CHALMERS

Model Predictive Control of Engine Idle Speed

Master of Science Thesis

LENNIE EDMAN
PATRIK LJUNGVALL

Department of Signals and Systems
Division of Automatic Control
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden, 2007
Report No. EX100/2007

Abstract

The aim with this thesis work was to investigate if there is a possibility to reduce the torque reserve in vehicles during idle conditions using a model predictive controller (MPC) instead of the present PID controller. The reason for having a torque reserve is to prevent the engine speed from dipping to an unacceptable level, caused by unexpected disturbances, but since the torque reserve also affects the fuel consumption it is important to keep it as low as possible.

The MPC was designed based on a black-box system identification of the engine air mass flow dynamics of the intake system. Measurements were performed in a rig and the identification results were later on compared to an engine model (MODEC).

The results showed that the torque reserve could be reduced from 8 Nm to 5 Nm without violating any constraints. Thus, replacing a PID controller with a model predictive controller is considered to be plausible. However, one must keep in mind that the results are based on an engine model and not on a real vehicle.

Acknowledgements

We would like to thank all the colleagues at VCC, department 97542 for help and support during this thesis work. A special thanks to Hans Englander and Marcus Rubensson for solving the problems with the MODEC engine model and to Anders Jonsson for helping us out with the FPD-rig measurements. We would also like to thank Professor Bo Egardt for guidance and Jan-Ola Olsson for making this interesting and rewarding project possible.

Preface

This report is the documentation of the thesis work done as part of the Master of Science degree in Electrical engineering at Chalmers University of Technology, Gothenburg, Sweden. The project was performed at the department of Complete Powertrain (97542) at Volvo Car Corporation, Gothenburg, Sweden and examined by the department of Signals and System at Chalmers University of Technology. The project time reached from 23^{rd} of April to 1^{st} of November 2007.

Thesis outline

- **Chapter 1 - Introduction**
  Problem formulation, used methods along with the framework for the thesis.
- **Chapter 2 - Modelling**
  Description of the system that is the subject for the control strategy.
- **Chapter 3 - System Identification**
  Identification of the intake air dynamics through statistical models.
- **Chapter 4 - Model Predictive Control**
  Construction of a Model Predictive Controller along with the results of this controller controlling the system and a comparison to the existing PID controller.
- **Chapter 5 - Conclusions and future work**
  Closer discussion of the results and recommendations to future work.

Notations

All quantities in this thesis are given in SI-units unless else stated. Abbreviations and variables are used to simplify explanations and calculations. These abbreviations and variables are presented in the tables below.

Abbreviation	**Explanation**
ARMAX	Auto Regressive Moving Average model with eXogenous inputs
ARX	Auto Regressive model with eXogenous inputs
CPU	Central Processing Unit
ECR	Equal Concern for the Relaxation
FP	FunktionsProvning (Functional testing)
FPD	FunktionsProvning Dynamisk (Functional testing Dynamic)
HIL	Hardware In the Loop
I/O	Input/Output
INCA	Integrated Calibration and Application tools
ISC	Idle Speed Control
LPF	Low Pass Filter
LSQ	Least Square
MBT	Maximum Brake Torque
MIMO	Multi Input, Multi Output
MODEC	MODels for Engine Control
MPC	Model Predictive Control
MV	Manipulated Variables
MVEM	Mean Value Engine Model
n.a.	Not available
OV	Output Variables
PID	Proportional Integral Derivative
PPP	Peak Pressure Position
PRBS	Pseudo Random Binary Signal
SI6	Short Inline 6
SISO	Single Input, Single Output
TDC	Top Dead Centre
VCC	Volvo Car Corporation
VVT	Variable Valve Timing

Table 1: Abbreviations used in the thesis.

Nomenclature	Explanation	Units
ϵ	Relaxation	
$\eta_{ign,ch}$	Gross indicated efficiency (including chamber losses)	
η_{ign}	Efficiency corresponding to the ignition angle	
η_{vol}	Volumetric efficiency	
γ	Ratio of specific heats	
λ	Air/Fuel-ratio	
μ	Mean value	
Ω	Observability matrix	
ω_B	Cut-off frequency	rad/s
ω_e	Engine speed	rad/s
$\dot{\omega}_e$	Differentiated engine speed	rad/s^2
ω_N	Nyquist frequency	rad/s
ω_s	Sampling frequency	rad/s
σ	Standard deviation	
τ_{63}	Time constant of the system	s
τ_{th}	Time constant for the throttle response	s
τ_d	Ignition delay	s
θ_{ign}	Ignition angle	deg
$\theta_{ign,opt}$	Optimum ignition angle	deg
ζ	Controllability matrix	
C_{f0}, C_{f1}, C_{f2}	Constants derived from the friction mean effective pressure	
c_0, c_1, c_2	Constants for the volumetric efficiency	
$e(t)$	Gaussian white noise	
$G(q, \theta)$	Transfer function of the system	
H_c	Control Horizon	
H_p	Prediction Horizon	
$H(q, \theta)$	Transfer function of the disturbances	
J	Moment of inertia	$kg \cdot m^2$
M	Maximum length PRBS	
m_f	Fuel mass in the cylinder	kg
$\dot{m_{fc}}$	Fuel mass flow into the cylinder	kg/m^3
$\dot{m}_{ac}$	Air mass flow into the cylinder	kg/m^3
$\dot{m}_{at}$	Air mass flow past the throttle plate	kg/m^3
$\dot{m}_{at,ref}$	Reference air mass flow past the throttle plate	kg/m^3
N	PRBS clock period	
N_e	Engine speed	rpm

Table 2: Nomenclature used in the thesis.

Nomenclature	**Explanation**	**Units**
n_{mv}	Number of manipulated variables	
n_r	Number of engine revolutions per cycle	
n_y	Number of plant outputs	
p_{cyl}	Pressure in the cylinder	Pa
p_{em}	Exhaust manifold pressure	Pa
p_{im}	Intake manifold pressure	Pa
q	Discrete shift operator	
q_{HV}	Specific heat value for the fuel	J/kg
R	Specific gas constant	$J/(kg \cdot K)$
r_c	Compression ratio	
$r(t)$	Reference trajectory	
$s(t)$	Set-point trajectory	
T_{im}	Intake manifold temperature	C
T_s	Sampling interval	s
$TqBase$	Torque delivered by the air dynamics in the intake system	Nm
$TqBaseTgt$	Desired torque delivered by the air dynamics in the intake system	Nm
$TqBrkBase$	Break base torque	Nm
$TqBrkInst$	Break inst torque	Nm
$TqInst$	Torque corresponding to the ignition positioning	Nm
$TqInstTgt$	Desired torque corresponding to the ignition positioning	Nm
$TqOut$	Output engine torque	Nm
$TqRsv$	Torque reserve	Nm
$u(t)$	Input signal	
V_d	Displaced volume (one cylinder or whole engine)	dm^3
V_{im}	Intake manifold volume	dm^3
w^u	Manipulated variables weights	
w^y	Output variables weights	
W_{ig}	Indicated gross work per cycle	[J]
W_f	Friction work	[J]
W_p	Pumping work	[J]
$y(t)$	Output signal	

Table 3: Nomenclature used in the thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter a brief background and a motivation behind this thesis work is presented. The methods used to perform the work and the framework for this thesis are found in this chapter along with a problem formulation divided into three parts; formulation of modelling problem, formulation of control problem and formulation of simulation problem.

1.1 Motivation and background

The fact that vehicles now are computerized machines has an enormous effect on the possibilities for functionality of vehicles. This fact together with requirements and needs from customers and society have caused energetic activities in the field of research and development. New mechanical designs are made possible by the existence of control systems. Together with the upcoming availability of network and computing technology, completely new areas has been opened revealing many interesting possibilities [Eri05].

Emissions from the sector of transports are considered one of the major contributors to pollution and global warming. There is a need to develop alternative and more efficient ways of vehicle control and propulsion since there is an increasing consumption of limited energy sources such as oil [Pet06]. Legislation demands in the emission area together with increasing oil prices, forces the industry to find more efficient solutions.

In this thesis a somewhat new control strategy, namely model predictive control (MPC), has been constructed and compared to the present Proportional-Integral-Derivative (PID) controller. The reason for constructing the MPC was to investigate if there is a possibility to reduce the fuel consumption during idle speed using this new control strategy instead of the present PID.

1.2 Method and limitations

The procedure of this thesis was as follows

- First a literature study was performed in order to understand the physical

model of the system which was going to be identified and later on controlled.

- The behaviour of the present PID controller, controlling the Models for engine control (MODEC) was studied. MODEC is a model developed at Volvo Car Corporation, that is used for simulations in the Matlab/Simulink environment.

- A literature study on the field of system identification was performed.

- Tests in vehicle, FPD-rig and on MODEC engine model were performed on the intake air dynamics in order to obtain valuable input/output (I/O) signals for the forthcoming system identification.

- A system identification process of the intake air dynamics was performed in System Identification Toolbox in the Matlab environment.

- Resulting models from the system identification process were evaluated and a proper model was selected for the controlling task.

- A literature study on the field of model predictive control was performed.

- A model predictive controller was created in Model Predictive Control Toolbox in the Matlab environment.

- The model predictive controller along with its control strategy was simulated in Simulink environment against a simplified model and the MODEC engine model.

- An Optimization algorithm was applied to the controller parameters.

- Finally, the simulation results were evaluated and the system behaviour under influence of the MPC was compared to the PID controller.

Limitations

Some limitations had to be made in order to meet the time aspect and complexity of the project:

- The procedure for collecting data in the FPD-rig, later used for system identification of the intake air mass flow dynamics, was kept short and simple due to a very limited amount of time available for testing.

- The controller parameters in the model predictive controller were not optimized.

- The work was to develop a test version of an MPC used for evaluation of the method in Simulink environment simulated against a virtual engine model and not to implement an MPC for use in a vehicle.
- The MODEC engine model was improved during the project which led to changed conditions for the analysis and experiments.

1.3 Problem formulation

This thesis has three major parts. The first part contains modelling and a study of the system. The second part is to develop a control system, MPC, for controlling the engine speed and the torque reserve during idle speed. The third part concerns the evaluation of the models and the control strategy. The evaluation is to be done by simulations of the system.

1.3.1 Formulation of modelling problem

Models are to be developed to represent the intake air flow dynamics and the ignition positioning. These models are later on used as an important part of the model predictive controller, namely for prediction of future plant behaviour. The models will be developed in Matlab and the Matlab/Simulink environment.

1.3.2 Formulation of control problem

A control system is to be developed for controlling the engine speed and the available amount of torque reserve during idle conditions. The controller shall be based upon model predictive control theory. The torque reserve is to be kept as small as possible without having a major impact on the controller performance.

The control system is to be implemented together with the system models in Matlab/Simulink environment.

1.3.3 Formulation of evaluation problem

The aim of the simulations is to evaluate the developed models of the system and the control strategy. The main outputs are the engine speed and the torque reserve. It is also important how the system behaves under influence of different kind of disturbances. The engine speed and the torque reserve for the model predictive controller should be compared to PID controller ditto. The model predictive controller should also manage to handle the same magnitude of the disturbances as the existing PID controller.

Chapter 2

Modelling

To be able to understand the system, which is going to be modelled and also later on controlled, a brief system overview is needed. This chapter provides such an overview which includes the different subparts of the total system that will be focused on. More detailed parts which will be presented are the air dynamics, ignition positioning and the creation of a torque reserve. These parts are divided into several subsections which will be of interest in for instance the forthcoming system identification. As main reference in this chapter, Lars Eriksson and Lars Nielsen's "Modeling and Control of Internal Combustion Engines" has been used.

2.1 Air dynamics

One of the major and vital parts in the engine is the air path or in this thesis called the air dynamics. This part in turn consists of two different parts namely the air flow past the throttle plate and the air flow and pressure characteristics of the intake manifold. Modelling of these areas will be processed in the following sections.

2.1.1 Throttle dynamics

The throttle is a metal plate, (see figure 2.1), which controls the rate of air mass flow into the engine, or more directly into the intake manifold. Positioning of this metal plate is performed by a positioning servo called a throttle servo. A direct current (DC) electrical motor is controlled by a simple controller to position the throttle plate in a desired position in order to get the needed amount of output torque in the other end of the engine. When a reduction of air flow is desired the air path is being reduced by closing the throttle not only by the DC electrical motor but also by a spring attached to the metal plate. This is done to increase the speed of the system.

The air mass flow past the throttle, $\dot{m}_{at}$, can be approximated as the output of a first order low pass filter, LPF.

$$\frac{d\dot{m}_{at}}{dt} = \frac{1}{\tau_{th}}(\dot{m}_{at,ref} - \dot{m}_{at}(t)) \tag{2.1}$$

2.1 and ?? describes the air mass flow past the throttle plate.

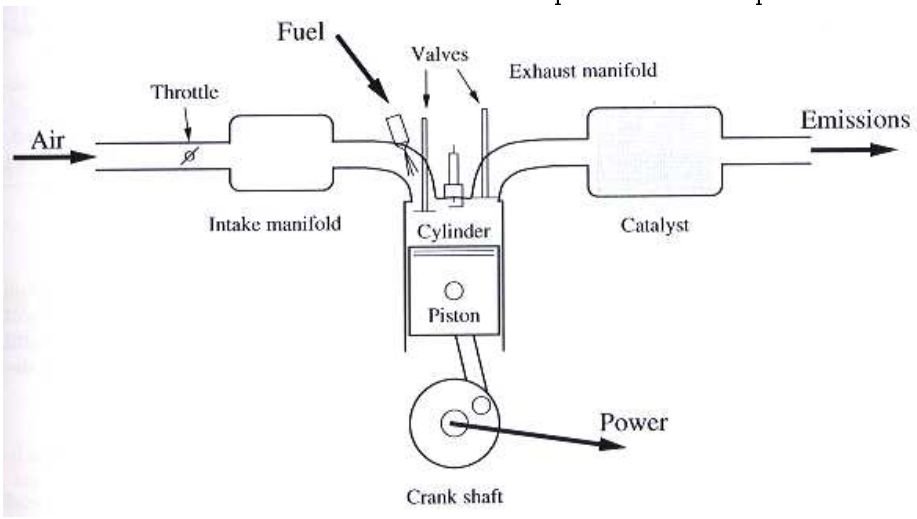

Figure 2.1: Air mass flow through the engine. [Eri05]

2.1 describes the air mass flow past the throttle plate, where $\dot{m}_{at,ref}$ is the reference air mass flow past the throttle at a certain operation point and τ_{th} is the time constant of the throttle response.

2.1.2 Intake manifold

The intake manifold is the volume of the pipe connected to the cylinder, figure 2.1. The primary function of the intake manifold is to evenly distribute the combustion mixture (or just air in a direct injection case) to each intake port in the cylinders heads. Even distribution is important to optimize the efficiency and performance of the engine.

The intake manifold pressure dynamics can in the same way as the air flow past the throttle be approximated by a first order LPF.

The pressure dynamics of the intake manifold is described by

$$\frac{dp_{im}}{dt} = \frac{RT_{im}}{V_{im}}(\dot{m}_{at} - \dot{m}_{ac}) \tag{2.2}$$

where the air mass flow into the cylinder, $\dot{m}_{ac}$, is given by

$$\dot{m}_{ac}(N_e, p_{im}, T_{im}) = \eta_{vol}(N_e, p_{im}, ...)\frac{V_d N_e p_{im}}{n_r R T_{im}} \tag{2.3}$$

and the volumetric efficiency is given by

$$\eta_{vol} = c_0 + c_1\sqrt{p_{im}} + c_2\sqrt{N_e} \tag{2.4}$$

where n_r is the number of engine revolutions per cycle, N_e is the engine speed, R is the specific gas constant, V_{im} is the intake manifold volume, V_d is the displaced volume, T_{im} is the intake manifold temperature, p_{im} is the intake manifold pressure and finally the constants c_0, c_1 and c_2 are used for black box identification of the volumetric efficiency denoted η_{vol}.

2.2 Ignition positioning

This section describes the effects of ignition positioning, starting with a brief description of the operating principles of a four stroke spark ignited engine. The expressions *Spark advance* and *Maximum brake torque* are explained and finally a mean value model for the engine torque is presented.

2.2.1 The four stroke cycle

In a four stroke spark ignited engine, an important property is the pressure in the cylinder, p_{cyl}, which settles the output torque of the engine, due to the fact that it creates the force on the piston.

The four strokes are called intake, compression, expansion and exhaust. During the intake stroke, the cylinder is filled with a mixture of air and fuel. The piston is moving downwards and the intake valve is open, which makes the pressure in the cylinder become close to the intake manifold pressure, p_{im}. When the piston moves further, the mixture of air and fuel is compressed to a higher temperature and pressure. This is the compression stroke and in this phase also a spark ignites the mixture and starts the combustion (around 25 degrees before top dead centre (TDC)). The expansion stroke starts when the combustion is about to finish (around 40 degrees after top dead centre). Here, work is done by the fluid in the combustion chamber when the volume expands. The exhaust stroke occurs around 130 degrees after top dead centre when the exhaust valve is opened. The fluid in the combustion chamber is pushed out into the exhaust system and the pressure in the cylinder gets close to the exhaust manifold pressure, p_{em}. Eventually, the piston reaches the top dead centre and a new stroke cycle takes place.

2.2.2 Spark advance

By controlling the timing of the spark it is possible to affect the output torque. This is called spark advance control. Spark advance is measured in crank angles before top dead centre and is used to position the combustion relatively to the crank shaft rotation. Besides the output torque, which will be the focus in this thesis, the spark advance also affects emissions, efficiency and engine knock. Mostly, the spark advance is controlled using the intake manifold pressure and the engine speed.

2.2.3 Maximum brake torque

An early ignition results in an early pressure build-up and a lower pressure during the later part of the expansion stroke. Consequently, a late ignition results in a later pressure build-up and a higher pressure during the expansion stroke. An optimum ignition timing results in maximum brake torque (MBT). Timings that are advanced or retarded from MBT results in a lower output torque. The MBT gives a certain peak pressure position (PPP), which commonly lies around 15-16 degrees after top dead centre. The PPP varies from cycle-to-cycle, which leads to cycle-to-cycle variations in the output torque, but when the mean PPP is at optimum the variations in the output torque are minimal. Figure 2.2 shows the output torque as a function of PPP.

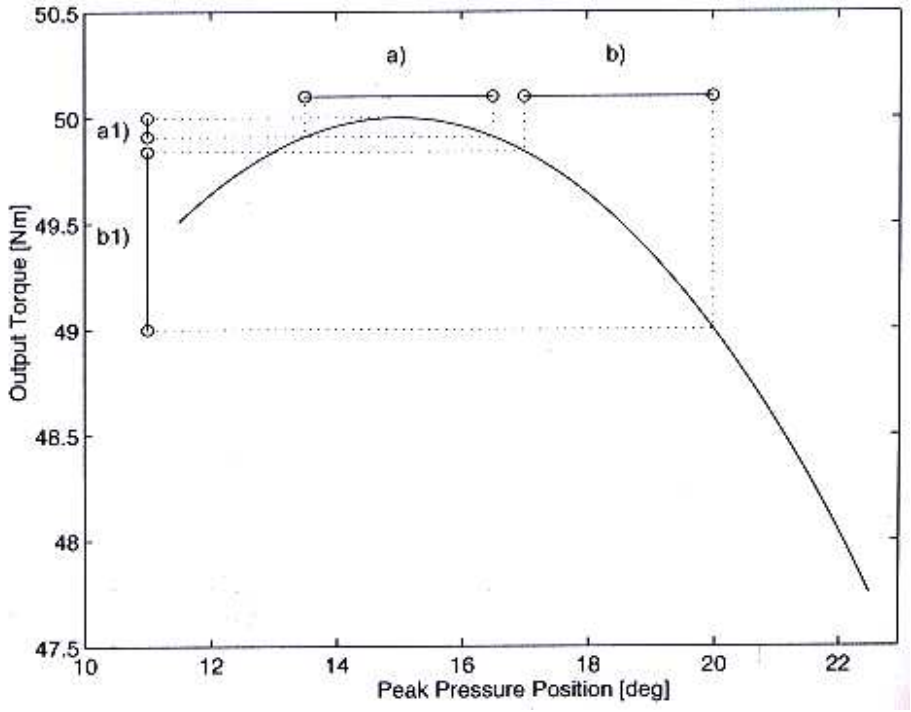

Figure 2.2: Output torque as a function of the peak pressure position (PPP).

At a, in figure 2.2, the mean PPP lies at optimum which gives small variations in the output torque at $a1$. At b, the mean PPP deviates from its optimum which gives larger variations in the output torque at $b1$.

2.2.4 Engine torque

The output engine torque, $TqOut$ is given by

$$TqOut = \frac{W_{ig} - W_p - W_f}{n_r \cdot 2\pi} \tag{2.5}$$

where W_p is the pumping work from the difference in intake and exhaust manifold pressure, W_f is the friction work and W_{ig} is the gross indicated work per cycle. These are given by the following expressions:

$$W_p = V_d(p_{em} - p_{im}) \tag{2.6}$$

$$W_f = V_d(C_{f0} + C_{f1}N_e + C_{f2}N_e^2) \tag{2.7}$$

$$W_{ig} = m_f \cdot q_{HV}\left(1 - \frac{1}{r_c^{\gamma-1}}\right) \cdot min(1, \lambda_c) \cdot \eta_{ign}(\theta_{ign}) \cdot \eta_{ign,ch}(\omega_e, V_d) \tag{2.8}$$

where C_{f0}, C_{f1} and C_{f0} are constants derived from the friction work, m_f is the fuel mass in the cylinder (see 2.9), $\dot{m}_{fc}$ is the fuel mass flow into the cylinder, q_{HV} is the specific heat value for a certain fuel, r_c is the compression ratio and γ is the ratio of specific heats. θ_{ign} is the ignition timing and η_{ign} is the efficiency corresponding to θ_{ign} (see equation 2.10). $\eta_{ig,ch}$ is the gross indicated efficiency that take combustion chamber losses (like heat transfer) into account.

$$m_f = \frac{n_r}{N_e} \cdot \dot{m}_{fc} \tag{2.9}$$

$$\eta_{ign}(\theta_{ign}) = 1 - C_{ign} \cdot (\theta_{ign} - \theta_{ign,opt}(\omega_e, m_f, \lambda))^2 \tag{2.10}$$

The latter is a simplified model, describing the efficiency decrease due to deviations from the optimum ignition timing, $\theta_{ign,opt}$. For each engine speed (ω_e) and load (specified by m_f and λ) there is an optimal position and ignition timing.

2.3 Moment of inertia

The rotational dynamics of the engine crank shaft gives a moment of inertia, J. According to Newton's second law, it can be written as [Nor06]

$$J\dot{\omega}_e = TqOut \tag{2.11}$$

Here, $\dot{\omega}_e$ is the derivative of the engine speed, given in radians per square-second.

2.4 System overview

Figure 2.3 shows a simplified, linearized overview of the system. Here, $TqBaseTgt$ is the desired torque and $TqBase$ the actual torque delivered by the air dynamics in the intake system. $TqBase$ is the maximum available torque in the system. $TqInstTgt$ is the desired torque related to the ignition timing. $TqInst$ is considered to be instantaneous, that is, it is considered to have no dynamics at all. However, there is a time delay that differs between the desired torque, $TqInstTgt$ and the actual torque, $TqInst$.

$TqInst$ is directly corresponding to the output torque, $TqOut$, but $TqOut$ is limited by the air mass flow in the intake system. This is illustrated by the rightmost saturation block in figure 2.3 and it means that it is not possible to get a higher output torque in the system than the maximum torque available, $TqBase$. Thus, a requirement is that $TqBase$ is greater than $TqInst$, which means that we will always get a difference between the two. This difference is called the torque reserve, $TqRsv$ (see section 2.4.1).

The air dynamics in the intake system is more complex to model than the ignition part. Obviously, it has physical limitations, since it is impossible for the intake system to handle an infinite amount of air. The system is also limited by the time it takes to open and close the throttle. These limitations are represented by the leftmost saturation block and the rate limiter, in figure 2.3, respectively. There are also two separate first order low pass filters in the figure, which approximates the air flow through the throttle and pressure dynamics of the intake manifold, as mentioned in section 2.1.1 and 2.1.2. T_{loss} represents the losses in the output torque, due to pumping work and friction work.

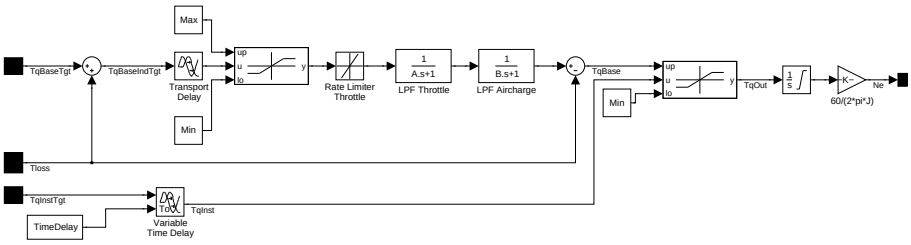

Figure 2.3: A simplified overview of the system

2.4.1 Torque reserve

As mentioned in the previous section 2.4, the torque reserve, $TqRsv$, is the difference between $TqBase$ and $TqInst$.

$$TqRsv = TqBase - TqInst \tag{2.12}$$

The advantage having a torque reserve is that the engine easier can handle unexpected disturbances without decreasing the engine speed to an unacceptable level. However, the torque reserve should not be unnecessary large, since it increases the fuel consumption. Thus, a control problem is a trade-off between having enough torque reserve to handle disturbances and at the same time minimize the fuel consumption.

According to (2.12), the $TqRsv$ is affected by increasing or decreasing the air mass flow in the intake system and by changing the torque related to the ignition timing.

By letting the spark advance deviate from its optimum, a contribution to the torque reserve is created. This margin is later used for quick compensations for disturbances. When this is not enough for compensating the disturbances,

the air mass flow in the intake system can be increased and used for further compensations. The latter method takes more time to carry out why it is necessary to let the ignition timing deviate from its optimum during steady state and thereby making quick compensations possible.

2.4.2 Simplified modelling of the system

Due to the complexity of the system and to its nonlinear behaviour, a system identification, based on a black-box model was decided to be made. Therefore, the air dynamics was considered a black-box, later represented by an Auto Regressive model with exogenous inputs, ARX-model, while the ignition part was modelled as a time delay. Figure 2.4 shows a simplified model of the system, where the air dynamics is represented by a black-box and the ignition positioning is represented by a time delay.

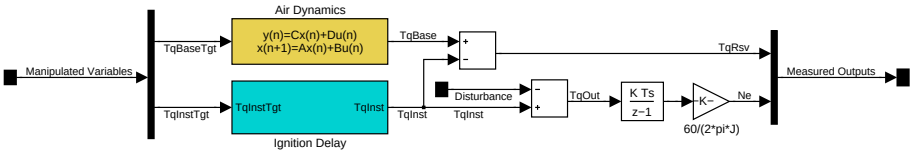

Figure 2.4: A simplified model of the system.

The controllable variables are $TqBaseTgt$ and $TqInstTgt$ and the outputs are the engine speed, N_e, and the torque reserve, $TqRsv$. As can be seen in the figure, the torque reserve is calculated according to 2.12. The necessity that $TqBase$ must be greater than $TqInst$ was later handled by the model predictive controller, that was given the condition that the torque reserve must not be negative. The output torque was directly converted into engine speed by using 2.13 where the moment of inertia was determined by experiments (see section 3.3 and Appendix A).

$$N_e = \frac{60}{2\pi} \int_0^t \frac{TqOut}{J} dt \tag{2.13}$$

2.5 MODEC

To be able to perform simulations of both the existing controller and the MPC that is to be constructed there is a need of a model containing the most vital parts of the engine. MODEC is a model that is used for simulations in the Matlab/Simulink environment. An overview of the systems involved in the MODEC model can be seen in figure 2.5. This model has been developed over the years at Volvo Car Corporation (VCC) and in the beginning its focus was primarily on the mean value engine models (MVEM) concerning the air path and the torque generation in the engine. As the work proceeded it now also contains turbo charging, variable valve timing (VVT) dynamics and fuel dynamics. There are also a number of systems implemented for automatic control of different parts

in the engine.

The main purpose for MODEC is to provide a fundament of dynamic real-time capable models to be used in VCC's hardware in the loop (HIL) simulators. There are several versions of the MODEC model available but the version used in this thesis is a model of the short inline 6 (SI6) engine. Several releases have also been used through the period of this thesis work. The release dates of these different versions were 2007-05-07, 2007-08-23 and finally 2007-08-31 where the last one was used to generate the control results seen in section 4.5. [Rub07]

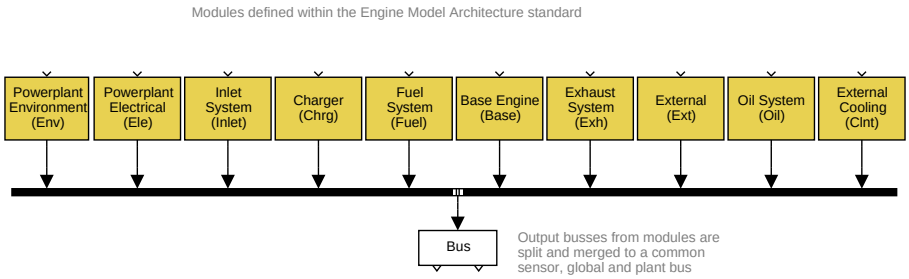

Figure 2.5: An overview of the systems involved in MODEC

Chapter 3

System identification

To be able to identify and later on control the air dynamics, represented as a black-box, a system identification was decided to be made. This chapter describes this system identification process. The basic theory of Black-box models and general advices for preparing the experiments and later on handle the data sets are given in section 3.2. The experiments are reviewed in section 3.3 and the identification and treatment of the data in section 3.4. The results from the system identification are finally presented in section 3.5.

3.1 Introduction

"System identification is the subject of constructing or selecting models of dynamical systems to serve certain purposes." [Lju99]

The first thing to do in a system identification procedure is to collect data. Here, a number of choices have to be made, such as selecting input signal and sampling time. This is included in the experiment design-part (see section 3.2.2. After the data has been collected, it usually needs to be filtered due to disturbances and deficiencies. This procedure is called preprocessing the data. Finally, a suitable model set is to be selected and from this set the most appropriate model should be chosen. If the model is not considered good enough, a new model has to be found. [Lju99]

System identification methods can be divided into the two main groups, non-parametric methods and parametric methods. This thesis will only focus at parametric methods.

Parametric methods

As the name suggests, the purpose of parametric methods is to estimate parameters using statistical methods. If the physics behind the model is well-known, it is appropriate to use tailor-made models. Here, the parameters represent unknown values that are physically interpretable. Black-box models are used when the physical background of a system is unknown or too complicated to model. In these cases the parameters have no physical meaning. They are just parameters used for describing the input-output relationship of the system. [Lju04]

3.2 Theory

In this section, the Black-box models ARX (Auto Regressive model with eXogenous inputs) and ARMAX (Auto Regressive Moving Average model with eXogenous inputs) are explained and derived. Basic principles for experiment design and preprocessing data are also given.

3.2.1 Black-box models

When the physical background of a system is unknown or too complicated to model, a convenient way is to use a standard linear model, a black-box model. Such a model set contains adjustable parameters to be determined by estimation procedures. [Lju04]

If the parameters to be determined are denoted by the vector θ, then a complete model set can be described as

$$y(t) = G(q,\theta)u(t) + H(q,\theta)e(t) \tag{3.1}$$

where $y(t)$ is the output signal, $u(t)$ is the input signal and $e(t)$ is white Gaussian noise. $G(q,\theta)$ and $H(q,\theta)$ are the transfer functions of the system and the disturbances, respectively.

By representing G and H as rational functions of the discrete shift operator q and letting the parameters be the numerator and denominator coefficients, G and H can be parameterized as follows [Lju99]

$$G(q,\theta) = \frac{B(q)}{F(q)} = \frac{b_1 q^{-nk} + b_2 q^{-nk+1} + \cdots + b_{nb} q^{-nk-nb+1}}{1 + f_1 q^{-1} + \cdots + f_{nf} q^{-nf}} \tag{3.2}$$

$$H(q,\theta) = \frac{C(q)}{D(q)} = \frac{1 + c_1 q^{-1} + \cdots + c_{nc} q^{-nc}}{1 + d_1 q^{-1} + \cdots + d_{nd} q^{-nd}} \tag{3.3}$$

ARX (Auto Regressive model with eXogenous inputs)

A simple input-output relationship can be described as a linear difference equation. If it also includes a white-noise term $e(t)$, as a direct error, it is called an equation error model.

$$y(t) + a_1 y(t-1) + \cdots + a_{na} y(t-na) = b_1 u(t-1) + \cdots + b_{nb} u(t-nb) + e(t) \tag{3.4}$$

where the coefficients $a_1 \ldots a_{na}$ and $b_1 \ldots b_{nb}$ are the parameters to be determined in the vector θ, which becomes

$$\theta = \begin{bmatrix} a_1 & \ldots & a_{na} & b_1 & \ldots & b_{nb} \end{bmatrix}^T \tag{3.5}$$

where

$$A(q) = 1 + a_1 q^{-1} + \cdots + a_{na} q^{-na} \tag{3.6}$$

and

$$B(q) = 1 + b_1 q^{-1} + \cdots + b_{nb} q^{-nb} \tag{3.7}$$

If the denominators in (3.2) and (3.3) are equal to $A(q)$, we have

$$A(q) = F(q) = D(q) \tag{3.8}$$

Combining (3.8) with (3.1) gives the ARX model

$$y(t) = \frac{B(q)}{A(q)} u(t) + \frac{1}{A(q)} e(t) \tag{3.9}$$

The advantage of the ARX model is its simplicity which makes it easy to calculate. The backside is that the $A(q)$ polynomial is also used to describe the properties of the disturbance term, which causes some errors in the estimation of the system dynamics. [Lju04]

ARMAX (Auto Regressive Moving Average model with eXogenous inputs)

For the ARMAX model, the equation error model is described as

$$y(t)+a_1 y(t-1)+\cdots+a_{na} y(t-na) = b_1 u(t-1)+\cdots+b_{nb} u(t-nb)+e(t)+c_1 e(t-1)+\cdots+c_{nc} e(t-nc) \tag{3.10}$$

where

$$C(q) = 1 + c_1 q^{-1} + \cdots + a_{nc} q^{-n_c} \tag{3.11}$$

Now the adjustable parameters are

$$\theta = \begin{bmatrix} a_1 \dots a_{na} & b_1 \dots b_{nb} & c_1 \dots c_{nc} \end{bmatrix}^T \tag{3.12}$$

The ARMAX model is then given by

$$y(t) = \frac{B(q)}{A(q)} u(t) + \frac{C(q)}{A(q)} e(t) \tag{3.13}$$

Both ARX and ARMAX are useful models when the disturbance enters the system at an early stage. This is because the disturbance term $e(t)$ and the input signal $u(t)$ share the same poles. [Lju99]

3.2.2 Experiment design

Before the experiments can be performed, it is important to make some important choices, such as selecting input signal and sampling interval.

Selecting input signal

The purpose of the input signal is to excite all frequencies of interest in the system. Thus, it has to contain sufficiently many distinct frequencies. A good idea for linear systems is to use a binary signal that has a certain probability to change from one level to another. When the changes between the two levels are random, the signal will contain all frequencies. The two levels should be set

to the maximum variation that is allowed. The input signal should change between its levels fast enough to catch the shortest time constant, but also remain constant for a period of time, letting the system settle. [Lju99], [Lju04]

Pseudo-Random Binary Signal (PRBS)

A common input signal for system identification is the Pseudo-Random Binary Signal. A PRBS is a periodic, deterministic signal with white-noise-properties and it can be generated by letting

$$u(t) = rem[A(q)u(t), 2] = rem[a_1 u(t-1) + \cdots + a_n u(t-n), 2] \qquad (3.14)$$

where $rem[x, 2]$ is the remainder as x is divided by 2. Here, $u(t)$ can only become 0 or 1, but after the signal has been generated, it could assume any value. The PRBS is a periodic signal, with the maximum periodic length, $M = 2^n - 1$. Choosing different A(q) polynomials in (3.14) gives different actual periods. Table 3.1 shows choices of A(q) polynomials that gives the maximum length PRBS for different orders n. [Lju99]

Order n	$M = 2^n - 1$
2	3
3	7
4	15
5	31
6	63
7	127
8	255
9	511
10	1023
11	2047

Table 3.1: A(q) polynomials that generate maximum length PRBS for different orders n.

A maximum length PRBS has white-noise second order properties, which is described by its signal spectrum, but to make sure that the PRBS preserve its good properties, it must be generated over one full period, $M = 2^n - 1$, and then be repeated. [Lju99]

The PRBS has a clock period (N), which is the minimum number of samples that the signal is held constant. According to [dK02] it is suitable to choose the clock period to be a 10th of the slowest time constant of the system.

Choice of sampling interval

Sampling always causes information losses. To make these losses insignificant, it is important to select an appropriate sampling interval, T_s. [Lju99]

The choice of sampling interval depends on the time constants of the system. If the sampling is much faster than the system dynamics, the new data points will not add any new relevant information. If the sampling is too slow, informative data will be missing. A good choice of sampling interval should be a trade-off between noise reduction and relevance for the dynamics. In general, it is better to choose a short sampling interval than one that is too long. [Lju99], [Lju04]

Aliasing

Sampling the data affects the signal spectrum. The sampling frequency is denoted as

$$\omega_s = \frac{2\pi}{T_s} \tag{3.15}$$

and the Nyquist frequency as

$$\omega_N = \frac{\omega_s}{2} = \frac{\pi}{T_s} \tag{3.16}$$

Frequencies higher than the Nyquist frequency will be interpreted as contributions from lower frequencies. Hence, information about these higher frequencies is lost by sampling. This is known as the alias phenomenon. [Lju99]

To avoid these misinterpretations, an antialiasing filter is being applied before the sampling. The antialiasing filter is a low-pass filter with a cut-off frequency (ω_B) just below the Nyquist frequency (ω_N). [Rob03]

3.2.3 Preprocessing data

When the identification experiment is all set, it could be a good idea to plot the collected data. There are probably several deficiencies in the data and these are easier to detect in a plot. Due to these deficiencies, it is important to preprocess the data before going into the procedure of identification.

Some common deficiencies in the data are high-frequency disturbances, outliers, missing data, drifts, offsets and trends. [Lju99], [Lju04]

High-frequency disturbances

High-frequency disturbances can be removed by low-pass filtering the data before the work of identification starts. If the sampling interval turned out to be too short, the data can also be decimated. [Lju99], [Lju04]

Outliers and missing data

Sometimes single data points are missing in the measurement output or input. This is often due to malfunctions in the sensors. A similar problem is the so-called outliers, which are obvious bad errors that are caused by failures in the

measurement. These bad values are often easier to detect in a residual plot. [Lju99]

One way to avoid getting missing data and outliers into the identification procedure is to cut out segments of the data set. Later, it is possible to use these different parts separately for estimation and validation, respectively. [Lju99], [Lju04]

Drifts and offsets

Low frequency disturbances like drift, offset and trend often occurs during the measurement. These slow variations can be taken care of by high-pass filtering the data before the identification process takes place. [Lju99]

3.2.4 Model validation

To investigate and decide if a model is good enough for the intended purpose is called model validation. [Lju99]

Choice of model structure

First, a model structure has to be chosen. Decisions regarding type of model, model order and model parameterization are to be made. [Lju99]

Controllability

Controllability is related to the possibility of forcing the system into a particular state by using an appropriate control signal. [Lju03]

For a discrete-time linear state-space system, the state equation is

$$\mathbf{x}(k+1) = A\mathbf{x}(k) + B\mathbf{u}(k) \tag{3.17}$$
$$\mathbf{y}(k) = C\mathbf{x}(k) + D\mathbf{u}(k) \tag{3.18}$$

where A is a $n \times n$ matrix. Then, the controllability matrix ζ is defined as

$$\zeta = \begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix} \tag{3.19}$$

If the controllability matrix ζ has full rank, the system is said to be controllable. [Lju03]

Observability

Again, for a discrete-time linear state-space system, the state equation can be written as (3.17) and (3.18). A system is then said to be observable if, for

any possible sequence of state and control vectors, the current state can be determined in finite time using only the inputs and outputs.

Let the observability matrix Ω be defined as

$$\Omega = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} \tag{3.20}$$

The system is observable if the rank of Ω is equal to n. [Lju03]

3.3 Experiments

The system identification experiments were performed in three different environments. The first time, the measurements took place in a vehicle (Experiment 1). Second, the measurements were performed in an FPD-rig (Experiment 2). In experiment 3, no measurements were performed. Instead, the collecting of data was done by simulating the MODEC engine model.

For collecting the data from the measurements we used the software INCA (Integrated Calibration and Application tools). The step functions were implemented manually during the measurements and the PRBS signal was generated by using the command *idinput* in Matlab.

During this thesis work, the MODEC engine model was updated and improved as model errors were discovered. Consequently, simulation results from the engine model may vary from different experiments. Primarily, overshoot characteristics and offset errors were reduced. This also affected the time constants of the model, which made us change the clock period of the PRBS input signal during the project.

3.3.1 Experiment design

Before going into the real measurements, an early version of the MODEC engine model was simulated to approximate the time constants, τ_{63}, and the delay, τ_d, of the system. The slowest time constant was found out to be $\tau_{63} = 138$ ms and the delay $\tau_d = 80$ ms. The approximated time constant was then used to generate an appropriate PRBS signal, that had a clock period which excited the important frequencies of the system.

The objectives of the experiments were to collect data that made it possible to:

- Identify the air dynamics of the system.
- Validate the air dynamics of the MODEC engine model.
- Determine the two time constants of the system, τ_{63}.

- Determine the built-in delay for the ignition part of the system, τ_d.
- Calculate the moment of inertia from the crank shaft dynamics, J.

Choice of sampling interval

When simulating the MODEC engine model, the sampling interval is $T_s = 1$ ms. Such a high resolution is not necessary and more important, it is not viable in the measurements, why we chose $T_s = 8$ ms as the sampling interval.

Choice of input signals

For the measurements, we used two different types of input signals. In both cases, the input signal represents the $TqRsv$ target. The first input signal was a number of step functions. The steps differed in amplitude and the purpose was to determine the time constants, τ_{63}, of the system and the moment of inertia, J.

Second, a PRBS signal was chosen as an input signal. The purpose here was to identify the air dynamics of the system.

After generating the PRBS in Matlab, the signal was cut, so that the period length became $M = 2^n - 1$. The maximum length PRBS was then divided into two equally long parts; one part for estimation and one for validation. Finally, we duplicated the two parts into three identical estimation parts and three identical validation parts. The reason for the duplication was to get a periodic behaviour, making it easier for the identification process to sort out disturbances. All together, the six parts built the PRBS input signal.

3.3.2 Data collection

This section presents the experiments made for collecting data. The data were later used for identification.

Measurements in vehicle - Experiment 1

The first experiment was performed in a Volvo XC90, SI 6.

From the simulation of the engine model, we got the time constant $\tau_{63} = 138$ ms and according to [dK02], the clock period for the PRBS was then calculated to $N = 14$ samples.

Figure 3.2 shows a comparison between the measured and simulated $TqBase$ during the measurement that was given step functions as the input signal. It should be noticed that the input target signal corresponds to the torque reserve $TqRsv$ and the output is the torque given by the air from the intake system, $TqBase$. For analysis convenience, the amplitude of $TqRsv$ has in the plot been modified to the same magnitude as the output $TqBase$. The heavy overshoot behaviour that the simulated model (red curve) shows was later reduced by changes in the MODEC engine model, as it turned out that the engine model

Figure 3.1: Picture of a Volvo XC 90, the type of car used in experiment 1.

needed improvements.

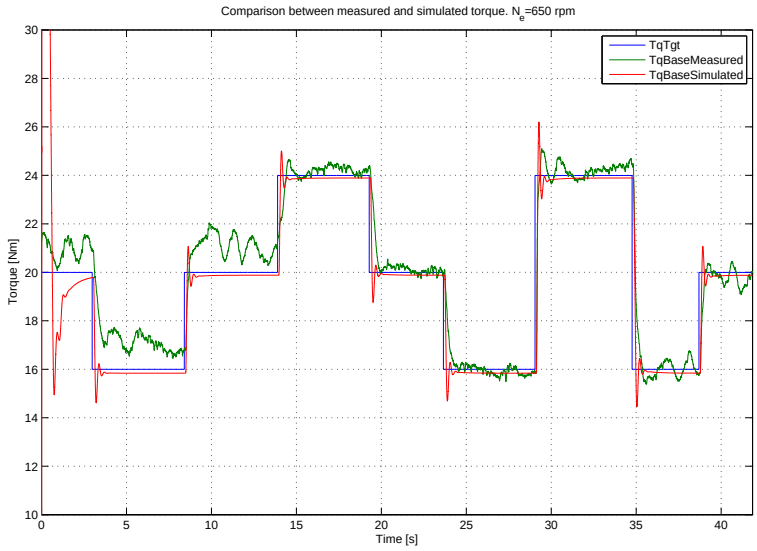

Figure 3.2: Experiment 1 - Comparison between measured and simulated torque with step functions as the input signal.

As can be seen in figure 3.2, the measured torque is fluctuating a lot. This is because the engine speed did not succeed to remain constant at its target level of $N_e = 650$ rpm as desired. Figure 3.3 shows the unfiltered engine speed for the two different input signals. Clearly, the engine speed varies in a way that can not only be explained by measurement disturbances. For the step input, the standard deviation of the engine speed was calculated to be $\sigma = 3.48$ rpm and for the PRBS input $\sigma = 24.08$ rpm. The means for the step input and the PRBS input were calculated to $\mu = 649.9$ rpm and $\mu = 655.6$ rpm, respectively. Since the target for the engine speed is $N_e = 650$ rpm, this indicates that the deviation for the PRBS from its target is even larger than the results for the

standard deviation shows.

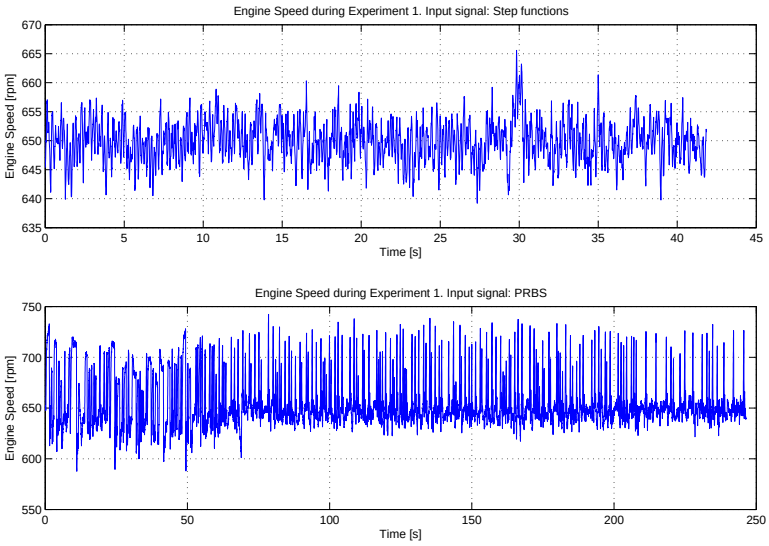

Figure 3.3: Experiment 1 - The unfiltered engine speed for the step input and the PRBS input, respectively.

Due to the fluctuation of the engine speed, a system identification based on this data set would not be suitable for our purpose. The engine speed need to be steady and hence we decided to carry out the following measurements in an FPD-rig. The moment of inertia was also decided to be calculated from future measurements, since the various engine speed made the results unreliable.

Measurements in FPD-rig - Experiment 2

The second experiment was performed in an FPD-rig. The FPD-rig is an environment for dynamical testing of functions. The rig consists of a real engine with a simulated gearbox. The dynamical environment makes it possible to perform driving cycles tests that simulate how the engine will behave in reality during certain conditions. An older version is the FP-rig, which only allows statically testing. The FP-rig does not include a simulated gearbox.

We used the same PRBS signal as in the vehicle measurements. One problem though, was that INCA was not able to update the changes in the input signal as fast as desired. This problem was solved by a decimation of the input signal.

In the FPD-rig, the engine brake keeps the engine speed at a constant level. As figure 3.5 shows, the engine speed is now kept closer to its target value, $N_e = 650$ rpm. The standard deviations are now $\sigma = 1.07$ rpm and $\sigma = 0.89$ rpm for the step input and the PRBS input, respectively. The means are $\mu = 649.9$

Figure 3.4: Picture of the FPD-rig environment, used for measurements in experiment 2.

rpm for both input signals. These data was considered stable enough to go on with the system identification procedure.

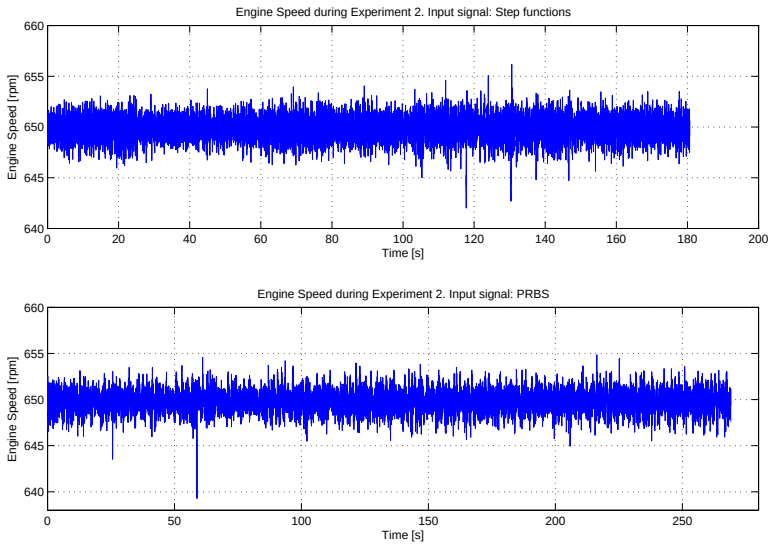

Figure 3.5: Experiment 2 - The unfiltered engine speed for the step input and the PRBS input, respectively.

Figure 3.6 shows that the measured and the simulated torque now are much more similar than in the vehicle measurements. The small variations in the measured torque come from small engine speed fluctuations and from measurement disturbances. From these results the moment of inertia was calculated to be $J = 0.22\ kg \cdot m^2$. A detailed description of how the moment of inertia was determined is found in Appendix A. The delay was determined to be $\tau_d = 80$ ms.

Again, we were at this time in the project using an older version of the engine model, which explains the distinguished overshoots for the simulated torque. As

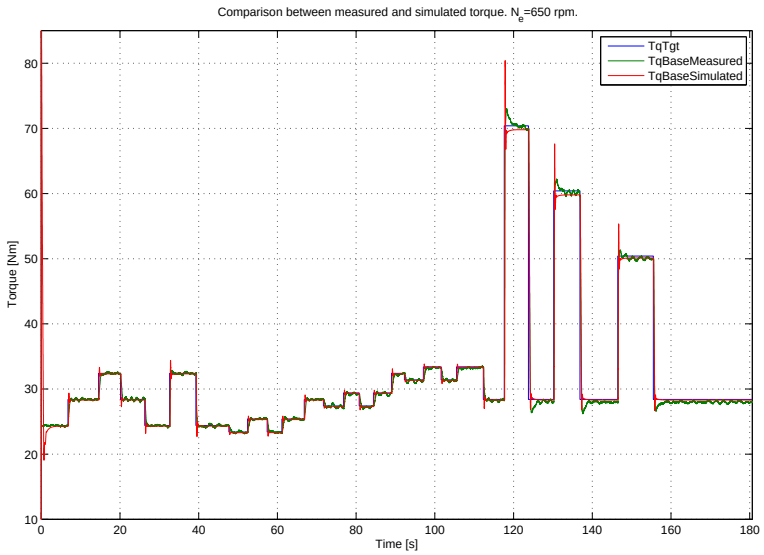

Figure 3.6: Experiment 2 - Comparison between measured and simulated torque with several step functions as the input signal.

we discovered that the overshoots are not that substantial for the real measurements, the engine model was later improved to behave more like reality.

Simulation in MODEC engine model - Experiment 3

A final experiment was to run a simulation in the MODEC engine model and later use these results for system identification. At this stage, the improved engine model was used. Figure 3.7 shows the differences between an older version of the MODEC engine model and the latest version. Obviously, the overshoot characteristics from the older model have been reduced significantly.

The time constants for the latest engine model were determined to $\tau_{63} = 108$ ms and $\tau_{63} = 191$ ms for the uphill and downhill slope, respectively. We calculated the clock period to $N = 19$ samples.

3.4 Identification

This section describes the identification procedure including the part of preprocessing data. It also contains the procedure of model validation.

3.4.1 Preprocessing data

After the experiments were finished, the data was preprocessed before the identification procedure took place.

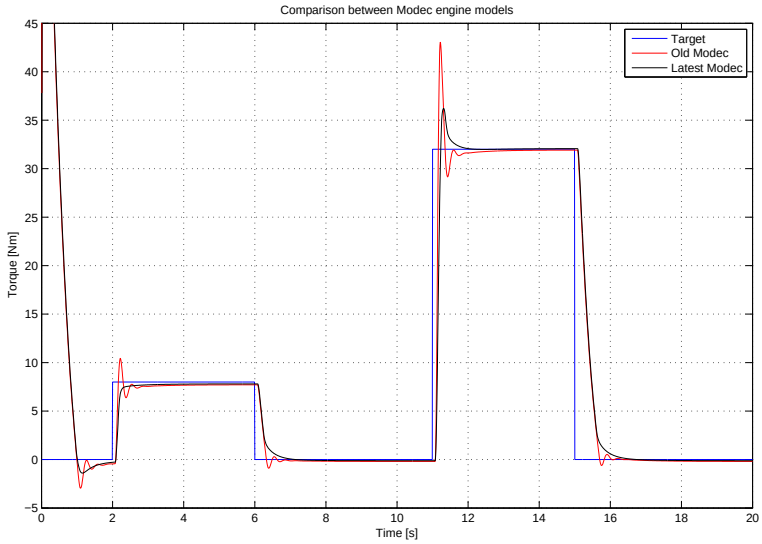

Figure 3.7: Comparison between an older version of the MODEC engine model and the latest version, used in experiment 3.

Detrend/Remove means

By using the Matlab-command *detrend*, the mean value from the original output data was removed. A possible linear trend was also removed by the same command. Figure 3.8 illustrates the original and detrended output data from the measurements in the FPD-rig (experiment 2) and from the simulation of the MODEC engine model (experiment 3). In both cases, a PRBS signal was used as the input signal.

Filtering

In experiment 2 there were high-frequency disturbances that were unwanted in the identification procedure. A Butterworth-filter of the 10^{th} order was used to low-pass filter the data from the output signal in experiment 2. In this experiment, the Nyquist frequency was calculated to

$$\omega_N = \frac{\pi}{T_s} = \frac{\pi}{0.008} = 392.7 rad/s = 62.5 Hz \tag{3.21}$$

The fastest time constant was in experiment 2 determined to be $\tau_{63} = 56.5$ ms, which corresponds to a frequency of $f = \frac{1}{\tau_{63}} = 17.7$ Hz. Frequencies above this have no significance and were decided to be filtered. However, it was not possible to determine the time constant in an accurate way and for that reason we chose to filter frequencies above $f = 20$ Hz. These are frequencies that correspond to a time constant that is less or equal to $\tau_{63} = 50$ ms.

The Butterworth filter frequency response can be seen in figure 3.9. To implement the filter, the Matlab-command *filtfilt* was used. The advantage of

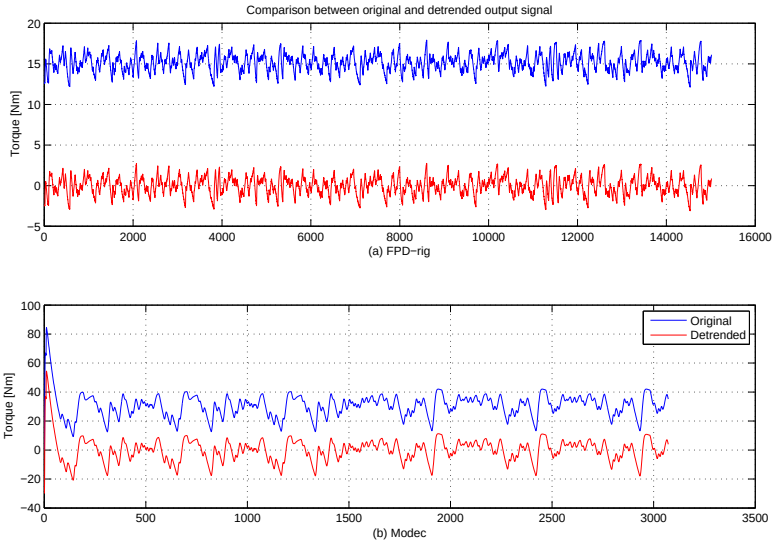

Figure 3.8: Original and detrended output data. (a) Data from measurements in the FPD-rig (experiment 2). (b) Data from simulation of the MODEC engine model (experiment 3).

filtfilt is the zero phase modification. After filtering in the forward direction, the filtered sequence is reversed and run back through the filter. [**?**]

An extract from the filtered and non-filtered output signal is illustrated in figure 3.10. The figure shows that the behaviour from the highest frequencies is reduced by the filter and the filtered signal demonstrates a smooth curve in comparison. Figure 3.11 shows the spectrum of the filtered and non-filtered output signal. Here, the peaks that occur at frequencies above $f = 20$ Hz have been heavily reduced.

Downsampling/decimation

As mentioned in section 3.3.1, the sampling time of the MODEC engine model is $T_s = 1$ ms. By picking every 8^{th} point from the input and output signal, we downsampled the signals and the new sampling interval was $T_s = 8$ ms.

However, questions arose whether the signals should be downsampled even more. Trials were made with the sampling times of $T_s = 16$ ms and $T_s = 32$ ms and the data sequences were stored to be used later in the identification process.

Outliers and missing data

In experiment 3 there could not be any missing data or outliers. In experiment 2, we wanted to avoid the starting period because it could contain data that are not representative. The start-up sequence was avoided by letting the engine

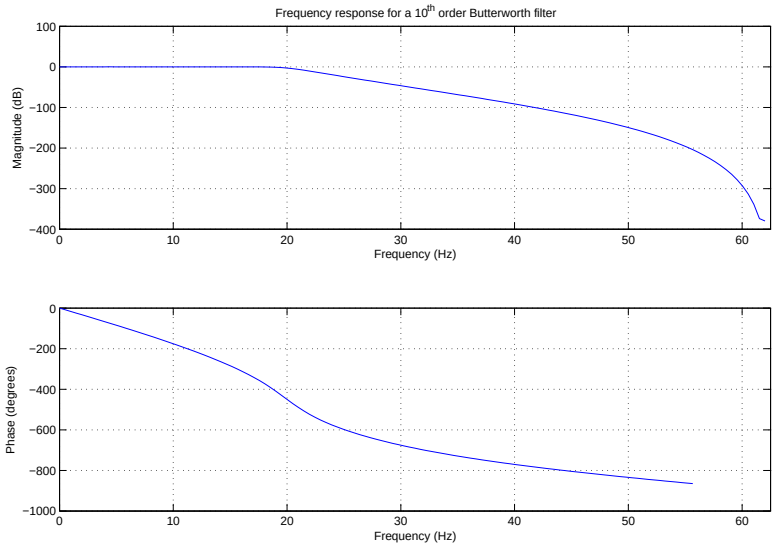

Figure 3.9: Characteristics of the applied Butterworth 10^{th} order filter, having a cut-off frequency, $\omega_B = 20$ Hz.

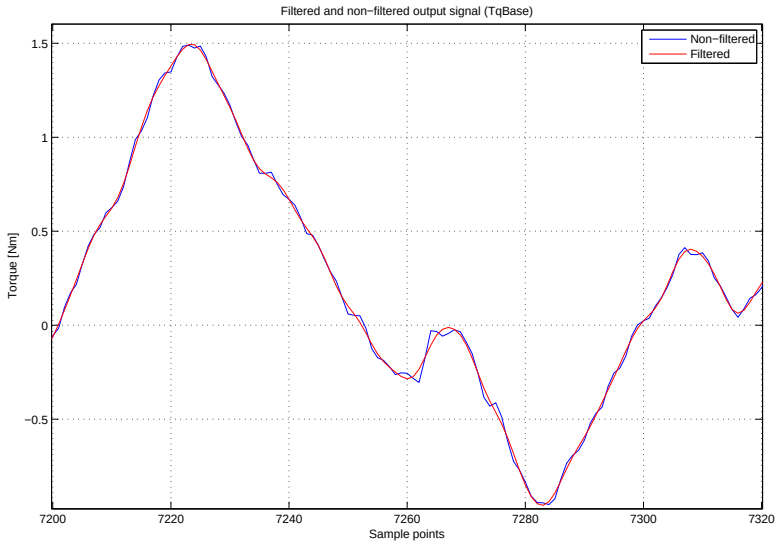

Figure 3.10: An extract of the output signal. The non-filtered output signal compared to the output signal, filtered by a Butterworth 10^{th} order filter

run for a while before the measurements started. In both experiments, half the data set was used for estimation and the other half for validation.

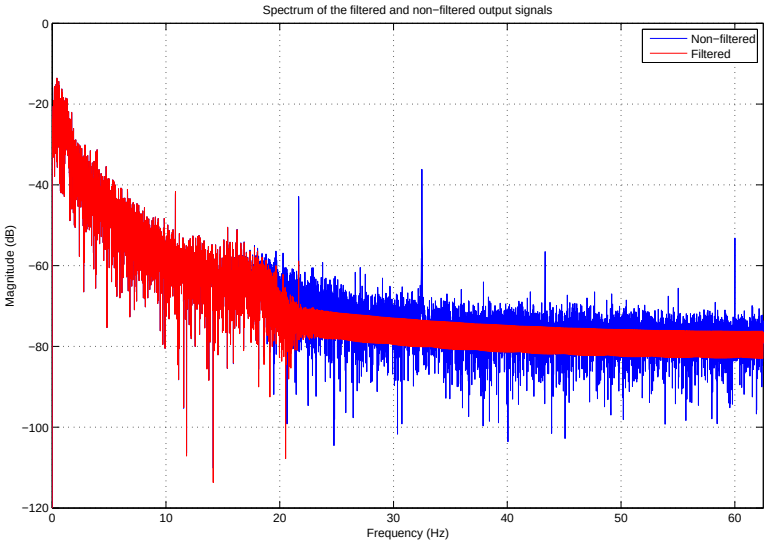

Figure 3.11: The spectrum of the non-filtered output signal compared to the spectrum of the output signal, filtered by a Butterworth 10^{th} order filter

3.4.2 Model validation

When the data was preprocessed and ready for identification, we created a script in Matlab that read the input and output signals and looped through a number of models. If the model passed certain stability requirements its properties were printed out for further analysis. The stability requirements were that all poles and zeroes must be within the unit circle and that the Nyquist curve must be to the right of -1 [Ben02]. Furthermore, the model had to be observable and controllable.

Choice of model structure

In the choice of model structure, we decided to go by the principle "Try simple things first". Since the physical background of the system is complicated, we chose to focus at Black-box models and we especially put our effort into the investigation of ARX-models. A problem with the Black-box models is that they are linear models. The system is in fact a nonlinear system, but since this thesis focus only at idle speed, we decided to consider the system as a linear system close to its operation point, $N_e = 650$ rpm.

This assumption is a simplification and in figure 3.12, it is possible to see that the system is not linear. One indication of this is that the system is acting faster on its way up than on the way down. The MODEC engine model is trying to catch this behaviour and clearly shows an overshoot on its way up, which does not exist on its way down. A Black-box model could never reproduce this behaviour, why we had to find a Black-box model that goes in the middle of the road.

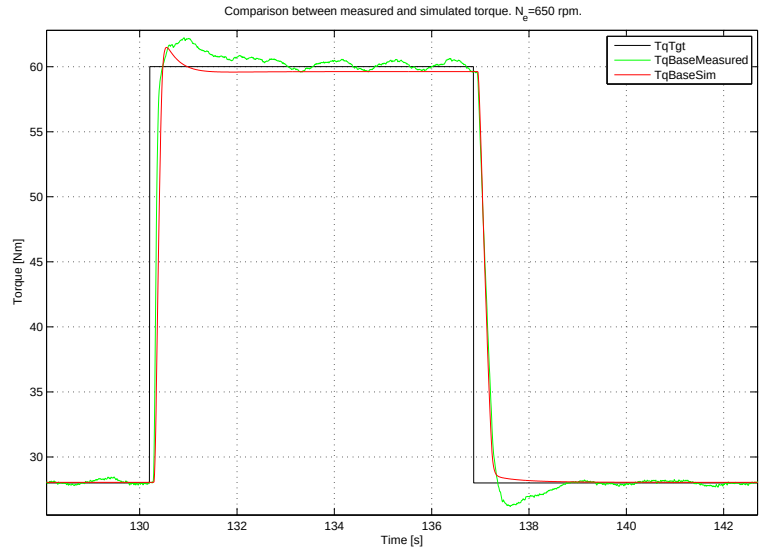

Figure 3.12: An extract of $TqBase$. The simulated torque from the MODEC engine model (experiment 3) compared to the measured torque from the FPD-rig (experiment 2).

Data from experiment 2 and experiment 3 were used to find an appropriate model. For the identification procedure a PRBS signal with a clock period of $\frac{1}{B} = 56s^{-1}$ was used in both cases. After preprocessing the data, the input and output signals were used for system identification in Matlab, using the commands *arx* and *armax*. These commands compute ARX and ARMAX models, given the input and output signals and the orders and delays of the models. The delay was earlier determined to be $\tau_d = 80$ ms which means that when the sampling time was $T_s = 8$ ms, the delay should be 10 sample points, since 80 divided by 8 equals 10. The different combinations of parameters tested are listed in table 3.2

Model	n_a	n_b	n_c	n_k
ARX, $T_s = 8$ ms	1-10	1-10	n.a.	10-12
ARX, $T_s = 16$ ms	1-10	1-10	n.a.	4-6
ARX, $T_s = 32$ ms	1-10	1-10	n.a.	2-3
ARMAX, $T_s = 8$ ms	1-10	1-10	1-5	10-12
ARMAX, $T_s = 16$ ms	1-10	1-10	1-5	4-6
ARMAX, $T_s = 32$ ms	1-10	1-10	1-5	2-3

Table 3.2: Different model orders tested for ARX and ARMAX models with three different sampling intervals. All combinations of the parameters in the table were tested.

Here n_a=number of poles, n_b=number of zeroes plus 1, n_c=number of C

coefficients, n_k=delay. The parameters n_a, n_b, and n_c (for ARMAX) is said to be the model order. The selected models turned out to be ARX-models with two poles, three zeroes and a delay of 11 sample points for both the FPD-rig measurements and the MODEC simulations. Note that even if the same model order and delay were chosen for both the experiments, it is in fact two different ARX-models, based on different output data. The models are denoted $ARX2411_{FPD}$ and $ARX2411_{MODEC}$, respectively and some of their properties are listed in table 3.3.

Model	Steady state	Fit	τ_{63}
$ARX2411_{FPD}$	18,9%	43,9%	232 ms
$ARX2411_{MODEC}$	78,1%	73,2%	248 ms

Table 3.3: Some properties for $ARX2411_{FPD}$ and $ARX2411_{MODEC}$. The steady-state gain and fit are compared to the measured output signal (in percent).

Model validation

All combinations of the parameters in table 3.3 were tested using several nestled for-loops in Matlab. For all model sets, its steady-state gain and its dominant time constant were calculated. Due to the fact that the Black-box models are linear, it has only one dominant time constant, τ_{63}, that is the same for the uphill part as for the downhill part. The hard part here was to find a model that was fast enough at the uphill part and at the same time did not overshoot too much.

The ARMAX-models were overshooting too much and hence these models were abandoned. For the ARX-models the time constant was too slow when the steady-state value was close to 1, but when the time constant was fast enough, the steady-state value became too low. This problem was solved by normalizing the ARX-model, making sure that the steady-state always was equal to one. By letting

$$k \cdot \frac{b_1 + b_2 + \dots b_n}{1 + a_1 + a_2 + \cdots + a_n} = 1 \Longleftrightarrow k = \frac{1 + a_1 + a_2 + \cdots + a_n}{b_1 + b_2 + \dots b_n} \tag{3.22}$$

the parameter k was calculated and then used to manipulate the b-coefficients, making the steady-state level always become equal to one. The downside of this manipulation is that it will lead to changes in the frequency domain. The magnitude curve in a Bode plot will increase if the steady-state was too small before the normalization. The Bode plot for $ARX2411_{FPD}$ and $ARX2411_{MODEC}$ is shown in the upper left in figure 3.13 and figure 3.14 respectively.

In the Bode plot it is clear that the normalized model (green curve) has a greater magnitude than the original model. When the model has been normalized, it is no longer adapted in the best way corresponding to the data sets given in the identification. On the other hand, the steady state value becomes exactly 1.

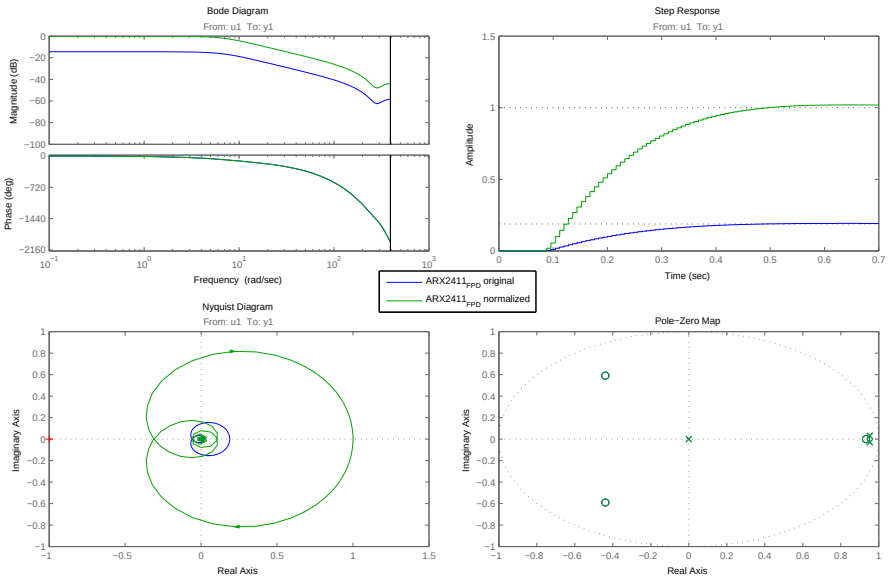

Figure 3.13: Bode diagram, step response, Nyquist diagram and a pole-zero map for $ARX2411_{FPD}$.

Looking at figure 3.14, the magnitude of the normalized model and the original model are much closer to each other. Obviously, this is because the steady state for the $ARX2411_{MODEC}$ is much closer to 1.

Normalizing the models does not affect the phase, but it does change the stability margin. This can be observed in the Nyquist diagram, where the normalized model apparently moves closer to -1. The poles and zeroes are not affected by the normalizing procedure and since they are within the unit circle at the same time as the Nyquist curve is to the right of -1, the models are considered to be stable.

According to (3.9), a general ARX-model can be re-written as

$$A(q)y(t) = B(q)u(t) + e(t) \tag{3.23}$$

The A- and B-polynomials for the normalized and original $ARX2411_{FPD}$ and $ARX2411_{MODEC}$ are shown in table 3.4 and 3.5.

Model	**A(q)**
Original $ARX2411_{FPD}$	1 - 1.905 q^{-1} + 0.9081q^{-2}
Normalized $ARX2411_{FPD}$	1 - 1.905 q^{-1} + 0.9081q^{-2}
Original $ARX2411_{MODEC}$	1 - 1.843 q^{-1} + 0.8509q^{-2}
Normalized $ARX2411_{MODEC}$	1 - 1.843 q^{-1} + 0.8509q^{-2}

Table 3.4: A-polynomials for original and normalized $ARX2411_{FPD}$ and $ARX2411_{MODEC}$

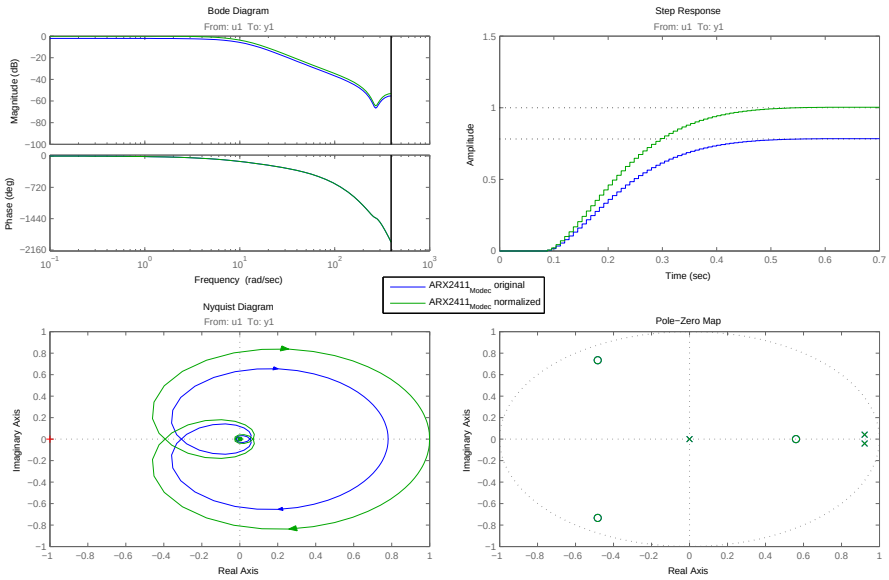

Figure 3.14: Bode diagram, step response, Nyquist diagram and a pole-zero map for $ARX2411_{MODEC}$.

As expected, the normalization caused no changes in the A-polynomials (see 3.22). In the B-polynomials, however, the difference between the original and normalized ARX-models is significant. Since the steady-state value for $ARX2411_{FPD}$ is as low as $18,9\%$, its B-polynomial increased substantially after normalization. The B-polynomial related to $ARX2411_{MODEC}$ did also increase after normalization, but the difference here is not as large, since its steady-state value was closer to 1 for the original model.

Model	**B(q)**
Original $ARX2411_{FPD}$	$0.00365\ \mathrm{q}^{-11} - 0.0001957q^{-12} - 0.001016q^{-13} - 0.001843q^{-14}$
Normalized $ARX2411_{FPD}$	$0.01934\ \mathrm{q}^{-11} - 0.001037q^{-12} - 0.005382q^{-13} - 0.009764q^{-14}$
Original $ARX2411_{MODEC}$	$0.005073\ \mathrm{q}^{-11} + 0.002064q^{-12} + 0.001168q^{-13} - 0.002195q^{-14}$
Normalized $ARX2411_{MODEC}$	$0.006493\ \mathrm{q}^{-11} + 0.002642q^{-12} + 0.001495q^{-13} - 0.002809q^{-14}$

Table 3.5: B-polynomials for original and normalized $ARX2411_{FPD}$ and $ARX2411_{MODEC}$

Another part of the model validation was to confirm that the models were controllable and observable. This was done by the Matlab commands *ctrb* and *obsv*, respectively. The command *ctrb* computes the controllability matrix ζ and *obsv* computes the observability matrix Ω. Observability is necessary for calculations in MPC (see section 4.2.1).

3.5 Results

This section provides the results from the system identification. The results from both the FPD measurements and the MODEC simulation are presented and compared. The model validation results have already been demonstrated in the previous section.

Figure 3.15 shows the scenario that has been used for result analysis. The input signals represents disturbances of different magnitude. The aim was to find a model that catches the systems behaviour for both small and large disturbances.

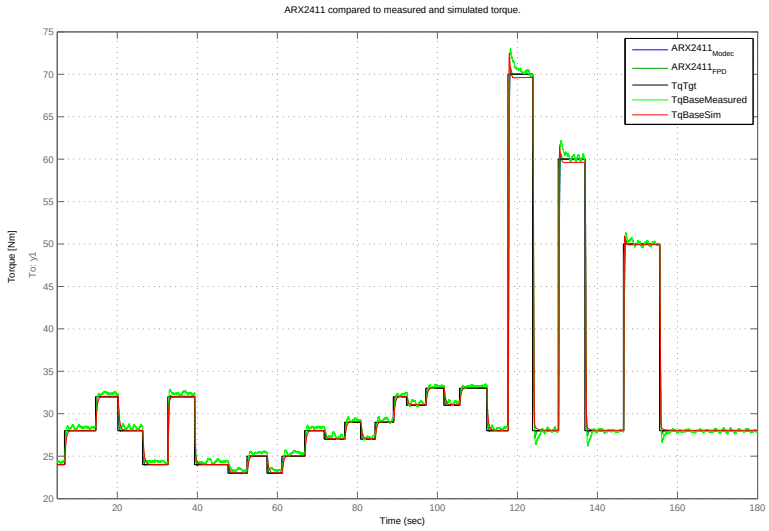

Figure 3.15: Scenario for result analysis.

Figure 3.16, illustrates selected parts from the scenario above. Both the ARX2411-models are compared to the measured and simulated torque. The comparison focus at two different disturbances; 32 Nm represents a large disturbance and 8 Nm represents a small. Since the characteristics for the positive flank and negative flank differ, these parts are plotted separately.

$ARX2411_{FPD}$ has a slightly shorter dominant time constant than $ARX2411_{MODEC}$ and as expected this can be seen in figure 3.16. The two ARX-models are very similar despite the fact that their properties differs a lot (see table 3.4.2). The dominant time constants were crucial in the process of model selection, since this play an important role in describing the system. Both models have one sample point (8 ms) extra delay compared to the true delay.

For the large disturbance of 32 Nm, (Figure 3.16 (a) and (b)), the ARX-models are too slow on the way up (a), but slightly too fast on the way down (b). This is a price we had to pay for using a linear model, representing a non-

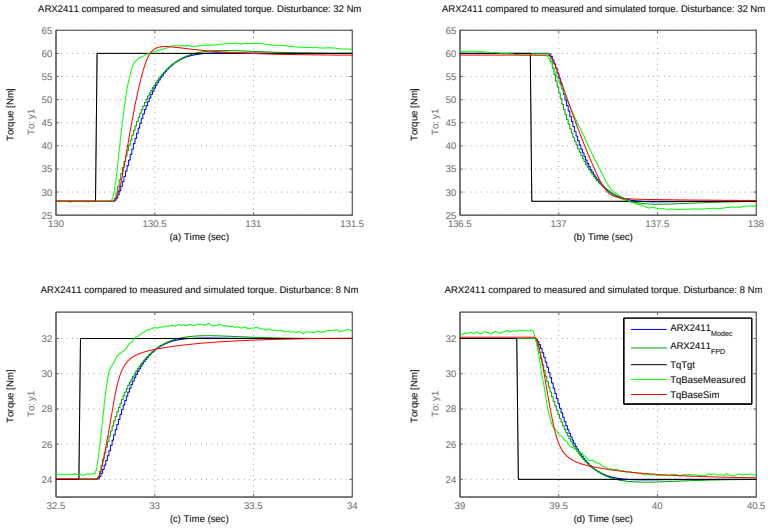

Figure 3.16: ARX2411 compared to measured and simulated torque for disturbances 8 and 32 Nm.

linear system.

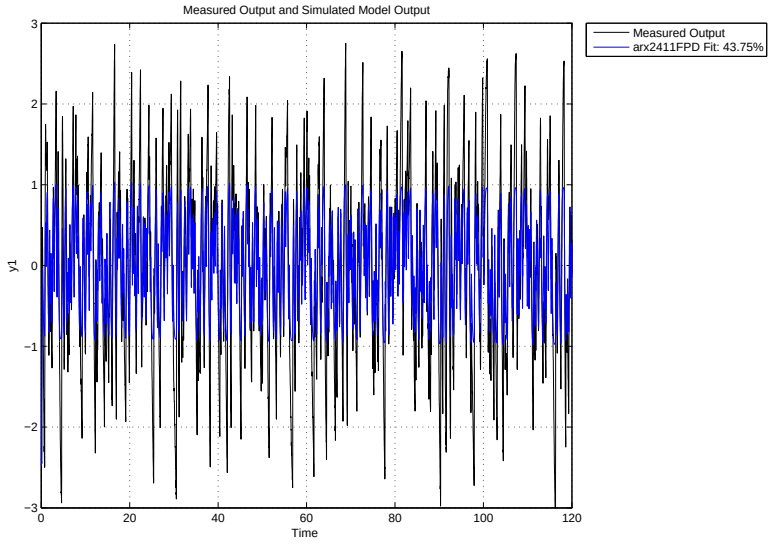

Figure 3.17: Comparison between $ARX2411_{FPD}$ and the measured output.

Figure 3.17 and 3.18 show the overall fit for $ARX2411_{FPD}$ and $ARX2411_{MODEC}$, respectively. The simulated output results from the models are here compared

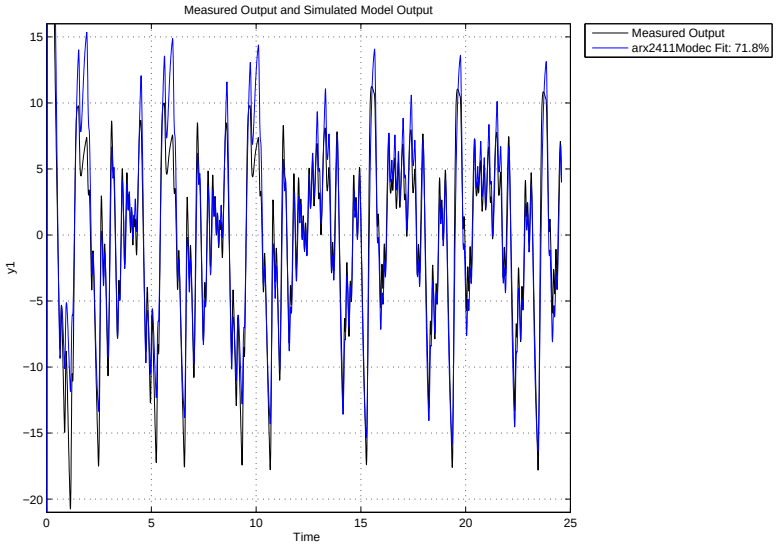

Figure 3.18: Comparison between $ARX2411_{MODEC}$ and the measured output.

to the measured output results. In figure 3.17 it is clear that the steady-state for $ARX2411_{FPD}$ is very low (see also table 3.4.2), but the model manage to follow the output characteristics in a relatively satisfying way. For $ARX2411_{MODEC}$ the steady-state value is higher, which is illustrated in figure 3.18. It is hard to tell the difference between the models when it comes to following high-frequency dynamics. Both $ARX2411_{FPD}$ and $ARX2411_{MODEC}$ are considered to imitate the output characteristics quite well.

Chapter 4

MPC

In this chapter a thorough examination of the control strategy of model predictive control will be made. First the background of MPC along with aspects as early history of MPC, advantages using MPC etc. will be presented. Secondly a review of the available theoretical framework will be outlined. Thereafter a modelling part along with a simulation section will be presented which deals with the construction phase and the outline and the conditions present during the simulations. Finally the results of the simulations will be presented followed up by a discussion of ditto.

4.1 Background

In this section a brief survey of the historical background for model predictive control will be given. This will be presented along with a definition of the term model predictive control, its use in the industry today and also some thoughts about the future of MPC. Some advantages of using MPC compared with one of the most widespread control strategies, namely PID control, will also be presented. As main reference in the theoretical part of Model Predictive Control, J.M. Maciejowski's Predictive Control with Constraints, from 2002, has been used.

4.1.1 What is Model Predictive Control (MPC)?

There are many different kinds of control strategies out at the market today. Many of the strategies are quite old and well tested, but there are also a few upcoming, not that old, methods trying to make new ways on the field of automatic control. Many of these relatively recent methods of automatic control are based on some kind of predictive control; MPC belongs to this group of methods.

4.1.2 Predictive control in the control hierarchy

Today's typical use of model predictive control in the process industry can be seen in figure 4.1. The top level in the hierarchy contains the determination of set-points, usually by means of steady-state optimization, which in turn may be performed at more than one level; plant-wide static set-point optimization

may be performed daily, while the set-point optimization at unit level may be performed hourly or even more often. This set-point optimization is based upon economic requirements but it excludes, usually, the dynamic behaviour of the system (plant).

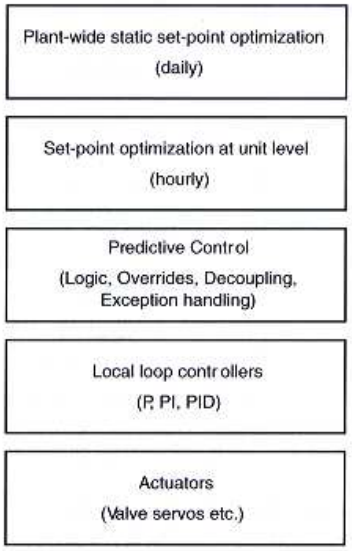

Figure 4.1: Today's typical use of model predictive control [Mac02]

Below predictive control in the hierarchy, traditional local controllers, which control for instance temperatures, flows and pressures, can be found. Typically these controllers consist of proportional and proportional-integral (PI) parts or sometimes even a part with derivative action added (PID). At the lowest layer in the hierarchy all the actuators, associated with individual control loops, are situated.

At the position where the predictive control is located, traditionally there is a complex layer containing overrides, exception handling, decoupling networks and logic, to deal with the variety of conditions which set-point single loop collaboration can not take care of. Usually this kind of layer is compound by a series of features constructed to solve individual problems and so tends to evolve during the lifetime of the system or plant. The behaviour of the entire plant is unlikely to be included in each solution which results in a badly optimized layer. Predictive control can thus be seen as a potential, and also powerful, link between the levels of set-point optimization at unit level and local loop controllers in the process industry. Figure 4.2 shows the future use of Model Predictive Control.

The reason why predictive control is being more and more accepted in the industry is that it is usually implemented on top of traditional local controllers. The method of predictive control is a highly integrated solution handling the

most common kind of exception handling problems and can thus provide a much better performance than the technology which it is replacing. When the predictive control technology is applied on top of the usually very reliable local loop controllers, the operating companies is allowed to increase their audacity in introducing this relatively new technology. If the predictive controller starts to misbehave, it is relatively easy to disable it and let the local loop controllers hold the plant at the last set-point and the process will still be safe to run.

In this analysis performed on the combustion engine trying to control the engine speed and the torque reserve the Model Predictive Controller will be used as a controller controlling local loop controllers, but also as a replacement of another more simple kind of controller; a PID type controller.

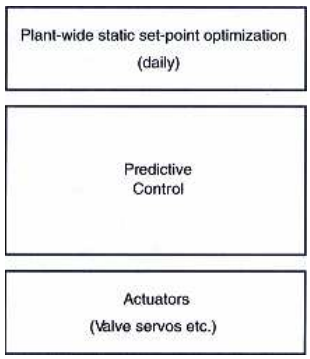

Figure 4.2: Future use of model predictive control [Mac02]

4.1.3 Advantages of MPC

The main reasons for the success of predictive control used in applications in the industry are several. The most important reason is the handling of constraints. Predictive controllers can take account of existing limitations of the actuators which are used to control the system. It will also allow operating-points closer to constraints, introduced to make the operation more profitable, in case of comparison to conventional control methods. Another important advantage of using model predictive control, which will also facilitate control of complex system, is that it handles multivariable control problems naturally.

4.2 Theory

Predictive control is model based since it uses an explicit internal model to generate predictions of future plant behaviour. This part containing the theory behind model predictive control will be revealing a variety of fundaments which is vital for both understanding and to be able to construct a model predictive controller. Closer examination of the receding horizon idea, the internal model, optimal inputs, optimization and constraints will be performed in this section.

4.2.1 Receding Horizon

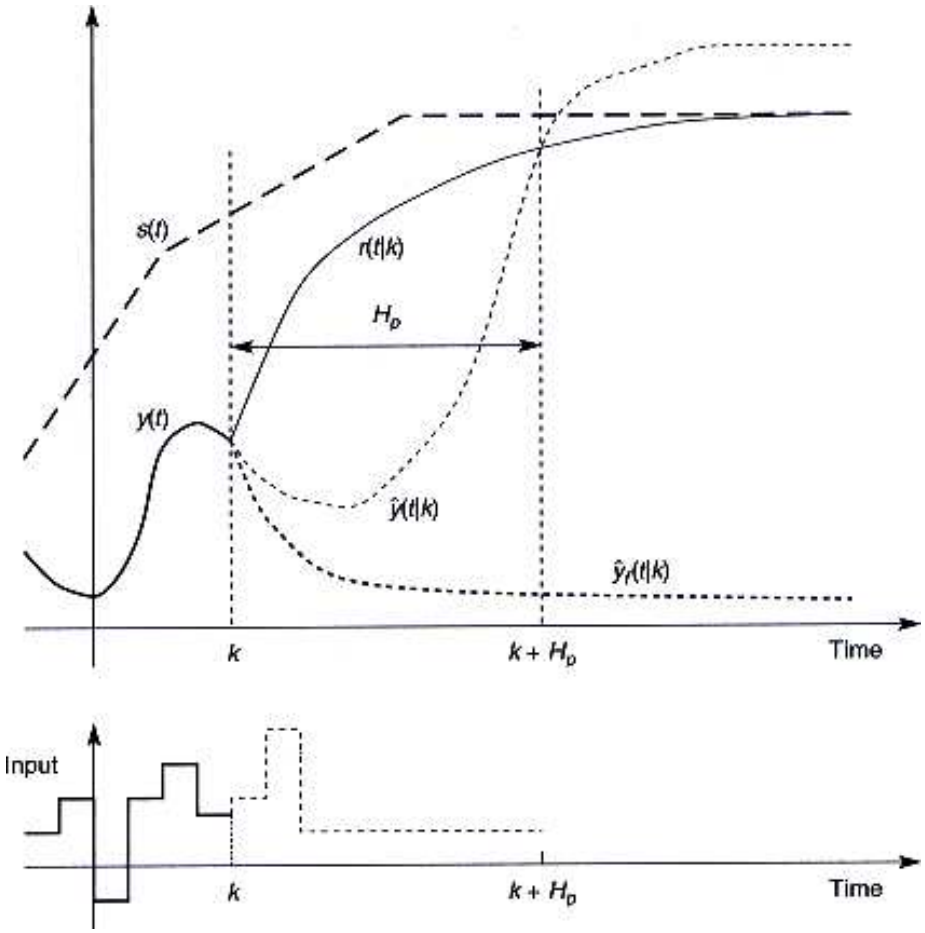

Figure 4.3: The basic idea of Model Predictive Control [Mac02]

The basic idea of model predictive control is illustrated in figure 4.3. In this overview of the general idea behind MPC, a plant configured as a system having one input signal as well as one output signal (SISO), is illustrated. $y(k)$ is denoted as the plant output the current time, where a discrete time setting is applied. The past output trajectory as well as a set-point trajectory, $s(t)$, which is the trajectory that the output should follow in case of ideal conditions is also shown in the figure. Denoted as $r(t|k)$ is the reference trajectory, which starts at the current output value $y(k)$ and defines an ideal trajectory along which the plant should return to the set-point trajectory. The plant is not necessarily driven back to the set-point trajectory as fast as possible which reveals the importance of the reference trajectory as a very vital part of the closed-loop mechanism. The reference trajectory is often assumed to be approaching the set-point exponentially from the current output value with a time constant of the exponential, T_{ref}, which also defines the speed of the actual response.

The existing current error 4.1, constitutes as a base for calculating the reference trajectory, which will be chosen so that the error would be like described in 4.2 if the output would follow it precisely.

$$\epsilon(k) = s(k) - y(k) \tag{4.1}$$

$$\epsilon(k+i) = e^{-i\frac{T_s}{T_{ref}}}\epsilon(k) \tag{4.2}$$

where T_s is the sampling time. The reference trajectory can thus be described as

$$r(k+i|k) = s(k+i) - \epsilon(k+i) = s(k+i) - e^{-i\frac{T_s}{T_{ref}}}\epsilon(k) \tag{4.3}$$

According to the name of the controller, predictions are made starting at the current time over a prediction horizon, and these are made by an internal model. The assumed input trajectory states the predicted future values, which can be defined as $\hat{u}(k+i|k)$, where $i = 0, 1, \ldots, H_p - 1$, and will be applied over the entire prediction horizon. This strategy of predictive character aims to achieve a appropriate predicted behaviour.

One way of utilizing this strategy is to choose the input trajectory such as to bring the output value of the plant to the required reference value $r(k+H_p)$ at the end of the prediction horizon $k+H_p$. The result of this action is referred to having one single coincidence point at the time of $k+H_p$. Most often there are not only one way to achieve this goal and since there are more than one input trajectory that satiefies this aim, there are also more than one solution; $\hat{u}(k|k), \hat{u}(k+1|k), \ldots, \hat{u}(k+H_p-1|k)$. One have to make a choice what to prioritize and for instance one could choose the solution which requires the smallest input energy, but it is preferable to impose some kind of simple structure on the input trajectory.

In the case shown in figure 4.3, the input $\hat{u}$ is assumed to vary over the first three steps of the prediction horizon, but to remain constant thereafter: $\hat{u}(k+2|k) = \hat{u}(k+3|k) = \ldots \hat{u}(k+H_p-1|k)$, there are thus three parameters to choose; $\hat{u}(k|k), \hat{u}(k+1|k), \hat{u}(k+2|k)$. If the input is assumed to remain constant over the prediction horizon one receive a structure that is the simplest possible: $\hat{u}(k|k) = \hat{u}(k+1|k) = \cdots = \hat{u}(k+H_p-1|k)$. Since there is only one equation to satisfy, $\hat{u}(k|k)$ is the only parameter; there is a unique solution.

A input trajectory which is representing the future has now been chosen. In this trajectory the first element is applied to the plant representing the input signal; $\hat{u}(k) = u(k)$, where $u(k)$ is the signal which is actually applied. The first cycle of the algorithm is now to end. Every new sampling interval, this cycle is repeated, and thus performing the output measurement, prediction, and determination of input trajectory. The result of this cycle is a new output measurement $y(k+1)$, a new trajectory $r(k+i|k+1)$, predictions, a new input trajectory $\hat{u}(k+1+i|k+1)$ is chosen and in the end the next input is applied to the plant. If a plant is being controlled this way, it is often referred to receding horizon control since the length of the prediction horizon remains the same as before, but slides along by one sampling interval at each step.

Internal Model

The internal model is linear, which in turn makes the calculations of the best input relatively straightforward. Our internal model is strictly proper, that is that we have the output measurement $y(k)$ available when deciding the value of the input $u(k)$ and thus the model output $y(k)$ depends on past inputs $u(k-1), u(k-2), \ldots$, but not on the input $u(k)$. Further, the internal model has to be presented in a proven form granting observability to assure the prediction functionality of the internal model.

The internal model is formulated as a state-space model in discrete-time. The following notation in state-space-form is used:

$$x(k+1) = Ax(k) + Bu(k)$$

$$y(k) = C_y x(k)$$

$$z(k) = C_z x(k)$$

where z contains the outputs which are supposed to be controlled, either to particular set-points or to satisfy some constraints, or both.

4.2.2 Optimization and Constraints

An important function of the model predictive controller is how to make the movements of the manipulated variables not only in a desired way, without violating different constraints, but also how to perform this in an optimal way not to make the movements more cost full than necessary. Therefore, set-point tracking, different kinds of constraints and cost functions will be dealt with in this part.

Set-point tracking

A vital part of the predictive control is set-point tracking. A primary control objective is to force the plant outputs to track certain set-points. If a case of no constraints is considered the controller predicts, within the prediction horizon, how much the outputs will deviate from its set-points. Each deviation is multiplied with the outputs weights, and then this product is squared and summed up as 4.4. [Bem07]

$$S_y(k) = \sum_{i=1}^{H_p} \sum_{j=1}^{n_y} w_j^y [r_j(k+i) - y_j(k+i)]^2 \tag{4.4}$$

where k is the current sampling interval, $k+i$ is a future sampling interval within the prediction horizon, n_y is the number of plant outputs, H_p is the prediction horizon and w_j^y is the weight of output j.

There are several methods of set-point tracking; zones, funnels and coincidence points. In this thesis the focus has been at a funnel objective with a straight line boundary, since this is the way Matlab/Simulink:s Model Predictive Control Toolbox works. Figure 4.4 shows the strategy of funnels with a

straight line boundary.

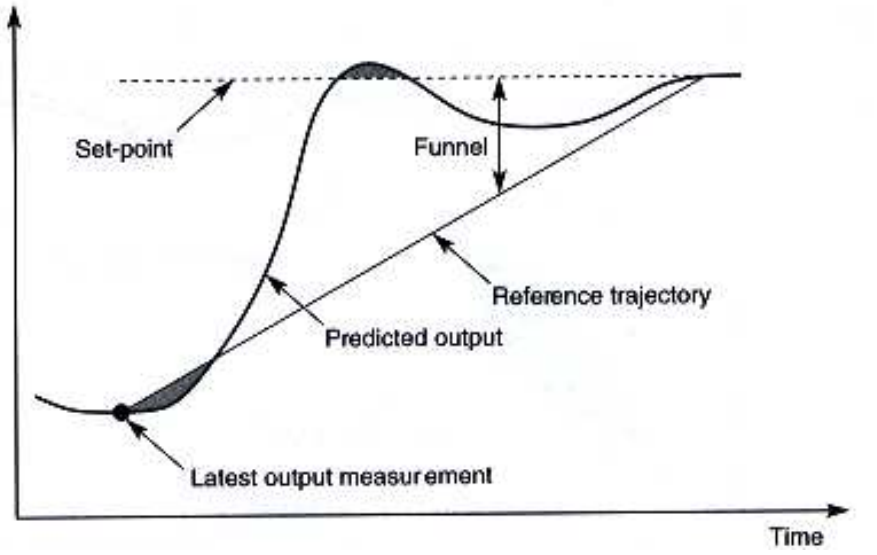

Figure 4.4: Set-point tracking using funnel objective [Mac02]

The controller might choose to make unnecessary large adjustments in the manipulated variables, if the controller exclusively focuses on the set-point tracking task, that could be impossible to achieve and lead to an unstable control system but also accelerate equipment wear. The Model Predictive Controller therefore manages a way of suppressing moves in the manipulated variables. This is done by monitoring a weighted sum of controller adjustments according to 4.5. [Bem07]

$$S_{\Delta u}(k) = \sum_{i=1}^{H_c} \sum_{j=1}^{n_{mv}} w_j^{\Delta u} \Delta u_j (k+i-1)^2 \tag{4.5}$$

where H_c is the control horizon, n_{mv} is the number of manipulated variables and $w_j^{\Delta u}$ is the weight put on the manipulated variables.

These weights put on the manipulated variables will have the effect of:

- Degraded set-point tracking of the controller.
- Less sensitiveness in the controller to inaccuracies in prediction.

The ability to move freely (within a constrained area) for the manipulated variables is most often not restricted, because the desire to compensate for set-point changes and disturbances. If a manipulated variable is being hold in a region the set-point tracking will also degrade, but some plants have more manipulated variables than output set-points. If all manipulated variables were allowed to move freely in such a plant, the values of the manipulated variables needed to achieve a particular set-point to reject for instance a disturbance

would be non unique; the manipulated variables would be drifting within the operating region. To address this, set-points for manipulated variables are defined as:

$$S_u(k) = \sum_{i=1}^{H_c} \sum_{j=1}^{n_{mv}} w_j^u [\bar{u}_j - u_j(k+i-1)] \tag{4.6}$$

where $\bar{u}_j$ is the manipulated variables set-point with w_j^u as the corresponding weight. [Bem07]

Constrained inputs/outputs

Constraints on the inputs and/or outputs forces a change from the simple least-square solution which is linear, to a least-squares solution that is constrained. In this case some kind of iterative optimization algorithm has to be employed because it is no longer possible to make use of a closed form solution. If the constraints are constructed as some kind of linear inequalities, the problem is of quadratic programming type. Since predictive control problems usually include constraints, the resulting control law is usually nonlinear. As long as the constraints are inactive the controller is linear and will act as if the controller was unconstrained, but if constraints become active then the controller is nonlinear, because the optimization function computes a nonlinear function of its inputs.

Constraints

Constraints exist in two types; hard and soft. Hard constraints must not be violated while soft constraints could be violated to some extent. Under some conditions a violation of a constraint is inevitable, for instance under influence of an unexpected large disturbance, and a realistic controller should be able to handle these kind of situations. This calls for softening of the otherwise hard constraints, by specifying the degree of softness for different constraints leading to acceptable violations and it is also possible to direct the violations to less dangerous areas, by specifying different tolerance bands of the constraints. [Mac02]

Cost function

Besides trying to reach the set-point there are also other kinds of control objectives to be satisfied. One of the most important control objectives is to minimize some kind of cost function. This function could be described in many ways but usually as some kind of quadratic criterion. The model predictive control action at time k is thus obtained by solving the optimization problem according to 4.7.

$$\min_{\Delta u(k|k),\ldots,\Delta u(m-1+k|k),\epsilon} \sum_{i=1}^{p-1} (\sum_{j=1}^{n_y} |w_{j+1,j}^y (y_j(k+i+1|k) - r_j(k+i+1))|^2 +$$

$$+ \sum_{j=1}^{n_{mv}} |w_{i,j}^{\Delta u} \Delta u_j(k+i|k)|^2 + \sum_{j=1}^{n_{mv}} |w_{i,j}^u (u_i(k+i|k) - \bar{u}_j(k+i))|^2) + p_\epsilon \epsilon^2 \tag{4.7}$$

This cost function provides the control strategy with penalties for neglecting the constraints, and it also gives the inputs/outputs penalties moving in a

non advantageous way. In this cost function, constants p_ϵ and ϵ are relaxation factors. [Bem07]

4.2.3 Stability

Predictive control is a type of feedback policy, when using the receding horizon idea. The resulting closed loop might suffer from instability, even though the performance of the plant is being optimized over the prediction horizon and even though the optimization keeps being repeated; each optimization will not consider the actions beyond the prediction horizon and will thus maybe position the plant in such a state that it will eventually be impossible to stabilize it. These kinds of situations are particularly likely to occur when there are several constraints on the available control input signals. One might also guess that a reason why this kind of problem can occur is for instance when the prediction horizon is too short, which in turn will result in a too short-sighted control horizon. The solution to very short prediction horizons is of course an increment of ditto; stability can usually be ensured by making the prediction horizon long enough or even infinite long.

There are several other ways of ensuring stability for the closed loop system. One way is to have any length of horizon, but to add some kind of terminal constraint which will force the state to take a particular value at the end of the prediction horizon, H_p. This is done by assuming that the optimization problem has a solution at each step, and that the global optimum can be found at each step. Constrained optimization problems in general can be difficult to solve, and just adding a terminal constraint could cause infeasibility in some cases. There is however possible to achieve stability without imposing conditions as severe as single-point terminal constraints.

There are several methods that could be regarded as modifying the weight on the terminal control error in some way. These methods are for instance the already mentioned way of imposing terminal constraints and the way of using infinite horizons, but it could also be done using fake algebraic Riccati equations. In the sense of modifying the weight put on the terminal control error the three different approaches are not that different from each other but one can note that the way of achieving stability through imposing terminal constraints is in general unnecessarily severe since one does not have to make the terminal weight infinitely large in order to obtain stability.

4.3 Controller design

A very vital part of the model predictive controller is the internal model used for estimation of future plant behaviour. The model, which was more carefully described earlier in chapter 2, consists of two major parts (air dynamics and ignition positioning). These parts were implemented in Matlab/Simulink. The part describing the air dynamics was identified and modelled as an ARX-model.

4.3.1 Internal model

The internal model, depicted in figure 4.5, was connected to the MPC-block in Simulink, to be able to use the design tool in MPC-toolbox. After having inserted the MPC-block in the Simulink diagram, the Input/Output (I/O) points were declared. The inputs, also called Manipulated Variables (MV), are ignition based, instantaneous torque, $TqInstTgt$, and the air based torque, $TqBaseTgt$, while the outputs are the engine speed, N_e, and the torque reserve, $TqRsv$. I/O points have to be declared since the design function in the MPC-toolbox will try to linearize the model and finally create a state space model representing the connected simulink system. This linearized state space model will then serve as the internal model embedded into the model predictive controller.

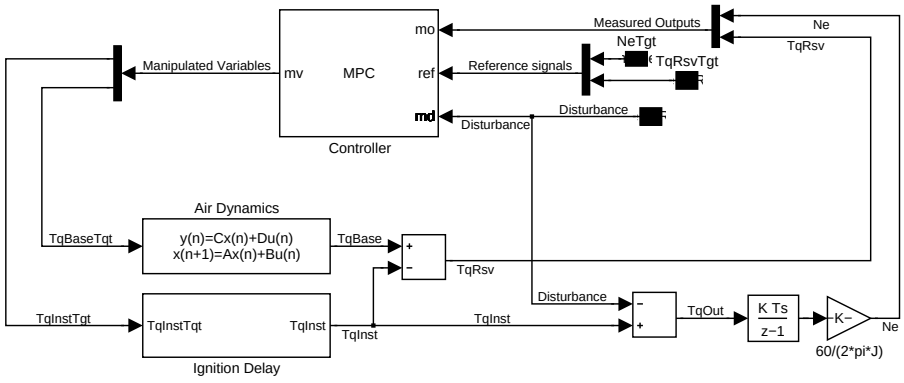

Figure 4.5: Simulink model of the internal model containing the MPC- block.

Operating point

The operating point of interest in this case is where the idle speed controller is supposed to operate the majority of its time in respect to N_e, $TqRsv$, $TqBase$ and $TqInst$. In case of no influence of external, output disturbances $TqInst$ has its operating point at $TqInst = 0$ Nm while in order to enforce the desired steady state torque reserve, both $TqBase$ and $TqRsv$ have their operating point at the desired nominal value of torque reserve; in the standard case $TqRsv = 8$ Nm.

Trimming the model

After having identified the operating point for the internal model, the Simulink model is subject to a trimming operation. During the design of the internal model for the model predictive controller in the MPC toolbox the command *trim* is used to compute the steady state values of the MPC-controller state for given input and output values. The exact description of the command *trim* taken from the MPC toolbox documentation is:

"The trim function finds a steady state values of the plant state vector such that $\dot{x} = Ax + Bu$, $y = Cx + Du$ or the best approximation of such an x in least squares sense, sets noise and disturbance model state at zero, and forms the extended state vector" [Bem07]

Behaviour of the internal model

In figure 4.6 the resulting behaviour of a unit step applied on the inputs $TqBase$ and $TqInst$ respectively, is shown. This is performed to receive some kind of validation of the plant behaviour. A step applied on $TqInst$ results in a ramp-like increment of the engine speed, while the resulting behaviour of the torque reserve is an inversion of the positive unit step. When the positive unit step is applied on $TqBase$ the result is a second order filter-like increment in the torque reserve, while the engine speed remains none affected. Overall plant behaviour was expected and thus the model behaves in a desired way.

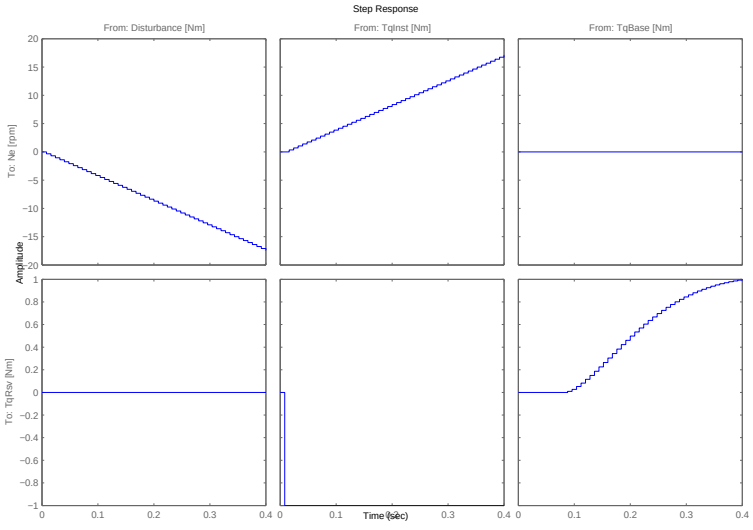

Figure 4.6: The result when applying a unit step to the input signals for the internal model

:

4.3.2 Controller construction

After having constructed an internal model that will be used for estimation purposes in the controller, there are still several parameters to set to make the controller work in a desired way.

Parameter selection

Parameters that have to be chosen during the controller construction phase are different horizons (how far in the future should the controller be able to predict plant behaviour?), set-points (target values for both input and output signals), input as well as output weights (which one of the I/O: s is more important when talking about set-point tracking?), and constraints (are there any constraints, or forbidden areas, for the signals?).

Sampling interval

The sampling interval is an important factor when choosing the upcoming prediction and control horizons. In today's existing idle speed controller a sampling interval of $T_s = 16$ ms is being used. Many of the systems onboard the vehicle has a sampling interval which is based on the engine speed, and not directly in time. This leads to increased availability in CPU power when the operating point of the system is located in an area where only lower engine speed is allowed. The chosen ARX-model for the air dynamics is triggered every $T_s = 8$ ms (the reason behind this has been declared earlier in chapter 3) and the model predictive controller will also be using this trigger timing.

Horizons

When trying to assure stability of the closed-loop system (plant including the controller), the length of the prediction as well as the control horizon are essential, since the controller when forcing the control signals to perform large changes can put the system in an unstable mode. The choice of prediction horizon was in this case to have a prediction horizon that is long enough to both be as fast as desirable but also assure stability for the closed-loop system. A prediction horizon of about $H_p = 35$ sample intervals was chosen in the simulations performed trying to suppress disturbances of different magnitudes. The length of the control horizon which also decides the response speed when trying to suppress different kinds of disturbances is also essential for the stability as earlier discussions statutes. The length of the control horizon was chosen to about $H_c = 5$ sample intervals. The mutual relation between the prediction horizon and the control horizon says that $H_c \ll H_p$ which is assured by the choices made.

Nominal values or Set-point selection

There are set-points which have to be chosen for each of the inputs and outputs. These set-points are the same as the operating points for the system. Set-points used in the basic simulation setup are shown in table 4.1. The reason behind the value $TqInst = 0$ Nm is that no excessive torque is needed to run the engine at $N_e = 650$ rpm; all internal disturbances, such as frictions etc., are compensated for. Nominal standard idle engine speed was set to $N_e = 650$ rpm and in the initial simulations the torque reserve was chosen as $TqRsv = 8$ Nm, because this value is also the default value for the PID controller.

Weights

Set-point tracking of the torque reserve is generally less important than ditto for the engine speed; the weight put on the engine speed should be higher than the weight associated with the torque reserve. In respect to the weights related to the input signals their task is secondary to the set-point tracking of the output signals. Hence, the resulting weights related to the input signals are set to zero

Signal	Set-point value
$TqBase$	8 Nm
$TqInst$	0 Nm
N_e	650 rpm
$TqRsv$	8 Nm

Table 4.1: I/O set-points used in the MPC.

to provide the largest possible quantity of freedom in the task of keeping the engine speed at $N_e = 650$ rpm as well as keeping the torque reserve at $TqRsv = 8$ Nm. In table 4.2 the values of the inputs/outputs related weights are compiled. An optimization algorithm has been applied to find the best values possible for each weight (see section 4.4.2).

Signal	Weight
$TqBase$	0
$TqInst$	0
N_e	1860
$TqRsv$	4.8

Table 4.2: Input and output variables weights.

Constraints

There are several constraints on the input as well as output signals, since certain movements are not allowed. Signals are not supposed to enter areas which are either not possible to enter, are adverse in some sense, or even hazardous. Constraints can be issued for entering a restricted area as a kind of hard boundaries, but also as regulations on the derivative of the movement on the signal. In the modelled system there are hard boundaries on the manipulated variables, namely -50 Nm$\leq TqBaseTgt \leq 150$ Nm, -50 Nm$\leq TqInstTgt \leq 150$ Nm. These constraints are due to functional restrictions, *max* and *min* values, and it is thus impossible to end up in the area outside these restrictions.

Constraints are also necessary on the output signals, to penalize entrance to certain areas. The torque reserve has a hard lower boundary that is practically impossible to violate; $TqRsv$ can not be negative, that is $1 \leq TqRsv$ is a valid lower limit that ensures a safety margin, which in turn implies that $TqInst$ can not be greater than $TqBase$ in any moment, $TqInst \leq TqBase$. As an upper boundary for the Torque reserve $TqRsv = 150$ Nm can not be infringed; $TqRsv \leq 150$ Nm.

Constraints related to the engine speed can be considered as more relaxed compared to other constraints. The upper boundary for the engine speed is set to $N_e = 750$ rpm but this is of minor importance. The lower boundary for the engine speed is set to $N_e = 500$ rpm and is preferred to be a very strict limit, but it's possible to violate this constraint in.

Not only constraints upon instantaneous values are introduced but also constraints related to the movement of the manipulated variables. These constraints

are denoted as $RateMin$ and $RateMax$ further on. As the constraints depend on the properties of the internal model of the model predictive controller, the constraints can be seen as limitations of the behaviour of the internal model. As mentioned earlier (in chapter 3), the air dynamics are not really as fast as desired responding to changes from negative to positive flank and not as slow as desired responding to changes from positive to negative flank. This can be compensated for, but only partially; the system can not be forced to respond faster, but it can be forced to respond slower. This calls for a constraint on movements in negative direction (defined as movements towards a smaller value). Therefore $TqBase$ has to be constrained introducing a $RateMin = -30$ (The value $TqBase = -30$ has been determined through an optimization algorithm, see section 4.4.2). All other signals are unconstrained with respect to $RateMin$ and $RateMax$. A summary of all constraints can be found in table 4.3.

Constraints	$TqBaseTgt$	$TqInstTgt$	N_e	$TqRsv$
Max	150 Nm	150 Nm	750 rpm	150 Nm
Min	-50 Nm	-50 Nm	500 rpm	1 Nm
RateMax	Inf Nm/unit	Inf Nm/unit	n.a.	n.a.
RateMin	-30 Nm/unit	-Inf Nm/unit	n.a.	n.a.

Table 4.3: Constraints on manipulated as well as output variables.

Constraints softening

In all cases, constraints of hard character are not necessary and sometimes one utilizes a kind of safety margins so that the constraints applied are a constrained zone with a certain width, rather than a constrained border with an infinitely small width. This way of exploiting constraints can be seen as applying soft constraints, or even softening the already hard constraints, rather than using hard constraints. There are yet another reason to change all hard constraints into soft ones, as discussed in earlier theory for stability of MPC, see section 4.2, namely to reduce the risk of infeasibility in the optimization algorithm. The higher the value of the relaxation factor the higher softness is applied, see table 4.4.

Constraint softening	$TqBaseTgt$	$TqInstTgt$	N_e	$TqRsv$
MaxECR	0.001	0.001	0.001	0.001
MinECR	0.001	0.001	0.001	0.001
RateMaxECR	$3.73 \cdot 10^{-4}$	0.001	n.a.	n.a.
RateMinECR	$2.20 \cdot 10^{-14}$	$2.20 \cdot 10^{-14}$	n.a.	n.a.

Table 4.4: Constraint softening on manipulated as well as output variables.

4.4 Simulation

This section will present the conditions under which the model predictive controller and also the PID controller were controlling both a simplified model and also the MODEC serving as the system. A disturbance scenario will be defined

facilitating comparison between different controllers. Finally a part dealing with the chosen strategy of finding the optimal parameter values for the model predictive controller will be presented.

4.4.1 Simulation prerequisites

Conditions which were applied to the organization of the simulations consisted of mainly two parts. First the MPC was connected to the plant which consists of the internal model which in turn means that plant equals the internal model and internal model errors are thus non-existent. Arrangements can be seen in figure 4.5. The decision to make this kind of previous validation of the MPC was due to the complexity provided by the more extended MODEC model which would make the survey more manageable but it would also offer the possibility to study the behaviour of the model predictive controller under the conditions where internal model errors were non-existent.

When tuning the MPC in a more rough way there is an advantage not having a too complex, but correct enough, plant used for simulations. The MPC was inserted in the arrangements provided in the Modec model, which implies that there are now a number of model errors affecting the performance of the model predictive controller. This model is also as close to the reality as the possibilities concedes.

When performing the simulations using both the simplified plant (internal MPC model equals the plant) and using the MODEC version a $TqRsv = 8$ Nm is being the standard magnitude of the torque reserve. This is valid for both MPC and the PID controller. Later on when both controllers behaviour have been examined and compared the torque reserve will be decreased in the sequence of $TqRsv = 8, 7, 6, 5$ Nm to see if it is possible to use a lower magnitude of torque reserve while still preserving the disturbance rejecting performance of the controller.

Scenario

To make it easier to compare the behaviour of the model predictive controller and the PID controller, the disturbances scenario were standardized. As the major disturbance affecting the engine is related to handling the steering wheel, the assumed appearance of this disturbance was approximated with a series of pulses. The reason choosing a pulse-like approximation is to resemble the reality, modelling the steering wheel servo, but also to create some kind of worst case scenario since the pulse-like approximation is more harmful than for instance a ramp-like ditto.

An important distinction to make is the one separating small and large disturbances. In this case the decision to categorize small disturbances as ≤ 10 Nm while large disturbances are ≥ 25 Nm. Finally a pulse train containing three small disturbances, 2, 4 and 8 Nm respectively, and three large disturbances, 20, 30 and 40 Nm respectively, was composed. In figure 4.7 the pulse train which

was applied as disturbances can be seen.

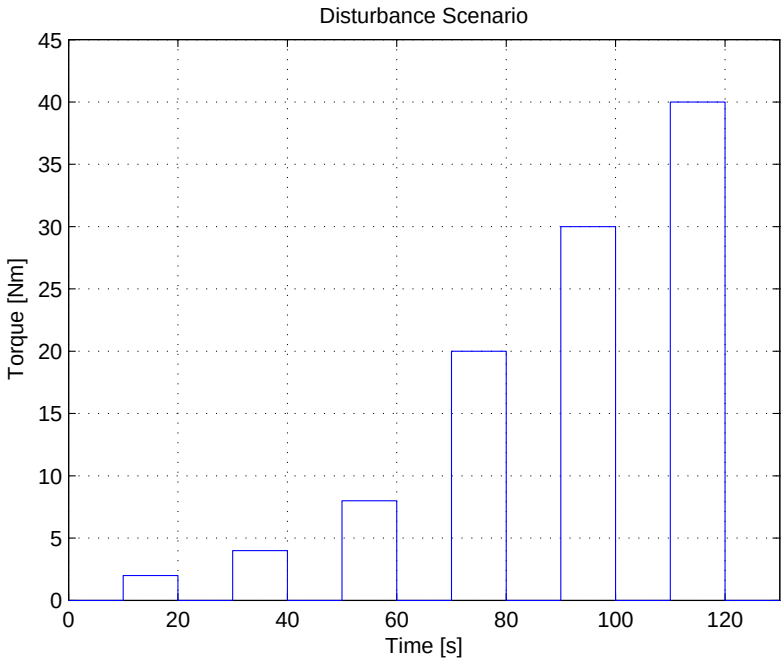

Figure 4.7: Pulse train describing the disturbance scenario used in the simulations.

4.4.2 Finding the optimal values for the parameters

There are several methods of finding the optimal values for the many parameters that are involved in the controller. In this thesis three different kind of strategies were applied; trial and error, least square error (LSQ) and finally one factor experiments. The first of these three methods was used only in the early starting phase of the tuning procedure to perform rougher tuning while the other two methods were used to fine tune the controller.

Optimization through least square error

One way of performing an optimization of the parameters used in the controller is to focus on the LSQ. This error is subject to the difference between the actual output value from the controller and a target function describing the desired value for the output. The objective for the optimization was to keep the engine speed, N_e, as close to the desired idle speed for the engine, $N_e = 650$ rpm, as possible. Realization of the LSQ algorithm in Matlab was done utilizing the *LSQ* command. This command provides the option to fine tune several of the parameters at the same time to be able to reach the final optimal value for each parameter. One major problem that occurred during the optimization process was that the starting values provided to the *LSQ* command most often resulted

in an optimal solution not too far away from the same values, almost no matter where the initial values were located. This indicates that there are a large number of local minima in which the optimization algorithm gets stuck. Thus there is a problem not being able to find the global minima, referring to the least square error, for the optimal values.

One factor experiment
Another way of finding the optimal values for the controller parameters is to make use of one factor experiments. The outline of the experiment was to change one parameter at the time, within a predefined area, while trying to track the effect of the changes and at the same time striving to minimize the error between the target function for the engine idle speed and the output value of the engine speed from the plant. This work is done very simple using *for*-loops in Matlab. The major disadvantage using this method is that it is very time consuming and not as effective as the *LSQ* command alternative.

4.5 Results

This section will provide the results from the earlier described scenarios and simulation conditions. Results from four different groups of simulation activities will be presented. First of all the MPC will be examined when trying to control the simplified model (denoted as plant) where the plant and the internal model of the MPC are equal. Secondly the MPC will be examined when striving to control the MODEC model which implies that the internal model is not equal to the plant. The performance of the MPC will be compared to the behaviour of the PID controller. The final intention is to present the results from the simulations showing the comparison between the MPC and the PID controller decreasing the level of torque reserve.

4.5.1 Simplified model

The MPC was connected to the plant (as in figure 4.5). As a disturbance scenario the one shown in figure 4.7 was applied. The result from these conditions can be seen in figure 4.8, where the behaviour of the output variables (OV) is plotted, and in figure 4.9, where the behaviour of the manipulated variables (MV) is plotted.

As can be seen in figure 4.8 the engine speed, N_e, returns quite fast to its set-point value and suppresses the disturbances in a desirable way. The MPC deals with small as well as large disturbances without violating too many constraints; in fact the torque reserve, $TqRsv$, adopts negative values which is a distinct violation of a constraint. The torque reserve is also responding very fast to the disturbances and for small ditto indications of a possible decrement of the set-point for the torque reserve is present. In figure 4.9 the control signal activity can be seen as a function of time. The control signals, both $TqBase$ and $TqInst$, follow their target values in a preferable way and the activity seems normal; no violations of constraints for the manipulated variables. This examination of the controller behaviour, when the MPC is trying to control the simplified plant, does mainly serve one purpose; to benchmark the early rougher

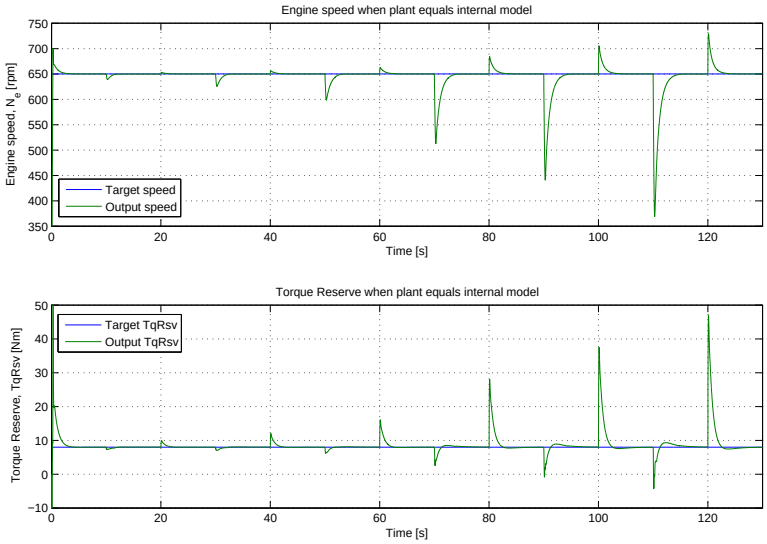

Figure 4.8: Behaviour of OV: s MPC controlling the simplified plant.

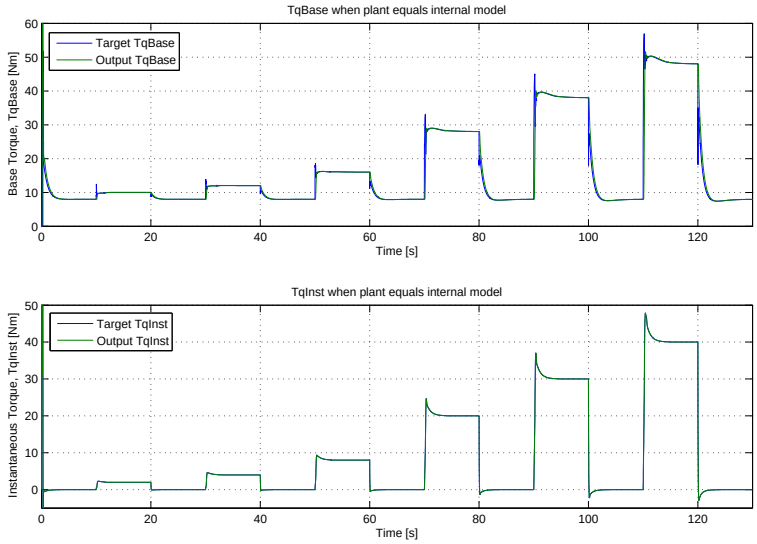

Figure 4.9: Behaviour of MV: s MPC controlling the simplified plant.

tuning of the model predictive controller parameters. The simplified plant itself is not valid enough to make any major conclusions as it differs a great deal from the controller performance when trying to control the more complex MODEC version of the plant. It is important to catch misbehaviour, such as constraint violations etc., in this early state of the simulation process to reduce the more

time consuming optimization process and make it less demanding if possible. Already in this state it is possible to see the importance of an accurate, and most of all fast enough, internal model dealing with the air dynamics of the plant. The control signal activity is much greater in respect to $TqBaseTgt$ than $TqInstTgt$ and there is also a much higher activity in $TqBaseTgt$ control signal than needed.

4.5.2 MODEC

The MPC was connected to the plant, which now consists of the MODEC model. Thisimplies that there is a significant difference between the internal model within the model predictive controller and the plant representing the physical engine. Disturbances influencing the output engine speed, N_e, behave as the scenario described in earlier which is also the same scenario as was used when examining the model predictive controller performance controlling the simplified plant.

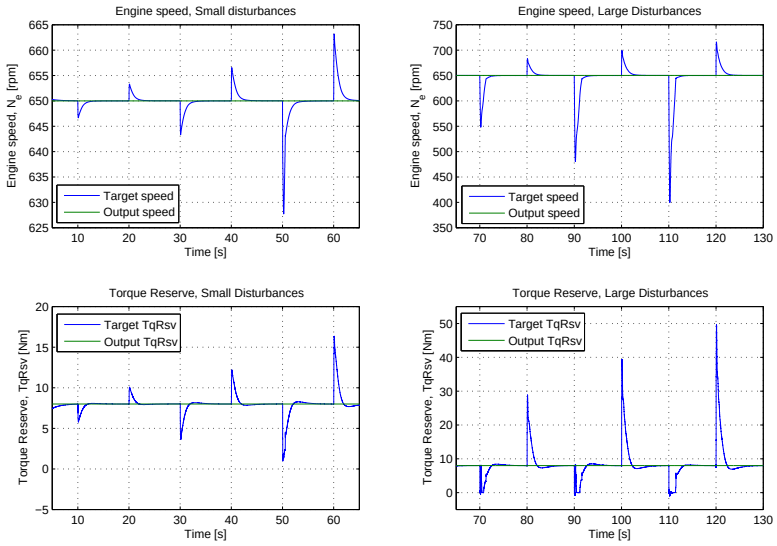

Figure 4.10: Engine speed, N_e, and torque reserve, $TqRsv$, affected by small and large disturbances respectively.

In figure 4.10 the engine speed, N_e, and the resulting torque reserve, $TqRsv$, is shown. When the controller is sensing small disturbances the fluctuation in engine speed is also small. These small disturbances, $2, 4, 8$ Nm do not have any major impact on the engine speed nor the torque reserve, as expected. The largest among the small disturbances, 8 Nm, forces the engine speed down to $N_e = 627$ rpm before returning to its set-point of $N_e = 650$ rpm only three seconds later. The corresponding torque reserve for this disturbance reaches its lowest value, $TqRsv = 1$ Nm before returning to its set point $TqRsv = 8$ Nm two seconds later. There are no violations of the constraints when the controller

only senses small disturbances.

Large disturbances have greater impact on the engine speed than the small ditto. When the controller is sensing large disturbances the engine speed drops to a much lower value than in the case of a small disturbance. For example when the disturbance of 30 Nm occurs the engine speed drops to about $N_e = 480$ rpm before returning to the set-point two seconds later. This drop in engine speed is also noted as a necessary violation of the lower boundary, $N_e = 500$ rpm, for the engine speed. In this case the disturbance is of such magnitude that all the available torque reserve is consumed (it reaches $TqRsv = 0$ Nm for more than 1 second). The focus will mainly be on the disturbance of magnitude 30 Nm in future examination and also in case of comparison to the PID controller performance later on.

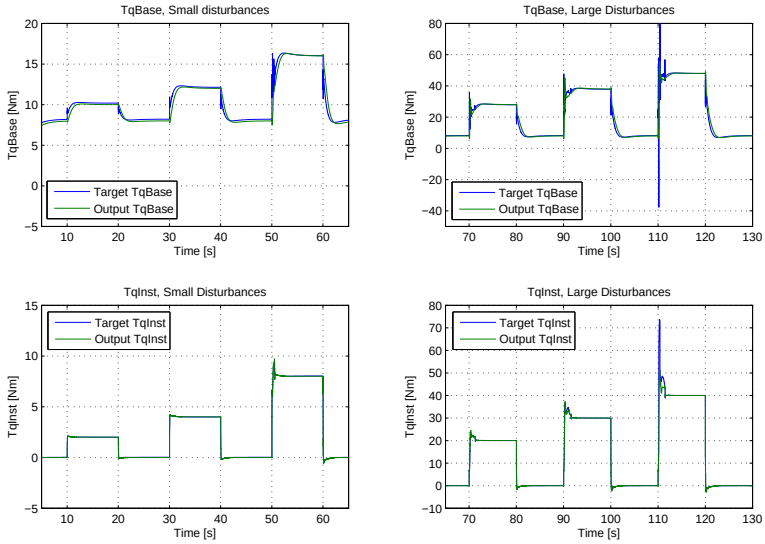

Figure 4.11: Manipulated variables, $TqBase$ and $TqInst$, behaviour when output signals are affected by small and large disturbances respectively.

Control signals, or manipulated variables activity is shown in figure 4.11. The performance of the control signals are in this figure divided into behaviour during small and large disturbances. During small disturbances the control signal activity is generally small, while during large disturbances much higher. The activity is maybe even higher than what is needed to achieve the same performance, since there are substantial differences between target and actual value.

In figure 4.12 (small disturbances) and figure 4.13 (large disturbances) throttle angle opening, average air mass flow into cylinder, different ignition angles and different torque efficiencies can be seen. These parameter values are the result of the disturbance scenario applied on the output variable engine speed. Variations in the measured parameters in these figures are naturally very small

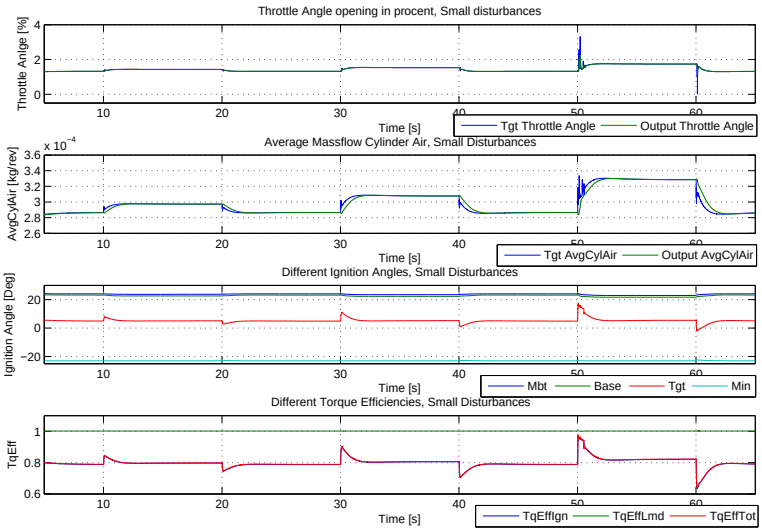

Figure 4.12: Closed loop system behaviour during small disturbances; throttle angle, average cylinger air, ignition angle and ignition efficiency.

when there are influences of categorized small disturbances present, while experiencing large disturbances the variations become greater. Looking deeper into the behaviour of the throttle opening angle and the ignition angle, when the large interesting disturbance of 30 Nm affects the engine output torque at the time of $t = 90$ s the throttle angle opening increases rapidly from a very small opening percentage to about 10%. At the same time the ignition efficiency changes quickly towards its optimal point as the ignition angle strives towards its optimal values.

4.5.3 Comparison to PID controller

To be able to evaluate the performance of the model predictive controller some kind of comparison has to be made to the existing PID controller. In this part such a comparison will be made where both small and large disturbances will be examined.

In figure 4.14 the PID controller performance, compared to the MPC, of tracking engine speed and torque reserve under similar conditions is shown. The engine speed tracking is slower for PID which naturally results in poorer ability of rejecting both small and large disturbances, but in return the effect in terms of control signal activity, figure 4.15, will be less when rejecting larger disturbances. It takes more than twice the time for the PID controller compared to the MPC to reach the set-point value after each disturbance entrance. In all six disturbance cases the magnitude of the speed is less for the model predictive controller than the PID controller. Regarding the non PID-like behaviour

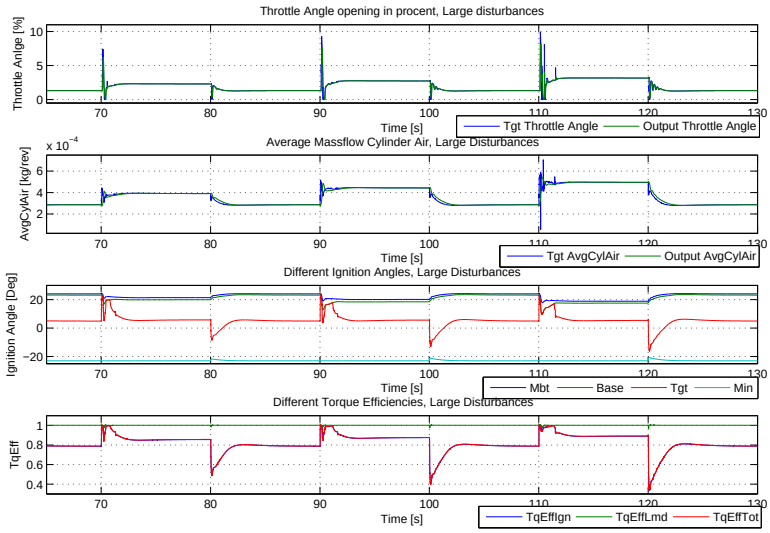

Figure 4.13: Cloesd loop system behaviour during large disturbances; throttle angle, average cylinger air, ignition angle and ignition efficiency.

with respect to the torque reserve, seen in figure 4.14, there is an extra feature implemented in the existing idle speed controller which provides the PID with an extra torque reserve in case of influence of a significant large disturbance.

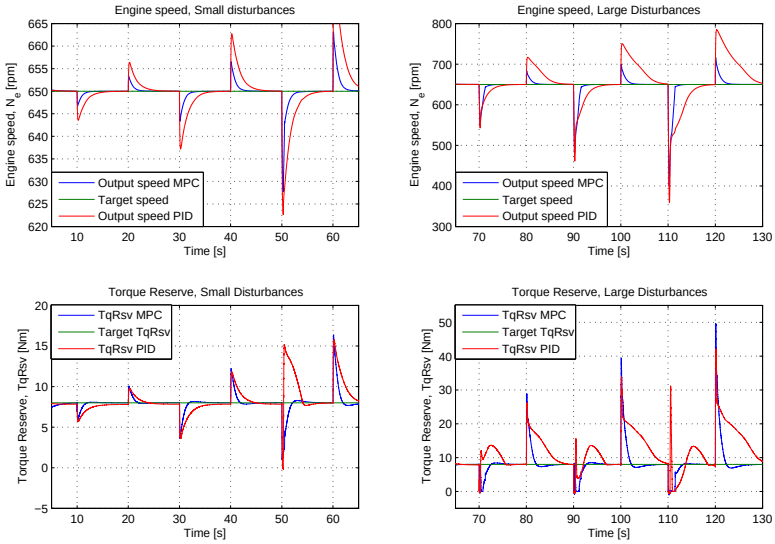

Figure 4.14: Comparison of engine speed and torque reserve set-point tracking for MPC and PID controllers.

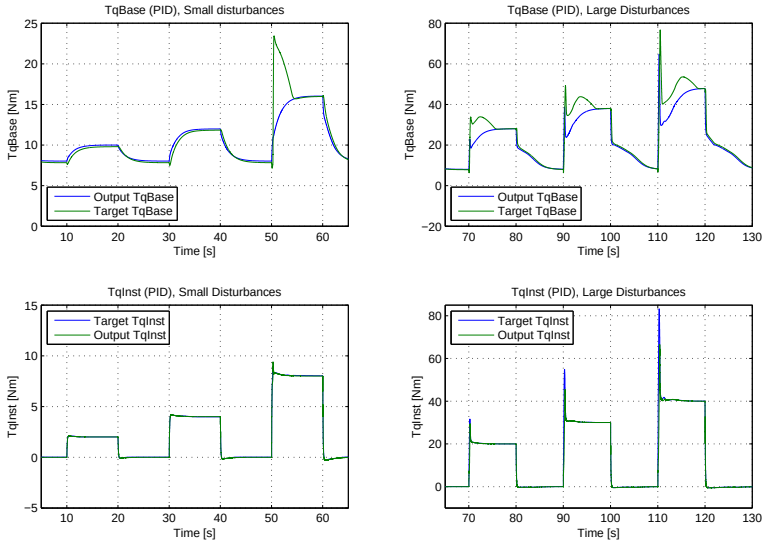

Figure 4.15: Control signal activity at small and large disturbances, respectively for PID controller.

To clarify the comparison between the MPC and the PID controller the focus will now be on the disturbance having the magnitude of 30 Nm. In figure 4.16 a closer examination of the controller behaviour at this time is shown.

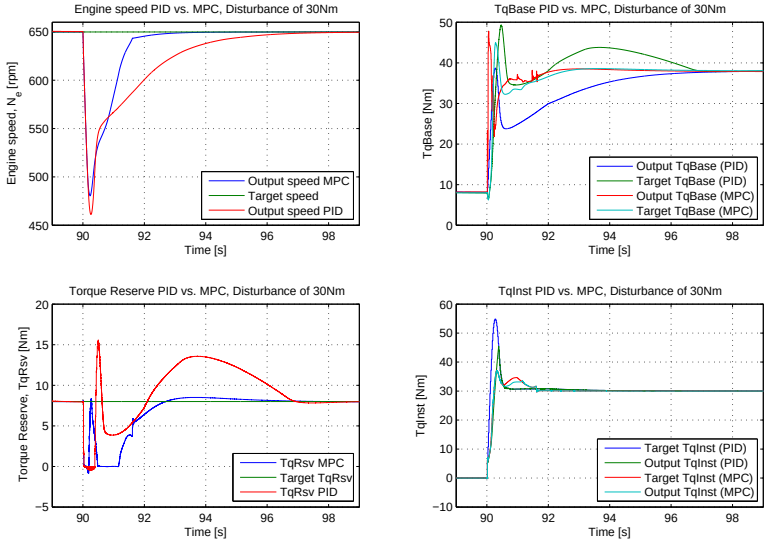

Figure 4.16: Comparison between PID and MPC closed loop system behaviour during influence of a 30 Nm disturbance

4.5.4 Extended scenarios

Now when the MPC and the PID controller has been compared, it may be interesting to examine the impact of the nominal values, or set-point, for the torque reserve, $TqRsv$. Figures 4.17, 4.18, 4.19 and 4.20, shows the results of simulations using the sequence of $TqRsv = 8, 7, 6, 5$ Nm as nominal values for the torque reserve. The overall behaviour of the model predictive controller in all these cases, varying the nominal value of the torque reserve, is quite similar. With respect to the engine speed there are two major effects coming from decreasing the nominal $TqRsv$ value which are the magnitude of the lowest point of the engine speed in the interval but also the time which the controller needs to be able to return to the set-point; decreased nominal $TqRsv$ leads to a lower lowest engine speed value and also a larger overall correction time to suppress the disturbance. When comparing this performance of the MPC to the PID controller it seems possible to use a MPC with a decreased nominal $TqRsv$ value to about $TqRsv = 5$ Nm and at the same time be able to not let the engine speed below the lowest value for the PID controller that in turn make use of $TqRsv = 8$ Nm as its nominal value. The lowest engine speed value under influence of a 30 Nm disturbance for the PID controller is about $N_e = 461$ rpm ($TqRsv = 8$ Nm) while ditto for the model predictive controller is $N_e = 467$ rpm ($TqRsv = 5$ Nm).

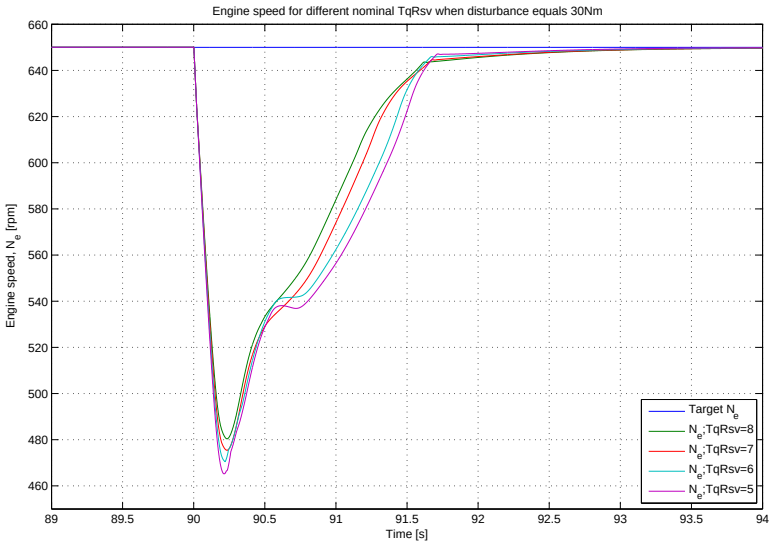

Figure 4.17: Engine speed for different nominal values of torque reserve when controlled by MPC.

There are also some significant differences in the control signal activity, see figures 4.19 and 4.20, comparing the cases when using different nominal value of the torque reserve. When this particular nominal value is decreased the control signal activities tend to increase, which can be explained by reduction of the margins available to suppress disturbances and especially large disturbances;

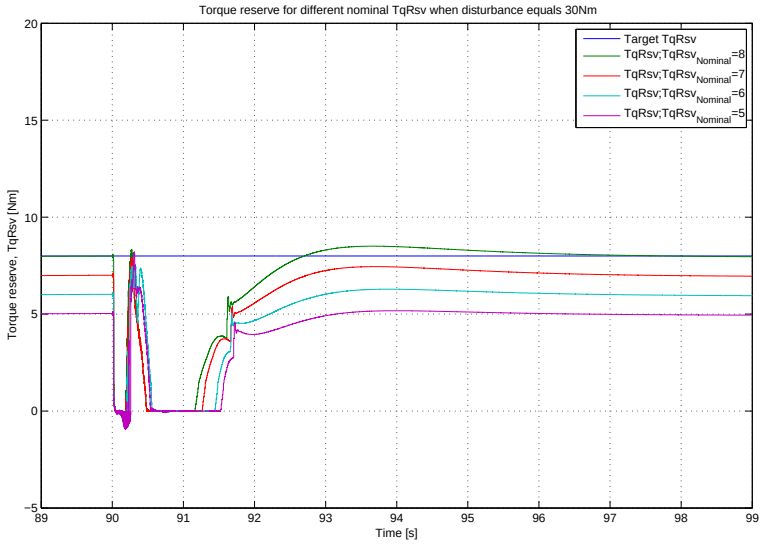

Figure 4.18: Torque reserve for different nominal values of torque reserve when controlled by MPC.

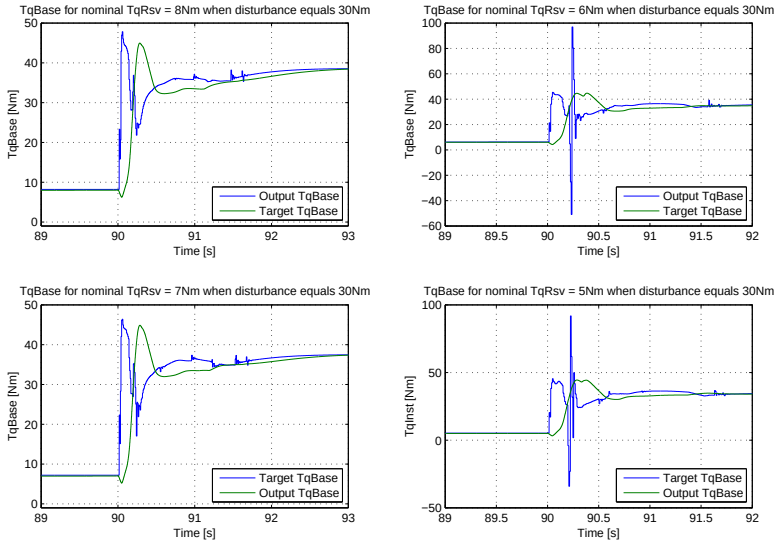

Figure 4.19: $TqBase$ for different nominal values of torque reserve when controlled by MPC.

the margins are still enough in the region of categorized smaller disturbances and will thus not have such a significant increment in control signal activity under those conditions.

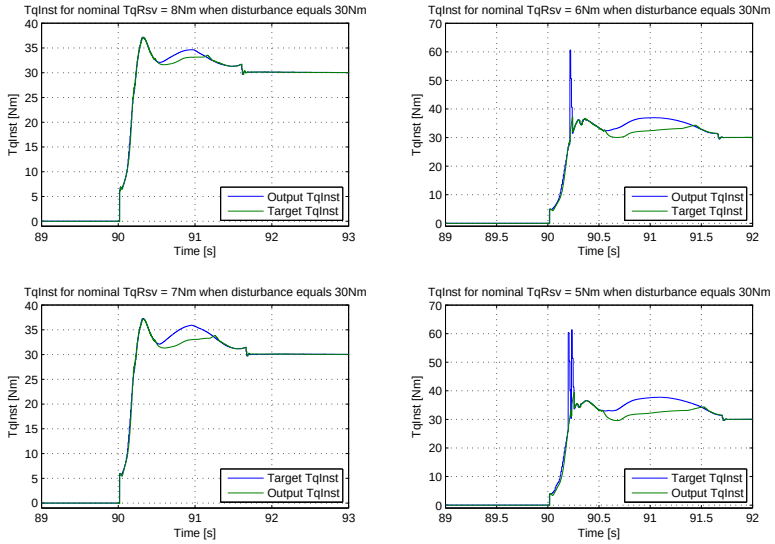

Figure 4.20: $TqInst$ for different nominal values of torque reserve when controlled by MPC.

The risk of ending up in constraint violations increases when the safety margins are decreased, and in case of the nominal $TqRsv = 5$ Nm, even drastically decreased (almost 40% lowered torque reserve). When the nominal value of $TqRsv = 5$ Nm is applied the manipulated variable $TqBaseTgt$ is in the vicinity of its lower boundary -50 Nm. Indications of constraints violations in any of the manipulated variables are not present. On the other hand there are constraint violations with respect to the engine speeds lower boundary but this is inevitable since there is not enough buffer of torque reserve available to fend off the large disturbance.

Chapter 5

Conclusion and future work

5.1 Conclusion

In this master thesis a model predictive idle speed controller was constructed based on a black box identification of the engine intake air mass flow dynamics. This model predictive controller was then allowed to control two different kinds of plants or system models; a simplified version and the more advanced version called MODEC, representing the engine. The target for the controller was to be able to withstand disturbances, of a on beforehand determined quantity and characteristics, without letting the engine speed during idle conditions assume too low levels and at the same time managing to preserve the level of torque reserve needed at a reasonable low level.

The results from the simulations of the model predictive controller under influence of a predetermined disturbance scenario shows that the engine speed at idle conditions is faster re-established to the desired set-point value when compared to the today used PID controller as a reference in case of using the same nominal value for the torque reserve for both controllers. Comparing the lowest assumed value of the engine idle speed for MPC and PID gives a significant advantage for the model predictive controller.

Examining the role of the nominal value for the torque reserve resulted in a possible reduction of the nominal value for the torque reserve, from 8 Nm to 5 Nm, without being forced to forgo the advantageous aspects for the MPC compared to the PID controller. This fact implies in an early state that the probability of being able to reduce the fuel consumption during idle speed is considerable, when not having performed tests in a vehicle. Hence, the potential of implementing a proper model predictive controller as a replacement of the PID controller that is being used today has to be considered plausible.

When performing the system identification of the intake air mass flow dynamics, difficulties were encountered which made it difficult to create a model of this part of the system that was of such quality that it could meet the expectations. The received ARX-model from the system identification part serves as a limitation of the performance of the model predictive controller, which can

be considered one of the plausible causes of the controller not being able to make use of an even further reduction of the nominal value put on the available amount of torque reserve.

5.2 Recommendations to future work

As we encountered different kinds of problem during this thesis work, some recommendations of future work can be made. The most important aspects are as follows:

- More thorough tests in vehicle so that better statistical models can be developed for use as internal models in the model predictive controller.

- If possible abandon the statistical ARX model and instead make way for a model based on the law of physics and thus can capture the non linearities of the system (both MODEC engine model and the physical engine).

- The result of the optimization strategy for the model predictive controller had the consequence that only local minima where investigated which implies that the possibility of optimal parameters are plausible. Hence, there is a need of some kind of optimization algorithm which has a global minimum as an outcome.

- Is there a possibility to reduce the computer power needed for the MPC further; can the length of the horizons be decreased not losing the advantageous performance?

- Improvements of the MODEC model can be made so that this model captures the physical behaviour of the engine in a more precise way. Implementation of model predictive controller in vehicle to be able to decide whether or not the MPC will outperform the existing PID controller.

- Is there a more efficient way of choosing control strategy than the focus put on the engine speed and the torque reserve?

Bibliography

[Bem07] Morari Manfred. Ricker N Lawrence. Bemporad, Alberto. *Model Predictive Control Toolbox 2. User's Guide.* The MathWorks Inc, 3 Apple Hill Drive, Natick, MA, 01760-2098, USA, 2007.

[Ben02] Lennartsson Bengt. *Reglerteknikens grunder.* Studentlitteratur, Lund, first edition, 2002.

[dK02] Craig IK. de Klerk, E. Closed-loop system identification of a mimo plant controlled by a mpc controller. *IEEE Africon*, 2002.

[Eri05] Nielsen Lars. Eriksson, Lars. *Modeling and Control of Internal Combustion Engines.* Vehicle Systems, ISY, Linköping Institute of Technology, Linköping, first edition, 2005.

[Lju99] Lennart. Ljung. *System Identification - Theory for the user.* Prentice Hall PTR, Upper Saddle River, New Jersey 07458, USA, second edition, 1999.

[Lju03] Glad Torkel. Ljung, Lennart. *Reglerteori: Flervariabla och olinjära metoder.* Studentlitteratur, Lund, first edition, 2003.

[Lju04] Glad Torkel. Ljung, Lennart. *Modellbygge och Simulering.* Studentlitteratur, Lund, first edition, 2004.

[Mac02] J.M. Maciejowski. *Predictive Control with Constraints.* Pearson Education Limited, Essex, England, first edition, 2002.

[Nor06] Österman J. Nordling, C. *Physics Handbook for Science and Engineering.* Studentlitteratur, Lund, eight edition, 2006.

[Pet06] Egardt B. Bruzelius F. Pettersson, S. *Automotive Control engineering: Powertrain Modelling Control.* Gröna Bilen, Göteborg, Sweden, 2006.

[Rob03] M.J. Roberts. *Signals and Systems: Analysis using transform methods and Matlab.* McGraw-Hill Education, Singapore, international edition, 2003.

[Rub07] Marcus. Rubensson. The engine model architecture (ema) project - concept and directions. Technical report, Volvo Cars Corporation, 2007.

Appendix A

Determining the moment of inertia

The moment of inertia, J, was determined for two reasons. First, a proper value for J was desired in the MODEC engine model and second, J was needed for converting the output torque into engine speed (see equation 2.13).

The determination of J was based on the data collected from experiment 2 (see section 3.3.2). From equation 2.11, the moment of inertia can be solved according to equation A.1

$$J\dot{\omega}_e = Tq \Leftrightarrow J = \frac{Tq}{\dot{\omega}_e} \tag{A.1}$$

A vector containing the measured engine speed as a function of time was stored and then differentiated to get the acceleration, $\dot{N}_e$. The new vector containing the acceleration was then filtered by the Matlab command *filtfilt* to reduce the high frequencies. The high frequencies due to measurement disturbances would make the results unreliable.

The measured output torque was stored after the means and trends was removed by the Matlab command *detrend*. A new vector with output torque as a function of time was then calculated for different values of J, according to equation A.1. The calculated torque was compared to the measured torque and analyzed by the least square method. The value of J that gave the most similar results between the calculated torque and the measured torque was considered to be the moment of inertia for the crank shaft dynamics. The moment of inertia was determined to be $J = 0.21914\ [kg \cdot m^2]$.