Bypass Valve Modeling and Surge Control for turbocharged SI engines

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

by
Eric Wiklund and Claes Forssman

Reg nr: LiTH-ISY-EX-3712-2005

August 29, 2005

Bypass Valve Modeling and Surge Control for turbocharged SI engines

Master's thesis

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

by **Eric Wiklund and Claes Forssman**

Reg nr: LiTH-ISY-EX-3712-2005

Supervisor: **Richard Backman**
GM-Powertrain
Johan Wahlström
LiTH
Per Andersson
LiTH

Examiner: **Associate Professor Lars Eriksson**
Linköpings Universitet

Linköping, August 29, 2005

Avdelning, Institution
Division, Department

Vehicular Systems,
Dept. of Electrical Engineering
581 83 Linköping

Datum
Date

August 29, 2005

Språk
Language

☐ Svenska/Swedish
☒ Engelska/English
☐ ________

Rapporttyp
Report category

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

ISBN
—

ISRN
LITH-ISY-EX-3712-2005

Serietitel och serienummer
Title of series, numbering

ISSN
—

URL för elektronisk version

http://www.vehicular.isy.liu.se
http://www.ep.liu.se/exjobb/isy/2005/3712/

Titel Bypassmodellering och surgereglering av turboladdade ottomotorer

Title Bypass Valve Modeling and Surge Control for turbocharged SI engines

Författare
Author

Eric Wiklund and Claes Forssman

Sammanfattning
Abstract

Since measurements in engine test cells are closely coupled with high costs it is of interest to use physically interpretable engine models instead of engine maps. Such engine models can also be used to do off-line tests of how new or altered components affect engine performance.

In the thesis an existing mean value engine model will be extended with a model of a compressor bypass valve. A controller for that valve will also be developed. The purpose with that controller is to save torque and boost pressure but at the same time avoid having the compressor entering surge during fast closing transients in the throttle position.

Both the extension and controller is successfully developed and implemented. The extension lowers the pressure after the compressor and increases the pressure before the compressor when the bypass valve is being opened and the controller shows better results in simulations than the present controller used in the research lab. By using the proposed controller, as much as 5 percent higher torque can be achieved in simulations.

Finally, there is a discussion on wastegate control alternatives and the use of TOMOC for optimization of wastegate control.

Nyckelord
Keywords

Mean Value Engine Modeling, Bypass valve, Surge control, Wastegate, TOMOC

Abstract

Since measurements in engine test cells are closely coupled with high costs it is of interest to use physically interpretable engine models instead of engine maps. Such engine models can also be used to do off-line tests of how new or altered components affect engine performance.

In the thesis an existing mean value engine model will be extended with a model of a compressor bypass valve. A controller for that valve will also be developed. The purpose with that controller is to save torque and boost pressure but at the same time avoid having the compressor entering surge during fast closing transients in the throttle position.

Both the extension and controller is successfully developed and implemented. The extension lowers the pressure after the compressor and increases the pressure before the compressor when the bypass valve is being opened and the controller shows better results in simulations than the present controller used in the research lab. By using the proposed controller, as much as 5 percent higher torque can be achieved in simulations.

Finally, there is a discussion on wastegate control alternatives and the use of TOMOC for optimization of wastegate control.

Keywords: Mean Value Engine Modeling, Bypass valve, Surge control, Wastegate, TOMOC

Preface

This master's thesis has been performed in Vehicular systems at Linköpings Universitet but the idea and the assignments comes from GM Powertrain in Södertälje. All the work has been done during the spring of 2005, February to June.

Acknowledgment

First of all we would like to give our thanks to Lars Eriksson at Linköpings Universitet and Richard Backman at GM Powertrain for giving us the chance to write this Master thesis.

We would also like to thank Per Andersson for giving us so much help on the MVEM engine model and Johan Wahlström for guidance during our work. We are grateful to Martin Gunnarsson for always letting us look at the engine parts in real life and for helping us with the experiments.

We would like to thank Lars Nielsen for his numerous discussions on sport, and we hereby express our sympathies for Anders Fröberg and his belief that a certain Italian car is the best there is.

Last but not least we thank Per "Kulan" Öberg for always being willing to share his pain.

Contents

Chapter 1

Introduction

The purpose of the work behind this master thesis can be formulated in a few words:

> *Shorten the time being spent in an engine test cell during development or redesign of an engine.*

That is of course a very general specification and in this case, the major area of interest is the ability to use physical models instead of static engine maps in order to control certain parts of the engine. There is already a MVEM engine model developed in Vehicular Systems at the department of Electrical Engineering at Linköpings Universitet but some parts are missing and there is a need for a partly new control system. This thesis will describe how a new valve is introduced to the model and how a new control system is developed for that valve. There will also be a discussion on what can be done to the control of the wastegate on a turbocharged engine.

1.1 The assignment

This master thesis can be divided into three areas, but they all have strong connections to each other. As shortly mentioned above the already developed MVEM engine model is missing one significant part, the compressor bypass valve, and a part of the assignment is to add that one to the existing model. The bypass valve is located on the compressor side of the turbocharger and it is there in order to make sure that the compressor do not enter surge. The bypass valve is sometimes called a recirculating blow-off valve. For all readers who are not familiar with the working principles of a turbocharger there is a short introduction in the end of this thesis in appendix B on page 58. The concept surge, will be explained in 4.1.2.

Closely coupled to the first assignment is the second one; develop a surge control system, which uses physical models instead of static maps. The actuator for that system is the bypass valve. The third and last assignment is to look for alternative ways of controlling the wastegate. The wastegate is a bypass valve located on the turbine side of the turbocharger. A short summary of the three assignments is listed below:

1. **Bypass valve**
2. **Surge Control**
3. **Wastegate Control**

When looking back on the general description given earlier and trying to link it to the assignments, the link may not be that obvious. The thing is that if it would be possible to use physical models to control surge, there would not be necessary to spend as many hours in the lab doing measurements for engine maps as done today.

The assignments are a result of an ongoing project at GM-Powertrain in Södertälje where the implementation of physical models and the use of models for simulations are of great importance.

1.2 Method

For the three assignments different methods is to be used in order to reach the individual goals.

1.2.1 Bypass modeling method

When beginning with the bypass valve model, the already implemented wastegate valve was tempting to look at. The basic working principles of the two valves are the same but one significant difference made it impossible to use the wastegate model as a blueprint for the bypass valve. The flow through the bypass valve is reversed compared to the flow in the rest of the engine model since it is recirculating air. Due to this fact a new model had to be derived with help from the mean value engine modeling library which will be introduced in chapter 2 on page 5.

Adjustments then had to be done so that the model behaved according to requirements when simulating it in Simulink. The model was finally tested and validated with data from actual measurements on an engine situated in an engine test cell.

1.2.2 Surge control method

The strategy is to build a controller that is as simple as possible and therefore some sort of PID-controller is suitable. The first attempt is to control the bypass valve with respect to pressure quotient over the compressor. This will lead to great difficulties since the pressure quotient, Π, as a function of air mass flow is almost a vertical line, which would lead to tremendous problems when trying to keep Π on one or the other side of a the surge line. Therefore a new strategy is developed.

In the second attempt, the controller is supposed to make sure that the air mass flow is not heading for the wrong side of the surgeline and if it does, the bypass valve is to be opened. Also this strategy has some flaws and another attempt is made to achieve a better controller.

A final controller is constructed, which controls the closing of the bypass valve. The valve is with this approach opened, as soon as a rapid negative trend in throttle position is detected, this is the same way of controlling as in the existing control system. The controller then try to close the bypass valve and thereby keeping the air mass flow just on the right side of the surge line.

1.2.3 Wastegate control method

Since there is an existing control system operating today the first approach is to find out whether there are improvements to be made or not. There are some different methods that can be used in order to investigate if there is anything to gain with a new control strategy and they will be further discussed in chapter 5 on page 36.

Since the aim was to find out how good it could possibly get, optimal control was suitable to use. There is a software called TOMOC which is developed for solving optimal control problem and that software has been used. There will also be a short discussion about the use of model predictive control or MPC, which is a promising modern control strategy suitable for this kind of problem.

1.3 The outline

The outline of this thesis is pretty straight forward. First of all a short introduction to Mean Value Engine Modeling, MVEM, will be given since some understanding of MVEM will make it easier to understand the thesis. That will be followed by the three major chapters in this thesis, which are devoted to the sections mentioned above, i.e. the bypass valve, surge control and wastegate control. Those chapters are followed by a chapter which summarizes conclusions drawn during the work with this thesis and finally there is a short chapter giving ideas to future work.

In the last pages there is a list of references to material used and two appendices. The first appendix has a presentation of the setup in the engine lab which was used for the measurements. The second appendix is a short introduction to the principles of supercharging and turbocharging.

Chapter 2

Mean Value Engine Modeling

In this chapter there will be an introduction to Mean Value Engine Modeling and a definition of the concept. There will also be a short introduction in 2.2 to the method used when designing a new model with help from the MVEM-library. That will lead to the description of the existing model in 2.3.

2.1 Why mean value engine modeling?

There are a couple of different kinds of modeling approaches which can be used when it comes to modeling spark ignited engines and one of them is MVEM, Mean Value Engine Modeling. There are modeling methods used which describes the operation of the engine with even more physically based equations and there are also examples using less physically based models, e.g. Black box models. In [1] the following definition of the MVEM concept is to be found:

> ***Mean Value Engine Models*** *are models where the signals, parameters, and variables that are considered are averaged over one or several cycles.*

The models concerned in MVEM are in great extent physically interpretable, i.e. the parameters have a physical meaning. There are of course approximations made but they do not in general affect the performance of the parameters of interest in a negative way.

So what is the benefit of using MVEM? The best answer to that question is accuracy. The models have a high accuracy when compared to measured values and on top of that, they are possible to simulate without too much trouble. Models that are even more complex than MVEM models can be hard to simulate since they demand a great deal of computer power. A number of articles about MVEM exist, three of them that are recommendable to start off with are, [8], [9] and [7].

2.2 The MVEM-library

When working with MVEM in Matlab/Simulink there is a predefined simulink library, which can be of great use. The so called *mvem_lib*-library was developed by Lars Eriksson during his post doc at ETH in Switzerland. This section will be a short introduction to the library and the elements it contains, for more information, reading [6] is recommended.

The general idea with *mvem_lib* is that many of the parts in the engine have basically the same functions and could therefore be modeled in the same way with exception for parameters such as length and other geometrical differences, take as an example all the manifolds connecting different engine components. All manifolds have the same basic function but different length, diameter and shape. The similarities have been used in the development of the library, for instance a block called receiver, which is a model of the manifolds, can be found in the library. Manifolds are in some literature also referred to as control volumes and sometimes also adiabatic control volumes. All in all there are ten prefabricated blocks to be found in *mvem_lib* version 0.3:

1. **Receiver**, all manifolds with exception for the exhaust manifold. Modeled with two states for securing the energy and mass balance. All this under the consideration of heat transfer.
2. **Inertia with friction**, models the turbocharger speed from a model with inertia.
3. **Compressible Restriction**, used for the throttle, valves etc. For all those restriction the area can be changed.
4. **Incompressible Restriction**, for the air filter and intercooler. Here there is no change in area.
5. **Adiabatic Mixer**, for mixing of gases of different temperature and flow velocity at a constant pressure.
6. **Intercooler Temperature Model**, is a model for the temperature drop in the intercooler.

7. **Engine Flow**, fuel and air mass flow through the engine. Everything in this model is based on the volumetric efficiency, η_{vol}.

8. **Engine Torque**, a model to describe the torque produced by the engine. Based on the gross indicated work, pumping work and friction work.

9. **Engine out Temperature**, only valid for an engine operating at $\lambda = 1$. Note that this is a black box model based on the evaluation of measurement data.

10. **Exhaust Temperature Drop Model**, for the heat transfers of the exhaust gases to the surroundings.

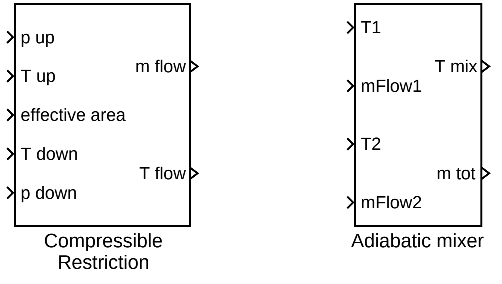

Figure 2.1: *Two of the blocks that can be found in mvem-lib. They are masked but the idea is to show in- and output.*

For more precise information on the working principle of each block, [6] is strongly recommended.

2.3 The existing model

By using MVEM-lib, Per Andersson at LiU has developed a MVEM model of a SAAB L850 engine. The structure of the model follows a specific pattern, first there is a restriction, then a control volume, then a restriction again and after that another control volume and so on and so forth. In the model it is easy to follow the air and the fuel on its way through the engine thanks to this pattern.

The model can be seen in figure 2.2 and in order to illustrate that every second block is a control volume, the control volume blocks, i.e. receiver blocks has a drop shadow. The block, TC dynamics also has a drop shadow but is not a receiver. This model will in the rest of this thesis be referred to as the original model. For further reading on the original model, reading [15] is recommended.

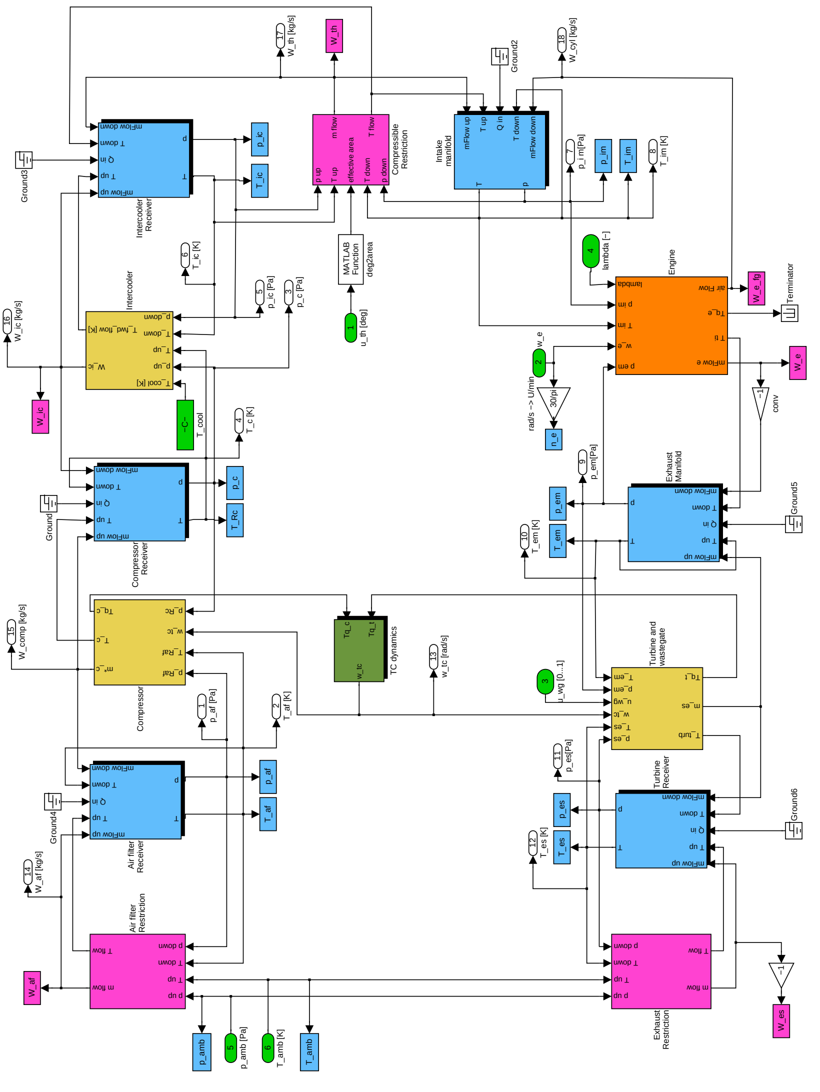

Figure 2.2: *MVEM model of a turbocharged spark ignited engine developed by Per Andersson, PhD student at Linköpings Universitet.*

Chapter 3

Bypass Modeling

In the following sections everything concerning the work with modeling the bypass valve and then mounting it into the original engine model will be presented.

3.1 Introduction

The assignment is to extend the original MVEM-model with a model block representing the compressor bypass valve. The reason for doing this is of course the fact that the engine from which the measurement data is collected has a bypass valve and for a better matching between model and reality, it is necessary to include that valve in the model as well. The bypass valve is used to avoid that the compressor enters surge. In 3.2 there will be a description of how the model was built. Some test results will be presented in 3.3 and in 3.4 an alternative solution is discussed. The chapter will end with 3.5 where some conclusions on the bypass model are presented. For an introduction to turbocharging see appendix B.

3.2 The model

This part of the thesis will describe how the bypass valve is modeled and then implemented in the original MVEM engine model.

3.2.1 Parameter identification

To build a model of the bypass valve an identification of the variables going in and out of each component has to be done. The components involved in this identification can be seen in figure 3.1. In the figure it is also easy to see what the bypass valve does, it recirculates air. Pressurized air from the compressor is recirculated and mixed with fresh air from the air filter.

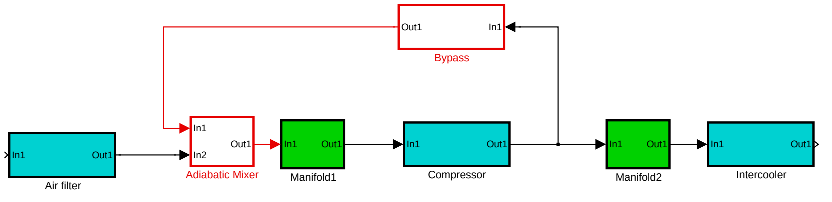

Figure 3.1: *Block figure of engine components that are of interest when modeling a bypass valve*

Furthest to the left is a model of the air filter, and there air mass flow and a temperature of the air are outputs.

The next gray block is Manifold1 and it is a model of the manifold connecting the air filter and the compressor and it has air mass flow and temperature of the air as input. As output it has a temperature and a pressure and those are then input to the compressor.

After the compressor comes another manifold and it has just as Manifold1, air mass flow and temperature as input and pressure and temperature as output. The pressure and temperature are then input to the intercooler.

The bypass has, as can be seen in the figure, connections to the two manifolds and therefore an identification of the variables going in and out of the two blocks can be made as in the table down below.

Bypass Block		**Adiabatic Mixer**	
IN	**OUT**	**IN**	**OUT**
$u_{BP} * A_{eff}$	$\dot{m}_{BP}$	$\dot{m}_{BP}$	m_{tot}
$p_{up} = p_c$	T_{BP}	T_{BP}	T_{mix}
$T_{up} = T_c$		$T_2 = T_{af}$	
$T_{down} = T_{Raf}$		$\dot{m}_2 = \dot{m}_{af}$	
$p_{down} = p_{Raf}$			

It can also be seen that the identification is done from a *mvem-lib* point of view since all the variables are named the same as in the *mvem-lib*-block, e.g.

p_{up}. On the right hand side of the sign of equality, the names used in the original MVEM model can be found, e.g. p_c. The $u_{BP} * A_{eff}$-signal going in to the bypass block is actually the controller signal, u_{BP} from the bypass controller or surge controller, multiplied with the area of the bypass hole.

The only thing left before starting to build the model is to point out the governing equations for the system, i.e. the equations describing the connection between in and out variables.

3.2.2 Bypass block equations

Since the simulink blocks are already constructed there are equations describing the process to be found. In [6] all the necessary equations are presented and in some aspects derived so here there will only be a list of them with the subscripts changed to match the ones in the original MVEM-model.

$$T_{BP} = T_c \tag{3.1}$$

$$\dot{m}_{BP} = \frac{u_{BP} * A_{eff} * p_c}{\sqrt{R * T_c}} * \Psi(\Pi) \tag{3.2}$$

With the equations above it is possible to describe the air mass flow through the bypass valve $\dot{m}_{BP}$ and the temperature T_{BP} of it. R in the equation is the ideal gas constant for air and u_{BP} is the control signal to the valve. The subscript BP means bypass and c means compressor. A_{eff} is the effective area of the bypass and Ψ describes how the flow behaves, i.e. if it is choked or not. This Ψ is calculated using the following equations:

$$\Psi(\Pi) = \begin{cases} \sqrt{\gamma}\left(\frac{2}{\gamma+1}\right)^{\frac{\gamma+1}{2(\gamma-1)}}, & \Pi \leq \Pi_{crit} \\ \sqrt{\frac{2\gamma}{\gamma-1}[\Pi^{\frac{2}{\gamma}} - \Pi^{\frac{\gamma+1}{\gamma}}]}, & \Pi > \Pi_{crit} \end{cases}$$

$$\Pi = \frac{p_{down}}{p_{up}} = \frac{p_{Raf}}{p_c} < 1$$

$$\Pi_{crit} = \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma}{\gamma-1}}$$

Where γ is the ratio of specific heats for the two air flows. These are all equations needed in order to describe the flow through the bypass valve.

3.2.3 Adiabatic mixer equations

When the air mass flow passes the bypass valve, it mixes with the air coming from the air filter before entering the compressor. This mixture has to be described and the equations have to be implemented in the model. In figure 3.1 on page 10 the adiabatic mixer block is the one of interest here. As expected the equations can be found in [6] and as done with the bypass block equations, there will only be a short summary here. The subscripts are according to the original MVEM model.

$$T_{mix} = \frac{\dot{m}_{BP} * c_{p,BP} * T_{BP} + \dot{m}_{af} * c_{p,af} * T_{af}}{\dot{m}_{BP} * c_{p,BP} + \dot{m}_{af} * c_{p,af}} \tag{3.3}$$

$$\dot{m}_{tot} = \dot{m}_{BP} + \dot{m}_{af} \tag{3.4}$$

Where $c_{p,x}$ is the specific heat of air.

3.2.4 Building and implementation

First of all an implementation of all the equations mentioned above is done in the original engine model with a great deal of help from *mvem-lib*. As can be seen in figure 3.2 the implementation is a bit confusing but in order to make all connections visible, the model will be kept like this for now. In the top right corner of the figure the ramp-blocks used for the simulations can be seen. The six blocks at the bottom of the figure are the same as the six blocks at the top of figure 2.2.

3.3 Test results

The model has only one unknown parameter, A_{eff} and in the following subsections it will be shown that the model works, how the parameter is chosen and how well it matches the reality. Since the bypass valve in reality is a pneumatic and mechanical system there is a time delay which has to be compensated for and how that is done will also be shown.

3.3.1 Simulations

The goal with the simulations is to establish whether the model is correct or not. As validation for the correctness of the model, four signals associated

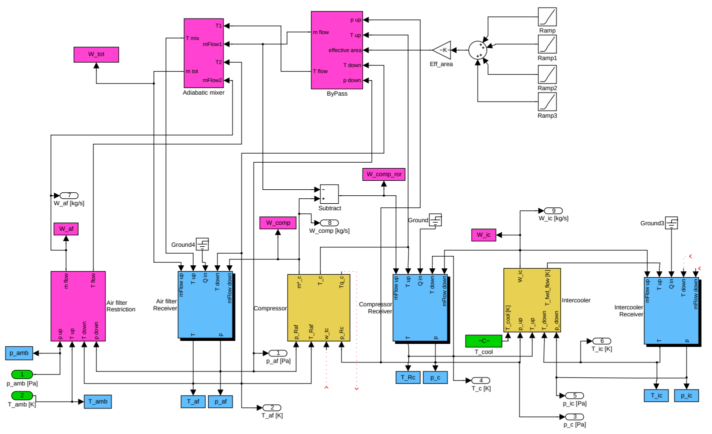

Figure 3.2: *The developed model which is used for simulations and for decisions upon the parameter* A_{eff}.

with the bypass valve comes into focus.

- **Air flow after compressor.** When opening the bypass valve the flow through the manifold after the compressor should diminish since some of the air is being recirculated.
- **Air flow before compressor.** When opening the bypass valve the flow of air going into the manifold before the compressor should increase since air is coming back from after the compressor.
- **Pressure after compressor.** The whole idea of using a bypass valve is to lower the pressure after the compressor and thereby avoiding surge.
- **Pressure before compressor.** Since high pressurized air is being recirculated the pressure on the air filter side of the compressor should be a bit higher than before.

As a test of whether the assumptions were right or not, three simulations in Simulink with the purpose of confirming all four conditions mentioned in the list above is done. The three simulations have some settings in common:

- **Throttle angle**, is locked at 40 degrees.
- **Engine speed**, is set to be 3000 rpm.
- **Wastegate**, is closed during the simulations.
- **Simulation time and solver**, is 50 seconds respectively ode15s.

The three different simulations will here be given a short presentation and thereafter an evaluation.

Simulation I
The first simulation is with the extended model, i.e. the original model extended with the bypass valve. During the whole simulation the bypass valve is kept closed just for making sure that the model is working. The important aspect here is to make sure that the model do not have any flaws that makes it impossible to simulate, e.g algebraic loops etc.

Simulation II
The second simulation is performed with the original model, i.e. the model without a bypass valve, and also here the model ran with the settings mentioned above. This test is done so that a comparison between the output from the original model with the output from the extended model can be done. The reason for doing this is to confirm that the model extension, which was simulated in simulation I, does not affect the performance of the whole model when the valve is held closed.

Simulation III
The third and last simulation is done with just the extended model in order to see if it had the behavior it was supposed to have. After about 19 seconds of the simulation, the bypass valve is opened, kept open for 1 second and then closed again. Everything with the help from the ramp-blocks referred to earlier and visible in figure 3.2. To get an idea of how the model performs and if it is correct the four parameters mentioned above is investigated.

Comments
In figure 3.3 the output of the three simulations can be seen plotted together in four different graphs. The output from the first and second simulation plotted with solid lines respectively dash dotted lines are, as they should, identical and are therefore a bit difficult to see. The output from the third simulation are the dashed lines visible in all four graphs. From the four graphs, two important conclusions can be made.

Similarity. The output from the original model and the output from the extended model with the bypass valve closed are identical. That means that the original model's performance has in no way been changed when adding the extension and keeping the bypass valve closed.

Correctness. According to the four validating parameters mentioned above the extended model has a correct behavior. When opening the bypass valve the flow going into the compressor increases a bit and the flow after the compressor diminish as can be seen in the two top graphs. The pressure in the air filter increases and the pressure after the compressor has a drop as can be seen in the bottom graphs.

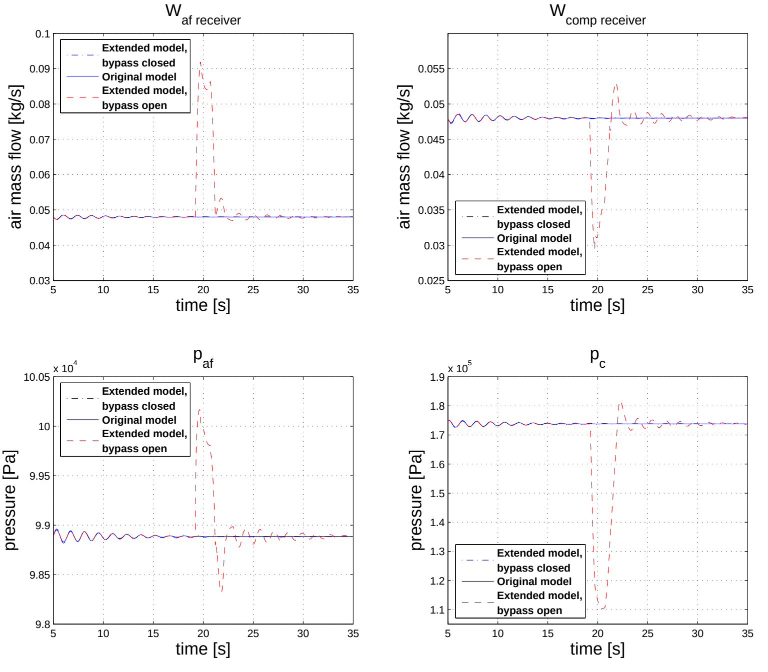

Figure 3.3: *Results from simulations, all four graphs are from the same three simulations but they describe four different outputs. There are three signals in each graph, two of them are almost identical, as expected. The third (dashed) signal deviates since it comes from a simulation where the bypass was opened once.*

With respect to the four measured parameters and their behavior, the bypass valve model is correctly implemented and the only thing left to do is to tune in the area parameter and to add a time delay.

3.3.2 Tuning the parameter

As mentioned earlier in this chapter there is only one parameter to tune, the effective flow area, A_{eff}, in the bypass valve. On top of that a model for the time delay in the valve mechanism also has to be developed.

A rough estimation of the flow area can easily be made by measuring the inlet diameter of the compressor bypass valve. By such a measurement the area is determined to be about $3.8 \times 10^{-4}\text{m}^2$ and that is multiplied with a discharge coefficient of 0.9. The discharge coefficient is a compensation for the fact that, even though the area has a specific value, the mass of air that can pass through it is a bit lower because of whirls etc. The best way to determine whether the area is correct or not is to compare results from simulations with measured data from the engine lab.

All graphs in figure 3.4 are produced through simulations with measured values from the L850 engine in the lab as input. The engine in the research lab is run at about 2500 rpm with the throttle locked in a position which generates a pressure before the throttle of about 118 kPa. Under these conditions the bypass valve is opened and closed twice and as many signals as possible are measured. Thereafter the measured throttle position, the engine speed and the signal from the engine control system to the bypass valve are used as input signals to the engine model.

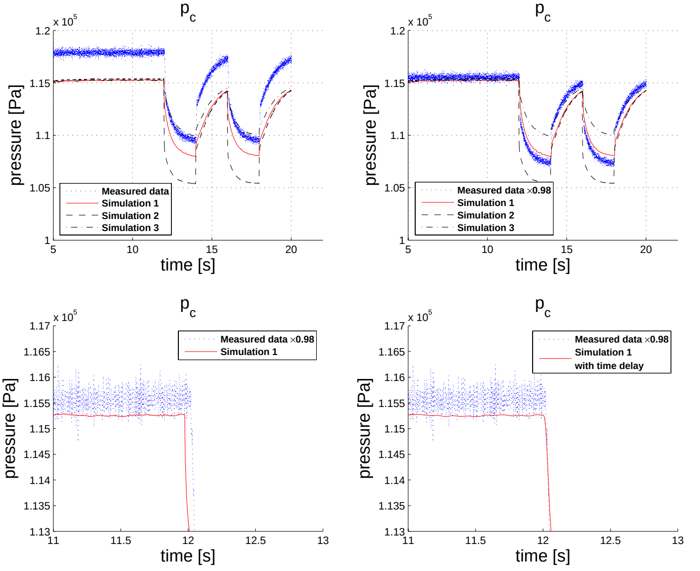

Figure 3.4: *Simulations with the same settings. In the top two graphs, the parameter A_{eff} has three different values and in the two bottom graphs a time delay is the only difference*

In all four graphs, the dotted line represents measured pressure after the compressor which is obvious since it is the only signal with measurement noise.

In the top left graph it can be seen that there is a static error in the output from the model. This static error or offset is seemingly large in the picture but is actually as small as two to three percent. In the top right graph, the measured values are simply multiplied with 0.98 when plotted with the same simulated values as in the top left graph.

In both top graphs it is also possible to see how the choice of effective area, A_{eff} affects the result. The lowest dashed line is the area coming from our estimations, the line following the measured values almost perfect in the top right graph is produced with an area corresponding to a third of the estimated area. The top dashed line corresponds to the estimated area divided by six. Worth noticing is that the pressure in the model does not build up as quick as it does in reality and this is probably due to some minor flaws in the original model.

From the measured data an estimation of the time delay in the system can also be done. In the bottom left graph the first negative step from the two graphs above has been enlarged. It can clearly be seen that there is a time delay in the real system but not in the model. This time delay consists of two parts, the time it takes to change pressure in the hose connected to the bypass valve and the time it takes to open the valve. With the settings and conditions during the measurements they were found to be about 25 ms each.

The time it takes for the pressure in the hose to build up is represented in the simulink model with a time delay block. The time it takes to open the valve is represented with a Butterworth filter of third degree. The result of this implementation can be seen in the lower right graph where the same simulation is done once again but with the time delay implemented.

3.3.3 Validation

In the previous section the area of the bypass valve was determined to be approximately $0.9 \times 3.8 \times 10^{-4}/3\text{m}^2 = 0.3 \times 3.8 \times 10^{-4}\text{m}^2$ and the total time delay to be about 50 ms. To validate this, new measurements are made and once again used as input to a simulation. This time the throttle was locked in a position which gives a pressure before the throttle of about 125 kPa and an engine speed of about 3500 rpm. As in the previous measurements the bypass valve is opened and closed twice.

In figure 3.5 the result from the simulation with the measured values as input can be seen. As before the dotted line represents measured pressure after the compressor. Note that there is no static error in the pressure this time.

In the left graph the simulated pressure after the compressor from the extended model can be seen and in the right graph the simulated pressure after the compressor from the original model can be seen. As expected there is no way for the original model to know that the pressure should decrease whereas the extended model almost perfectly models the pressure drop.

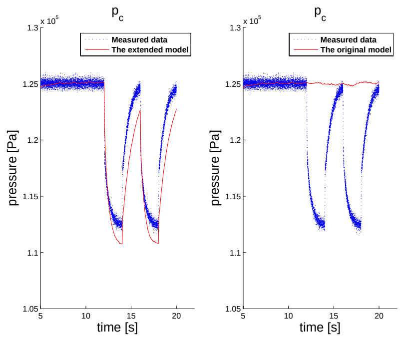

Figure 3.5: *Validation through simulations with measured values.*
The left graph describes how the extended model reacts and the graph on the right hand side shows how the original model reacts on the same simulation.

3.4 Alternative

In figure 3.6 an alternative model is shown which is completely encapsulated in the compressor block, which is the third block from the top left corner in figure 2.2. In this case the air is not recirculated so it is more of a non-recirculating blow-off valve than a bypass valve. The adiabatic mixer block could actually be removed from the model since it is not used at all. The reason for not using it, is that an algebraic loop will occur if one would connect the block to the rest of the model.

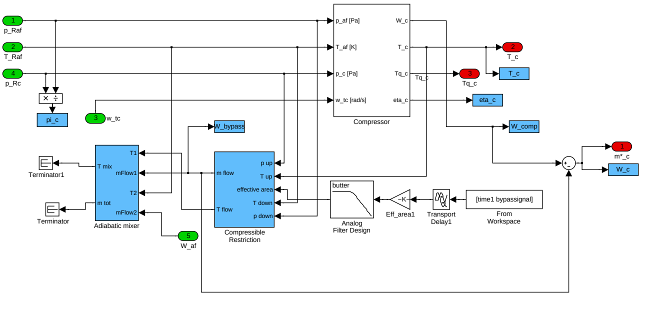

Figure 3.6: *The block figure shows how an alternative model could be constructed*

Surprisingly this model actually shows almost as good results as the extended model does. For evaluation of this model the same data as mentioned in the validation section above was used, i.e. 3500 rpm and 125 kPa.

In figure 3.7 some output signals from the extended model and from the alternative model are plotted. In the top graph the flow through the air filter is

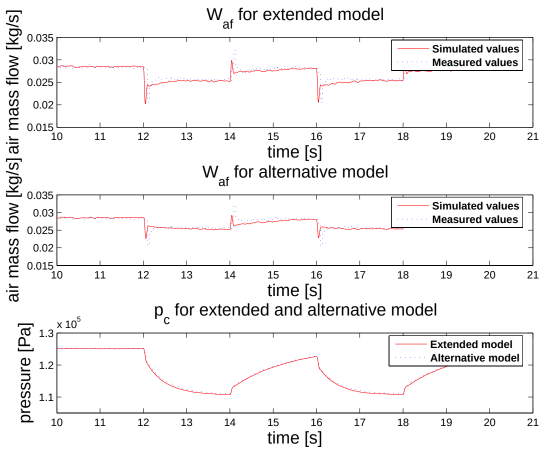

Figure 3.7: *Simulations done with the extended model and the alternative one. Noticeable is that in the two top graphs there is a slight difference between the performance of the two models. In the bottom graphs on can see that the two models have almost the same performance.*

plotted for the extended model and for measured values. In the middle graph it is the same thing but with the flow from the alternative model. Comparing the two graphs it is possible to see that the extended model models the drop in air flow, which occurs when opening the bypass valve, in a better way than the alternative model.

In the graph at the bottom, the pressure after the compressor from the extended model and the pressure after the compressor from the alternative model are plotted and there is almost impossible to discover a difference. When enlarging the graph with respect to time between 12 and 14 seconds, it is possible to see that the pressure from the alternative model, which is the dotted line, is half a percent higher than the pressure from the extended model when the bypass valve has been open for a short time.

3.5 Conclusions

In 3.2 a description of how the model was built and the necessary equations were introduced. The thoughts and equations concerning the bypass are confirmed to be correct in 3.3 and finally in 3.4 an alternative model is presented. From this there are four conclusions to be drawn:

Implementation. The equations, i.e. the blocks, are implemented in a correct way since all flows and pressures behaves as expected, and when the valve is closed the model behaves as the original one. This can be seen when looking at the four validation parameters mentioned earlier in this chapter.

Parameters. The area, which the air can pass through in the bypass valve, has been measured, determined and validated. The time delay which exists in the reality has been modeled and implemented in the model. All other parameters involved in the equations are already chosen during the development of the MVEM library so they should be correct.

Validations. The whole model has been validated with actual data from the engine lab and it performed as expected. The reason for not building up the pressure as quick as done in the reality is probably due to some minor flaws in the original model.

The alternative. It is actually almost as good as the extended model. The advantage of this model is that it can be hidden in the compressor block and thereby the structure and design of the engine model is preserved. The disadvantage is that it is not a correct model if one would look at it from a physical perspective. Most likely it would show greater flaws than the proposed model if other airflows could be measured and compared to those coming from the models.

With the conclusions mentioned above the assignment is fulfilled, i.e. the original model is extended with a bypass valve.

Chapter 4

Surge Control

This chapter will be a presentation of the difficulties with controlling surge and how those difficulties can be overcome. The reason for surge control is to maintain as high pressure as possible after the compressor without entering surge. A typical situation where better surge control could give improved performance is during acceleration, i.e. when shifting up gears.

4.1 Introduction

There are several different phenomena which have to be taken into consideration when constructing a modern turbocharged engine. In this introduction there will be a short presentation of two of them, surge and choke.

In figure 4.1 on the following page a so called compressor map can be seen, it is typically used to describe the characteristics of a compressor. On the X-axis the air mass flow through the compressor is to be seen and on the Y-axis the pressure quotient over the compressor. The pressure quotient or pressure ratio, Π, is pressure in the manifold connecting the compressor and the intercooler, p_{comp} divided by the pressure in the manifold going from the air filter to the compressor, p_{af}. The almost horizontal lines are speed lines describing how fast the compressor blades spins expressed in revolutions per minute. In general, compressor maps are only describing the characteristics for the compressor working area, i.e. where it has its highest efficiency and that is in most cases for a personal vehicle when it spins with 90 000 rpm and more. In a vehicle however the compressor mainly operates at speeds below 90 000 rpm.

The circles are describing the compressor efficiency and as can be seen the compressor has its highest efficiency for flows between $0.05\mathrm{m}^3/\mathrm{s}$ and $0.08\mathrm{m}^3/\mathrm{s}$ and for Π between 1.6 and 2.4. The line furthest to the left is the surge line, to the left of this line the compressor will enter the stage of surge, that phenomena will be explained in 4.1.2 on the next page.

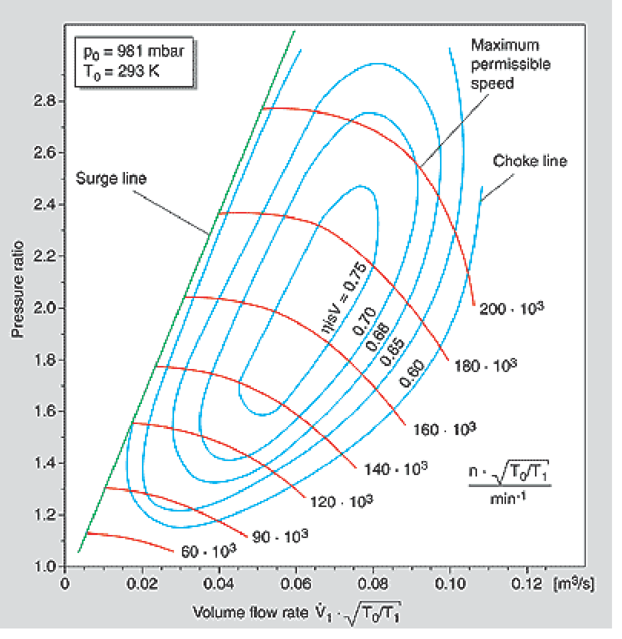

Figure 4.1: *A typical compressor map where the surge and choke line is to be seen.*

4.1.1 Choke

In figure 4.1, the efficiency circles can be seen and that they do not continue growing forever to the right and the reason for that is that far to the right in the compressor map there is natural limit. This limitation is called choke and occurs when the speed of the air flowing through the compressor reaches sonic speed. When those sonic conditions are fulfilled there is no way for the pressure wave to travel upstream since it also travels at the speed of sound. Therefore it is not possible for the pressures downstream and upstream to communicate and with no communication the flow will simply not change. In literature the flow is, under these conditions, called choked and it is not possible to increase the flow above this point. For further reading see [16].

4.1.2 Surge

The phenomena which is of most interest for this thesis is surge and it is far more serious than choke. Since surge will destroy the compressor and result in high repair costs. In the compressor map, figure 4.1, the surge line can be seen and to the left of this line the compressor is in surge. Surge can be looked upon as a standing (pressure-) wave traveling up- and downstream in the manifold connecting the compressor and throttle. If the pressure quotient, Π, would be plotted in the compressor map as function of flow during surge it would result in an elliptic shaped curve, placed parallel to the X-axis and so that it crosses the surge line.

Surge typically occurs when the throttle is closed very fast. In literature there are often distinctions between different kinds of surge and in [10], among many other things, three different forms of surge is classified.

- **Mild surge**. No flow reversal and just small vibrations in the pressure. The vibrations has a periodicity governed by Helmholtz resonance frequency.
- **Classic surge**. No flow reversal but larger oscillations in the pressure. The vibrations have a lower frequency than what is being produced during mild surge.
- **Deep surge**. In this case flow reversal is possible and in a compressor map it could, in some cases, be plotted as an unsteady flow symmetric with respect to the Π-axis.

Of course mild surge can be handled for a short period of time but if deep surge would occur, it would probably be the end of the turbocharger.

The surge problem in a vehicle engine occurs mainly, as mentioned earlier, when there is a rapid negative change in throttle position, typically as the driver is about to shift up gears. When the driver takes his/her foot off the gas pedal during gearshift the requested air mass to the cylinders will be heavily reduced causing the control system to close the throttle.

During the closing of the throttle, the mass flow trough the compressor drops very fast but unfortunately the pressure ratio over the compressor does not change nearly that fast. This slower change in pressure is due to the inertia of the turbo meaning that it takes some time for the compressor wheel to reduce its speed. Looking back at figure 4.1 on the preceding page it can be seen that for a drop in mass flow combined with unchanged compressor speed leads to a small rise in Π rather than a drop. Π starts to drop when the speed of the compressor reduces. This behavior of the compressor is the reason for entering the stage of surge.

4.1.3 Surge control methods

Surge is a well known problem and since it works as a performance limiter for the turbocharger, extensive research efforts have been made to solve this problem. Most of the work however only consider industrial turbo machinery and do not so much take the vehicle engine point of view into consideration. A lot of the work and research that has been done in this area is summarized in [11], and in [14] there are a number of control alternatives given. In [10] there is a classification of the different control strategies that can be used for surge control.

- **Active surge control**. This is a very complicated method and typically specific for each machinery, very few general methods, if any, has been derived. It basically uses different actuators to move the surge line further to the left. Examples of actuators are geometry affecting valves and microphones. A different geometry will affect the pressure vibrations and the microphone can transmit sound waves which also affects the pressure vibrations. For further information [11] is a good starting point.

- **Surge detection and avoidance**. This a classic method when it comes to turbo machinery but it is not so well investigated when it comes to turbochargers. Probably there would be problems using it on a turbocharger since it depends heavily on the response time of the sensors. The basic principle is to have sensors detecting when the compressor is about to enter surge and then use actuators, like a bypass valve, to avoid surge. The problem is that in a car the actuator system is not fast enough.

- **Surge control/ avoidance/ protection**. The simplest way of controlling surge. Here the idea is to make sure that the turbocharger does not have a chance to enter the region where surge can occur. This is typically done in cars today where the control system uses some sort of predicted airflow to detect critical changes which could cause surge. An example of such a system is given in 4.1.4.

4.1.4 Surge control in the research lab today

More information on the research lab can be found in appendix A and will therefore not be discussed here. To reduce the risk of entering surge the present engine management system in the research lab, Trionic 9 or T9, opens the compressor bypass valve when there is a strong negative trend in the requested air mass to the cylinder. This has the advantage of very fast response and thereby ability to avoid surge but since the system keeps the bypass valve open for a certain time, typically around 1.5 to 2.0 seconds, there is also a disadvantage in form of an unnecessary loss in boost pressure.

An ideal solution would be if the controller could open the bypass valve as soon as there is a risk for surge but then, by slowly closing the valve keeping the flow just to the right of the surge line. The result would be that some of the pressure which is being lost normally, during fast transients, is kept. That leads to better efficiency for the engine and a higher torque when shifting up gears.

4.2 The controller

The aim of the controller for the bypass valve is to avoid surge and to minimize the time the bypass valve is being held open. This part of the thesis will explain how some of the major challenges when constructing such a controller can be overcome.

4.2.1 Control challenges

It might not seem that hard to construct a simple controller that makes sure that the surge line is never being crossed, but there are many obstacles that have to be overcome in order to get a controller that works properly. The issues affecting the controller design is listed below.

- **Discrete controller signal.** The controller signal has to be either one or zero since the actuator is discrete, this results in the valve being fully open or fully closed.
- **Fast changes.** The mass flow through the compressor changes very fast when about to enter surge so the controller has to be prepared for these changes.
- **Time delays.** Both in pressurizing the hose between the bypass valve actuator and the bypass valve, as well in lifting the actual valve.
- **Implementation in simulink.** It is not that easy to implement time delays in simulink and getting them to work properly.
- **Tuning the controller.** Problems finding good values for the parameters in the controller.
- **Noise.** Measuring signals always leads to measurement noise and this has to be taken in consideration.

- **Non measurable flows.** Not all signals are being measured, e.g. air mass flow through compressor and bypass valve so there are few available signals for a controller.

- **Controller structure.** To choose a structure of the controller that handles the challenges listed, in the best way.

4.2.2 Method of control

The main idea of how to control the bypass valve is to take the surge line, in the compressor map, as reference value and from that being able to say if the bypass valve should open or close. The idea is to have an ordinary PID-controller taking either the air mass flow through the compressor, or the pressure ratio over the compressor as input signal. Output from the controller should be a discrete signal used for opening or closing the bypass valve entirely, since the actuator is discrete.

4.2.3 Choice of controller

Due to the problems mentioned in section 4.2.1 on the previous page a standard PID-controller can not just be implemented without violating the surge line restriction, so therefore other alternatives has to be searched for. The solution is a final controller consisting of two parts. First there is a part that reacts on negative changes in the throttle position combined with a "timer"-block that keeps the bypass valve open for 1.5 seconds which is about the same as done in the present control system. Second there is a P-controller that has the mass flow through the compressor as controller signal.

This design can be seen as an extension of the present controller with a P-controller handling the closing of the bypass valve. Where the second part, reacting on throttle position in combination with the timer would react similar to the present control system. Take a look at figure 4.2 on the facing page to see how it has been implemented in simulink.

The reasons for choosing this way of controlling are several. Having the air mass flow, instead of the pressure ratio, as input to the P-controller gave a faster response from the controller. This is because the changes in mass flow are much faster than the ones in pressure. Unfortunately it turned out that this did not generate a fast enough control system, therefore the part which reacts on negative throttle changes were implemented.

The present control system in the research lab reacts on requested mass flow, which is an even faster signal than the throttle position. The reason for, despite the increased reaction time for the system, choosing the throttle position as input is simply that the requested mass flow is not available in the model but the throttle position is. In reality there might be benefits to be won in using the requested mass flow instead.

Having the controller to keep the valve open for a seemingly long fixed time and then adding a P-controller which is trying to close the valve when it is being open for a too long time, leads to a more stable performance. A P-controller that handles the opening of the valve often leads to, due to the time delays, that the surge line restriction can not be meet.

Because of the simplicity of a P-controllers design, that type of controller is chosen instead of a PI-, PD-, or PID-controller. A P-controller is simple to implement and simple to test in the engine research lab. When using the controller in simulations and reality, having measured signals as inputs, a D-part complicates the performance due to all the measurement noise and also because the controller has to work in discrete time. In section 4.4 on page 31 further information and discussions about other control strategies can be found.

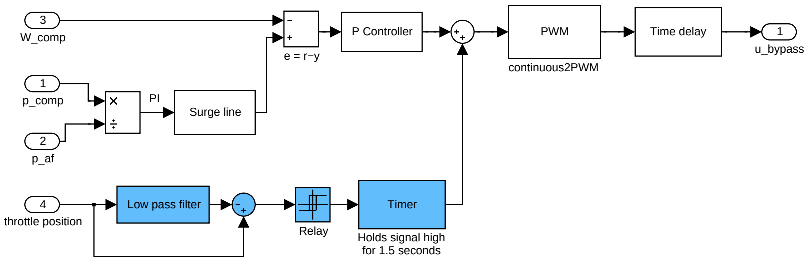

Figure 4.2: *The bypass valve controller with its two parts implemented in simulink.*

In the following list some of the major blocks in the simulink model in figure 4.2 are described.

- **Surge line:** Calculates the reference value to the P-controller, i.e. the reference mass flow, for the given pressure ratio over the compressor. It is done by multiplying the pressure ratio, Π, with a second order polynomial which is derived from data given by the compressor manufacturer.

- **Time delay:** Consist of two parts, first a Butterworth filter which is meant to mimic the dynamics in the actual valve, i.e. the time it takes to lift the valve. The second part is an ordinary time delay block, which models the time it takes before the valve begins to open after the control signal is set to one. The time-delay block has got to be placed after the Butterworth filter in order to get the model to work properly during simulation. If it would be built the other way around the control restriction would not be able to be fulfilled. This is because it is easier to delay a continuous signal than a Bang-bang signal.

- **PWM:** Converts the continuous control signals into a pulse width modulated (PWM) signal. Meaning that the continuous signal, taking values between zero and one, tells the PWM-block how long it shall keep the signal high in comparison to keeping it low. Meaning that when the input signal is 0.3 the PWM sets the output signal to one 30 percent of the time and to zero 70 percent of the time. The PWM works with a frequency of 50 Hertz.

- **Timer:** It keeps the control signal high for 1.5 seconds when the relay in series with the low pass filter detects a large negative trend in throttle position. The task of this block is to create a behavior similar to the one given by the present control system, T9.

- **Low pass filter:** Taking the throttle signal and running it through a first order low pass filter and then comparing it to the original signal, gives a signal out that has a peak every time an excessive change is made in the throttle signal.

All the four input signals comes, during simulation, from the MVEM-model. When testing the controller in the engine research lab the mass flow through the compressor, W_{comp}, is not measurable and therefore an observer has to be used to estimate the mass flow.

4.3 Test results

As can be seen in figure 4.2 on the previous page there is the constant K_p in the P-block and the threshold of the relay that have to be parameterized. In this section it will be shown how this is done and also an evaluation of the chosen values will be made. Finally the controller will be compared to the present control system to see if there are substantial benefits to be won by choosing the proposed controller.

4.3.1 Tuning parameters

When it comes to choosing the parameters for the P-controller it is not that straight forward, because of the fast changes in flow and the time delays in the bypass actuator. Tuning the threshold value for the relay caused no greater problems, since just the excessive negative changes were of interest. The threshold for the relay therefore has to be set high enough so that it will not react on the measurement noise or the smaller changes which are not dangerous for the compressor. But the threshold must be low enough to detect the excessive changes. A threshold equal to 0.25 volt, gave a response from the relay in all the necessary scenarios.

When the second part of the controller has opened the valve, the P-controller will operate as a closing controller, i.e. if the mass flow in the compressor map is to the right of the surge line the controller will reduce the opening of the bypass valve. Parameterizing this controller is done in a way similar to Ziegler-Nichols method of tuning PID-controllers, see [3] for further reading about this method. The problem here is that the system can not be brought to self-oscillation in an obvious way. What can be done is to increase K_p until the control signal starts to oscillate severely, this value for K_p is then called K_0. According to Ziegler-Nichols method, K_p should be set to 50 percent of the value of K_0, in this case however a slightly higher value than this is chosen and K_p ends up equal to 55. K_0 in this case is about 75. In figure 4.3 the results of different K_p can be seen.

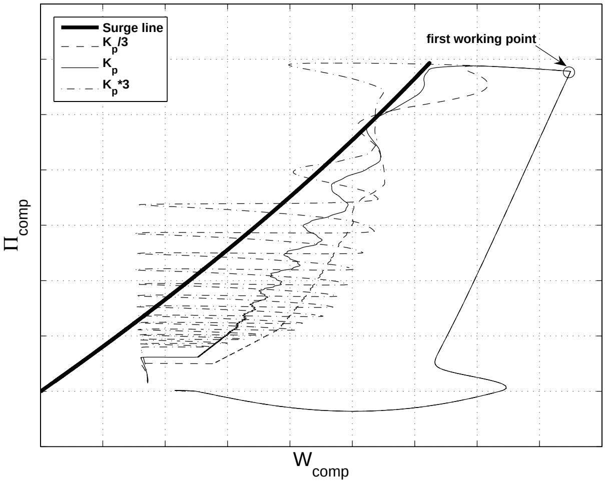

Figure 4.3: *A simplified compressor map showing the effect from different K_p on a P-controller*

The graph is from a simulation in simulink with the extended MVEM-model. During the simulation the engine speed was set to 2 000 rpm, the wastegate was kept closed and a large negative step was made in the throttle position. A simulation with these settings gives a scenario with a high pressure quotient and an extensive mass flow, meaning that this parametrization should work for a large variety of engine working points since most other scenarios are not so extreme as this.

In figure 4.3 it can be seen how the mass flow and pressure quotient changes and if the surge line is being crossed during the simulation. The thick line is the surge line, which act as a reference for the controller. The solid thin line is the proposed choice of K_p and the differently dotted lines comes from when simulations with K_p three times smaller and three times larger than the proposed K_p.

In the top right corner of the figure is the first stationary working point, i.e. for the open throttle. As can be seen the way to that point is the same for all the three simulations. When the throttle then is being closed the mass flow is quickly being reduced, how much depends on the value of K_p. It can clearly be seen that the proposed choice of K_p is good choice, since it does not ever cross the surge line.

It can easily be shown that if a D-part would be added, the performance should improve but since keeping the controller simple to implement in the engine lab is of importance, adding an I- or D-part is not preferable. In section 4.4.1 on the next page it can be seen what a PID-controller could do to the results.

To claim that these parameters, K_p and the threshold, are optimally chosen would be a bit daring, but they are neither way off track. The goal is not to locate the optimal solution but rather to find one solution that is working better than the control method used in the present control system.

This way of implementing the controller is a safer way than having a PID-controller that both opens and closes the bypass valve since the mass flow then tends to end up on the wrong side of the surge line due to the time delays in the system. The chosen controller is more reliable but has the disadvantage of having the valve open a bit longer than what is necessary.

In figure 4.4 on the facing page it can be seen how the results in boost pressure and on the torque from the engine crankshaft differs from the simulated present control system and the controller described earlier in figure 4.2 on page 27. The left graph shows how much more boost pressure there is directly after the compressor in percent, the right one shows how much more engine torque this way of controlling can gain, also in percent. Before the step in throttle position is carried out, after ten seconds of the simulation, there is no difference in either pressure nor torque. However one second later there is an almost 3 percent higher boost pressure and more than 5 percent extra torque from the engine crankshaft.

What should be noticed is that these graphs comes from simulations, carried out in the same way as mentioned earlier when deciding upon K_p, on the engine model and not from actual measurements. Meaning that the time of the opening for the simulated present control system might not be totally accurate, but it gives at least an indication that there are benefits to be won in using this way of controlling. As can be seen there are substantial benefits in

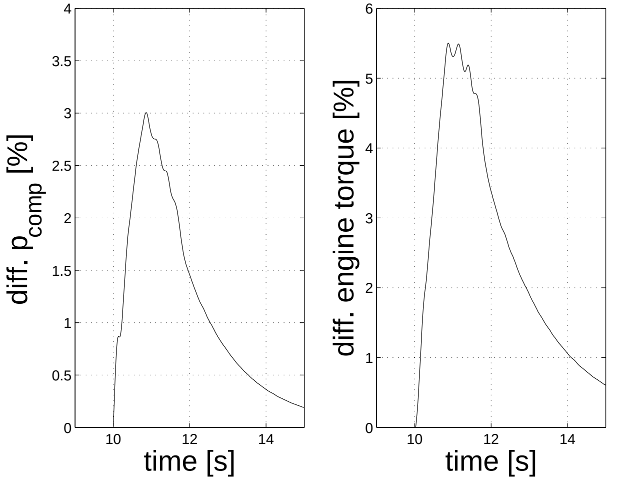

Figure 4.4: *Difference in percent between P-controller vs. simulated present control system referring to compressor pressure and engine torque.*

both pressure and, most important, the engine torque.

4.4 Alternatives

There are many interesting alternatives to the proposed P-controller and some of them will be discussed here. A couple of adjustments and improvements that can be done to the proposed controller will also be presented.

4.4.1 PID controller

One natural extension of the controller would be to extend the P-controller with an integrating and a derivative part. As mentioned earlier, a P-controller is chosen because it is simple to implement and test in the engine research lab, but of course extending it to a full PID-controller could be of interest. In figure 4.5 on the following page it can be seen how a PID-controller, behaves in comparison to the P-controller. As can be seen there is somewhat of an overshoot before the mass flow is stabilized on the right side of the surge line. The overshoot is the dash-dotted line to the left of the surge line. That overshoot occurs because the integrating part is trying to close the valve more than a P-controller would do, and that when situated in the allowed area of the compressor map, resulting in a smaller opening of the bypass valve than what is recommendable. This could perhaps be solved by either better tuning

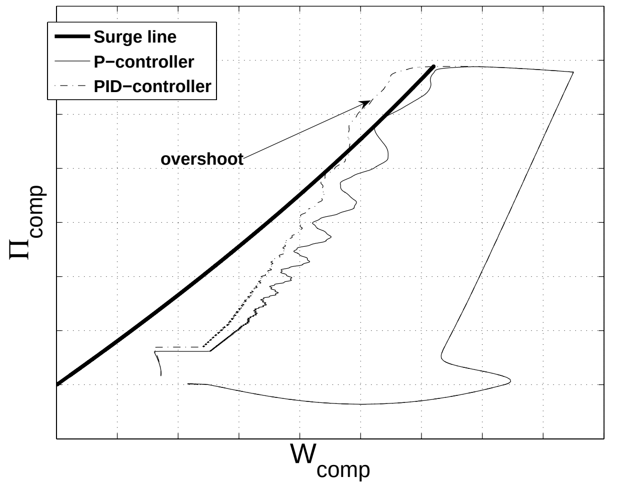

Figure 4.5: *A compressor map showing the flow from the proposed P-controller and the flow from an alternative PID-controller*

of the PID-controller or a better anti-windup function for the I-part than the one used here.

4.4.2 Changing the actuator dynamics

The most severe problems, when it comes to controlling the bypass valve, are the time delays. The delays in combination with very fast and large oscillation in mass flow results in controller difficulties. One way to get around this problem would be to try to reduce all these delays. Ideal would of course be if there were no time delays at all and if the system would not be discrete, i.e. the control signal could be continuous.

The results from a simulation with such a system controlled only by a simple P-controller can be seen in figure 4.6 on the next page. The simulation is the same as mentioned earlier, i.e. all the settings are the same as in previous simulations. This gives an indication of how good it theoretically can become. This would however require a change of actuator and that would of course increase the costs of manufacturing. As can be seen in the top left graph in figure 4.6 a simple P-controller is enough to get a controller that almost perfect follows the surge line and thereby maintaining as much boost pressure and engine torque as possible. In the bottom left figure the change in compressor wheel speed can be seen. Having an ideal actuator would result in as much as 40 percent increase in turbine speed and an extra seven percent of boost pressure, as can be seen in the top right figure. But most important

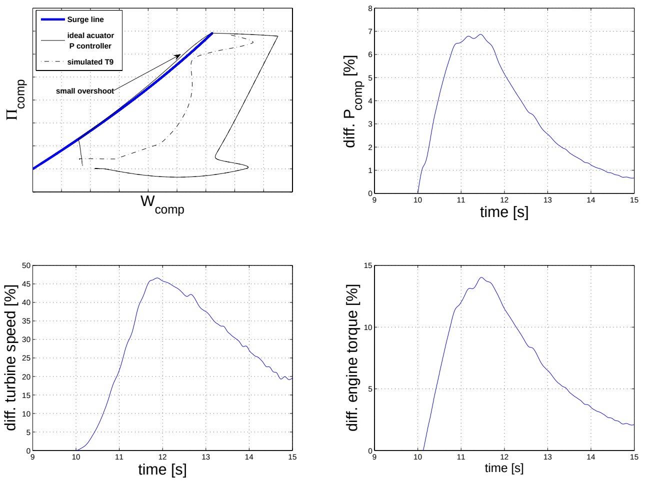

Figure 4.6: *The benefits of an ideal actuator vs. the simulated present system*

of all, in the bottom right figure, there is a 14 percent increase in the engine torque 1.5 seconds after the throttle is being closed.

4.4.3 Miscellaneous control ideas

Besides the two alternatives presented above, some other ideas exists and they will be analyzed here.

An air mass flow controller: An alternative way of solving the control of the bypass valve which do not include the requested air mass signal would be to have a PID-controller to open and close the valve. This way of controlling is not very effective and simple simulations can show that due to the time delays in the system the controller is not fast enough.

Even if the line that is used as reference, is moved further to the left in the compressor map the controller is still to slow. Performing this kind of experiments is an easy way to show that the time delays in the system makes controlling difficult and that some sort of predicted signal is needed.

Requested air mass flow: As mentioned before the requested air mass flow into the cylinders can be used as controller signal, as done today, instead of throttle position to open the bypass valve. That would lead to an even earlier opening of the valve and perhaps better results. The reason why this is not implemented in the chosen controller is because of the fact that in the MVEM-model the requested mass flow is not available but the throttle position is.

4.5 Conclusions

In this chapter a description of surge was given and the problems with the phenomena motivates the development of a controller which controls the by-pass valve developed in the previous chapter.

In 4.2.1 on page 25 there are a couple of obstacles which had to be overcome and how that is done can be found in the text but some solutions are also listed here:

- **Discrete controller signal.** A discrete controller signal can be constructed by having a continuous signal and pulse width modulate it.

- **Fast changes.** The fast changes in pressure and especially flows occurs when in surge and therefore an early indicator of changes in throttle area is needed. In a real system this can be solved by using the signal "requested air mass flow" which comes directly from the gas pedal. In simulations when working with the MVEM-engine model, the throttle position can be used to generate a preview of how the air mass flow will look like.

- **Time delays.** In the controller model, time delay blocks were implemented in order to mimic the reality so that the final controller could be able to handle such delays in the reality.

- **Implementation in simulink.** The implementation in simulink was successful and is described in 4.2.3 on page 26.

- **Tuning the controller.** The controller ended up with only one parameter that had to be tuned and that is done with the theory from Ziegler-Nichols as base.

- **Measurement noise.** In order to diminish the effect of measurement noise, the use of a Butterworth filter comes in handy. This should be done for all measurement data going in to the model or controller.

- **Non measurable flows.** The deficit with not having the flow through the compressor can be overcome by using an observer and in that way calculate the flow.

- **Controller structure.** Construct a controller that can handle the items listed above can be done and is described in 4.2.3 on page 26.

The developed controller shows great results in simulations and is definitively worth further work. With the settings proposed in this chapter, the controller could save as much as five percent more torque than the present control system used in the engine research lab. A five percent higher torque means a more rapid response when shifting up gears during acceleration. This is a result of having a three percent higher boost pressure as can be seen in figure 4.4 on page 31.

It has also been shown in simulations what can be done when using a continuous controller and an ideal actuator. That result is perhaps even more interesting since it shows that great improvements can be made when looking at torque. As much as 12-14 percent higher torque is achievable. Such a controller is in the simulations able to completely keep the air mass flow on the right side of the surge line even though the steps in throttle position are extremely fast.

Due to technical problems in the engine research lab, the controller is not yet tested on a real engine in a test bench. The controller is however possible to compile to a real time system and then be used in real time so with a correct interface to the engine there should be no problems using it on a real engine.

There is actually one benefit when trying to use the controller on the engine in the research lab and that is the fact that T9 has the "requested air mass flow"-signal which makes sure that the bypass can open much earlier, i.e. it is an even safer system.

Since the settings used during simulations are pessimistic and the steps made are extreme there should be no problem for the controller to handle most cases also in a real application.

In 4.4.3 on page 33 it is made clear that there is need for a predicted signal of some sort which gives a hint if the airflow is about to enter surge, otherwise the controller will not be fast enough.

One way to improve the systems dynamics, that would not cost so much, would be to rearrange the location of the components. Placing the actuator of the bypass valve in the immediate surroundings to the valve, would eliminate the time delay that comes from changing pressure in the hose connecting the bypass valve actuator and the actual bypass valve. Today the location is one meter away which results in a substantial time delay. A relocation like this would render in only one time delay, the time it takes to lift the valve, and hopefully an easier system from a control point of view.

Chapter 5

Wastegate Control

In this chapter there will first of all be a description of the control problem and thereafter a section concerning optimal control in general and TOMOC in special.

5.1 Control goals

The purpose with this chapter is not to develop a new controller but to investigate if there are any improvements to be done to the existing controller. A controller that uses both the throttle and wastegate can perhaps be more efficient than the controllers normally used. One challenge is the fact that most modern non-linear control methods are tricky to implement in real time.

5.2 Optimal control

The purpose of optimal control is, as the name reveals, to find the optimal way of controlling a specific system. This is done in off-line mode and can either be used for bench marking of different controllers or be used for deriving good reference values for the controller. Some of the benefits of using optimal control is that it handles both open-loop and close-loop control problems, it is also possible to solve non-linear control problems. One drawback is that even for rather simple control problems it is difficult to find the optimal solution. In many cases simplifications have to be made in order to solve the problem. The high complexity of the problems also leads to great demand on computer power in order to be able to solve the optimal control problem in a reasonable amount of time, therefore optimal control can not be used on-line. For further reading about optimal control please consult [4].

In this thesis optimal control is used in order to get a feeling of how good a controller theoretically can become and for getting an understanding of how the response time affects the fuel consumption.

5.3 TOMOC

TOMOC is a tool for solving optimal control problems based on TOMLAB. The big difference between TOMLAB and TOMOC is TOMOC's ability to handle dynamic functions, which is a requirement for solving optimal control problems. It works in a Matlab environment and is build up around different m-functions and in this thesis the built in optimization solver in Matlab is used. TOMOC was developed by Adam Lagerberg at School of Engineering at Jönköpings universitet in his Ph.D thesis but had to be slightly modified in order to be suitable for this problem. The reasons for choosing TOMOC as a solver for the optimal control problem in this thesis are mainly because it was available and seemed reasonably easy to use. Below, a short description of how TOMOC works is given but for additional information regarding TOMOC, please see [13].

5.3.1 Structure

Optimal control in TOMOC is all about finding the best (optimal) control function $u(t)$, that minimizes the cost function described in equation 5.1 subject to constraints. The dynamics of the system is defined through the state equations where *x* is the state vector and *u* is the vector of control signals as seen in 5.2.

$$\min J = \phi(x(t_I), x(t_F), t_I, t_F) + \int_{t_I}^{t_F} L(x(t), u(t), t)dt \tag{5.1}$$

$$\dot{x}(t) = f(x(t), u(t), t), \quad t \in [t_I, t_F] \tag{5.2}$$

These state equations and also the control variables can be under simple or complex boundary conditions, e.g. having a certain start and/or finishing value. There can also be ordinary simple constraints on the state equations and control variables meaning they have to stay between certain values but these constraints can also be of a more complex nature, i.e being a function of the states, control variables and time.

The final time, t_F, can be fixed or allowed to vary and then be a part of the optimization problem.

What TOMOC then does in order to solve the optimal control problem is to transform the differential equations, 5.2 on the previous page, into discrete form with methods such as, Euler, Runge-Kutta, Trapezoidal or Hermite-Simpson. By choosing how many segments the time, $t \in [t_I, t_F]$, should be divided into, the desired accuracy can be achieved. When making the problem discrete the integral in equation 5.1 turns into a sum with as many segments as the time is being divided into.

Another advantage with TOMOC is its ability to divide the problem into several phases with different characteristics. Each phase of the problem can then have its own cost function, dynamics and constraints. This feature will however not be used in this thesis.

There are of course also some drawbacks using TOMOC, for starter it requires a lot of computer power. Secondly, it can not handle time delays and finally, the user interface is not particularly user-friendly designed.

5.3.2 Implementation

The two control problems that are to be solved are to go from one level in engine torque to another as fast possible, or at a fixed time but with as low fuel consumption as possible. When it comes to implementing this specific problem into TOMOC there are two major challenges that has to be dealt with.

First of all, a challenge is how to describe the dynamics of the engine model used. An engine is very complex and has many different state equations, so therefore it would be of great interest to reduce that complexity by reducing the number of state equations.

Second there is the cost function. How shall the cost function be written in order to describe a real driving scenario and capture the dynamics around the change in requested torque that is of interest in this study?

States and control signals

When starting the implementation of the problem in TOMOC a simple problem already implemented served as model, however that problem only had two states and one control signal so it has to be widely extended. The original engine model described in figure 2.2 on page 8 has 13 states and 6 control signals, listed below.

$$x = \begin{bmatrix} p_{af} \\ T_{af} \\ p_{comp} \\ T_{comp} \\ p_{ic} \\ T_{ic} \\ p_{im} \\ T_{im} \\ p_{em} \\ T_{em} \\ p_{turb} \\ T_{turb} \\ \omega_{tc} \end{bmatrix} \qquad u = \begin{bmatrix} \alpha \\ N \\ Wastegate \\ \lambda \\ p_{amb} \\ T_{amb} \end{bmatrix} \tag{5.3}$$

13 states and 6 controls signals represent an optimal control problem with very high complexity so therefore it is desirable to reduce both states and control signals as much as possible. The reason for this reduction in complexity is to make the problem easier to solve in a reasonable amount of time. Since interest lies in studying how the wastegate and throttle shall be controlled when a large increase in requested engine torque appears it is rather a necessity than a limitation setting all control signals, but the wastegate and the throttle position, to constant values.

The states, on the other hand, are not that easy to reduce since the states depend on each other, meaning that if one state is removed others will be affected. Nevertheless a solution with just 8 states is found, the new control problem, with its 8 states and 2 control signals, is as follows:

$$x = \begin{bmatrix} p_{comp}[Pa] \\ p_{ic}[Pa] \\ p_{im}[Pa] \\ p_{em}[Pa] \\ T_{em}[K] \\ p_{turb}[Pa] \\ T_{turb}[K] \\ \omega_{tc}[rpm] \end{bmatrix} \qquad u = \begin{bmatrix} \alpha \\ Wastegate \end{bmatrix} \tag{5.4}$$

The rest of the states and control signals are all set to constants. Setting the temperatures on the inlet side to constant values is a fair simplification since they vary quit slow and can therefore be seen as constant during a transient. Having the pressure in the air filter set to the ambient pressure can also be seen as a reasonable assumption.

The values for all the constants are selected so that they represent a specific working point for the engine. They all come from measurements carried out in the engine research lab. This working point is also used as starting point for the optimization problem. The values are:

$$\begin{bmatrix} p_{af}[Pa] \\ T_{af}[K] \\ T_{comp}[K] \\ T_{ic}[K] \\ T_{im}[K] \end{bmatrix} = \begin{bmatrix} 99500 \\ 307 \\ 340 \\ 310 \\ 300 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} N[rpm] \\ \lambda \\ p_{amb}[pa] \\ T_{amb}[K] \end{bmatrix} = \begin{bmatrix} 3000 \\ 1 \\ 101300 \\ 293 \end{bmatrix}$$

Cost function

As mentioned earlier designing the cost function is not that straight forward. Consideration has to be taken on what actually should be punished and what consequences that will result in for the states and control signals. Normally states and/or control signals are being punished, but since there is no gain in punishing a temperature, a pressure, the wastegate position or the throttle position explicitly, another approach when designing the cost function has to be taken.

The solution here is a rather simple cost function used together with complex finish values and complex boundaries. Of interest, would be to punish the overall fuel consumption and deviations from the requested engine torque, in some cases perhaps also time. The mass flow of fuel and the engine torque can be calculated by using some of the states and is presented in [15]. Under the assumption that the temperature in the intake manifold, the engine speed and λ is constant, the fuel mass flow and torque can be calculated as:

$$W_{fuel} = \left(C_1 + C_2 \left(\frac{p_{em}}{p_{im}} \right)^{C_3} \right) p_{im} \tag{5.5}$$

$$Tq_{crankshaft} = K_1 + \left(K_2 + K_3 \left(\frac{p_{em}}{p_{im}} \right)^{K_4} \right) p_{im} + K_5 * p_{em} \tag{5.6}$$

Both friction losses and pumping losses is included in the expression for engine torque. With these calculated for every segment in the time interval, the cost function becomes:

$$\min J = \sum_{i=1}^{n} \left(C_I * W_{fuel}(i) + C_{II} \left(Tq_{crankshaft}(i) - Tq_{requested} \right)^2 \right) \tag{5.7}$$

Where C_I and C_{II} are parameters that can be adjusted in order to get a good relationship between response and fuel economy. If the end time, t_F, is a part of the optimization a part that punish time can be added to equation 5.7. In this thesis the end time will be set before the optimization starts and therefore not be an extra optimization parameter. The present system has a response time somewhere between 0.5 and 1.5 seconds so therefore is the end time in the optimization chosen to 0.5 seconds.

Constraints

In order to get the desired engine torque at the end time of the optimization complex boundary conditions can be used, meaning that there are no specific end values for specific states or control signals. Instead equation 5.8 is used, i.e. the torque at the end time must be the same as the requested one.

$$Tq_{requested} - Tq(t_F) = 0 \tag{5.8}$$

Having a requested engine torque as final value means that the states p_{im} and p_{em} together must fulfill specific final values that depends on the chosen final torque.

Having only a requested engine torque at the end time together with the cost function described in equation 5.7 can under certain conditions generate a system that at first drives the torque toward zero and then, as late as possible, generate the specified end torque.

This is because such a behavior would minimize the overall fuel consumption. Of course this is not an acceptable behavior, therefore a complex path constraint can be added, forcing the torque to stay above a certain level. In this case the level is chosen to be slightly under the engine torque generated at the optimization starting point. This kind of behavior can occur when the punishment on fuel consumption is very high and the solution is presented here just to show how that problem can be overcome, although it is not a problem for this thesis.

Constraints are also forced on the states and control signals to make sure they do not do anything physically obscene. The constrains chosen can be seen below:

$$\begin{bmatrix} 80000 \\ 80000 \\ 80000 \\ 20000 \\ 273 \\ 80000 \\ 273 \\ 80000 \end{bmatrix} \leq \begin{bmatrix} p_{comp} \\ p_{ic} \\ p_{im} \\ p_{em} \\ T_{em} \\ p_{turb} \\ T_{turb} \\ \omega_{tc} \end{bmatrix} \leq \begin{bmatrix} 400000 \\ 400000 \\ 400000 \\ 400000 \\ 2500 \\ 400000 \\ 2500 \\ 200000 \end{bmatrix}$$

$$\begin{bmatrix} 10 \\ 0 \end{bmatrix} \leq \begin{bmatrix} alpha \\ Wastegate \end{bmatrix} \leq \begin{bmatrix} 1500 \\ 1 \end{bmatrix}$$

It is necessary to have these constraint in order to get a meaningful optimization.

Expected results

This way of implementing the optimization will hopefully show how the wastegate, but also the throttle, shall be controlled in order to get a feeling for how response time and fuel consumption work together. Worth mentioning is also that the wastegate is being considered as an ideal actuator and can thereby be opened and closed without any limitations. Meaning that the results here can not directly be transformed to how the situation is on a real engine.

5.3.3 Results

The results from some optimizations done with TOMOC is to be seen in figure 5.1 on the next page. In the graphs there are the results from three different optimizations carried out all in the same way but with different cost functions. It is only parameter C_{II} in equation 5.7 on page 40 that has been altered between the different optimizations.

The optimizations is being carried out with the reduced number of states and control signals that is mentioned in 5.3.2. The constants used instead are also described in 5.3.2. The initial working point generate an engine torque equal to 68 Nm and it is supposed to reach 168 Nm at the end time of the optimization that is set fix to 0.5 seconds. Meaning that the engine torque shall increase 100 Nm in 0.5 seconds, from 68 Nm to 168 Nm. The number of segments the time interval is divided into is nine since it gives a good result in reasonable time. A complex path boundary, as mentioned earlier, is added to make sure the engine torque stays above 60 Nm during the entire optimization. As can be seen in figure 5.1 on the next page, TOMOC manage to fulfill the constraints as well as reaching the requested torque in time.

During all three optimizations $C_I = 5 * 10^4$ in equation 5.7 and for the solid line $C_{II} = 1 * 10^{-2}$, for the dash-dotted line $C_{II} = 0.5 * 10^{-2}$ and for the dashed line $C_{II} = 2 * 10^{-2}$. As can be seen in the left lower graph a higher value on C_{II}, i.e. a higher cost for not reaching the requested torque as fast as possible, results in a higher torque faster, as can be expected. But this higher torque is reached to the cost of increased fuel flow, i.e. a higher over all fuel consumption, that can be seen in the lower right graph.

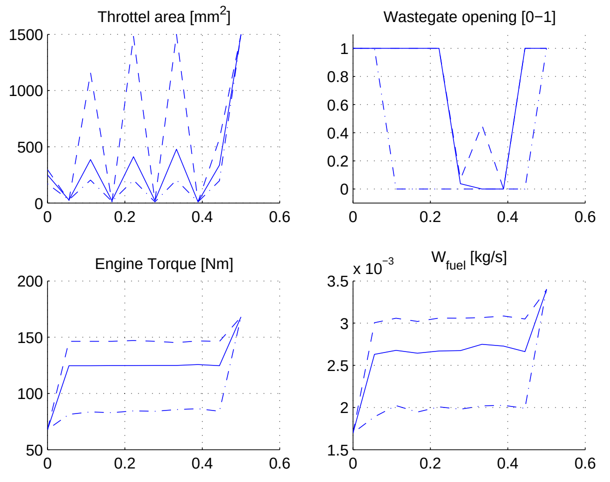

Figure 5.1: Results from three optimizations with TOMOC. The top graphs show the control signals and the two bottom graphs show how the control affects the engine torque and the fuel consumption. C_{II} from equation 5.7 has for the dashed dotted line its lowest value and for the dashed line it highest.

The both upper graphs in figure 5.1 show the two control signals, throttle area in mm^2 and wastegate opening where zero represent a fully closed wastegate and one represent it being fully open.

5.4 Conclusions

From how the optimal control problem was implemented in TOMOC described in 5.3.2 and the results presented in 5.3.3 some conclusions can be drawn.

Throttle area
All three optimizations shows that it is optimal for the throttle to open as much as possible at the end to reach the requested torque. The oscillations that can be seen in throttle area, especially in the case with higher cost on the torque, is probably because there are too few segments.

One way to lower the oscillations would be to put a punishment on fast changes in the control signals. Another way would be to use more segments since it gives better resolution and higher accuracy. There has however been some problem using too many segments and is probably because it then finds some local optimums instead of the global optimum. However the tendency for the throttle area is clear, less opened at the beginning and as late as possible open it to the maximum.

Wastegate
It can be seen in all three cases that it is optimal in the beginning to have the wastegate open. Then the wastegate shall be closed, for how long time depends on the level of the engine torque, the higher torque the shorter time it needs to be closed. The wastegate is probably held closed to build up turbine speed and boost pressure before the final leap in engine torque. The bigger leap the longer the wastegate is needed to be held closed. At the end of the optimization the wastegate is again opened, probably in order to reduce the fuel consumption in the last segments of the optimization.

Normally the wastegate shall be kept open, in order to save fuel, but these simulations indicate that if the engine knows that it shall deliver a certain amount of extra torque in a certain amount of time it can be more fuel economic to keep the wastegate closed.

Engine torque
A higher cost for deviation in the engine torque from the requested torque results as expected in a higher level for the torque. This effect can also be seen if the parameter C_I changes value, an increase in C_I will result in a lower level for the torque. It is not the actual values of C_I and C_{II} that determines the level of the torque, it is the relationship between them.

Fuel flow
The higher the cost for deviations in engine torque is, the higher the fuel consumption is.

Chapter 6

Conclusions

The intention with this chapter is to summarize and extend the conclusions drawn in the preceeding chapters. Basically everything of importance from this thesis should be found here whereas implementations etc. can be found in the other chapters.

6.1 Bypass implementation

According to the validation parameters mentioned in chapter 3, i.e. pressure and flow in the manifold after the airfilter, pressure and flow in the manifold after the compressor, the bypass model is correctly built and implemented in the original engine model. Furthermore the parameter that controlls the opening area for the bypass valve had to be determined. This was done by roughly estimating the area through geometric measurements and thereafter by adjusting and fine tuning through simulations with data in Simulink.

The model and especially the area parameter is thereafter validated with new data from measurements. All other models come from [6] so no further validation is needed. On top of that the performance of the model and the correctness is very good, it can therefore be looked upon as a correctly implemented bypass valve and the end of assignment one.

When it comes to the alternative implementation of a bypass valve, that dumps the gases to the atmosphere there is one important feature which must not be forgotten. Even though it looks rather nice and is easier to hide away in other Simulink blocks it is not a correct model of the reality. It has in many ways the same behavior as the final implementation used but it is important to remember that the alternative model do not recirculate the air and the temperature of it and therefore there is a loss of energy in the system.

6.2 Surge control

According to the tests and simulations performed in chapter 4 the proposed controller works in a proper way in an off-line environment. The controller has the advantage that it keeps a higher pressure and a higher torque for a longer time than the present control system used in the research lab. In some cases the gain in torque is as high as five percent which of course means a lot to a driver when shifting gears.

The possibilities and advantages that comes when using a continuous controller instead of the proposed discrete controller are also shown. A continuous controller in combination with no time delays, i.e. an ideal controller can give as much as an extra 12-14 percent torque while shifting gears.

The controller can be used in a real-time environment and has been compiled in Matlab's Real-Time workshop. When used in real-time in the engine lab there is possible to use the predicted air mass flow as input to the controller and thereby have an even faster system that can solve the control issue faster and better.

The controller is a simple P-controller and is therefore very robust and easy to implement in an existing engine management system, if needed. So the second assignment is fulfilled. There are however other possibilities when it comes to controlling the valve and they will be discussed in the chapter Future work, i.e. chapter 7.

An important observation is that it was hard to control both the opening and closing of the bypass valve with a P-controller using the air mass flow as reference. Therefore there is a need for some sort of predicted air mass signal that makes it possible to open the bypass valve early.

6.3 Wastegate

The present idea of monitoring the wastegate valve is having it open in order to save fuel and having it closed in order to reduce the turbo lag. It is therefore a choice the manufactures have to make when designing the performance of the car.

In 5.3.3 on page 42 the indications are however that in certain driving scenarios it can be economic to close the wastegate to save fuel. Fuel can be saved since closing the wastegate can lead to a later opening of the throttle. This is of course when the engine knows that a higher torque will be requested in, for instance, half a second. It is therefore of great interest to study this further to get a more general idea how to control the wastegate in an optimal way. The optimizations carried out here are rather rough and must be refined before any final conclusions can be drawn.

Results from these tests also indicate that some kind of central control for the wastegate and the throttle can be beneficial for the relationship between fuel consumption and turbo lag. Therefore this ought to be investigated further, and in chapter 7.2.2 and 7.2.3 on page 52 some controller concepts will be introduced that are of particular interest to this case.

Chapter 7

Future Work

7.1 Bypass and surge control

Since the bypass-/surge controller never was tested in reality there is an obvious road to continue on but there are also other things that can be interesting to look into and they will all be presented here.

7.1.1 Testing

The controller that was presented in chapter 4 showed great improvements to the engine power output in simulations. There was also in that chapter a description of how a real time implementation of the model is done so the only thing needed in order to be able to use the controller in real life is an observer that gives the air mass flow through the compressor.

7.1.2 Improvements

When it comes to the controller developed it is just a P-controller and the gain K_p is only adjusted to one specific engine working point. An optimization of K_p should therefore be done so that the controller works as good as possible over the engines whole operating area.

There would perhaps be possible to gain even more in torque if a complete PID-controller was used and that is absolutely an interesting subject to investigate. The few tests conducted in this thesis can be improved and the parameters can be tuned even better.

Instead of testing with an ideal actuator at once an initial and easy attempt should be to shorten the length of the hose connecting the actuator and the bypass valve and extending the other hoses instead. To make the complete model a bit more good looking there would be an idea to put it all in one and the same block in simulink, i.e. make a subsystem of the adiabatic mixer and the bypass block.

7.1.3 Alternatives

If there were no time delays in the system, i.e. having an ideal actuator, there would perhaps be possible to use an ordinary PID controller that totally controlls both the opening and closing of the bypass valve.

Another interesting alternative would be to try and control surge with help from both the air mass flow and the pressure quotient, Π.

7.2 Wastegate

In this chapter two interesting alternatives, called MPC and NMPC, for control of the wastegate will be introduced. They are closely connected to each other and they are also probably possible to use in on-line control of an engine but the chapter will be started off with a discussion on Tomoc.

7.2.1 TOMOC

As mentioned earlier the optimizations that have been done are rater rough, i.e. there are few segments and that means lower resolution and accuracy. Therefore a natural extension of the work would be to add more segments to the control problem. The results presented in this thesis also have to be tested for how robust they are. Of course some fine tuning of the entire problem also can be done since no time has been spent on making it 100 percent realistic.

To get a better understanding of how the wastegate and throttle should be controlled in more general terms, several other optimization scenarios have to be carried out. Of great interest would be to see how wastegate and throttle can work together in order to reduce fuel consumption or the turbo lag.

Of interest would also be to add the actual limitations and constraints that the present wastegate valve operates under in order to see what can be done with the present hardware.

7.2.2 Model predictive control

Of all the modern control strategies that exist, Model Predictive Control or MPC probably is one of few that really has been implemented and continuously used in the industry. It is mostly used in processes where sampling times measured in minutes or hours are no problem since it takes a lot of computer power to do all calculations that are necessary. However, with todays computers that should be a problem that can be solved and according to [5] even an anti spin system on a vehicle has successfully been implemented. Two of the reasons for this method to become so popular are that it easily explained to process operators and the fact that constraints on the control signal and the output signal can explicitly be handled. For a more intricate explanation of MPC, [4] and [5] is recommended as an introduction.

Basics

The Basic idea with MPC is to formulate the control problem as an optimization problem and to solve that problem on-line. The system has to be represented on state space form and in discrete time i.e.:

$$x(k+1) = Ax(k) + Bu(k) \tag{7.1}$$

$$y(k) = Cx(k) \tag{7.2}$$

There are several other variants as well, such as model predictive heuristic control and dynamic matrix control mentioned in [12]. The problem to be solved when talking about state space MPC is the minimization of the function:

$$\sum_{j=0}^{N-1} \|y(k+j+1) - r(k+j+1)\|_{Q_1}^2 + \|u(k+j)\|_{Q_2}^2 \tag{7.3}$$

Where N = prediction horizon, k = time for which the problem has to be solved, Q_1 and Q_2 are weight matrices, y = output/output vector, u = control signal/ control signals, r = reference signal/ reference signals. This means that the objective is to minimize the difference between the output and a reference signal but at the same time minimize the control signal. An example would be to minimize the difference between actual torque from an engine and a reference signal and at the same time minimize the throttle position and wastegate position. The sequence of control signals, $u(0), u(1)...u(N-1)$ are the ones that can be optimized with optimization techniques and thereby minimizing the expression 7.3. An algorithm for solving that problem would look like this:

1. Measure $x(k)$ (or use an observer to estimate it)
2. Minimize 7.3 and thereby get the control sequence, $u(k)...u(N-1)$
3. Use the first of the signals in the sequence, $u(k)$
   (or the first signal vector, if there a several control signals)
4. Increase the time, $k := k+1$
5. Start over from step one

What it takes and what can be done

The reason for mentioning this method is that when trying to control the response time on a turbocharged engine with the wastegate and throttle, constraints on the wastegate signal and throttle signal has to be taken into account. The constraints make it difficult to use any other modern control method.

In order to implement a MPC-controller a linearized model of the engine is needed, but that is a problem for all modern control methods. They are nonlinear methods in theory but in practice almost all models have to be implemented as linearized models.

When such a linearized model exists it is possible to optimally control the wastegate and the throttle at the same time. Optimally means that as little fuel as possible is used for a fixed response time. Ideally such a controller could probably controll the wastegate, bypass and throttle, i.e. all the air actuators, at the same time and thereby increase the engines performance.

Drawbacks and challenges

A challenge to be faced is the creation of a linearized engine model which is easy to use and implement. Such a model is not that easy to create and will have to be done for a number of engine operating points. That means that an extensive engine map consisting of the linearized model have to be implemented in the engines control system. A search function also has to be implemented so that the MPC- controller easily can find out which operating point that is closest and therefore should be used.

If that challenge can be overcome, then MPC is a promising method and due to the fact that it handles constraint better than any other method, probably the method most likely to be possible to use.

Another challenge is the extensive computer power needed. This can probably be overcome with a smart implementation of the controller and the fact that computers get faster and more powerful every month.

7.2.3 Nonlinear model predictive control

This part is based on an article, [12], written by a number of French scientists and published by SAE. The reason for discussing it here is that it offers a solution to the challenges mentioned earlier on when discussing MPC.

Introduction and background

A disadvantage with turbochargers is the turbine inertia that results in a long response time before reaching the supercharging pressure. This problem would be possible to solve with prediction of the needed pressure and a control which can make use of that prediction. Model predictive control offers both.

Since most processes are nonlinear and the linearizations are often not good enough, there is a need for nonlinear model predictive control, NMPC. A method to model processes that has become more and more popular in recent years is neural networks. It has the ability to model non-linear systems with flexibility and arbitrary accuracy and it is often less time consuming than building a physically interpretable model.

Combining the two methods, NMPC and neural networks, leads to a method which the authors of [12] calls Neural Predictive Control. This method has been developed for and tested on a turbocharged spark ignited engine with good result. The method also makes it easy to generate linear models of the non-linear problem. Even though it is a non-linear method the implementation has to be done with a linearized model since it has to be implemented in real time.

The neural network used can be trained with some training method available, in the article the Levenberg-Marquardt method was used. Five different engine speeds and number of steps in throttle and wastegate position were measured and then used for the training of the neural network.

After the training and the building of a control scheme, it was time for simulations and for that purpose three different neural predictive controllers were used. Two of them were ruled out because of their poor computational times, the third, called saturated linearized neural predictive control, SLNPC was tested on an engine test bench. The result was satisfying even though the tests were performed on an engine speed for which the neural network was not trained.

Possibilities

It is perhaps a bit unrealistic to implement such a controller in an engine control system today, but there is clearly benefits from such a method. The problem with linearization can be solved through the implementation of a neural network and as mentioned earlier on in this thesis the use of MPC is perhaps the most promising modern control method.

References

[1] Lars Nielsen, Lars Eriksson
Course material vehicular system
Bokakademin Linköping 2004

[2] Lars Nielsen, Lars Eriksson
Laborationskompendium Fordonssystem TSFS05
Bokakademin Linköping 2004

[3] Torkel Glad, Lennart Ljung
Reglerteknik. Grundläggande teori
Studentlitteratur, Lund 1989

[4] Torkel Glad, Lennart Ljung
Reglerteori. Flervariabla och olinjära system
Studentlitteratur, Lund 1989

[5] Glad, Gunnarsson, Ljung, McKelvey, Stenman och Löfberg
Digital Styrning, Kurskompendium
Bokakademin Linköping 2003

[6] Simon Frei, Lars Eriksson
Modelling of a Turbocharged SI Engine
IMRT Technical report 2001

[7] Jensen, Kristensen, Sorenson and Houbak
Mean Value Engine Modeling of a small Turbocharged Diesel Engine
SAE International 1991

[8] Müller, Hendricks, Spencer and Sorensen
Mean Value Modeling of Turbocharged Spark Ignition Engines
SAE International 1998

[9] Sorensen, Hendricks, Magnusson and Bertelsen
Compact and Accurate Turbocharged Modelling for Engine Control
SAE International 2005

[10] Bram de Jager
Rotating stall and surge control: A survey
IEEE 1995

[11] Frank Willems and Bram de Jager
Modeling and control of Rotating stall and surge control: An Overview
IEEE 1998

[12] Colin, Chamaillard, Charlet, Bloch and Corde
Linearized Neural Predictive Control, A Turbocharged SI Engine Application
SAE International 2005

[13] Adam Lagerberg
Open-loop optimal control of a backlash traverse Chalmers University of Technology 2004

[14] Jan Tommy Gravdahl
Modeling and Control of Surge and Rotating Stall in Compressors
Report 98-6-W, Norwegian University of Science and Technology, 1998

[15] Per Andersson
Air Charge Estimation on Turbocharged Spark Ignition Engines
Ph D thesis, Linköpings Universitet (to be published in 2005)

[16] John L.Lumley
Engines, An introduction
Cambridge University Press 1999

[17] V.A.W Hillier
Fundamentals of MOTOR VEHICLE Technology 4th edition
Stanley Thornes (Publishers) Ltd 1991

[18] Heinz Heisler
Vehicle and Engine Technology
SAE International 1999

[19] HowStuffWorks inc.
http://www.howstuffworks.com
2005-05-13

Appendix A

Experimental setup

In this appendix the environment where all tests were conducted will be described. The research laboratory is located at Linköpings Universitet and consists of an Engine test cell and a control room. For more information on the research laboratory [2] is recommended.

A.1 Engine test cell

The engine test cell is equipped with two spark ignited engines from SAAB. One of the engines has variable compression and the other is turbo charged.

An electrical dynamometer is connected to each engine so that the load can be adjusted with good accuracy. They are also used for starting up the engines since no start engine is connected. The dynamometers are controlled by an individual control system which includes protection of the engines from overrun and overload.

A.2 Control room

Two computers are used in the control room, one for communication with the engines control system and one for control of instruments used for measurements and in some cases it is also used for control of the dynamometers.

Normally the dynamometers are controlled from a user interface called X-ACT. It is possible to control either two of the following three parameters at the same time:

- Throttle position
- Engine speed
- Break Torque

In most cases the throttle position and the engine speed is set to fixed values and the resulting break torque is observed. Controlling the dynamometers from one of the computers is basically only done when making an engine map.

A.3 The Engine

For this thesis, the turbocharged engine in the engine test cell mentioned earlier has been used. It is a L850 engine which is a four stroke, two liter, turbocharged engine from SAAB with four cylinders. The engine is equipped with a control system from SAAB called Trionic 9 which is a prototype system used for research and has never been taken into large scale production. The engine is with few exceptions the same as used in SAAB 9^3 aero today.

For measurements the standard sensors mounted on the engine are used but there has also been extra sensors mounted on the engine, e.g. a sensor measuring the turbine rotation speed. All data from measurements are saved on a computer and can thereafter be viewed in for instance Matlab.

Appendix B

Introduction to Supercharging and Turbocharging

B.1 Superchargers

In order to increase an engines performance one would like to have air with as high density as possible coming into the cylinder. Increasing the inlet air density can according to [16] be done either by manifold tuning, supercharging or turbocharging. This text will be focused on supercharging and especially turbocharging. Reading literature on supercharging and turbocharging can be a bit confusing since authors use different definitions.

On the web page HowStuffWorks [19] they refer to supercharging and turbocharging as two different forced induction systems where the supercharger get its power supply from the engine through a belt and the turbocharger get its power supply from the exhaust stream.

The author of the book *Engines, An Introduction* John L. Lumley [16] is of the opinion that supercharging used to be the generic name for using mechanical devices in order to increase the inlet density but that the word nowadays has a slightly different meaning. Nowadays, he says, supercharging refers to compressors which has no connection between inlet and outlet, i.e. the air is taken into a chamber which then is closed and thereafter one reduces the volume of the chamber and when enough pressure is reached the chamber outlet is opened. According to Lumley the turbocharger has a compressor with a direct connection between inlet and outlet so when it is not operating there is no pressure difference over it.

The general opinion, i.e. the one which most authors share, is that supercharging is the generic name for air pumps of some sort which increases the inlet air charge density. This definition is used by Nielsen and Eriksson in [1], by Heisler in [18] and by Hillier in [17]. They all make, even though they use different names, a distinction between two different types of superchargers:

Supercharger type I.
Positively driven or mechanical driven superchargers. With that they mean superchargers driven by belt, chain or gear from the engines crankshaft. Under this category supercharges like the Roots blower, the (sliding) vane compressor and the centrifugal blower/compressor are sorted.

Supercharger type II.
Non positive driven superchargers. All superchargers in this category is driven by the energy in the exhaust gases. To this category the turbocharger belongs and also the not so well known pressure wave supercharger.

The advantage with the positively drive supercharges is that they are able to deliver boost power even for low engine speeds. The disadvantage is that they consume engine power and often the fuel consumption is distinctively increased. There are also electric driven compressors which not directly steel power from the engine but they are of little interest since they are expensive.

The rest of this text will be dedicated to one of the non positively driven superchargers, the turbocharger. The advantage with the turbocharger is that it does not directly consume engine power since it gets its energy from the exhaust gases. The disadvantage is the fact that it is not able to deliver boost power for low engine speeds but despite that, it has become the most widely used supercharger.

B.1.1 The Turbocharger

The basic turbocharger is made up by a radial turbine wheel and a centrifugal compressor wheel mechanically connected through a shaft. When fully operating the shaft spins with a speed of more than 90 000 rpm. The shaft is, in order to diminish friction losses, mounted on floating bearings which are supported with clean oil from the engine's lubrication system. For the turbocharger this lubrication is of great importance and any flaws in oil pressure, or if there is dirt in the oil, will soon put an end to the turbocharger. The oil also serves as cooling for the surrounding walls so the oil has to be able to exit the shaft case and have a free way to the engine sump so that it can be recirculated.

The turbine side

The turbine side consists of a heat resisting wheel, normally made of nickel-based alloy mounted in a cast iron casing. It has an inlet where exhaust gases with a temperature of about 900-1000˚C and a high velocity are able to enter. Usually a radial flow turbine is used, i.e. the air enters from the side and exits in a direction which has a 90 degree angle to the entering direction.

An important feature on the turbine side of the turbocharger is the exhaust gas bypass valve¨, also known as the boost limiting valve, or as most people know it, the wastegate. It is mounted so that by opening the valve, some of the exhaust gases will not pass by the turbine wheel but go directly to the catalytic converter and thereby not contribute to the speed of the turbine wheel. The reason for doing this is that with a lower speed of the turbine wheel, and thereby a lower speed of the compressor wheel, one can avoid building up a too high pressure in the manifolds before the cylinders.

A too high pressure in the intake manifold could lead to damage on the cylinders through knock but it also leads to high emissions of NO_x, so the pressure has to be limited.

The compressor

At the other side of the turbine shaft there is a impeller wheel also called a compressor wheel made out of aluminum-alloy. The compressor is a so called centrifugal blower, i.e. the air is entering trough an inlet which is perpendicular to the direction of the blades.

The air is sucked into the space between the blades and is then subjected to a centrifugal force which forces it toward the outer side of the wheel. The air is thereafter forced into the area between two parallel walls at the outer side of the compressor called diffusers where the velocity energy is converted into pressure energy. This high pressure air is then collected in a discharge involute which encircles the diffuser. The shape of this involute makes sure that the air is finding its way to the manifold connected to the compressor.

The intercooler

Another important part that has to be mentioned when talking about turbocharging is the charge-air coolers or intercoolers. When the charged air leaves the compressor it has a high temperature and therefore a low density. This is a problem since the engines performance depends on the amount of air that can be inducted into the cylinder at each stroke.

In literature there are two different types of intercoolers to be found:

- Air to liquid intercooler
- Air to air intercooler

Air to liquid intercoolers uses the coolant liquid from the engine cooling system to lower the temperature of the charge air. They are able to lower air from 150˚C down to about 85˚C. The most frequently used intercooler however is the air to air intercooler.

The air to air intercooler is in most cases mounted directly in front of the engine radiator so that air is drawn trough it by the engine fan. It has the capability to lower air from about 120˚C to about 60˚C and that lower temperature is the reason for them to be more popular than the air to liquid intercoolers. A lower temperature means higher volumetric efficiency but also less risk for knocking.

Worth knowing about turbochargers

When adding a supercharger to an engine other components in the engine and on the car have to be strengthened in order to withstand the extra pressure, the higher loads and higher speeds.

When adding a turbocharger one also have to lower the compression ratio in order to avoid knocking in the engine. Some engine manufactures also uses an ignition timing system to better fit the ignition with the boost pressure and thereby avoiding knock.

The advantages with turbocharging are among many things the following:

- Better engine performance from a small engine.
- Lower exhaust noise and emissions.
- Better fuel economy than a large scale engine with the same performance.

The disadvantages with turbocharging are also easily listed:

- High repair and service costs. When something in the turbocharger brakes it will most likely affect the rest of the engine.
- Slow response when stepping on the gas pedal. Mostly a problem in older cars.

Usually the turbocharger is matched to the engine so that it can give a maximum boost pressure of about 1.5-2.0 bar, i.e. the total intake manifold pressure is the atmospheric pressure times 1.5-2.0.