Simulation Testbed of a Combustion Engine in Simulink for Control Purposes

Erik Bolin

LUND
UNIVERSITY

Department of Energy Science

MSc Thesis
ISRN : LUTMDN/TMHP-19/5448-SE
ISSN : 0282-1990

Department of Energy Science
Lund University
Box 118
SE-221 00 LUND
Sweden

Printed in Sweden.
Lund 2019

Abstract

Control theory plays a big part in the development of more efficient and environmental friendly combustion engines. But the process of designing good controllers can be very time consuming and complex. One way to reduce the time spent on design and testing of a controller is to use simulations to get fast feedback of the controllers performance. Therefore, in this thesis it is investigated if a fast and simple but yet accurate enough simulation of a combustion engine can be implemented in Simulink. It is also of interest to see if this simulation can be used to investigate possibilities of using iterative learning control (ILC) to improve an already know pressure tracking model predictive control (MPC) strategy.

This is done by implementing already validated models in the simulation interface and comparing the simulated results to experimental data. Already know and tested MPC strategies are tested to see the applicability of this controller together with the simulation and software related to Simulink.

Because of the possibly iterative behaviour occurring with the pressure tracking MPC, this controller is tested more thoroughly to see if an ILC strategy can be used to improve pressure tracking performance by compensating for model errors and possibly noise disturbance errors.

It is concluded that this simulation can be used in early design stages of a controller to find out the general control performance. But that to many dynamics are lost because of assumptions and simple models so it can not be used for tuning of controllers. Results regarding ILC is that no ILC design could be found that improves the pressure tracking but that ILC can not be ruled out and there is still much more to investigate both in simulation and in the real process.

Acknowledgements

I would like to give my thanks to my supervisor Per Tunestål who have given me the opportunity to work with a really interesting project and also helped me along the way. Next I would like thank Rolf Johansson who also has been giving me great advise through out the project.

Now I would like to acknowledge three PhD students who gave me a lot of their time and helped me keep this project going at all times. First I want to thank Alvin Barbier who in the beginning of the project introduced me and taught me a lot about combustion engines in general in a very pedagogical and interesting way which gave me a lasting interest of combustion engines. Next I want to acknowledge Björn Olofsson who I have had many interesting and productive discussions with about the project and control in general. Lastly I want to thank Xiufei Li who has been a great advisor in questions about both combustion engines and control but I am also thankful for him giving me a hospitable and enjoyable working experience as we have been sharing office.

And to finish I would like to thank the whole Department of Energy Science for lending me a working space, equipment and many new connections. But also for giving me a very educational and rewarding time that I will have great use of in all my future endeavours.

Contents

1

Introduction

A majority of vehicles today run with an internal combustion engine. Most of these are driven with fossil fuels, for example, petrol, diesel and natural gas. Burning of these fuels form compounds and greenhouse gases which results as exhausts from the engine. Emissions of greenhouse gases is one of the reasons for global warming and one example of greenhouse gas that is formed in the engine is carbon dioxide (CO_2). Other exhausts that have more hazardous effects on local environment and human health are soot and nitrogen oxides (NO_x). The transportation sector is one of the biggest contributors to emissions of greenhouse gases globally. Therefore, to achieve the global and international emission goals improvements in terms of fuel reduction and efficiency of combustion engines has to be made. With alternatives as electrical vehicles there is a vision of zero emissions from the transport section. But because of the vast majority of vehicles running with combustion engines it would be impossible to completely replace the combustion engine in a near future. A transition phase is needed to phase out less efficient combustion engines with more efficient once to reach the emission goals and follow legislations.

Sensors has gone through great development and it is today possible to measure values that were not possible before. This has opened up possibilities for more advance control methods which have improved fuel reduction and efficiency of combustion engines. Control have a big impact on performance of the combustion engine but designing new controllers can be time consuming and require a lot of tuning. An efficient way of reducing the time spent in the design stage of a controller is through simulation where the controller can be tested, tuned and analyzed before tests on the real system are conducted. Not only makes a good simulation the design stage faster but can also find if the designed controller lacks in stability and robustness which could lead to unnecessary damaging of the real engine if applied directly.

1.1 Contribution

The contributions in this thesis will be the implementations of a simulation of an combustion engine in the mathematical computing software Simulink and MATLAB. What this thesis is to investigate is if it is possible to implement a fast and accurate Simulation good enough for control purposes in Simulink. Simulink and Matlab was chosen for their fast and accurate simulations and also their availability in the academic and research field. The goal of the simulation is that it can be used in early design stages to save time and confirm new or existing control strategies.

The simulation is built on models introduced and validated in Gabriel Turesson's doctoral thesis "*Model-Based Optimization of Combustion-Engine Control*" [Turesson, 2018]. Control strategies in this work are also based on earlier work from Turesson. These control strategies are used to test the applicability of controllers in the simulation and also to further investigate options with iterative learning control (ILC) together with pressure tracking model predictive control (MPC).

The hypothesis is that because of the iterative behaviour occurring with the pressure tracking MPC and the desire to follow the same reference trace for many cycles, ILC might be able to compensate for model errors and possibly recurring noise dynamics. ILC has been used before in combustion engine control but in different applications [Zweigel et al., 2015] [Slepicka and R. Koch, 2016] [Yan and Wang, 2011].

1.2 Outline

The thesis starts with two chapters about theory used. The Chapter 2 is about the models used in the simulation and controllers then a Chapter 3 is about theory behind the control strategies MPC and ILC. Chapter 4 is about the implementation of the simulation, how the models are used and what software has been used. Then in Chapter 5 and 6 the implemnetation and results from ignition delay MPC and pressure tracking MPC are presented. In Chapter 6 a discussion about ILC is also included. Last is Chapter 7 where results from all the chapters are concluded.

Chapter 2

In this chapter a four stroke combustion cycle and models used in this thesis are introduced and explained.

Chapter 3

In this chapter the basic theory about control principles MPC and ILC are explained.

Chapter 4

This chapter contains discussion of the software Simulink and the toolbox MPC-tools and modifications made in this toolbox. The chapter also contains the implementation of the simulation and how the models are used and what preperations are needed for the simulation.

Chapter 5

This chapter is about the ignition delay MPC introduced in [Turesson, 2018] and how this controller works together with the simulation, MPCtools toolbox and with the modified MPCtools. This controller was used to investigate the compatibility with the simulation and MPCtools rather than its control performance.

Chapter 6

Chapter 6 contains the implementation of a pressure tracking MPC based on a controller introduced in [Turesson, 2018]. Here the pressure tracking performance is investigated with and without noise disturbances and also possibilities for ILC implementations are discussed.

Chapter 7

This last chapter concludes all the results from previous chapters.

Table 1.1: Table with model variables

Notation	*Description*	*Units*
V_d	Displaced Volume	$[\mathrm{m}^3]$
V_c	Clearance volume	$[\mathrm{m}^3]$
B	Bore	[m]
A_{ch}	Cylinder head surface area	$[\mathrm{m}^2]$
A_p	Piston head crown area	$[\mathrm{m}^2]$
l	Connecting rod length	[m]
L	Piston stroke length	[m]
a	Crank radius	[m]
θ	Crank angle degrees	[deg]
N	Rotational speed of crankshaft	$[\frac{\mathrm{rev}}{\mathrm{min}}]$
$\bar{S}_p$	Mean piston speed	$[\frac{\mathrm{m}}{\mathrm{s}}]$
τ	Ignition delay	[ms]
p	In cylinder pressure	[Pa]
Q_c	Heat release	[J]
Q_{ht}	Heat transfer	[J]
V	In cylinder volume	$[\mathrm{m}^3]$
γ	Specific heat ratio	[-]
h_c	Convection coefficient	$[\frac{\mathrm{W}}{\mathrm{m}^2\cdot\mathrm{K}}]$
T	In cylinder temperature	[K]
T_w	Cylinder wall temperature	[K]
w	Average cylinder gas velocity	$[\frac{\mathrm{m}}{\mathrm{s}}]$
p_m	Motored pressure	[Pa]
C_1	Empirical parameter	[-]
C_2	Empirical parameter	$[\frac{\mathrm{m}}{\mathrm{s}\cdot\mathrm{K}}]$
m_f	Injected fuel mass	[kg]

2

Modelling

Models in this chapter are the base for the simulation and the model-based controllers. The models come from [Heywood, 1988] and [Turesson, 2018] where most of the models are derived and validated. In the coming sections the models are introduced and described. Later in Chapter 4 it is more thoroughly discussed how these models are implemented.

2.1 Combustion cycle

A four stroke cylinder cycle in a diesel engine consist of four stages over the coarse of two revolutions of the crank shaft, this is illustrated in Figure 2.1.

1. First is the intake stage that starts at top dead center (TDC) and ends at bottom dead center (BDC). During this stage the inlet valve is open and the exhaust valve is closed and air or an air/fuel mixture goes into the cylinder through the intake valve because of the vacuum created at the down stroke.

2. The second stage is the compression stage which completes the first revolution and here the inlet valve closes. The air is compressed and due to the decrease in volume and increase in pressure the temperature in the cylinder increases.

3. Starting the second revolution from TDC is the expansion stage where heat is released from combustion where the heat energy is converted to work on the piston on the down stroke. Fuel for the combustion is injected close to TDC either late in the compression stage or early expansion stage.

4. The fourth stage is the exhaust stage starting at BDC of the second revolution and here the exhaust valve is open to let the exhausts out of the cylinder.

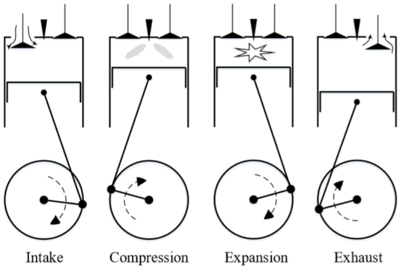


Figure 2.1: Results from simulated pressure compared with experimental data [Li, 2018].

2.2 Cylinder geometry

A lot of necessary variables can be found in the geometry of the cylinder. The combustion dynamics are highly dependant on the geometry of both cylinder and piston. The rest of this section is describing geometry for a common cylinder with flathead piston which is shown in Figure 2.2. A useful ratio is the ratio of connecting rod length to crank radius [Heywood, 1988]

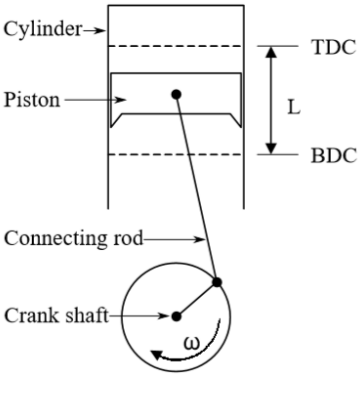


Figure 2.2: Illustrative picture of a cylinder. TDC and BDC stands for top dead center and bottom dead center, respectively [Li, 2018].

$$R = \frac{l}{a} \tag{2.1}$$

Where l is the length of the connecting rod and a is the radius of the crankshaft. Another useful relation is between a and the stroke length, which is the same as the length from TDC to BDC in Figure 2.2, is given as

$$L = 2a \tag{2.2}$$

The cylinder volume can be calculated as [Heywood, 1988]

$$V = V_c + \frac{\pi B^2}{4}(l + a - s) \tag{2.3}$$

where V_c is clearence volume which is the volume in the cylinder at TDC, B is the diameter of the cylinder bore and s is defined as

$$s = a\cos(\theta) + \sqrt{l^2 - a^2\sin^2(\theta)} \tag{2.4}$$

where θ is the crank angle degree (CAD). With this the volume can be expressed as a function of θ and rewritten as

$$V = V_c + \frac{V_d}{2}\left(R + 1 - \cos(\theta) - \sqrt{R^2 - \sin^2(\theta)}\right) \tag{2.5}$$

where V_d is the displaced volume which is the maximum volume in the cylinder subtracted by the clearence volume. In a similar manner as the volume, the surface area in the chamber can be expressed as

$$A = A_{ch} + A_p + \pi B(l + a - s) \tag{2.6}$$

where A_{ch} is the cylinder head area and A_p is the piston crown surface area. For flat-top pistons the chamber surface area is

$$A = A_{ch} + A_p + \frac{\pi BL}{2}\left(R + 1 - \cos(\theta) - \sqrt{R^2 - \sin^2(\theta)}\right) \tag{2.7}$$

The volume rate in volumes per CAD is analytically calculated as

$$\frac{dV}{d\theta} = \frac{V_d}{2}\left(\sin(\theta) + \frac{1}{2}\sin(2\theta)(R^2 - \sin^2(\theta))^{-\frac{1}{2}}\right) \tag{2.8}$$

The mean piston speed $\bar{S}_p$ is calculated as

$$\bar{S}_p = 2LN \tag{2.9}$$

Where N is rotational speed of the crankshaft.

2.3 Combustion dynamic models

From the first law of thermodynamics for an open system the following expression is given [Heywood, 1988]

$$\frac{dU}{dt} = \frac{dQ}{dt} - p\frac{dV}{dt} + \sum_i \dot{m}_i h_i \tag{2.10}$$

where U is the internal energy, Q is the total heat transfer across the boundary to or from the system, $\dot{m}_i$ mass flow rate across the boundary at location i, p is the pressure and h_i is the enthalpy of flux entering or leaving from i. If U and h_i are substituted with the sensible internal energy, U_s and sensible enthalpy, $h_{s,i}$, then the heat release becomes the difference of the heat released from combustion and the heat transferred from the boundaries

$$\frac{dQ}{dt} = \frac{dQ_c}{dt} - \frac{dQ_{ht}}{dt} \tag{2.11}$$

The sensible internal energy and enthalpy is defined as the changes in internal energy and enthalpy resulting only from changes in temperature, changes from chemical reactions and phasing are not taken into consideration. With Equation (2.11) and that $h_{s,i}$ is assumed to be very small it results in

$$\frac{dU_s}{dt} = \frac{dQ_c}{dt} - \frac{dQ_{ht}}{dt} - p\frac{dV}{dt} \tag{2.12}$$

With further assumptions that the mixture in the cylinder has reached a fully vaporized state and that this gas is modeled as an ideal gas then the sensible internal energy becomes

$$\frac{dU_s}{dt} = mc_v\frac{dT}{dt} \tag{2.13}$$

where m is the mass of the gas, T is the temperature of the gas and c_v is the specific heat for constant volume. In an ideal gas it follows that

$$pV = mR_sT \tag{2.14}$$

and

$$\frac{dT}{T} = \frac{dp}{p} + \frac{dV}{V} \tag{2.15}$$

where R_s is the specific gas constant and is assumed to be constant and has the property

$$R_s = c_p - c_v \Leftrightarrow \gamma = \frac{c_p}{c_v} = 1 + \frac{R_s}{c_v} \tag{2.16}$$

Where c_p is the specific heat at constant pressure and γ is the specific heat ratio. From these results Equation (2.12) can be rewritten as

$$\frac{dp}{dt} = -\frac{\gamma}{V}p\frac{dV}{dt} + \frac{\gamma - 1}{V}(\frac{dQ_c}{dt} - \frac{dQ_{ht}}{dt}) \tag{2.17}$$

With a constant engine speed dt has a linear relation to $d\theta$ and the differential equation for the pressure rate becomes

$$\frac{dp}{d\theta} = -\frac{\gamma}{V}\frac{dV}{d\theta}p + \frac{\gamma - 1}{V}\left(\frac{dQ_c}{d\theta} - \frac{dQ_{ht}}{d\theta}\right) \tag{2.18}$$

Equation (2.18) can be rearranged for finding the heat release rate

$$\frac{dQ_c}{d\theta} = \frac{1}{\gamma - 1}V\frac{dp}{d\theta} + \frac{\gamma}{\gamma - 1}p\frac{dV}{d\theta} + \frac{dQ_{ht}}{d\theta} \tag{2.19}$$

Heat transfer

The heat transfer is modeled as convection between the mixture in the cylinder and the in-cylinder surfaces [Heywood, 1988]

$$Q_{ht} = h_c A(T - T_w) \tag{2.20}$$

where T is the in-cylinder gas temperature, T_w is the temperature of the cylinder walls, A is the surface area in the cylinder and h_c is the convection coefficient. Because of the constant engine speed the heat transfer rate the is obtained by adding a term of $1/6N$

$$\frac{dQ_{ht}}{d\theta} = \frac{h_c A}{6N}(T - T_w) \tag{2.21}$$

The convection coefficient h_c follows Woschni's correlation and is calculated as

$$h_c = 3.26B^{-0.2}p^{0.8}T^{-0.55}w^{0.8} \tag{2.22}$$

where w is the mean in-cylinder gas velocity which is modeled as [Turesson, 2018]

$$w = C_1\bar{S_p} + C_2\frac{V_d T_{IVC}}{p_{IVC}V_{IVC}}(p - p_m) \tag{2.23}$$

C_1 and C_2 are empirical values, p_m is the motored pressure and T_{IVC}, p_{IVC} and V_{IVC} are the in-cylinder temperature, pressure and volume at inlet valve closing, respectively. From the assumption of an ideal gas law the combined gas law can be applied which says that the following ratio are the same

$$\frac{p_1 V_1}{T_1} = \frac{p_2 V_2}{T_2} \tag{2.24}$$

By reformulating this the in-cylinder temperature is calculated by using the inlet valve closing conditions

$$T = \frac{V T_{IVC}}{p_{IVC}V_{IVC}}p \tag{2.25}$$

2.4 Ignition delay

The ignition delay is defined as the time between start of injection and start of combustion

$$\tau = \frac{1}{t_c}(\theta_{SOC} - \theta_{SOI}) \tag{2.26}$$

where t_c is the conversion constant from milliseconds to CAD, θ_{SOI} is the start of injection crank angle and θ_{SOC} is the crank angle where combustion starts. The following model is used in [Turesson, 2018] and is one of the more simple ignition delay models

$$\tau = A_t \overline{[O_2]}^{\alpha} e^{E_a/\tilde{R}\overline{T}} \tag{2.27}$$

where A_t, α and E_a are fuel dependant empirical parameters, $\tilde{R}$ is the universal gas constant, $\overline{[O_2]}$ and $\overline{T}$ are the mean oxygen concentration and mean temperature between θ_{SOI} and θ_{SOC}. The oxygen concentration calculated using the model

$$[O_2] = \frac{[O_2]_{IVC}}{V} V_{IVC} \tag{2.28}$$

The in-cylinder temperature is derived from the adiabatic model

$$T = T_{IVC}\left(\frac{V_{IVC}}{V}\right)^{\gamma-1} \tag{2.29}$$

With Equations (2.28) and (2.29), Equation (2.27) can be rewritten on the following form

$$\tau = A_t \exp\left(\frac{E_a}{\frac{\tilde{R}}{\tau}\int_{\theta_{SOI}}^{\theta_{SOI}+\tau} T_{IVC}\left(\frac{V_{IVC}}{V(\theta)}\right)^{\gamma-1} d\theta}\right)\left(\frac{1}{\tau}\int_{\theta_{SOI}}^{\theta_{SOI}+\tau} \frac{[O_2]_{IVC}}{V(\theta)} V_{IVC} d\theta\right)^{\alpha} \tag{2.30}$$

3

Control

3.1 Model Predictive Control

Model predictive control (MPC) is categorized as a receding-horizon controller [Åström and Wittenmark, 1989]. What characterizes a receding-horizon controller is that it with a model of the process to be controlled and the current state of the process, predicts the outcome of different control sequences. The one sequence that achieves the best predicted behaviour in comparison to desired control goal is chosen. Only the first signal in this sequence is applied to the real process and the rest are discarded. This whole procedure is then repeated every time step to always find an "optimal" control signal in terms of predicted outcome based on the current measured states.

MPC computes these predictions online over a so called prediction horizon H_p that limits how many time steps into the future that controller should predict. The prediction horizon is an important design parameter. It has to be sufficiently large to cover the necessary process dynamics—e.g. process delay d—to be able to give valid predictions but with a larger horizon follows more computational requirements so there is a trade-off that has to be considered. The control signal is kept constant until the next instant when a new calculated control signal is applied.

Another design parameter is the control horizon H_c which limits how many of the first steps in the prediction that the control signal can vary. After the control horizon the control signal is kept constant at the last control signal used in the H_cth step. A small control horizon reduces the amount of possible control sequences and therefore the computation time which yet again is a trade-off. An intuitive way to understand the necessary conditions of these two horizons is [Johansson, 2008]

$$H_p > d, \quad H_c < H_p - d \tag{3.1}$$

The prediction horizon has to be larger than the process delay otherwise the effects of chosen control signals can not be observed in the prediction. The limitation on the control horizon is to make sure that the full effect of the control signal is manifested in the prediction.

MPCtools

There are many different formulations and ways to design an MPC. The MPC design that has been used during the project comes from the Simulink toolbox MPCtools which is discussed more in Chapter 4. However, in this section relevant theory about the mathematical formulation of the MPC design in MPCtools is described. Understanding the design is important for the coming chapters where certain modifications to the system and added adaptive functionality is discussed. More thorough information about the MPC design in MPCtools can be found in the MPCtools reference manual [Åkesson, 2006].

MPCtools MPC Design

The linear model used for prediction are on the form [Åkesson, 2006]

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= C_y x_k \\ z_k &= C_z x_k + D_z u_k \\ z_k^c &= C_c x_k + D_c u_k \end{aligned} \tag{3.2}$$

where k is the time index, x_k is the state vector, u_k is the control signal, y_k is the measured output, z_k is the controlled output, z_k^c is the constrained output and A, B, C_i, D_i are state space matrices. This separation of z_k and z_k^c can have its benefits but for simplicity it is assumed that $z_k = z_k^c$ which is true for the implementations in this project. Part of the optimization problem to be solved by the MPC is the quadratic cost function

$$J_k = \sum_{i=1}^{H_p} ||\hat{z}(k+i|k) - r(k+i|k)||_Q^2 + \sum_{i=0}^{H_c-1} ||\Delta\hat{u}(k+i|k)||_R^2 \tag{3.3}$$

where r is the reference, Q and R are weight matrices, $\hat{z}(k+i|k)$ and $\Delta\hat{u}(k+i|k)$ are the predicted controlled output and control signal increments at time instance $k+i$ calculated in time instance k, respectively. The control signal increment is the the forward difference of the control signal, $\Delta u_k = u_k - u_{k-1}$. The cost function can be rewritten with vectors on the form

$$J_k = ||\mathcal{Z}_k - \mathcal{T}_k||_{\mathcal{Q}}^2 + ||\Delta\mathcal{U}_k||_{\mathcal{R}}^2 \tag{3.4}$$

where

$$\mathcal{Z}_k = \begin{bmatrix} \hat{z}(k+1|k) \\ \vdots \\ \hat{z}(k+H_p|k) \end{bmatrix}, \quad \mathcal{T}_k = \begin{bmatrix} r(k+1|k) \\ \vdots \\ r(k+H_p|k) \end{bmatrix}, \quad \Delta\mathcal{U}_k = \begin{bmatrix} \Delta\hat{u}(k|k) \\ \vdots \\ \Delta\hat{u}(k+H_c-1|k) \end{bmatrix} \tag{3.5}$$

and the weights

$$\mathcal{Q} = \begin{bmatrix} Q & 0 & \dots & 0 \\ 0 & Q & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & Q \end{bmatrix}_{H_p x H_p} , \quad \mathcal{R} = \begin{bmatrix} R & 0 & \dots & 0 \\ 0 & R & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & R \end{bmatrix}_{H_c x H_c} \tag{3.6}$$

With Equation (3.2) a linear expression for $\mathcal{Z}_k$ can be found as

$$\mathcal{Z}_k = \Phi x_k + \Gamma u_{k-1} + \Theta \Delta \mathcal{U}_k \tag{3.7}$$

where

$$\Phi = \begin{bmatrix} C_z \\ C_z A \\ C_z A^2 \\ \vdots \\ C_z A^{H_p} \end{bmatrix}, \quad \Gamma = \begin{bmatrix} D_z \\ C_z B + D_z \\ C_z AB + C_z B + D_z \\ \vdots \\ C_z \sum_{i=0}^{H_p-1} A^i B + D_z \end{bmatrix} = \begin{bmatrix} \Gamma_0 \\ \Gamma_1 \\ \Gamma_2 \\ \vdots \\ \Gamma_{H_p-1} \end{bmatrix} \tag{3.8}$$

$$\Theta = \begin{bmatrix} \Gamma_0 & 0 & 0 & \dots & 0 \\ \Gamma_1 & \Gamma_0 & 0 & \dots & 0 \\ \Gamma_2 & \Gamma_1 & \Gamma_0 & \ddots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \Gamma_{H_c-1} & \Gamma_{H_c-2} & \Gamma_{H_c-3} & \dots & \Gamma_0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \Gamma_{H_p-1} & \Gamma_{H_p-2} & \Gamma_{H_p-3} & \dots & \Gamma_{H_p-H_u+1} \end{bmatrix}_{H_p x H_c} \tag{3.9}$$

By defining some new variables

$$\mathcal{E}_k = \mathcal{T}_k - \Phi x_k - \Gamma u_{k-1} \tag{3.10}$$

$$\mathcal{G}_k = 2\Theta^T \mathcal{Q} \mathcal{E}_k \tag{3.11}$$

$$\mathcal{H} = \Theta^T \mathcal{Q} \Theta + \mathcal{R} \tag{3.12}$$

and inserting 3.7 into 3.4 the cost function can be written as [Åkesson, 2006]

$$J_k = \Delta \mathcal{U}_k^T \mathcal{H} \Delta \mathcal{U}_k - \Delta \mathcal{U}_k \mathcal{G} + \mathcal{E}_k^T \mathcal{Q} \mathcal{E}_k \tag{3.13}$$

3.2 Iterative Learning Control

Iterative learning control (ILC) is a controller that is suitable on systems where a periodic behaviour is desired. Further on it will be described how ILC can be implemented on a controlled closed loop systems. The idea of using an ILC on a controlled closed loop system is that an external control signal is calculated by the ILC, which is then added to the existing reference signal or control signal to compensate for recurrent deviations from the desired reference. The ILC algorithm can be formulated as follows [Norrlöf, 2004]

$$\begin{aligned} y_k(t) &= T_c(q)r(t) + T_c(q)u_k(t) \\ e_k(t) &= r(t) - y_k(t) \\ u_{k+1} &= Q(q)[u_k(t) + L(q)e_k(t)] \end{aligned} \tag{3.14}$$

where T_c is the controlled closed loop system, r is the reference signal, u is the control signal and e is the tracking error. Q and L are filters to be designed. The subscript k denotes the iteration index. Now what is interesting from a design perspective here is the error dynamics

$$e_{k+1}(t) = [(1-Q)(1-T_c)]r(t) + [Q(1-L\cdot T_c)]e_k(t) \tag{3.15}$$

The design goal is to make the error dynamics converge to zero by choosing appropriate filters for Q and L. Worth to be noted is that Q and L can be non-causal. Since the system has a periodic behaviour, values from earlier iterations can be seen used as "future" values.

Heuristic ILC Design

With the Heuristic method not much information about the system model is taken into consideration. Q and L is chosen in a way such that the stability criterion is fulfilled but no real consideration for performance is taken into account [Johansson, 2008].

Algorithm

1. *Choose the Q filter as a low-pass filter with cut-off frequency such that the band-width of the learning algorithm is sufficient.*
2. *Let $L(q) = \kappa q^{\delta}$. Choose κ and δ such that the stability criterion is fullfilled. Normally it suffice to choose δ as the system time delay and $\kappa : 0 < \kappa \leq 1$ to get a stable ILC system.*

Pros and cons The advantage with this method is that it is simple and not much knowledge about the system is needed. But this comes with a price in terms of ability to improve performance.

4

Simulation

The engine simulation is supposed to simulate a combustion engine with a four stroke cylinder cycle. The simulation is built on the models from Chapter 2 and is as mentioned implemented in Simulink. A number of assumptions and estimations had to be made for the simulation to run smoothly. These assumptions and estimations and the overall setup of the simulation will be discussed in this section.

4.1 Software

All code in this project has been written in the programming language MATLAB and the simulations have been implemented in Simulink. MATLAB and Simulink are both products from MathWorks who is one of the leading corporations in developing mathematical computing software. MATLAB and Simulink goes hand in hand and incorporating both these software gives the opportunity to combine both textual and graphical programming. The simulation of the system dynamics, models from Chapter 2 were used and the MPC controllers were implemented using an already existing Simulink toolbox for setting up MPC controller blocks. The toolbox is MPCtools created by Johan Åkesson from the Department of Automatic Control at Lund University [Åkesson, 2006].

MPCtools

As mentioned MPCtools is a toolbox that has finished codes for initializing and implementing MPC controllers ready to be used as blocks in Simulink. The controllers created in MPCtools are linear MPCs which uses linear state space models for prediction. The problems are set up as quadratic cost functions and are solved with one of two QP solvers available. There are more features in the toolbox but these were not used.

Adaptive MPC Implementation

Because of the non-linear nature of the engine a solely linear MPC can have bad or even unstable performance, therefore an adaptive functionality was implemented.

The adaptivity is based on linearizing the system used in the MPC every cycle. The linearization was made in a separate block which then sends new state space matrices to the MPC block created with MPCtools. In both MPC implementations it is only the B matrix in the state space representations that varies when the system is linearized and all other matrices stay the same. Introduced in Section 3.1, the prediction matrices Γ, Θ and the defined matrices $\mathcal{E}_k$, $\mathcal{G}_k$ and $\mathcal{H}$ are the ones affected by a change in B. By updating these matrices each cycle with the new linearized B matrix adaptivity is achieved.

4.2 Engine

The parameter values used in this simulation are based on values from a Scania D13 six-cylinder heavy-duty diesel engine which is the engine used in [Turesson, 2018] where values of the empirical parameters are found. This engine is used in one of the test cells in the combustion lab at Lund University and in Table 4.1 parameter values from the engine are shown.

Table 4.1: Technical specification

Crank radius	0.080	$[m]$
Connecting rod length	0.2550	$[m]$
Compression ratio	18	[-]
Bore	0.130	$[m]$
Displacement volume	0.0021	$[m^3]$
Clearance volume	$1.2490 \cdot 10^{-4}$	$[m^3]$

In the engine in the lab a cooled low-pressure EGR (Exhaust gas recirculation) and air path has been added to the intake manifold. This added path is used in the controller and simulation in Chapter 5.

4.3 Pressure

The simulated pressure is calculated from Equation (2.18) and as a reminder is defined as

$$\frac{dp}{d\theta} = -\frac{\gamma}{V}\frac{dV}{d\theta}p + \frac{\gamma - 1}{V}\left(\frac{dQ_c}{d\theta} - \frac{dQ_{ht}}{d\theta}\right) \tag{4.1}$$

The pressure is updated every time step, variables $V(\theta)$ and $\frac{dV(\theta)}{d\theta}$ are known from the geometry properties of the cylinder and γ is assumed constant. $dQ_{ht}/d\theta$ is modeled with Equation (2.21) and $dQ_c/d\theta$ has to be calculated before each cycle. In the simulation the pressure is calculated by integrating $dp/d\theta$ but since $dp/d\theta$ is dependent on p Equation (4.3) is modified such that p is delayed one sample

$$\frac{dp(\theta)}{d\theta} = -\frac{\gamma}{V(\theta)}\frac{dV(\theta)}{d\theta}p(\theta-\theta_{res}) + \frac{\gamma-1}{V(\theta)}\left(\frac{dQ_c(\theta)}{d\theta} - \frac{dQ_{ht}(\theta)}{d\theta}\right) \tag{4.2}$$

To make sure the simulation has smooth transitions between cycles, the pressure is implemented to end and start each cycle at a constant value, p_{in}. The $dp/d\theta$ model is not perfect in this sense and if the pressure is not set to a constant value the integral is affected and cause integral windup problem. For simplicity the intake pressure and exhaust pressure is assumed to be the same and therefore the pressure can start and end each cycle at the same value. The method used to implement this was to let the pressure decrease as a sinus curve for a set amount of samples after EVO. This approach is very simple but is believed to be enough for control purposes [Königsson, 2010]. In Figure 4.1 simulated result and experimental data is compared.

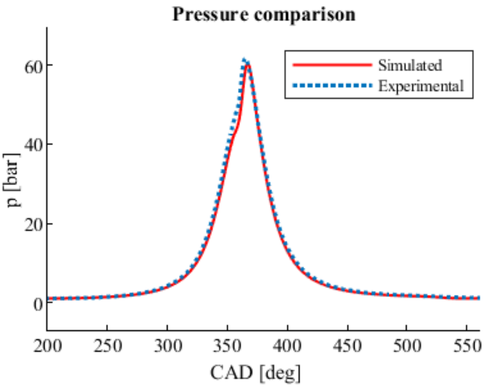


Figure 4.1: Results from simulated pressure compared with experimental data.

4.4 Heat Release Rate

The pre-calculated heat release rate is calculated with a Wiebe function each cycle. Two options are available, either set parameters in the Wiebe function by hand or use a Wiebe estimation on experimental data to set the parameters. Equation (2.19) is used to calculate the heat release rate of the experimental data. The calculated result is then filtered for better approximation.

Wiebe estimation

Wiebe estimation is used to estimate the heat release, Q_c, as a so called Wiebe function which has the form [Abbaszadehmosayebi, 2014]

$$\frac{Q_c(\theta)}{Q_{tot}} = \begin{cases} 1 - \exp(-a(\frac{\theta - \theta_{SOC}}{\theta_{CD}})^{b+1}), & \theta \geq \theta_{SOC} \\ 0, & else \end{cases} \tag{4.3}$$

where Q_{tot} is the total accumulated energy from the combustion, a and b are empirical constants and θ_{CD} is the combustion duration. To go from experimental Q_c to the estimation a, b and θ_{SOC} are assumed to be known. Common values for the empirical values are $a = 6.9$ and $b = 2$. Q_{tot} and θ_{CD} are then found using a nonlinear least-square solver in Matlab which fits the Wiebe function to the experimental data by minimizing the Euclidean norm of the error. This estimated heat release is used as the initial heat release of the simulation. From Equation (4.3) an analytical solution for $dQ_c/d\theta$ can be found

$$\frac{1}{Q_{tot}} \frac{dQ_c(\theta)}{d\theta} = \begin{cases} \frac{a(b+1)}{\theta_{CD}} (\frac{\theta - \theta_{SOC}}{\theta_{CD}})^b \exp(-a(\frac{\theta - \theta_{SOC}}{\theta_{CD}})^{b+1}), & \theta \geq \theta_{SOC} \\ 0, & else \end{cases} \tag{4.4}$$

With this equation the heat release rate can be calculated each time sample. The start of combustion is calculated as

$$\theta_{SOC} = \theta_{SOI} + t_c \tau \tag{4.5}$$

and Q_{tot} is modelled with a proportional relation between the lower heat value of the fuel and the injected fuel mass and the combustion efficiency is assumed to be 100 % which results in

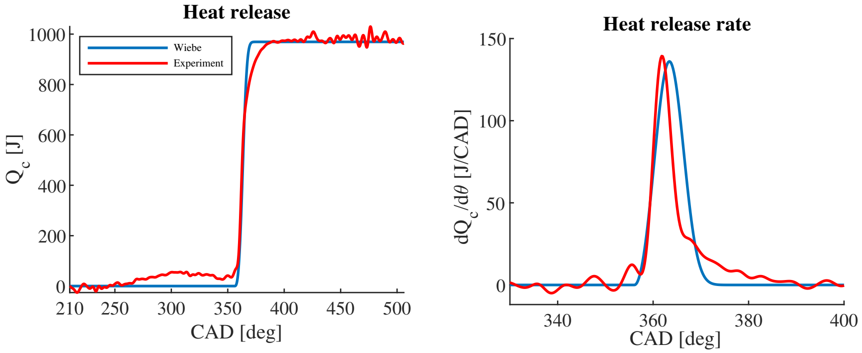


Figure 4.2: Estimated heat release and heat release rate compared to experimental data. The estimated $\hat{Q}_{tot}$ and $\hat{\theta}_{CD}$ is 969.1956 J and 15.9324 CAD.

$$Q_{tot} = m_f \cdot Q_{LHV} = m_f \cdot 44.2 \cdot 10^6 \tag{4.6}$$

With these equations the heat release rate is given a relation to the main control signals m_f and θ_{SOI}.

4.5 Ignition delay

Equation (2.30) has τ on both the left and right hand side of the equal sign. Therefore, the ignition delay from last cycle is used to calculate the next cycle ignition delay

$$\tau = A_t \exp\left(\frac{E_a}{\frac{\tilde{R}}{\tau_0}\int_{\theta_{SOI}}^{\theta_{SOI}+\tau_0} T_{IVC}\left(\frac{V_{IVC}}{V(\theta)}\right)^{\gamma-1} d\theta}\right)\left(\frac{1}{\tau_0}\int_{\theta_{SOI}}^{\theta_{SOI}+\tau_0} \frac{[O_2]_{IVC}}{V(\theta)} V_{IVC} d\theta\right)^{\alpha} \tag{4.7}$$

where τ_0 is the ignition delay from last cycle. $[O_2]_{IVC}$ and T_{IVC} are found using static models shown in Figure 5.1 The way the ignition delay is implemented in the simulation is that the heat release is placed such that θ_{SOC} coincides with TDC. Then the ignition delay is calculated and then the whole heat release rate vector is shifted $\theta_{SOI} + t_c\tau$ CAD from TDC.

4.6 Heat transfer

The heat transfer rate is modeled as Equation (2.21). Values used for the different parameters are shown in Table 4.2. The wall temperature is assumed to be constant and and the motored pressure is estimated and will be more discussed in the next section.

Table 4.2: Parameter values

C_1	2.28	[-]
C_2	0.0032	$[\frac{m}{s \cdot K}]$
N	1200	$[\frac{rev}{min}]$
T_w	365	$[K]$
T_{IVC}	303	$[K]$
V_{IVC}	0.0022	$[m^3]$
p_{IVC}	1.0170	[bar]

Estimated motored pressure

The heat transfer is dependent on the average cylinder gas velocity which is defined in Equation (2.23). As can be seen in that equation the average cylinder gas velocity is dependent on the motored pressure. The motored pressure can either be measured in advance or if only experimental data with combustion is available the motored pressure can be estimated. The motored pressure is estimated with pressure data by assuming it is an isentropic process

$$p_m V^\gamma = constant \tag{4.8}$$

Before combustion the experimental pressure is assumed to be equal to the motored pressure. By choosing a small section of the pressure curve in that area the motored pressure for the whole cycle can be estimated. This chosen section of the pressure is used to fit Equation (4.8). Different notations are used since the fitted values might not represent the real parameters

$$p_m V^\kappa = C \tag{4.9}$$

With V and selected section of p known, the value of κ and C can be estimated by taking the logarithm and then perform a linear regression

$$\log(p_m) + \kappa \log(V) = \log(C) \tag{4.10}$$

4.7 Discussion

A simulation based on these simplified models is far from reflecting reality. Much more complex and computation heavy models are need to give a good representation of the engine. To have the heat release rate shaped by a Wiebe function is big a simplification where a lot of dynamics are lost. As can be seen in Figure 4.2 the heat release rate can not be fully estimated and the heat release is not always symmetric. Therefore this simulation is very limited to how the heat release can be shaped but this estimation is a decent way to simulate the basic combustion dynamics of the engine. The main reason for the difference in heat release compared to the estimation is model errors, a Wiebe function is still considered a good estimation though. One other thing to take into consideration is that the data is taken from experiments with two injections, one pilot and one main injection which can lead to either one or two combustion stages and the Wiebe function estimates a single combustion event. The experimental data looks most certainly like one combustion stage and it was decided that this data could still be used.

The models calculating the pressure comes from assumptions of ideal conditions which is also not true in reality. The discretization of the pressure rate function in Equation (4.2) is also surly to have an impact but by comparing experimental data and simulated results in Figure 4.1 it is believed to be good enough for control purposes. The assumption that γ is constant is also something that has room for improvement where a simple but more accurate model might be applicable.

Heat transfer models are most often not very accurate but Woschni's model has been proven to work in control applications [Turesson, 2018] and is a good choice for its simplicity. But for further improvements more accurate models could make a big difference and also if possible use a measured motored pressure instead of estimated.

The ignition delay model has also been proven to be pretty accurate but not perfect. Variables $[O_2]_{IVC}$ and T_{IVC} are usually measured but here static models are used. These static models can change over time and has to be found experimentally therefore it is needed to update these for better results. Best option would be to not rely on static models and have a continuous model instead.

Overall this simulation loses a lot of real life dynamics but the most essential dynamics required works good enough for controlling purposes. This simulation can be a good option for a first phase of a new controller to get an idea of its performance. The simulation is not good enough for tuning purposes or determining robustness.

5

Ignition Delay MPC

This less advanced controller is introduced in [Turesson, 2018] and was chosen as test object for investigating how well MPCtools works with the rest of the simulation and to test the adaptive feature added to the MPC. Both a linear MPC and adaptive MPC is implemented and compared. This MPC regulates the ignition delay and combustion timing to follow a reference by computing the fuel-injection timing and valve positions in the gas-exchange system. The combustion timing is defined here as the crank angle where 50 % of the total heat is released, θ_{50}. The models used originates from the models in Section 2.4 and to relate the outputs to the control signals partial derivatives are approximated.

5.1 Ignition Delay Model

The non-linear model in Equation (2.30) is the base for the linearized model. Here is a reminder

$$\tau = A_t \exp\left(\frac{E_a}{\frac{\tilde{R}}{\tau} \int_{\theta_{SOI}}^{\theta_{SOI}+\tau} T_{IVC} \left(\frac{V_{IVC}}{V(\theta)}\right)^{\gamma-1} d\theta} \right) \left(\frac{1}{\tau} \int_{\theta_{SOI}}^{\theta_{SOI}+\tau} \frac{[O_2]_{IVC}}{V(\theta)} V_{IVC} d\theta \right)^{\alpha} \tag{5.1}$$

From this model the partial derivatives with respect to θ_{SOI}, T_{IVC} and $[O_2]_{IVC}$ can be approximated as [Turesson, 2018]

$$\frac{\partial \tau}{\partial \theta_{SOI}} \approx \frac{\tau(\theta_{SOI} + \frac{\Delta\theta_{SOI}}{2}) - \tau(\theta_{SOI} - \frac{\Delta\theta_{SOI}}{2})}{\Delta\theta_{SOI}} \tag{5.2}$$

$$\frac{\partial \tau}{\partial T_{IVC}} \approx \frac{\tau(T_{IVC} + \frac{\Delta T_{IVC}}{2}) - \tau(T_{IVC} - \frac{\Delta T_{IVC}}{2})}{\Delta T_{IVC}} \tag{5.3}$$

$$\frac{\partial \tau}{\partial [O_2]_{IVC}} \approx \frac{\tau([O_2]_{IVC} + \frac{\Delta[O_2]_{IVC}}{2}) - \tau([O_2]_{IVC} - \frac{\Delta[O_2]_{IVC}}{2})}{\Delta[O_2]_{IVC}} \tag{5.4}$$

With these derivatives a linearized model of the ignition delay can be written as

$$\tau(k+1) = \tau(k) + \left[\tfrac{\partial\tau}{\partial\theta_{SOI}}, \quad \tfrac{\partial\tau}{\partial T_{IVC}}, \quad \tfrac{\partial\tau}{\partial[O_2]_{IVC}}\right] \begin{bmatrix} \Delta\theta_{SOI} \\ \Delta T_{IVC} \\ \Delta[O_2]_{IVC} \end{bmatrix} \tag{5.5}$$

where k is the current cycle. θ_{50} is modeled as

$$\theta_{50}(k+1) = \theta_{50}(k) + \Delta\theta_{SOI} + t_c\left[\tfrac{\partial\tau}{\partial\theta_{SOI}}, \quad \tfrac{\partial\tau}{\partial T_{IVC}}, \quad \tfrac{\partial\tau}{\partial[O_2]_{IVC}}\right] \begin{bmatrix} \Delta\theta_{SOI}(k) \\ \Delta T_{IVC}(k) \\ \Delta[O_2]_{IVC}(k) \end{bmatrix} \tag{5.6}$$

where t_c is a conversion variable from milliseconds to CAD.

5.2 Gas-Exchange System

In the gas-exchange system there are three valves of interest. The first valve is θ_{hot} which determine how much air is going from the turbocharger directly into the intake manifold. The two last valves of interest are the ones regulating the high and low pressure EGR paths θ_{HP} and θ_{LP}, respectively. To relate the valve positions to T_{IVC} and $[O_2]$, static models of how these valves affect these values derived from experiments by [Turesson, 2018] were used. These static models are found in Figure 5.1. From the static functions the partial derivatives $\partial T_{IVC}/\partial\theta_{hot}$, $\partial[O_2]_{IVC}/\partial\theta_{HP}$ and $\partial[O_2]_{IVC}/\partial\theta_{LP}$ can be calculated and results in the following relation

$$\frac{\partial\tau}{\partial\theta_{hot}} = \frac{\partial\tau}{\partial T_{IVC}}\frac{\partial T_{IVC}}{\partial\theta_{hot}} \tag{5.7}$$

$$\frac{\partial\tau}{\partial\theta_{HP}} = \frac{\partial\tau}{\partial[O_2]_{IVC}}\frac{\partial[O_2]_{IVC}}{\partial\theta_{HP}} \tag{5.8}$$

$$\frac{\partial\tau}{\partial\theta_{LP}} = \frac{\partial\tau}{\partial[O_2]_{IVC}}\frac{\partial[O_2]_{IVC}}{\partial\theta_{LP}} \tag{5.9}$$

These partial derivatives can be used to relate the control signals to the control variables

$$\tau(k+1) = \tau(k) + \left[\tfrac{d\tau}{d\theta_{SOI}}, \quad \tfrac{d\tau}{d\theta_{hot}}, \quad \tfrac{d\tau}{d\theta_{HP}}, \quad \tfrac{d\tau}{d\theta_{LP}}\right] \begin{bmatrix} \Delta\theta_{SOI}(k) \\ \Delta\theta_{HP}(k) \\ \Delta\theta_{HP}(k) \\ \Delta\theta_{LP}(k) \end{bmatrix} \tag{5.10}$$

$$\theta_{50}(k+1) = \theta_{50}(k) + t_c\left[\tfrac{1}{t_c} + \tfrac{d\tau}{d\theta_{SOI}}, \quad \tfrac{d\tau}{d\theta_{hot}}, \quad \tfrac{d\tau}{d\theta_{HP}}, \quad \tfrac{d\tau}{d\theta_{LP}}\right] \begin{bmatrix} \Delta\theta_{SOI}(k) \\ \Delta\theta_{HP}(k) \\ \Delta\theta_{HP}(k) \\ \Delta\theta_{LP}(k) \end{bmatrix} \tag{5.11}$$

Now the linearized model is on a form that can be used in the MPC model in Equation (3.2).

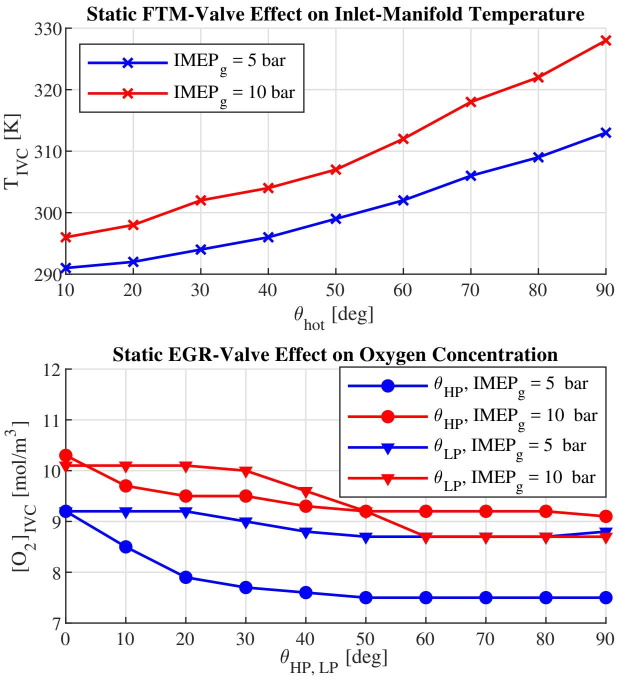


Figure 5.1: Experimental data found in [Turesson, 2018] to describe the static gas-exchange system models.

5.3 MPC Formulation

The true control signal sent into the engine is denoted

$$u_k = \begin{bmatrix} \theta_{SOI}(k) \\ \theta_{hot}(k) \\ \theta_{HP}(k) \\ \theta_{LP}(k) \end{bmatrix} \tag{5.12}$$

but since the models in Equation (5.10) and (5.11) are dependent on Δu some modifications to the system has to be made. The calculated control signal from the MPC will be Δu which must be added to the current true control signal separately outside of the MPC. Thus to handle constraints on the true control signal, the true control signal must be measured and will then become states of the system. With this in mind combined with the models in Equation (5.10) and (5.11), the state space representation of the system takes the form

$$x_{k+1} = x_k + \mathbf{B}\Delta u_k \tag{5.13}$$

$$x_k = \begin{bmatrix} \tau(k) \\ \theta_{50}(k) \\ \theta_{SOI}(k) \\ \theta_{hot}(k) \\ \theta_{HP}(k) \\ \theta_{LP}(k) \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \frac{d\tau}{d\theta_{SOI}} & \frac{d\tau}{d\theta_{hot}} & \frac{d\tau}{d\theta_{HP}} & \frac{d\tau}{d\theta_{LP}} \\ t_c(\frac{1}{t_c} + \frac{d\tau}{d\theta_{SOI}}) & t_c\frac{d\tau}{d\theta_{hot}} & t_c\frac{d\tau}{d\theta_{HP}} & t_c\frac{d\tau}{d\theta_{LP}} \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.14}$$

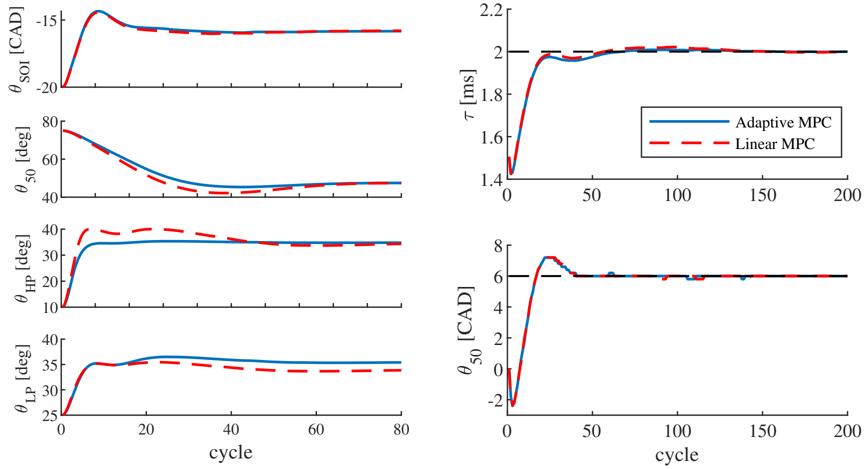
The cost function is set up such that a reference errors in θ_{50} and τ will be minimized, weights on θ_{HP}, θ_{LP} and $\Delta^2 u$ are only used for smoother control. Here $\Delta^2 u$ is defined as $\Delta u_{k+1} - \Delta u_k$. This value is used because of how a MPC controller is implemented in MPCtools. In Equation (3.4) the cost function that is solved by the controller is described and is dependant directly on the increment of the calculated control signal. The system model used in this ignition delay controller calculates the increment of the real control signal and with this follows that the cost function that is minimized takes $\Delta^2 u$ into consideration instead of Δu. The MPC problem is therefore formulated as

$$\begin{aligned} \min_{\substack{\Delta\theta_{SOI},\Delta\theta_{hot}, \\ \Delta\theta_{HP},\Delta\theta_{LP}}} \quad & \sum_{k=1}^{H_p} \big(Q_{11}|\theta_{50}^r(k) - \theta_{50}(k)|^2 + Q_{22}|\tau^r(k) - \tau(k)|^2 \\ & + Q_{55}|\theta_{HP}(k)|^2 + Q_{66}|\theta_{LP}(k)|^2\big) + \\ & \sum_{k=0}^{H_c-1} R_{11}|\Delta^2\theta_{SOI}(k)|^2 + R_{22}|\Delta^2\theta_{hot}(k)|^2 \\ & + R_{33}|\Delta^2\theta_{HP}(k)|^2 + R_{44}|\Delta^2\theta_{LP}(k)|^2 \\ \text{subject to} \quad & \theta_{min} \le \begin{pmatrix} \theta_{SOI} \\ \theta_{hot} \\ \theta_{HP} \\ \theta_{LP} \end{pmatrix} \le \theta_{max}, \quad \Delta u_{min} \le \begin{pmatrix} \Delta\theta_{SOI} \\ \Delta\theta_{hot} \\ \Delta\theta_{HP} \\ \Delta\theta_{LP} \end{pmatrix} \le \Delta u_{max}, \\ & x_{k+1} = x_k + \mathbf{B}\Delta u_k \end{aligned}$$

Adaptive and linear MPC comparison

Simulation results when using both a linear and an adaptive MPC are shown in Figure 5.2 and 5.3 where the difference is the references. The linear model chosen for the linear MPC was taken in steady state around the reference point in Figure 5.2. The models calculated in the separate linearizing block used by the adaptive MPC. When comparing the results from Figure 5.2 and 5.3 it is shown that there is no significant difference in convergence speed between the linear and the adaptive MPC. Even when the linear model does not match the steady state as in Figure 5.3 it converges. Using the adaptive MPC does not require any extra work with finding a linear model in before using the controller and will have better performance if the linear model is poorly chosen.

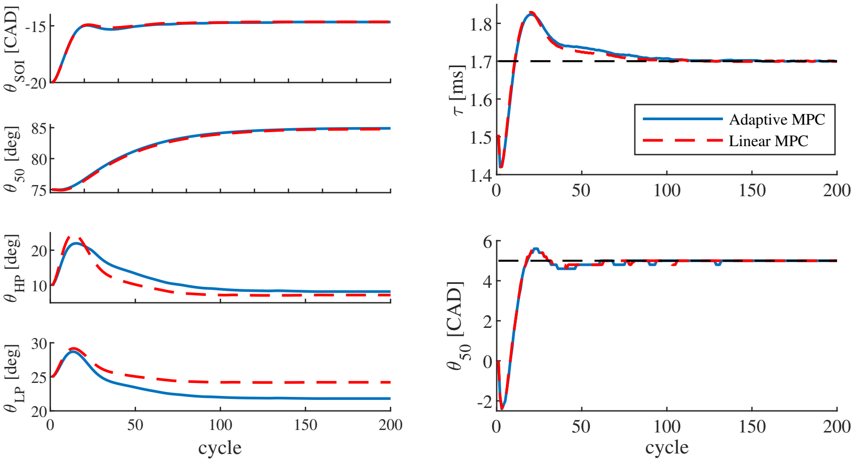
All results show that the simulation works well with controllers created with the MPCtools toolbox and controllers with the adaptive functionality also gives desirable results. From this the adaptive MPC controller is ready to be tested on the more advanced pressure predictive MPC.



(a) Control signals.

(b) Outputs.

Figure 5.2: These figures compares the simulated results when using an adaptive MPC and a pure linear MPC. Blue line shows the adaptive MPC signals, dashed red line show linear MPC signals and the dashed black line in Figure 5.2b shows the reference. The references are 2 ms and 6 CAD from TDC for the ignition delay and θ_{50}, respectively



(a) Control signals.

(b) Outputs.

Figure 5.3: These figures compares the simulated results when using an adaptive MPC and a pure linear MPC. Blue line shows the adaptive MPC signals, dashed red line show linear MPC signals and the dashed black line in Figure 5.3b shows the reference. The references are 1.7 ms and 5 CAD from TDC for the ignition delay and θ_{50}, respectively

6

Pressure Prediction MPC

In this chapter the structure, results and discussion about a pressure prediction MPC controller will be presented. The idea is to set up a linear system that predicts the pressure in the next cycle and then using this system to implement a MPC that tracks an ideal pressure reference. The control signals used are the start of injection and the injected fuel mass. These control signals are used for their association to the heat release rate. The adapted feature discussed in Section 4.1 is used to linearize the system and update the MPC every cycle. The reason for using this controller, apart from the advantages discussed in [Turesson, 2018], is that with this controller an iterative behaviour occurs. The idea is to investigate if this iterative behaviour can be exploited with ILC to compensate for flaws in the pressure predictive controller.

6.1 Relation between P and Δu

To find the relation between deviation in heat release and deviation in pressure a linearized model of Equation (2.18) combined with Equation (2.21) will be used. With this the Equation (2.18) can be formulated as

$$\frac{dp}{d\theta} = -\frac{\gamma}{V}\frac{dV}{d\theta}p + \frac{\gamma-1}{V}\left(\frac{dQ_c}{d\theta} - \frac{h_c A}{6N}(T - T_w)\right) \tag{6.1}$$

To simplify the linearization both γ and T_w are assumed constant. Linearizing this equation with respect to p and $dQ_c/d\theta$ then results in the following equation [Turesson, 2018]

$$\frac{d\Delta p}{d\theta} = -\left(\frac{\gamma}{V}\frac{dV}{d\theta} + \frac{d\mu(p_0,\theta)}{dp}\right)\Delta p + \frac{\gamma-1}{V}\frac{d\Delta Q_c}{d\theta} \tag{6.2}$$

where p_0 is the pressure from previous cycle and μ is defined as

$$\mu(p,\theta) = (\gamma-1)\frac{h_c A T_{IVC}}{6N p_{IVC} V_{IVC}} p \tag{6.3}$$

h_c will be defined here as

$$h_c = \alpha B^{0.2} p^{0.8} T^{-0.55} w^{0.8} \tag{6.4}$$

where α is used as a tuning parameter otherwise it follows the Woschni correlation. Differential equation (6.2) can be solved with integrating factor and results in

$$\Delta p(\theta) = \int_{\theta_{IVC}}^{\theta} \Phi(\theta, \vartheta)\Gamma(\theta)\frac{d\Delta Q_c(\vartheta)}{d\theta} d\vartheta \tag{6.5}$$

where

$$\begin{aligned} \Phi(\theta, \vartheta) =& \exp\left(-\int_{\vartheta}^{\theta} \frac{d\mu(p_0, \phi)}{dp} d\phi\right) V(\vartheta)^{\gamma-1} \\ \Gamma(\theta) =& \frac{\gamma-1}{V(\theta)^{\gamma}} \end{aligned} \tag{6.6}$$

$d\mu/dp$ is calculated using the forward Euler method and the whole expression of 6.5 is approximated as a sum of discrete values multiplied with the CAD resolution

$$\begin{aligned} \Delta p(\theta) \approx& \theta_{res} \sum_{k=\theta_{IVC}}^{\theta} \exp\left(-\sum_{l=k}^{\theta} \frac{d\mu(p_0, l)}{dp}\right) V(k)^{\gamma-1} \frac{\gamma-1}{V(\theta)^{\gamma}} \frac{d\Delta Q_c(k)}{d\theta} \\ =& \theta_{res} \sum_{k=\theta_{IVC}}^{\theta} \Phi_d(\theta, k)\Gamma(\theta)\frac{d\Delta Q_c(k)}{d\theta} \end{aligned} \tag{6.7}$$

This can be writing on vector form

$$\Delta p(\theta_n) = \theta_{res}\Gamma(\theta_n)\begin{bmatrix}\Phi_d(\theta_n, \theta_1), & \Phi_d(\theta_n, \theta_2), & \dots & \Phi_d(\theta_n, \theta_n)\end{bmatrix} \begin{bmatrix} \frac{d\Delta Q_c(\theta_1)}{d\theta} \\ \frac{d\Delta Q_c(\theta_2)}{d\theta} \\ \vdots \\ \frac{d\Delta Q_c(\theta_n)}{d\theta} \end{bmatrix} \tag{6.8}$$

and further expanding this to matrix form results in

$$\begin{bmatrix} \Delta p(\theta_1) \\ \Delta p(\theta_2) \\ \Delta p(\theta_3) \\ \vdots \\ \Delta p(\theta_n) \end{bmatrix} = \begin{bmatrix} b_{11} & 0 & 0 & \dots & 0 \\ b_{21} & b_{22} & 0 & \dots & 0 \\ b_{31} & b_{32} & b_{33} & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 \\ b_{n1} & b_{n2} & b_{n3} & \dots & b_{nn} \end{bmatrix} \begin{bmatrix} \frac{d\Delta Q_c(\theta_1)}{d\theta} \\ \frac{d\Delta Q_c(\theta_2)}{d\theta} \\ \frac{d\Delta Q_c(\theta_3)}{d\theta} \\ \vdots \\ \frac{d\Delta Q_c(\theta_n)}{d\theta} \end{bmatrix} = \mathbf{B}_d \frac{d\Delta Q_c}{d\theta} \tag{6.9}$$

where

$$b_{ij} = \theta_{res}\Gamma(\theta_i)\Phi_d(\theta_i, \theta_j) \tag{6.10}$$

With these results the state space representation for predicting the pressure in the next cycle is as follows

$$\begin{aligned} P_{k+1} &= P_k + \mathbf{B}_d \frac{d\Delta Q_c}{d\theta} \\ y_k &= P_k \end{aligned} \tag{6.11}$$

Next is to establish a relation between $\frac{d\Delta Q_c}{d\theta}$ and the control signals. The control signals are the mass of fuel injected, m_f, and the start of injection, θ_{SOI} and is denoted

$$u_k = \begin{bmatrix} m_f \\ \theta_{SOI} \end{bmatrix}_k \tag{6.12}$$

The assumption is that a variation in m_f will affect the total energy added and a variation in θ_{SOI} is a shift in the heat release. This is illustrated in figure 6.1. From the assumption that the mass only affects the accumulated heat release the partial derivative [Turesson, 2018]

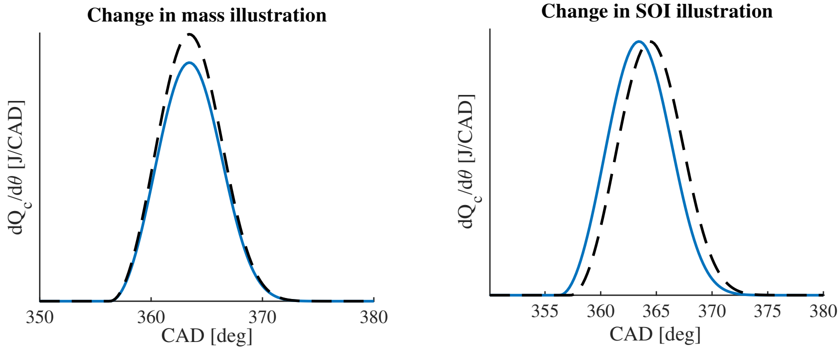


Figure 6.1: Left and right figure illustrates the assumption of how $\frac{d\Delta Q_c}{d\theta}$ is affected by an increase in m_f and a later θ_{SOI}, respectively. Blue line represents current $\frac{d\Delta Q_c}{d\theta}$ and the dashed line represents next cycle.

$$\frac{\partial}{\partial m_f}\frac{dQ_c}{d\theta} = \frac{Q_{LHV}}{\int dQ_c}\frac{dQ_c}{d\theta} \tag{6.13}$$

is given. The shift in heat release is described with the partial derivatives

$$\frac{\partial}{\partial \theta_{SOI}}\frac{dQ_c}{d\theta} = -\frac{d^2Q_c}{d\theta^2} \tag{6.14}$$

$$\frac{\partial \theta_{CT}}{\partial \theta_{SOI}} = 1 \tag{6.15}$$

where θ_{CT} is the combustion timing. Equation (6.15) says that a change in θ_{SOI} results in an equal change in the combustion timing. With Equation (6.13) and (6.14) a gradient can be set up

$$\nabla \frac{dQ_c(\theta)}{d\theta} = \begin{bmatrix} \frac{Q_{LHV}}{\int dQ_c} \frac{dQ_c(\theta)}{d\theta}, & -\frac{d^2 Q_c(\theta)}{d\theta^2} \end{bmatrix} \tag{6.16}$$

and from this the relation between $\frac{d\Delta Q_c}{d\theta}$ and Δu follows

$$\frac{d\Delta Q_c}{d\theta} = \begin{bmatrix} \nabla \frac{dQ_c(\theta_1)}{d\theta} \\ \nabla \frac{dQ_c(\theta_2)}{d\theta} \\ \vdots \\ \nabla \frac{dQ_c(\theta_n)}{d\theta} \end{bmatrix} \Delta u = \nabla \frac{dQ_c}{d\theta} \Delta u \tag{6.17}$$

Inserting (6.17) in Equation (6.11) gives the state space representation

$$\begin{aligned} P_{k+1} &= P_k + \mathbf{B}_d \nabla \frac{dQ_c}{d\theta} \Delta u_k \\ y_k &= P_k \end{aligned} \tag{6.18}$$

which can be implemented in the MPC. In Figure 6.2 the performance of the linear pressure prediction model is shown. As can be seen the linear model almost predicts the pressure for the next cycle.

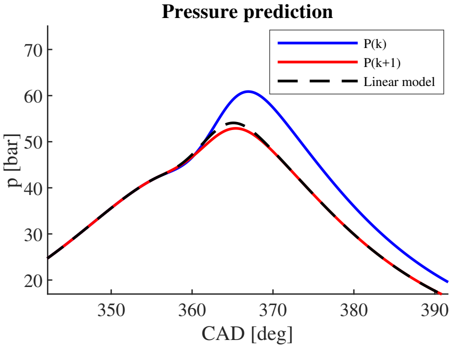


Figure 6.2: Pressure prediction results.

6.2 MPC Formulation

The model in Equation (6.18) is used for prediction but since changes in the control signal is used and not the accumulated signal, some alterations has to be made for the MPC to be able to take constraints on m_f and θ_{SOI} into consideration. The calculated control signal from the MPC will be Δu which must be added to the current control signal separately outside of the MPC. In this controller the control horizon is equal to the prediction horizon which gives the opportunity to measure Δu as a state and minimize this in the cost function instead of $\Delta^2 u$. Both the true control signal u and Δu is measured and becomes states of the system which takes the form

$$x_k = \begin{bmatrix} P_k \\ u_k \\ \Delta u_k \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} \mathbf{I}_{(n+2)} & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{B}_d \nabla \frac{dQ_c}{d\theta} \\ \mathbf{I}_{2x2} \\ \mathbf{I}_{2x2} \end{bmatrix} \tag{6.19}$$

$$x_{k+1} = \mathbf{A} x_k + \mathbf{B} \Delta u_k \tag{6.20}$$

To achieve pressure tracking the cost function is setup in two parts, one is to minimize the error between the pressure reference and the pressure and the second part is to lessen the changes of the control signal. The complete MPC problem formulation results in

$$\min_{\Delta m_f, \Delta\theta_{SOI}} \quad \sum_{k=1}^{H_p} q||P_k^r - P_k||_2^2 + \sum_{k=0}^{H_c-1} \Delta u_k^T \mathbf{R} \Delta u_k \tag{6.21a}$$

$$\text{subject to} \quad y_{min} \leq \begin{pmatrix} P \\ m_f \\ \theta_{SOI} \end{pmatrix} \leq y_{max}, \quad \Delta u_{min} \leq \begin{pmatrix} \Delta m_f \\ \Delta\theta_{SOI} \end{pmatrix} \leq \Delta u_{max}, \tag{6.21b}$$

$$x_{k+1} = \mathbf{A} x_k + \mathbf{B} \Delta u_k \tag{6.21c}$$

Scaling

When implementing this controller directly using MPCtools matlab will run into numerical errors. This is because of the difference in scaling of the control signals are too large and no feasible solution could be found. m_f and θ_{SOI} is of size 10^{-6} and 10, respectively. Therefore a scaling matrix u_t was introduced

$$u_t = \begin{bmatrix} 10^{-6} & 0 \\ 0 & 10 \end{bmatrix} \tag{6.22}$$

and to achieve same dynamics as in Equation (6.21) but with scaling the system model had to be reformulated. The new state space becomes

$$\hat{x}_k = \begin{bmatrix} P_k \\ u_t^{-1} u_k \\ u_t^{-1} \Delta u_k \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} \mathbf{I}_{(n+2)} & 0 \\ 0 & 0 \end{bmatrix}, \quad \hat{\mathbf{B}} = \begin{bmatrix} \mathbf{B}_d u_t \nabla \frac{dQ_c}{d\theta} \\ \mathbf{I}_{2x2} \\ \mathbf{I}_{2x2} \end{bmatrix} \tag{6.23}$$

$$x_{k+1} = \mathbf{A} x_k + \hat{\mathbf{B}} \Delta \hat{u}_k \tag{6.24}$$

where $\hat{u}_k$ is the calculated control signal which is proportionally related to the desired control signal $u_k \propto u_t \hat{u}_k$. u_t is multiplied to $\hat{u}_k$ outside the controller block the same is done for the measured states x_k, u_t^{-1} is multiplied to u_k and Δu_k outside the controller block. With scaling the final formulation results in

$$\min_{\Delta\hat{m}_f, \Delta\hat{\theta}_{SOI}} \quad \sum_{k=1}^{H_p} q||P_k^r - P_k||_2^2 + \sum_{k=0}^{H_c-1} \Delta\hat{u}_k^T \hat{\mathbf{R}} \Delta\hat{u}_k \tag{6.25a}$$

$$\text{subject to} \quad \hat{y}_{min} \leq \begin{pmatrix} P \\ \hat{m}_f \\ \hat{\theta}_{SOI} \end{pmatrix} \leq \hat{y}_{max}, \quad \Delta\hat{u}_{min} \leq \begin{pmatrix} \Delta\hat{m}_f \\ \Delta\hat{\theta}_{SOI} \end{pmatrix} \leq \Delta\hat{u}_{max}, \tag{6.25b}$$

$$\hat{x}_{k+1} = \mathbf{A}\hat{x}_k + \hat{\mathbf{B}}\Delta\hat{u}_k \tag{6.25c}$$

where

$$\hat{\mathbf{R}} = (u_t^{-1})^T \mathbf{R} (u_t^{-1}), \quad \hat{y}_i = (u_t^{-1}) y_i, \quad \hat{u}_i = (u_t^{-1}) u_i \tag{6.26}$$

6.3 Reference trace

The pressure reference is based on two ideal pressure traces for an isentropic process, a process where energy is only transferred as work and there is no transfer of heat or matter. These ideal pressure traces are called ideal isochoric cycle and ideal isobaric cycle where isochoric means that heat is added with constant volume and isobaric is when heat is added with constant pressure [Johansson et al., 2014]. It is also possible to do a mix mode cycle where both isochoric and isobaric heat addition is used. For a isentropic process a relation can be derived with the ideal gas law that gives

$$p_2 = p_1 \left(\frac{V_1}{V_2} \right)^{\gamma} \tag{6.27}$$

With this relation a full ideal pressure trace can be calculated. For this controller only a small section of the pressure trace is taken into consideration and this section is in between inlet valve closing and exhaust valve opening. Therefore the reference will only be calculated between inlet valve closing and exhaust valve opening.

Isochoric cycle

In the isochoric cycle for this case there are four points of interest:

1. The starting point (inlet valve closing)
2. Top dead center before heat is added
3. Top dead center after heat is added with constant volume
4. The end point (exhaust valve opening)

The pressure reference can be calculated with the following equation

$$p_{ref} = \begin{cases} p_1\left(\frac{V_1}{V(\theta)}\right)^{\gamma}, & \theta_1 \leq \theta \leq \theta_2 \\ p_3\left(\frac{V_2}{V(\theta)}\right)^{\gamma}, & \theta_2 < \theta \leq \theta_4 \end{cases} \tag{6.28}$$

where p_i, V_i and θ_i are the pressure, volume and crank angle in interest point i and in the iscochoric case it is known that $V_2 = V_3$. p_3 is the pressure after heat is added and this is decided by desired gross indicated mean effective pressure, $IMEP_g$, which is defined in this thesis as

$$IMEP_g = \frac{1}{V_d}\int_{V_{IVC}}^{V_{EVO}} pdV = \frac{1}{V_d}\int_{\theta_{IVC}}^{\theta_{EVO}} p\frac{dV}{d\theta}d\theta \tag{6.29}$$

The $IMEP_g$ for the ideal isochoric cycle can be calculated as [Johansson et al., 2014]

$$IMEP_g = \frac{p_3V_3 - p_4V_4 + p_1V_1 - p_2V_2}{(\gamma-1)V_d} \tag{6.30}$$

By choosing a desired $IMEP_g$ and the assumption that

$$p_2 = p_{IVC}\left(\frac{V_{IVC}}{V_{TDC}}\right)^{\gamma}, \quad p_4 = p_3\left(\frac{V_{TDC}}{V_{EVO}}\right)^{\gamma} \tag{6.31}$$

all variables are known to calculate p_3 as

$$p_3 = \frac{IMEP_g(\gamma-1)V_d - p_1V_1 + p_1\left(\frac{V_1}{V_2}\right)^{\gamma}V_2}{V_2 - \left(\frac{V_2}{V_4}\right)^{\gamma}V_4} \tag{6.32}$$

With these results the complete isochoric cycle can be calculated. A new variable p_{ic} is defined as

$$p_{ic} = p_3 - p_2 \tag{6.33}$$

which is the change in pressure in the point TDC before and after heat is added in the isochoric cycle. This variable will be used in the mixed mode cycle.

Mixed mode cycles

The mixed mode cycle is when isochoric and isobaric cycle is combined and from how the the mix mode cycle is defined the pressure reference for the isobaric cycle comes naturally. The mixed mode cycle consists of five points of interest:

1. The starting point (inlet valve closing)
2. Top dead center before heat is added
3. Top dead center after heat is added with constant volume

4. Top dead center after heat is added with constant pressure
5. The end point (exhaust valve opening)

In the case of a isobaric cycle there will be no heat added in point 3.

$$p_{ref} = \begin{cases} p_1\big(\frac{V_1}{V(\theta)}\big)^{\gamma}, & \theta_1 \leq \theta \leq \theta_2 \\ p_2 + \varepsilon p_{ic}, & \theta_2 < \theta \leq \theta_3 \\ p_3\big(\frac{V_2}{V(\theta)}\big)^{\gamma}, & \theta_3 < \theta \leq \theta_4 \end{cases} \tag{6.34}$$

where p_{ic} is defined in Equation (6.33) and ε is a number between 0 and 1 to how many percent of the isochoric pressure rise should be used. In the case when $\varepsilon = 0$ the cycle is isobaric and when $\varepsilon = 1$ the cycle is isochoric. Because the reference pressure is divided into three parts then the integral for $IMEP_g$ can also be divided into three parts

$$IMEP_g = \frac{1}{V_d}\left(\int_{V_1}^{V_2} pdV + \int_{V_2}^{V_3} pdV + \int_{V_3}^{V_4} pdV\right) \tag{6.35}$$

In the case of mixed mode cycle and isobaric cycle V_3 is the unknown. By solving the integrals in Equation (6.35) and some rewriting results in the expression

$$\frac{V_d IMEP_g - p_1V_1 + p_2V_2}{p_2 + \varepsilon p_{ic}} - (\gamma - 1)V_2 = \varepsilon\gamma V_3 - V_4\left(\frac{V_3}{V_4}\right)^{\gamma} \tag{6.36}$$

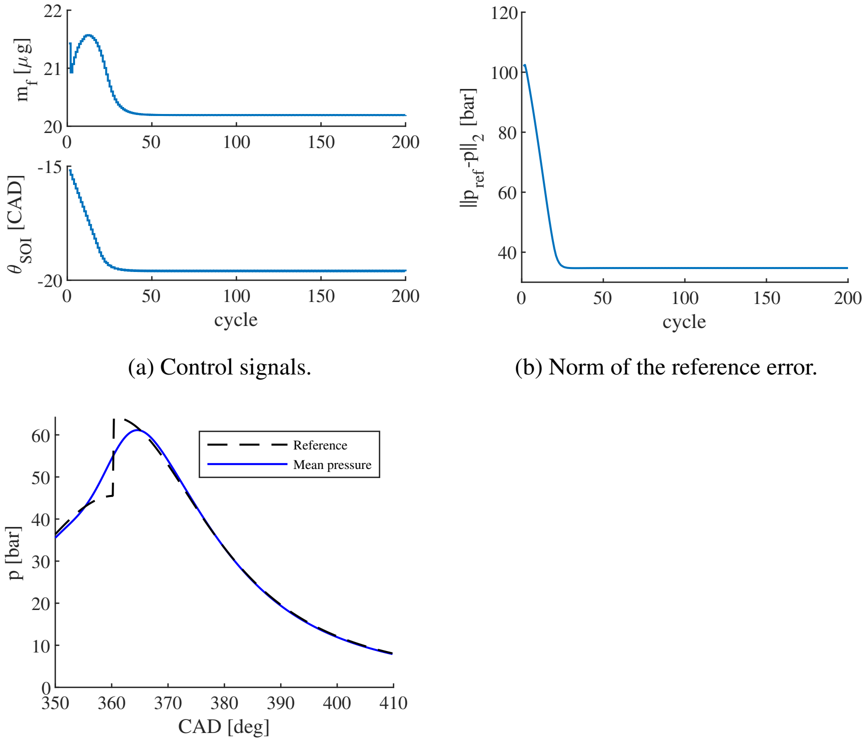
All variables except from V_3 are known and V_3 is found in matlab by inserting all values from a known volume vector and then choosing the best fit to the expression in (6.36). With this all values are known to calculate the full mixed mode cycle reference.

6.4 Simulation results

The simulations made are aimed to investigate possibilities of improvement. The performance of this type of controller is discussed thoroughly in [Turesson, 2018]. What is of most interest is to study if any iterative error occurs due to nonlinearities in the process, model errors or when disturbances are added to the system. Noise is introduced as a disturbance to the system to make the simulations relate closer to reality. Gaussian white noise is added to the ignition delay, combustion duration and pressure because these have a significant contribution to the pressure shape and affects the pressure tracking performance.

No noise

To determine how well the controller handles nonlinearities and model errors, no noise is added to the system. The simulation results in 6.3 are done where the parameter values in the system models and linearized models used by the MPC are the same. Figure 6.3 shows that the controller manages to converge to a pressure trace that is a good fit to the reference. To examine the steady state dynamics the mean in Figure 6.3c is calculated from the pressures in cycles between 60-200. A combustion fast enough to match the step in the pressure reference at TDC can not be achieved and a too large pressure rate wants to be avoided because it can lead to unnecessary tear in the cylinder. Therefore, the norm of the error will not converge towards zero. The euclidean norm of the error was investigated because it gives a single value that is a good representation of how well the pressure trace matches the reference at all measured point in the cycle. This norm is also part of the optimization problem in the controller which gives an idea when the controller reached its optimal solution. When comparing results from figure 6.3b and 6.3c the controller is not affected much by nonlinearities in the system.



(a) Control signals.

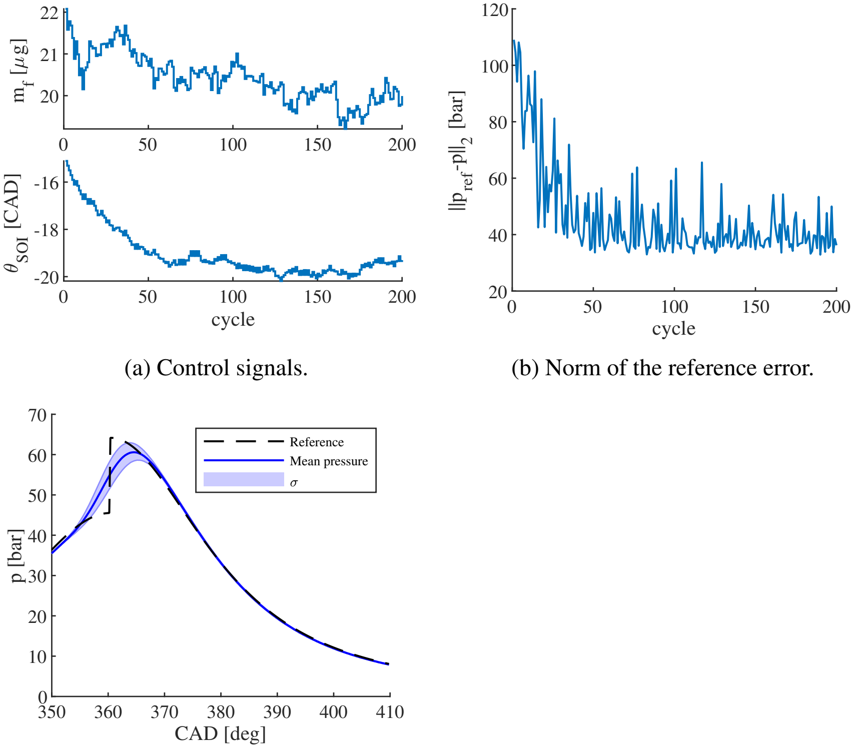
(b) Norm of the reference error.

(c) Mean pressure trace from cycle 60 to 200.

Figure 6.3: Simulation results from when no noise is added to the system. The reference is an isochoric cycle with $IMEP_g = 2$ bar

Noise disturbances

Noise is added to the ignition delay, combustion duration and pressure. The ignition delay and combustion duration are heavily dependant on chemical reactions which usually are very stochastic. Noise on the pressure comes from the pressure sensors which generate a measurement noise on the pressure signal. All noise added are zero mean Gaussian white noise and the standard deviation of the noise is chosen to match reality but also within a range that the simulation can handle. When too large noise is added the simulation run into numerical problems. The standard deviations on the noise were chosen as $\sigma_\tau = 0.1$ ms, $\sigma_{CD} = 2$ CAD and $\sigma_p = 10^4$ bar for the ignition delay, combustion duration and pressure, respectively. The results in Figure 6.4 are from simulation with the same simulation conditions as in Figure 6.3 but with the added noise. As can be seen in Figure 6.4c the pressure converges to the



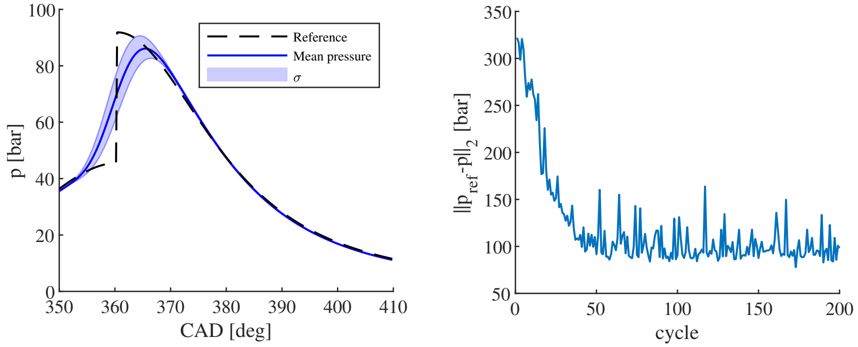
(a) Control signals.

(b) Norm of the reference error.

(c) Mean pressure trace from cycle 60 to 200.

Figure 6.4: Simulation results from when noise is added to ignition delay, combustion duration and pressure. All noises are zero mean Gaussian white noise with standard deviations $\sigma_\tau = 0.1$ ms, $\sigma_{CD} = 2$ CAD and $\sigma_p = 10^4$ Pa.

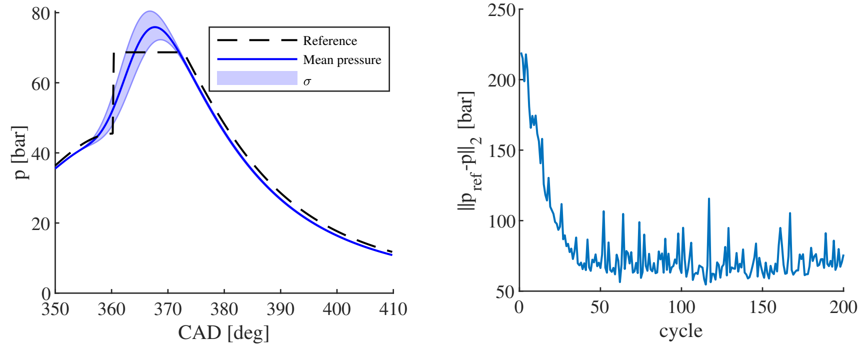
same mean as when there is no noise. The blue shaded area around the mean represents one standard deviation, σ, in pressure and together with the norm in Figure 6.4b it is seen that noise have a big impact on the pressure tracking performance. The results in Figures 6.5 and 6.6 are from simulations with the same noise but with different reference traces. The reference in Figure 6.5 is again an isochoric cycle but in this case with $IMEP_g = 5$ bar instead of 2 bar and in Figure 6.6 the reference is a mixed mode cycle with $\varepsilon = 0.5$ and $IMEP_g = 5$ bar. In both cases the controller manages to converge to a good fit but the noise seems to have the same effect no matter the reference trace.



(a) Mean pressure trace from cycle 60 to 200.

(b) Norm of the reference error.

Figure 6.5: Simulation results with a isochoric reference with $IMEP_g = 5$ bar.



(a) Mean pressure trace from cycle 60 to 200.

(b) Norm of the reference error.

Figure 6.6: Simulation results with a mixed mode reference with $IMEP_g = 5$ bar and $\varepsilon = 0.5$.

When analyzing the results from the simulation in Figure 6.4 it is seen that the pressure is mostly affected by the noise at points between 356-370 CAD. The error dynamics in steady state is of interest therefore the reference errors in some sample points over the cycles 60-200 are compared in Figure 6.7 and it is noticed that the errors seems to have similar dynamics and high correlation. To investigate the correlation between errors in steady state at different CAD, correlation coefficients between the errors in the interval 356-370 CAD are calculated. These correlation coefficients are represented in Figure 6.8 as a color plot where the yellow color represents high positive correlation and dark blue represents low correlation. As can be seen many points are highly correlated and by examining for example CAD 365 at the x-axis and look in a vertical line, most correlation coefficients are between 0.7 and 1. By visually presuming that errors in the interval 356-370 have similar dynamics and by verifying this with the correlation coefficients, the error dynamics at 365 CAD is assumed to be a good representation for all errors in this interval. The error dynamics of steady state reference error at 365 CAD are to be studied to determine if white noise cause some frequency dynamics or if it goes through to the system as white noise as well. To find out if the error is white noise the sampled autocorrelation function is studied in Figure 6.9. The autocorrelation function is used to find if there is any periodic behaviour in a signal and the autocorrelation of white noise is 1 at lag 0 and 0 for all lags larger than 0. The blue lines in Figure 6.9 is a 95 % confidence boundary and since the boundary is close to zero and most values are within or close to the boundary the steady state error is most likely white noise.

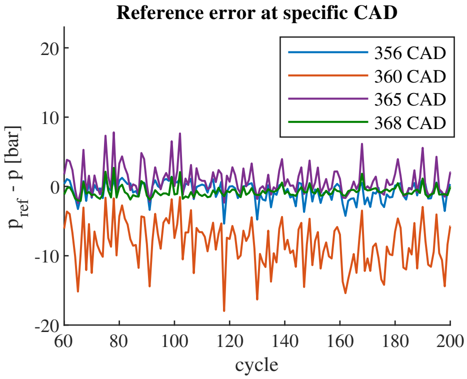


Figure 6.7: The reference error at different CAD.

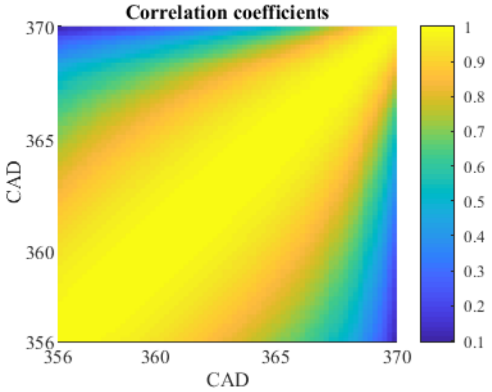


Figure 6.8: Correlation coefficients of the reference errors at 356 to 370 CAD. Yellow color indicates positive correlation and dark blue color indicates next to none correlation.

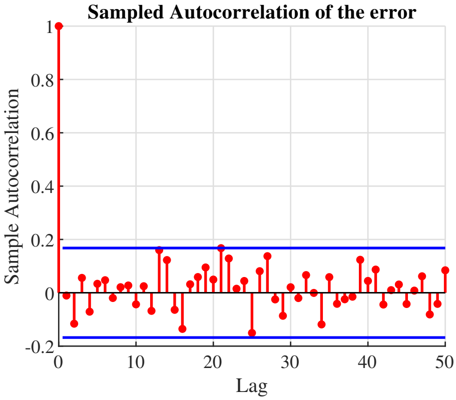


Figure 6.9: Sampled autocorrelation function of the error at 365 CAD. The blue lines shows a 95 % confidence boundary.

6.5 Colored noise

When examining experimental data it is shown that the noise on the combustion duration has colored characteristics. This can be explained physically that for example wall temperature and residual gases from last cycle effects the next cycle. In Figure 6.10 the autocorrelation of the experimental combustion duration compared to the autocorrelation of the simulated combustion duration with colored noise added. It was found that colored noise defined as brown noise had a good match in terms of autocorrelation. Wall temperature and residual gases should also have affects on the ignition delay therefore simulation with brown noise on both ignition delay and combustion duration was conducted. Results from this simulation is shown in

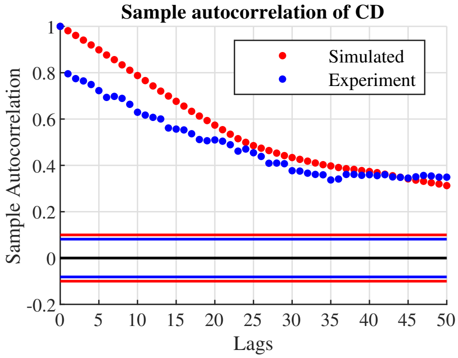
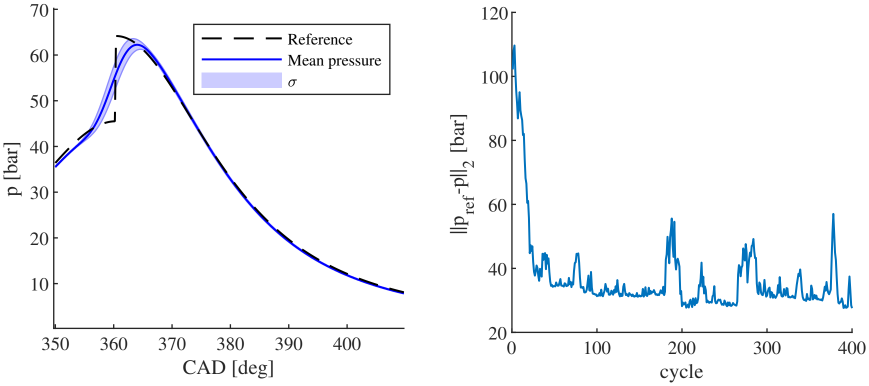


Figure 6.10: Sampled autocorrelation of combustion duration from experimental and simulated data.

Figure 6.11 and as can be the mean steady state pressure is similar to previous simulations but the standard deviation is not as large as when white noise is added. One reason for this is because the brown noise is not zero mean and a too large brown noise would lead to infeasible results. But now that colored noise was added an autocorrelation between the reference error at CAD 365 in next cycle and previous cycles, see Figure 6.12, occurs which indicates some kind of dynamics that are not stochastic.



(a) Mean pressure trace from cycle 60 to 400.

(b) Norm of the reference error.

Figure 6.11: Simulation results With brown noise on the ignition delay and combustion duration.

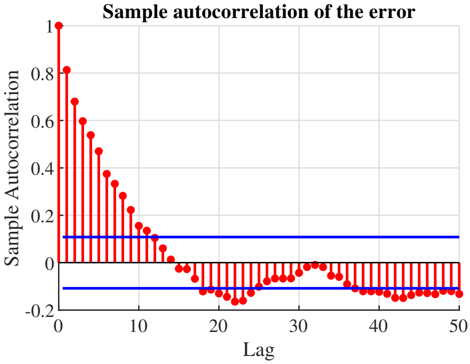
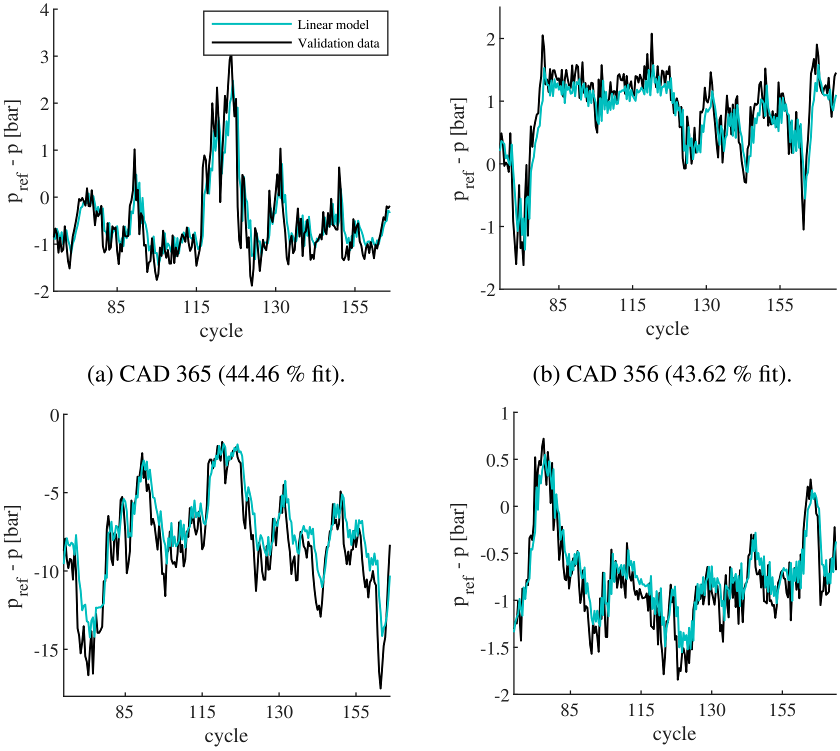


Figure 6.12: Sampled autocorrelation function of the reference error at 365 CAD. The blue lines shows a 95 % confidence boundary.

To find any dynamics between previous errors and the error in the next cycle, system identification toolbox in matlab was used. In this toolbox one set of data of the reference error at CAD 365 was divided in two parts estimation data and validation data. A second degree transfer function model was found that showed promising system dynamics and was estimated as

$$e_{k+1} = \frac{0.6147}{1+0.1166q_k^{-1}-0.4203q_k^{-2}}e_k \tag{6.37}$$

where e_k is the reference error and q_k is the cycle shift operator defined as $e_k = q_k^{-1}e_{k+1}$. Validation results for this model are shown in Figure 6.13. The model only fits around 45 % compared to the validation data but when looking at the figures it is clear that there are some linear dynamics between previous and next cycle error. When doing residual analysis the autocorrelation of the residuals and the cross-correlation between the input signal and the residuals is shown in Figure 6.14. Since the autocorrelation is not inside the confidence interval for all lags it shows that there are still more information in the residuals that can be identified. Though the autocorrelation is still small enough to say that the model is fairly accurate. The cross-correlation is inside the confidence interval for positive lags which is what matters when validating the model. But for negative lags it there is a significant cross-correlation which is a sign of feedback in the system . Since the data is taken from the controlled closed loop feedback is highly plausible. This simple model achieved around a 45 % fit for not only validation data from CAD 365 but also for reference error data from different CAD which shows that the noise affects the error similarly at most points in the interval of interest between 356-370 CAD. Now that error dynamics have been established similar methods was used to find dynamics between reference signal and the error. Since ILC is applied to the reference signal to minimize the error a model from reference to error is preferred for



(a) CAD 365 (44.46 % fit).

(b) CAD 356 (43.62 % fit).

(c) CAD 360 (46.11 % fit).

(d) CAD 368 (46.2 % fit).

Figure 6.13: Results when validating linear model for the reference error at different CAD. Black line are simulated validation data and turquoise line is linear model

controller design purposes. The otherwise constant reference signal was exited with white noise to produce data where dynamics could be identified and then the system identification toolbox was again used. This time no model between reference and reference error could be identified. Either because the reference was not excited enough to show any dynamics, more advanced identification methods are needed or the nonlinearities in the controlled closed loop from both the engine and the MPC can not be estimated well with a linear model.

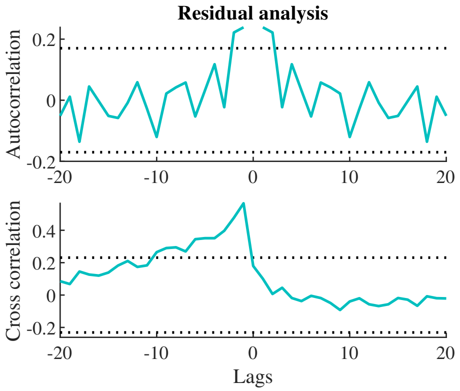


Figure 6.14: Autocorrelation of the residuals and cross correlation of the input and the residuals are shown in this figure. The dotted lines represents a 99 % confidence interval

6.6 ILC

Even though no dynamics between the reference to the error could be identified and no obvious ILC design could be found, implementation options for MPC was still investigated. This is tested because ILC can not be ruled out from working on the real process from the results of this simplified simulation. First it was concluded that an ILC working in a crank angle profile would be hard to implement with this specific MPC because the reference for the whole next cycle has to be calculated and sent into the MPC before the next cycle starts. This meant a heuristic ILC design could not be implemented and more details about the system dynamics are needed. To not be able to implement in a CAD profile limited the implementation to an ILC working in a cycle profile where the ILC control signal was added to the reference. Adding the ILC to the MPC control signal was discouraged since this could disturb the constraint handling properties of the MPC. A cycle profile ILC was implemented such that control signals and reference errors from current and past cycles. But as mentioned earlier because of no obvious iterative errors or reference to error dynamics could be found no ILC design could be made to improve the pressure tracking with this ILC.

Discussion

Pressure tracking MPC has very good performance in simulation and can follow different reference traces. One reason for this good performance is that the models used to simulate the engine are the same models that are linearized and used in the MPC. Because the controller has the adaptive functionality the linearized model will always be similar to the nonlinear model. For this reason, controller performance as good as the simulated results can not be expected in a real life implementation. But since this type of controller have been tested on a real engine with good results it was also expected that it would give good results in simulation.

When adding white noise to the system the results shows that there definitely is room for improvement when it comes to noise reduction. From the results it is seen that the pressure is mostly affected at CAD that are during or close to the heat release which is to be expected since both ignition delay and combustion duration affects this directly. When analyzing the correlation between reference errors at different CAD it can be concluded that these white noise disturbances affect the pressure in a similar way where the biggest errors occur. Finding that the error dynamics does not have any iterative behaviour and has a high probability to be white noise this leads to believe that it is hard to handle these noise disturbances.

When adding brown noise which is supported by the experimental data linear cycle to cycle error dynamics could be found. This opens up the possibility for ILC to improve the the pressure tracing performance. Though since no reference to error model could be identified no ILC design improving the pressure tracing performance could be found. It can not be concluded that ILC is not a possible way of improvement on the real process. Because of lost dynamics and possibly too simple models in the simulation makes the simulated result insufficient to say that ILC can not be used to improve the pressure tracking. An ILC working in a cycle profile is possible to implement and if dynamics between reference and error are found a model based ILC design can possibly used.

7 Conclusion

This thesis has been about the making of a simulation testbed of a combustion engine for control purposes. The control strategies in the simulation is aimed towards MPC and from those results has opportunities for other control strategies are investigated. Results regarding the engine models includes simplifications, assumptions, software implementation, performance and improvements. First a conclusion about the simulation performance is made and then conclusions about performance of tested MPC controllers and opportunities for other control strategies are drawn.

7.1 Simulation

The simulation uses very simple models for the engine and many assumptions are also made. This of course leads to that a lot of dynamics are lost and make this simulation less trustworthy. But even with these simple models the results compared to experimental data shows that many of the main dynamics can be simulated. There are more complex models that are possible to implement but this would lead to a trade off between computation time and model accuracy. It is concluded that this simulation is good enough for control purposes and early controller design stages.

When it comes to software Simulink is an accessible and easy to use software. It has been proven to yield good results based on the models implemented and it is easy to add improvements to the simulation program when needed. The MPCtools Toolbox is a good choice for implementation of MPC controllers in Simulink. It has many features built in that can easily be used and the possibility to implement an adaptive feature makes it preferable for this project. It would have been possible to implement an MPC controller optimized for just this simulation but this would have been very time consuming and the time that might have been won in simulation time is not significant enough. Even though this project has focused on MPC it would have been easy to replace and implement other control strategies in this simulation.

7.2 MPC

Regulating this simulation with MPC gave good results but a big reason for this is that very simple models are used and also that the models in the MPC are linearized models of the ones used to simulate the engine. This leads to better results than to be expected in the real engine. A big improvement to increase how trustworthy the MPC performance is would be to simulate the engine with models that the MPC are not based on. With this, more possible improvements to the controller could maybe be found and investigated.

When analyzing how noise disturbances affect the system controlled with pressure tracking MPC it is concluded that the nonlinearities in the engine models do not show any noticeable change in frequency dynamics on the output when exposed to white noise. Though when colored noise was introduced a strong autocorrelation between past and present reference errors was found which indicates there might be dynamics that can be exploited. A linear transfer function that have a decent representation of the present error to the error in next cycle could be identified but no linear dynamics between the reference and the error could be identified which made it impossible for any model based ILC design. There are still more work that can be done regarding system identification and more refined identifications methods than the ones used in the system identification toolbox might find dynamics between the reference and the error.

7.3 ILC

Since reference error dynamics could be found it is concluded that ILC might be useful to improve the controlling performance of the system in this simulation. But since no linear dynamics between the reference and the error was found and because of that no model based ILC design could be found either. There were no obvious iterative errors occurring in neither a CAD profile or cycle profile. But as explained earlier a lot of dynamics are not present in this simulation and the performance of the pressure tracking MPC most likely better than expected. Therefore, the applicability of ILC can not be ruled out just from results of this simulation. The possible ILC in cycle profile implementation was found but due to lack of understanding of the system dynamics it could not be tested enough. Suggested future work are either more work on system identification in the simulation or experiments on the real process to find iterative dynamics when pressure tracking MPC is used.

Bibliography

Abbaszadehmosayebi, G. (2014). *Diesel engine heat release analysis by using newly defined dimensionless parameters.* PhD thesis. uk.bl.ethos.629962. School of Engineering and Design, Brunel University, London, United Kingdom.

Åkesson, J. (2006). "MPCtools 1.0 — reference manual". ISSN: 0280-5316.

Åström, K. J. and B. Wittenmark (1989). *Adaptive control.* 2nd ed. Addison-Wesley series in electrical engineering. Addison-Wesley, Boston, USA.

Heywood, J. B. (1988). *Internal combustion engine fundamentals.* 2nd ed. McGraw-Hill series in mechanical engineering. McGraw-Hill, New York, NY, USA.

Johansson, B., Ö. Andersson, P. Tunestål, and M. Tunér (2014). *Combustion Engines*. v. 1. Department of Energy Sciences, Lund University, Lund, Sweden. ISBN: 9789176230954.

Johansson, R. (2008). *Predictive and Adaptive Control*. Department of Automatic Control, Lund University, Lund, Sweden.

Königsson, F. (2010). *A Combustion Model for Diesel Engines*. M.Sc diva2:460060. Dept. Machine Design, KTH, Stockholm, Sweden.

Li, C. (2018). *Stratification and Combustion in the Transition from HCCI to PPC*. PhD thesis. TMHP-18/1140-SE. Dept. Energy Science, Lund University, Lund, Sweden, pp. 10–11.

Norrlöf, M. (2004). "Disturbance rejection using an ILC algorithm with iteration varying filters". *Asian Journal of Control (Vol.6, No. 3).*

Slepicka, C. and C. R. Koch (2016). "Iterative learning on dual-fuel control of homogeneouse charge compression ignition". *IFAC-PapersOnLine* **49**:2, pp. 347–352.

Turesson, G. (2018). *Model-Based Optimization of Combustion-Engine Control*. PhD thesis. TFRT-1120. Dept. Automatic Control, Lund University, Lund, Sweden. ISBN: 978-91-7753-705-2.

Yan, F. and J. Wang (2011). “Common rail injection system iterative learning control based parameter calibration for accurate fuel injection quantity control.” *International Journal of Automotive Technology* **12**:2, pp. 149–157.

Zweigel, R., F. Thelen, D. Abel, and T. Albin (2015). “Iterative learning approach for diesel combustion control using injection rate shaping”. In: European Control Conf., p. 213.