Master of Science Thesis in Electrical Engineering
Department of Electrical Engineering, Linköping University, 2020

Model-Predictive Control for Gas Exchange in a Gasoline Engine

George Jajji

Master of Science Thesis in Electrical Engineering

Model-Predictive Control for Gas Exchange in a Gasoline Engine:

George Jajji

LiTH-ISY-EX--20/5345--SE

Supervisor: **Robin Holmbom**
ISY, Linköpings universitet

Examiner: **Lars Eriksson**
ISY, Linköpings universitet

Division of Automatic Control
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping, Sweden

Copyright © 2020 George Jajji

Abstract

The process to induct air into engine cylinders, via the air inlet system and cylinder port valves, is referred to as the "gas-exchange". Control is achieved by the turbo-charger, the intake throttle plate and the variable valve timing (VVT) system. These actuation systems traditionally use separate control with independent SISO feedback. There are however physical couplings that affect the control performance. This thesis work looks at MPC control methods for a robust control strategy. MPC methods are typically used for systems with slow dynamics, due to computational limits. But new advances in CPU performance should allow for real-time implementations for engine control.

Sammanfattning

Processen för att suga in luft i motorcylindrarna, via luftinloppssystemet och cylinderportventilerna, kallas gasbytet". Styrning uppnås med turboladdaren, insugningsgasplattan och VVT-systemet (variable valve timing). Dessa aktuatorer använder traditionellt separat styrning med SISO-feedback. Det finns dock fysiska kopplingar som påverkar kontrollens prestanda. Detta examensarbete tittar på MPC-kontrollmetoder för en robust kontrollstrategi. MPC-metoder används vanligtvis för system med långsam dynamik på grund av beräkningsgränser. Men nya framsteg inom CPU-prestanda bör möjliggöra realtidsimplementeringar för motorstyrning.

Acknowledgements

I would like to thank Volvo Cars for giving me the opportunity to write this thesis. I also want to thank my supervisor at Linköping University, Robin Holmbom, who patiently guided and assisted me throughout this thesis. Finally I would like to thank my examiner Professor Lars Eriksson for making this thesis a reality and for his persistent enthusiasm on the subject.

George Jajji

Contents

1 Introduction

Due to technological advances in the automotive industry, engine control units are more powerful than ever. With such ECUs more advanced control systems for vehicles can be realized. The process to induct air into the cylinders, via the inlet system and cylinder port valves, is referred to as the gas-exchange. Controlling this process is done by the intake throttle plate, variable valve timing (VVT) and the turbo-charger. Today these actuators are controlled separately with single-input-single-output (SISO) controllers and deal with physical couplings with the help of look-up tables and/or calibration of the actuators which are to be controlled. For this reason it is desired to implement model based control (MPC) which is more robust and intended for multiple-input-multiple-output (MIMO) control strategies. A MPC scheme could deal with such physical couplings by design unlike for example a PID controller which requires external help with such couplings.

1.1 Problem Formulation

Because of legislation it is desirable to achieve better fuel economy in the automotive industry. Today the engines in the automotive industry are advanced and perform well, this can be indicated by the fact that today's commercial line engines fulfill their purpose and also fulfill at least Euro 6. However they are multi-variable systems that are controlled with single variable control which is non ideal. Engines keep advancing and are becoming more sophisticated and complex, therefore a SISO control strategy will inevitably become unviable for this type of application. Therefore there is a desire to develop a new control method which can handle multi-variable vehicle systems better.

1.2 Purpose and Objectives

Volvo Cars has requested a study around MPC for their gas-exchange system. The main purpose of this master thesis is to develop a MPC which is designed around dynamic state models for the gas-exchange system that is able to achieve better fuel economy. The aim of the MPC controller is to be robust and capable to control the system while having good performance at all operating points. The proposed MPC should have an easy to understand structure and connect to:

- Related research
- Defined ordinary differential equations for the system
- Defined control and feedback signals for the system

1.3 Related Work

In this section related research about the subject is presented. A summary is available for papers on MPC in a couple of fields and also papers about variable valve timing (VVT) are presented.

1.3.1 Model Predictive Control

Automotive

In the automotive industry the standard system for gas-exchange is a SISO-system. For example the throttle is controlled by an electric servo motor which is the standard in the automotive industry according to Eriksson and Nielsen (2014). In this case the input is a control signal which then results in one output, the throttle angle. This is typically what is meant by single-input-single-output systems.
Studies around MPC for gas-exchange in gasoline engines are few and not in depth. The approach in Bemporad et al. (2018) is general and explains the conditions and requirements necessary to achieve MPC. The article goes into specifics on how the MPC is structured and discusses chosen input and output signals. Furthermore they present that results are sufficiently good, which proves that a control strategy consisting of a MPC with the computational power available today is achievable.
A MPC requires an accurate model to work properly, Fantenberg (2018) is a master thesis that specifically designed models for the air mass flow but with dual independent VVT. The models in this thesis are designed with physical insight and grey-box approach and they show good results.
In S. Joe and Thomas A. (1997) it is stated that MPC technology was mainly used for power plants previously but now can be found in a variety of applications including the automotive industry. According to S. Joe and Thomas A. (1997) they have observed that process identification and control design are separated in current MPC technology and suggest that identification and control design could be integrated, producing significant benefits.

Von Wissel et al. (2016) explains that today's automotive industries must introduce advanced powertrain solutions to meet strict environmental regulations and market expectations. This results in powertrain architectures with high complexities, according to Von Wissel et al. (2016) this is why hardware and software development should not be decoupled anymore. Thus the development of hardware and software should run in parallel. Furthermore according to the paper the use of MPC in the automotive industry has increased due to advancement in MPC technology and due to the increased computing power of the ECUs. It is stated that at Renault several powertrain control projects are already using MPCs and that Renault uses the Honeywell OnRamp software toolset for the MPC part.
Honeywell has developed a software tool set which is called OnRAMP. It has been proven useful in several implementations. Honeywell claims that the software can help auto makers improve fuel economy, reduce material costs, and minimize warranty costs by optimizing powertrain controls, Honeywell (2015).
In Polterauer et al. (2019) the flexibility of model predictive control can be seen. The control problem for EcoDriving is formulated which utilizes the advantages of a MPC and emphasizes the need for a well thought out cost-funtion and usage of constraints. For example EcoDriving according to the formulation in Polterauer et al. (2019) can be achieved if fuel consumption and number of gear shifts can be minimized with respect to torque, number of gear shifts and brake force. The control problem must achieve this in subject to the vehicles dynamics, velocity limits, journey time and comfort.
Considering this kind of flexibility with the control problem in model predictive control one can say that the achieving good control results relies heavily on how the control problem is formulated and that several control strategies are possible.

Design parameters

Two important parameters are the receding horizon N and sampling time T_s. When configuring the MPC controller for automotive purposes they should be chosen so that the QP-solver used in the MPC law is within the MPC control period. The QP-solver has to be fast enough to provide solutions within sample intervals, Bemporad and Patrinos (2012). Furthermore depending on the cost-function and the control purpose the control period which is $\frac{N}{T_s}$ should be long enough to include the dynamics involved in the control scheme.
The penalty matrices Q_1 and Q_2 are weight matrix coefficients. The first one allows one to adjust the accuracy with which each out-put must track it's reference point and the second allows adjusting how aggressive control signal increments may be. Testing until desirable out-put behaviour is achieved is an important aspect of MPC application. Improper weight of the controlled states could result in the MPC not being able to control the out-puts to their references, Aimin et al. (2009).
When choosing state and input constraints there is a need to understand the systems physical limitations and consider the feasibility of the chosen constraints.

Polterauer et al. (2019) is a good paper to read about constraints in relation to cost-function within an automotive field.

General usage and requirements

In Nilsson (2018) it is concluded that an offline MPC is not ideal because of the heavy computational power required. They suggest that a control strategy with several MPCs working in parallel is more beneficial. Furthermore they use a PI-controller in cascade structure to support the MPC because of model errors.
In Jalali and Nadimi (2006) it is stated that MPC is adopted in industries as a mean to deal with multi-variable constrained control problems. The purpose of this paper is to give an overview of robustness in MPC technology and it also states that robustness to model uncertainties and noise in MPCs is questionable.
According to Yang and Boyd (2010) MPCs can usually only be used for systems with slow dynamics, due to computational limits. However they state that if a fast MPC is desired, a well known method to achieve this is to compute the control law offline and then implement it as a lookup table, which has obvious disadvantages according to the paper.
In Allgöwer et al. (1999) it is concluded that MPC for linear constrained systems is a good control solution practically. However the paper suggests the need for nonlinear MPC mainly because of the fact that higher performance specification are desirable. According to the paper this would mean that these kind of demands often only can be satisfied when nonlinearities and constraints are considered in the controller.
According to practitioners in Manfred and Jay H. (1997) the reason MPC is limited is not because of the control algorithms deficiencies but due to the fact there are difficulties in modeling, sensing, state estimation, etc. The paper states that model development is the most critical and time consuming step in implementation of MPCs. Furthermore the paper proposes several approaches to reduce the computational efforts by linearizing the system in some manner. One proposal is to apply a feedback linearization and then use MPC in a cascade arrangement for the resulting linear system. However this approach is limited to low order systems.
In Forbes et al. (2015) some of the drawbacks of having one MPC control an entire MIMO-system are considered. When a large applications performs poorly due to model mismatches it could be difficult to determine which sub-models need updating. The paper considers the advantages of a MPC divided into smaller systems.
Evaluating and updating sub-models can be time consuming, having several smaller MPCs working together, when one of the small MPC performs poorly there are fewer models to evaluate. Another advantage is the benefit of turning off one MPC and still be able to control the process to a certain degree, instead of turning of the control entirely.
According to Bemporad (2016) MPC is widely used in industries to optimize closed-loop response of multi-variable systems under constraints. A linear MPC requires a quadratic programming (QP) solver to compute the control signals.

According to Bemporad (2016) there are several good algorithms for QP available today. The paper presents different approaches to solving the QP-problem by transforming it into a Non-negative Least Squares (NNLS) problem.
In Bemporad and Patrinos (2012) it is stated that for a QP-solver to be implemented in the MPC law, it must be fast enough to provide solutions within sampling intervals. It also requires some hardware and memory. This is necessary to define the QP-problem and store the result and algorithm in code form to enable the MPC law. This subject is important in order to achieve hard real-time system requirements.

1.3.2 Variable Valve Timing

According to Eriksson and Nielsen (2014) cam phasing is common and can with help of an hydraulic actuator be implemented by rotating the camshaft in relation to the crank and timing belt.
In Li et al. (2009) it is stated that variable valve timing is another way to control the amount of fresh air in the engine. By decoupling the camshaft and crankshaft; intake and exhaust valve timing can be enabled. VVT allows control of the amount of fresh air and residual gases inside the cylinders. According to the paper VVT can reduce emissions, fuel consumption and engine noise.
In Leroy et al. (2009) the fresh air charge is considering commercial-line sensors and designed for real-time purposes. The model is composed of three terms and focuses a lot on the phasing of the exhaust and intake independently. The article concludes that the model has good physical behaviour and is sufficient for control purposes.
In Andreas (2018), they present an air charge estimator which is a physical model for estimating the in-cylinder trapped mass and residual gas fraction by using the cylinder pressure measurements and intake and exhaust valve lift profiles. The model is validated on a wide range of engine operating conditions and has a fairly low maximum error over these conditions. The model is designed keeping in mind commercial-line sensors. There are many qualitative papers and works which are related to the modeling of the system however there is not a lot of work that is related to the use of traditional MPCs in engine systems.

1.3.3 Placement of thesis work

The related work and research around the use of MPC is almost always around systems with slower dynamics than engines. However statements are made in several papers in the related work section that MPC methods could be used for fast systems like an engine. The work done in this thesis assumes that real implementations of MPC in vehicles with a gasoline engine is feasible.
The main problem in these type of systems are the nonlinearities. If a nonlinear MPC method for controlling an engine would be feasible it would enhance performance. However implementing such a controller is difficult and would perhaps not be computable with ECUs available in conventional vehicles. There are several approaches to linearize such a system and use a linear MPC. The ap-

proach in this thesis is not mentioned in related work, but the type of linear MPC is. Linearization of the engine system is done by taylor expansion of low order, the reasoning behind this is that for such a fast system it is sufficient to know in which direction the states are headed. Other proposed methods are applying feedback linearization and then use MPC in a cascade arrangement. However this method is limited to low order systems.
Linearizing by using taylor expansion required models of the system in symbolic expression to perform linearizations for each sampling time. This approach differs from others where linearization was done analytically.
A difference in approach from the the sources in the literature is that the MPC controller is developed in Matlab and Simulink. They provide the capability to generate symbolic expressions for the linearization in each sampling time, which was necessary for this approach.

1.4 Outline

- **Introduction**

  In the introduction chapter the problem formulation, purpose and objectives is discussed. Also a summary of related research is presented in this chapter.

- **System Overview**

  This chapter contains a brief overview of the gasoline engine and proposed controller.

- **Modeling**

  This chapter contains the modeling process and the theory for the models that are used including the defined ordinary differential equations for the dynamics.

- **Control System**

  This chapter will briefly explain model predictive control and how it is developed in detail.

- **Results**

  Here the results of the MPC controller are presented.

- **Discussion**

  The results of the previous chapter are discussed here.

- **Conclusions**

  In the final chapter conclusions and future work are presented.

1.5 Methodology

At first a study of related research and related work is integrated in the thought process of this thesis. Mainly to give structure to the development of the MPC. Secondly a systematic work flow for the MPC is considered where initially only one control signal is considered, in this case the throttle angle. This control signal controls the air mass flow in to the cylinders. After successfully designing a MPC with this control signal and then validating the results the next step will be taken. The next step is to introduce another control signal, the VVT. After successfully extending the MPC with another control signal it will be tested in simulation and then documented.

2

System Overview

This chapter gives an overview of the the type of engine that is used for the development of the MPC controller.

2.1 Engine

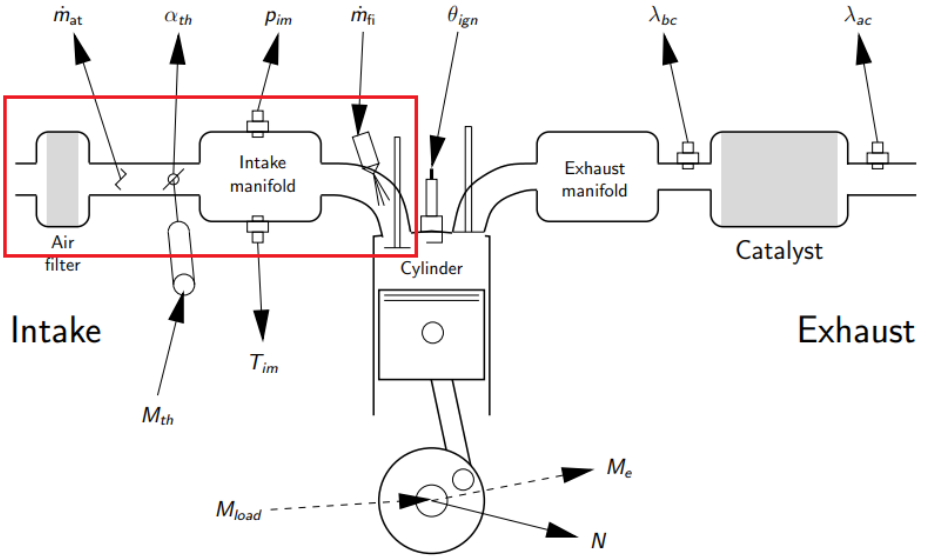

Figure 2.1: *An overview of some of the sensors and actuators that are used for controlling and diagnosing spark ignition engines. Reprinted and adjusted with permission from Eriksson and Nielsen (2014)*

.

The process to induct air into the cylinders, via the air inlet system and cylinder port valves, is referred to as the gas-exchange system. The process is marked in Figure 2.1. The MPC controller that is developed will control this part of the engine and, therefore models for the gas-exchange system are needed. Relevant parameters for this system will be presented in the Appendix.

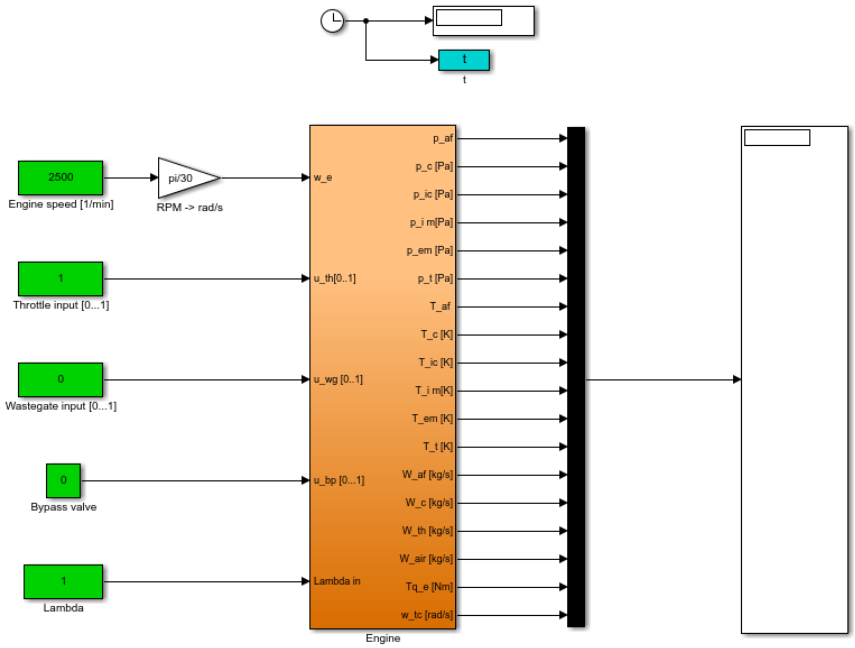

Figure 2.2: *MVEM in Simulink for the engine.*

The engine that is used for this master thesis is an SI-engine. There is a virtual engine available and is a mean value engine model (MVEM). In Eriksson and Nielsen (2014) MVEMs are defined as models where the signals, parameters, and variables that are considered are averaged over one or several cycles. The one that is used in this master thesis can be seen in Figure 2.2.

2.2 Controller

The MPC controller's final version can be seen in Figure 2.3. The MPC controller will operate on the gas-exchange as marked in Figure 2.1. Further motivation and details on the structure of the MPC controller will be presented in Chapter 4.

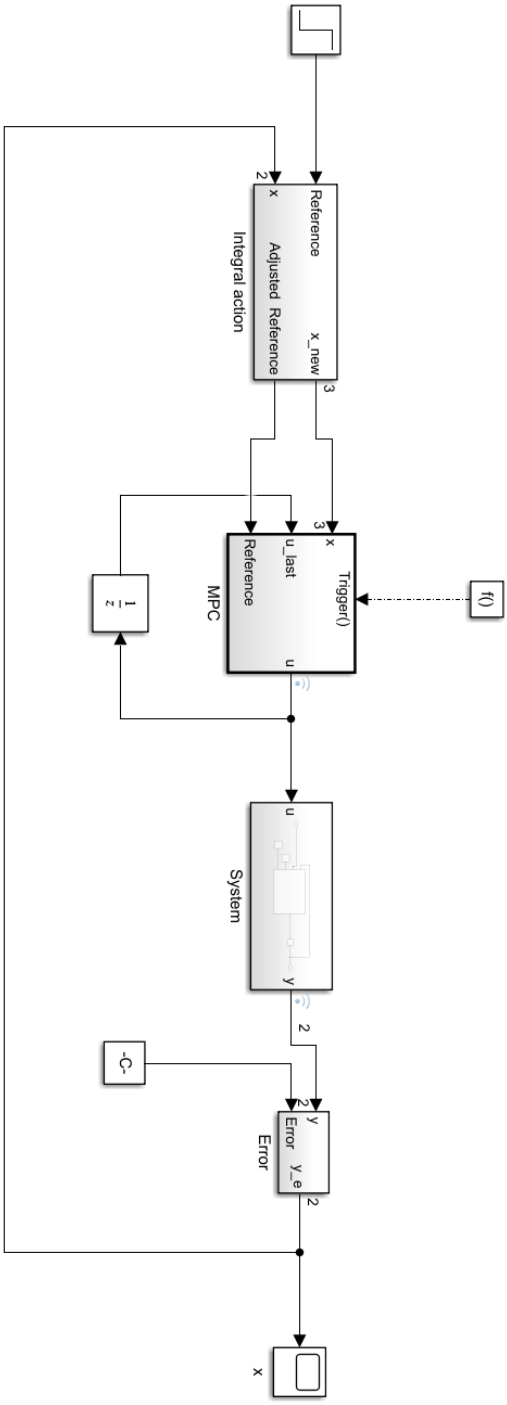

Figure 2.3: *An overview of the Simulink model of the MPC controller.*

3
Model

For MPC it is necessary to design models of the gas-exchange system in the engine. In this chapter the models for the throttle, intake manifold and VVT will be presented. These models will be used for different type of implementations of the MPC. The throttle model will be used in all implementations. However one MPC will use an isothermal control volume and another will use an adiabatic control volume. Lastly an implementation of MPC with a first order system for the VVT will be made.

3.1 Throttle

The throttle model is a central part of the system model that will be used by the MPC controller. Therefore data from the real throttle was collected and evaluated. The data was a series of step responses in the throttle in a lab environment and then the dynamics of the step responses were studied. To model the throttle the Ohata model used in Holmbom and Eriksson (2018) is used and formulated as suggested in Eriksson and Nielsen (2014).

3.1.1 Throttle Model

Flows that pass through an opening with a small cross-section area are well defined by isentropic compressible flow. Air mass flow past the throttle depends on the throttle angle, ambient pressure and temperatures as described in Eq. (3.1).

$$\dot{m}_{\text{at}} = \frac{p_{\text{us}}}{\sqrt{RT_{\text{us}}}} A_{\text{eff}}(\alpha_{\text{th}}) \Psi_{\text{cv}}(\Pi) \tag{3.1}$$

Where p_{us} and T_{us} are assumed to be p_{amb} and T_{amb} respectively. For simple

geometry, the effective cross-section area of the throttle can be described by a second order polynomial, as seen Eq. (3.2).

$$A_{\text{eff}}(\alpha_{\text{th}}) = a_0 + a_1\alpha_{\text{th}} + a_2\alpha_{\text{th}}^2 \tag{3.2}$$

Where α_{th} is the throttle plate position and a_0, a_1 and a_2 are estimated parameters which depend on the system geometry. The flow model suggested in Eriksson and Nielsen (2014) is replaced by the Ohata model which is proven to be a better choice. According to Holmbom and Eriksson (2018) Ohata's model using conservation of energy, mass and momentum, coincides with other models that use more parameters when estimating the Ψ-function.

$$\Psi_{\text{cv}}(\Pi) = \sqrt{\frac{\gamma+1}{2\gamma}(1-\Pi)(\Pi+\frac{\gamma-1}{\gamma+1})} \tag{3.3}$$

$$\Pi = \begin{cases} \frac{p_{\text{ds}}}{p_{\text{us}}}, & \text{if } \frac{p_{\text{ds}}}{p_{us}} \geq \frac{1}{\gamma+1} \\ \frac{1}{\gamma+1}, & \text{otherwise} \end{cases} \tag{3.4}$$

Where γ is the ratio of specific heats.
While using this Ψ-function several cases need to be kept in mind.

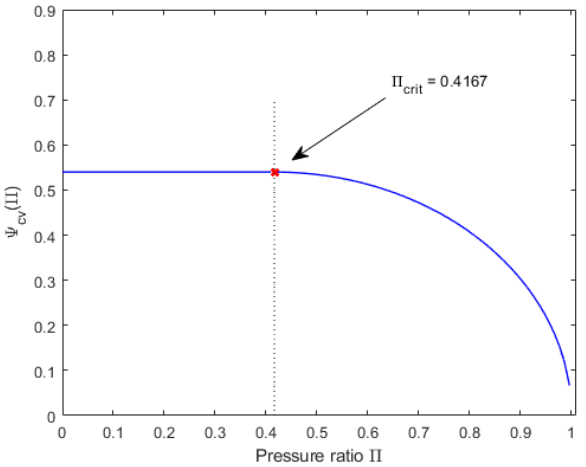

Figure 3.1: *The Ψ-function in Eq.(3.3) against the pressure ratio, Π . The critical pressure ratio can be seen in red.*

Figure 3.1 is inspired by the same type of plot in Eriksson and Nielsen (2014) and shows the behaviour of the flow as a function of pressure ratio. In the figure the cases in Eq.(3.4) can be seen. Where Π_{crit} is determined by the choice of γ. The critical pressure ratio occurs when the flow goes from sonic velocity to sub-sonic velocity which is indicated by the dotted line. Thus, the flow can be divided in two cases depending on their velocity, however there is a third case

where $\Pi = 1$ and $\Psi_{cv}(\Pi) = 0$ which is not acceptable since it doesn't fulfill the Lipschitz condition ,Eriksson and Nielsen (2014) and can be overcome by using a linear function around $\Pi = 1$, this region is seen in Figure 3.1 where the blue curve ends.
This case can be described by the following equation.

$$\Psi_{cv,li}(\Pi) = \Psi_{cv}(\Pi_{li})\frac{1-\Pi}{1-\Pi_{li}} \tag{3.5}$$

Where $\Pi_{li} = 0.99$ and the linear region is defined by $\frac{p_{ds}}{p_{us}} \in [\Pi_{li}, 1]$.
Data was collected from the real throttle and the data is presented in Figure 3.2 and 3.3. The data includes a series of steps taken in the throttle.

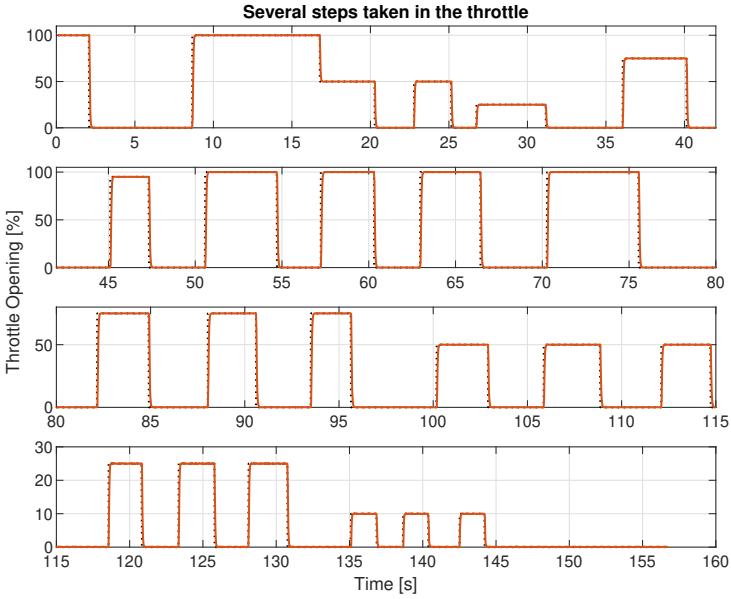

Figure 3.2: *A series of steps made in the real throttle.*

The real throttle has physical delays and therefore it's dynamics also need to be considered when modeling. Data from the real throttle was collected, four steps in the throttle can be seen in Figure 3.3 which when studied suggested that a first order system would suffice.

$$\frac{d\alpha_{th}}{dt} = \frac{1}{\tau_{th}}(\alpha_{th,ref} - \alpha_{th}) \tag{3.6}$$

Where $\alpha_{th,ref}$ is the requested throttle angle and τ_{th} is the time constant determined by the collected data. Higher order systems as seen in Figure 3.3 fit the measured data slightly better and pick up some of the dynamics a first order system would not, however the improvements are only marginally better than

the first order system in Eq.(3.6) and thus the systems of higher order were discarded since they require more computation.

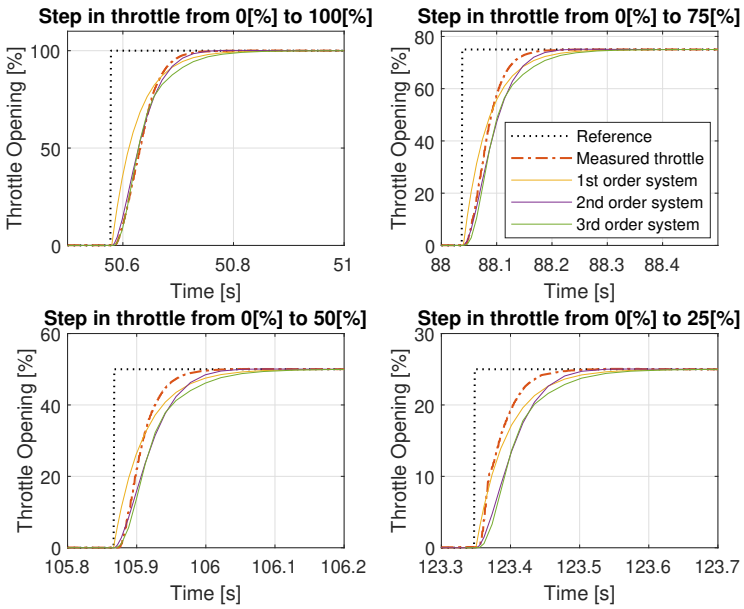

Figure 3.3: *Four steps were taken from the measured data for comparison against a first, second and third order system. The second and third order systems were estimated with Matlab's identification toolbox.*

3.2 Intake Manifold

The model for the intake manifold will describe some of the dynamic behaviour of the system. These are described with ordinary differential equations (ODE). The intake manifold can be viewed as a thermodynamic control volume that stores mass and energy, Eriksson and Nielsen (2014).
Consider that the intake manifold is a control volume with a fixed volume V. The increase and decrease of the mass is determined by $\dot{m}_{\text{in}}$ and $\dot{m}_{\text{out}}$, the mass and energy conservation will be used to derive the ODEs for the dynamics in the volume. The change of mass in the control volume can be described by

$$\frac{\mathrm{d}m}{\mathrm{d}t} = \dot{m}_{\text{in}} - \dot{m}_{\text{out}}. \tag{3.7}$$

Since mass and energy are difficult to measure, these equations are usually manipulated to express the system states in measurable quantities such as pressure and temperature. The intake manifold can be modeled as either isothermal or adiabatic.

3.2.1 Isothermal Model

The isothermal model does not consider the change in temperature and thus $T = T_{\text{in}} = T_{\text{out}}$. To derive the ODE for the pressure it is assumed that the gas is ideal and the ideal gas law is differentiated as seen in Eq.(3.8).

$$\frac{\mathrm{d}p}{\mathrm{d}t} = \frac{RT}{V}\frac{\mathrm{d}m}{\mathrm{d}t} \tag{3.8}$$

Inserting the mass conservation Eq.(3.7) in the differentiated ideal gas law, yields the following model for the intake manifold.

$$\frac{\mathrm{d}p_{\text{im}}}{\mathrm{d}t} = \frac{RT_{\text{amb}}}{V_{\text{im}}}(\dot{m}_{\text{at}} - \dot{m}_{\text{ac}}) \tag{3.9}$$

Where the intake manifold temperature is equal to the ambient temperature, $T_{\text{amb}} = T_{\text{im}}$ and $\dot{m}_{\text{ac}}$ is seen below.

$$\dot{m}_{\text{ac}}(N, p_{\text{im}}, T_{\text{amb}}) = \eta_{\text{vol}}\frac{V_{\text{D}}Np_{\text{im}}}{n_{\text{r}}RT_{\text{im}}} \tag{3.10}$$

Where η_{vol} is the volumetric efficiency.

3.2.2 Adiabatic Model

While working on this master thesis the question arose if an adiabatic model for the control volume would enhance the control performance. Therefore an adiabatic model is also considered. This model has a variable temperature as a state and the heat transfer $\dot{Q}$ is set to zero, Eriksson and Nielsen (2014).

$$\frac{dT_{\text{im}}}{dt} = \frac{RT}{p_{\text{im}}Vc_{\text{v}}}[\dot{m}_{\text{in}}c_{\text{v}}(T_{\text{in}} - T_{\text{im}}) + R(T_{\text{in}}\dot{m}_{\text{in}} - T_{\text{im}}\dot{m}_{\text{out}}) - \dot{Q}] \tag{3.11}$$

Furthermore the pressure model is extended upon as seen in Eq.(3.12).

$$\frac{dp_{\text{im}}}{dt} = \frac{RT_{\text{im}}}{V_{\text{im}}}(\dot{m}_{\text{in}} - \dot{m}_{\text{out}}) + \frac{p_{\text{im}}}{T_{\text{im}}}\frac{dT_{\text{im}}}{dt} \tag{3.12}$$

The results of the comparison between the adiabatic and isothermal models will be presented in Chapter 5.

3.2.3 Comparison between the Adiabatic and Isothermal models

To evaluate the two models they are compared by making the same step, from 8 [kPa] to 3.5 [kPa] and back. For this comparison Eq.(3.6) - Eq.(3.12) were used. The comparison can be seen in Figure 3.4 the pressure dynamics is insignificantly faster and there is no big difference. The isothermal model's temperature is constant thus there are no temperature dynamics and for the adiabatic model the temperature drops and rises when the steps in pressure are made. At steady state there is no difference between the models, Eriksson and Nielsen (2014).

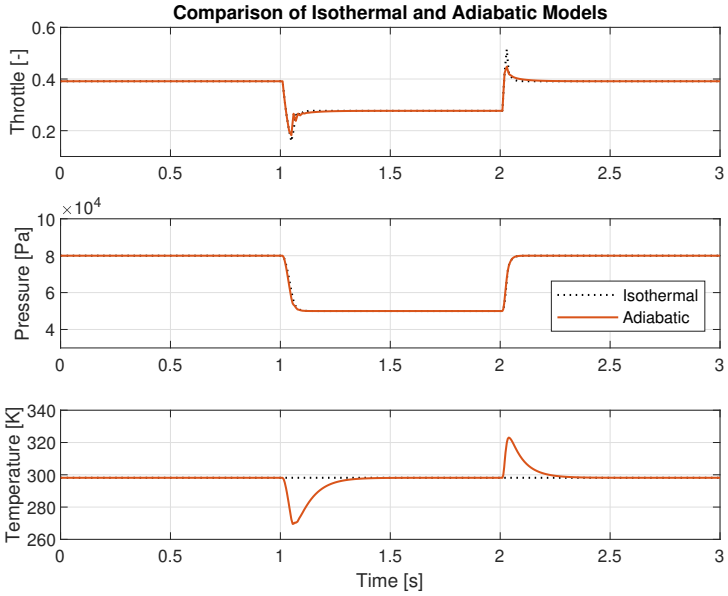

Figure 3.4: *Comparison between the isothermal and adiabatic model.*

However when looking at the requested control signals for the step responses in Figure 3.5, an interesting observation can be made. When the step is made at 1 [s] for both models the throttle is initially requested to be closed but when pressure converges to 4.5 [kPa] the adiabatic model does not request a full signal strength unlike the isothermal model which requests an fully open throttle for a short instance. This is most probably because of the temperature drop as seen in Figure 3.4.

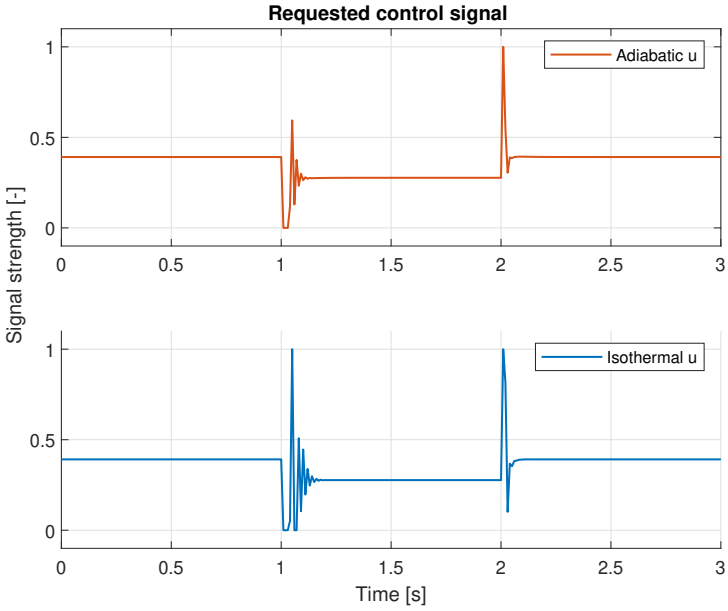

Figure 3.5: *MPC comparison between the isothermal and adiabatic model's requested control signals. Signal strength implies fully closed to fully opened throttle.*

Since the isothermal model has one less state than the adiabatic model, the former is preferred for this system.

3.3 Variable Valve Timing

Variable valve timing (VVT) is the ability to vary the timings of the valve openings and profiles. This can influence the combustion and torque generation of an engine. By doing so the potential of choosing the correct conditions for the gas-exchange system can be achieved resulting in a more efficient operating point which potentially could optimize fuel economy, emissions and maximum torque.

3.3.1 VVT Model

A complete model for VVT would mean a better understanding of the dynamics of this system and given an operating point, optimizing the η_{vol} value for that specific operating point.
When the MPC controller is fully realized a simplified model for the VVT can be tested which is modeled as a first order system and can be seen in Eq.(3.13).

$$\frac{d\alpha_\eta}{dt} = \frac{1}{\tau_\eta}(\alpha_{\eta,ref} - \alpha_\eta) \tag{3.13}$$

Where α_η is the requested volumetric efficiency and τ_η is the time constant for the dynamics of the VVT. For the simplified VVT model, τ_η is designed to be slower than τ_α by a factor of five.

3.4 Dynamic State Models

A summary of the ODE formulations of flows, temperatures and pressures in the air path.
ODE formulations:

$$x = \begin{bmatrix} \alpha_{th} \\ p_{im} \\ \alpha_\eta \end{bmatrix} \tag{3.14}$$

$$\dot{x} = f_i(x, u) \tag{3.15}$$

$$f_1(x, u) = \frac{u_1 - x_1}{\tau_{th}} \tag{3.16}$$

$$f_2(x, u) = \frac{RT_{\text{amb}}}{V_{\text{im}}}(\dot{m}_{\text{at}}(p_{\text{amb}}, x_2, T_{\text{amb}}, x_1) - \dot{m}_{\text{ac}}(N, x_2, T_{\text{amb}})) \tag{3.17}$$

$$f_3(x, u) = \frac{u_2 - x_2}{\tau_\eta} \tag{3.18}$$

$$\dot{m}_{\text{at}}(p_{\text{amb}}, x_2, T_{\text{amb}}, x_1) = \frac{p_{\text{amb}}}{\sqrt{RT_{\text{amb}}}} A_{\text{eff}}(x_1)\Psi_{\text{cv}}(\Pi(\frac{x_2}{p_{\text{amb}}})) \tag{3.19}$$

$$\dot{m}_{\text{ac}}(N, x_2, T_{\text{amb}}) = x_3 \frac{V_{\text{D}} N x_2}{n_{\text{r}} RT_{\text{amb}}} \tag{3.20}$$

$$A_{\text{eff}}(\alpha_{\text{th}}) = a_0 + a_1 x_1 + a_2 x_1^2 \tag{3.21}$$

$$\Psi_{\text{cv}}(\Pi) = \sqrt{\frac{\gamma + 1}{2\gamma}(1 - \Pi)(\Pi + \frac{\gamma - 1}{\gamma + 1})} \tag{3.22}$$

$$\Pi = \begin{cases} \frac{x_2}{p_{\text{amb}}}, & \text{if } \frac{x_2}{p_{\text{amb}}} \geq \frac{1}{\gamma+1} \\ \frac{1}{\gamma+1}, & \text{otherwise} \end{cases} \tag{3.23}$$

$$\Psi_{\text{cv,li}}(\Pi) = \Psi_{\text{cv}}(\Pi_{\text{li}})\frac{1 - \Pi}{1 - \Pi_{\text{li}}} \tag{3.24}$$

4

Control System

The main focus of this chapter will be on model predictive control (MPC). This chapter will include a brief explanation of MPC and how it works in general. In this chapter several implementations of MPC will be presented

- MPC with $x = [\alpha_{\text{th}}, p_{\text{im}}, T_{\text{im}}]^T$ and $u = \alpha_{\text{th,ref}}$
- MPC with $x = [\alpha_{\text{th}}, p_{\text{im}}, I_{\text{e}}]^T$ and $u = \alpha_{\text{th,ref}}$
- MPC with $x = [\alpha_{\text{th}}, p_{\text{im}}, \alpha_{\eta}]^T$ and $u = [\alpha_{\text{th,ref}}, \alpha_{\eta}]^T$

The MPC will be compared with a PID controller which is the solution used in conventional vehicles. Therefore a section for PID will be included in this chapter as well.

4.1 Model Predictive Control

There are several types of MPC, however for this master thesis a linear MPC will be developed. The controller's purpose is to control the actuators in the gas-exchange system. A MPC uses a model of the system which is to be controlled to predict the future behaviour of said system given a certain input. The MPC solves an optimal control problem where a cost-function is minimized over a time horizon. Then, an optimal control signal sequence is obtained for that time horizon. The first control signal in the sequence is used for controlling the system and the rest of the proposed control signals are discarded. Then this process is repeated for every time step. The advantage of model predictive control over other model based strategies is that it can explicitly take in to account constraints and limitations, Martin Enqvist (2014).
In Figure 4.1 the general structure of a MPC can be seen. In the first step the state vector x will be the in-put for the state model where it is also linearized.

In the second step the model is descretized and in the third step the MPC controller will be formulated according to quadratic programming and lastly the QP-solver runs and solves the cost-function which produces the optimal control signal sequence.

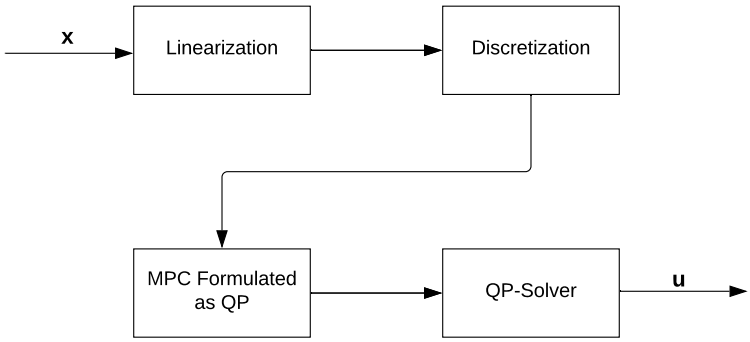

Figure 4.1: *The structure of the MPC controller.*

4.1.1 Definition of Control and Feedback signals for the multi-variable MPC controller

The state model consists of,

$$x = \begin{bmatrix} \alpha_{\text{th}} \\ p_{\text{im}} \end{bmatrix} \tag{4.1}$$

where α_{th} is the actual throttle angle and p_{im} is the intake manifold pressure. For defining the control and feedback signals for the MPC, the ODE formulations presented in Chapter 3 will be rewritten such that

$$\begin{cases} \dot{x}_1 &= \frac{1}{\tau_{\text{th}}}(u - x_1) \\ \dot{x}_2 &= \frac{RT_{\text{amb}}}{V_{\text{im}}}(\dot{m}_{\text{in}} - \dot{m}_{\text{out}}) \\ u &= \alpha_{\text{th,ref}} \end{cases} \tag{4.2}$$

Where $\alpha_{\text{th,ref}}$ is the desired throttle angle. The state model consists of the first order system for the throttle and the ODE for the intake manifold pressure. Below are the models for the air mass flow in and out of the intake manifold.

$$\dot{m}_{\text{in}} = \dot{m}_{\text{at}}(p_{\text{amb}}, x_2, T_{\text{amb}}, x_1) = \frac{p_{\text{amb}}}{\sqrt{RT_{\text{amb}}}} A_{\text{eff}}(x_1) \Psi_{\text{cv}}(\Pi(\frac{x_2}{p_{\text{amb}}})) \tag{4.3}$$

$$\dot{m}_{\text{out}} = \dot{m}_{\text{ac}}(N, x_2, T_{\text{amb}}) = \eta_{\text{vol}} \frac{V_{\text{D}} N x_2}{n_{\text{r}} R T_{\text{im}}} \tag{4.4}$$

The next step for the state model in Eq. (4.2) is to linearize it.

4.1.2 Control and Feedback signals for Adiabatic model

Implementing the adiabatic model for the intake model requires for an extended state model which consists of,

$$x = \begin{bmatrix} \alpha_{\text{th}} \\ p_{\text{im}} \\ T_{\text{im}} \end{bmatrix} \tag{4.5}$$

The reasoning of the inclusion of an adiabatic control volume model is for performance comparison and demonstration of constraint usage in MPC.

4.1.3 Model Linearization

For this master thesis a linear MPC controller is considered, therefore it is necessary to linearize the dynamic state models presented in Chapter 3. The linearized state model will be in the following form,

$$\dot{x} = Ax + Bu + f_0 \tag{4.6}$$
$$y = Cx \tag{4.7}$$

Considering a non-linear system,

$$\dot{x} = f(x, u)$$
$$y = h(x)$$

it can be linearized with Taylor expansion. Assuming that,

$$x = \Delta x + x_0$$
$$u = \Delta u + u_0$$

By Taylor expanding the non-linear system above,

$$\dot{x} \approx \Delta\dot{x} = f(x_0 + \Delta x, u_0 + \Delta u) \approx f(x_0, u_0) + \frac{\partial f(x_0, u_0)}{\partial x}\Delta x + \frac{\partial f(x_0, u_0)}{\partial u}\Delta u \tag{4.8}$$

is obtained. The assumption is made that the state vector x is known to the controller thus $y = Cx$ is neglected. With this Taylor expansion the matrices for Eq.(4.6) can be constructed as,

$$A = \begin{pmatrix} \frac{\partial f_{\text{i}}}{\partial x_{\text{i}}} & \cdots & \frac{\partial f_{\text{i}}}{\partial x_{\text{n}}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{\text{n}}}{\partial x_{\text{i}}} & \cdots & \frac{\partial f_{\text{n}}}{\partial x_{\text{n}}} \end{pmatrix} \quad , \quad B = \begin{pmatrix} \frac{\partial f_{\text{i}}}{\partial u} \\ \vdots \\ \frac{\partial f_{\text{n}}}{\partial u} \end{pmatrix}$$

and

$$f(x_0, u_0) = f_0$$

With the linearized state model the next step is to discretize it.

4.1.4 Model Discretization

For model predictive control the models always have to be in discrete time and will be presented as in Martin Enqvist (2014).

$$x(k+1) = Fx(k) + Gu(k) \tag{4.9a}$$
$$y(k) = Cx(k) \tag{4.9b}$$
$$z(k) = Mx(k) \tag{4.9c}$$

Where $z(k)$ is the quantity that is to be controlled. For this MPC controller the assumption will be made that the complete state vector $x(k)$ is available for the controller. This is only possible if all states can be measured exactly meaning that $C = I$ and $y(k) = x(k)$.
For discretization of the state model Euler's forward method is used.

$$x(k+1) = x(k) + T_s f(x(k), u(k)) \tag{4.10}$$

Applying method (4.10) on the state model in Eq.(4.6) results in

$$x(k+1) = x(k) + T_s(Ax(k) + Bu(k) + f_0) = [I + T_s A]x(k) + T_s Bu(k) + T_s f_0 \tag{4.11}$$

and the discrete model will be

$$x(k+1) = Fx(k) + Gu(k) + K \tag{4.12}$$

in the same form as Eq.(4.9a). Using this form recursively the two-step prediction becomes

$$\begin{aligned} x(k+2) &= Fx(k+1) + Gu(k+1) + K \\ &= F^2x(k) + FGu(k) + Gu(k+1) + FK + K \end{aligned} \tag{4.13}$$

The step prediction continues similarly until N which is the receding time horizon. For simplification of the notation two vectors containing future control signals and their respective states will be used.

$$U = \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+N-1) \end{bmatrix}, \quad X = \begin{bmatrix} x(k) \\ x(k+1) \\ \vdots \\ x(k+N-1) \end{bmatrix}$$

The discrete state model Eq.(4.12) for receding time horizon N can now be written as

$$X = \mathcal{F}x(k) + \mathcal{G}U + \mathcal{D} \tag{4.14}$$

where

$$\mathcal{F} = \begin{bmatrix} I \\ F \\ \vdots \\ F^{N-1} \end{bmatrix} \quad , \quad \mathcal{G} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ G & 0 & 0 & \cdots & 0 \\ FG & G & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ F^{N-2}G & \cdots & FG & G & 0 \end{bmatrix}$$

and

$$\mathcal{D} = \begin{bmatrix} 0 \\ IK \\ FK + IK \\ F^2K + FK + IK \\ \vdots \\ F^{N-2}K + \cdots + F^2K + FK + IK \end{bmatrix}$$

4.1.5 Cost-function and QP-problem

This section will serve as a in depth look on the optimal control problem (OCP) and propose a cost-function and lastly formulate the OCP as a QP-problem.

OCP

The reason a MPC is proposed for this type of application is because it can handle multiple in-puts and out-puts well. The objective is to control the gas-exchange process and ideally control it more efficiently. A car is driven by pushing the gas pedal which translates to a throttle angle which translates to a requested amount of torque. After a brief delay the car will accelerate and converge to the requested amount of torque. To control this, Eq. (4.15) is considered.

$$M_\mathrm{e} = \frac{W_\mathrm{e}}{n_\mathrm{r} 2\pi} = \frac{W_\mathrm{i,g} - W_\mathrm{i,p} - W_\mathrm{fr}}{n_\mathrm{r} 2\pi} \tag{4.15}$$

$W_\mathrm{i,g}$, $W_\mathrm{i,p}$ and W_fr are gross indicated work, pumping work and engine friction respectively. In Eq.(4.15) $-W_\mathrm{i,p}$ and $-W_\mathrm{fr}$ indicate losses. The engine friction is unchangeable and the pumping work losses depend indirectly on the air mass flow as seen in Eq.(4.4) which requires a degree of freedom in p_im. This can be achievable with VVT control which affects the air mass flow to the cylinders. Since this is not the case the torque is considered as a cost-function as seen in Eq.(4.16) which has a linear dependency on the intake manifold pressure p_im and will be presented below.

$$\min_{u} \int_{t_0}^{t_1} (M(t,u)_{\mathrm{e,ref}} - M(t,u)_{\mathrm{e}})^2 dt \tag{4.16}$$

The cost-function in the optimal control problem (OCP) in Eq.(4.16) is a difference in torque and requested torque. Minimizing this difference results in going from a certain torque to a reference torque.
For an SI-engine a simple model is obtained by Eriksson and Nielsen (2014), where the engine operates on nominal conditions and depends linearly on the intake manifold pressure p_{im} as seen in Eq.(4.17) and Eq.(4.18).

$$M_{\mathrm{e}} = \frac{BMEP(p_{\mathrm{im}})V_{\mathrm{D}}}{n_{\mathrm{r}} 2\pi} \tag{4.17}$$

$$BMEP(p_{\mathrm{im}}) = -C_{\mathrm{p1}} + C_{\mathrm{p2}} p_{\mathrm{im}} \tag{4.18}$$

With Eq.(4.17) and Eq.(4.18) the torque can be formulated as

$$M_{\mathrm{e}} = \frac{V_{\mathrm{D}}(-C_{\mathrm{p1}} + C_{\mathrm{p2}} p_{\mathrm{im}})}{n_{\mathrm{r}} 2\pi} \tag{4.19}$$

This is a simplified model of the torque and because of the linear behaviour the cost-function can be formulated in continuous time as

$$\min_{u} \int_{t_0}^{t_1} (p(t,u)_{\mathrm{im,ref}} - p(t,u)_{\mathrm{im}})^2 dt. \tag{4.20}$$

Future Work

For improving the fuel economy in a future iteration of the MPC a possible cost-function will be proposed in this section. The work in Eq.(4.15) can be rewritten to and simplified as

$$W_{\mathrm{e}} = m_{\mathrm{f}} q_{\mathrm{LHV}} \eta_{\mathrm{ig}} - V_{\mathrm{D}}(p_{\mathrm{em}} - p_{\mathrm{im}}) - V_{\mathrm{D}} FMEP \tag{4.21}$$

Where q_{LHV} and η_{ig} are positive constants and

$$m_{\mathrm{f}} = \frac{m_{\mathrm{a}}}{(A/F)_{\mathrm{s}} \lambda} \tag{4.22}$$

Implementing Eq.(4.22) in Eq.(4.23) gives

$$W_{\mathrm{e}} = \frac{m_{\mathrm{a}}}{(A/F)_{\mathrm{s}} \lambda} q_{\mathrm{LHV}} \eta_{\mathrm{ig}} - V_{\mathrm{D}}(p_{\mathrm{em}} - p_{\mathrm{im}}) - V_{\mathrm{D}} FMEP. \tag{4.23}$$

To improve the total work W_e it is possible to minimize the losses from the pumping work. The engine friction is fixed and thus unchangeable. The proposed cost-function would be

$$\min_u \int_{t_0}^{t_1} (m(t,u)_{\mathrm{a,ref}} - m(t,u)_{\mathrm{a}})^2 - p_{im}^2 dt. \tag{4.24}$$

Where the objective is to achieve the mass air reference to the cylinders with as high intake manifold pressure as possible. This would result in reaching the desired torque but with the pumping work losses minimized.

Solving the OCP

Before taking the next step towards the MPC controller which is developed in discrete time, the feasibility of the (OCP) is validated in continuous time. This is done for proof of concept purposes and to evaluate the OCP and its in-put and out-put signals. Comparing the results of the numerical optimal control problem against the MPC is a way to validate the feasibility of the problem and study the differences of the problem in continuous time against discrete time.

YOP

YOP is a MATLAB Toolbox for numerical optimal control. It utilizes CasADi which is an open-source tool for nonlinear optimization, to interface to integrators and non-linear optimization solvers, Leek (2016).

$$\min_u \int_{t_0}^{t_1} (p(t,u)_{\mathrm{im,ref}} - p(t,u)_{\mathrm{im}})^2 dt \tag{4.25}$$
$$0.01 \leq u \leq 0.99$$

The OCP was basically constructed as a MPC problem, the only significant difference is that this toolbox only solves the problem once. The main reason for this test is to confirm that the cost-function is solvable and investigate the input signal's behaviour. The cost-function as seen in Eq. (4.25) was used and the state model was implemented with all sub-models Eq.(4.26).

$$\begin{cases} \dot{x}_1 = \frac{d\alpha_{\mathrm{th}}}{dt} = \frac{1}{\tau_{\mathrm{th}}}(\alpha_{\mathrm{th,ref}} - \alpha_{\mathrm{th}}) \\ \dot{x}_2 = \frac{\mathrm{d}p_{\mathrm{im}}}{\mathrm{d}t} = \frac{RT_{\mathrm{a}}}{V_{\mathrm{im}}}(\dot{m}_{\mathrm{at}} - \dot{m}_{\mathrm{ac}}) \\ u = \alpha_{\mathrm{th,ref}} \end{cases} \tag{4.26}$$

With this formulation a couple of step responses were done with $\Pi = \frac{p_{\mathrm{im}}}{p_{\mathrm{amb}}}$ and different initial values. The steps compared against a MPC can be seen in Chapter 5.

4.1.6 QP formulation of the cost-function

Model predictive control is one of few control strategies that has become popular in modern day. MPC has basically become the standard for some industries because of its capability to handle constraints well in the control process, Martin Enqvist (2014). It's popularity also originates in the fact that many consider it to be easy to understand and use.
The basic idea in MPC is to formulate the control problem as an optimization problem instead. The optimization problem will be solved which means minimizing a cost-function online, for each time new output signals are received.
The expression (4.27) as presented in Martin Enqvist (2014) is how the optimization problem could be formulated.

$$J_N(x(k)) = \sum_{j=0}^{N-1} \|z(k+j)\|^2_{Q_1} + \|u(k+j)\|^2_{Q_2} \tag{4.27}$$

The expression is in discrete time k and N is the receding time horizon. $x(k)$ is the state, $z(k)$ is the control quantity to be controlled and $u(k)$ is the control signal.
Basically the model is used to predict the future output behaviour for a particular input signal, where the difference between predictions and reference is minimized over a future horizon (N). The result of the optimization is applied according to the receding time horizon, at time step k, only the first control signal of the optimal control sequence is applied and the rest is discarded. Then the same process is repeated at time step $k + j$.
To formulate the cost-function Eq.(4.27) as a quadratic programming problem. The quadratic forms in the cost-function are expanded and the expression in Eq.(4.28) is obtained.

$$\sum_{j=0}^{N-1} \|z(k+j)\|^2_{Q_1} + \|\Delta u(k+j)\|^2_{Q_2} =$$
$$(\mathcal{F}x(k) + \mathcal{G}U)^T \mathcal{M}^T \mathcal{Q}_1 \mathcal{M}(\mathcal{F}x(k) + \mathcal{G}U) + U^T \mathcal{Q}_2 U \tag{4.28}$$

Where

$$\mathcal{Q}_1 = \begin{bmatrix} Q_1 & & & \\ & Q_1 & & \\ & & \ddots & \\ & & & Q_1 \end{bmatrix}, \quad \mathcal{Q}_2 = \begin{bmatrix} Q_2 & & & \\ & Q_2 & & \\ & & \ddots & \\ & & & Q_2 \end{bmatrix}$$

and

$$\mathcal{M} = \begin{bmatrix} M & & & \\ & M & & \\ & & \ddots & \\ & & & M \end{bmatrix}$$

These are block diagonal matrices of Q_1, Q_2 and M which are repeated N times. The penalty matrix Q_1 for the states dictates how much the states are punished in the cost-function. Penalty matrix Q_2 acts similarly for the control signals. Lastly N is the receding time horizon, which basically is the number of steps in the future that the MPC will predict.
The reason MPC is widely accepted is that the optimal control problem can be solved with quadratic programming, Martin Enqvist (2014). In quadratic programming the constraints are linear but the cost-function is quadratic, a generic formulation can be seen below.

$$\begin{aligned} \min_{w} \quad & \tfrac{1}{2} w^{\mathrm{T}} H w + f^{\mathrm{T}} w \\ \text{s.t.} \quad & A w \leq b \end{aligned} \tag{4.29}$$

Eq.(4.35) is the MPC problem as seen in Eq.(4.28) but formulated as QP problem according to Eq.(4.29).

$$\min_{U} \quad \tfrac{1}{2} U^{\mathrm{T}} (\underbrace{\mathcal{G}^{\mathrm{T}} \mathcal{M}^{\mathrm{T}} \mathcal{Q}_1 \mathcal{M} \mathcal{G} + \mathcal{Q}_2}_{\mathrm{H}}) U + \underbrace{(\mathcal{G}^{\mathrm{T}} \mathcal{M}^{\mathrm{T}} \mathcal{Q}_1 \mathcal{M} (\mathcal{F} x(k) + \mathcal{D}))^{\mathrm{T}}}_{\mathrm{f}} U \tag{4.30}$$

$$\text{s.t.} \quad A_{\mathrm{u}} U \leq b_{\mathrm{u}}$$

4.1.7 Reference tracking

In this section the QP problem is further expanded upon so it can follow a reference.

$$J_N(x(k)) = \sum_{j=0}^{N-1} \|z(k+j) - r(k+j)\|_{Q_1}^2 + \|\Delta u(k+j)\|_{Q_2}^2 \tag{4.31}$$

As can be seen in the cost-function in Eq.(4.31) the difference between the control quantity $z(k)$ and $r(k)$ is punished, this means that the cost-function will minimize the difference as fast as possible, thus reaching the reference point.

$$\min_{U} \quad \tfrac{1}{2} U^{\mathrm{T}} (\underbrace{\mathcal{G}^{\mathrm{T}} \mathcal{M}^{\mathrm{T}} \mathcal{Q}_1 \mathcal{M} \mathcal{G} + \mathcal{Q}_2}_{\mathrm{H}}) U + \underbrace{(\mathcal{G}^{\mathrm{T}} \mathcal{M}^{\mathrm{T}} \mathcal{Q}_1 \mathcal{M} (\mathcal{F} x(k) + \mathcal{D}) - \mathcal{R})^{\mathrm{T}}}_{\mathrm{f}} U \tag{4.32}$$

Where

$$\mathcal{R} = \begin{bmatrix} r(k)) \\ r(k+1) \\ \vdots \\ r(k+N-1) \end{bmatrix}$$

The results of programming the past sections in MATLAB and building them in Simulink can be seen in Figure 4.2.

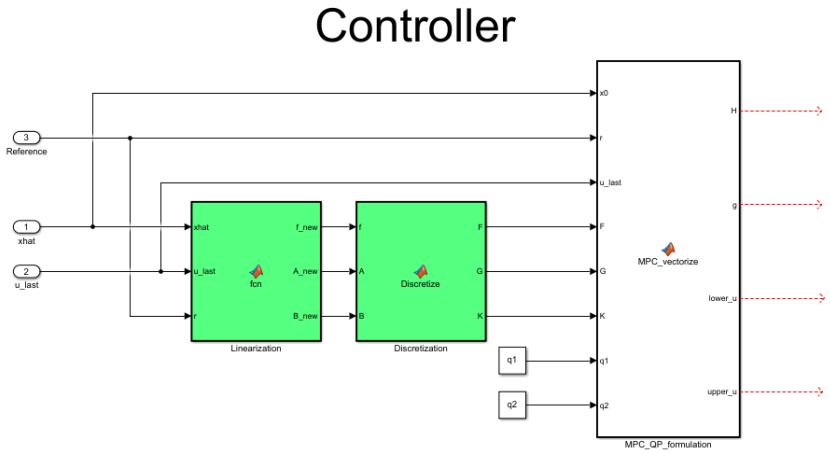

Figure 4.2: *Showcasing the Simulink model for the controller. The Simulink blocks seen contain the programming for the linearization, discretization and QP formulation in symbolic expression.*

The white block in Figure 4.2 outputs vectors in the form as seen in Eq.(4.29). Where $H = H$, $g = f^T$ and $loweru$ and $upperu$ are the lower and upper constraints for the control signals. These outputs are then fed to the QP-solver.

4.1.8 Integral Action

In the simulation environment, the MPC controller performs without stationary error. However this would not be the case in a real environment because of noise in estimations and other disturbances, therefore a stationary error was implemented in the system. To minimize the stationary error, integral action was also implemented. This can be implemented as suggested in Martin Enqvist (2014), however this method requires very precise models and thus was not feasible.
The method used instead is by extending the state model with a state for the error as seen in Eq.(4.33).

$$x = \begin{bmatrix} \alpha_{\text{th}} \\ p_{\text{im}} \\ I_{\text{e}} \end{bmatrix} \tag{4.33}$$

This state contains the integral of the stationary error e. Since $C = I$ and $y = x$ the state will be

$$I_{\text{e}} = \int (e)\text{dt} = \int (r - y)\text{dt} = \int (r - x)\text{dt}.$$

The state model in (4.2) will be expanded upon with

$$\dot{x}_3 = \dot{I}_{\text{e}} = r - x_2 \tag{4.34}$$

The integral action is implemented in the simulation environment as seen in Figure 4.3.

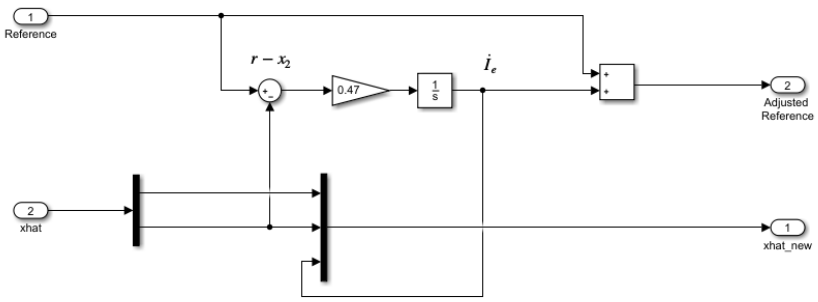

Figure 4.3: *The integral action as modeled in Simulink. The error $r - x_2$ is integrated and then $\dot{I}_e$ is added to the new state vector.*

4.1.9 QP-Solver

In this master thesis qpOASES which is a QP-solver is used to solve Eq.(4.29). qpOASES is an open-source C++ implementation of a QP-solver, Ferreau et al. (2019). qpOASES is chosen for this thesis because it is developed with model predictive control in mind and also offers interfaces for MATLAB and Simulink. Lastly it can generate code for further implementation. Since the MPC controller in this thesis is built in MATLAB and Simulink, qpOASES is an easy to use QP-solver with an interface as seen in Figure 4.4

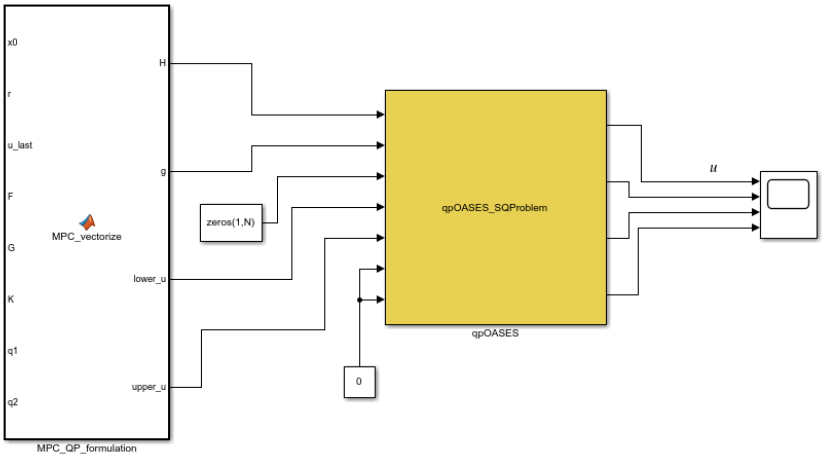

Figure 4.4: *The MPC controller produces the QP matrices H and g which are fed to qpOASES and solved. The QP-solver then outputs the optimal control signal u.*

4.1.10 Constraints

One of the advantages of MPC is the ability to limit the control signals by programming constraints as seen in Eq.(4.29). In the same fashion this can also be extended for limiting the states.

$$\min_{U} \quad \frac{1}{2} U^{\mathrm{T}} \underbrace{(\mathcal{G}^{\mathrm{T}} \mathcal{M}^{\mathrm{T}} \mathcal{Q}_1 \mathcal{M} \mathcal{G} + \mathcal{Q}_2)}_{\mathrm{H}} U + \underbrace{(\mathcal{G}^{\mathrm{T}} \mathcal{M}^{\mathrm{T}} \mathcal{Q}_1 \mathcal{M} (\mathcal{F} x(k) + \mathcal{D}))^{\mathrm{T}}}_{\mathrm{f}} U \tag{4.35}$$

$$\text{s.t.} \quad \begin{bmatrix} A_{\mathrm{u}} \\ A_{\mathrm{z}} \end{bmatrix} U \quad \leq \quad \begin{bmatrix} b_{\mathrm{u}} \\ b_{\mathrm{z}} \end{bmatrix}$$

This can be done in the programming of the MPC controller and can also be seen in Figure 4.4 where the upper and lower bounds for the control signal u are fed to the QP-solver. In the figure it is also seen that there no limitations on any of the states because zeros are fed to the QP-solver which turns these off. For the different MPC implementations in this master thesis the constraints that are of interest are presented below.

$$\begin{array}{rcrcl} 0 & \leq & u_1 & \leq & 1 \\ 0.4 & \leq & u_2 & \leq & 1 \\ & & T_{im} & \leq & 330[K] \\ & & T_{im} & \leq & 320[K] \end{array} \tag{4.36}$$

Where $u_1 = \alpha_{th}$ and $u_2 = \alpha_{\eta}$. The constraint on the control signal for the throttle translates to fully closed to fully opened throttle. In the volumetric efficiency

it is a requirement to have a lower bound that at least allows for ignition in the cylinders. The constraints on the intake manifold temperature are for demonstrating the implementation of bounds for states in MPC.

4.1.11 Penalty matrices Q

With the penalty matrices Q_1 and Q_2 you can determine how strict the cost-function and changes in the control signals will be penalized.
For all implementations below the penalty for $x_2 = p_{\text{im}}$ was chosen to be high since they all use the same cost-function where the objective is to track the reference pressure.

- MPC with $x = [\alpha_{\text{th}}, p_{\text{im}}, T_{\text{im}}]^T$ and $u = \alpha_{\text{th,ref}}$
  Where $x_1 = \alpha_{\text{th}}$ and $x_3 = T_{\text{im}}$ are not penalized in Q_1 and with a very low penalty on the control signal u in Q_2.
- MPC with $x = [\alpha_{\text{th}}, p_{\text{im}}, I_{\text{e}}]^T$ and $u = \alpha_{\text{th,ref}}$
  Where $x_1 = \alpha_{\text{th}}$ and $x_3 = I_{\text{e}}$ are not penalized in Q_1 and with a very low penalty on the control signal u in Q_2.
- MPC with $x = [\alpha_{\text{th}}, p_{\text{im}}, \alpha_\eta]^T$ and $u = [\alpha_{\text{th,ref}}, \alpha_\eta]^T$
  Case 1: High penalty on $x_1 = \alpha_{\text{th}}$ and no penalty on $x_3 = \alpha_\eta$ in Q_1 and with a very high penalty on the control signal u_1 and no penalty on u_2 in Q_2.
  Case 2: No penalty on $x_1 = \alpha_{\text{th}}$ and high penalty on $x_3 = \alpha_\eta$ in Q_1 and with a very high penalty on the control signal u_2 and no penalty on u_1 in Q_2.

4.2 PID with λ-tuning

To compare the control performance of the MPC controller, a PID controller with the lambda tuning method was made in MATLAB and Simulink. There are several ways to set up the PID. In this master thesis it is done with the lambda tuning set-up and the method in Rivera (1999) was used.
The general expression for PID control in the S-domain is:

$$c(s) = K_{\text{c}}(1 + \frac{1}{\tau_{\text{I}}} + \tau_{\text{D}} s) \tag{4.37}$$

Where K_{c}, τ_{I} and τ_{D} are the gain, integration time and derivative time respectively. To apply the PID controller on the system the second order model in Rivera (1999) was chosen as presented below.

$$c(s) = \frac{K}{\tau^2 s^2 + 2\zeta\tau s + 1} \tag{4.38}$$

$$K_{\text{c}} = \frac{2\zeta\tau}{\lambda K}, \quad \tau_{\text{I}} = 2\zeta\tau, \quad \tau_{\text{D}} = \frac{\tau}{2\zeta}$$

By using MATLAB and transforming the state model in Eq.(4.2) to the frequency domain with Laplace transform the coefficients above were derived. The final PID controller was developed in Simulink and used for evaluation purposes in the next chapter.

5

Results

This chapter presents the results of the basic MPC controller, the MPC controller with an adiabatic model for the intake manifold, the MPC controller compared to a PID controller and lastly a multi-input-multi-output MPC controller.

5.1 The Basic MPC

This section presents the MPC in its most basic version, all other applications of MPC in this thesis are based on this version.
In the step response seen in Figure 5.1 a step in p_{im} is made from 20 [kPa] to 65 [kPa] with an implemented error of 5 [kPa] in the measurement of p_{im}. This error is implemented in the measurement of state x_2 which is fed back to the controller. There is an overshoot which is taken care of quickly by the integral action. In the step response seen in Figure 5.2 a step in p_{im} is made from 20 [kPa] to 65 [kPa] with no error implemented. There is no overshoot. Several step responses with the MPC with integral action can be found in the Appendix.

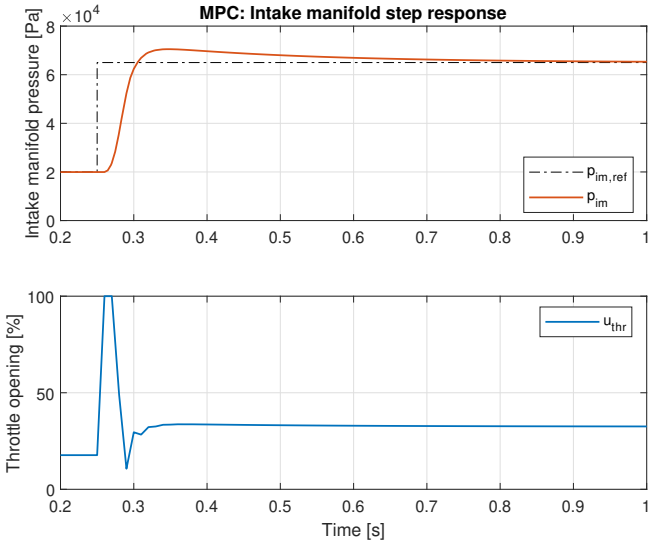

Figure 5.1: *A step response from 20 [kPa] to 65 [kPa] with a measurement error of 5 [kPa].*

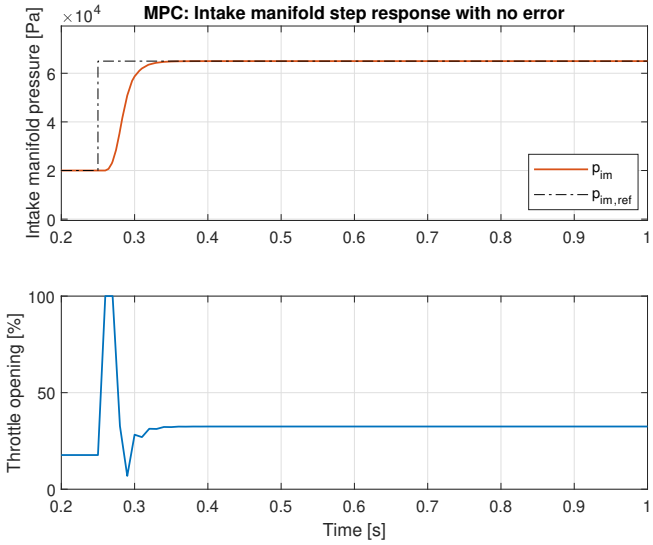

Figure 5.2: *A step response from 20 [kPa] to 65 [kPa] without a measurement error.*

5.2 YOP vs MPC

In Chapter 4 the optimal control problem defined as seen in Eq.(4.25) was solved with YOP. To validate the results of the MPC with no integral action they are compared against each other in this section. In Figure 5.3 a step in the intake manifold pressure is made from 20 [kPa] to 60 [kPa]. The MPC is faster in reaching the reference p_{im}. The difference can be more clearly seen in the upper plot where the angle α_{th} reacts faster and also does not open as much as in the YOP case. The requested control signals for both cases can be seen in Figure 5.4. A significant difference can be seen here, the MPC control signal is more exact, unlike the YOP control signal were it oscillates before converging. The reason the MPC signal is more refined is because it solves the QP-problem for each time step, meaning an optimal control sequence is retained where the MPC uses the first control signal and discards the others and then repeats this.
In Figure 5.5 a step from 35 [kPa] to 90 [kPa] is made. The same behaviour can be seen in this case as well. The pressure p_{im} in both cases are close to each other but differ a little. The MPC reacts faster than YOP to the step that is made. In Figure 5.6 the same behaviour can be observed by looking at the oscillating YOP control signal. Again, this is the result of the difference in how the MPC and YOP solve the QP-problem.

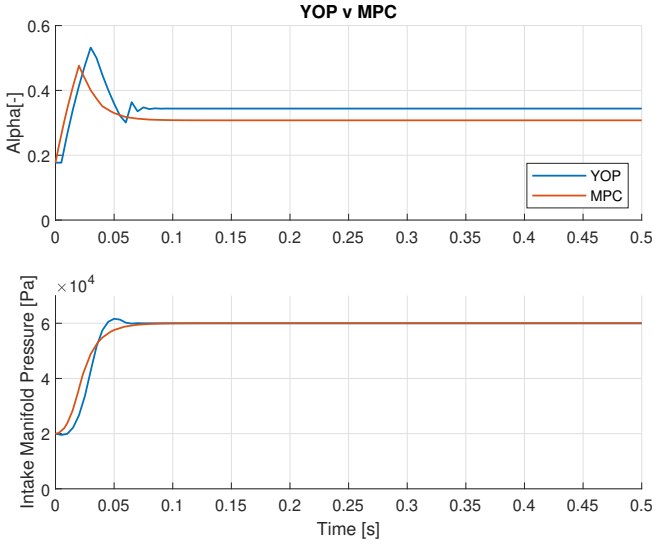

Figure 5.3: *A step response in YOP and the MPC. The step in the lower plot is made in the intake manifold pressure from 20 [kPa] to 60 [kPa]. In the upper plot the angle alpha for the throttle can be seen.*

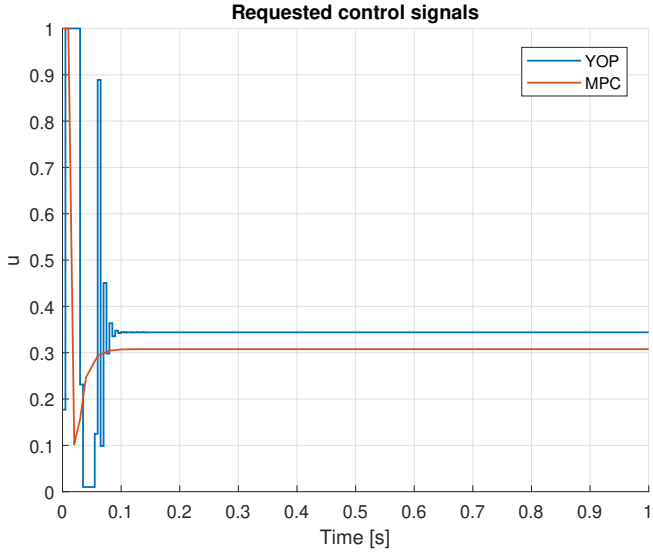

Figure 5.4: *The requested control signals for the step response in Figure 5.3*

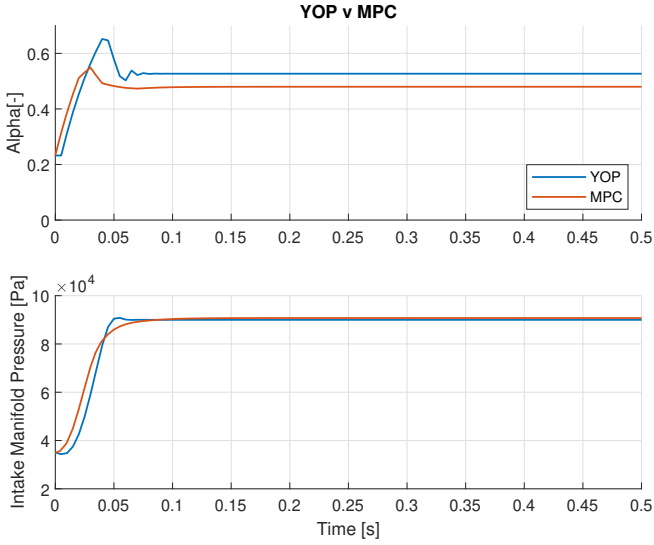

Figure 5.5: *A step response in YOP and the MPC. The step in the lower plot is made in the intake manifold pressure from 35 [kPa] to 90 [kPa]. In the upper plot the angle alpha for the throttle can be seen.*

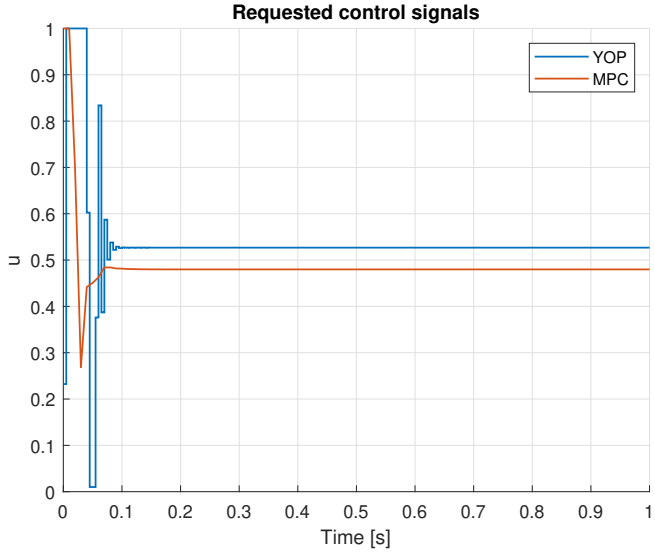

Figure 5.6: *The requested control signals for the step response in Figure 5.5*

5.3 MPC with adiabatic model

In Chapter 3 an adiabatic control volume for the intake manifold was proposed. An implementation of the MPC with the intake manifold temperature as seen in Eq.(3.11) was made to study the performance of such a MPC and also test the flexibility of the controller with constraints. At first a step in p_{im} was made with

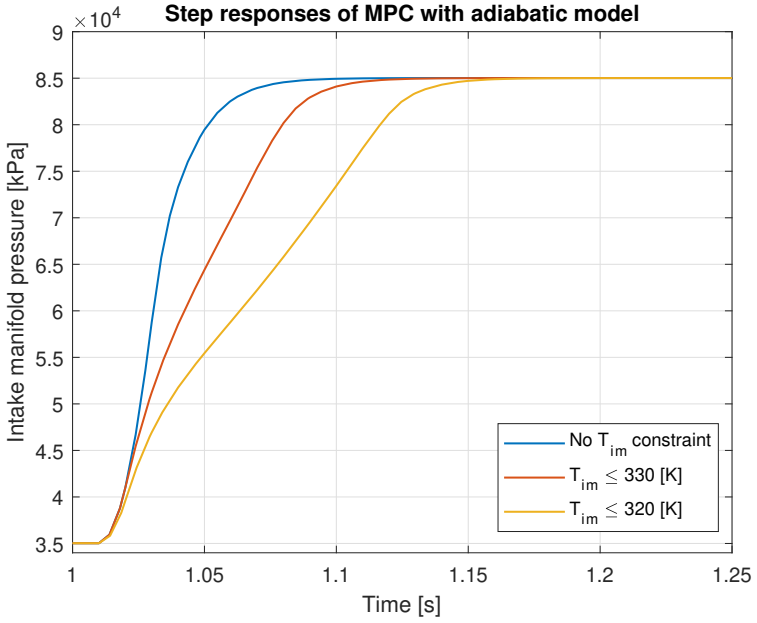

Figure 5.7: *A step in the intake manifold pressure from 35 [kPa] to 85 [kPa] with different constraints on the intake manifold temperature can be seen.*

no constraints on T_{im}. Then the peak temperature could be determined by looking at the temperature as seen in Figure 5.8 which reached a maximum temperature of 341$[K]$. To demonstrate the advantages of constraints in MPC, two more steps were made where T_{im} was limited to 330$[K]$ and 320$[K]$. The results of these tests can be seen in Figure 5.7. The step in p_{im} with no constraints on the T_{im} is clearly faster than the steps where T_{im} is constrained. There is an clear improvement in performance when the temperature in the intake manifold is not constrained. The reason the build up in the intake manifold pressure is slower in the constrained cases is because that will slow down the compression in the intake manifold. With a slower compression the temperature in the intake manifold can be lower which in turn is necessary to achieve the boundaries set on T_{im}.

These comparisons with the adiabatic control volume serve as a demonstration of how constraints can be implemented and handled. The way constraints on states are handled can be seen when comparing the steps. In Figure 5.7 the step with $T_{im} \leq 330[K]$ performs worse than the no constraint case. This is the con-

sequence of the control signal usage as seen in Figure 5.9. When comparing the first and second plots it can be seen that the MPC requests a full throttle opening for a longer time in the constraint free case and also settles faster than the case with $T_{im} \leq 330[K]$. This difference in control signal affects the performance of the pressure build up which as stated before is necessary to keep T_{im} under or equal to the set boundaries. The same reasoning can be applied for the $T_{im} \leq 320[K]$ case. In Figure 5.8 it can be seen that the temperature for the two

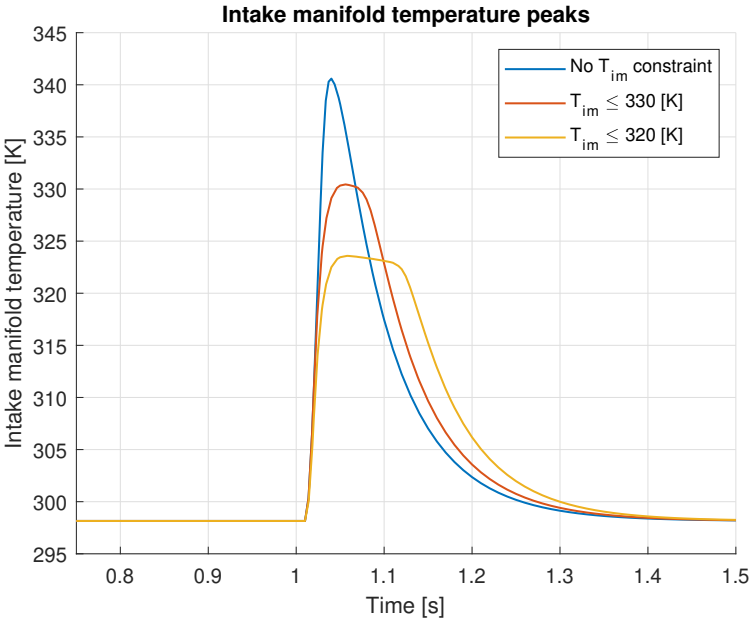

Figure 5.8: *Corresponding intake manifold temperature for the step responses in Figure 5.7 can be seen.*

constraints are not strictly respected. The red curve reaches $T_{im} = 330.4[K]$ and the yellow curve almost reaches $T_{im} = 324[K]$. The reason this occurs is because the models used for the MPC are not perfect but also because the MPC linearizes these non-linear models poorly which ultimately leads to a problem in the prediction of the T_{im}.

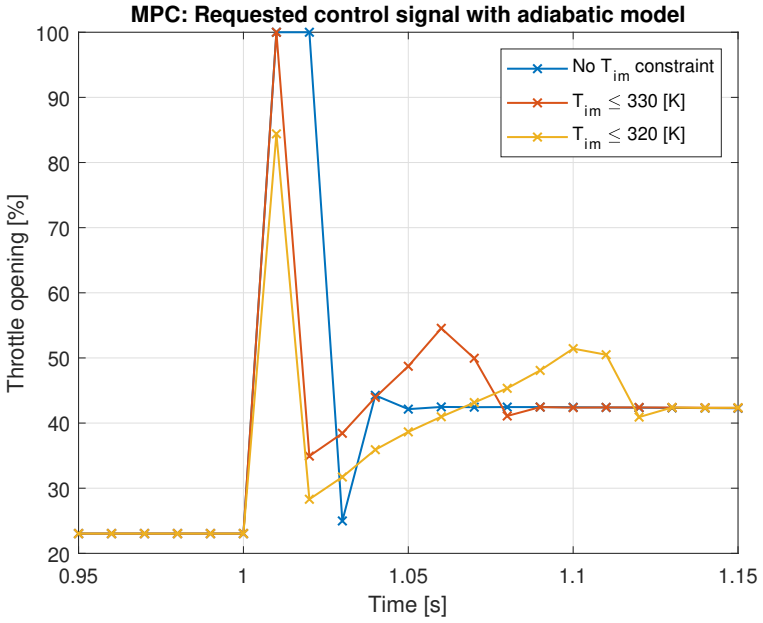

Figure 5.9: *Corresponding intake manifold temperature for the step responses in Figure 5.7 can be seen.*

5.4 MPC vs PID

The MPC controller with integral action is compared to a PID controller with λ-tuning. The PID controller is the conventional solution for the gas-exchange process today. The comparison of the two controllers is done as SISO controllers. Which means that the MPC like the PID only considers one control signal.
The first comparison can be seen in Figure 5.10 with an error of 1000 [Pa] implemented in x_2. A step in p_{im} is made from 20 [kPa] to 50 [kPa] and the corresponding requested control signals for both controllers can be seen in Figure 5.11. Here it can be seen that the MPC is faster than the PID controller in reaching the reference pressure, however there is no significant difference, thus both are acceptable. The overshoot in the MPC is much smaller than the PID's overshoot and there is no stationary error in both cases and they follow the pressure reference well.
The second comparison can be seen in Figure 5.12 with an error of 1000 [Pa]. A step in the intake manifold pressure is made from 20 [kPa] to 70 [kPa] and the corresponding requested control signals for both controllers can be seen in Figure 5.13. In this case too, the MPC is faster than the PID controller. Both controllers have a little overshoot and stationary error which is eliminated when the step is settled. Both controllers follow the reference well in this case too.
The overshoot in the MPC controller takes time to settle, this could be changed by tuning the integral action more aggressively. Looking at the requested con-

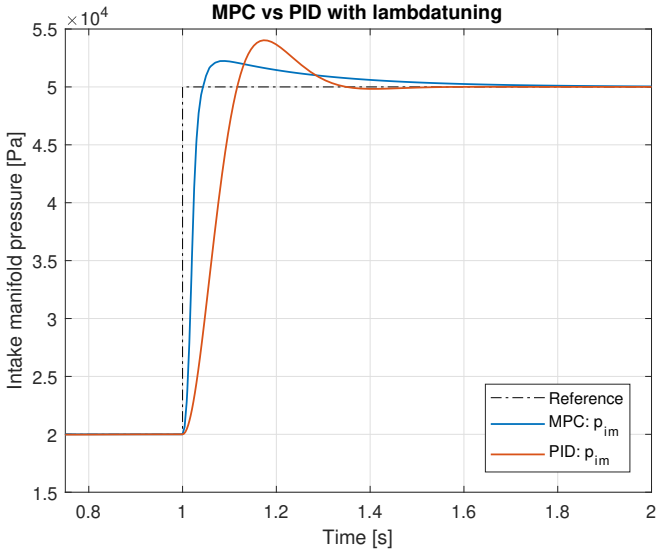

Figure 5.10: *A step from 20 [kPa] to 50 [kPa] is made with both the MPC controller and a PID controller. Also a 1000 [Pa] error is implemented.*

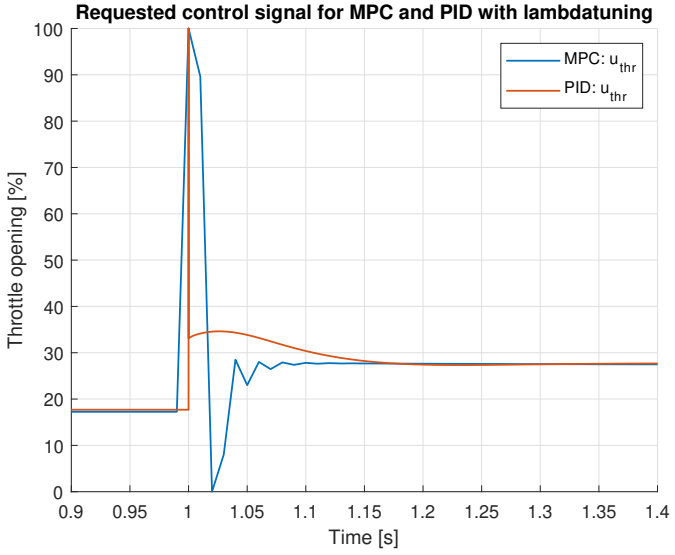

Figure 5.11: *The requested control signal, u_{th} for the step response in Figure 5.10 for both controllers separately.*

trol signals in Figure 5.11, it can be seen that the PID control signal reaches 100 [%] but instantly drops to a 35[%] - 40[%] range and settles at approximately 28

[%] . On the other hand the MPC control signal requests 100 [%] a longer time than the PID and then also settles at 28 [%].

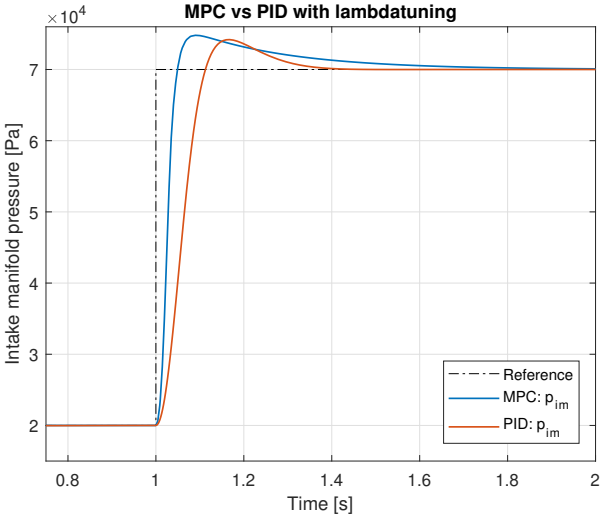

Figure 5.12: *A step from 20 [kPa] to 70 [kPa] is made with both the MPC controller and a PID controller. The intake manifold pressure has a measurement error of 1000 [Pa] implemented.*

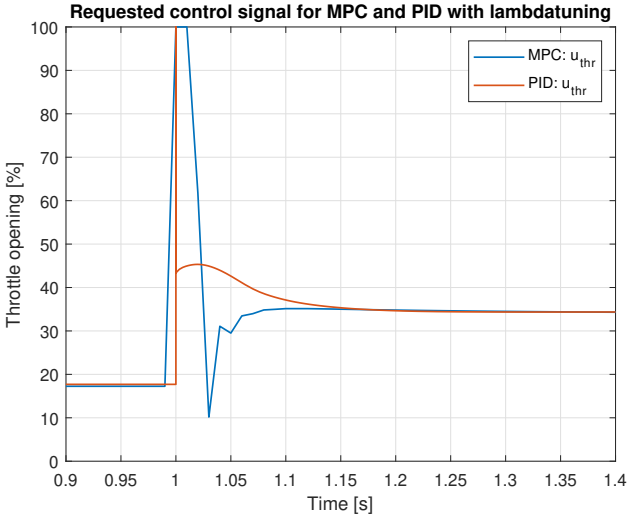

Figure 5.13: *The figure shows the requested control signal u_{th} for the step response in Figure 5.12 for both controllers separately.*

5.5 The MIMO MPC

For testing purposes the state model as seen in Eq.(4.1) is expanded with a state for the simplified VVT model as seen in Eq.(3.13). With this the MPC can now control two in-puts which are the throttle angle control signal u_1 and volumetric efficiency control signal u_2.
This section will present the results of two cases which are compared in the figures below. In these cases the following cost-function was used:

$$\min \quad J = (p_{\text{im,ref}} - p_{\text{im}})^T Q_1 (p_{\text{im,ref}} - p_{\text{im}}) + u^T Q_2 u \tag{5.1}$$

and the following penalty matrices:

$$\text{Case 1}: \quad Q_1 = 1\text{e}16, \qquad Q_2 = \begin{bmatrix} 1\text{e}9 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\text{Case 2}: \quad Q_1 = 1\text{e}16, \qquad Q_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1\text{e}9 \end{bmatrix}$$

In Figure 5.14 it can be seen that the pressure reference tracking is similar in both cases, this is because both cases share the same Q_1

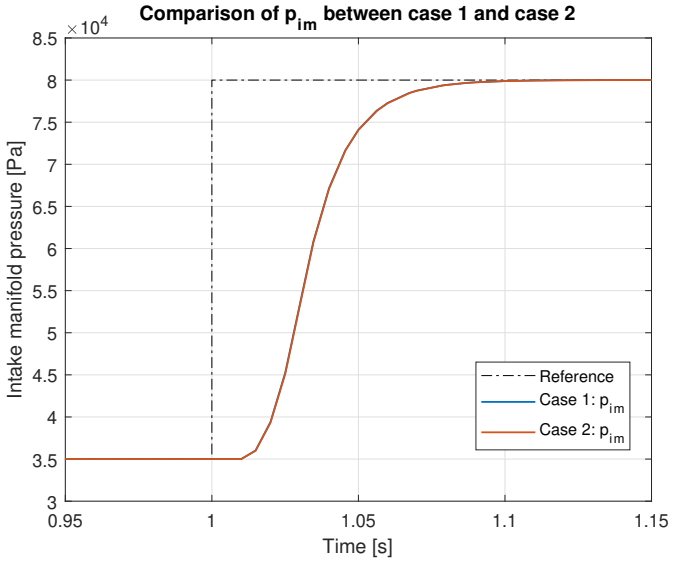

Figure 5.14: *The two steps for the two cases from 35 [kPa] to 80 [kPa] in p_{im} with the MIMO MPC.*

In Figure 5.15 and 5.17 the blue curves represent the case where u_1 is penalized and where the red curve is not penalized at all. Even though the use of u_1 is penalized the blue case assumes slightly higher values than the red case where u_1 is not penalized.

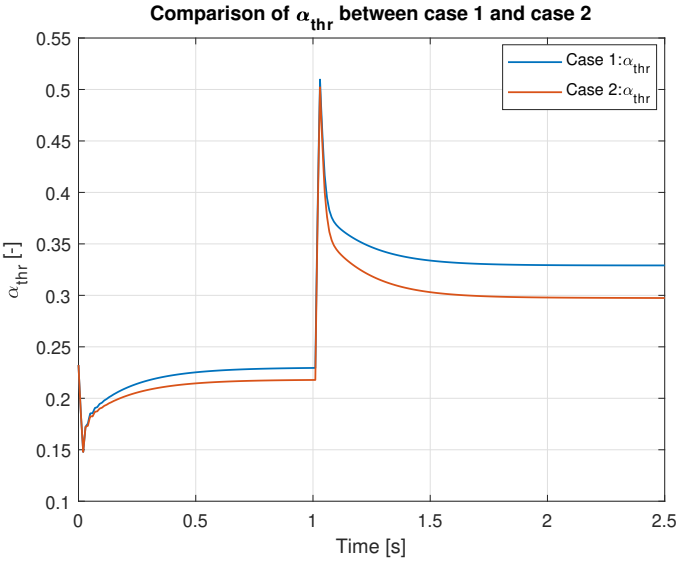

Figure 5.15: *A comparison of the α_{thr} state between case 1 and case 2.*

Furthermore it can be seen in Figure 5.16 and 5.18 that when u_2 is penalized which is seen in the red curves it assumes lower values than in the blue curves.

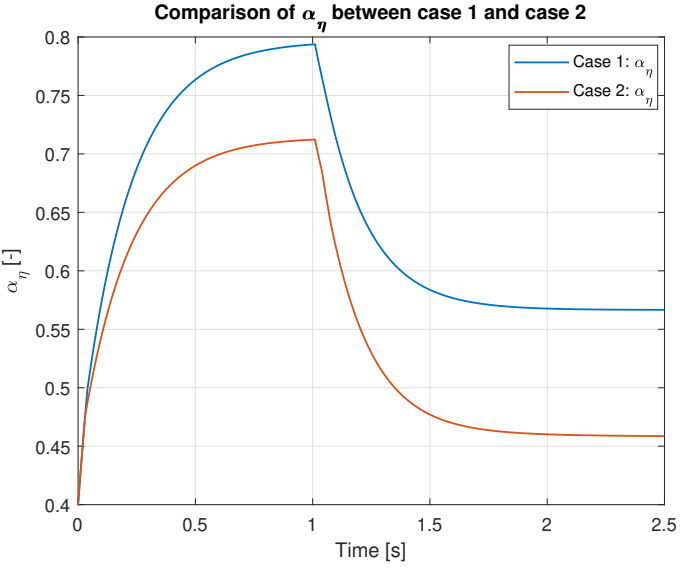

Figure 5.16: *A comparison of the α_{eta} state between case 1 and case 2.*

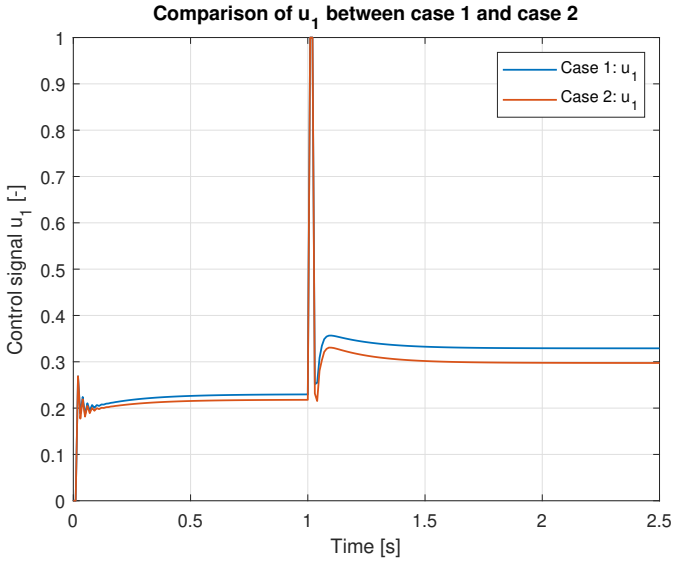

Figure 5.17: *A comparison of control signal u_1 between case 1 and case 2.*

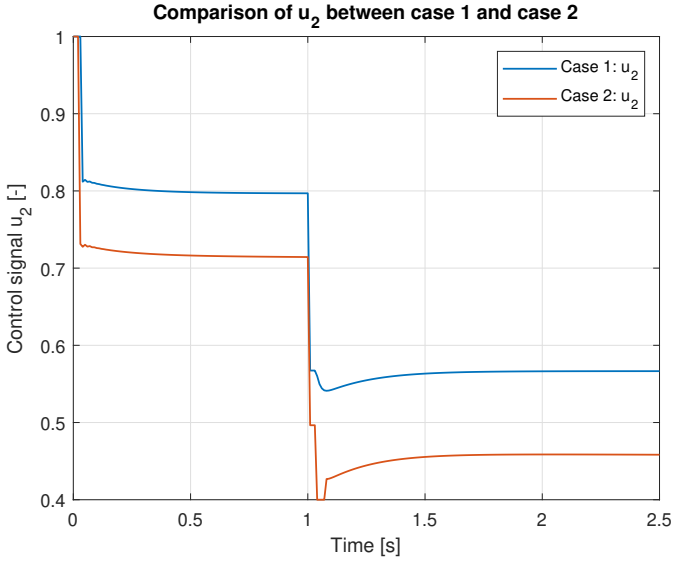

Figure 5.18: *A comparison of control signal u_2 between case 1 and case 2.*

6

Discussion

6.1 The Basic MPC

With a cost-function that tracks the intake manifold pressure towards a given reference the results in Figure 5.1 are as expected. The step response is quick and given the implemented static error in x_2 the overshoot is acceptable and settles fairly quickly. In the lower plot the requested control signal for the throttle can be seen. At the time the step is taken the control signal is requested to be at full throttle, this is an expected result since the pressure reference is prioritized in the control problem hence the throttle opens fully to realize the objective as fast as possible.
In Figure 5.2 the step response has no static error implemented and the step response is quick with no overshoot. However the behaviour of the requested control signal is the same as in the case with a static error since they both use the same cost-function.

6.2 YOP vs MPC

For this comparison the integral action in the MPC was turned off. The results of these tests show that the MPC closely follows the results of the numerical solution from YOP. There are however differences in the states and also the control signals. The states between the two most probably differ because of pressure ratio Π. The state-model used in YOP only considers one case which is $\Pi = \frac{p_{\text{im}}}{p_{\text{amb}}}$. The MPC however has three different cases of Π which describe the flow Ψ more accurately see Figure 3.1. For the comparison a step was made where both configurations were in the same Π-region. When comparing the behaviour of the control signals a big difference can be observed. The YOP control signal is probably affected for the same reason as the states are. When looking at the

MPC control signal it is less oscillate, this is the result of the MPC iterating and solving the control problem for each sample time. The comparison between the two gives a validation that the MPC was implemented correctly which was the purpose of the use of YOP.

6.3 Adiabatic control volume

The tests seen in Figure 5.7 show that there are differences in performance between constraining the intake manifold temperature and not constraining it. All three cases achieve the reference that is set however the pressure buildup is slower and not as smooth when constraining the temperature in the intake manifold. The reason is that with faster pressure build up that also means a faster compression rate which would heat up the intake manifold and thus exceed the boundaries set on the temperature. If there is a need of temperature limitation this is a good way to do so in MPC. Since there are no other conditions specified on the system all results seen in Figure 5.7 are acceptable. If several aspects are to be considered such as rise time or smooth pressure build up this can be translated to constraints which then are all considered by the MPC. As long as a constraint can be deduced linearly by the optimal control signal sequence U, it can be implemented as was done with the intake manifold temperature T_{im}.
For the other implementations of MPC in this thesis, the model for an isothermal control volume was used. The model uses the constant temperature $T_{\text{im}} = 298.15[K]$ and lacks a state for T_{im}.The adiabatic control volume results in a slightly faster pressure build up, however the difference is insignificant.
The adiabatic control volume model for the intake manifold was not chosen for the MPC since the difference in dynamics between the adiabatic and isothermal models was insignificant. For the purposes of this master thesis it sufficed to use the isothermal model meaning one less state in the state model.

6.4 MPC vs PID

The reason that the MPC controller requests a maximized control signal when the step is taken is because of the control problem which in this case optimizes the reference tracking and has a very low penalty on the usage of the control signal. For the requested control signals in Figure 5.13 the same reasoning can be applied.
Since both controllers are of SISO type the results are both acceptable and this can be validated by the fact that PID controllers are conventional in this field of work. However the PID controller can be tuned further to match the speed and overshoot of the MPC. In an ideal situation these controllers are equally good in performance but the MPC controller's main advantage in this case would be the flexibility the MPC offers in the form of constraints and penalty matrices but also the ability to expand the models that are used for higher accuracy, where the PID controller would not be able to handle this without external help.

6.5 The MIMO MPC

When comparing the steps in p_{im} as seen in Figure 5.14 the performance is the same. The pressure reference is achieved quickly since in both cases x_2 is prioritized by putting a high value on it in Q_1. The difference in these tests is the high penalty put on either u_1 or u_2. In the first case there is a high penalty on the use of the throttle control signal, but when the step in pressure is taken the MPC requests maximum throttle signal and the volumetric efficiency signal drops to the 0.55 range, which can be seen in Figure 5.17 and 5.18 . When directly compared to case 2, where instead u_2 is highly penalized it can be seen that the volumetric efficiency signal drops to the 0.4 range. An explanation for this could be that the solution is more optimal when using the throttle for pressure build-up even though that it is highly penalized in this case, meaning that the QP-solver profits more from minimizing u_2.
For the other implementations of MPC in this master thesis the volumetric efficiency was constant at $\eta_{\text{vol}} = 0.8$, these tests may serve as an indication that a fully controlled η_{vol} is desirable.

7

Conclusions

This chapter's aim is to draw conclusions based on the testing of several implementations of MPC in Chapter 5 and MPC in general.
The MPC with integral action has good performance in pressure tracking. This can be seen in Figure 5.11 where it reaches the reference pressure fast and also how it deals with a large static error. When turning the integral action off and then comparing it to the results of the numerical solution in YOP of the same step it can be concluded that the MPC controller was implemented correctly, even though there are slight differences between the results. These differences are explained and identified in Chapter 6.
When comparing the MPC with a PID controller, which is the conventional type of controller for this field of work, it can be seen that the results are close to each other. The PID controller could be tuned further however the pressure tracking is fast enough and sufficient for the purpose of this comparison. There is no clear preference in controller in this case since they both perform well in this context however the interesting question is how physical couplings are dealt with using these types of controllers. The MPC deals with such physical couplings by design whereas the PID controller would need external assistance. If development time is not an issue the MPC would still be the more flexible and long lasting choice since it's structure is easy to understand and continue to expand upon within a certain system. The choice of an MPC controller over a PID controller can be further motivated by the demonstration of constraints. An implementation of MPC with an adiabatic control volume was tested with three levels of constraints on the intake manifold temperature where the advantages of such constraints are discussed in Chapter 6.

The results suggest that an isothermal control volume is sufficient however the implementation of an adiabatic model could easily be added further in the development when deemed necessary for example if limiting a temperature is

desired. Overall this test proves the argument against a PID controller which cannot handle the constraints of a system naturally. If a system is of MIMO type with physical couplings and development time is not an issue, MPC is the long lasting choice.
Lastly an implementation of MPC with two control signals was tested. The pressure tracking is sufficient as before however the tests that were made focused on penalizing one or the other control signal more. The interesting part of this test is the behaviour of the requested control signals. In both cases the control signal for the throttle angle seemed to be used maximally even though it was highly penalized. The control signal for the volumetric efficiency however was affected more clearly when comparing both tests, in the test where the volumetric efficiency was highly penalized it dropped to the lower bound of the defined constraint when the step in pressure was made. A reasonable conclusion of this specific implementation of MPC is that the volumetric efficiency varies depending on several factors and that a freely controlled volumetric efficiency is better than a constant one, which was the case in all previous implementations of MPC in this master thesis. With the MPC in it's current state the fuel economy can not be improved since the cost-function does not focus on that. However better fuel economy should be possible if the cost-function is altered, as the proposed cost-function in Section 4.1.5

7.1 Future work

This section is meant to propose future work and what could be done to improve on this subject.

- How does the MPC behave for different cost-functions? It would be interesting to see how the MPC would perform when for example minimizing the pumping work losses. When developing a MPC with multiple in-puts and multiple out-puts several approaches should be evaluated.
- Full implementation of models concerning the gas-exchange process. A MPC that includes a higher order system for the throttle, the adiabatic control volume model, a better VVT model and a model for the wastegate. It would be interesting to see a MPC with higher accuracy and three control signals.

A

Appendix

A.1 Engine Parameters

Parameter	Value	Unit
R	2.93000e2	[g/mol]
V_{im}	1.80000e-3	
T_{im}	2.98150e2	[K]
η_{vol}	0.8	[-]
V_D	1.95300e-3	
N	1980	[rpm]
n_r	2	[-]
p_{amb}	1.01325e5	[Pa]
T_{amb}	2.98150e2	[K]
τ_α	0.04426	[-]
γ	1.4	[-]
a_0	5.18849e6	[-]
a_1	-6.38593e5	[-]
a_2	1.23068e3	[-]

A.2 SISO MPC

The MPC controller developed for this master thesis is flexible, it is programmed with symbolic code and has an interface in Simulink. Because of this it is easily re-configured and allows for expanding the dynamic of the system by simply coding them by terms of states in symbolic expression. Furthermore the definitions of the MPC controller can be easily changed for example the number of

states and in-puts. The MPC controller in Simulink allows for easy adjustments or further modeling if desirable.
The SISO MPC controller with integral action developed for this master thesis can be found at `https://github.com/georgejajji/MPC-for-gasexchange-in-gaso`

A.3 MATLAB

A.3.1 MATLAB Plots

This section covers the intake manifold step responses made with the MPC with integral action. Five step responses are presented and all of them have an error of 5000 [Pa] implemented.

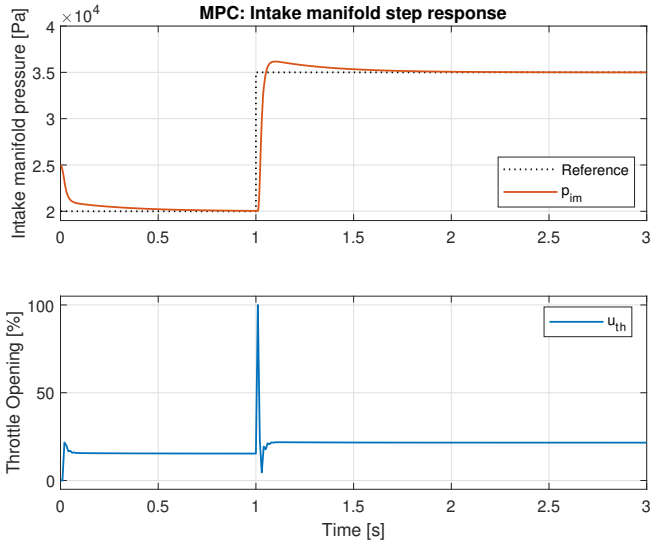

Figure A.1: *A step response from 20 [kPa] to 35 [kPa] with an implemented error of 5 [kPa].*

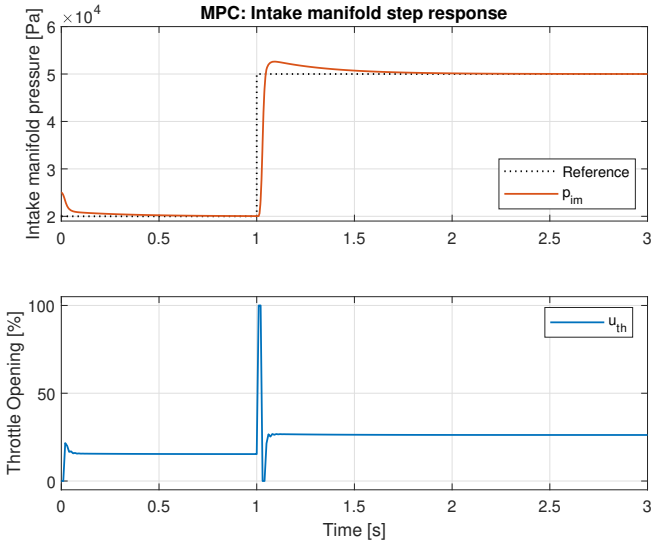

Figure A.2: *A step response from 20 [kPa] to 50 [kPa] with an implemented error of 5 [kPa].*

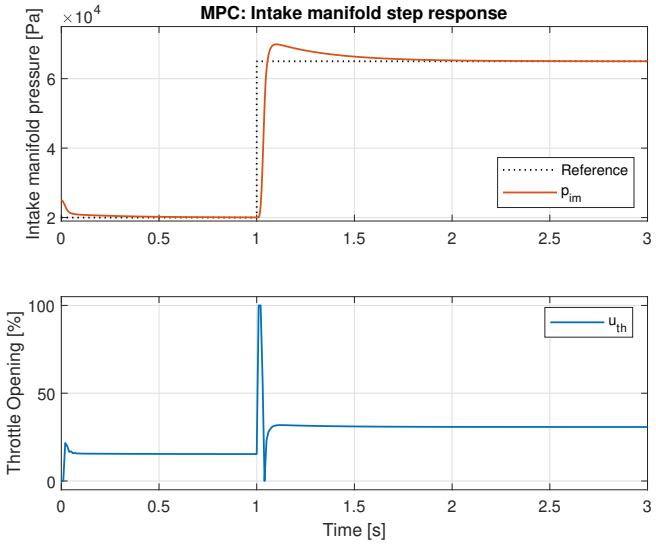

Figure A.3: *A step response from 20 [kPa] to 65 [kPa] with an implemented error of 5 [kPa].*

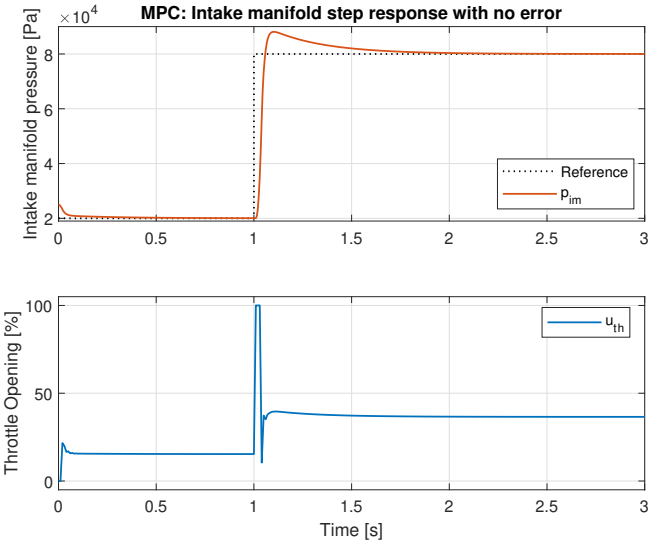

Figure A.4: *A step response from 20 [kPa] to 80 [kPa] with an implemented error of 5 [kPa].*

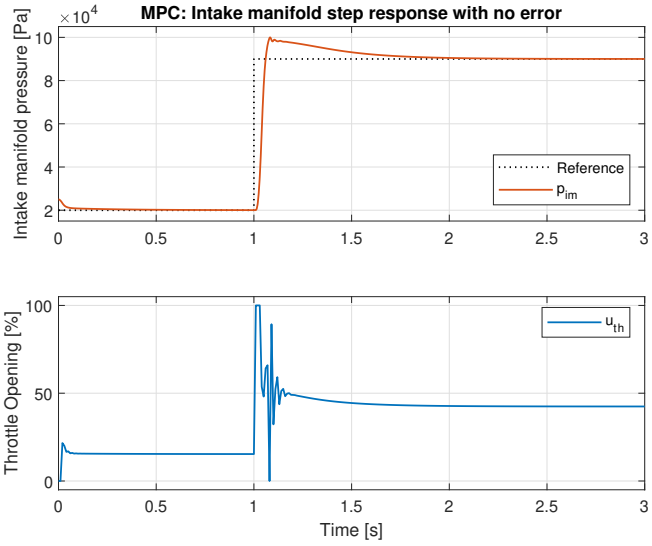

Figure A.5: *A step response from 20 [kPa] to 90 [kPa] with an implemented error of 5 [kPa].*

Bibliography

Aimin, A., Xiaohong, H., Chao, Z., and Hongye, S. (2009). A pragmatic approach for selecting weight matrix coefficients in model predictive control algorithm and its application. *2009 IEEE International Conference on Automation and Logistics, Automation and Logistics, 2009. ICAL '09. IEEE International Conference on*, pages 486 – 492.

Allgöwer, F., Findeisen, R., and Ebenbauer, C. (1999). Nonlinear model predictive control.

Andreas, T. (2018). Cylinder pressure based cylinder charge estimation in diesel engines with dual independent variable valve timing.

Bemporad, A. (2016). A quadratic programming algorithm based on nonnegative least squares with applications to embedded model predictive control. *IEEE Transactions on Automatic Control, Automatic Control, IEEE Transactions on, IEEE Trans. Automat. Contr*, 61(4):1111 – 1116.

Bemporad, A., Bernardini, D., Long, R., and Verdejo, J. (2018). Model predictive control of turbocharged gasoline engines for mass production.

Bemporad, A. and Patrinos, P. (2012). Simple and certifiable quadratic programming algorithms for embedded linear model predictive control. *IFAC Proceedings Volumes*, 45(17):14 – 20.

Eriksson, L. and Nielsen, L. (2014). *Modeling and control of engines and drivelines.* Automotive series. Wiley.

Fantenberg, E. (2018). Estimation of air mass flow in engines with variable valve timing.

Ferreau, H. J., Potschka, A., and Kirches, C. (2019). Open-source c++ implementation of the recently proposed online active set strategy.

Forbes, M. G., Patwardhan, R. S., Hamadah, H., and Gopaluni, R. B. (2015). Model predictive control in industry: Challenges and opportunities. *IFAC PapersOnLine*, 48(8):531 – 538.

Holmbom, R. and Eriksson, L. (2018). Analysis and development of compact models for mass flows through butterfly throttle valves. *SAE Technical Papers*, 2018-April.

Honeywell (2015). Honeywell's software design suite onramp helping auto industry with powertrain development.

Jalali, A. and Nadimi, V. (2006). A survey on robust model predictive control from 1999-2006. *2006 International Conference on Computational Inteligence for Modelling Control and Automation and International Conference on Intelligent Agents Web Technologies and International Commerce (CIMCA'06), Computational Intelligence for Modelling, Control and Automation, 2006 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on*, page 207.

Leek, V. (2016). An optimal control toolbox for matlab based on casadi.

Leroy, T., Alix, G., Chauvin, J., Duparchy, A., and Le Berr, F. (2009). Modeling fresh air charge and residual gas fraction on a dual independent variable valve timing si engine. *SAE International Journal of Engines*, 1(1):627.

Li, J., Julien, V., Hakan, Y., and Anna, S. (2009). Parameterization and simulation for a turbocharged spark ignition direct injection engine with variable valve timing.

Manfred, M. and Jay H., L. (1997). Model predictive control: Past, present and future.

Martin Enqvist, Torkel Glad, S. G. P. L. L. L. J. L. T. M. A. S. J.-E. S. (2014). *Industriell Reglerteknik Kurskompendium*. Reglerteknik, Institutionen för Systemteknik, Linköpings universitet.

Nilsson, Amanda, W. L. M. S. E. E. B. A. A. M. J. L. S. G. (2018). Flervariabla reglerstrategier för avancerade motorer.

Polterauer, P., Incremona, G. P., Colancri, P., and del Re, L. (2019). A switching nonlinear mpc approach for ecodriving. *2019 American Control Conference (ACC), Control Conference (ACC), 2019 American*, pages 4608 – 4613.

Rivera, D. E. (1999). Internal model control: A comprehensive view.

S. Joe, Q. and Thomas A., B. (1997). An overview of industrial model predictive control technology.

Von Wissel, D., Talon, V., Thomas, V., Grangier, B., Lansky, L., and Uchanski, M. (2016). Linking model predictive control (mpc) and system simulation tools to support automotive system architecture choices.

Yang, W. and Boyd, S. (2010). Fast model predictive control using online optimization. *IEEE Transactions on Control Systems Technology, Control Systems Technology, IEEE Transactions on, IEEE Trans. Contr. Syst. Technol*, 18(2):267 – 278.