Knock mitigation study on alternative fuel heavy duty engines

Master's thesis in Mobility Engineering

Dimitra Katsikioti

DEPARTMENT OF MECHANICS AND MARITIME SCIENCES

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024
www.chalmers.se

MASTER'S THESIS IN MOBILITY ENGINEERING

Knock mitigation study on alternative fuel heavy duty engines

DIMITRA KATSIKIOTI

Department of Mechanics and Maritime Sciences
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2024

Knock mitigation study on alternative fuel heavy duty engines
DIMITRA KATSIKIOTI

Supervisor: Axel Hansson, Volvo Penta & Prajwal Shankaregowda, Volvo Penta
Examiner: Petter Dahlander, Department of Mechanics and Maritime Sciences

Department of Mechanics and Maritime Sciences
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: + 46 (0)31-772 1000

Cover:
Cover image sourced from “Maybe there is a future for the internal combustion engine after all” by Milan Lenters, published in Innovation Origins, 2021

Department of Mechanics and Maritime Sciences
Göteborg, Sweden 2024-02-10

Knock mitigation study on alternative fuel heavy duty engines
Master's thesis Mobility Engineering
DIMITRA KATSIKIOTI
Department of Mechanics and Maritime Sciences
Chalmers University of Technology

Abstract

With increasingly stricter emissions regulations and a growing demand for higher efficiency, alternative fuels are becoming more viable alternatives to fossil fuels. Methane, for example, can be used in conventional combustion engines with minimal modifications, meeting lower emission targets. However, engines running on alternative fuels still face issues common to conventional engines, such as "knock," which limits their efficiency.

This thesis investigates the manifestation of knock in alternative fuel engines and proposes a method to prevent it while maintaining high efficiency. The method utilizes Model Predictive Control (MPC) and Artificial Neural Networks (ANNs). A detailed outline of the proposed controller is provided, along with the rationale behind its key structural characteristics. The use of ANNs to model the system's state is also explored and evaluated. The controller is tested under various operating conditions and tuning settings to verify its effectiveness and identify optimal tuning.

The proposed control method performed well in the selected cases and setup. Further improvements to both the controller and neural network structure are suggested to enhance performance. This thesis establishes that MPC can improve engine performance and prevent unfavorable operating conditions. Initially useful for engine mapping, the controller has future potential for direct implementation on the ECU.

Key words:

Alternative fuels, Combustion, Knock, Neural Networks, Control Theory, Power Generation

List of acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

ANN	Artificial Neural Network
ECU	Engine Control Unit
EGR	Exhaust gas recirculation
GHG	Greenhouse gas
ICE	Internal combustion engine
MPC	Model predictive control
TDC	Top dead centre

Contents

List of figures

List of tables

1 Introduction

1.1 Background

In the past years, reducing emissions, has become of outmost importance due to a convergence of pressing environmental and societal concerns. The rapid development of the industry sector, as well as the transport sector, has raised overall the living standards but at the same time has brought on negative effects to the environment and as an extension, to human health. Internal combustion engines using fossil fuels are responsible for about 25% of the global power production and at the same time 17% of total GHG emissions [1]. Among the emissions produced by internal combustion engines the main ones are carbon dioxide (CO_2), carbon monoxide (CO), nitrogen oxides (NO_x) and particulate matter (PM) [2] .

There have been many efforts to mitigate this situation with increasingly strict emissions legislation, electric powertrains, and alternative fuels. At the same time, while the main stressor is the environmental issue, another really key factor to consider is that fossil fuels are a finite source of energy and thus replacing them at some point will be inevitable [2].

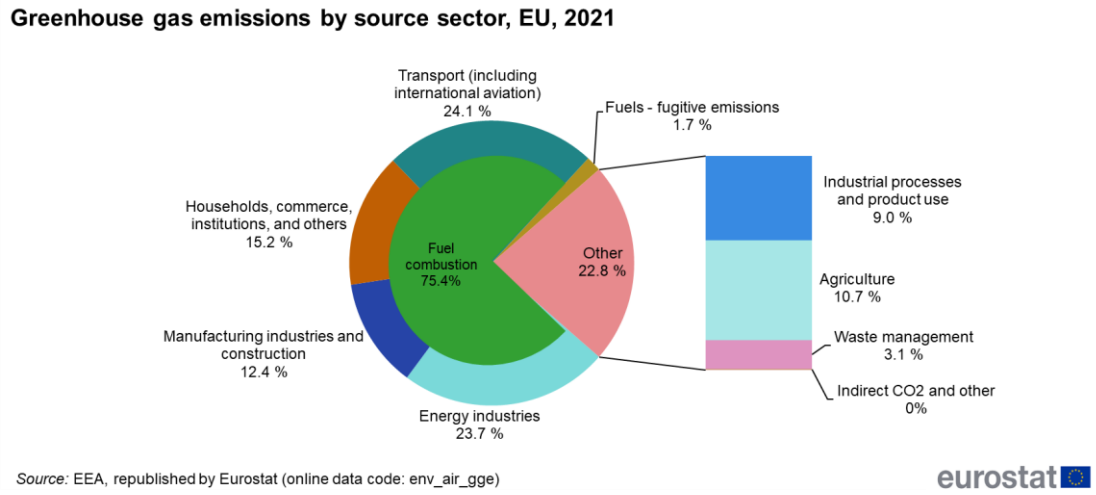

Figure 1.1 Greenhouse gas emissions by source sector, EU 2021 *[1]*

While electric powertrains have been a promising solution for the light duty vehicle category mostly, there are still areas where using electric alternatives is not feasible or efficient enough. Especially for heavy duty engines, that are used either for transportation or power generation the focus has shifted more on alternative fuel engines, which utilize either biofuels or other types of fuels like natural gas, hydrogen etc. Among the main advantages of alternative fuels is firstly, the lower overall emissions, the abundance of sources and consequently availability and the fact that they can be used on "conventional" internal combustion engines with minor modifications [3] [4]. Since alternative fuels, such as natural gas, are used on modified internal combustion engines, it is easy to conclude that they could encounter the same problems that affect their operation. A very prominent problem among internal combustion engines encounter during their operation is the "knocking" phenomenon [5].

When it comes to knocking, the main solution up until now has been altering the spark timing to prevent it from occurring. However, at the same time this strategy reduces the engine's overall efficiency. So, it becomes clear, that understanding knock occurrence in depth and developing a strategy to limit it while also maintaining efficiency and safe operating conditions is a topic worth investigating further [5] [6].

1.2 Purpose

The purpose of this thesis is to investigate the knock phenomenon on alternative fuel engines and more specifically natural gas engines. By understanding the root causes and the operating variables that lead to this abnormal combustion, it helps one to understand on how it can be mitigated. After the most important affecting factors are located, it is necessary to figure out a strategy on how to both monitor combustion settings and how to prevent knock from happening. There is a vast literature with developed methods and indicators aiming to prevent knock while optimizing engine efficiency. When it comes to alternative fuel engines specifically, the available literature is less comprehensive, so the challenge is to determine which of the developed methods can also be applied to engines that operate with natural gas instead of gasoline. By taking all the above into consideration, the final aim of this thesis is to develop a method that can be used to mitigate knock in an internal combustion engine run on methane. This method will be applied to a controller, which coupled with the engine simulation model will be evaluated to assess its results.

1.3 Objectives

The objectives that are to be fulfilled by the end of this master's thesis are the following:

1. Literature study to find existing solutions for engine knocking. Studying of engine combustion and identification of control objective and available actuators.
2. Creation of simulation model for software control and optimization. Continuation of combustion model development.
3. Development of controller by choosing suitable strategy, controller states, cost function etc.
4. Evaluation of results to suggest an optimal combustion controller and analysis of results based on knock mitigation and efficiency increase.

1.4 Scope

The scope of the thesis is limited to:

- Locating and adapting already existing methods and strategies for knock measuring and mitigation.

- Developing a method based only on measurements from a heavy-duty engine, that is used for power generation, and operates with natural gas, available at Volvo Penta.
- Data for knock occurrence that originate from simulations of the heavy-duty engine on GT-Suite. Other types of actuators are not considered.
- The model for the simulations is provided mostly completed, with the possibility of minor modifications to be needed.

The thesis is conducted by one student and the time frame is limited to the end of the semester (June).

1.5 Limitations

Limitations of the thesis or topics that were not studied include:

- Prevention of knock by altering spark timing and EGR position and no other actuators.
- Training of the controller with transient profiles.
- Development and verification of controller are based only on simulation results and not actual data from the physical engine.

Furthermore, it is important to note that some values, like the torque request in Section 5.2, or details of the engine simulation model are omitted due to them being confidential information. The same goes for brake efficiency values. However, this does not affect the conciseness and clarity of the presented results.

1.6 Ethical Aspects

This thesis proposes a controller to mitigate knock and increase efficiency for alternative fuel engines. It is a first step in an effort to accelerate the traditional engine mapping process and in the future, it is possible that the controller is implemented on the engine ECU directly. However, it is obvious that during these initial stages the controller needs to be supervised and verified with actual engine mapping results before any further steps can be taken. At the same time, it is quite clear that when it comes to software control there is always the possibility of it malfunctioning, so especially at the initial stages any implementation of the controller should be handled with caution.

On the other hand, the successful implementation of the controller could assist in further promoting the use of alternative fuel engines. Alternative fuel engines play a crucial role in becoming increasingly independent from fossil fuels. This is in line with the 17 Sustainable Development Goals stated by the United Nations (UN) [7]. Among those goals, the most relevant are the goals for affordable and clean energy, sustainable cities and climate action. All these goals are inline with the thesis objectives. By reducing the engine mapping process and having a controller that ensures the engine's proper function with the highest possible efficiency combined with the already lowered emissions of alternative fuel engines, the transition from fossil fuels to sustainable energy sources could be

accelerated. Also, by shortening the engine mapping and replacing it with virtual means, the physical engines will have to be tested/ operated for a shorter amount of time or not at all. In that way the emissions and overall pollution that are related with engine operation could also be avoided.

1.7 Disposition

The definition of knocking in ICEs and all the operational conditions that lead to it are detailed in Chapter 2. Also, at the end of that chapter the selected actuators for the scope of this thesis are presented. In chapter 3, the concept of control systems and MPC controllers is discussed. Furthermore, the definition of ANNs is given in the same chapter and how they are relevant to the development of the controller. By chapter 4, an elaborate description of all the relevant controller components is provided and how they all interact with each other to achieve the desired results. In chapter 5, the results are discussed. In the last chapter, some conclusions along with some suggestions for future work are given and some further improvements that could be made.

2 Abnormal combustion: Knock

2.1 Definition and characteristics

As has already been mentioned briefly, abnormal combustion occurrences, such as backfiring, pre-ignition and knocking have been causes of significant concern for engineers. These problems are also present with alternative fuel engines, but this research is far more recent and less developed [4]. There have been efforts to understand if the same principles that apply for fossil fuel combustion engines are also the same for engines that operate for example using natural gas/ methane. It is therefore important to establish what knocking is, the mechanism behind it, the most crucial factors that lead to it and lastly what are the most established mitigation strategies.

In general, it is widely accepted that knock occurs "*due to the rapid autoignition of the end gas (unburnt mixture ahead of the flame), while the ignition process has already begun in the combustion chamber and the flame propagates"* [8]. For a certain quantity of unburnt mixture, the autoignition results in a sudden heat release which causes both high in-cylinder pressure and significant pressure oscillations. This happens usually between 10^{o} and 50^{o} after TDC (Top Dead Centre) [9].

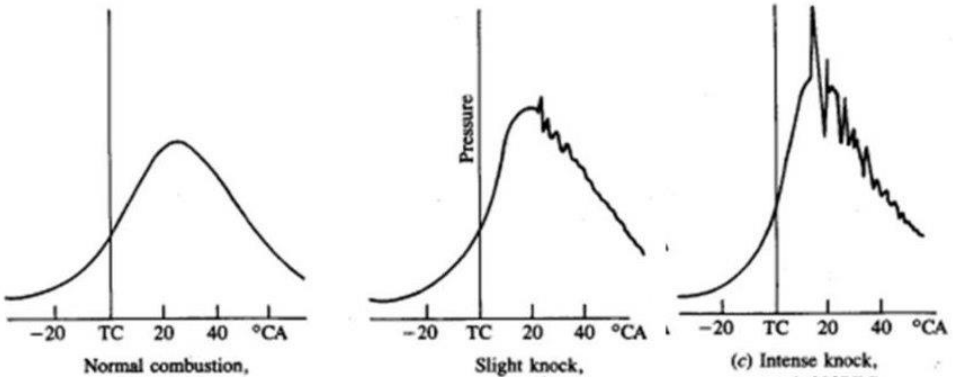

Figure 2.1 Normal combustion vs Knocking Combustion *[9]*

Usually, the fuel's octane number is an indicator of its resistance to knock. However, this rating method has not been developed to accurately represent gaseous fuels and fuels that are less sensitive to knock than iso-octane [10].

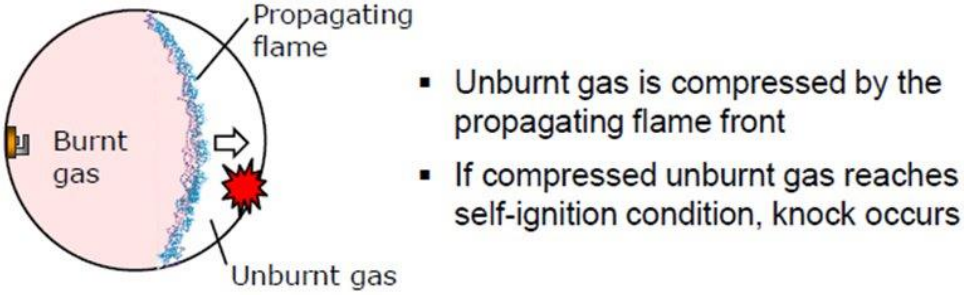

Figure 2.2 Knock phenomenon in combustion chamber *[11]*

In an effort to avoid knock occurrence, usually lower compression ratios are preferred. Gasoline engines are more sensitive when it comes to knock, while for diesel engines it is not really a cause of concern [12]. As a result, the efficiency of the combustion and the net power output of the engine is also limited. Other negative implications of knock occurrence are component thermal degradation. The thermal degradation occurs due to the increased heat transfer through the chamber walls (reduced boundary layer thickness), which in turn is the result of the intense pressure fluctuations [13]. At the same time, the pressure fluctuations can cause significant vibration of engine components and the results can vary from an audible "knocking" sound (hence the name knock), to severe structural damage [9]. To sum up, some common negative effects of knock can be:

- Breakage of pistons rings.
- Cylinder head erosion.
- Limited engine efficiency or vehicle acceleration performance.
- Increased air pollution.
- Considerable rise in engine specific fuel consumption (SFC).
- Possibility of long-term structural harms to engine.
- Source of noise during engine operation [13].

It is therefore clear, that avoiding knock or developing an effective mitigation strategy is very crucial to both the efficiency of a combustion engine and its longevity.

2.2 Key factors

A very important step towards that direction is to locate the main factors that are closely related to knock occurrence. The most notable factors are:

- **Fuel octane rating:** The higher the octane rating of a fuel is, the more resistant to knock it is. Accordingly, fuels with low octane number are prone to preignition [12].
- **Fuel composition:** It has been noticed that knock occurrence is connected to the presence of certain additives and hydrocarbons in the fuel composition [12].
- **Compression ratio:** It has been briefly mentioned that the higher the compression ratio is, the more likely it is for knock to occur. The main reason behind that is that increased compression ratio leads to an increased temperature and pressure for the air-fuel mixture, thus making it more susceptible to autoignition [12] [13].
- **High temperature:** Apart of the compression ratio, also elevated engine operating temperatures could affect the temperature of the air-fuel mixture. To avoid such phenomena, it is important to have an effective cooling system which is regularly inspected [8].
- **Ignition timing:** It is known that by advancing the spark angle by a certain amount can lead to increased combustion efficiency, but advancing too much can lead to knock. Accordingly, by retarding the

spark angle, some of the engine efficiency is "sacrificed" but at the same time knock is avoided [8] [13].

- **Air-fuel mixture:** Lean mixtures are more prone to knock because they lead to higher combustion temperature and pressure. But that doesn't mean necessarily that rich mixtures do not cause knock. When it comes to richer mixtures, an overly rich mixture can also lead to knock due to incomplete combustion and carbon deposits. Another important aspect to consider, is the homogeneity of the mixture. The more homogenous a mixture is, the less prone to knock it is [12].
- **Engine speed & load:** At higher engine speeds the combustion time is shortened, and the risk of incomplete combustion is increased. Also, at higher loads, the pressure and temperature are increased, thus the possibility of knocking .
- **Engine design & components:** Various aspects of engine design could influence knock. A very prominent example is the combustion chamber design. As it has already been stated, the more homogeneous the air-fuel mixture is, the less likely knock is to happen. So, by having a combustion chamber design that promotes efficient mixture and an even combustion knock could be reduced. Also, a particularly important aspect is the chosen materials. Materials with the appropriate thermal properties (good thermal conductivity), could help dissipate heat more effectively and reduce knock [14] [15].
- **Carbon deposits:** Buildup of carbon in various spots such as the piston crown, cylinder head and valves create hotspots that could lead to pre-ignition and knock [8].
- **Ambient conditions:** Outside temperatures can obviously affect the intake air temperature and affect the combustion process. The same goes for altitude, which can affect the air pressure and subsequently the amount of combustible oxygen intake per stroke [5].

2.3 Detection methods

It has already been established that knock is a highly complex phenomenon, which also becomes clear by the plethora of factors that could directly or indirectly lead to it. Therefore, finding a methodology to mitigate it or avoid it all together, can be also a very delicate process. The most common process is called engine screening. During engine screening, the operating areas where knock is happening are located by varying several key states of the engine [3] [15]. Apart from locating the region where knock is happening, engine screening is important also for finding the "optimal" engine settings to get the desired engine performance. In other words, the purpose of mapping (when it comes to knock specifically), is to achieve a spark timing as close to knocking as possible. In that way, the efficiency is maximized, without having knock at the same time [5].

For the engine screening and knock data, sensors are used. There are different types of sensors that can be used to detect knock, depending on the preferred methodology. The most prominent categories are:

- Temperature sensors placed in the combustion chamber.
- Sensors for noise produced by engine during operation.
- In- cylinder pressure sensors.
- Pressure sensors for the coolant.
- Measurement of light emitted during combustion [5].

Accordingly, the processes are separated into methods based on cylinder pressure analysis, methods based on engine block vibration analysis, methods based on gas ionization analysis and methods based on heat transfer analysis. Usually, the process includes collecting data from the sensors (e.g. pressure signal) and then processing them to determine whether knock is happening or not [16] [17] . The post processing signal usually includes filtering the signal to separate the noise and then calculating an indicator which is compared to a threshold value usually. If the indicator exceeds the threshold value, then knock is taking place. This threshold value is engine specific [18]. Some of the most common indicators based on pressure are presented in Figure 2.3- Figure 2.5.

Metric

Maximum Amplitude of Pressure Oscillation [40]

Abbreviation:

MAPO

Signal:

Filtered in-cylinder pressure

Formula:

$$KI = \max\left(\left|p_{filt}\right|_{\theta_0}^{\theta_0+w}\right)$$

Illustration:

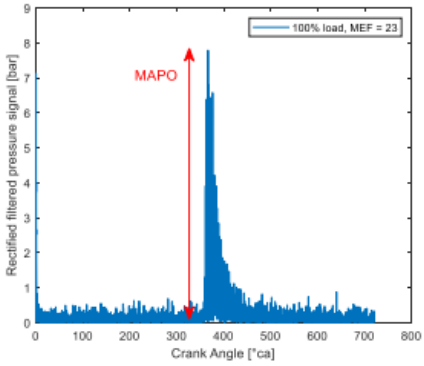

Figure 2.3 Knock indicator- MAPO *[17]*

Metric

Integral of Modulus of Pressure Oscillation [41]

Abbreviation:

IMPO

Signal:

Filtered in-cylinder pressure

Formula:

$$KI = \int_{\theta_0}^{\theta_0+w} \left|p_{filt}\right| \bullet d\theta$$

Illustration:

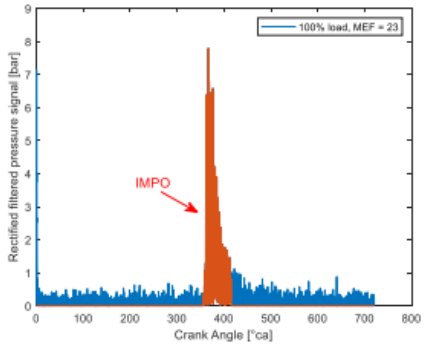

Figure 2.4 Knock indicator- IMPO *[17]*

Metric

Average Energy [33]

Abbreviation:

AE

Signal:

Filtered in-cylinder pressure

Formula:

$$KI = \frac{1}{N_s} \bullet \sum_{i=1}^{N_s} p_{filt}^2(i)$$

Illustration:

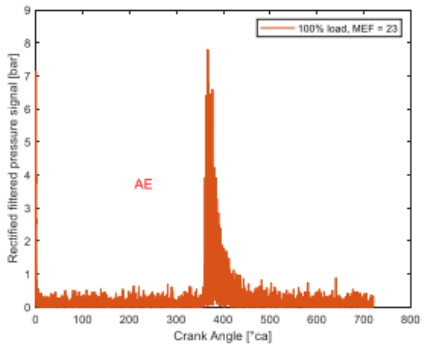

Figure 2.5 Knock indicator- AE *[17]*

2.4 Controller actuators

The described process can be quite time consuming and requires lots of resources. So, the aim of this thesis is to investigate whether a control method can be developed that could possibly reduce the engine screening process while also achieving similar or even better performance [19]. In order to do that, the first step is to establish which actuators can be used for a control scheme and then choose the appropriate ones that can be used to predict/prevent knock. For the scope of this thesis, the actuators/ signals chosen are:

1. ***Engine Speed [rpm]***

The correlation between engine speed and knock has already been described in Section 2.2. Furthermore, engine speed is a very crucial operating characteristic so it's important to include it in the controller [20].

2. ***Spark Timing [deg]***

Spark timing has also been mentioned in Section 2.3, as the most common variable that is varied in an effort to prevent knock.

3. ***EGR Position [deg]***

EGR position is proportionate to how much exhaust gases are present in the combustion chamber. The correlation between EGR and knock is quite clear if one considers how drastically EGR affects the mixture conditions inside the combustion chamber. In more detail, by increasing the amount of exhaust gases in the chamber, the specific heat capacity of the intake charge is increased and as a result the peak temperature during combustion is reduced. Along with the spark timing, they are the most critical variables, that can be altered to prevent knock [20] [21].

4. ***Throttle Position [%]***

It is known that throttle affects the amount of air entering the combustion chamber and in a sense the engine load. The effect is even more evident in our case where the engine studied operates on stoichiometric conditions. Again, in Section 2.2, the impact of engine load on knock occurrence is mentioned [21].

5. Wastegate valve position [%]

The wastegate valve is responsible for the amount of exhaust gases going into the turbocharger. So, in other words it indirectly controls the resulting boost pressure and thus it affects the combustion conditions. If the wastegate is completely open [100 %], the lower boost pressure will reduce the chances of knock occurring, due to the resulting lower compression pressure [8] [20].

3 Controller Development Background

In this chapter the most important components/ aspects of the development process are presented. At this point it is important to give some background information on controllers and MPC controllers specifically, how they are structured and defined. Then a brief background on Artificial Neural Networks is presented and why they are relevant to the scope of this thesis.

3.1 Control Theory

Control systems have been implemented on a large variety of fields from automotive operations to the household heating system. There are a lot of details and parameters that make up a control system, and some aspects may differ depending on the application. In this chapter, the main components and variables that are encountered in a control system are described.

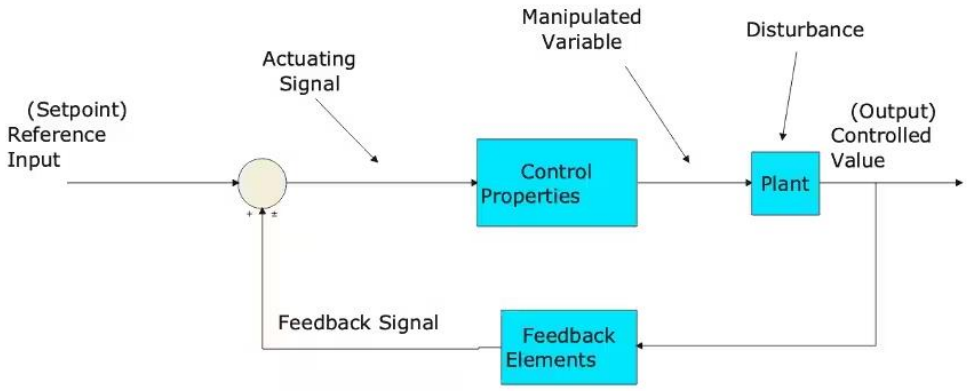

Figure 3.1 Control system outline *[22]*

The main variables are:

- **State variables/ Control variables:** They are used to represent the mathematical "state" of the dynamic system and what the system aims to control [23] [24].
- **Manipulated variables/ Control moves:** They are the variables that the controller alters in order to minimize the cost function. In other words, the manipulated variables are altered to affect the value of the state variables.
- **Disturbances:** They are extra inputs that affect the operation of the controller and the calculation process. Their value is predetermined and cannot be affected by the controller. They can be further divided into two categories:
  - *Measured:* This type of disturbance can be directly measured and compensated for.
  - *Unmeasured:* Accordingly, this type of disturbance cannot be directly measured but it still affects the system [25].
- **Plant/ Process:** The mathematical representation of the system or device that is under control.

- **Outputs:** They are the result of the plant process. They can either be the systems states or some expression combining the system states and inputs [22].
- **References:** They are the desired values that the controller is trying to achieve for the system outputs.
- **Sampling time:** The rate at which the system receives its inputs, disturbances, performs computations and produces outputs.
- **Constraints:** They represent other limitations that the controller needs to consider while optimizing the problem and minimizing the cost function [23].

3.2 Model predictive control

Control has been playing a crucial role in ICE operation, especially in automotive applications. Through the ECU, a large variety of actions are controlled such as injection and ignition timing, the EGR, monitoring and diagnostics etc. Traditionally all these actions are achieved by feedback control often coupled with feedforward control to appropriately portray and take into account the effect of the various operating points and parameters. In order for that to process to succeed, two-dimensional look-up tables have been utilized or in other words, calibration maps. There are two types of actions that need to be successfully fulfilled. Firstly, the feedforward control aspect enables the rapid change between operating points and secondly the feedback control aspect performs the error correction. Considering the ever-stricter legislations on emissions and the desire to keep decreasing fuel consumption, the calibration process of these controllers tends to be strenuous and time consuming. A very promising alternative to this issue could be the MPC controller [6] [26].

MPC has proven to be very effective when it comes to solving nonlinear, highly constrained problems and that makes it a great fit for ICE issues. Traditional controllers, seem unable to effectively deal with the nonlinear and seemingly stochastic dynamic behaviour of the phenomena that take place in an ICE. Firstly, as it has already been mentioned, the calibration process can be quite challenging. Furthermore, the time-consuming aspect can lead to compromises when it comes to the efficiency of the controller and despite that it cannot be guaranteed that the controller will be effective on a wide range of operating conditions [26] [27]. On the other hand, MPC seems to be more suited to this type of problems due to offering the following advantages:

- Implicitly considers constraints on state, input and output variables.
- Provides closed loop control performance and stability for the optimal problem with constraints.
- Exploits the use of a future horizon when while optimizing the current control law.
- Offers the possibility of both offline and real time implementations.
- Provides the capability to handle uncertainty in the system's parameters, delays, and non-linearity in the model [26].

The main target of MPC as it has been already stated, is to calculate a future profile for the manipulated variables u, so as to optimize the behaviour of the plant according to the targets set for the output y. The future profile for the manipulated variables is a finite time window defined as the horizon. Firstly, there is the prediction horizon, which is basically how far into the future the controller will attempt to optimise the plant behaviour. Then, there is the control horizon, which is a subset of time steps and it's the time over which the controller decides the control variables that will be applied. A general guideline is that the control horizon value is around 10-20% of the prediction horizon. The reference value is the target that has been set for the output y of the system [24] [26] [27].

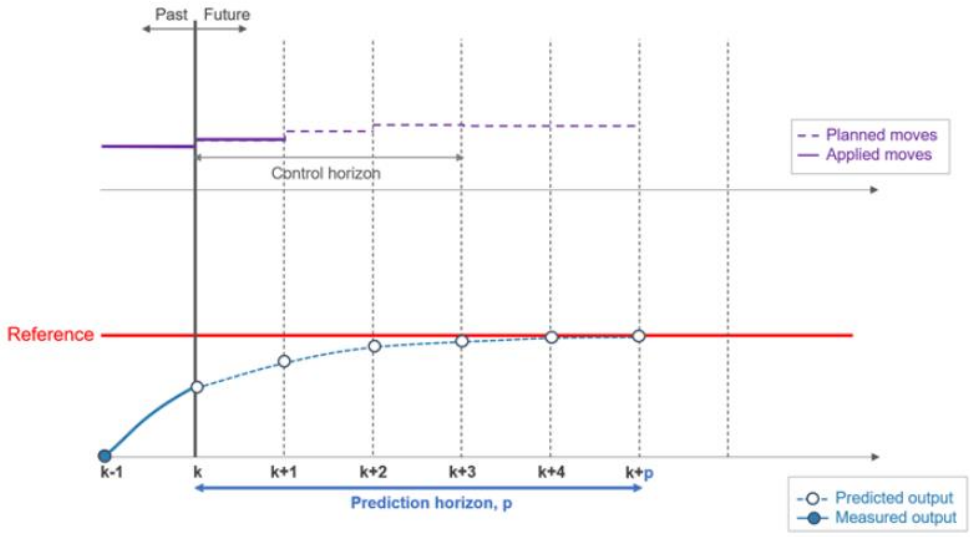

Figure 3.2 Prediction and control horizon in MPC *[24]*

The selection of the prediction horizon depends mostly on the type of problem that is being optimized. If the system has highly unpredictable dynamic behaviour with high disturbances, it's better to select a shorter prediction horizon. However, when that is not the case and the system behaves in a more "predictable" way, then by increasing the control horizon, the performance of the controller is expected to increase too. At the same time though, the computational time will increase as well [24]. So there has to be a balance between performance and computational time when selecting the values for the prediction and control horizon.

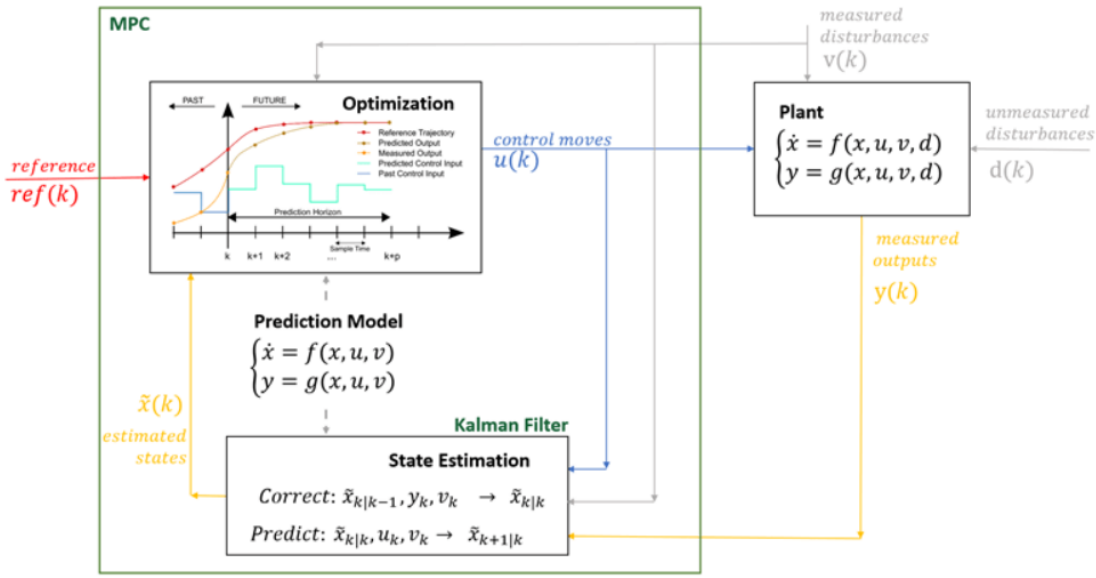

Figure 3.3 MPC controller basic outline *[24]*

Apart from the control and prediction horizons, some other important aspects of MPC are the state function and the cost function. The state function's role it to estimate the states of the system over the prediction horizon, in a way that will minimize the defined cost function. The state space model is the mathematical representation of the system as a set of input, output and state variables. The state space model is also mentioned as the prediction model. The cost function is essentially what the controller is trying to achieve. It can be either minimizing or maximizing the difference between the states and the reference value or it can include extra constraints like the manipulated variables having a specific minimum or maximum value. The cost function is part of the optimizer aspect of the controller [24] [27].

3.3 Artificial Neural Networks (ANNs)

An Artificial Neural Network is defined as:

"An adaptive system that learns by using interconnected nodes or neurons in a layered structure that resembles the human brain".

These types of networks learn by examples that are given to them through data. In other words, by having a set of data that include the necessary inputs and outputs, the network can be trained to recognize the patterns that link the inputs with the outputs (e.g. pattern recognition, data classification etc.). What makes ANNs so useful is that firstly, the user does not need to define the relationships between the data (e.g. in the form of equations) and secondly that they have been proven to provide high accuracy even in the most complex of problems. This process is called adaptive learning [28]. The network has the ability to "learn" how to perform the required task based on the data given for its training. So, for problems that are highly nonlinear and dynamic, they can be a great option. They can be implemented for a great variety of purposes from simple data classification to image processing [28] [29].

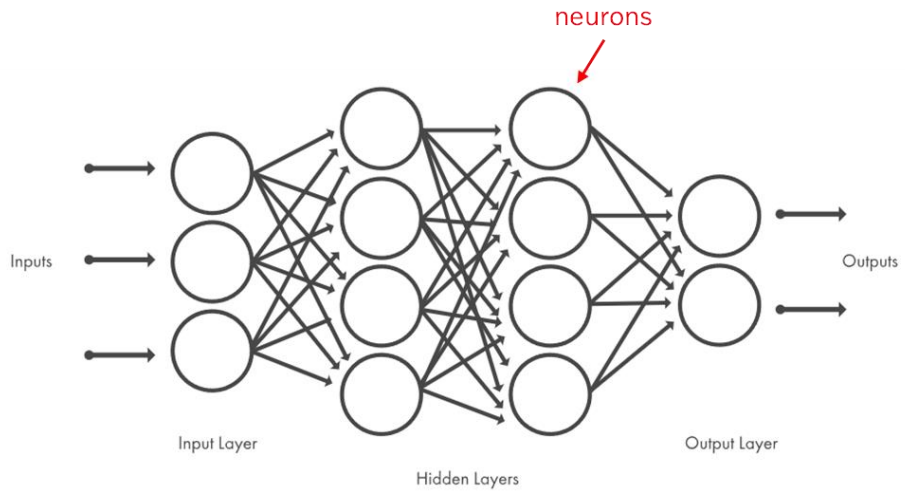

Figure 3.4 Outline of ANN structure *[30]*

The most important components that make up an ANN, and are crucial to understanding how they operate, are:

- **Inputs:** The information/variables that the network requires to perform its calculations.
- **Outputs:** The results of the network after the calculations.
- **Layers:** They are classified as input layer, output layer and hidden layers. Each layer comprises of a specific number of neurons.
- **Neurons:** The artificial neuron is developed to operate in a similar way to a biological neuron. It can accept a number of different signals/inputs and process them in a predefined way. While the neuron can accept a multitude of inputs, it only has one output, as illustrated in Figure 3.5.

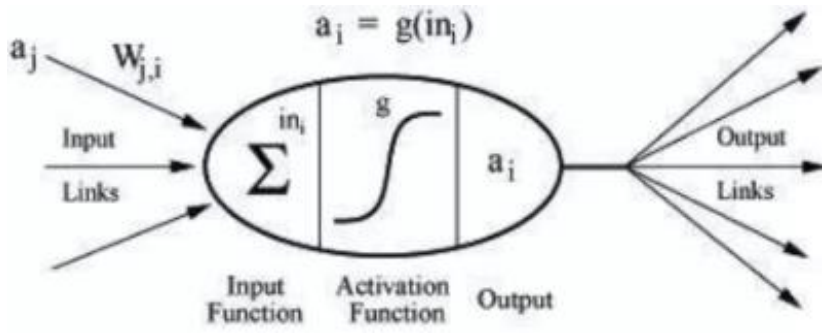

Figure 3.5 Artificial neuron outline *[31]*

The neuron and the entire network as a result, operates in two modes. Training and testing mode. Each neuron input has a weight applied to it. All the inputs, multiplied by their corresponding weights, are summed. Then the neuron applies its activation function on the sum and that result is the neuron output. This output is either forwarded to another layer of neurons or it's the network output. During the training phase, the weights of the inputs are changed to achieve the desired output. After the training, during the testing, the weights of the inputs remain unchanged [29].

- **Activation function:** It is an indicator on whether the specific neuron will be activated or not. In other words, it is decided whether the output of the neuron will contribute to the output of the process or not. There are different types of activation functions as illustrated in Figure 3.6.

- **Batch:** The amount of data from the training dataset that go through the network before updating its parameters.
- **Epochs:** The training of the network with all of the training data for one full cycle.
- **Iterations:** Each iteration is the number of batches required to complete one epoch.
- **Learning rate:** How quickly the update of the parameters is happening. A very low learning rate has higher chances of converging but makes the training process computationally heavy. On the other hand, a large learning rate can speed up the process, but it is possible that it will not converge [29] [30].

The number of neurons and hidden layers, batch size, number of epochs, learning rate and activation functions are also known as the networks hyperparameters. Hyperparameters carry information crucial to the networks structure and training process.

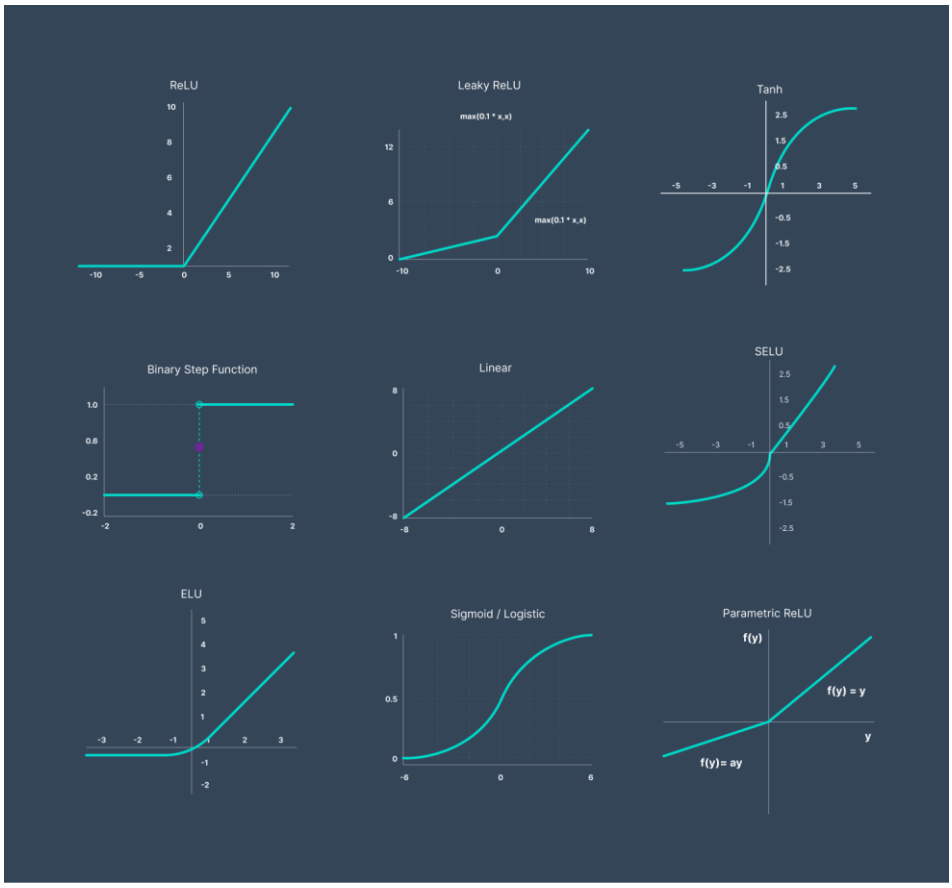

Figure 3.6 Different types of activation functions *[32]*

The main advantage of ANNs is that they can successfully perform operations that linear networks are unable to. In order to do that, it is better to have large quantities of data for the training process. The length of the training process can be controlled by defining the number of epochs [28]. Choosing the number of epochs can be a very delicate process. By having too few epochs, the network may not have enough time to accurately figure out the patterns and achieve a good level of accuracy. This is called underfitting. On the other hand, by having a very large number of epochs "overfitting" can happen. Overfitting means that the network has become "too specific" and has lost its ability to generalize [30].

As it has already been mentioned, ANNs are able to portray complex phenomena with high accuracy while avoiding the time-consuming process of using equations. It has also been stated, that for the prediction component of a controller, a state space model is required to accurately portray the selected system. When talking about knock, it has been established that it is really a highly nonlinear, multifaceted issue. So, within the limits of the thesis scope, it has been investigated how ANNs could be utilized to make the controller more efficient and lead to accurate results. More details on the structure of the controller are available in the next chapter.

4 Controller Structure & Outline

For this Section of the report, the structure of the controller is defined based on the concepts explained in Chapters 2 and 3. The main components that will be described are illustrated in Figure 4.1, which is a preliminary sketch of the controller.

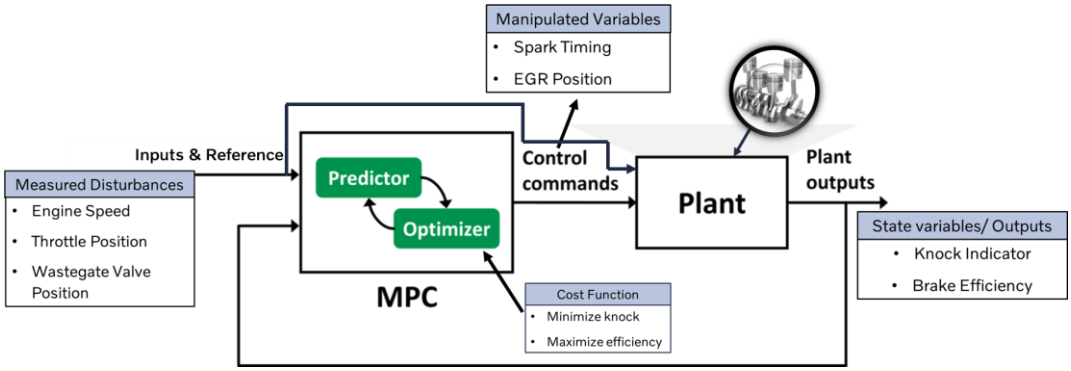

Figure 4.1 Controller structure

4.1 Control Variables

The actuators that have been selected for the development of the MPC controller have been extensively explained in Section 2.4. Furthermore, the main variables that affect the control process have been stated in Section 3.1. Based on that information the most important controller variables, for this thesis, are:

- **State variables/Outputs:** Knock Indicator and Brake Efficiency. The Knock Indicator was logged from the engine simulation model. In GT-Suite the Knock Induction Time Integral is available. This coefficient has a value of zero when there's no knock occurrence. Then from values from zero until just below one, there is still no-knock occurrence, but the higher the value the closer the engine is to knocking. Lastly, if the value of the coefficient is one or above, there is knock occurrence and the higher the value the more sever the knocking is.
- **Manipulated variables:** Spark Timing and EGR position. In other words, the controller will attempt to prevent knock from happening by altering the spark timing or the position of the EGR valve.
- **Measured disturbances:** Engine Speed, Throttle Position and Wastegate Valve Position. All three of these variables, are directly related to knock occurrence and therefore are considered for the control process. However, none of these variables are altered, hence they are treated as disturbances.

4.2 Plant: Engine Simulation Model

The plant is defined as the system that is under control. In this case, the system under control is the engine simulation model. The studied engine has been simulated in GT-Suite. The engine's specifications are the following:

- **Engine Type:** Spark Ignited Methane Gas Engine
- **Displacement:** 17 Litres
- **Number of cylinders:** 6 (in-line)
- **Combustion specifications:** Stoichiometric combustion (λ=1)
- EGR
- Three-way (TWC) catalyst

The simulation model, as seen in GT-Suite, is presented in Figure 4.2.

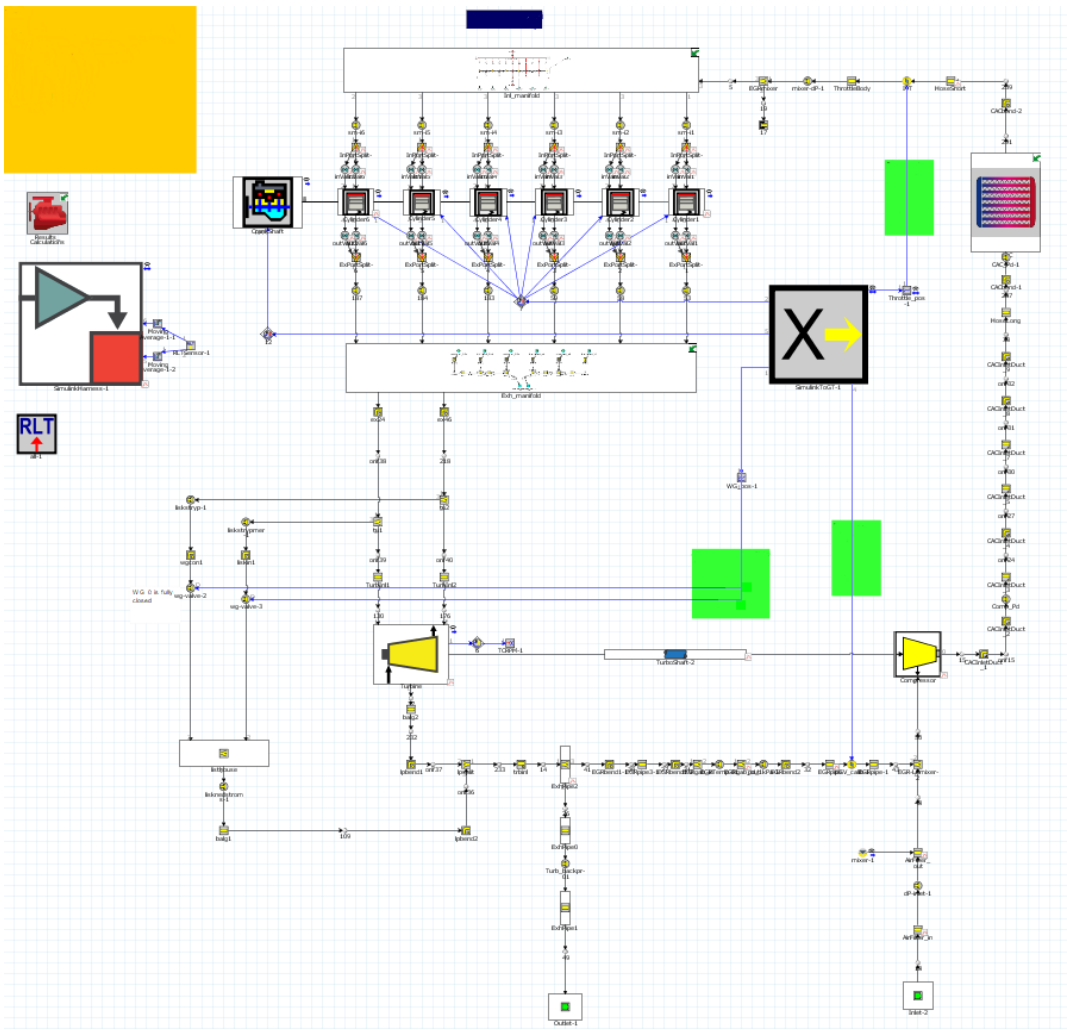

Figure 4.2 Engine model in GT-Suite

The controller and thus the engine simulation are implemented through Simulink. According to Figure 4.1, the simulation model, apart from the manipulated variables that the controller chooses, also receives the measured disturbances as inputs which are user defined. While this choice may not reflect actual engine operating conditions, it was deemed as a necessary first step to establish the proper operation of the controller and verify its effectiveness.

4.3 MPC Controller: State Space model

For the state space model, a neural state space model has been trained, to identify knock occurrence and the corresponding efficiency at different operating conditions. The entire process of defining the network and the training process, along with the results are explained in Sections 4.3.1-4.3.3.

4.3.1 State space model definition

Starting by the network structure the necessary inputs and outputs must be defined. The state space model's inputs and outputs have to correspond to the control variables mentioned in Section 4.1. Considering all that, the specifications for the neural state space model are given in Table 4.1.

Table 4.1 Neural State Space Inputs and Outputs

Inputs	Outputs
Engine Speed	Knock Indicator
Spark Timing	Brake Efficiency
EGR Position	
Throttle Position	
Wastegate Valve Position	

Before the training process, it is necessary to define a range for all the aforementioned variables. The range needs to reflect the actual operating conditions under which the engine will be operated. For each of the input variables the ranges are given in Table 4.2.

Table 4.2 State Space Model Inputs Range

Variable	Range	Unit	Further Specifications
Engine Speed	1300-1900	rpm	-
Spark Timing	-50-0	deg	-
EGR Position	0-100	%	0 indicates fully closed. 100 indicates fully open.
Throttle Position	5-100	%	0 indicates fully closed. 100 indicates fully open.
Wastegate Valve Position	0-100	%	0 indicates fully closed. 100 indicates fully open.

When it comes to the outputs, to expedite the training process some limitations have been set. For the Knock Indicator (Knock Induction Time Integral) while it has been mentioned that it does not have a maximum value, one has been set. All the values that exceeded ten, where changed to ten. Accordingly, when it comes to Brake Efficiency, the simulation model can sometimes produce negative efficiency results. Since this does not reflect realistic operating conditions, all the negative values were replaced by zero values.

4.3.2 State space model training process

The training process has been briefly described in Section 3.3. Based on that, it is important for the neural state space training, to have training data that actually reflect the relationship between the inputs and the desired outputs. In this case, what needs to be assessed is how the engine speed, spark timing, EGR position, throttle position and wastegate valve position affect the knock indicator and the brake efficiency. In that direction, with the help of the simulation model, some training profiles we produced, ranging all of the variables from their minimum to maximum value and using their results for training the network. An example of those training profiles is illustrated in Figure 4.3 [33].

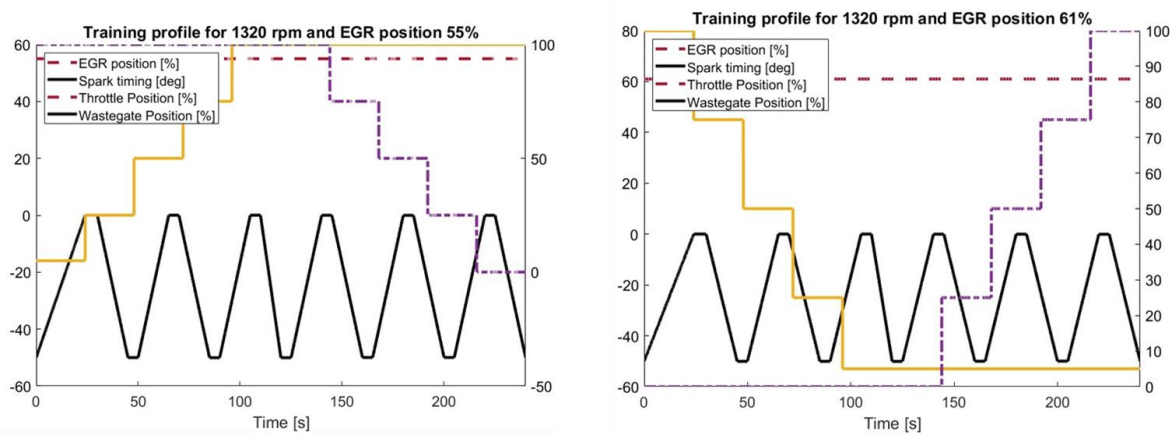

Figure 4.3 Training Profiles for Neural State Space

While in those figures it seems like the engine speed and EGR position are kept constant, there are corresponding training profiles for each of their values in the range that has been specified. The engine speed has been varied in increments of 50 rpm from 1300 rpm to 1900 rpm. The EGR position has been varied in increments of 12, starting from 1 up to 78. They could be considered as profiles that describe "steady state" operating conditions, because while the variables are changed, they are kept constant for specific increments of time. In total these profiles were compiled to 13 bigger files each having a length of 1920 s. From those profiles the 80% was used for the training process and 20% was used for testing.

When it comes to the training process, the data needs to be pre-processed before they are actually used for the training. All of the data needs to be rescaled in order to improve convergence and training stability. Also, it is preferred that all

of the variables have similar scales so as to prevent certain features from dominating the learning process. For example, in that case the engine speed could dominate over other inputs such as the spark timing due to its larger magnitude. To prevent that from happening all of the variables were scaled in a range from zero to one. Accordingly, the engine speed was rescaled to zero for the 1300 rpm and one for the 1900 rpm. In the same spirit, the spark timing was zero for -50^{o} and one for 0^{o}. The same process was followed for all the inputs and outputs. Lastly the sampling time for all the data was defined as 0.01 s.

The hyperparameters for the neural state space and an outline of the network are illustrated in Table 4.3 and Figure 4.4.

Table 4.3 Hyperparameters of Neural State Space

Number of Hidden Layers	2
Neurons (per layer)	150
Activation Function	tanh
Epochs	200
Optimizer	Adam
Bias Initializer	zero

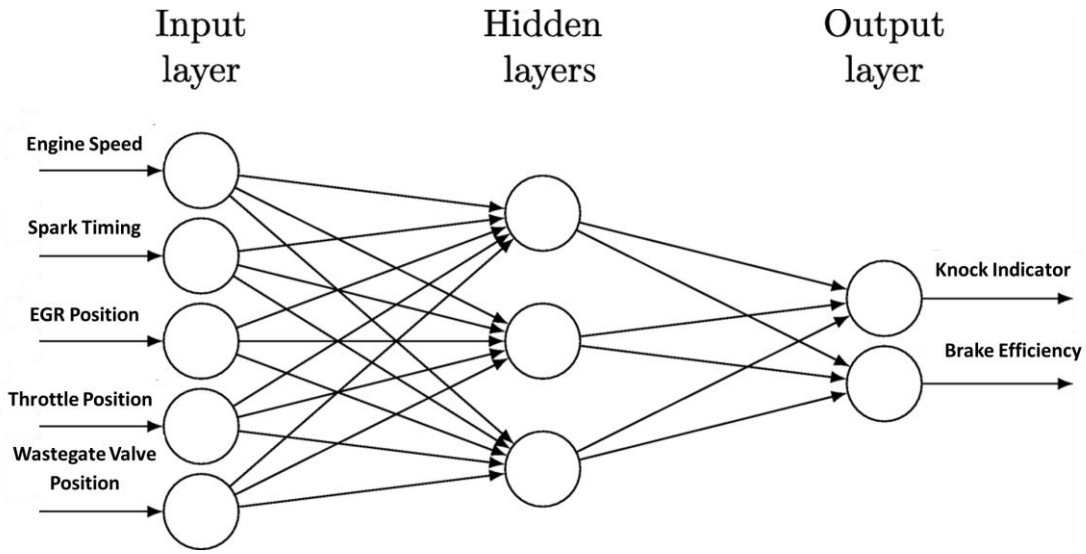

Figure 4.4 Neural State Space Outline

4.3.3 State space model results

The last stage for the neural is to test how accurate the training process was. As it is already clear from Section 4.3.2, 20% of the data were not used for training. This part of the data is kept to be used on the neural state space and actually

assess how it will respond to brand new data. The accuracy of the network when it comes to predicting knock and brake efficiency is illustrated in Figure 4.5.

For both outputs, the values predicted by the network correlate closely to the actual values, within the tolerances set for the project. For the Knock Indicator the accuracy is 89.8% and for Brake Efficiency the accuracy is 95.3%. The results presented in Figure 4.5 are normalized with the same method mentioned in Section 4.3.2.

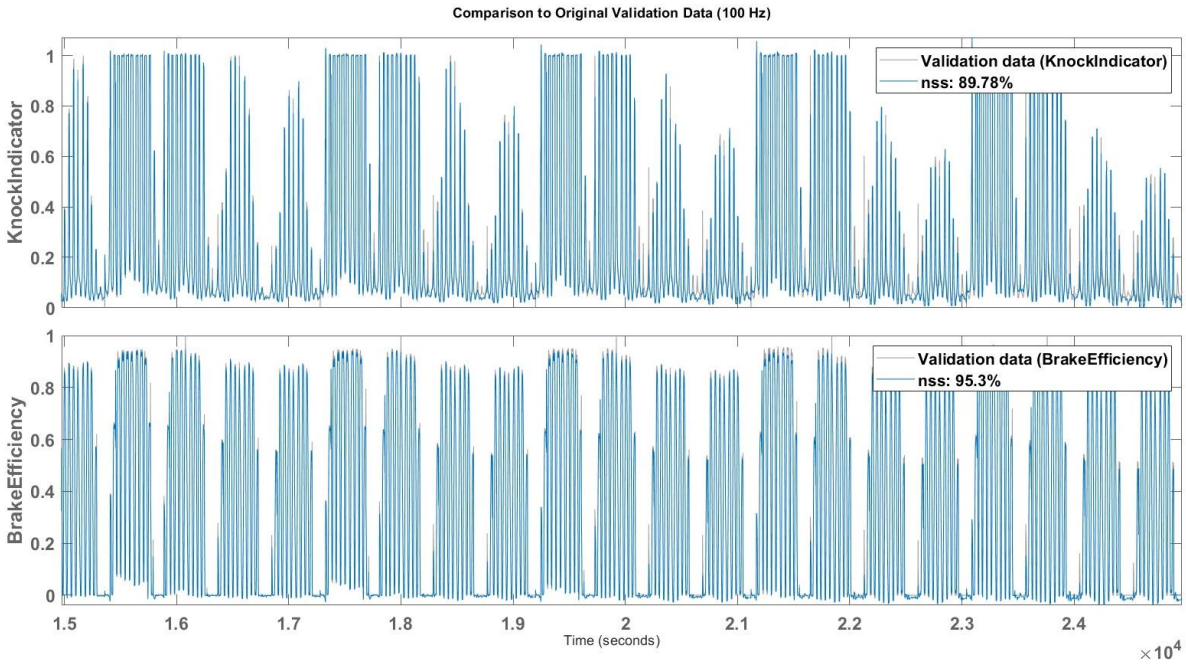

Figure 4.5 Neural State Space Testing Accuracy

4.4 MPC Controller: Setup in Simulink

Most of the crucial components of the controller have been extensively described in the previous sections. In this section some important aspects of the controller setup in Simulink will be described so as it is more enlightening for the cost function definition and the discussion of the results. The outline of the controller in Simulink is given in Figure 4.6.

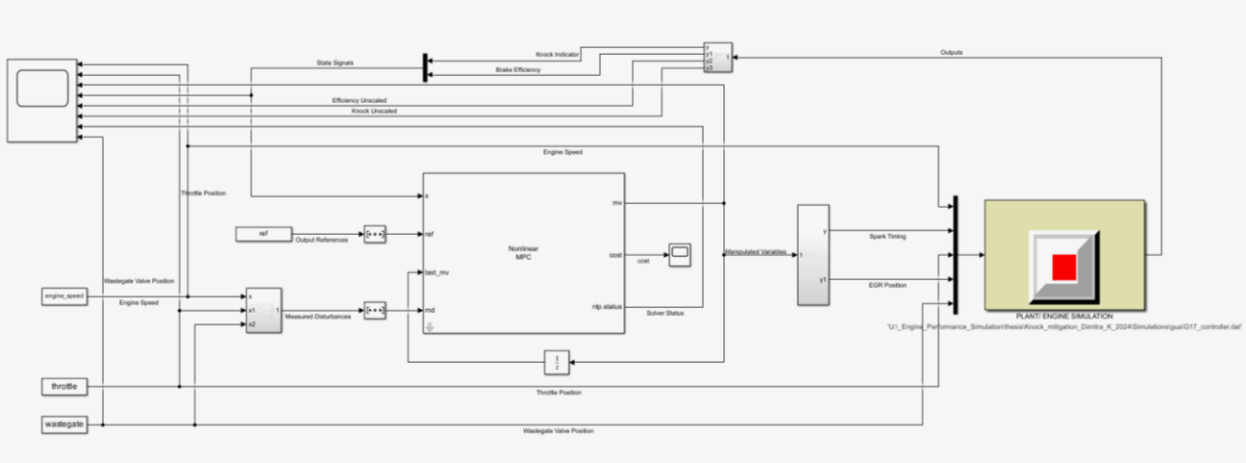

Figure 4.6 Controller in Simulink

The specifications that need to be defined are:

Sampling Time

The sampling time in accordance with the training process has been set to Ts= 0.01 s.

Prediction Horizon

The prediction horizon is the time over which the MPC controller is called to "predict" the system's behaviour. The prediction horizon is set to p=50. Accordingly, the duration of the prediction horizon is calculated as:

$$p \cdot T_s = 50 \cdot 0.01 = 0.5\ seconds$$

Control Horizon

The control horizon is defined as the number of manipulated variables moves to be optimized in one interval. According to Section 3.2, the control horizon should be 10-20% of the prediction horizon. Considering that, the control horizon was chosen to have a value of 10.

State Variables

The states of the system have already been established as the Knock Indicator and Brake Efficiency. They are also the outputs of the system. In terms of the controller the states are defined as:

$x_1 \rightarrow$ Knock Indicator
$x_2 \rightarrow$ Brake Efficiency

The minimum and maximum values for both states have been set as zero and one in accordance with the rescaling logic explained in Section 4.3.2.

Manipulated Variables

The manipulated variables that have been selected are the spark timing and the EGR position. Again, these variables have been rescaled to range from zero to one.

Another important parameter that is important for the results cases is the rate of change for the manipulated variables. The rate of change limits how much a manipulated variable can be increased or decreased over a time step. The rate of change has been varied to study its effect on the results of the controller.

State Function

The state function is the neural state space model that has been defined in Section 4.3.

The only aspect of the controller that is yet to be defined is the cost function which will be analyzed in Section 4.5.

4.5 MPC Controller: Cost Function

The cost function is the part of the controller in Figure 4.1 that is a part of the optimizer. It is the target of the controller and what it is trying to achieve. At each sampling time the cost function is minimized, across the set prediction horizon. Essentially, the cost function estimates the systems trajectory based on the current state of the system, which is used as an initial condition. The most important aspects of the cost function are:

- Satisfying the optimization objective. By setting the cost function, a target is set for the controller. In other words, it is a tool to assess the controller performance and the corresponding control functions. The manipulated variables are adjusted in a way that leads to the desired system behaviour.
- A cost function can have conflicting targets or targets that directly affect each other. For example, in an ICE the cost function may try to maximize efficiency while also that the same time minimizing fuel consumption. Each term in the cost function represents a different goal.
- These targets can be rated according to their importance by adding weighting factors. The higher a weighting factor is for a term; the more priority is given to the goal that it represents. On the other hand, if a term has a lower weighting factor that means that it represents a secondary target.

- Lastly, the cost function may include constraints. These constraints can be about the minimum or maximum value an input can have, bounding the states or anything that the user feels needs to be constrained. The violation of these constraints will lead to an increased cost and as a result lead to poor controller performance. Generally, the constraints can be either "hard" or "soft". Hard constraints must never be violated, whereas soft constraints are more flexible. It is crucial for the controller to not be over-constrained to ensure proper function [26] [27].

In nonlinear MPC it is possible to define a custom cost function according to the systems characteristics. For the purpose of this thesis, a custom cost function has been applied to the controller. Its general form is:

$$J(k) = w_1 \cdot \sum_{i=0}^{p-1} x_1(k+i) - w_2 \cdot \sum_{i=0}^{p-1} x_2(k+i)$$

Where:

w_1→ Weight applied to Knock Indicator Term

x_1→ Knock Indicator

w_2→ Weight applied to Brake Efficiency Term

x_2→ Brake Efficiency

The first term represents the controller's goal to minimize the knock indicator, while the negative second term indicates the goal to maximize the brake efficiency. For the next section different weighting factors have been applied to the cost function to assess its behaviour.

5 Results

Having covered all the controller's characteristics in the previous sections, the next step it to verify that the controller is effective and achieves the set targets. In Section 5.1, the controller is operated with some of the training profiles under steady state conditions to assess its performance. In Section 5.2, the controller is operated without preset inputs, which reflect real life operating conditions more accurately.

In both cases the brake efficiency values are scaled to vary from 0 to 1. The maximum scaled value (1) does not indicate a brake efficiency of 100% but instead it represents the maximum brake efficiency encountered during the training process of the state space model (i.e. the maximum brake efficiency encountered during the examined operating conditions). So accordingly, a scaled brake efficiency of 1.2 does not mean that the efficiency is 120%. This is necessary due to the confidentiality of the data as mentioned in Section 1.5.

5.1 Steady State Operation

All of the cases that were investigated have the same inputs. There have been also further combinations, which have been evaluated (e.g. higher engine speed) but the trends were similar so it would be redundant to present every single one of them. The most interesting cases are the ones presented because they are closer to how the controller will be used, at least on an initial stage. As it will be made clear also from Section 5.2, the engine is operated under constant engine speed, varying throttle and constant wastegate value.

Case 1

For this case, a weighting factor has been applied to the efficiency term of the cost function:

$$J(k) = \sum_{i=0}^{p-1} x_1(k+i) - 3 \cdot \sum_{i=0}^{p-1} x_2(k+i)$$

Furthermore, the EGR rate of change, as mentioned in Section 4.4 has been set to 0.3. The input profile for the measured disturbances is illustrated in Figure 5.1.

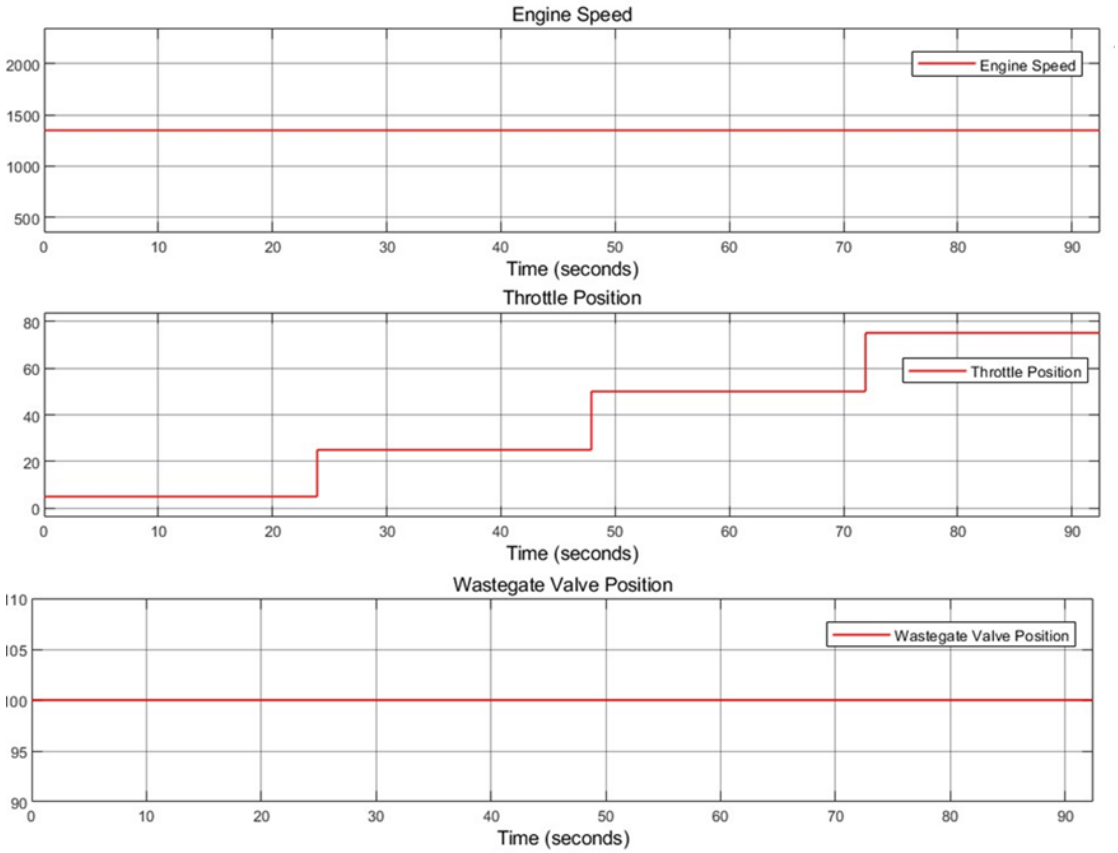

Figure 5.1 Input Profiles- Case 1

The engine speed is kept constant at 1400 rpm and the wastegate valve is fully open. The throttle position is varied from fully closed to fully open in intervals of 24 seconds.

Starting by the manipulated variables, in Figure 5.2, the controller corresponds quite well to the changes in the throttle position until around 70s. The spark timing is kept constant up until that time point and the EGR position while it varies it has a steady behaviour. However, when the throttle is increased to 75% the controller becomes slightly unstable for almost 10s before it settles again.

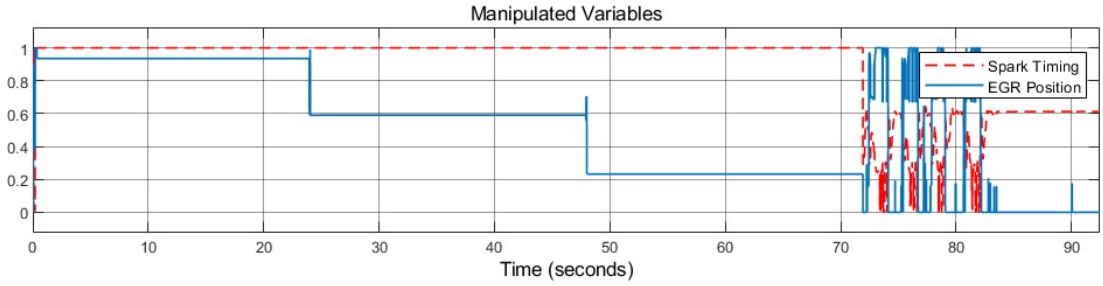

Figure 5.2 Manipulated Variables-Case 1

The same trend is noticed for the state variables, in Figure 5.3. For the lower values of throttle position a negative efficiency is noticed, which essentially means that the brake efficiency is zero. However, after the throttle is increased to 80%, the efficiency goes up. The knock indicator is at all times under the critical value of one, which means that there is no knock occurrence. Even at the time section where the controller is slightly unstable, the knock is still very low.

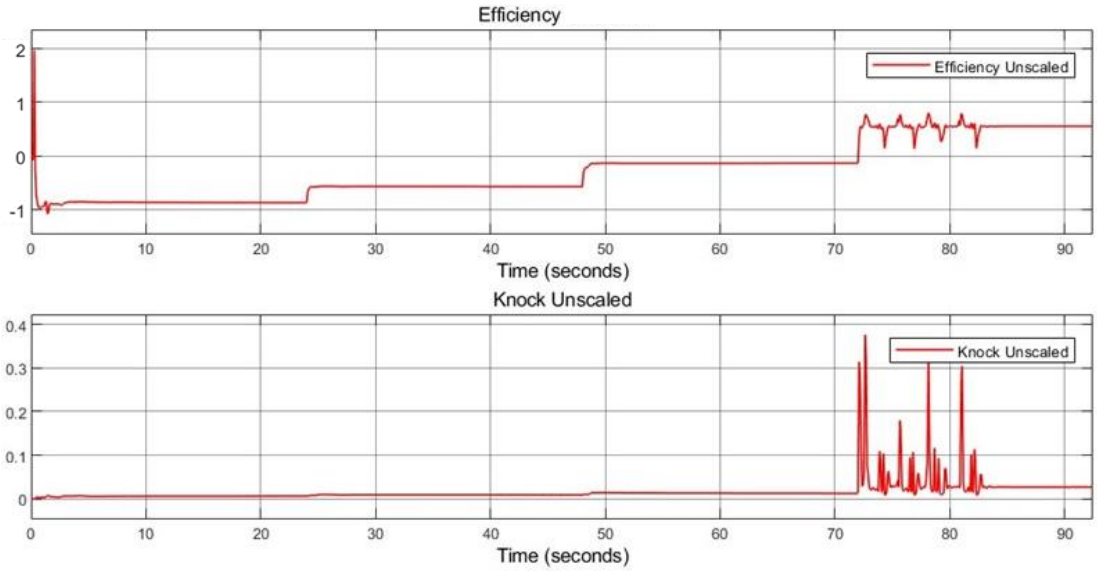

Figure 5.3 State Variables- Case 1

Case 2

For this case, a weighting factor for the efficiency term of the cost function remains unchanged:

$$J(k) = \sum_{i=0}^{p-1} x_1(k+i) - 3 \cdot \sum_{i=0}^{p-1} x_2(k+i)$$

On the other hand, the EGR rate of change, has been altered to 0.1, imposing a slightly stricter constraint. The input profile for the measured disturbances is the same as in Case 1 and is illustrated in Figure 5.4.

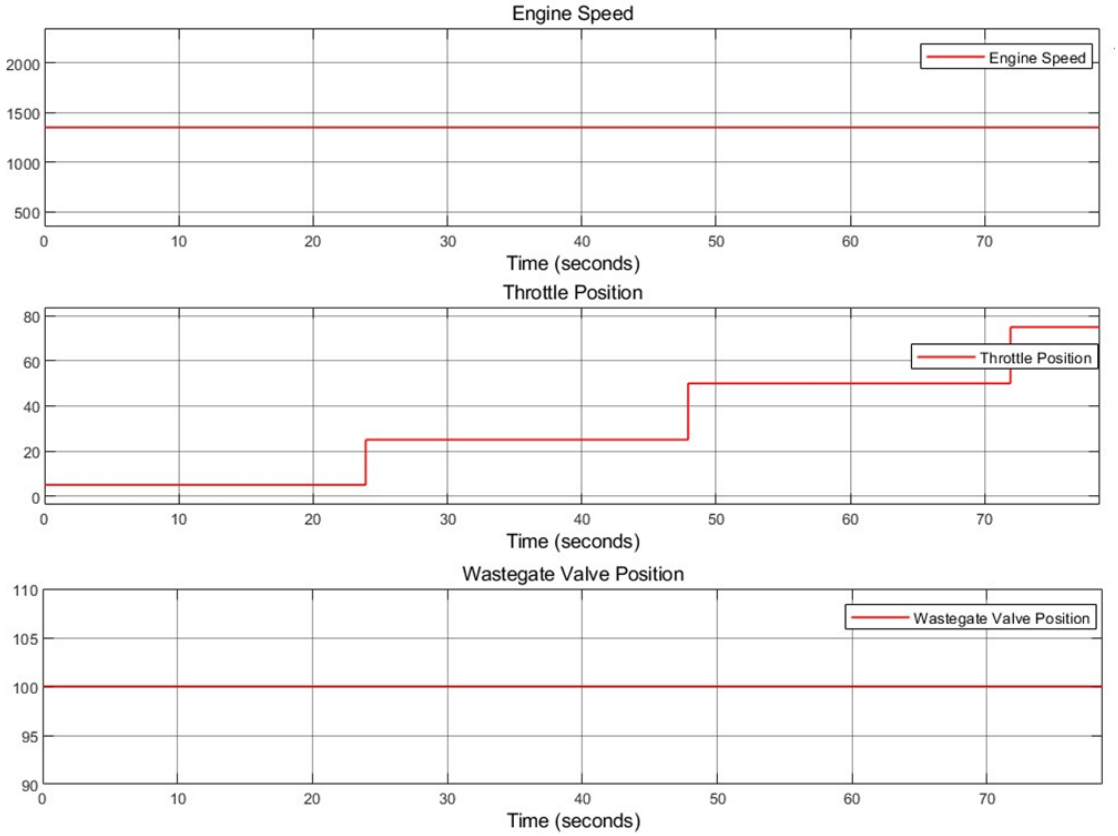

Figure 5.4 Input Profiles- Case 2

Again, starting with the manipulated variables, in Figure 5.5, there is a similar trend to Case 1, although this time the controller hardly becomes unstable. Furthermore, in this case although the inputs are the same, due to the stricter constraint, the spark timing is not kept at the same value throughout 80 s of operation. It is obvious that even though the constraints are stricter, the controller responds better.

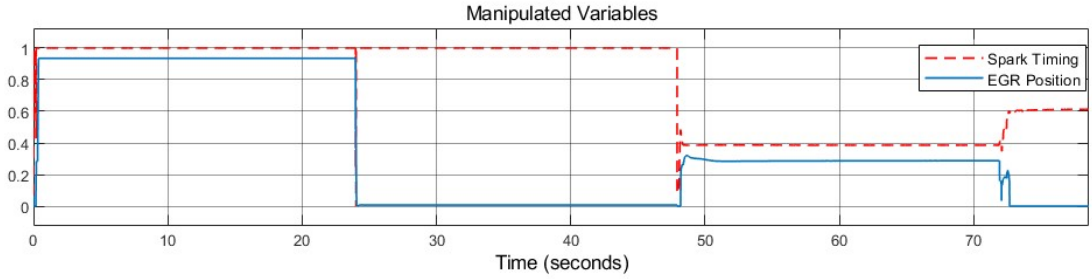

Figure 5.5 Manipulated Variables- Case 2

The same conclusions can be made from looking at Figure 5.6 , the state variables. The brake efficiency has improved from Case 1. The efficiency increases up to 0.4 from 48s in comparison to Case 1 where the efficiency reaches the same value after 73s. The knock indicator is well below the critical value for the entirety of the experiment.

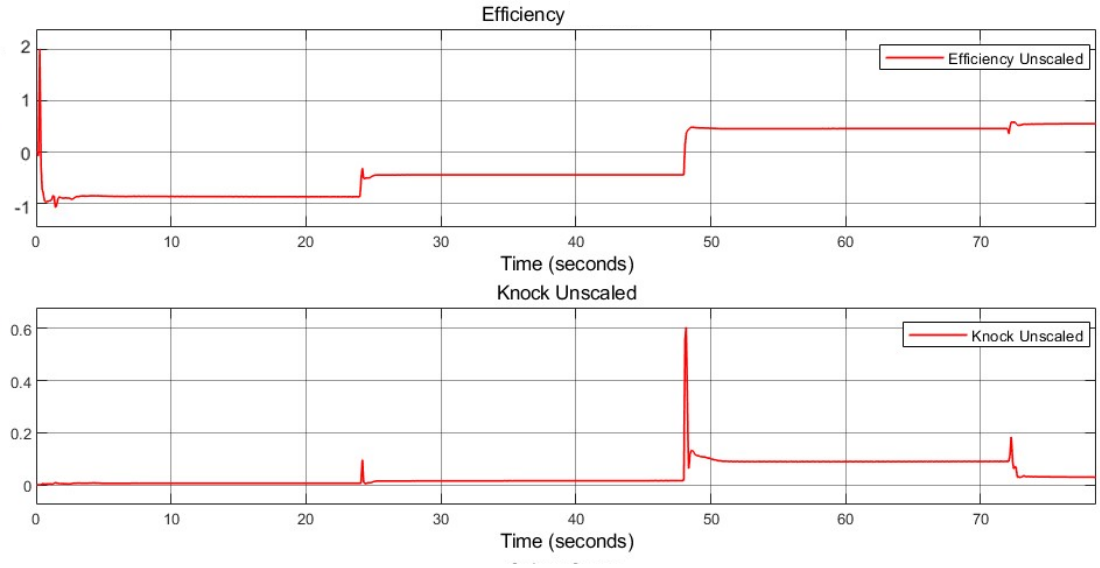

Figure 5.6 State Variables- Case 2

Case 3

For this case, the weighting factor for the efficiency term of the cost function has been doubled. In that way, the controller will give even more "priority" to maximizing the efficiency:

$$J(k) = \sum_{i=0}^{p-1} x_1(k+i) - 6 \cdot \sum_{i=0}^{p-1} x_2(k+i)$$

The EGR rate of change, remains unaltered when compared to Case 2. The input profile for the measured disturbances is the same as the previous cases and is illustrated in .

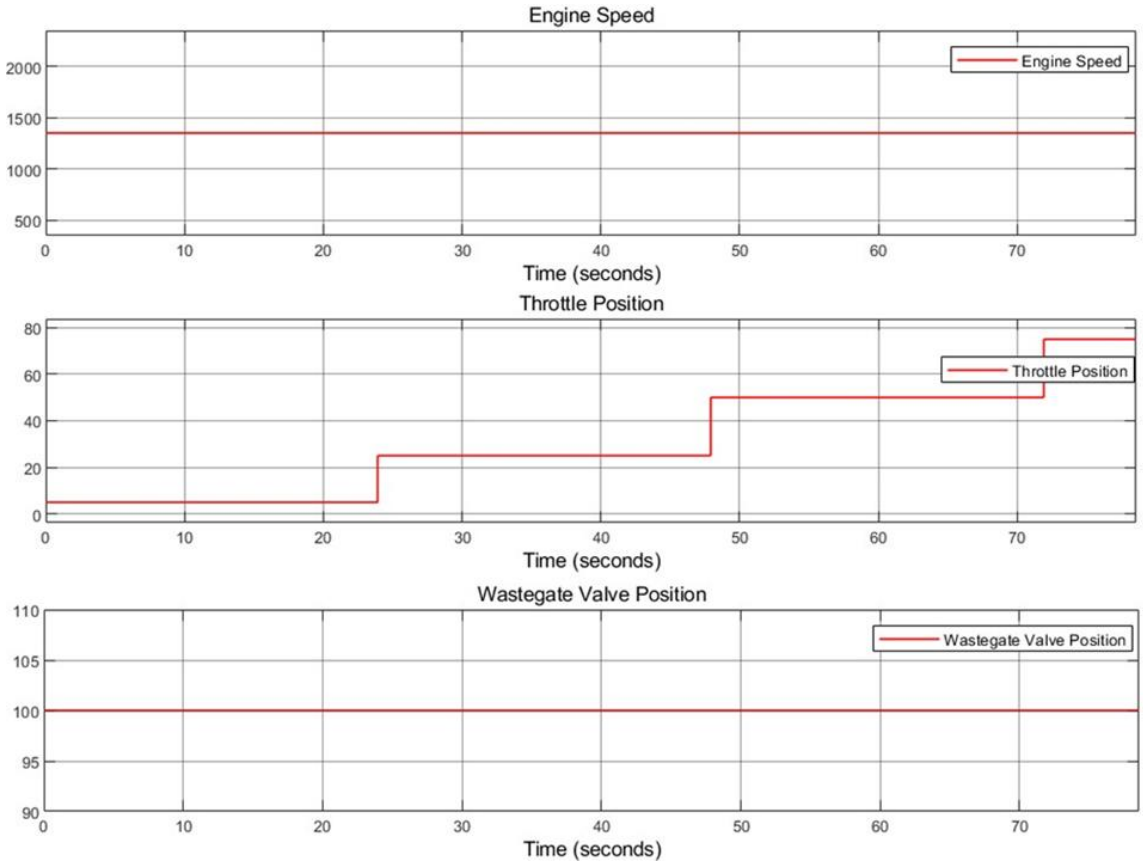

Figure 5.7 Input Profiles- Case 3

Starting from Figure 5.8, there are two important facts to be pointed out. First of all, the EGR position is barely changed, which indicates that the cost function-EGR rate combination maybe be too harsh, obstructing the controller from utilizing both manipulated variables fully. The spark timing behaviour is fairly similar to Case 2. Secondly, the controller becomes unstable again before the throttle is shifted from 50% to 75% and continues to be unstable while the throttle is at 75%.

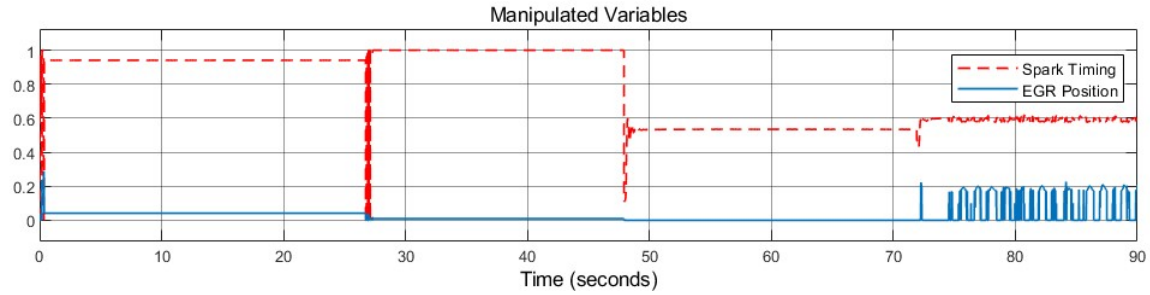

Figure 5.8 Manipulated Variables- Case 3

Moving on with the state variables in Figure 5.9, there are again two interesting facts that reflect the impact of the cost function- EGR rate dynamics. Starting with the knock indicator, its maximum value is 0.8. This value again is below 1 which indicates no knock occurrence; however, it is the highest value achieved when compared to the other two cases. The obvious reasoning for that, is that due to the cost function, the controller is mostly focused on increasing efficiency. That means that it can possibly "prefer" higher knock in favor of higher efficiency. Lastly, the efficiency after 70s is higher when compared to Case 2 but as it has already been mentioned before, the controller is slightly unstable.

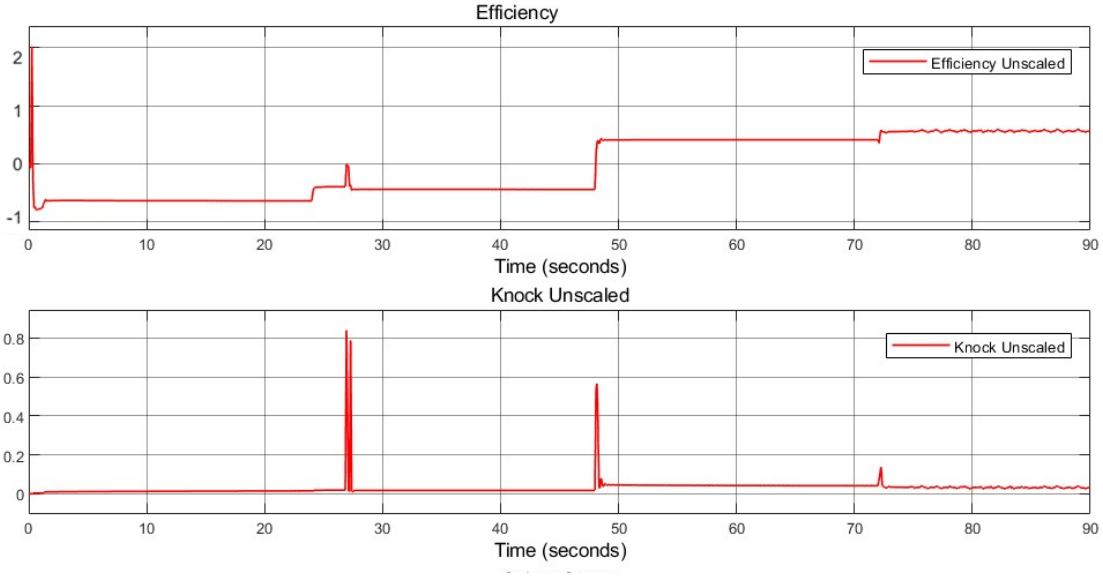

Figure 5.9 State Variable- Case 3

5.2 Transient Operation

In this section some structural characteristics have been changed to the controller's inputs. While the controller and its characteristics have remained unaltered, the measured disturbances are no longer constant values as in Figure 4.6. The changes are illustrated in Figure 5.10 and Figure 5.11.

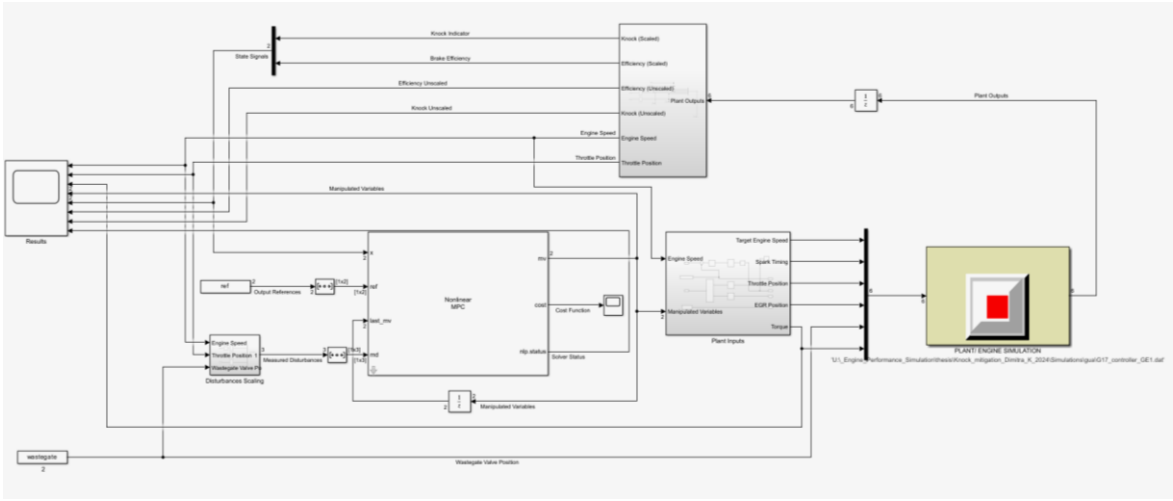

Figure 5.10 Updated controller in Simulink

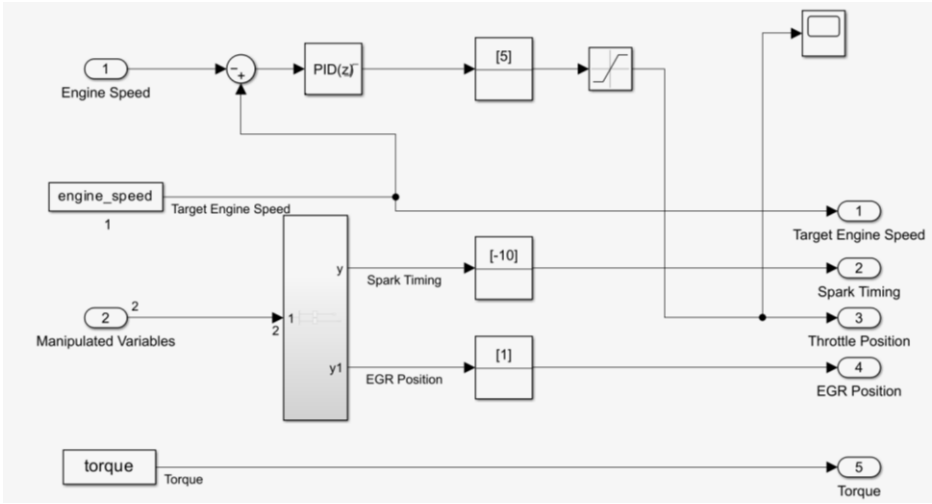

Figure 5.11 New controller plant inputs

Starting with the wastegate valve position, it has been kept as a constant value. The engine speed and the throttle position though, are given to the controller straight from the plant (i.e. the engine simulation model). The inputs to the engine simulation model are shown in Figure 5.11. The spark timing and the EGR position are given from the controller. The new additions are the torque parameter and the target engine speed along with a PID controller for determining the throttle position. The user now sets an target engine speed and a torque request to the engine and the throttle position is determined according to those two factors. The actual engine speed and throttle position from the engine simulation model are fed to the controller and so on. The torque request varies from 0 to 1. Here 1 represents the highest torque request value for the specific engine, which is confidential information.

5.2.1 PID Controller Tuning

The gains for the controller tuning where extracted from GT-Suite. For the proportional gain, the value is 1 and for the integral gain the value is 3. For small increments of torque, it has been noticed that the PID controller responds adequately. However, for larger increments of torque the controller becomes

somewhat unstable. So, all the results that are presented are obviously affected by this unstable behaviour of the PID controller. The effect will also be visible in the results figures.

5.2.2 Cases specifications

In this section, the setup for the cases that were tested is detailed. In all the cases the target engine speed is the same and equal to 1500 rpm. Furthermore, the minimum torque request at each case is 0.03. Also, the torque request is kept constant for 15s before it is increased. The torque steps are separated in two different cases. For the first case the torque request is increased by 0.045 every 15s and for the other case it is increased by 0.15. Again, these values are normalized and correspond to the actual torque request implemented. Also, the waste gate will be varied from fully closed to fully open to study its effect on the controller operation. In each result case, the specifics of the simulation will be mentioned.

When it come to the setup of the controller, the cost function for all the cases is:

$$J(k) = \sum_{i=0}^{p-1} x_1(k+i) - 6 \cdot \sum_{i=0}^{p-1} x_2(k+i)$$

When it come to the EGR rate that has been specified in cases 1-3, for the new cases it is not defined. At least as an initial step it felt more reasonable to try to see how the controller responds to the operating conditions and then try to add constraints/ tune the parameters.

Case 4

Starting with the first case the simulation settings are:

- Target Engine Speed: 1500 rpm
- Torque request step: 0.045 (small step)
- Wastegate position: 0% (fully closed)

In Figure 5.12, the controller inputs are illustrated along with the torque request.

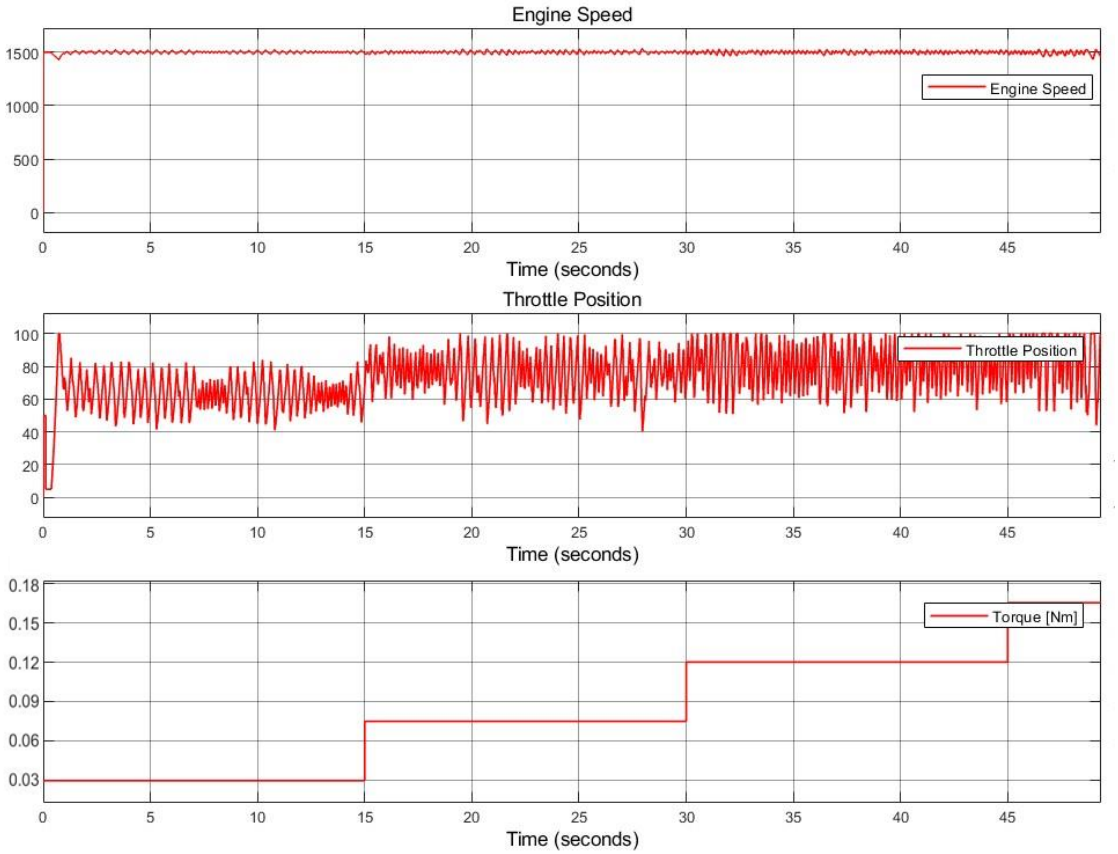

Figure 5.12 Controller Inputs- Case 4

Starting with the engine speed, it is obvious that while it stabilizes at 1500 rpm, it has small oscillations which are becoming more prominent as the torque request increases. This is also reflected by the throttle position where again it is noted that it fails to at least stabilize locally. While to torque request increases, the throttle position and engine speed fluctuate more rapidly.

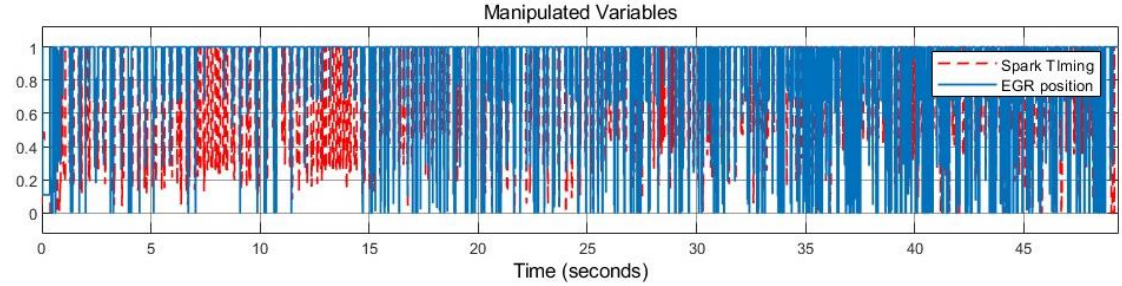

Figure 5.13 Manipulated Variables- Case 4

The profiles for the manipulated variables reflect that situation too. It is necessary for the EGR rate to be limited to avoid that type of behaviour. However, it doesn't make sense to limit the EGR rate at this point. It is more urgent to tune the PID controller first, so that it responds better to the torque request and then try to change the controller's parameters to achieve a better overall performance.

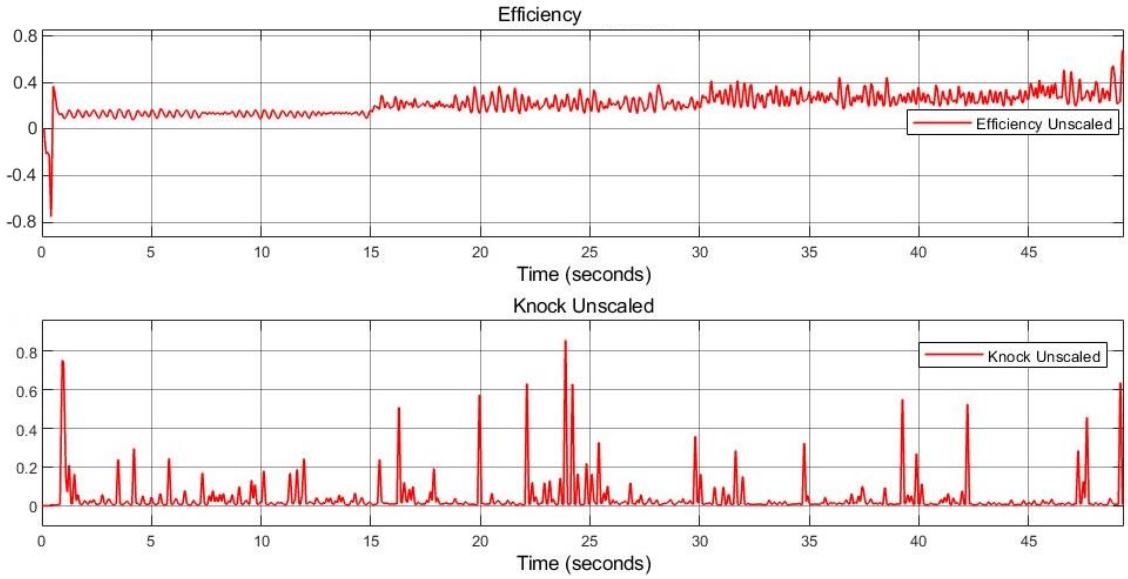

Figure 5.14 State variables- Case 4

Case 5

The next case to be investigated is identical to the one before, the only difference being the torque step:

- Target Engine Speed: 1500 rpm
- Torque request step: 0.15 (large step)
- Wastegate position: 0% (fully closed)

In Figure 5.15, the controller inputs are illustrated along with the torque request.

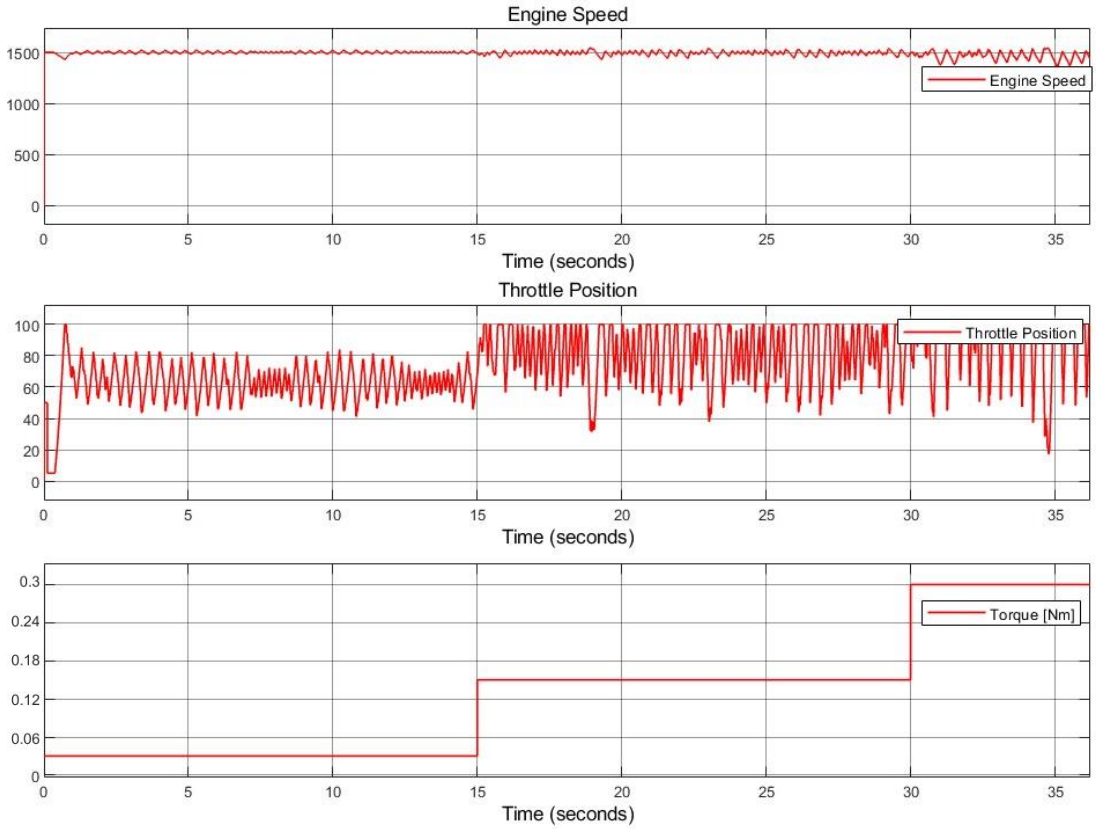

Figure 5.15 Controller Inputs- Case 5

The increase in torque request step has an obvious effect on the stability of the PID controller. While none of the other settings have been altered, the engine speed seems to oscillate more in comparison to case 4. The same goes for the throttle profile. Once again it varies rapidly and this overall behaviour is reflected also on the choice of manipulated variables, which fail to stabilize.

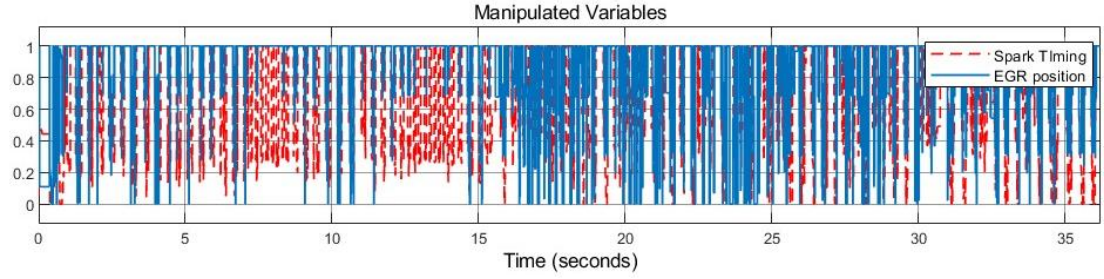

Figure 5.16 Manipulated Variables- Case 5

For both cases 4 and 5, the overall brake efficiency is rather low. In case 4, the knock indicator remains below the critical value of one the whole time, while for case 5, for the highest torque request the knock indicator is well above the critical value meaning that knock is occurring. It is quite obvious that the controller is not responding very well for both cases. The main reason for that being, that the controller was trained for more "steady state" throttle behaviour. Another reason, although relatively minor when compared to the previous one, is that for both cases the wastegate position is at 0%. So, for the scope of this thesis, it makes sense to assess how the wastegate position will affect the controller's response to the new setup that was introduced.

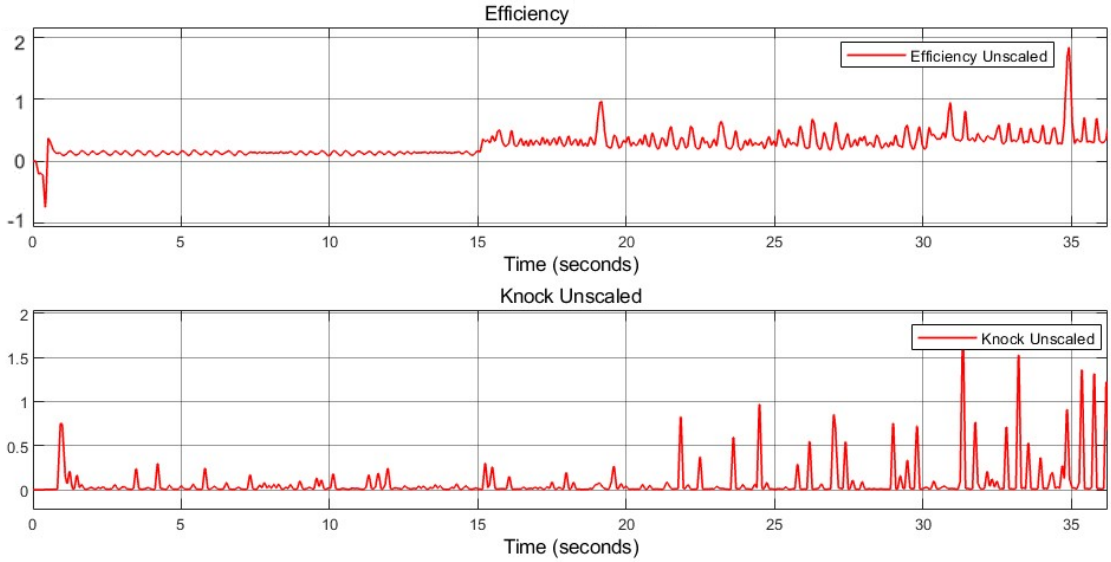

Figure 5.17 State variables- Case 5

Case 6

In this case, the low torque step has been selected and the wastegate is half open:

- Target Engine Speed: 1500 rpm
- Torque request step: 0.045 (small step)
- Wastegate position: 50% (half open)

In Figure 5.18,the controller inputs are illustrated along with the torque request. The effect of the wastegate is rather evident for the lowest torque request. With the assistance of the wastegate, the controller seems to be able to stabilize and keep the engine speed constant at 1500 rpm. The throttle also does not constantly oscillate, which is rather encouraging and underlines the complexity of the controller's behaviour.

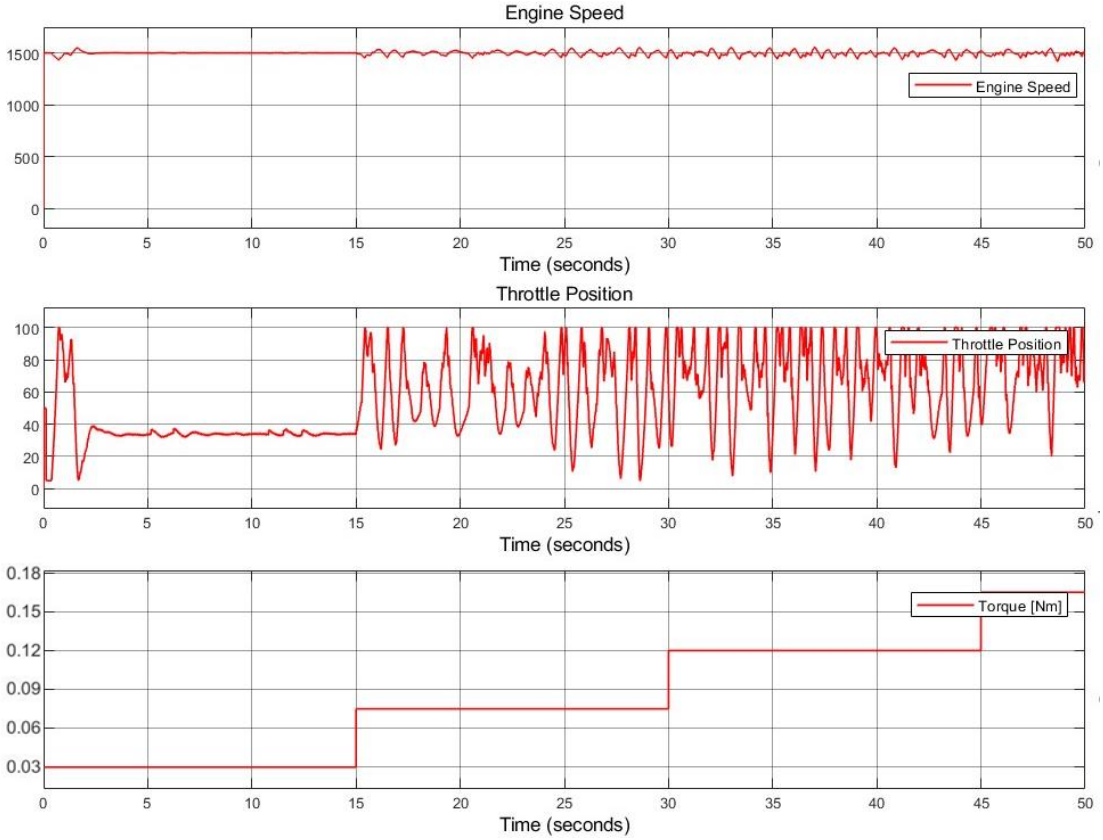

Figure 5.18 Controller Inputs- Case 6

Accordingly, in Figure 5.19, the manipulated variables exhibit the same trend. For the first 15 seconds, before the torque increases, both the EGR position and the spark timing are kept either relatively constant or oscillation is rather minimal compared to previous cases.

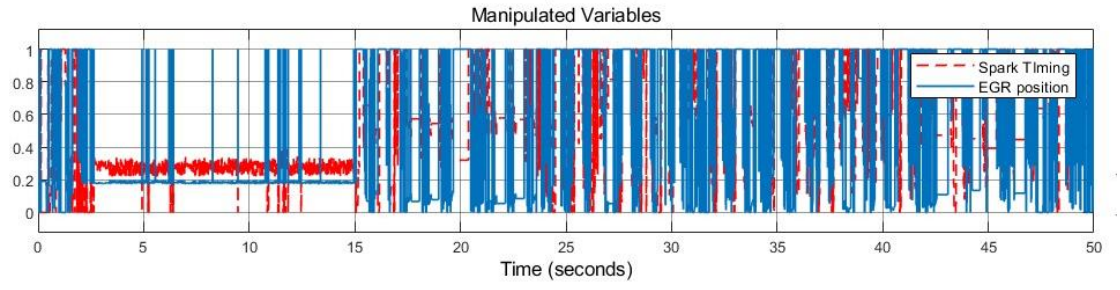

Figure 5.19 Manipulated Variables- Case 6

The same conclusions can also be made from the state variables. The brake efficiency and the knock indicator are rather stable for the lowest level of torque request. The brake efficiency is again rather low, indicating once more that the controller could benefit from some further tuning and the knock indicator for one specific time point (around 27s), is beyond one, indicating mild knock. Unfortunately, the effect of the wastegate "disappears" after the torque request is increased, so it makes sense to investigate whether this effect could be prolonged by further increasing the wastegate level.

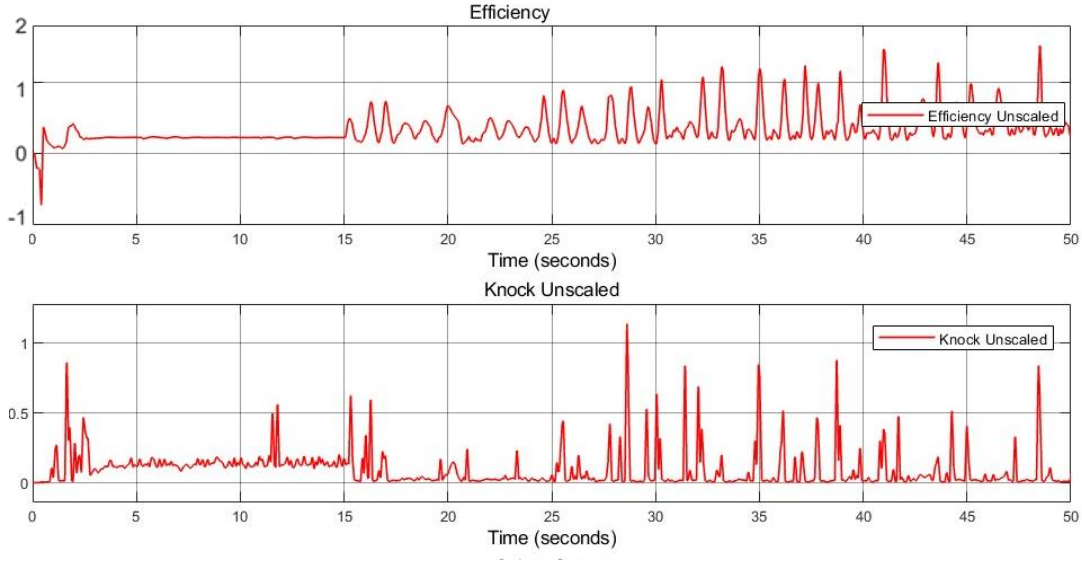

Figure 5.20 State variables- Case 6

Case 7

In this case the smaller torque step is selected, while the wastegate valve is fully open:

- Target Engine Speed: 1500 rpm
- Torque request step: 0.045 (small step)
- Wastegate position: 100% (fully open)

From Figure 5.21, the results are quite unexpected. While for the lowest torque request the controller fails to stabilize, for 0.074 of torque request it is rather stable. The same behaviour is located in the throttle trend. In other words, increasing the wastegate level, does not necessarily lead to increased stability/ performance of the controller.

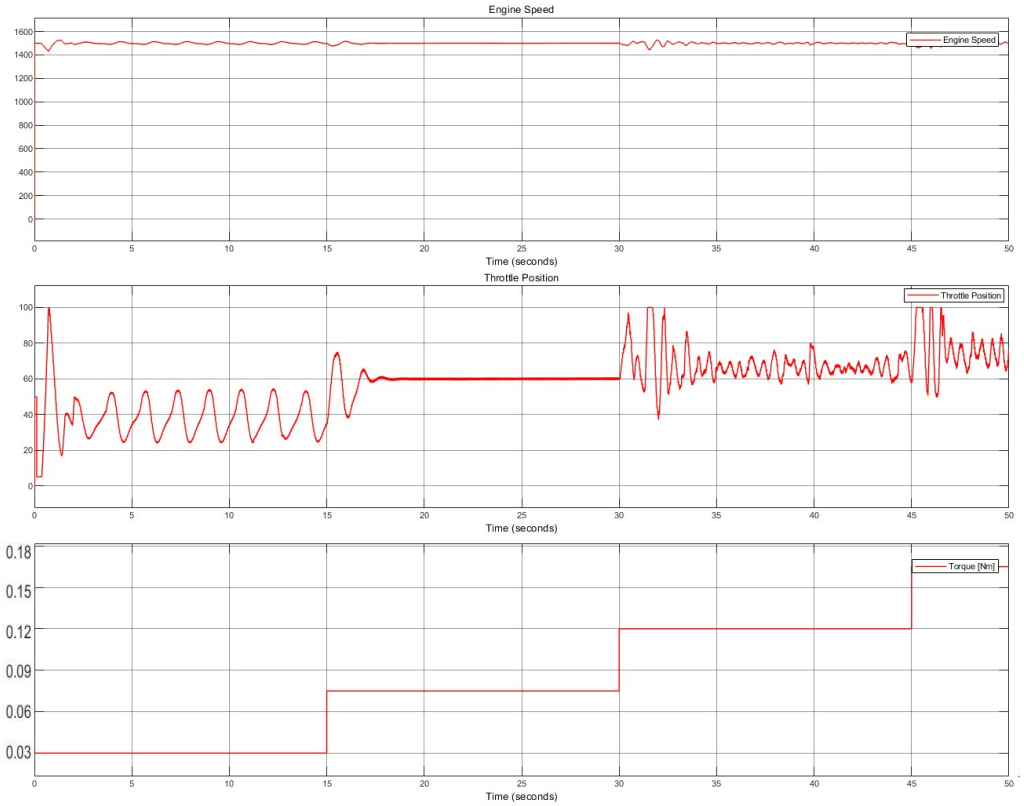

Figure 5.21 Controller Inputs- Case 7

When it comes to the manipulated variables, both of them are kept completely constant from 15 seconds until 30 seconds. In other words, they are kept constant for the second level of torque request. For the lowest level of torque request, they both fluctuate. However, the fluctuation is relatively milder when compared to after 30 seconds, where they both oscillate rapidly.

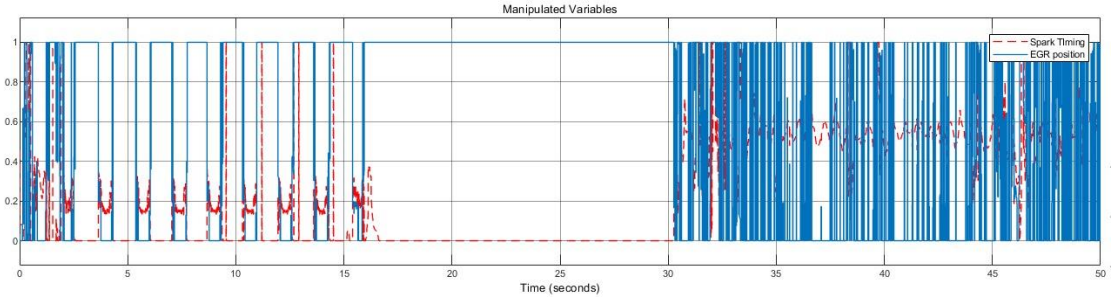

Figure 5.22 Manipulated Variables- Case 7

Again, the same conclusions are verified also by the state variables. They are both constant between 15 and 30 seconds. Throughout the whole experiment, the knock indicator is well below the critical value, which could be the effect of the fully opened wastegate when compared to the previous case. Again, the brake efficiency is relatively low, further verifying the need to alter the settings of the controller.

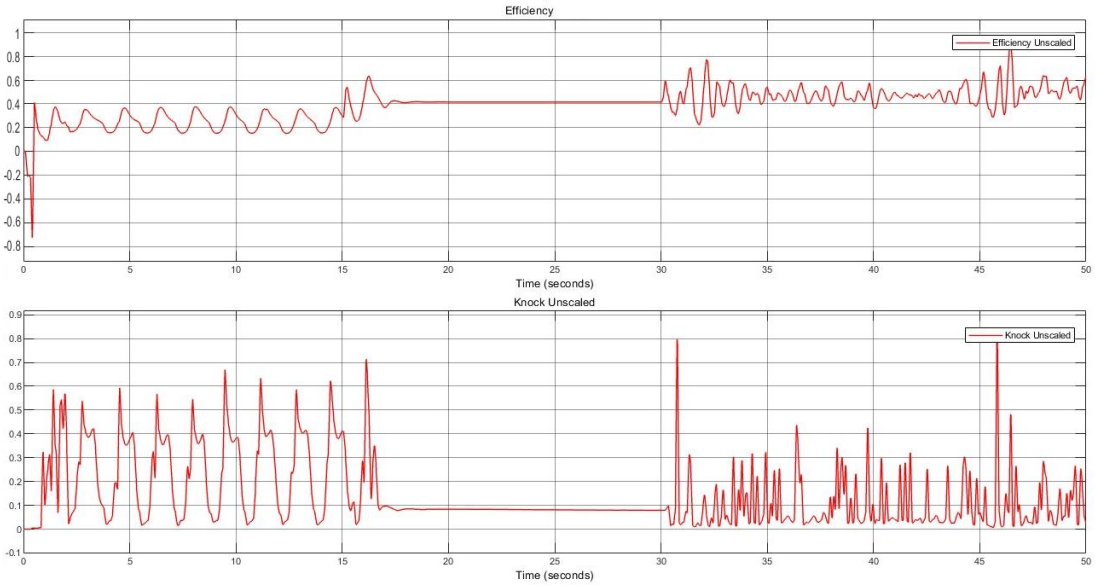

Figure 5.23 State variables- Case 7

Case 8

For the last case investigated, the higher torque step is selected while the wastegate is fully open:

- Target Engine Speed: 1500 rpm
- Torque request step: 0.15 (large step)
- Wastegate position: 100% (fully open)

Here the experiment length is slightly shorter than the previous cases, due to the increased computational load. It is quite evident that the wastegate is not able to prevent the controller from becoming unstable for higher torque requests. The same pattern is spotted also for the manipulated variables and the state variables. The manipulated variables in Figure 5.25, once again vary too quickly. This indicates unstable controller behaviour and obviously does not reflect real life operating conditions. When it comes to the state variables in Figure 5.26, once more the brake efficiency is rather low and the knock indicator for some points is above the critical value. Both of these facts, clearly illustrate that the controller is struggling to achieve its goal.

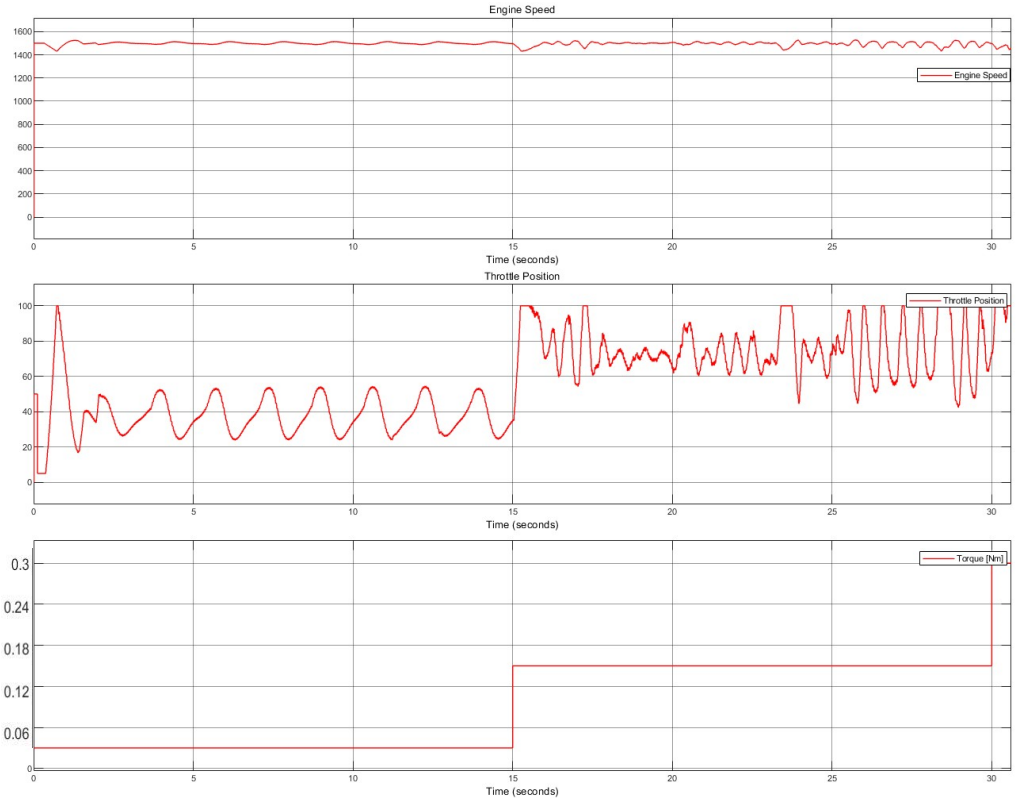

Figure 5.24 Controller Inputs- Case 8

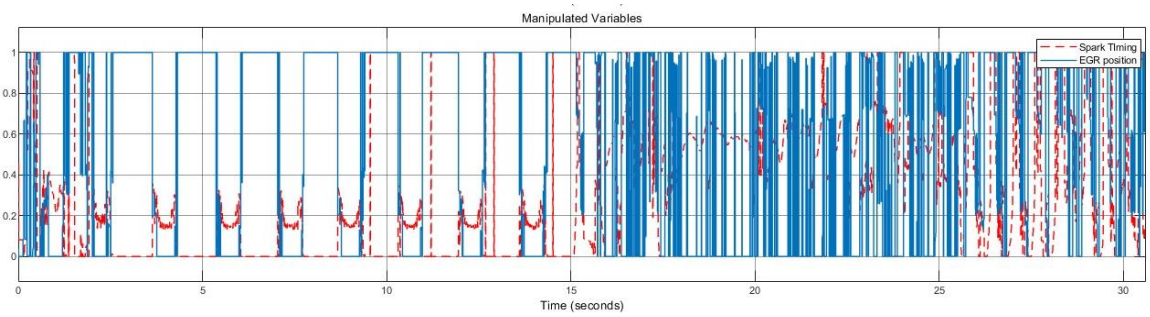

Figure 5.25 Manipulated Variables- Case 8

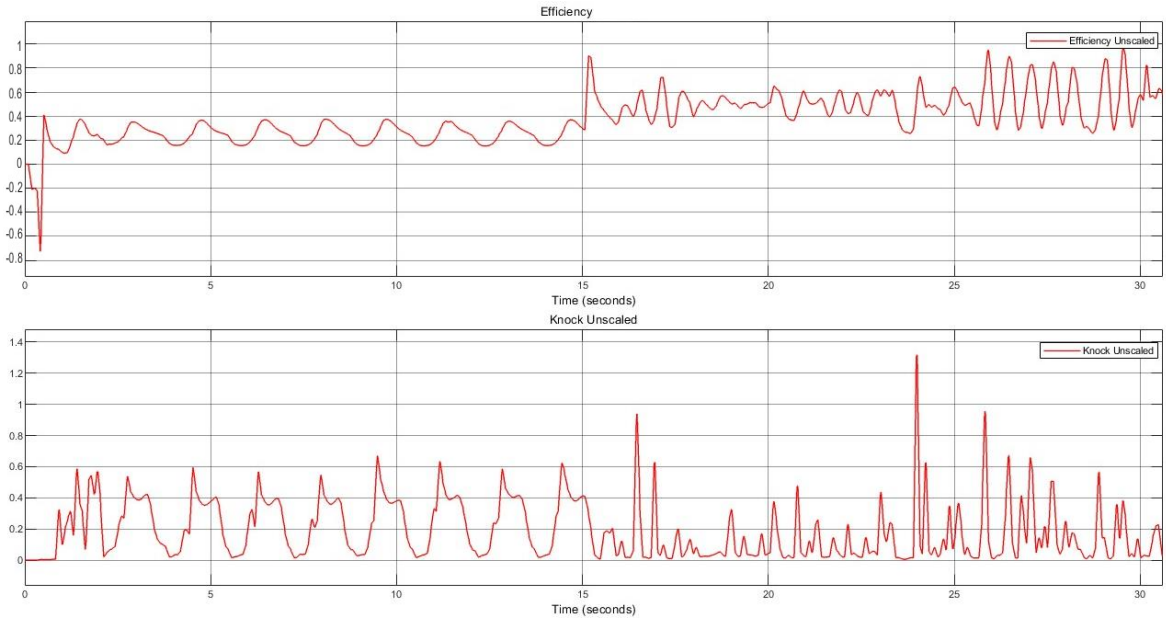

Figure 5.26 State variables- Case 8

6 Conclusions

The main goal of this thesis is to develop a control strategy that can be implemented to accurately predict knock occurrence in an IC engine and at the same time prevent it by altering its operating conditions. This purpose is fulfilled with respect to the objectives and limitations set in Chapter 1.

While the controller developed for the scope of this thesis has proven to be quite effective and has turned out to be a promising tool for future use, at the same time some issues and weaknesses have been located that need some further discussion. Some of those weaknesses are due to the limited time set for the completion of the thesis, while others are simply conclusions that have been made after the controller has been tested and assessed. Lastly, in this chapter some further improvements and future work are proposed.

6.1 Controller training & setup

It has already been mentioned in section 4.3.2, that the controller training process only included "steady state" profiles. By steady state it doesn't necessarily mean that the values of the input variables should be kept constant at all times. It does mean though that the controller will not be able to be as effective if the inputs are varied rapidly. This is further verified by section 5.2. However, effectively resolving this issue is slightly more complicated than just adding "transient" operation profiles in the training process.

As it has been made clear in the section 5.2, the throttle is varied by an added PID controller that alters it according to the requested engine speed. It has also been noticed that the aforementioned controller is struggling with maintaining the engine speed constant. So before moving on to any changes to the controller structure it is critical to assess how the controller responds when the PID controller for the throttle operates effectively. It is therefore important, to run some experiments again with the updated PID controller to establish a baseline for its behaviour and performance. After ensuring that the PID controller is effective, then the transient cases should be conducted again and see if any improvements have occurred. This process will contribute to locating where the controller fails to achieve its set goals.

The next step is to attempt to add transient profiles in the controller training process. This can be a lengthy and computationally heavy process, which is why it has been not included in the thesis scope. Apart from the time-consuming training process another aspect that needs to be considered is whether one controller can both accurately predict both steady state and transient behaviour or if two controllers will be necessary. One for steady state operation and one for transient operation. It has been established from section 5.1, that the controller is quite effective for steady state conditions, meaning that the overall strategy

and structure of the controller is achieving the desired goals. In other words, a base has been established and the next step is to further refine the controller.

The last stage of the "refinement" process is to vary the controller settings to be able to acquire the best result possible. This includes altering the cost function and the constraints. It has already been demonstrated in section 5.1, how different weights in the cost function and variations in the rate of change for the manipulated variables, affect the results of the controller. It is therefore necessary to balance those characteristics to achieve the desired outcome. However, once again this process can be very time consuming and will require a large amount of test runs, so to avoid unnecessary and costly runs, it is better to focus on that aspect once it has been established that all the previous steps have been completed and work as expected. Especially for the EGR position rate of change, it is important that it is limited. While spark timing can easily be altered from cycle to cycle, the same cannot be achieved for the EGR. In order for the effect of the EGR to be noticed in the combustion efficiency, it requires some time. So it makes more sense to not vary the EGR position abruptly.

6.2 Computational time

As stated in the previous section, the controller operation is rather slow. It is really important to reduce the computational time for two main reasons. The first reason being that it makes it easier to run more testing cases and establish a wider database of results. This will help to further assess the controller effectiveness and locate any further weaknesses in the setup. In section 5.2.2, it has been mentioned that some of the tested cases were especially time consuming and thus it was not possible, within the limitations of this thesis, to fully investigate the full requested torque range. At the same time, the training process of the controller state space model was especially time consuming. So, it is quite obvious that reducing the computational time either by altering the controller settings or possibly opting for more efficient hardware could benefit the controller performance.

6.3 Controller structure & Plant outputs

The presented controller is limited on only altering the spark timing and the EGR position to mitigate knock. While this is a reasonable choice, based on the literature review, there is a variety of different actuators that could be considered. Furthermore, the measured disturbances could also be increased. The more variables/ information added to the controller, the more accurate it is bound to be. However, it is really important to avoid making the controller excessively complex, to avoid increasing the computational cost. It is important when making any alterations, to consider which actuators are of high importance and which could be possibly omitted.

When it comes to the plant outputs, they do not really affect the computational load as much as changing the controller structure. With that being said, more information can be acquired by the simulation model to fully understand and

assess the effect of the controller on the engine operation. This selection can be personal to the user, depending on the area of interest.

6.4 Future Prospects

Concluding the assessment of the controller it can be stated that it fulfils most of the goals set on a satisfactory level. In other words, it constitutes a solid base for further work that exceeds the scope and limitations of this thesis. Some suggestions for both the controller and its implementation are the following:

- Further tuning of the controller to achieve optimal performance and shorten computational time. More details on that can be found in sections 6.1-6.3.
- The main goal behind the development of this controller was to shorten the engine mapping process and give a preliminary understanding of engine behaviour when it comes to knock. It is quite obvious from the results of the test runs that the controller is capable of fulfilling that goal. Furthermore, the controller can be a useful tool when it comes to software development, since at this point it's based on simulation results and not actual engine data.
- Adding to the previous point, by giving a preliminary understanding of engine behaviour, the controller can be a rather useful tool for engineers when comparing different engine hardware.
- Possibly the most important contribution of this thesis is the versatility of the developed strategy. While this controller has been developed specifically for a methane combustion engine, it can be applied on all different types of combustion engines. The main outline of the controller will remain unchanged. The only modifications required are the training of the state space model according to the individual engine data and possibly the tuning of the controller to fit the engine specifications.
- Lastly, while for the scope of the thesis, the controller has been trained and used exclusively with simulation data, in the future it can also be used with real life engine data. This is a step towards the final goal of this whole venture, which is to actually implement the controller on the engine control unit, as a means to avoid knock as an issue all together.

References

[1] Eurostat, "Greenhouse gas emissions by source sector, EU, 2021," Eurostat, 2021. [Online]. Available: https://ec.europa.eu/eurostat/statistics-explained/index.php?title=File:Greenhouse_gas_emissions_by_source_sector,_EU,_2021.png. [Accessed 12 June 2024].

[2] J. Inumaru, T. Hasegawa, H. Shirai, H. Nishida, N. Noda and S. Ohyama, "Fossil fuels combustion and environmental issues," *Thermal and Nuclear Power Generation,* vol. 2, pp. 1-56, 2021.

[3] C. Bae and J. Kim, "Alternative fuels for internal combustion engines," *Proceedings of the Combustion Institute,* vol. 36, no. 3, pp. 3389-3413, 2017.

[4] S. Verhelst, R. Sierens and S. Verstraeten, "A Critical Review of Experimental Research on Hydrogen Fueled SI Engines," *Journal of Engines,* vol. 115, pp. 264-274, 2006.

[5] X. Zhen, Y. Wang, S. Xu, Y. Zhu, C. Tao, T. Xu and M. Song, "The engine knock analysis – An overview," *Applied Energy,* vol. 92, pp. 628-636, 2012.

[6] L. Petrucci, F. Ricci, F. Mariani, V. Cruccolini and M. Violi, "Engine Knock Evaluation Using a Machine Learning Approach," *SAE Technical Paper,* 2020.

[7] U. Nations, "Sustainablle Development Goals," [Online]. Available: https://www.un.org/sustainabledevelopment/sustainable-development-goals/. [Accessed 20 June 2024].

[8] Z. Wang, H. Liu and R. D. Reitz, "Knocking combustion in spark-ignition engines," *Progress in Energy and Combustion Science,* vol. 61, pp. 78-112, 2017.

[9] M. Radwan, A. Helali, S. Elfeky and Y. Attai, "An Investigation on Knock and Pre-ignition with Tumble Induced Turbulence," *SAE Technical Papers,* 2007.

[10] T. W. Ryan, III, T. J. Callahan and S. R. King, "Engine Knock Rating of Natural Gases—Methane Number," *J. Eng. Gas Turbines Power,* vol. 4, no. 115, pp. 769-776, 1993.

[11] H. Terashima and M. Koshi, "Mechanisms of strong pressure wave generation in end-gas autoignition during knocking combustion," *Combustion and Flame,* vol. 162, no. 5, pp. 1944-1956, 2015.

[12] A. Egerton, "Engine Knock and Related Problems," *Nature,* vol. 122, pp. 20-26, 1928.

[13] D. J. Corrigan and S. Fontanesi, "Knock: A Century of Research," *SAE INTERNATIONAL JOURNAL OF ENGINES,* vol. 15, pp. 57-127, 2021.

[14] V. Mittal, "A Review of Recent Advancements in Knock Detection in Spark Ignition Engines," *Signals,* vol. 5, pp. 165-180, 2024.

[15] J. Wagner, J. Keane, R. Koseluk and W. Whitlock, "Engine Knock Detection: Products, Tools, and Emerging Research," *Journal of Engines,* vol. 107, pp. 636-646, 1998.

[16] W. Suijs, S. Broekaert, T. D. Cuyper and S. Verhelst, "The sensitivity of pressure-based knock threshold values to alternative fuels: A comparison of methanol vs. gasoline," *Fuel,* vol. 362, 2024.

[17] W. Suijs, R. D. Graeve and S. Verhelst, "An exploratory study of knock intensity in a large-bore heavy-duty methanol engine," *Energy Conversion and Management,*

vol. 302, 2024.

[18] G. Brecq, J. Bellettre and M. Tazerout, "A new indicator for knock detection in gas SI engines," *International Journal of Thermal Sciences,* vol. 42, no. 5, pp. 523-532, 2003.

[19] C. Tornatore, L. Marchitto and V. G., "Technologies for Knock Mitigation in SI Engines—A Review," *Engines and Fuels for Future Transport. Energy, Environment, and Sustainability.,* 2022.

[20] F. Millo and C. Ferraro, "Knock in S.I. Engines: A Comparison between Different Techniques for Detection and Control," *Journal of Fuel and Lubricants,* vol. 107, pp. 1091-1112, 1998.

[21] K. P. Schmillen and M. Rechs, "Different Methods of Knock Detection and Knock Control," *Journal of Engines,* vol. 100, pp. 1404-1415, 1991.

[22] EdrawMax, "What is a Control System Block Diagram," [Online]. Available: https://edrawmax.wondershare.com/electrical-engineering/control-system-block-diagram.html. [Accessed 20 June 2024].

[23] S. Zacher, Closed Loop Control and Management, Springer, 2022.

[24] MathWorks, "What is Model Predictive Control?," [Online]. Available: https://se.mathworks.com/help/mpc/gs/what-is-mpc.html. [Accessed 23 June 2024].

[25] R. Poley, in *Control Theory Fundamentals*, CreateSpace Independent Publishing Platform, 2014, p. 268.

[26] N. Armin, H. Heidarifar, M. Shahbakhti, C. R. Koch and H. Borhan, "Model Predictive Control of Internal Combustion Engines: A Review and Future Directions," *Energies,* vol. 14, no. 19, 2021.

[27] N. Khaled and B. Pattel, Practical Design and Application of Model Predictive Control: MPC for MATLAB and Simulink Users, Elsevier Inc. , 2018.

[28] E. Guresen and G. Kayakutlub, "Definition of artificial neural networks with comparison to other networks," *Procedia Computer Science,* vol. 3, pp. 426-433, 2011.

[29] A. Krenker, J. Bešter and A. Kos, "Introduction to the Artificial Neural Networks," in *Artificial Neural Networks - Methodological Advances and Biomedical Applications*, Rijeka, Croatia, InTech, 2011, pp. 3-18.

[30] MathWorks, "What Are Neural Networks Used For?," 2024 June 25. [Online]. Available: https://se.mathworks.com/discovery/neural-network.html.

[31] M. Simanjuntak, J. Simangunsong and H. Dai, "Epoch in a neural network for brain stroke," *International Journal of Basic and Applied Science,* vol. 11, no. 4, pp. 172-179, 2023.

[32] AnalytixLabs, "Activation Functions In Neural Networks — Its Components, Uses & Types," Medium, 2024 March 13. [Online]. Available: https://medium.com/@byanalytixlabs/activation-functions-in-neural-networks-its-components-uses-types-23cfc9a7a6d7. [Accessed 25 June 2024].

[33] R. Tan, C.-Y. Lin and M. Tomizuka, "Approximate nonlinear model predictive control of a gasoline engine with EGR," *2015 European Control Conference (ECC),* pp. 1414-1419, 2015.

[34] A. Norouzi, H. Heidarifar, M. Shahbakhti, C. R. Koch and H. Borhan, "Model Predictive Control of Internal Combustion Engines: A Review and Future

Directions," *Energies,* vol. 14, 2021.

DEPARTMENT OF MECHANICS AND
MARITIME SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY