT_{im}}{dt} \end{cases} \quad (3.25)$$

which is a summarization of the total control volume model (Eriksson, 2007). This is the basis of the adiabatic IM model.

The air mass flow out from the control volume is given by

$$\dot{m}_{out} = \dot{m}_{ac} = \frac{\eta_{vol} p_{im} V_d N}{n_r R T_{im}} \quad (3.26)$$

where η_{vol} is the volumetric efficiency of the engine, V_d is the swept volume of the entire engine, n_r is the number of crank revolutions per power stroke and N is the current engine speed (Heywood, 1988). η_{vol} has previously been identified for the engine, see chapter 3.1.

The air mass flow into the control volume is, when running normally aspirated, given by the flow through the main throttle. This flow is defined (Heywood, 1988) as

$$\dot{m}_{in} = \dot{m}_{at} = \frac{p_{amb}}{\sqrt{RT_{amb}}} A_{th}(\alpha) C_{D,th}(\alpha) \Psi(p_r) \quad (3.27)$$

where p_{amb} is the air pressure before the throttle (assumed to be the ambient pressure), T_{amb} is the air temperature before the throttle, $A_{th}(\alpha)$ is the open flow area of the throttle which depends on the throttle plate angle α , $C_{D,th}$ discharge coefficient which also depends on the throttle plate angle, p_r is the pressure ratio $\frac{p_{im}}{p_{us}}$ and Ψ is defined as

$$\Psi(p_r) = \begin{cases} \sqrt{\frac{2\gamma}{\gamma-1} \left(p_r^{\frac{2}{\gamma}} - p_r^{\frac{\gamma+1}{\gamma}} \right)} & \text{for } p_r > p_{r,crit} \\ \sqrt{\frac{2\gamma}{\gamma-1} \left(\left(\frac{2\gamma}{\gamma+1} \right)^{\frac{2}{\gamma-1}} - \left(\frac{2\gamma}{\gamma+1} \right)^{\frac{\gamma+1}{\gamma-1}} \right)} & \text{otherwise} \end{cases} \quad (3.28)$$

where γ is the ratio of specific heats ($\frac{c_p}{c_v}$) and

$$p_{r,crit} = \left(\frac{2}{\gamma+1} \right)^{\frac{\gamma}{\gamma-1}} \quad (3.29)$$

When p_r is above $p_{r,crit}$ the fluid velocity at the throttle is above the sonic velocity. With $\gamma = 1.4$, which corresponds to dry air, $p_{r,crit} = 0.5283$.

$A_{th}(\alpha)$ and $C_{D,th}(\alpha)$ in equation 3.27 is lumped together into something called effective area, $A_{eff}(\alpha) = A_{th}(\alpha) C_{D,th}(\alpha)$.

3.4.2 Experiments, Modeling and Identification

Some parameters of p_{im} model are unknown, and experiments are needed for the identification of them. The unknowns are the effective area of the main throttle and the volume of the intake manifold. These are necessary to be able to build a model that can be used to calibrate the PID controller for the p_{im} .

Measurements of p_{im} and \dot{m}_{at} with different main throttle plate angles, θ_{mt} , were done at different N , to be able to decide A_{eff} of the MT according to

$$A_{eff}(\alpha) = \frac{\sqrt{RT_{us}}}{\dot{m}_{at} p_{us} \Psi(p_r)} \quad (3.30)$$

which is 3.27 rewritten. See figure 3.21 for the result of two such mappings.

This was done with the engine running at steady-state at every measurement point, so that \dot{m}_{at} was the same as the air mass flow measured by the air mass flow meter. p_{us} and T_{us} are set to the value measured by the intake manifold pressure/temperature sensor with the engine shut off. Any influence on the pressure from for example the air filter is assumed to be zero. In reality there will be a small pressure drop over the air filter, but since there is no pressure sensor for the pressure before the throttle, or for that matter the ambient pressure, this is neglected.

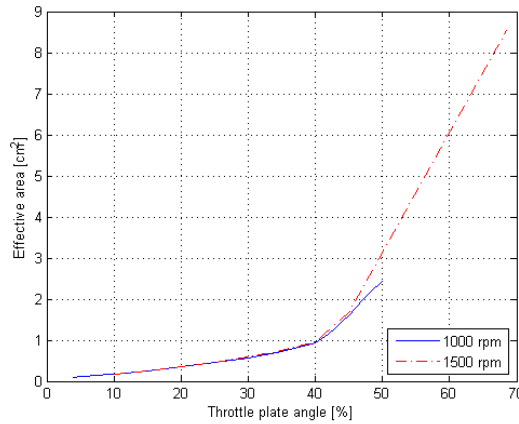


Figure 3.21. Map of effective area of the main throttle at two different engine speeds, 1000 and 1500 rpm. One can notice a knick on the curves where the flow velocity goes from sub-sonic to sonic. These maps are used in the modeling of the intake manifold pressure to get an identification of the volume of the intake manifold.

A simplified model of the intake manifold was created. The model was needed to get something to run the regulator on in simulation mode, so that some preliminary PID parameters could be found. The first task of the model was to identify the volume of the intake manifold, V_{im} . Since it is supposed to be a simple model with a simple task (parameterize the PID controller) several volume elements of the engine, like the SC, the IC and the IM, was lumped together into one control volume. V_{im} was identified as the "intake manifold volume", and used for the parameterization.

The model was simulated with different volume values and the result of each simulated step in p_{im} was compared to a measured step. The volume was considered identified when the curves of the modeled and the measured pressure coincided. This has been achieved in figure 3.23, and the volume found was 9 liters (or 0.009 m^3). This volume is larger than expected. One possible explanation is that there is sensor dynamics working on the intake manifold pressure sensor. This have not

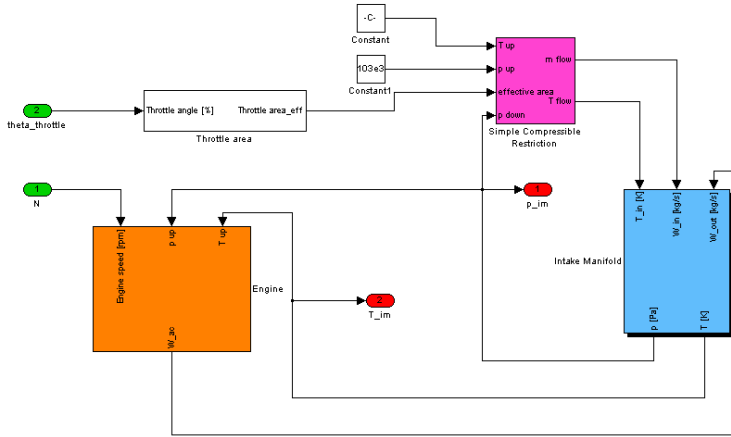


Figure 3.22. The intake manifold model used in the controller parameterization. The throttle area block is just the map from throttle angle to effective area seen in figure 3.21. The simple compressible restriction is a implementation of equation 3.27 ($T \text{ flow} = T \text{ up}$). The intake manifold implements equation 3.25. The engine, finally, implements equation 3.26. The temperature of the air before the throttle is set to a constant 25°C . The pressure upstream of the throttle is assumed to be the ambient pressure, and is set to 103 kPa (for that is what the intake manifold pressure sensor shows at ambient pressure).

been explored and should be followed up in future work.

With the unknown parameters of the model identified work on the controller can commence. It is implemented as a simple PID controller with anti-windup, as can be seen in figure 3.24. The input to the controller is the error between actual and desired p_{im} , and the output of the controller is the desired MT plate angle. The model and the controller are combined, and the controller feeds a MT angle to the model which in turn feeds a current p_{im} to the controller.

A time delay block was added between the controller and the "engine". This delayed the throttle plate angle signal for 0.05 seconds to emulate the time it takes for the throttle controller to move the throttle plate. This delay is actually a bit larger than it takes for the throttle plate to react to a change in desired throttle plate angle, as can be seen in figure 3.26. The time delay is the time identified as the rise time of the throttle, not the actual dead time. The model sets the actual value as soon as it gets it, therefore this added time delay was necessary (but may be too large). A smaller actual time delay, on the real system (the engine), may allow for some more aggressive PID parameters, where the parameters chosen in the next chapter is to guarantee stable operation. The controller is then run with different PID parameters until stable condition is found, starting out from identified, but sub-optimal, Ziegler-Nichols parameters. Throttle angle was chosen as

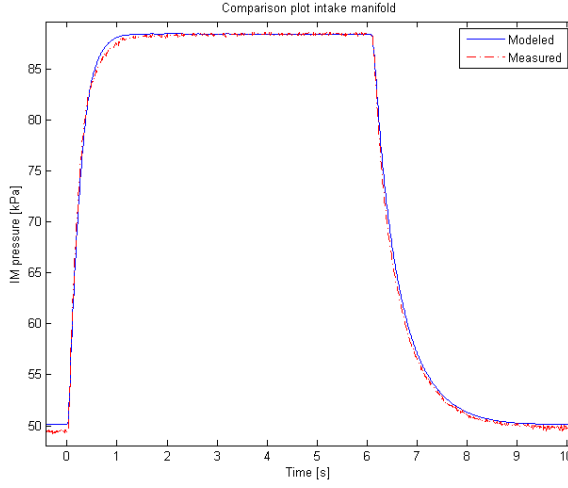


Figure 3.23. The intake manifold volume has been identified in this plot. It can be seen that the transient behavior of the model agrees rather good with the measured behavior. The volume that got good model behavior was found to be 9 liter, which is rather large.

output from the controller over throttle effective area because it exhibited a more linear behavior in respect to the intake manifold pressure (see figure 3.25).

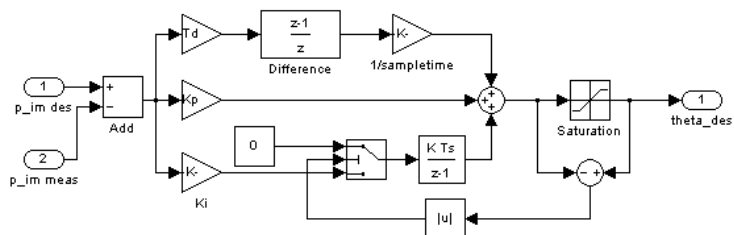


Figure 3.24. The p_{im} controller implementation. It is an ordinary PID controller with anti-windup that shuts off the integral part of the controller if the controller wants to output throttle angles smaller than 0% or larger than 100%.

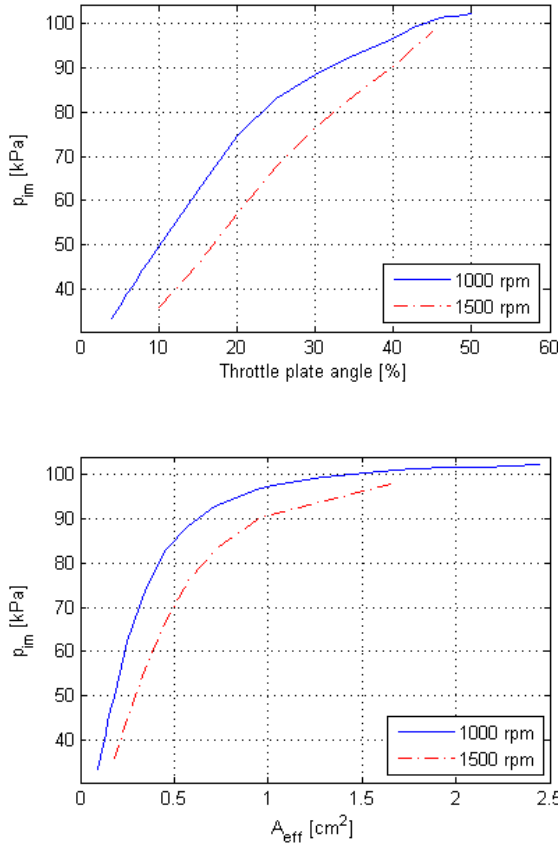


Figure 3.25. This plot is the basis of the decision to base the p_{im} controller around MT plate angle instead of throttle effective area. It can be seen that there is a better linear dependency of intake manifold pressure on MT plate angle than on MT effective area.

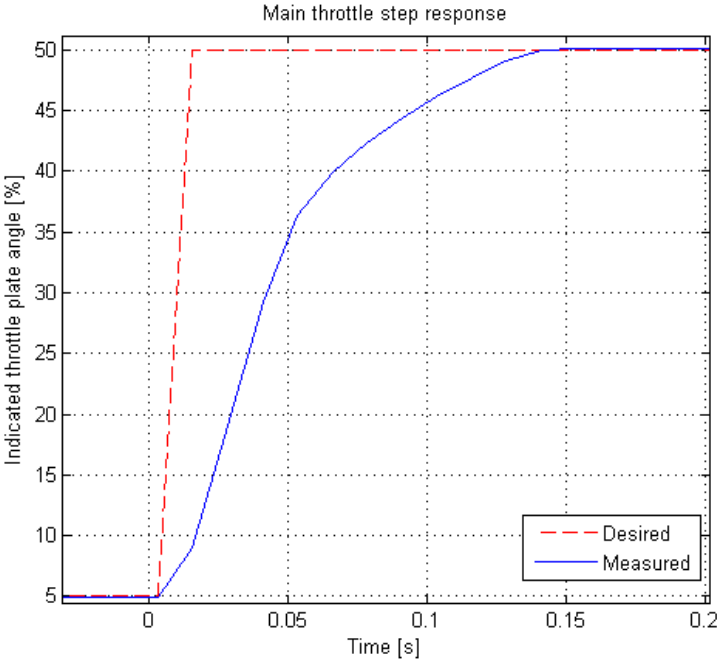


Figure 3.26. A typical step response for the main throttle. This plot shows the time delay and the rise time of the throttle, which the added time delay in the model and controller combination is based upon.

3.4.3 Resulting Controller

The controller is made up by the parts in figure 3.24. The parameters K_p , T_i and T_d was first approximated with the Ziegler-Nichols set of parameterization rules, which with a dead time, L , of 0.045s and a maximum slope of 168 kPa/s resulted in $K_p = 3.18 \cdot 10^{-3}$ (the controller works in Pa not kPa), $T_i = 0.09$ and $T_d = 0.0225$, (identified from figure 3.26). This produced a very bad controller. The parameters were then hand-tuned to something that actually worked. The resulting PID parameters became $K_p = 7 \cdot 10^{-4}$, $T_i = 666$ and $T_d = 0.05 \cdot 10^{-4}$ at 1000rpm and $K_p = 4 \cdot 10^{-4}$, $T_i = 666$ and $T_d = 0.05 \cdot 10^{-4}$ at 1500rpm. The controller is not run on the engine for final validation due to the limitations in chapter 1.2.

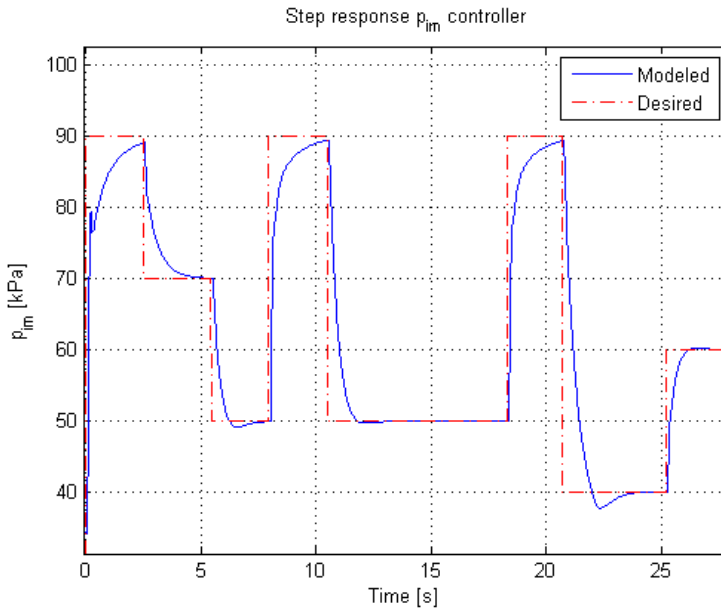


Figure 3.27. A plot showing the reference following of the p_{im} controller during steps in desired pressure when the engine is modeled running at 1000 rpm. There are some overshoots during down steps. The rise time is acceptable. The controller is made to be very stable during rather long delays, just to be sure there is no oscillations introduced in the system.

The controller and model are fed a reference desired p_{im} curve that the controller is supposed to follow as closely as possible. The result of one such experiment can be seen in figure 3.27. What can be seen in the plot is that there are small "over-shoots" on the down steps and that the controller manages to get the actual, modeled, pressure value to the right value in approximately 2-3 seconds on steps

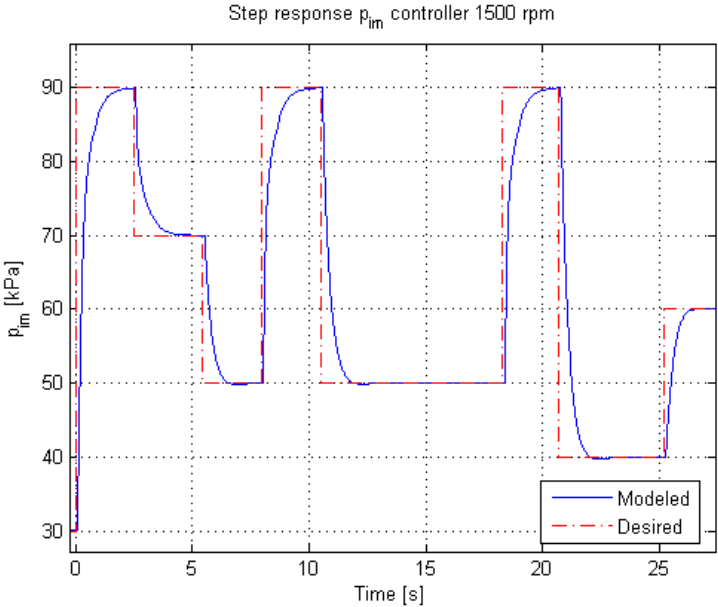


Figure 3.28. A plot showing the reference following of the p_{im} controller during steps in desired pressure when the engine is modeled running at 1500 rpm. The step responses here look better than the ones at 1000 rpm. It seems that the higher air mass flow, as a result of the higher engine speed, together with the different PID parameters makes for a slightly better controller.

in desired value.

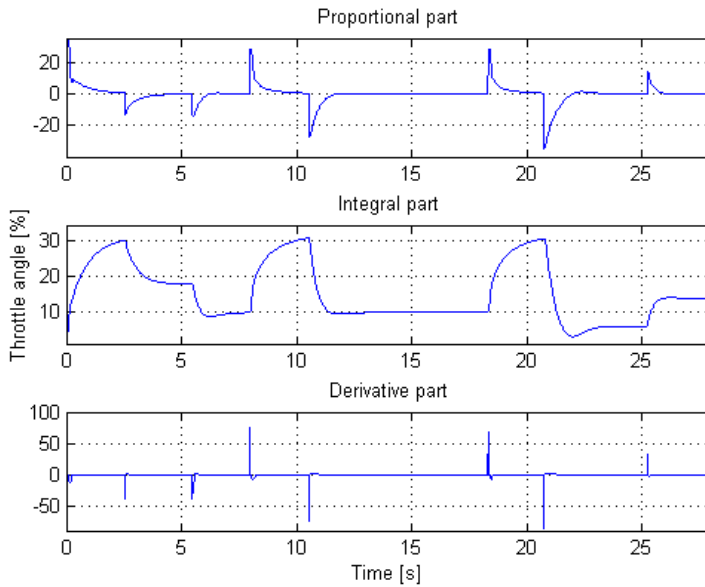


Figure 3.29. The same step responses as in figure 3.27, but this plot shows the internal workings of the controller. The first plot shows the influence of the proportional part of the controller. The second plot shows the influence of the integrating part and the third and final plot shows the derivative part of the controller.

Chapter 4

Fuel

The fuel system for the engine consists of a fuel pump, a fuel rail with a pressure regulator and five fuel injectors. The fuel pump just pumps fuel from the fuel tank to the fuel rail constantly, and the excess fuel is transported back to the fuel tank in a return pipe. The pressure regulator on this particular engine is a strict mechanical regulator, that is connected to the intake manifold and thus the fuel pressure is related to the intake manifold pressure in some unknown way. The proposal that the pressure difference between the fuel pressure in the fuel rail, p_{fuel} , and the intake manifold pressure, p_{im} , is kept constant will be tested, and if proven false some sort of black box model will be developed. The fuel injectors injects fuel sequentially into the intake runners of each individual cylinder.

4.1 Fuel Injectors

The fuel injectors consist of a solenoid, a valve and a spring. The actuator signal is the current through the solenoid, that opens the valve electromagnetically against the spring resistance. The spring closes the valve when the current is shut off.

4.1.1 Theory

One of the sensors present on the engine is the air mass flow meter, MAF. This is used to measure the air mass flow, \dot{m}_a , that passes the sensor, and the sensor is placed before the main throttle on this engine. Another sensor is the Universal Exhaust Gas Oxygen (UEGO) sensor that is located in the exhaust system. The UEGO measures the oxygen fraction in the exhaust gas, which is called the lambda value, λ . At steady-state the \dot{m}_a before the throttle, measured by the MAF, is equal to that entering the cylinders, $\dot{m}_{a,cyl}$ and this value, together with the measurement of the λ -value, can be used to calculate the amount of fuel flow that also enters the cylinders, according to

$$\dot{m}_a = \dot{m}_{ac} \quad (4.1)$$

and

$$\lambda_{cyl} = \frac{\dot{m}_{ac}}{\dot{m}_{fc}} \frac{1}{(A/F)_s} \quad (4.2)$$

where \dot{m}_{fc} the fuel flow into the cylinders and $(A/F)_s$ is the stoichiometric air to fuel ratio, which in this work is approximated with the value of gasoline ($= 14.7$).

Also at steady-state the amount of fuel injected, \dot{m}_{fi} , is the same as the fuel flow into the cylinders, \dot{m}_{fc} , according to

$$\dot{m}_{fi} = \dot{m}_{fc} \quad (4.3)$$

and the λ in the cylinders, λ_{cyl} , is also the same as the measured λ , λ_{meas} .

$$\lambda_{cyl} = \lambda_{meas} \quad (4.4)$$

The relation between the fuel flow and the mass of fuel injected is

$$\dot{m}_{fi} = \frac{N n_{cyl}}{n_r} m_{fi} \quad (4.5)$$

where N is the engine speed, n_{cyl} is the number of cylinders and n_r is the number of revolutions per cycle.

In Eriksson and Nielsen (2007) it is shown, by using equations for turbulent flow for an incompressible fluid, that the amount of fuel injected, m_{fi} , is proportional to the square root of the pressure difference between p_{fuel} and p_{im} , Δp , over the injector valve and the opening time t_{inj} , according to

$$m_{fi} = c\sqrt{\Delta p}(t_{inj} - t_0) \quad (4.6)$$

where t_{inj} is the injection time and t_0 is the opening and closing time of the valve.

One possible way to simplify the model is to approximate the pressure difference over the fuel injector to be constant, and if the resulting pressure constant is inserted into the constant c above the following is given

$$m_{fi} = c_1(t_{inj} - t_{0,mod}) \quad (4.7)$$

where $t_{0,mod}$ is a model of t_0 . This model will be evaluated below.

Two different ways to model $t_{0,mod}$ are evaluated. The first is that t_0 is a simple constant, which will be called $t_{0,mod1}$, and the second is that $t_0(U_{batt})$ is approximated as a linear dependency on u_{bat} , which will be called $t_{0,mod2}$, like

$$t_{0,mod1} = c_{t_{0,mod1}} \quad (4.8)$$

$$t_{0,mod2} = c_{1,t_{0,mod2}} U_{bat} + c_{2,t_{0,mod2}} \quad (4.9)$$

According to literature, see for example Eriksson and Nielsen (2007), the opening and closing time of the injector valve depends on the battery voltage U_{batt} since

the opening of the valve depends on the current through the solenoid and thus the voltage. So the extension that t_0 depends on U_{batt} is reasonable. The linear model chosen was a first try.

c_1 in equation 4.7 and $c_{t_0,mod1}$ in equation 4.8 or $c_{1,t_0,mod2}$ and $c_{2,t_0,mod2}$ in equation 4.9 are determined using the least square method.

The injector model found is then inverted to be able to calculate the injection time from the desired fuel flow, according to

$$t_{inj} = \frac{m_{fi}}{c_1} + t_{0,mod} \quad (4.10)$$

where the t_{inj} -models corresponding to the t_0 -models $t_{0,mod1}$ and $t_{0,mod2}$ are named $t_{inj,mod1}$ and $t_{inj,mod2}$.

4.1.2 Experiments

The steady-state map of the engine presented in 3.1.2 was used to determine the parameters of the fuel injector model. To improve the injector model dynamic fuel step measurements at different load points were done, where everything except the injection time was kept as close as constant as possible.

4.1.3 Results Assuming $p_{fuel} - p_{im} = \text{constant}$

Using the steady-state map of the engine, with the sensor values from the ‘production’ sensors on the engine, the amount of fuel injected was calculated using the equations (4.1) - (4.5). The calculated value of \dot{m}_{fi} was then used together with the least square method to determine the parameters of the injector model.

The modeled values are then compared to the real values, by calculating the injection time by the inverted models (4.10), $t_{inj,mod1}$ and $t_{inj,mod2}$, using the mapped value of \dot{m}_{fi} and equation 4.5, in table 4.1.

The mean absolute value of the relative errors for the two tested models are 1.49 % for $t_{inj,mod1}$ and 1.57 % for $t_{inj,mod2}$, and the variance of the errors are 3.29 % for $t_{inj,mod1}$ and 3.91 % for $t_{inj,mod2}$. It can be seen that $t_{inj,mod1}$, which is the simple model without a t_0 dependency on U_{batt} , is better than the extended model $t_{inj,mod2}$. $t_{inj,mod1}$ has both a smaller mean absolute relative error as well as a smaller spread of the error (variance) than $t_{inj,mod2}$. $t_{inj,mod2}$ should perform better, since it is of similar structure as $t_{inj,mod1}$ but with one added parameter. The reason it is not may be related to figure 4.2, where it can be clearly seen that there is no correlation between the measured U_{batt} values and t_0 . The added correlation error introduced is what makes $t_{inj,mod2}$ worse than $t_{inj,mod1}$.

t_{inj} Calculated [%]	$t_{inj,model1}$ Rel. error [%]	$t_{inj,model2}$ Rel. error [%]	$t_{inj,model3}$ Rel. error [%]	$t_{inj,model4}$ Rel. error [%]
2.96	-0.13	-0.79	-5.44	-1.60
4.30	-1.76	-2.24	-2.50	-0.24
6.29	3.13	2.45	3.70	2.48
3.07	1.74	1.73	-1.54	0.60
4.69	-1.90	-2.45	-1.69	-0.51
6.33	-0.53	-0.66	-0.29	-0.94
3.21	2.10	3.16	0.28	1.41
4.75	-1.07	-0.45	-0.24	0.25
6.35	-1.01	-0.18	-0.74	-1.40
Mean abs error [%]	1.49	1.57	1.82	1.05
Variance [%]	3.29	3.91	5.95	1.79

Table 4.1. Comparing the results of the different t_{inj} -models at different engine speeds and p_{im} steady-state points. It can be seen that $t_{inj,model4}$ gives the smallest absolute mean error and the smallest variance of relative errors.

4.1.4 Results Assuming $p_{fuel} - p_{im} \neq \text{constant}$

The resulting mean absolute value and variance of the errors of models $t_{inj,model1}$ and $t_{inj,model2}$ were deemed not good enough, so further analysis took place. The pressure difference in equation (4.6) was suspected not to be constant. The first experiments to analyze this was to insert an unknown variable, X , into the equation representing p_{fuel} , since there was no reliable sensor that measured this pressure. The equation is as follows

$$m_{fi} = c\sqrt{X - p_{im}}(t_{inj} - t_0) \quad (4.11)$$

Attempts to manually tune this unknown variable, and calculate new values of c and t_0 , resulted only in values of t_{inj} with a mean absolute error larger or equal to the variance of the original simple model, with equality only when X was very large (i.e. when the pressure difference between p_{fuel} and p_{im} was close to constant for the different p_{im}). This model extension was rejected, since p_{fuel} obviously is not kept constant.

The next step was to calculate, using the least square method, separate values of the simple model $t_{inj,model1}$ for each load point using data from injection time step responses. The result is presented in tables 4.2 and 4.3.

There seems to be a clear dependency of p_{im} and a weaker dependency of N on the constant c_1 .

The constant c_1 was plotted with respect to the square root of p_{im} . This model was chosen because the original model, see (4.6), was dependent on the square root

p_{im} [kPa] \ N [rpm]	1000	1500	2000	2500	3000
50	30e-004	31e-004	31e-004	30e-004	30e-004
75	31e-004	33e-004	34e-004	32e-004	32e-004
100	35e-004	35e-004	36e-004	35e-004	34e-004

Table 4.2. c_1 constants calculated for each different load point. It can be seen that there seems to be a relation between p_{im} and the constant value, where the value grows when p_{im} gets larger. There seems to be no dependency of N on c_1 .

p_{im} [kPa] \ N [rpm]	1000	1500	2000	2500	3000
50	0.71	0.66	0.54	0.44	0.46
75	0.46	0.63	0.77	0.48	0.44
100	0.70	0.68	0.79	0.58	0.54

Table 4.3. t_0 constants calculated for each of the different load point [ms]. No clear dependency can be seen on neither p_{im} or N .

of Δp . See figure 4.1, where a linear approximation done with the least square method also is plotted. This is done to illustrate the dependency, which seems to be a linear dependency. From the plot the conclusion is that there is a clear dependency of $\sqrt{p_{im}}$ on c_1 .

Table 4.3 reveals no evident correlation between t_0 and either p_{im} nor N . A possible correlation to the battery voltage, U_{batt} , was examined in figure 4.2, and no apparent correlation can be seen. So t_0 was implemented as a constant value, and calculated as the mean value of all the t_0 -values in table 4.3. The resulting t_{inj} -model was

$$t_{inj,mod3} = \frac{m_{fi}}{c_{1,mod}} + t_0 = \frac{m_{fi}}{c_{1,\sqrt{p_{im}}} \cdot \sqrt{p_{im}} + c_{1,const}} + t_{0,const} \quad (4.12)$$

where $c_{1,mod}$ is the linear approximation of c_1 from figure 4.1 and $t_{0,const}$ is the mean value of table 4.3.

The steady-state data set, the same as before, is used to test the model and the result is presented in table 4.1 together with the result from the other proposed models. The mean absolute value of the relative errors for $t_{inj,mod3}$ is 1.82 % and the variance of errors is 5.95. This is worse than the original simple model $t_{inj,mod1}$. This prompted a more thorough examination of the possible correlation between t_0 and p_{im} , done in figure 4.3. This revealed a possible, weak, correlation.

Since the way to identify the model variables are based upon equation 4.7 there is a slight problem when trying to identify any t_0 -model when c_1 is more complex than a constant, since you have to divide with c_1 to get t_0 . If a c_1 -model like

$$c_{1,mod} = c_{1,\sqrt{p_{im}}} \cdot \sqrt{p_{im}} + c_{1,const} \quad (4.13)$$

is chosen, the identifiable variables with $t_{0,mod}$ would be

$$c_{1,mod} \cdot t_{0,mod} = c_{1,\sqrt{p_{im}}} \cdot \sqrt{p_{im}} \cdot t_{0,mod} + c_{1,const} \cdot t_{0,mod} \quad (4.14)$$

To get $t_{0,mod}$ as a function of $C_1\sqrt{p_{im}} + C_2$, $c_{1,mod} \cdot t_{0,mod}$ was chosen to be $c_{1,t_0,mod} \cdot p_{im} + c_{2,t_0,mod} \cdot \sqrt{p_{im}}$. This would result in

$$\begin{aligned} t_{0,mod} &= \frac{c_{1,mod} \cdot t_{0,mod}}{c_{1,mod}} = \frac{c_{1,t_0,mod} \cdot p_{im} + c_{2,t_0,mod} \cdot \sqrt{p_{im}}}{c_{1,\sqrt{p_{im}}} \cdot \sqrt{p_{im}} + c_{1,const}} = \\ &= \frac{c_{1,t_0,mod} \cdot p_{im}}{c_{1,\sqrt{p_{im}}} \cdot \sqrt{p_{im}} + c_{1,const}} + \frac{c_{2,t_0,mod} \cdot \sqrt{p_{im}}}{c_{1,\sqrt{p_{im}}} \cdot \sqrt{p_{im}} + c_{1,const}} = \\ &= A_1\sqrt{p_{im}} + A_2 + \frac{A_3}{\sqrt{p_{im}}} + \dots + B_1 + \frac{B_2}{\sqrt{p_{im}}} + \frac{B_3}{p_{im}} + \dots \end{aligned} \quad (4.15)$$

where A_n is the result of the polynomial division from the $c_{1,t_0,mod} \cdot p_{im}$ part and B_n is the result of the polynomial division from the $c_{2,t_0,mod} \cdot \sqrt{p_{im}}$ part. In the identification it was found that the constant $c_{1,const}$ was ten times smaller than the constant $c_{1,\sqrt{p_{im}}}$. This means that both A_{n+1} and B_{n+1} will be ten times smaller than their respective predecessor A_n and B_n . This in turn means that both $\lim_{n \rightarrow \infty} A_n$ and $\lim_{n \rightarrow \infty} B_n = 0$ and that the most significant parts of equation 4.15 is $A_1\sqrt{p_{im}} + B_1$ which is the desired model structure of $t_{0,mod}$.

The following model $t_{inj,mod4}$ is presented

$$t_{inj,mod4} = \frac{m_{fi}}{c_{1,mod}} + t_{0,mod} = \frac{m_{fi} + c_{1,t_0,mod} \cdot p_{im} + c_{2,t_0,mod} \cdot \sqrt{p_{im}}}{c_{1,\sqrt{p_{im}}} \cdot \sqrt{p_{im}} + c_{1,const}} \quad (4.16)$$

where $c_{1,mod}$ is a linear approximation of c_1 similar, but not identical, to the linear approximation in figure 4.1.

Using the original steady-state data set, the parameters of this new model $t_{inj,mod4}$ is calculated using the least square method. The resulting model was then compared in a similar way to the three previous models, and the mean absolute value of the model $t_{inj,mod4}$ is 1.05 % and the variance of the relative error is 1.79 %. This is an improvement over the original simple model $t_{inj,mod1}$. The result is presented in table 4.1.

Which Model to Choose

As can be seen in table 4.1 the $t_{inj,mod4}$ is the model that is performing best of the four tested models. It is so much better than the simple model, $t_{inj,mod1}$, that the added degree of complexity is deemed worthwhile, and the model $t_{inj,mod4}$ is the one implemented in the control system.

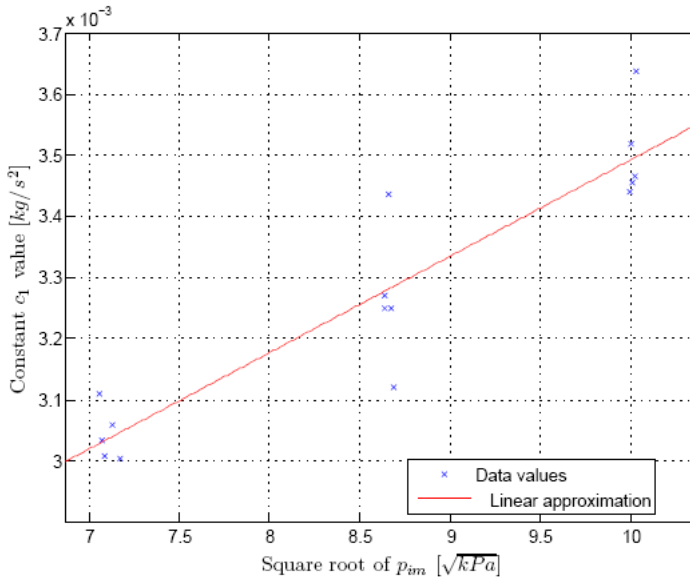


Figure 4.1. Using the dynamic measurements, the step responses from steps in t_{inj} , the parameters of the simple t_{inj} model $t_{inj,model1}$ was calculated using the least square method for each load point. A linear approximation done with the least square method based on the data values is also plotted, to visualize the dependency. There is a clear dependency of $\sqrt{p_{im}}$ on c_1 . The linear approximation plotted here is the basis of the c_1 -modeling in $t_{inj,model3}$.

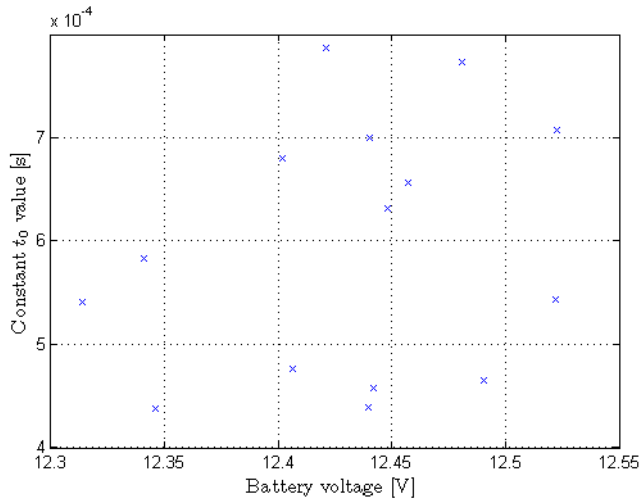


Figure 4.2. Using the dynamic measurements, the step responses from steps in t_{inj} , the parameters of the simple t_{inj} model $t_{inj,mod1}$ was calculated using the least square method for each load point. Here the value of t_0 are presented as a function of the battery voltage U . No correlation between U and t_0 can be seen.

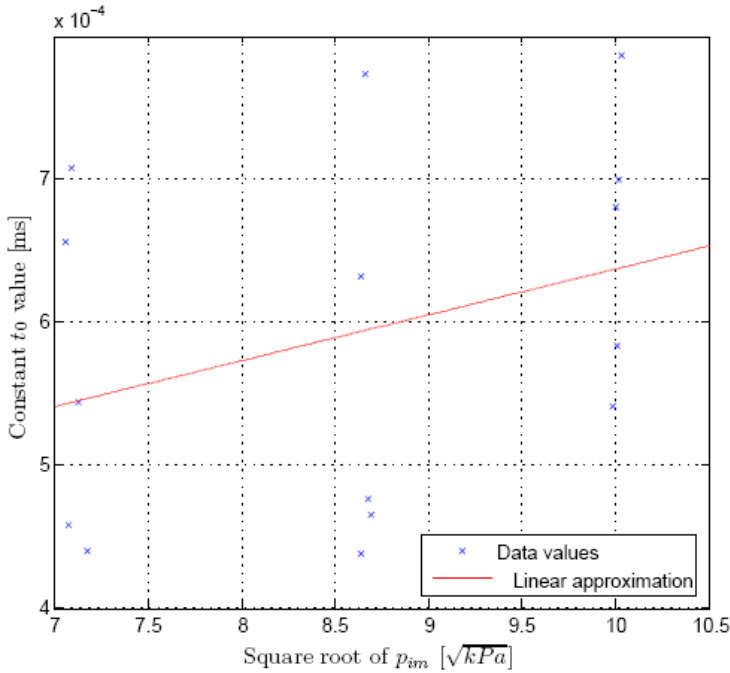


Figure 4.3. Using the dynamic measurements, the step responses from steps in t_{inj} , the parameters of the simple t_{inj} model $t_{inj,mod1}$ was calculated using the least square method for each load point. Here the constant values t_0 are presented as a function of $\sqrt{p_{im}}$. A possible weak correlation can be seen in the figure. A linear approximation done with the least square method based on the data values is also plotted, to visualize the dependency. This possible weak correlation was the basis of the decision to make $t_{inj,mod4}$.

4.2 Open Loop Part of the Fuel Controller

The feed forward part of the fuel controller is necessary to achieve the fast transient response speed needed to meet current emission regulations. It is also desired to have an fast responsive engine that feels right to a driver. Feed forward can, on the other hand, not sufficiently accurate handle engine uncertainties, like for example sensor aging. It is also difficult to get a feed forward system to run the engine in the very narrow λ -band necessary for optimum catalyst operation.

4.2.1 Theory

The feed forward part uses $\eta_{vol,mod3}$ from chapter 3.1.1

$$\eta_{vol,mod3} = k_0 + k_1N + k_2p_{im} + k_3Np_{im} \quad (4.17)$$

to calculate a modeled value of η_{vol} from the sensor values of the engine speed, N , and intake manifold pressure, p_{im} . This calculated value of η_{vol} is then used together with the inverse of equation (3.3) according to

$$\dot{m}_{ac} = \frac{\eta_{vol}p_{im}V_d n_{cyl}N}{RT_{im}n_r} \quad (4.18)$$

and the sensor values of the engine speed to calculate the amount of air entering the cylinders. This is then used to calculate the amount of fuel to enter in the cylinders according to

$$\dot{m}_{fc} = \frac{\dot{m}_{ac}}{\lambda_{ref}} \frac{1}{(A/F)_s} \quad (4.19)$$

where λ_{ref} is the desired lambda ratio in the cylinders. This is most often equal to 1 but can be different if for example the combustion gases needs to be cooled down with fuel excess to avoid knock.

4.2.2 Results

To evaluate the open loop model the η_{vol} model was combined with the injector model $t_{inj,mod4}$ presented earlier, and the resulting calculated values of t_{inj} was compared to values from the steady-state map. The result is presented in table 4.4.

The mean absolute error of the modeled t_{inj} value is 1.01 % and the variance of the relative error is 1.62 %. This is better than the model of the injector, $t_{inj,mod4}$, seen in table 4.1. This seems to indicate that the respective errors of that model and the model of volumetric efficiency is canceling each other out somewhat when the combined model is tested with the small engine map. This is a good result, which gives hope that the resulting complete model will work on the actual engine.

Map value [ms]	Model value [ms]	Relative error [%]
2.96	2.92	-1.45
4.30	4.33	0.72
6.29	6.40	1.82
3.07	3.05	-0.52
4.69	4.70	0.22
6.33	6.30	-0.39
3.21	3.37	1.99
4.75	4.77	0.33
6.35	6.25	-1.61

Table 4.4. Comparison between t_{inj} values of the engine map and calculated t_{inj} values based on the open loop and injector models. The mean absolute error of the modeled t_{inj} value is 1.01 % and the variance of the relative error is 1.62 %. This is better than the model of the injector, $t_{inj,mod4}$, seen in table 4.1. This seems to indicate that the respective errors of that model and the model of volumetric efficiency is canceling each other out somewhat when the combined model is tested with the small engine map.

4.3 Wall Wetting

A part of the injected fuel does not reach the cylinders directly, it is instead deposited on the intake manifold walls, either as film or as puddles (Eriksson and Nielsen, 2007). This fuel then evaporates over time, which explains, in part, the dynamic response in exhaust lambda to a change in fuel injection time. The other major part is the sensor dynamics, mainly from the lambda sensor.

4.3.1 Theory

One popular formulation of the fuel puddle dynamics is Aquino (1981)), where a part χ of the flow of injected fuel, \dot{m}_{fi} , is deposited on the walls of the intake manifold and forms a fuel puddle. The rest of the injected fuel, $(1 - \chi)$, is mixed with the air and injected into the cylinders. The fuel in the fuel puddle evaporates over time with a time constant τ_{fp} , and this evaporation is also mixed with the air. The evaporation of the fuel in the fuel puddle is assumed to be proportional to the area of fuel deposited, which in turn is assumed to be proportional to the mass of fuel deposited due to the thinness of the fuel puddles (the fuel is spread out). This gives

$$\frac{dm_{fp}}{dt} = \chi \dot{m}_{fi} - \frac{1}{\tau_{fp}} m_{fp} \quad (4.20)$$

This gives that the amount of fuel flow entering the cylinders, \dot{m}_{fc} , is the sum of the part of the injected fuel not deposited on the walls and the amount of fuel that evaporates from the mass of fuel already deposited on the walls, like

$$\dot{m}_{fc} = (1 - \chi) \dot{m}_{fi} + \frac{1}{\tau_{fp}} m_{fp} \quad (4.21)$$

4.3.2 Experiments

Steps in amount of injected fuel, which means steps in injection time, t_{inj} , at normally aspirated conditions have been done where all other variables were kept as close to constant as possible. These dynamic measurements were then used to try to identify the amount of fuel deposited on the intake manifold walls as fuel puddles, χ , and the time constant for the evaporation of fuel from the fuel puddles, τ_{fp} . A model of the system, the engine with lambda sensor, was implemented in Simulink. This was done to be able to tune the wall wetting model parameters. The model uses the current N , p_{im} and \dot{m}_a , which are assumed to be at steady-state, together with the model in equation 4.21 to model the effects of wall wetting. It has been assumed that the steady-state variables N and \dot{m}_a are constants, with the same value as the mean value of the measurements. This is done only to get a cleaner output from the model, because the measured data for those variables are rather noisy so a mean value is "better". The parameters χ and τ_{fp} , which are included in the model, are then hand tuned to achieve comparable results between the modeled and the measured lambda values. The model can be seen in figure 4.4 and a typical result from the model, with and without tuned wall wetting parameters, can be seen in figure 4.6.

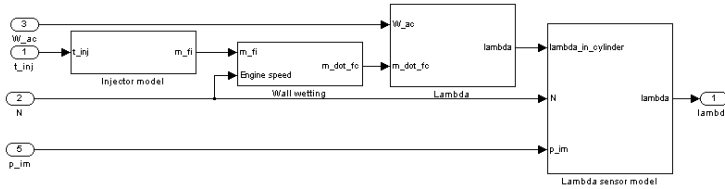


Figure 4.4. Simplified engine model for fuel puddle tuning. It consists of an injector model, a wall wetting model, a lambda value model and a lambda sensor model. The injector model is based on the findings in chapter 4.1. The wall wetting model can be seen in more detail in figure 4.5. The lambda model is a very basic cylinder model, it basically just calculates the current lambda value from the air mass flow meter measurement and the calculated injected fuel flow into the cylinders. The lambda sensor model uses the sensor time delay and the sensor time constant to simulate the UEGO sensor used in the engine cell. This is done with a first order sensor model according to description in Eriksson and Nielsen (2007). The inputs to the model are the air mass flow, the injection time, the engine speed and the intake manifold pressure. The output is the simulated lambda value. The air mass flow, the engine speed and the intake manifold pressure are assumed to be constant, at steady-state, in the experiment, due to the sensors being rather noisy. This assumption is a simplification, but it produces a much nicer lambda value in noise respect.

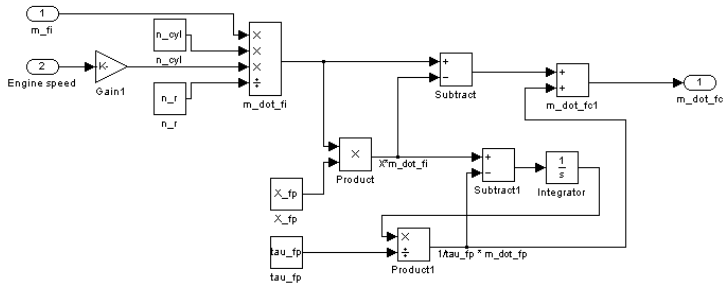


Figure 4.5. Wall wetting model implemented in Simulink. Model inputs are fuel mass to be injected and engine speed, which translates to fuel flow to be injected. This is then modeled using equations 4.20 and 4.21 to get the fuel flow into the cylinders. The "constants" X_{fp} and τ_{fp} in the model are χ and τ_{fp} which are set each time before running the model in Matlab. They both are incrementally changed, by hand before each running, to tune the model so that finally the output of the model is comparable to the measurement.

4.3.3 Results

After hand tuning the wall wetting constants, fraction of fuel injected that is deposited χ and fuel vaporisation time constant τ_{fp} , the following value maps were found to achieve the best curve fits

p_{im} [kPa] \ N [rpm]	750	1000	1500	2000
50	0.5	0.5	0.35	0.15
75	0.5	0.4	0.15	0.10
100		0.35	0.15	0

Table 4.5. Fraction of fuel injected deposited on the intake manifold walls (χ). It can be seen that χ is larger at low engine loads, which is consistent with the available literature (see for example Eriksson and Nielsen (2007)).

p_{im} [kPa] \ N [rpm]	750	1000	1500
50	0.4	0.3	0.2
75	0.4	0.25	0.2
100		0.2	0.2

Table 4.6. Time constant of fuel vaporisation (τ_{fp}). At higher engine speeds τ_{fp} is set to 0.2. It can be seen that the vaporisation is lower, the time constant is larger, at lower engine loads. This can be explained by the lower temperature and lower air flow at lower loads.

An example of a tuned model curve at a load point with clear wall wetting is shown in figure 4.6, and there are more plots in appendix A. It can be seen that the tuned wall wetting model produces much more similar results to the actual measured lambda value then the model without wall wetting.

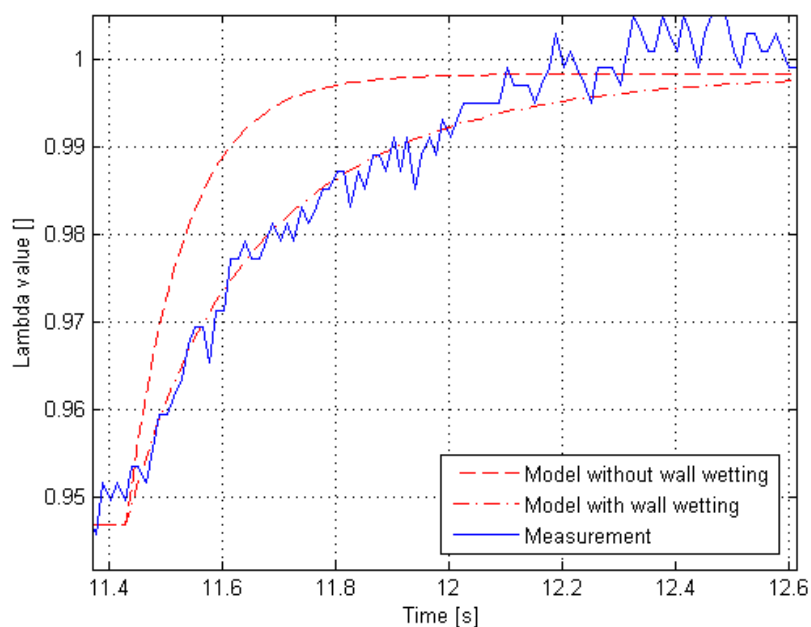


Figure 4.6. Comparison between a model with and a model without wall wetting compensation at 1000rpm and 50kPa intake manifold pressure. It can be seen that the tuned wall wetting model produces much more similar results to the actual measured lambda value then the model without wall wetting.

4.4 Closed Loop Part of the Fuel Controller

The closed loop part of the fuel controller is needed to adjust inaccuracies in the forward loop part, as well as to compensate for sensor and actuator aging. A lambda sensor measures the amount of oxygen in the exhaust gases and the closed loop controller compensates the amount of injected fuel to achieve the desired oxygen to fuel ratio.

4.4.1 Theory

The closed loop part of the fuel injection control system is implemented as a PI controller. The basic principle is that the output of the process is compared to a desired setpoint. The error between the set point and the actual output value is then used to influence the process so that, hopefully, the right output value is reached. In this case the process is the engine. The output is the lambda value of the exhaust gases of the engine. The set point is the desired lambda value that the engine should run at. A PI controller uses the error to influence the process with a proportional and an integrating part, where the proportional part uses the current error and the integrating part uses the sum of recent errors. The contributions of the two parts are then added together. A PI controller is used over a PID controller due to the fact that the lambda measurement signal is very noisy. This noise would, with a small sample rate, produce very frequent large changes in the derivative of the lambda signal. The lambda measurement noise can be seen in for example figure 4.6. A filter of the lambda measurement could be implemented, but that would produce further delays due to phase shifts in that filter.

The parameterization of the PI controller will be done by performing step response experiments at different engine speeds, N , and intake manifold pressures, p_{im} . The output of the controller is the injection time, t_{inj} sent to the fuel injectors. This changes the amount of fuel injected. There is a transport delay from when the injection time is changed to when the change is first seen by the measuring lambda sensor, since the air and fuel needs to travel through the engine to where the sensor is located in the exhaust system (and there is also a slight sensor delay in the λ -sensor). A typical step and step response experiment can be seen in figure 4.7, where the time delay also can be seen. This time delay will be important in the PI parameterization, in that it is the input for several of the different parameterization rules tested here. Everything but the injection time is kept as close to constant as possible, this includes the throttle position, the engine speed and the intake manifold pressure. The air mass flow may vary some due to changes in the cooling of the air in the intake manifold due to the changes in injected fuel, but the variation is considered small and can therefore be neglected.

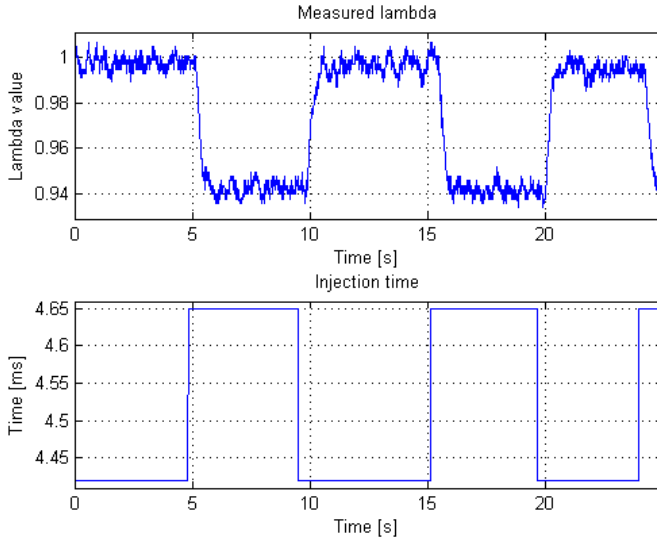


Figure 4.7. Step response measurement of the λ value, and the steps in t_{inj} , when $N = 1000$ rpm and $p_{im} = 75$ kPa. This is an example of a typical step response experiment done to parameterize the lambda PI controller. Everything but t_{inj} is kept as close to constant as possible, this includes the throttle position, N and p_{im} . The air mass flow may vary some due to changes in the cooling of the air in the intake manifold due to the changes in injected fuel, but the variation is considered small and can therefore be neglected. The possible perceived ramping of t_{inj} at the steps is due to the 80 Hz sample frequency. In reality the steps are instantaneous.

Three different sets of parameterization rules will be tested and evaluated, namely Ziegler-Nichols, Cohen-Coon and Åström-Hägglund. These will be used and compared to each other. The different parameterization rules are presented in table 4.7.

Parameter	Ziegler-Nichols	Cohen-Coon	Åström-Hägglund
K_p	$\frac{P_{pid}}{b_{pid}L_{pid}}$	$\frac{1}{K_{pid}} \frac{T_{pid}}{L_{pid}} (0.9 + \frac{L_{pid}}{12T_{pid}})$	$0.2 \frac{T_{pid}}{K_{pid}L_{pid}}$
T_i	$3L_{pid}$	$L_{pid} \frac{30 + 3\frac{L_{pid}}{T_{pid}}}{9 + 20\frac{L_{pid}}{T_{pid}}}$	$0.7T_{pid}$

Table 4.7. Parameterization rules sets used in this work. The Ziegler-Nichols and Åström-Hägglund sets are found in Reglerteknik (2008) and the Cohen-Coon in Cohen and Coon (1952). P_{pid} is the perturbation in t_{inj} , which in this case is not the actual change in t_{inj} value but rather the percentage change. K_{pid} is the process gain, which is $\frac{\text{change in manipulated value}}{P_{pid}}$. L_{pid} is the process dead time. T_{pid} is the time constant.

The Åström-Hägglund parameter functions are valid when the normalized time delay, τ , 4.22, is between 0.2 and 0.7 (Reglerteknik, 2008). τ is defined as

$$\tau = \frac{L_{pid}}{L_{pid}T_{pid}} \quad (4.22)$$

The above mentioned parameters can be identified in the step response experiments. In figure 4.8 one step of one such experiment has been enhanced by a filter, so that the different parameters can be easily seen. The filtering is done with a zero-phase forward and reverse low-pass filter. The low-pass filter is a Butterworth filter of second order with a cutoff frequency of 0.17, and it is implemented by the Matlab function "butter". The actual filtering is done with the Matlab function "filtfilt". The maximum slope of the step response is found. This has been done through derivation of the filtered lambda curve. Due to noise the identification of the maximum slope value is only done in an set interval where the actual step response is taking place (between 0.4 and 1 second in the figure). When the maximum slope b is found the dead time L_{pid} is easily found by the approximation in equation 4.23. The time constant T_{pid} is identified as the time when the lambda value has risen 63% of the difference between λ_{upper} and λ_{lower} . A semi-automatic Matlab function that does the identifying has been used, and it is presented in appendix B. The intervals where the mean values of the upper and lower lambda values, as well as the maximum slope, has been added manually in the function.

$$L_{pid} = t_{half} - \frac{\lambda(t_{half}) - \lambda_{lower}}{b} \quad (4.23)$$

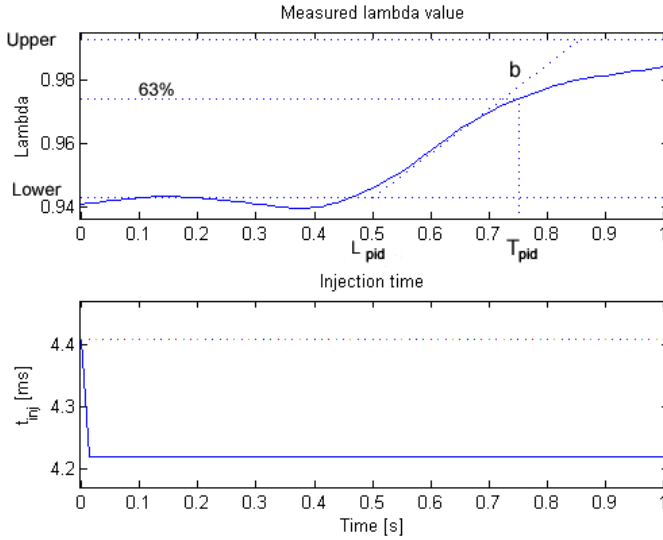


Figure 4.8. Step response measurement of lambda value, and the step in injection time. The lower and upper bounds of the lambda response, as well as the 63 % rise value of lambda, are also marked. The line with slope b , which is the steepest slope in the step response, is plotted. And L_{pid} , the process dead time, which is the time value where this line curve cuts the lower bound of the step response is marked. T_{pid} , the process time constant, is also marked in the plot. The step in t_{inj} is plotted separately. The percentage of the change in t_{inj} is the output of the lambda controller, and the input when calculating the parameters of the PI controller.

4.4.2 Identification

Three sets of PI controller parameters were identified by using plots of step responses in λ from steps in t_{inj} . The sets were identified using the three sets of rules presented in table 4.7, which are Ziegler-Nichols, Cohen-Coon and Åström-Hägglund. These sets will be validated later.

4.4.3 Implementation

The implementation of the PI lambda feed-back controller is done as shown in figure 4.9. The controller can change parameters online, and there are three different parameter sets implemented. The parameters themselves are found by the controller in look-up tables with interpolation between the different values at certain engine load points. It has been assumed that there are no sudden leaps in either intake manifold pressure, due to the transient behavior of the intake manifold air volume, and engine speed. This will lead to natural bumpless parameter change. The PI controller can be switched on and off by using a switch value called "Activate PI". This switch will also cause the integrator of the controller to reset when switched on. There is an anti-windup circuit present that countermeasures the integrator winding up when the controller has reached its saturation, but the saturation itself is set rather large so that is most often no cause of concern.

The controller works by taking the measured lambda value, λ , of the engine and subtracting the desired lambda value, λ_{des} . This gives an error value, e , which is used by the proportional and integrating parts of the controller. The output of the controller is an adjustment factor that the calculated fuel mass to inject, \dot{m}_{fi} , is multiplied with. The output of the PI part of the controller is the percentage that \dot{m}_{fi} needs to change to achieve λ_{des} . To this value a 1 is added to get the adjustment factor.

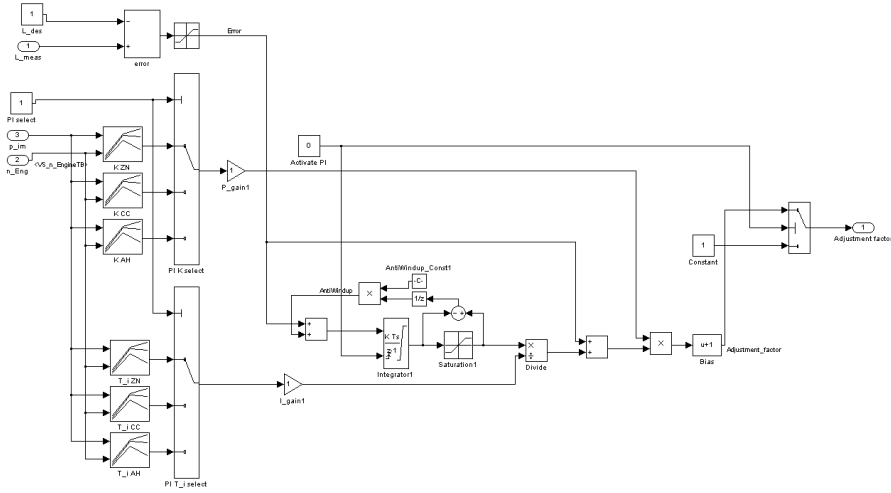


Figure 4.9. The implementation of the lambda test controller. The choice between the different parameter sets can be seen, where a switch, PI select, chooses which set is used. A variable, L_{des} , which is the model equivalent of λ_{des} , is used to set the desired lambda value the engine should run at. There is an anti-windup circuit. The switch value "Activate PI" allows engaging and disengaging the PI controller. The output of the controller is the percentage the t_{inj} should change, and this is added to 1 to get an adjustment factor that the current t_{inj} value is multiplied with.

4.4.4 Verification

The figures 4.10, 4.11 and 4.12 are plots of switching on the controller and of a disturbance in controller output at an engine speed of 1000rpm and 75kPa in intake manifold pressure. The different parameter sets will be tested by first switching the controller on with the engine running at a steady 0.8 in measured lambda value. Second there will be a disturbance introduced, where the injection time of the engine will be changed suddenly by instantly lowering it by 10%.

What can be seen in figure 4.10 is that the Ziegler-Nichols controller is behaving very strange. It is very slow in responding. It may be that it is an irregularity in just this load point, or it may be that Ziegler-Nichols is very sensitive to process dead time. The result is, however, that a Ziegler-Nichols parameter set can not be recommended from what can be seen here.

The Cohen-Coon parameter set, seen in figure 4.11, on the other hand is responding very fast, both to the initial start of the controller and the disturbance. There are however both overshoots and some oscillations in steady-state tracking of the desired lambda value.

The Åström-Hägglund set of parameters is tested in figure 4.12. It can be seen that the initial response to the start of the controller is slower than Cohen-Coon, but that there is no overshoot. The same can be said of the response to the disturbance. The following of the desired lambda value at steady state also seems to have far less oscillations than Cohen-Coon.

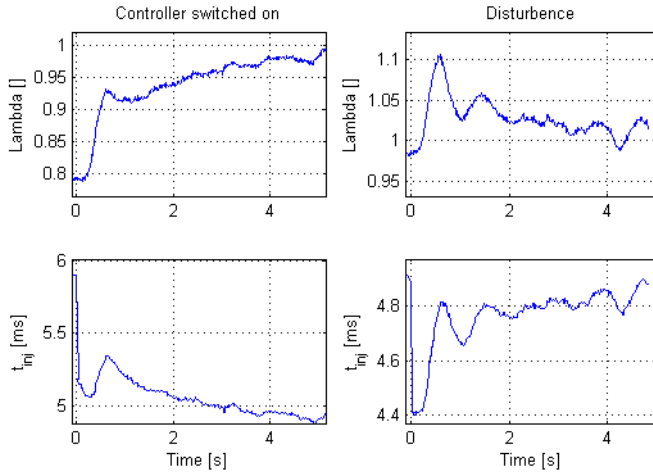


Figure 4.10. Ziegler-Nichols. The two plots to the left are the lambda value and injection time when the controller is being switched on. The time, on the x-axle, is set to 0 at the moment when the controller is switched on. The two plots to the right are the lambda value and injection time when there is a disturbance in the injection time. The time, on the x-axle, is set to zero at the time when the disturbance is introduced. The initial response to the controller being switched on is very slow. The desired lambda value, $= 1$, is reached in over 5 seconds. The response to the disturbance is also very sluggish. This is an example of bad controller parameters.

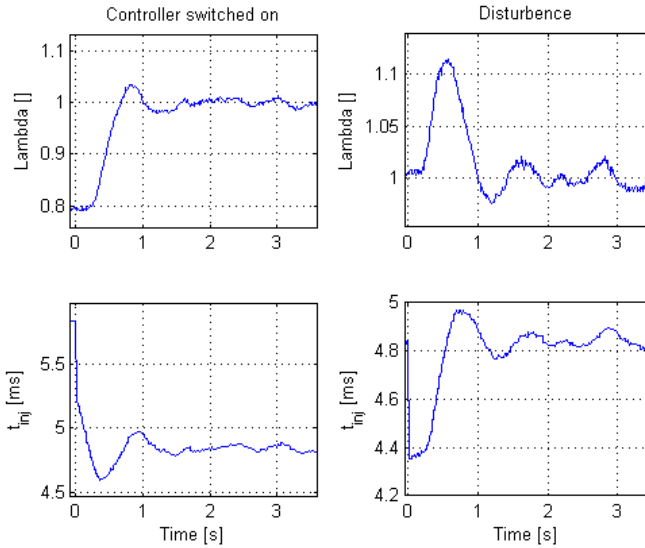


Figure 4.11. Cohen-Coon. The two plots to the left are the lambda value and injection time when the controller is being switched on. The time, on the x-axle, is set to 0 at the moment when the controller is switched on. The two plots to the right are the lambda value and injection time when there is a disturbance in the injection time. The time, on the x-axle, is set to zero at the time when the disturbance is introduced. This parameter set is responding much faster then the Ziegler-Nichols in figure 4.10. There is, however, a clear overshoot. It is, albeit, rather small, at least in this engine load point. The response to the disturbance is also rather quick, with a clear overshoot and some oscillations in lambda value after. This parameter set may be a bit too aggressive due to in part the overshoots and in part due to the oscillations when holding the desired lambda value in steady-state. There is also a clear direct through-put of λ behavior into t_{inj} behavior. It can be seen for example at 1.5 seconds where a small spike in λ is directly translated in the same small spike in t_{inj} . This is probably due to too large proportional part of the controller.

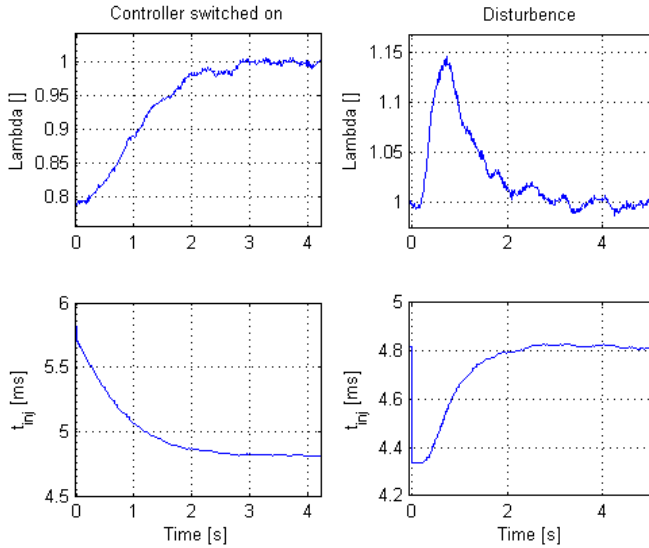


Figure 4.12. Åström-Hägglund. The two plots to the left are the lambda value and injection time when the controller is being switched on. The time, on the x-axle, is set to 0 at the moment when the controller is switched on. The two plots to the right are the lambda value and injection time when there is a disturbance in the injection time. The time, on the x-axle, is set to zero at the time when the disturbance is introduced. This parameter set is producing rather good results, at least in this load point. What can be seen is that the rise time is slower than Cohen-Coon, seen in figure 4.11, but much faster the Ziegler-Nichols, seen in figure 4.10. The response to the disturbance is similar in nature. There are no overshoots, and there is much smaller oscillations in steady-state (seen in the left-most part of the left plots). It may be that the parameter set is a bit too defensive, but the behavior of the controller is rather nice.

Choice of Parameter Set

The chosen parameter set for the λ PI controller, as can be decided with only one verification data point for each parameter set, is the Åström-Hägglund set. This choice is due to the stability of the controller. It is a bit slow, but not necessarily too slow. There is however a very limited experimental basis to base this decision on, and this is due to the limitations (see chapter 1.2).

There are lots of room for improvement with the closed loop part of the fuel controller.

Chapter 5

Summary of Models and Controllers

In this chapter a short summary of what models and controllers have been developed will be presented. There are several separately developed models and controllers in this thesis, but it is when those are put together something useful is created. They can either be put into the actual control system of the real engine or they could be put together to construct a simulation package. This package could be used to simulate how the engine behaves, and it is also a good way to present the models and controllers of this thesis to get a better picture of how everything works together. Such a package will be presented here. A table of model accuracies will be presented, where the models are presented in the same order as they are introduced in this chapter.

In section 3.1 three black box models of the volumetric efficiency are proposed, with a fourth added later in section 3.1.4. The fourth model was proposed for better transient response. The first three models used engine speed N and intake manifold pressure p_{im} to model η_{vol} , with the fourth instead using N and main throttle plate angle θ_{MT} .

Two throttle controllers were developed in section 3.2. They were implemented as two PID controllers with friction and spring torque compensation and in the case of the main throttle also compensation for the limp-home nonlinearity. The controller parameters were tuned manually.

In section 3.4 a controller for p_{im} when $p_{im,des} \leq p_{amb}$ is developed. A model of the intake manifold dynamics is developed to help with the identification and validation of the controller. The controller is manually tuned for two different engine speeds.

In section 4.1 four different models of the fuel injectors are developed, which are based on different assumptions regarding the dependencies on battery voltage and

fuel rail pressure.

In section 4.2 an open loop part of the fuel controller is developed by combining the results of the volumetric efficiency (section 3.1) and the fuel injectors (section 4.1).

Wall wetting compensation is developed in section 4.3 using a wall wetting model and the different proportions of wall wetting at different engine speeds and p_{im} are found.

The closed loop part of the fuel controller is developed and combined with the open loop part (section 4.2). Different parameter sets for the resulting λ PI controller are identified and tested in section 4.4.

There is an added model in this chapter, and that is the model for the main throttle introduced in figure 5.5. This model converts the driving signal u of the throttle to throttle position, and is based on the work done in Ahlberg et al. (2008). This model has not been fully parameterized, because it is not used in this thesis. It is however included here so that the overview of the simulation package is complete.

Model [%]	Mean abs error [%]	Variance [%]	Section
$\eta_{vol,mod1}$	2.65	12.04	3.1
$\eta_{vol,mod2}$	3.21	18.74	3.1
$\eta_{vol,mod3}$	0.78	1.07	3.1
$\eta_{vol,mod4}$	2.43	9.56	3.1.4
$t_{inj,mod1}$	1.49	3.29	4.1.3
$t_{inj,mod2}$	1.57	3.91	4.1.3
$t_{inj,mod3}$	1.82	5.95	4.1.4
$t_{inj,mod4}$	1.05	1.79	4.1.4
Open loop	1.01	1.62	4.2

Table 5.1. Summary of the accuracies of the developed models. The different models are tested with the steady-state dataset, where the mean absolute errors and the variance of the errors have been calculated. The model named "Open loop" is the open loop part of the fuel injection, and it is basically the chosen η_{vol} -model and the chosen t_{inj} -model combined. References to the thesis sections where the different models are introduced have also been added to this table, where more information can be found.

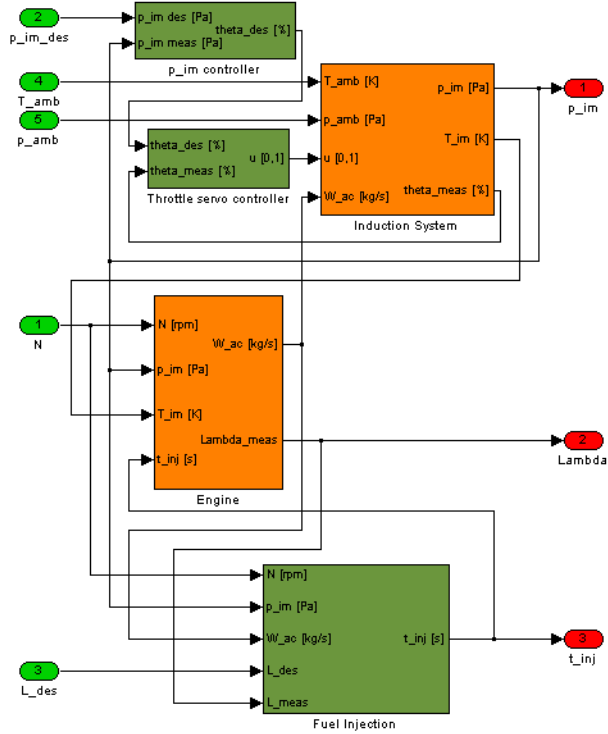


Figure 5.1. A top view of the collected models and controllers of this thesis. There is a p_{im} controller developed in section 3.4, an MT controller discussed in section 3.2, a collected set of models representing the induction system (see figure 5.4 for more details), a set of models representing a simplified engine (see figure 5.3 for more details) and a set of models and a λ controller representing the fuel injection (see figure 5.2 for more details).

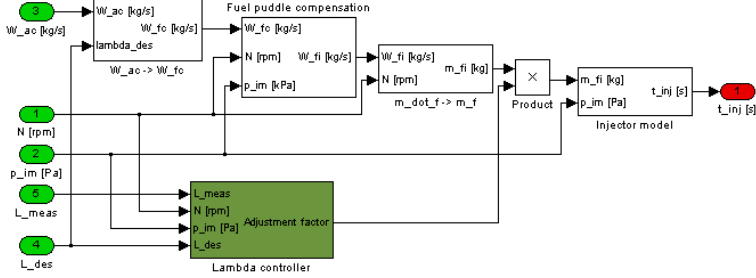


Figure 5.2. The collected models of the fuel injection together with the λ controller developed in section 4.4. To be able to run the fuel injection in an actual engine control system some sort of \dot{m}_{ac} approximation is needed. Here this is done in the engine part of the model, seen in figure 5.3. The approximated \dot{m}_{ac} is converted into desired \dot{m}_{fc} using equation 4.2 (substituting λ_{cyl} with λ_{des}). The desired \dot{m}_{fc} is then converted into fuel mass to inject by first using the fuel puddle compensation model developed in section 4.3 and then equation 4.5. The fuel mass is adjusted with the output from the λ controller. Finally the injection time t_{inj} is calculated by using the injector model developed in section 4.1.

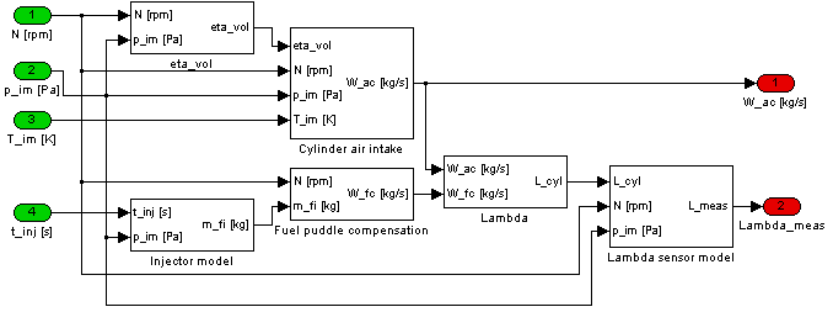


Figure 5.3. The collected models representing a simplified engine. The η_{vol} model implements the model chosen in section 3.1. \dot{m}_{ac} is then calculated using equation 3.3. The injected fuel mass is calculated by using the injector model developed in section 4.1. In the fuel puddle compensation model \dot{m}_{fi} is first calculated with equation 4.5 and then \dot{m}_{fc} is calculated using the findings of section 4.3. λ_{cyl} is calculated using equation 4.2. Finally λ_{meas} is calculated using the first order sensor model introduced in figure 4.4.

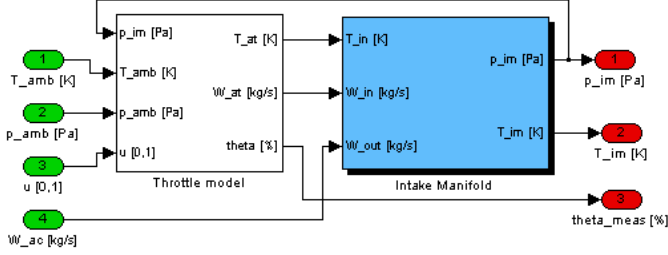


Figure 5.4. The model of the induction system uses the throttle model described by figure 5.5 and the intake manifold model described by equation 3.25.

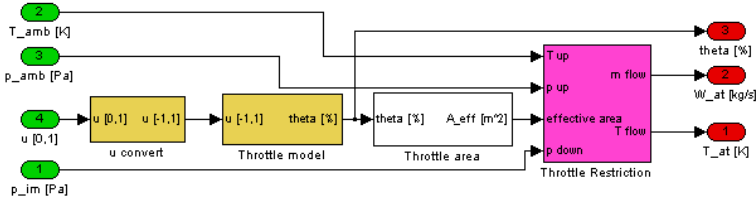


Figure 5.5. The models that the throttle model consists of. The block named "Throttle model" here is a model that implements the theory described in section 3.2.1. This model is not used in this thesis, so its parameters have therefore not been fully identified. The model is however presented here for a complete model representation. The throttle plate angle θ is converted to effective area by using look-up tables according to the findings in section 3.4.2. The throttle restriction implements equation 3.27 to get \dot{m}_{at} .

Chapter 6

Conclusions

The first three black box model of the volumetric efficiency were tested in steady-state conditions, and one model was chosen as satisfactory. Later, when testing the model on the engine in dynamic experiments with throttle transients, this model was not able to compensate for the transients in air flow. This prompted the proposition of an alternative model adapted for faster transient response. This new model was not as good for steady-state conditions, but is promising for fast transients. However it has not been validated on transients due to the engine standing still for repairs (see Limitations section 1.2).

The throttle controllers were validated with different step responses, in the case of the main throttle the steps used were performed under, over and through the limp-home nonlinearity. There were small over-shoots, but on the whole the performance of the controllers were found to be satisfactory.

Two different ways of controlling p_{im} when $p_{im,des} > p_{amb}$ are briefly examined in section 3.3. It is found that controlling p_{im} by actuating the by-pass throttle while keeping the main throttle fully open is the best way, since it produces less heat and more engine torque for the same amount of fuel.

The controller for p_{im} when $p_{im,des} \leq p_{amb}$ was validated using the developed intake manifold (IM) model, and found to work satisfactory at the engine speed for which it was tuned for. To get a functioning IM model the volume of the IM had to be identified from experiments. This volume was found to be 9 liter, which is larger than expected. One possible explanation is that there is sensor dynamics working on the intake manifold pressure sensor, this should be followed up in future work.

The four proposed models for the fuel injectors were tested with experimental data from steady-state experiments, and $t_{inj,mod4}$ was found to be the best model. It was much better than the other models and the added degree of complexity was thus deemed worthwhile, and this model is the one implemented in the control

system.

The fuel controller consists of two loops, one feed-forward part and one feed-back part where the feed-forward part consists of the volumetric model and the injector model described above.

The chosen parameter set for the feed-forward loop part of the fuel controller was the Åström-Hägglund set. This choice was motivated by the fact that it gave the best trade-off between sensitivity and response time. It is a bit slow, but not necessarily too slow due to the fact that the feed-forward part of the controller will handle large transients.

Chapter 7

Future Work

Proposals for future work:

- **Further improvements, extensions and validations of existing models and controllers:** Due to the limitations there was no time to implement for example a p_{im} controller for pressures exceeding ambient. The tuning of the p_{im} controller also needs be extended for a broader N spectrum. There was also no possibility to validate the proposed transient solution for the volumetric efficiency model. These things, amongst others, need to be done before the engine works as intended in the engine cell.
- **Ignition controller:** There need to be some sort of ignition angle and dwell angle controller. This is necessary for several other controllers that are needed to get a fully functioning engine, like idle speed controller and knock control.
- **Intake manifold pressure sensor:** Investigate the dynamics of the pressure sensor and the possibility to include a sensor model into the other models and controllers.

Bibliography

- D. Ahlberg, P. Axelsson, A. Hall, N. Lerede, T. Lindell, A. Myklebust, F. Petersson, A. Thomasson, and P. Wallebäck. *RATT: Technical Documentation*. 2008.
- C.F. Aquino. *Transient A/F Control Characteristics of the 5 Liter Central Fuel Injection Engine*. 1981. SAE Paper No. 810494.
- D.F. Caris and E.E. Nelson. *A new look at high compression engines*. 1959. SAE Technical Paper Series No. 590015.
- G.H. Cohen and G.A. Coon. *Theoretical consideration of retarded control*. 1952. Trans. ASME, vol 75 p 827-834.
- P.R. Crossley and J.A. Cook. A nonlinear engine model for drive train system development. In *IEE Conference 'Control 91'*, 1991.
- Lars Eriksson. Modeling and control of turbocharged SI and DI engines. *Oil & Gas Science and Technology - Rev. IFP*, 62(4):523–538, 2007.
- Lars Eriksson and Lars Nielsen. *Non-linear modelbased throttle control*. 2000. In *ElectronicEngineControls*, volume SP-1500, 47-51. SAE 2000 World Congress, March 2000, Detroit, MI, USA.
- Lars Eriksson and Lars Nielsen. *Vehicular Systems*. Vehicular Systems, ISY, 2007.
- Lars Eriksson, Lars Nielsen, Jan Brugård, Johan Bergström, Fredrik Pettersson, and Per Andersson. Modeling of a turbocharged SI engine. *Annual Reviews in Control*, 26(1):129–137, October 2002.
- Torkel Glad and Lennart Ljung. *Reglerteknik, Grundläggande teori*. Studentlitteratur, 1989. ISBN 91-44-17892.
- Elbert Hendricks and Spencer C. Sorensen. Mean value modelling of spark ignition engines. *SAE Paper 900616*, 1990.
- J.B Heywood. *Internal Combustion Engine Fundamentals*. McGraw-Hill, 1988. ISBN 0-07-100499-8.
- C. Kittel and H. Kroemer. *Thermal Physics (2nd Edition)*. W. H. Freeman, 1980. ISBN 978-0716710882.

- Lennart Ljung and Torkel Glad. *Modellbygge och simulering*. Studentlitteratur, 1991. ISBN 91-44-31871-5.
- Ylva Nilsson. *Modelling for Fuel Optimal Control of a Variable Compression Engine*. Vehicular Systems, ISY, 2007. ISBN 978-91-85831-36-4.
- Ylva Nilsson, Lars Eriksson, and Martin Gunnarsson. Modelling for fuel optimal control of SI VCR engines. In *Proceedings of New Trends in Engine Control, Simulation and Modelling*, IFP, Rueil-Malmasison, France, 2006.
- Institutionen för systemteknik Reglerteknik. *Industriell reglerteknik, Kurskompendium*. Reglerteknik, ISY, 2008.
- SAAB Automobile AB. Nytt, unikt motorkoncept ger ökad prestanda och lägre bensinförbrukning: SAAB variable compression engine. *SAAB Automobile AB Information 00/02*, 29 February 2000.
- Andreas Thomasson and Lars Eriksson. Model-based throttle control using static compensators and imc based pid-design. In *IFAC Workshop on Engine and Powertrain Control, Simulation and Modeling*, Paris, France, 2009.
- M. Vašak, M. Baotić, M. Morari, I. Petrović, and N. Preić. *Constrained optimal control of an electronic throttle*. 2006. *International Journal of Control*, Vol. 79, No. 5, p 465-478.

Appendix A

Wall Wetting Plots

These are the plots of the wall wetting compensation experiments that showed improvement with a model of the wall wetting. Load points with a higher load than this did not show any wall wetting.

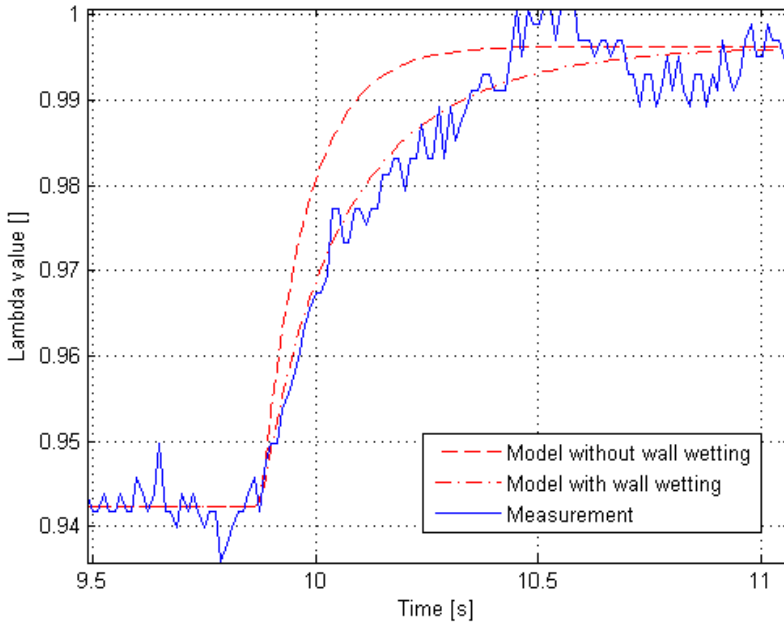


Figure A.1. Comparison between a model with and a model without wall wetting compensation at 1000 rpm and 75 kPa intake manifold pressure.

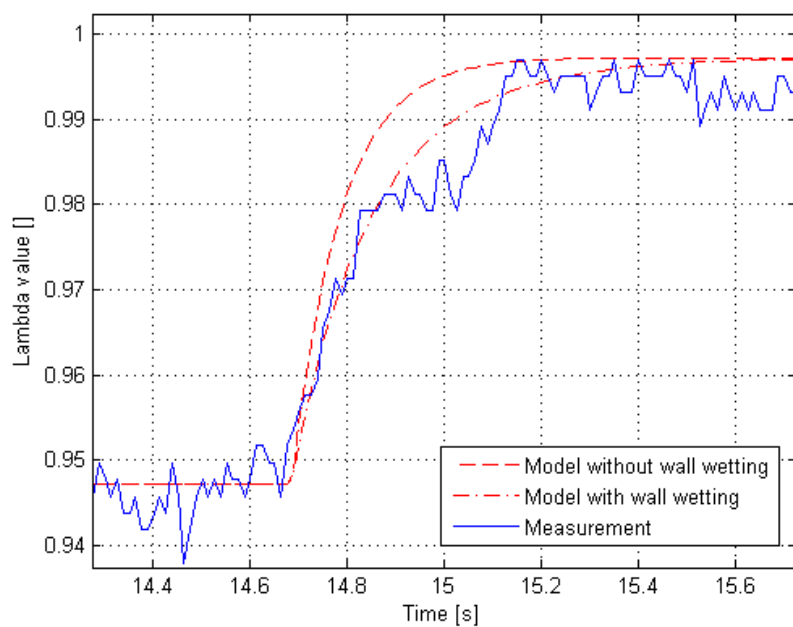


Figure A.2. Comparison between a model with and a model without wall wetting compensation at 1000 rpm and 100 kPa intake manifold pressure.

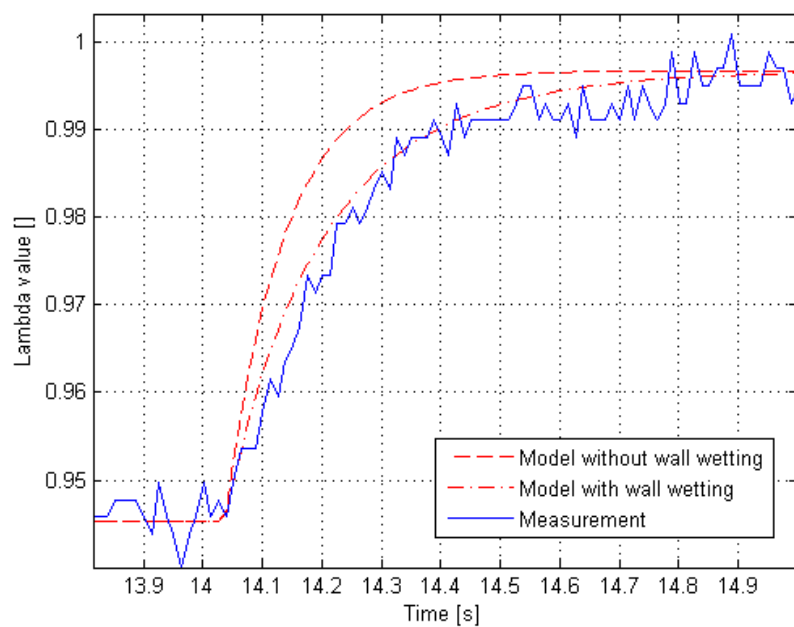


Figure A.3. Comparison between a model with and a model without wall wetting compensation at 1500 rpm and 50 kPa intake manifold pressure.

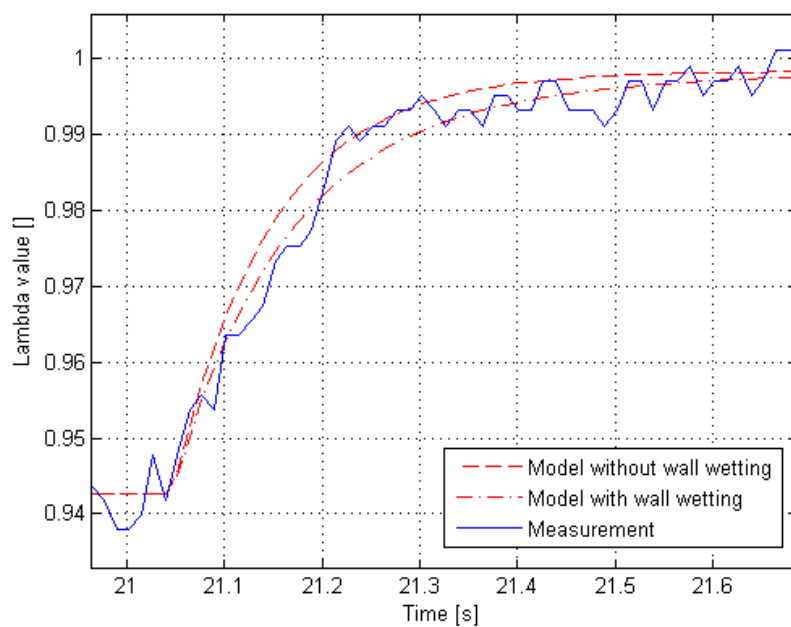


Figure A.4. Comparison between a model with and a model without wall wetting compensation at 1500 rpm and 75 kPa intake manifold pressure.

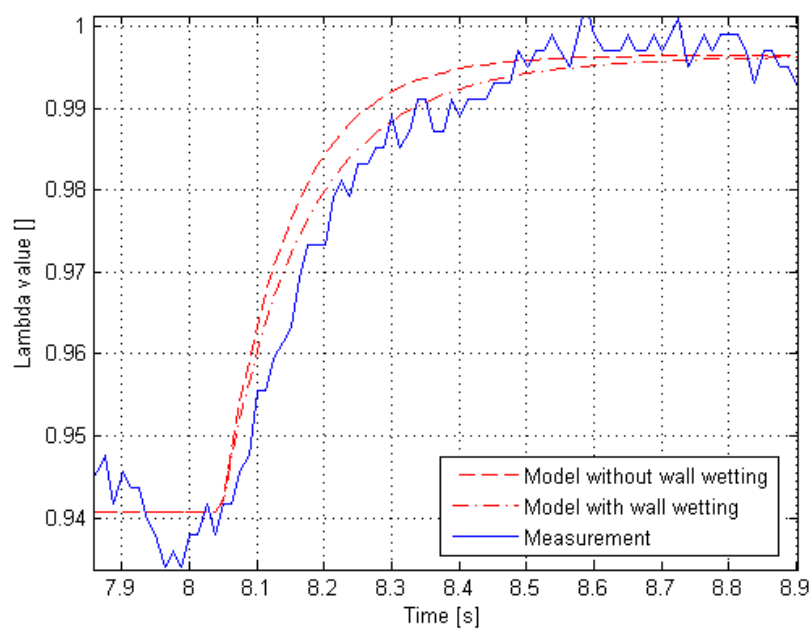


Figure A.5. Comparison between a model with and a model without wall wetting compensation at 1500 rpm and 100 kPa intake manifold pressure.

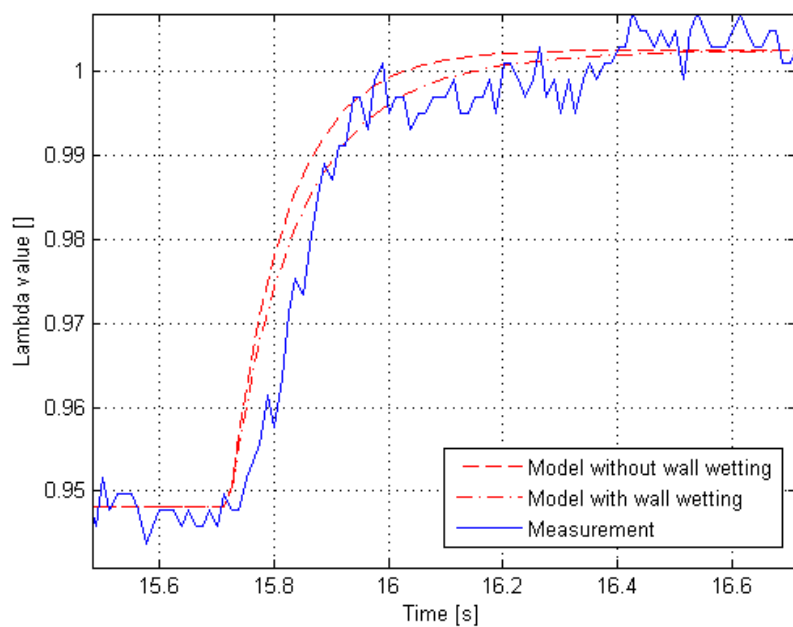


Figure A.6. Comparison between a model with and a model without wall wetting compensation at 2000 rpm and 50 kPa intake manifold pressure.

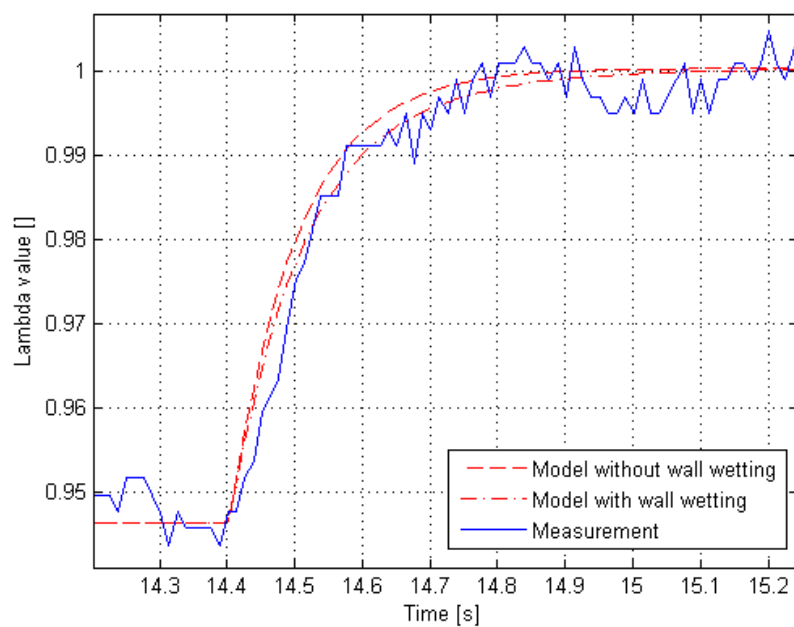


Figure A.7. Comparison between a model with and a model without wall wetting compensation at 2000 rpm and 75 kPa intake manifold pressure.

Appendix B

Source Code for Lambda Controller Parameter Finding

```
clear all;
[B,A] = butter(2, 0.17);
R = 10;

% load data set 1
load('lambda_n750_pim50.mat')
raw_data = lambda_n750_pim50;

% set parameters manually
freq = length(raw_data.X.Data) / max(raw_data.X.Data);
upper_l = mean([mean(raw_data.Y(2).Data(floor(0.1*freq):1:floor(3.5*freq)))
    mean(raw_data.Y(2).Data(floor(11*freq):1:floor(14*freq)))]);
lower_l = mean([mean(raw_data.Y(2).Data(floor(6*freq):1:floor(9*freq)))
    mean(raw_data.Y(2).Data(floor(16*freq):1:floor(19*freq)))]);
perturbation_offset = 2.89-0.93;

% filter the data set
y_filt = filtfilt(B,A,raw_data.Y(2).Data);

% derive and find the maximum slope value
y_derivata = diff(y_filt) ./ diff(raw_data.X.Data);
b = max(y_derivata(floor(9.2*freq):floor(10*freq)));

% calculate rise step size
step_l = (upper_l - lower_l)*0.63+lower_l;

t_inj = raw_data.Y(17).Data;
half_l = (upper_l - lower_l)/2+lower_l;
t_inj_max = max(t_inj);

y_filt2 = interp(y_filt,R);

j= 1;
last_t_inj = t_inj(1);
for i=1:length(t_inj)
    if t_inj(i) < t_inj_max && last_t_inj == t_inj_max
        start_index(j) = i;
        j = j + 1;
    end
    last_t_inj = t_inj(i);
end

for i=1:length(start_index)
    start_index_time(i) = raw_data.X.Data((start_index(i)));
end

j= 1;
last_y_filt = y_filt2(1);
last_last_y_filt = last_y_filt;
for i=1:length(y_filt2)
    if y_filt2(i) >= step_l && last_y_filt < step_l && last_last_y_filt < step_l
        index(j) = i;
    end
end
```

```

        j = j + 1;
    end
    last_last_y_filt = last_y_filt;
    last_y_filt = y_filt2(i);
end

j = 1;
last_y_filt = y_filt2(1);
for i=1:length(y_filt2)
    if y_filt2(i) >= half_l && last_y_filt < half_l
        half_index(j) = i;
        j = j + 1;
    end
    last_y_filt = y_filt2(i);
end

time_index = interp(raw_data.X.Data,R);

for i=1:length(index)
    index_time(i) = time_index(index(i));
end

for i=1:length(half_index)
    half_index_time(i) = time_index(half_index(i));
end

% calculate help variables
t_1 = half_index_time - (y_filt2(half_index)-lower_l)./b;
L = mean(t_1 - start_index_time);
T = mean(index_time - t_1);
a1 = max(raw_data.Y(17).Data);
a2 = min(raw_data.Y(17).Data);
P = (a1-a2)/a1;
K_p = (upper_l-lower_l)/P;

% calculate PI parameters for data set 1
K_ZN(1) = P*0.9/(b*L);
T_i_ZN(1) = 3*L;
K_CC(1) = (1/K_p)*(T/L)*(0.9*(1/12)*(L/T));
T_i_CC(1) = L*((30+3*(L/T))/(9+20*(L/T)));
K_AH(1) = 0.2*T/(K_p*L);
T_i_AH(1) = 0.7*T;

```