MARMARA UNIVERSITY

FACULTY OF ENGINEERING

COMPARISON OF COMMERCIAL QD SOFTWARE AND PREPERATION OF A BASIC INTERFACE FOR A SIMPLE 1-D CODE

Arda Öğütcü, Ulaş Şeker

Müdek Final Project Commission

The essential features of an acceptable design Project

		Satisfied	Unsatisfied
1	Mechanical or Thermal Design		✔
2	Report writing technique	✔	
3	Development of student creativity		✔
4	Use of open-ended problems		✔
5	Formation of design	✔	
6	Problem statement and specification	✔	
7	Synthesis of alternative solutions		✔
8	Feasibility	✔	
9	Detailed system description	✔	
10	Consideration of constraints (e.g. economic, safety, reliability, etc.)		✔
11	Utilization of engineering and scientific principle	✔	

Decision	Signature
Accepted but not approved	13.07.2021
Prof. Dr. Bülent Ekici,	
Dr. Öğr. Üyesi Uğur Tümerdem	
Ar. Gör Serkan Öğüt	

GRADUATION PROJECT REPORT

Department of Mechanical Engineering

Supervisor

Prof. Dr. Mehmet Zafer Gül

ISTANBUL, 2021

MARMARA UNIVERSITY

FACULTY OF ENGINEERING

COMPARISON OF COMMERCIAL QD SOFTWARE AND PREPERATION OF A BASIC INTERFACE FOR A SIMPLE 1-D CODE

by

Arda Öğütcü, Ulaş Şeker

Feb 11, 2021, Istanbul

SUBMITTED TO THE DEPARTMENT OF MECHANICAL ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE

OF

BACHELOR OF SCIENCE

AT

MARMARA UNIVERSITY

Signature of Author(s) ....

Department of Mechanical Engineering

Prof. Dr. M. Zafer Gül

Certified By ....

Project Supervisor, Department of Mechanical Engineering

Accepted By.......Prof. Dr. Bülent EKİCİ...

Head of the Department of Mechanical Engineering

ACKNOWLEDGEMENT

First of all, we would like to thank our supervisor Prof. Dr. Mehmet Zafer Gül, for the valuable guidance and advice on preparing this thesis and giving us moral and material support.

January, 2021 Arda Öğütcü, Ulaş Şeker

CONTENTS

ACKNOWLEDGEMENT i

CONTENTS ii

ÖZET iv

ABSTRACT v

SYMBOLS vi

ABBREVIATIONS vii

LIST OF FIGURES viii

1 INTRODUCTION 1

2 LITERATURE SURVEY FOR 3-CYLINDER DIESEL ENGINES 2

2.1 General Information 2

2.1.1 Objective 2

2.1.2 Brief History of Diesel Engines 2

2.1.3 Brief History of Straight Three Diesel Engines 3

2.2 Theory 3

2.2.1 Working Principle of the Diesel Engines 3

2.2.2 Working Principle of the Straight Three Diesel Engines 4

2.3 Problems of the 3-Cylinder Diesel Engines 4

2.4 Technologies Used in 3-Cylinder Diesel Engines 5

2.4.1 Balance Shaft 5

2.4.2 Valvetronic Technology 6

2.4.3 Turbocharger Technology 7

2.4.4 Cylinder Deactivation 10

3 SOFTWARE RESEARCH AND COMPARISON 11

3.1 General Information 11

3.1.1 Objective 11

3.1.2 Softwares used in Market.....11
3.1.3 Software Selection.....11
3.2 Explanation of Softwares.....11
3.2.1 Diesel-RK.....11
3.2.2 Lotus Engine Simulation.....12
3.2.3 Ricardo WAVE.....12
3.3 Comparison of Softwares.....12
3.4 Results.....17
3.4.1 Diesel-RK.....17
3.4.2 Lotus Engine Simulation.....19
3.4.3 Ricardo WAVE.....21
4 ENGINE SELECTION.....23
4.1 General Information.....23
4.2 Engine Specifications.....24
4.3 DPF Problems.....24
4.4 Results.....25
5 GOVERNING EQUATIONS.....28
5.1 Single-Zone Model.....28
5.2 Annand’s Heat Transfer Model.....30
5.3 Engine Friction Model.....32
5.4 Residual Gas Fraction Model.....32
5.5 Burned and Unburned Gas Temperature Calculations.....32
6 MATLAB APPDESIGNER ENGINE PERFORMANCE SIMULATION MODEL 34
6.1 Components of the App.....34
6.2 Program Inputs.....34
7 CONCLUSION.....49

8 REFERENCES 51

ÖZET

Bu projenin amacı, 1-D motor simülasyon yazılımı kullanarak turboşarjlı 3 silindirli bir dizel motor tasarlamaktır. Bunu gerçekleştirmek için 3 silindirli dizel motorlar için bir literatür araştırması yapıldı. Sırasıyla tarihçesi, çalışma prensipleri, sorunları ve bu sorunları önlemeye yönelik yeni teknolojiler araştırıldı. Pazardaki 3 silindirli dizel motorların popülaritesi görüldü.

Piyasadaki 1-D motor performans yazılımlarının karşılaştırması, farklı yazılımların farklı sonuçlarını simüle etmek ve karşılaştırmak için basit bir 3 silindirli dizel motor seçilerek yapıldı. Bu karşılaştırma, kullanıcının sonuçları ne kadar kolay alabildiğine ve yazılımların kullanıcı dostu olup olmadığına göre yapıldı.

Önceki adımlar tamamlandıktan sonra örnek bir motor seçildi. Önceki adımdan seçilen yazılım kullanılarak örnek motor simüle edildi. Simülasyon tamamlandıktan sonra, sonuçlar örnek motor özellikleriyle karşılaştırıldı. Karşılaştırma, simülasyon sonuçlarının ve spesifikasyonların tutarlı olduğunu ortaya çıkardı. Bundan sonra, turboşarj tasarımdan çıkarıldı ve örnek motorla karşılaştırıldı. Bu karşılaştırma, motor performansı ve verimliliğindeki artışı gösterdi.

ABSTRACT

The purpose for this project is to design a 3-cylinder diesel engine with turbocharger with using a 1-D engine simulation software. To achieve this, a literature survey research for 3-cylinder diesel engines was conducted. The history, working principles, problems and the new technologies to prevent these problems were researched, respectively. The popularity of the 3-cylinder diesel engines in the market was seen.

A comparison of the 1-D engine performance softwares in the market were made by selecting a simple 3-cylinder diesel engine to simulate and comparing the different results of the different softwares. This comparison was made based on how easily the user can get the results and whether the softwares are user-friendly or not.

After the previous steps are done, a sample engine was selected. Using the chosen software from the previous step, the sample engine was simulated. After the simulation was concluded, the results were compared to the sample engine specifications. Comparison revealed that the simulation results and the specifications were consistent. After this, the turbocharger was removed from the design and compared against the sample engine. This comparison shows the increase in the engine performance and efficiencies.

SYMBOLS

° : Degree, a unit of measurement of angles

L : Liter, a metric unit of capacity

°C **:** Centigrade Degree**,** unit of temperature on the Celsius scale

ABBREVIATIONS

1-D : One Dimension

CI : compression ignition

cc : Cubic Centimeter

TDC : Top Dead Centre

BDC : Bottom Dead Centre

V6 : Six Cylinder Engine

V8 : Eight Cylinder Engine

RPM : Revolutions Per Minute

CFD **:** Computational fluid dynamics

mm : millimeters

TDI : Turbocharge Direct Injection

CR : Common Rail

DOHC : Dual Over Head Camshaft

EGR : Exhaust Gas Recirculation

HP : Horse Power

lb.ft : Pound-feet

kW : kilowatt

N.m : Newton-meter

DPF : Diesel Particulate Filter

LIST OF FIGURES

Figure 2-1: Simple sketch for 3-cylinder engine 5
Figure 2-2: Simple sketch for 3-cyliner engine with balance shaft 6
Figure 2-3: 3-cyliner engine with balance shaft 6
Figure 2-4: Simple sketch for valvetronic technology 7
Figure 2-5: Diagram for turbocharger 7
Figure 2-6: Exhaust gas pulses for a 2-cylinder engine 8
Figure 2-7: Exhaust gas pulses for a 4-cylinder engine 8
Figure 2-8: Exhaust gas pulses for a 3-cylinder engine 9
Figure 2-9: Twin scroll turbocharger sketch 9
Figure 2-10: 8-cylinder engine with cylinder deactivation 10
Figure 3-1: Lotus Engine Simulation Interface Panel 13
Figure 3-2: Model of a 3-Cylinder Diesel Engine in Lotus Engine Simulation 13
Figure 3-3: Lotus Engine Simulation Parameter Panel 14
Figure 3-4: Diesel-RK Parameter Panel 14
Figure 3-5: Diesel-RK Parameter Panel 15
Figure 3-6: Model of a 3-Cylinder Diesel Engine in Ricardo WAVE 15
Figure 3-7: Help Section of Ricardo WAVE 16
Figure 3-8: Library and Session Trees of Ricardo WAVE 16
Figure 3-9: Diesel-RK Result Table 18
Figure 3-10: Diesel-RK Result Graph Section 18
Figure 3-11: Diesel-RK Result Graph Section 19
Figure 3-12: Lotus Engine Simulation Result Table 19
Figure 3-13: Inlet Valve Result Graphs 20
Figure 3-14: Exhaust Valve Result Graphs 20
Figure 3-15: Engine Performance Summary in Lotus Engine Simulation 21
Figure 3-16: Ricardo WAVE Result Table of the Engine Block 21
Figure 3-17: Ricardo WAVE Presentation of the Model in Solver Section 22
Figure 3-18: Ricardo WAVE Plot Section 22
Figure 3-19: Engine Performance Summary in Ricardo WAVE 23
Figure 4-1: 3-cylinder turbocharged diesel engine modelled in WAVE 25
Figure 4-2: Ricardo WAVE Presentation of the Model in Solver Section 26
Figure 4-3: Data for engine 26

Figure 4-4: Plot of break power-engine speed comparison between 3-cylinder turbocharged diesel engine vs. 3-cylinder diesel engine 27
Figure 4-5: Plot of total volumetric efficiency-engine speed comparison between 3-cylinder turbocharged diesel engine vs. 3-cylinder diesel engine 27
Figure 6-1 asdasda 35

1 INTRODUCTION

1.1 Thesis Objective

The purpose of this thesis is to design an engine performance program and to write a basic 1-D code. To achieve this, a general research will be conducted to find a suitable engine and a suitable engine simulation program for a base model.

After the engine and the engine simulation program is determined, app designing process begins. Some parts of the calculations part of the code was taken from “A User-Friendly, Two-Zone Heat Release Model for Predicting Spark-Ignition Engine Performance and Emissions”.[1] A simple app will also be created to take user inputs and use them in calculations.

1.2 Thesis Research

First three chapters of this report excluding the introduction part includes:

First chapter aims to understand the shift in the automotive industry towards 3-cylinder diesel engines and get a better look at this new trend. In order to achieve this, a 3-cylinder diesel engine will be designed in 1-D code and simulated based on an existing 3-cylinder diesel engine. To achieve our goal, a literature survey about diesel and 3-cylinder engines will be made, respectively. The new technologies and techniques that are being used in automotive industry for 3-cylinder diesel engines will also be mentioned in this report to understand the popularity of the new generation 3-cylinder diesel engines as well.

In second chapter, engine performance softwares used in the market will be researched and compared. A simple 3-cylinder diesel engine will be selected and tested in all engine simulation softwares that are available. After the tests are done, the comparison between softwares will be made based on availability of research materials and documents, user-friendly interface, straightforwardness of result data and plots.

After the comparison and selection of the software, the main 3-cylinder diesel engine that was picked will be tested. The purpose of engine selection was to see the difference between turbocharged non-turbocharged engine and the effects of this technology on new 3-cylinder diesel engines. To do that, a research for the engine specifications will be made. The test will be executed based on the data from the engine specifications research. After completing the test, the comparison for brake power will be made to see the differences between the real data that is given by the company and the data from the test.

1.3 Design and Coding of the App of Thesis

After the initial research phase, design and coding phase will be started. Using the chosen engine simulation program that is in the market as a basis, a new simplified version of it will be designed. The designed app will be capable of calculating single zone model calculations for diesel engines and presenting the results as plots and a summary table.

2 LITERATURE SURVEY FOR 3-CYLINDER DIESEL ENGINES

2.1 General Information

2.1.1 Objective

The purpose to write this report is to get a better understanding of the 4-stroke 3-cylinder diesel engines, learn how they behave and operate, and to understand the popularity of the 3-cylinder designs nowadays over 4-cylinder classic diesel engine design. To achieve it, firstly a research about the history of the 3-cylinder diesel engines will be made. Then, the advantages and disadvantages of the 3-cylinder diesel engine designs will be identified and compared them against 4-cylinder diesel engine designs.

Besides these topics, an explanation about the reason why the 3-cylinder engines are preferred over 4-cylinder engines and the technologies that are currently being used in 3-cylinder diesel engines to increase efficiency and power output to nearly the same level of 4-cylinder diesel engines.

2.1.2 Brief History of Diesel Engines

By 1892, Rudolf Diesel (1858-1913) had perfected his compression ignition engine into basically the same diesel engine known today. This was after years of development work which included the use of solid fuel in his early experimental engines. Early compression ignition engines were noisy, large, slow, single-cylinder engines. They were, however, generally more efficient than spark ignition engines. It wasn't until the 1920s that multicylinder compression ignition engines were made small enough to be used with automobiles and trucks.[2]

- Modern Diesel Engines

Carbon soot particulate generation has been greatly reduced in modern CI engines by advanced design technology in fuel injectors and combustion chamber geometry. With greatly increased mixing efficiency and speeds, large regions of fuel-rich mixtures can be avoided when combustion starts. These are the regions where carbon soot is generated, and by reducing their volume, far less soot is generated. Increased mixing speeds are obtained by a combination of indirect injection, better combustion chamber geometry, better injector design and higher pressures, heated spray targets, and air-assisted injectors. Indirect injection into a secondary chamber that promotes high turbulence and swirl greatly speeds the air-fuel mixing process. Better nozzle design and higher injection pressures create finer fuel droplets which evaporate and mix quicker. Injection against a hot surface speeds evaporation, as do air-assisted injectors. Some modern, top-of-the-line CI automobile engines (e.g., Mercedes) have reduced particulate generation enough that they meet stringent standards without the need for particulate traps.[3]

2.1.3 Brief History of Straight Three Diesel Engines

The very first commercially available straight three engine amongst the cars was, powered by a 900 cc two-stroke engine, the 1953-1955 DKW F91. Then, at 1956, Saab's 750 cc two-stroke engine was introduced, which was used in Saab 93. The same engine was also used in Saab 95 and Saab 96 until 1980.

After that, Subaru was introduced a 543 cc, 4-stroke 3-cylinder engine in 1980 which was used in Alto and Fronte models. The straight-three versions of the Ford EcoBoost engine - a turbocharged 1.0-litre petrol engine - was introduced in the 2012 Ford Focus. It uses of an unbalanced flywheel to shift the inherent three-cylinder imbalance to the horizontal plane where it is more easily managed by engine mounts, and so remove the need to use balancer shafts. In 2016, cylinder deactivation was added, claimed to be a world first for three-cylinder engines.[4]

2.2 Theory

2.2.1 Working Principle of the Diesel Engines

An engine in which the combustion process starts when the air-fuel mixture self-ignites due to high temperature in the combustion chamber caused by high compression. CI engines are often called Diesel engines, especially in the non-technical community.[5]

At first, while the pistons are near to TDC (Top Dead Centre), air is begin to enter the cylinder and fill the volume and causes to push the piston down to BDC (Bottom Dead Centre). Then, the inlet valve closes and piston is begin to move to TDC. That causes the air to compress and increase its temperature. When the piston reaches near to TDC, the fuel injects directly to the combustion chamber which contains high temperature air. Fuel injector ensures that the fuel is distributed evenly. The fuel is ignited by the compressed air with a high temperature and that causes the piston to move downwards to BDC. That whole process is called combustion. When the piston reaches near to BDC, exhaust valve opens and that causes the exhaust air to move out from the combustion chamber while piston moves upward to TDC. Lastly, while the piston is near to TDC, exhaust valve closes and inlet valve opens. With that, system returns its original position and the next cycle starts.

2.2.2 Working Principle of the Straight Three Diesel Engines

Straight three engine, also known as inline three cylinder engine, has a similar design as straight four engine but has one less cylinder. Working principle of the straight three engines can be categorized by looking at their engine cycles. Similar to straight four engine, these engines also have three different main engine cycles which are otto, diesel, and dual cycles.

The whole process of diesel cycle is explained briefly above. However, there are some differences between 3-cylinder and 4-cylinder engines. Firstly, since 3-cylinder engines has one less cylinder than 4-cylinder engines, their firing interval is different. While 4-cylinder engines has a 180˚ firing interval, it is 240˚ for 3-cylinder engines. That shows that there will be some moment that no power stroke occurs in the cycle which will be explained later.

2.3 Problems of the 3-Cylinder Diesel Engines

3-cylinder engines are popular nowadays but it was not the same at the past. Companies tried different 3-cylinder engine designs to get a better seat in the competition between companies in engine and automotive industry. In 20th century, many companies tried to product and sell their cars which contained a 3-cylinder engine and some of them succeed such as Geo Metro which manufactured by General Motors and Suzuki in 1989. However, it can only put out fifty five horsepower with his 1.0L 3-cylinder engine. Therefore, it could not become more popular during the years.

The first generation 3-cylinder engines had more serious problems than the lacking in power. Firstly, they were not balanced very well because of the design. As its explained above, three cylinder engine has a 240˚ firing interval, different than 4-cylinder engines which has a 180˚ firing interval. One cylinder of a four stroke engines need a total of 720˚ to complete one combustion cycle. However, only 180˚ of that 720˚ is the power stroke. When looking at the 4-cylinder engines, it can be seen that they have a 180˚ firing interval. When the engine starts with a power cycle in a 4-cylinder engine, there will be no gap between the power strokes since it has four cylinders total. So every 180˚, there will be another power stroke on the engine and that will provide a smooth balance between the cylinders and the power strokes.

However, there is a different situation in 3-cylinder engines and that causes one of the most common problems of the straight three engines. The 3-cylinder engines have a 240˚ firing interval that can be calculated with a very simple calculation which is 720˚/3 cylinders=240˚ per cylinder. However, like mentioned earlier, a power stroke of one combustion cycle only takes 180˚. Therefore, everytime a power stroke occurs, there will be a 60˚ of no power stroke which can be called as silence period on the engine since one firing interval takes 240˚ for one cylinder in 3-cylinder engines. What that means is there will be times when there is no power stroke occuring, so the power delivery of a straight three engine is not going to be as smooth as engine with more cylinders in it.

The other main problem of the 3-cylinder engine is the power balance of the engine with respect to the centerline. While looking at the straight three engines when they are operating, the unbalanced piston positions can be seen.

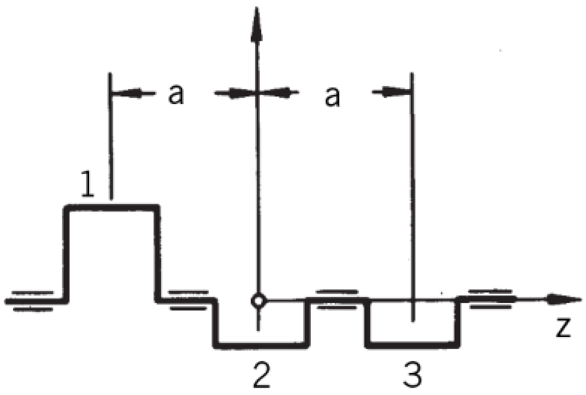


Figure 2-1: Simple sketch for 3-cylinder engine

Writing a moment of inertia equation with respect to centerline, it can be seen that the moment is not balanced. The unbalanced forces will go back and forth between the first cylinder and the last cylinder and as a result of that, the whole engine will begin to vibrate which is not desirable. This was the biggest problem in the past because when the engine vibrates, it shakes the whole body of the car so much even the passengers and the driver can feel it. That is why the 3-cylinder engines were not popular in the past, but they become more populer with the new technologies than ever before.

2.4 Technologies Used in 3-Cylinder Diesel Engines

2.4.1 Balance Shaft

When a straight three engine operates, the unbalanced forces caused by the design generates vibration and it rotates the engine end to end. So, to eliminate this problem, engineers started to use a balance shaft. The balance shaft is an extra engine part to reduce the vibration by cancelling out the unbalanced dynamic forces. It consists of two counter-weights opposite of each other. These counter-weights have a 180˚ phase difference. By phasing the balance shaft correctly, the out of phase up/down motion balances the rocking couple of the pistons.

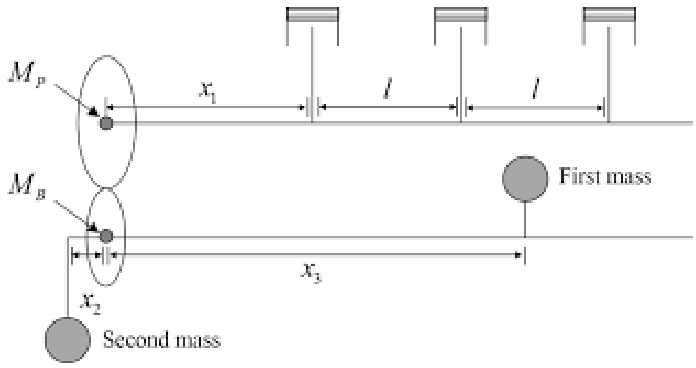


Figure 2-2: Simple sketch for 3-cyliner engine with balance shaft

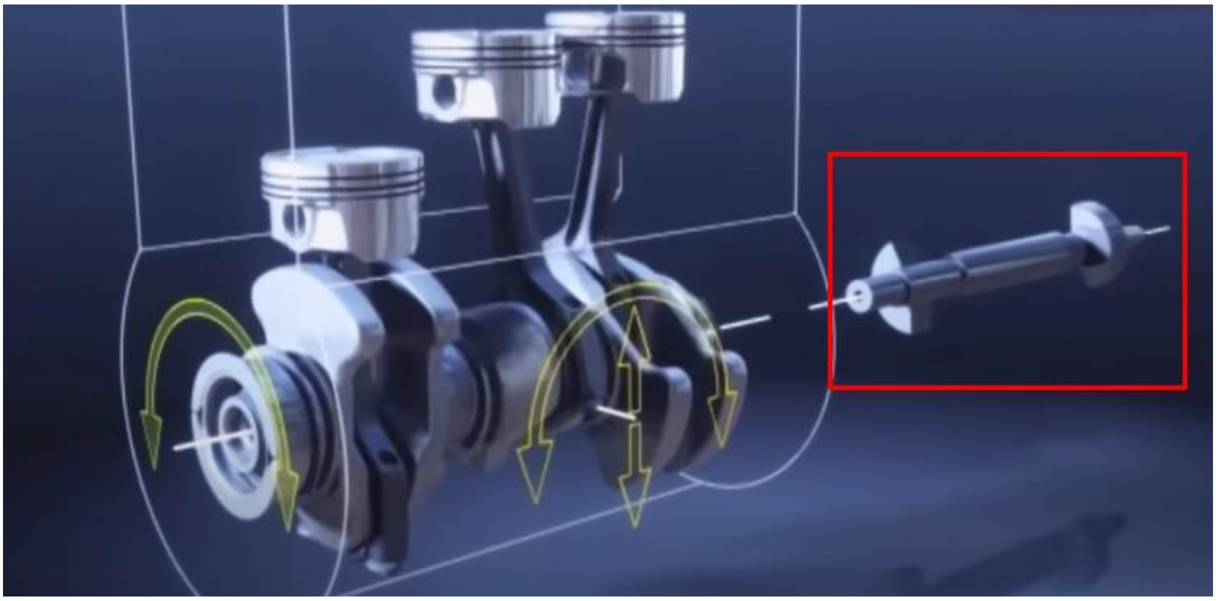


Figure 2-3: 3-cyliner engine with balance shaft

2.4.2 Valvetronic Technology

Valvetronic allows the engine to run without a throttle butterfly, the cylinder charge being determined under part load as a function of the valve-opening period.[6] An extra set of rocker arms, which called intermediate arms, are added to cylinder heads between the valve stem and the crankshaft. These intermediate arms are able to pivot on a central point, by means of an extra, electronically actuated camshaft. This movement alone, without any movement of the intake camshaft, can vary the intake valves' lift from fully open, or maximum power, to almost closed, or idle.

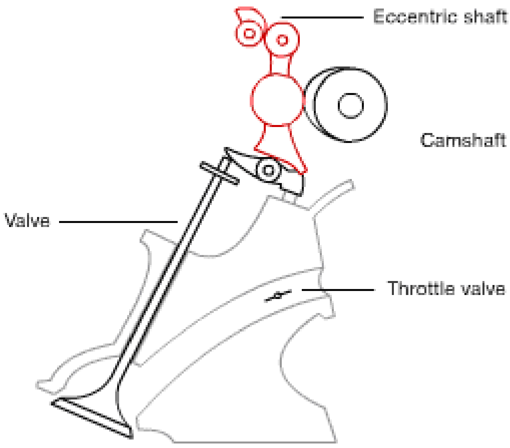


Figure 2-4: Simple sketch for valvetronic technology

2.4.3 Turbocharger Technology

On turbocharged engines exhaust gases leaving the exhaust manifold enter the turbine of the turbocharger, which drives the compressor that compresses the incoming air. Pressure of the exhaust gas entering the turbine is only slightly higher than atmospheric, and only a very small pressure drop is possible through the turbine.[7]

Turbochargers should be mounted as close as possible to the cylinder exhaust ports so that turbine inlet pressure, temperature, and kinetic energy can be as high as possible.[8]

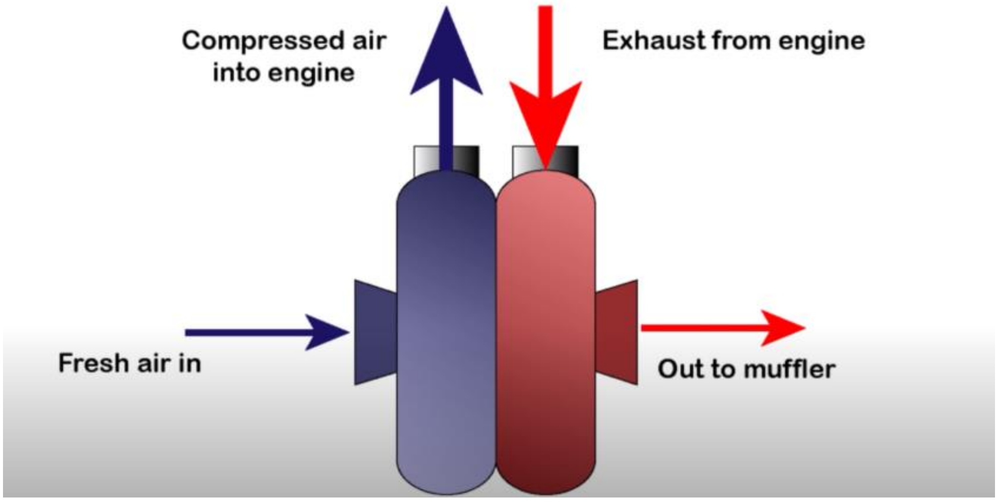


Figure 2-5: Diagram for turbocharger

- Turbochargers in 2-3-4 Cylinder Engines

The efficiency of the turbocharger is connected to the exhaust gas pulses occuring at the end of the combustion cycles. Therefore, engines with turbocharger technology give different results based on their cylinder numbers.

In 2-cylinder engines, the exhaust events are a little bit too far apart from each other and the there are big gaps between the exhaust gas pulses that can be seen in Fig. . That causes the turbocharger to lose energy and slow down while waiting for the next exhaust gas pulse to happens. Therefore, using a turbocharger in a 2-cylinder engine is not efficient.

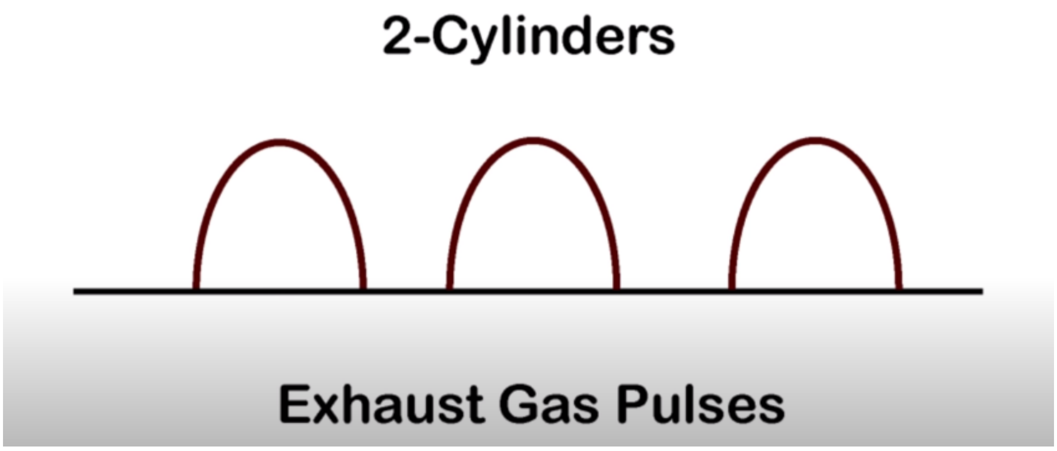


Figure 2-6: Exhaust gas pulses for a 2-cylinder engine

In 4-cylinder engines, exhaust gas pulses intersect each other and that can cause a turbulance in exhaust manifold and decrease the scavanging efficiency. This causes a decrease in overall efficiency.

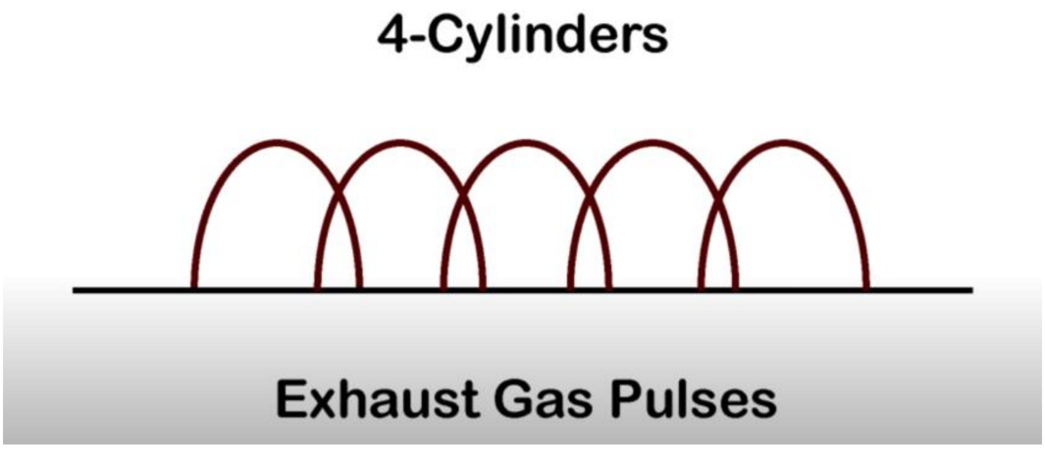


Figure 2-7: Exhaust gas pulses for a 4-cylinder engine

In 3-cylinder engines, exhaust gas pulses are successive with one another which in a way that the interferance is minimal while still having enough flow that the turbocharger does not slow down between pulses. This reason makes the 3-cylinder turbocharger engine designs optimal and explains the popularity of this design in the market.

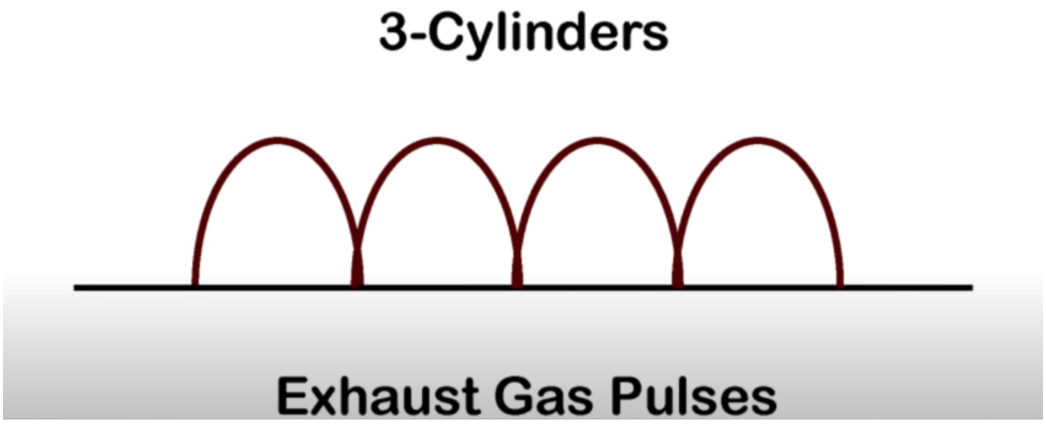


Figure 2-8: Exhaust gas pulses for a 3-cylinder engine

- Twin Scroll Turbocharger Technology

This technology is used by BMW and Volvo. It has a similar design to the normal turbocharger except two inlets for exhaust gasses which have turbines in them. These turbines are driven a large compressor which maximizes the overall efficiency.

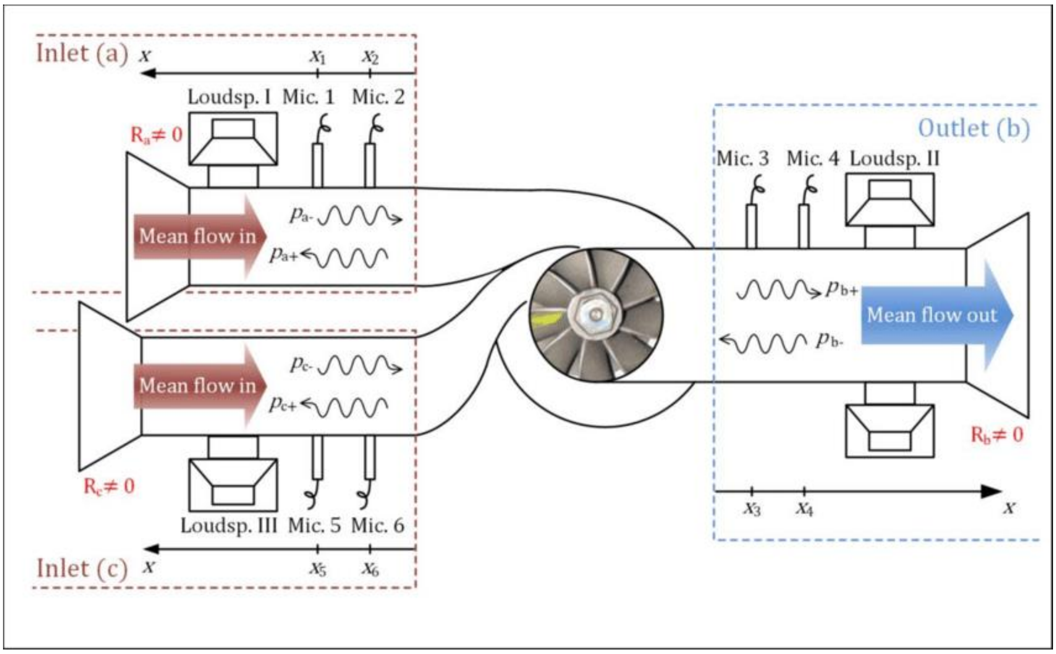


Figure 2-9: Twin scroll turbocharger sketch

2.4.4 Cylinder Deactivation

Cylinder deactivation, as known as variable displacement, is an engine technology which changes the engine displacement by simply deactivating cylinders to improve fuel efficiency. When a cylinder deactivated, there is no fuel and air entry to cylinder. This does not affect the movement of the deactivated piston, because it is attached to the crankshaft. Cylinder deactivation is usually being used in V6 and V8 engines such as V8 DaimlerChrysler Hemi engine of Mercedes-Benz.

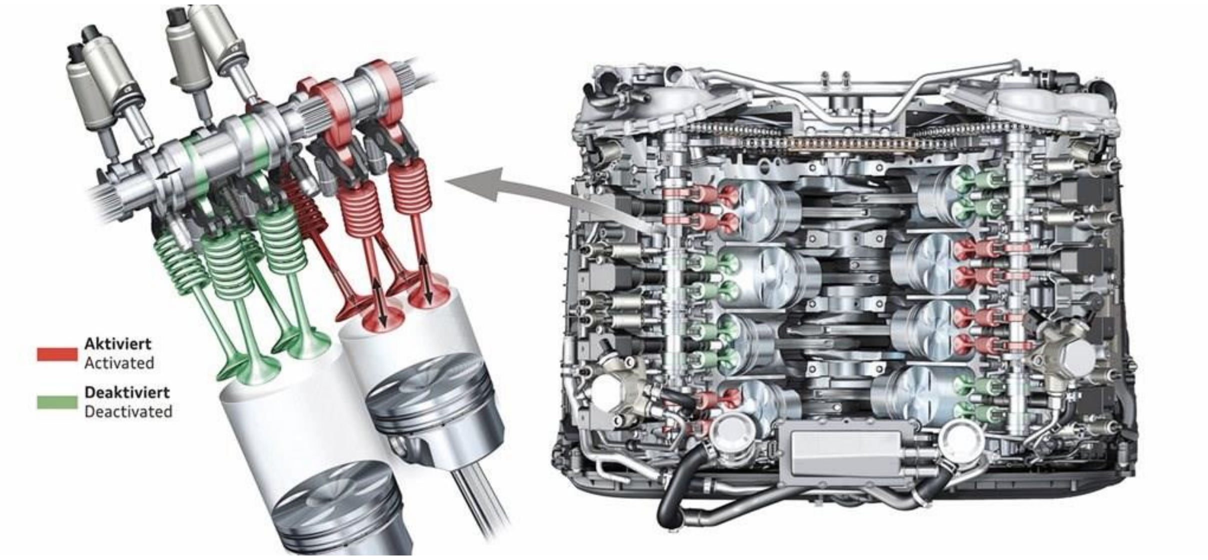


Figure 2-10: 8-cylinder engine with cylinder deactivation

However, this technology can be only used in specific operating conditions:

- It can only be used when the full power is not needed.
- It can not be used in lower or higher RPMs.
- Engine has to run at the same operational point to be able to gain the benefits of fuel efficiency.
- It adds an additional vibration to the engine which complicates the calculations of overall vibrations.

For the reasons above, companies moved away from 6-cylinder engines with cylinder deactivation technology to 3-cylinder engine designs. Moreover, in 2012, the 3-cylinder engine named Ecoboost 1.0L, introduced by Ford, is redesigned to a 3-cylinder engine with cylinder deactivation technology in 2018. This was the first time that the cylinder deactivation was used in a 3-cylinder diesel engine.

3 SOFTWARE RESEARCH AND COMPARISON

3.1 General Information

3.1.1 Objective

The main objective of the report is to researching the engine performance simulation softwares that are currently used in the market by the automotive industry and comparing these softwares against each other to detect which one is the best to use and write in our thesis. To achieve this, a simple three cylinder engine as the main design will be used for comparison.

3.1.2 Softwares used in Market

- Diesel-RK
- Lotus Engine Simulation by Lotus Cars
- GT Power by Gamma Technologies
- BOOST by AVL
- WAVE by Ricardo Engine
- Virtual Engine by FEV

3.1.3 Software Selection

Many softwares such as GT Power, BOOST, Virtual Engine etc. were inaccessable. No education or free-trial versions of these programmes can be found. Only the three softwares that can be found were Diesel-RK, Lotus Engine Simulation and WAVE. The explanations and comparisons of these three softwares will be in the discussion section.

3.2 Explanation of Softwares

3.2.1 Diesel-RK

DIESEL-RK is a full cycle thermodynamic engine simulation software. One is designed for simulating and optimizing working processes of two and four-stroke internal combustion engines with all types of boosting.[9] The program can be used to calculate:

- Fuel consumption prediction and optimization.
- Torque curve and other engine performances predictions.
- Combustion and emission analysis.
- Valve timing optimization.
- Turbocharger and bypasses matching and optimization.

3.2.2 Lotus Engine Simulation

Lotus Engıne Simulation is a simulation program capable of predicting the complete performance of an engine system.[10] The program can be used to calculate:

- the full- and part-load performance of the engine under steady-state and transient operating conditions;
- in-cylinder heat transfer data;
- instantaneous gas property variations within the engine manifolds;
- turbocharger and supercharger matching conditions.

3.2.3 Ricardo WAVE

WAVE is a state-of-the-art 1D gas dynamics simulation tool. It is used worldwide in industry sectors including ground transportation, rail, motor sport, marine and power generation. WAVE enables performance and acoustic analyses to be performed for virtually any intake, combustion and exhaust system configuration.[11] The program can be used to calculate:

- Engine performance
- Acoustics and noise
- Combustion and emissions
- Thermal analysis
- Dynamic system control
- Real-time analysis
- 1D/3D CFD co-simulation

3.3 Comparison of Softwares

Although, all three softwares are capable of doing engine performance analysis from simple to complex design and calculating the datas of the combustion cycle of an engine, they have differences. For example, in Lotus Engine Simulation, user needs to drag the components one by one to create the design by hand and after completing the design, user decides the parameters of the components by selecting them one by one. On the other hand, in Diesel-RK, the design will be completed by the software itself with selecting the parameters from the beginning. However, Lotus Engine Simulation allows the users to get the chance to see the whole steps of their engine designs and makes it more practical when any modifications needed. In contrast to that, Diesel RK has a more simplistic and user-friendly design which even the inexperienced users can use the software.

On the other hand, WAVE has a unique session tree that the other two softwares lack. The purposes of the session tree are modifiying the parameters easily, showing all the components of the design, adding different cases to the WAVE Solver. Besides that, WAVE can complete the design by adding appropriate components such as ducts and valves. Lastly, the help section of WAVE was simpler and more discriptive than the other two softwares.

The interface of Lotus Engine Simulation is shown in figures 3-1, 3-2 and 3-3 below.

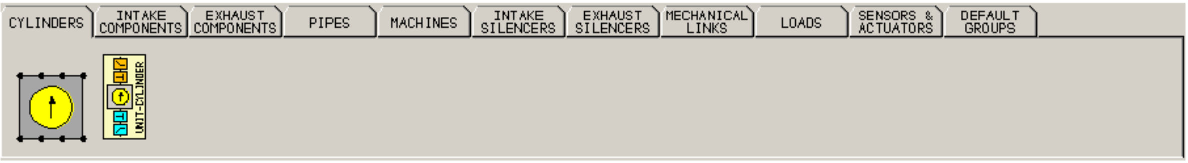


Figure 3-1: Lotus Engine Simulation Interface Panel

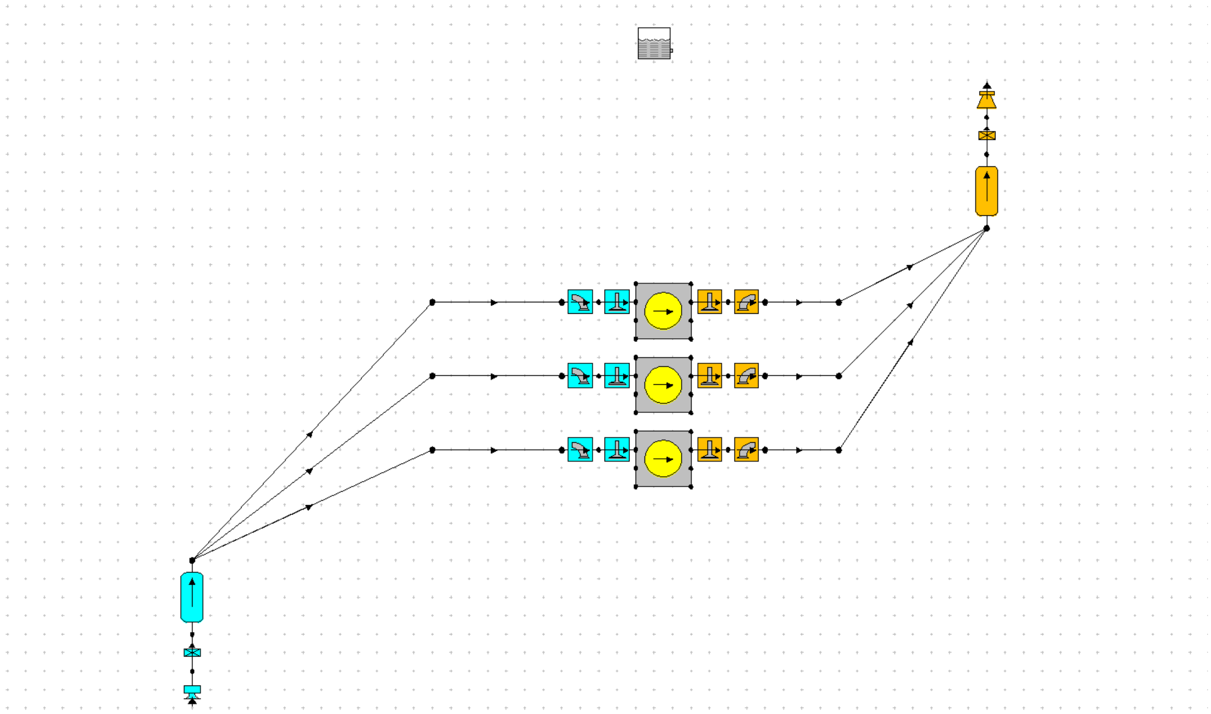


Figure 3-2: Model of a 3-Cylinder Diesel Engine in Lotus Engine Simulation

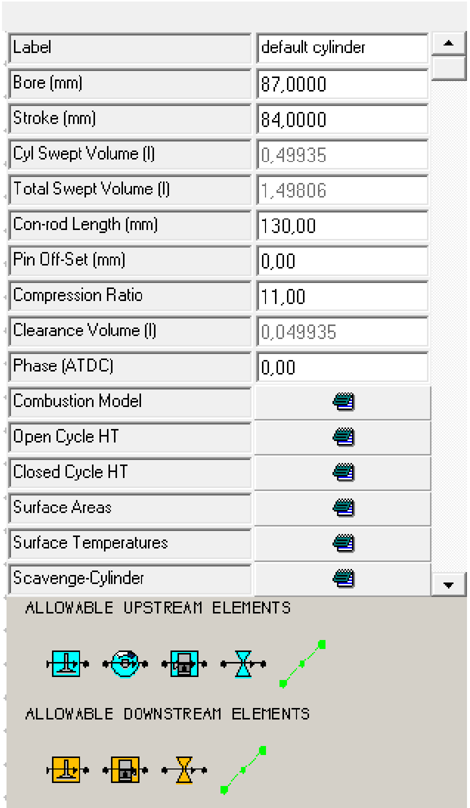


Figure 3-3: Lotus Engine Simulation Parameter Panel

The interface of Diesel-RK is shown in figures 3-4 and 3-5 below.

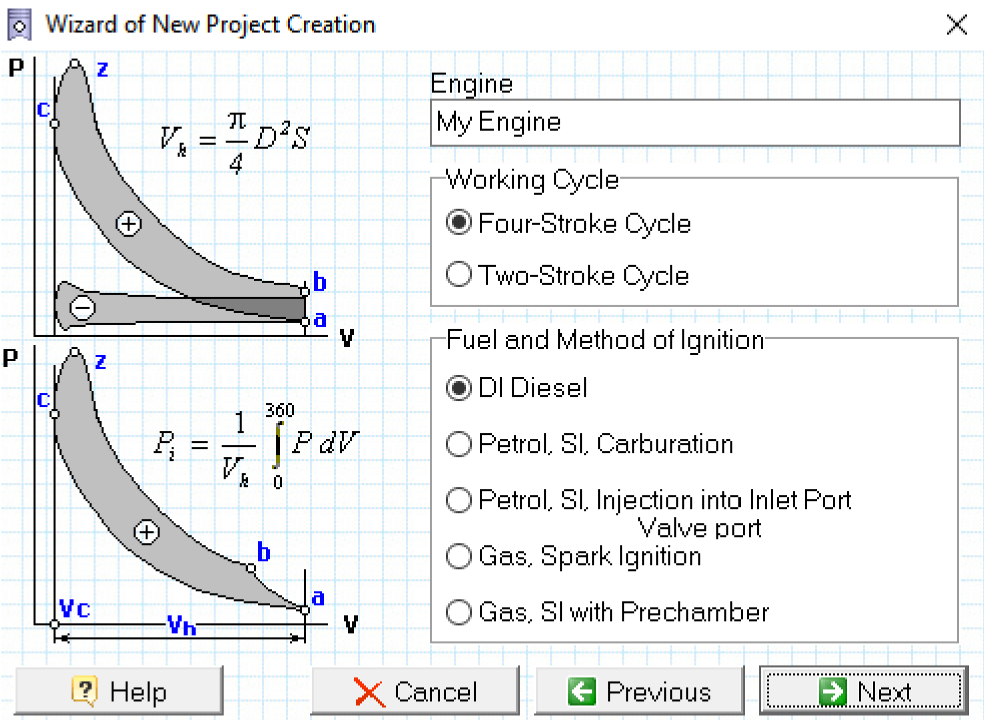


Figure 3-4: Diesel-RK Parameter Panel

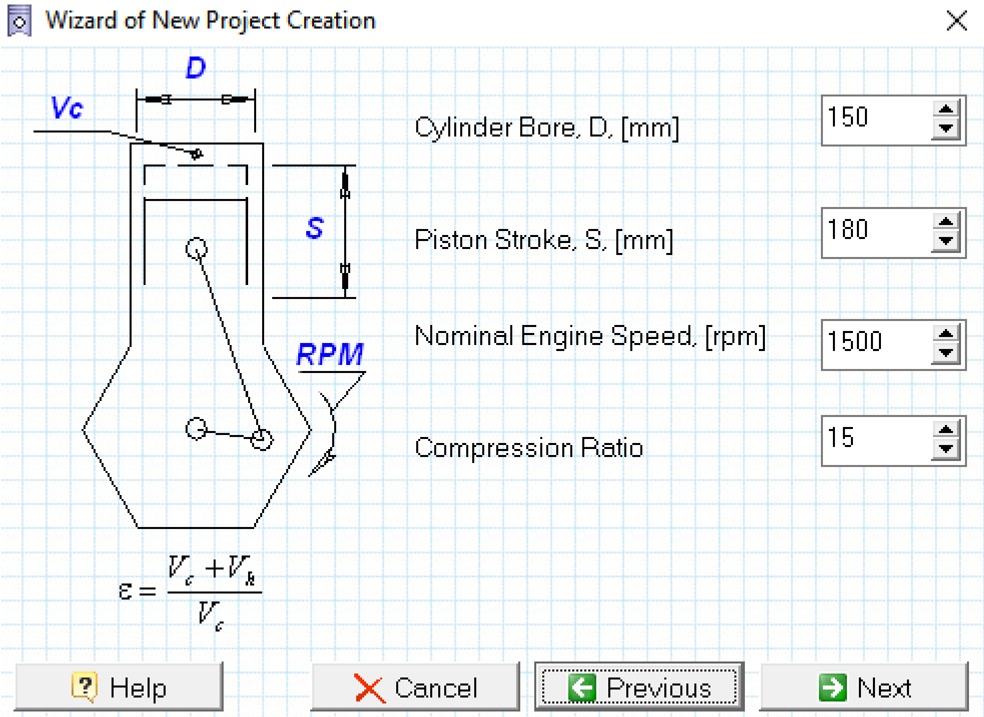


Figure 3-5: Diesel-RK Parameter Panel

The interface of WAVE is shown in figures 3-6, 3-7 and 3-8 below.

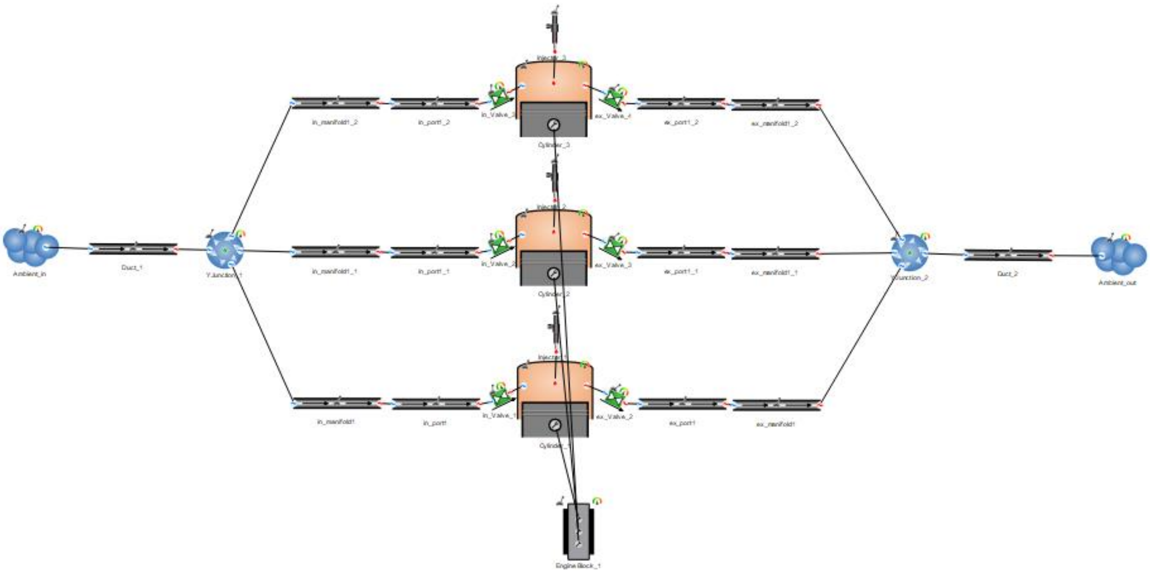


Figure 3-6: Model of a 3-Cylinder Diesel Engine in Ricardo WAVE

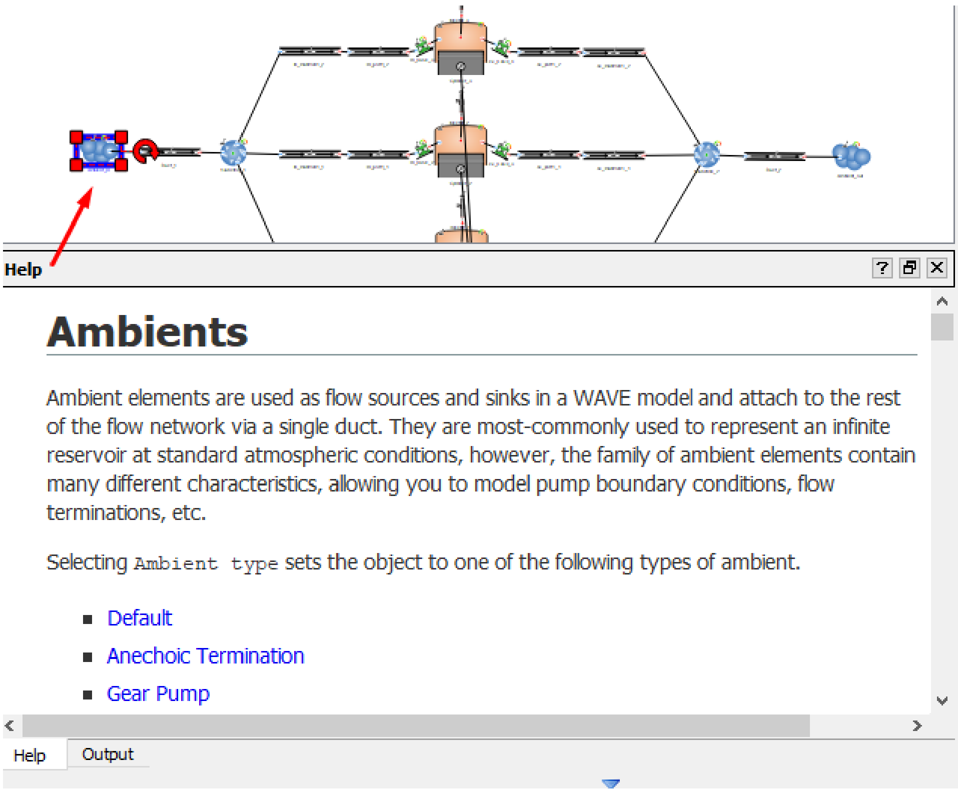


Figure 3-7: Help Section of Ricardo WAVE

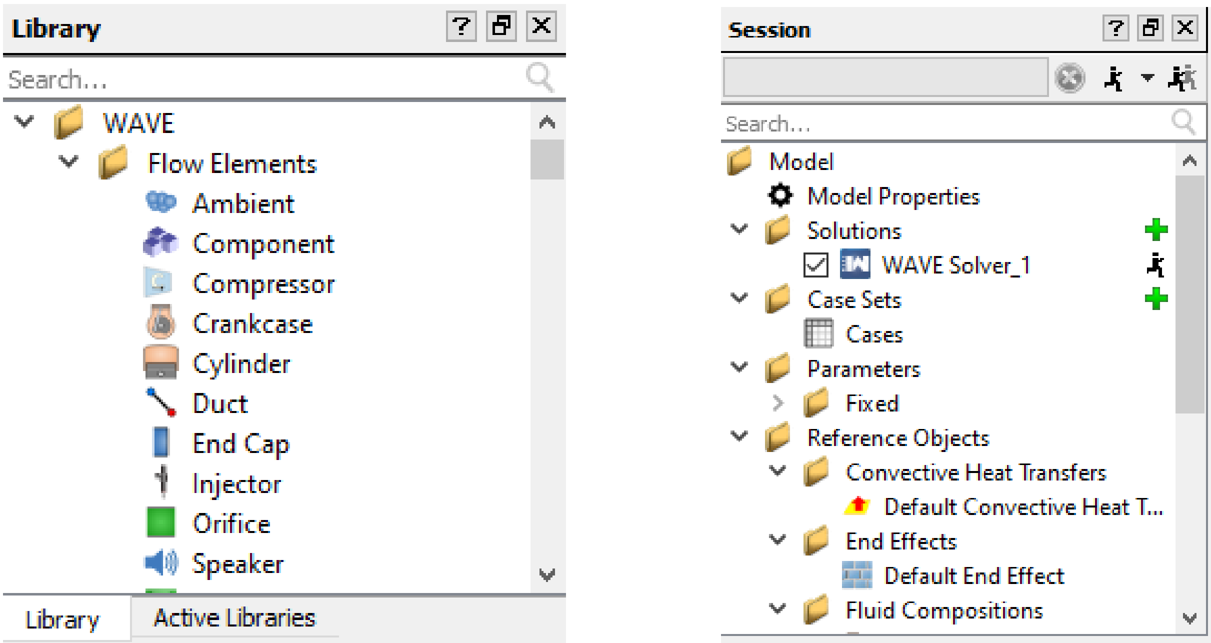


Figure 3-8: Library and Session Trees of Ricardo WAVE

Another difference is how they present the results and how easily the user can get the desired data. In Diesel-RK, user can access different results and graphs easily as all the results are better catagorized and each category has better access to all datas which can be dragged from data section to plot section. Both sections are in the same window. On the other hand, in Lotus

Engine Simulation, results do not have any overwiev section which makes them harder to access. However, Lotus Engine Simulation has one advantage over Diesel RK in result section. The user can see some specific data by clicking onto the graphs. Ricardo WAVE's plot and data selection is far superior than Lotus Engine Simulation and Diesel-RK. The result panel of WAVE is better categorized than the other two softwares and has more explanotary sub-sections for plots and data.

3.4 Results

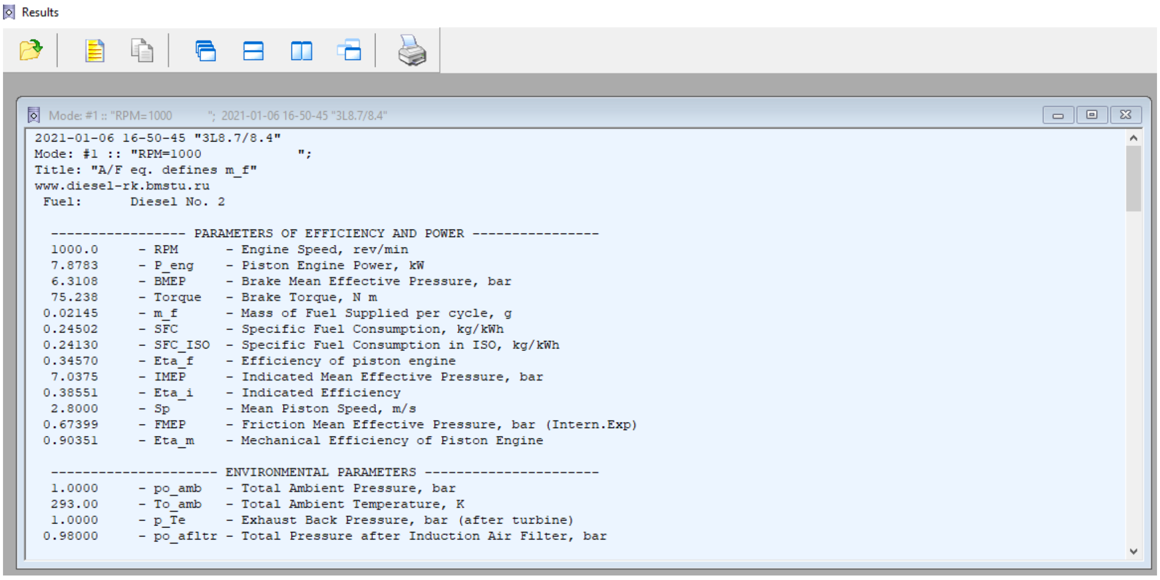
The engine that is chosen for this report was a 3-cylinder inline diesel engine. Parameters of the simulations are given below:

- Nominal engine speed: 1000 RPM
- Cylinder bore: 87.0 mm
- Piston stroke: 84.0 mm
- Compression ratio: 11.0
- Fuel type: Diesel
- Piston/Bore ratio: 1.05
- Head/Bore ratio: 1.10
- Inlet pressure: 100 kPa, 1 bar
- Inlet temperature: 20 ˚C
- Specific humidity: 0.130

The results for the given parameters are given below.

3.4.1 Diesel-RK

In Diesel-RK, the result table is easy to read and categorized. Each datum is presented with its information and explanation next to the values. That makes Diesel-RK more user-friendly without considering the graph section.



Results

Mode: #1 :: "RPM=1000 "; 2021-01-06 16-50-45 "3L8.7/8.4"

```
2021-01-06 16-50-45 "3L8.7/8.4"
Mode: #1 :: "RPM=1000          ";
Title: "A/F eq. defines m_f"
www.diesel-rk.bmstu.ru
 Fuel:      Diesel No. 2

  ---------------- PARAMETERS OF EFFICIENCY AND POWER ---------------
  1000.0     - RPM       - Engine Speed, rev/min
  7.8783     - P_eng     - Piston Engine Power, kW
  6.3108     - BMEP      - Brake Mean Effective Pressure, bar
  75.238     - Torque    - Brake Torque, N m
 0.02145     - m_f       - Mass of Fuel Supplied per cycle, g
 0.24502     - SFC       - Specific Fuel Consumption, kg/kWh
 0.24130     - SFC_ISO   - Specific Fuel Consumption in ISO, kg/kWh
 0.34570     - Eta_f     - Efficiency of piston engine
  7.0375     - IMEP      - Indicated Mean Effective Pressure, bar
 0.38551     - Eta_i     - Indicated Efficiency
  2.8000     - Sp        - Mean Piston Speed, m/s
 0.67399     - FMEP      - Friction Mean Effective Pressure, bar (Intern.Exp)
 0.90351     - Eta_m     - Mechanical Efficiency of Piston Engine

  -------------------- ENVIRONMENTAL PARAMETERS ---------------------
  1.0000     - po_amb    - Total Ambient Pressure, bar
  293.00     - To_amb    - Total Ambient Temperature, K
  1.0000     - p_Te      - Exhaust Back Pressure, bar (after turbine)
 0.98000     - po_afltr  - Total Pressure after Induction Air Filter, bar
```

Figure 3-9: Diesel-RK Result Table

However, when it comes to graph section, it was hard to get the desired graph. The graphs correlated with RPM could not be reached. Even though the graph section was seperated into different categories, transfering results from one section to another for comparison was not available.

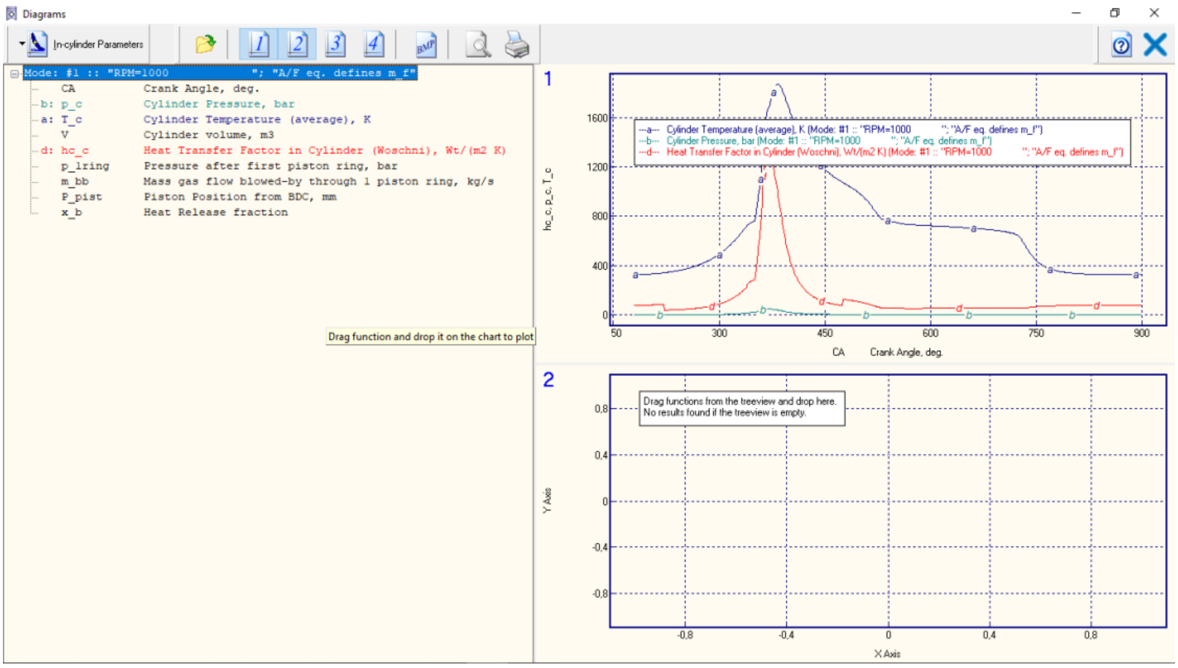


Figure 3-10: Diesel-RK Result Graph Section

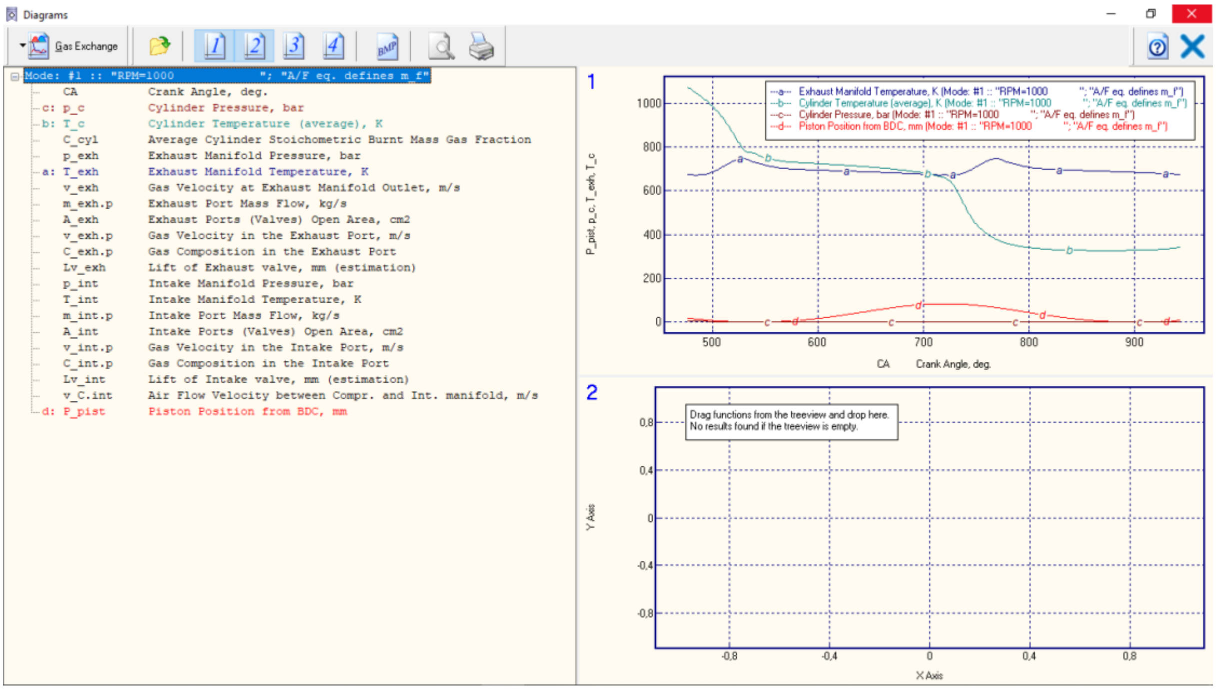


Figure 3-11: Diesel-RK Result Graph Section

3.4.2 Lotus Engine Simulation

In Lotus Engine Simulation, the result table is hard to read and disorganized compared to Diesel-RK and Ricardo WAVE.

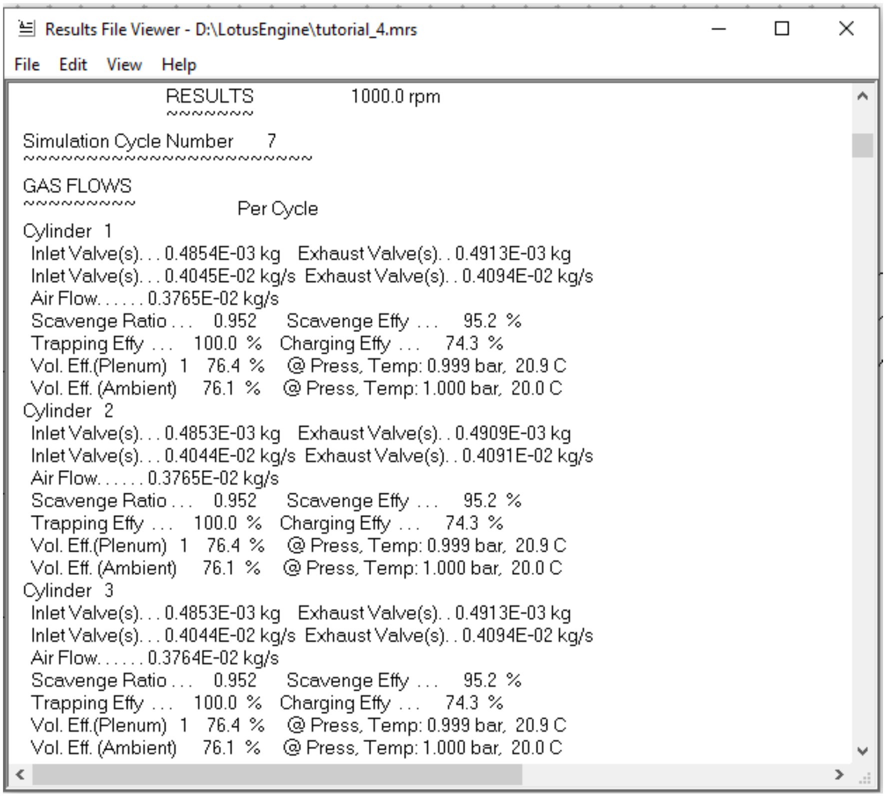


Figure 3-12: Lotus Engine Simulation Result Table

However, graph section is practical and easy to get the desired plots. Graph interface allows you to choose any parameter to compare with another. Lotus Engine Simulation allows users to select any point at the design to get data correlated with this specific point.

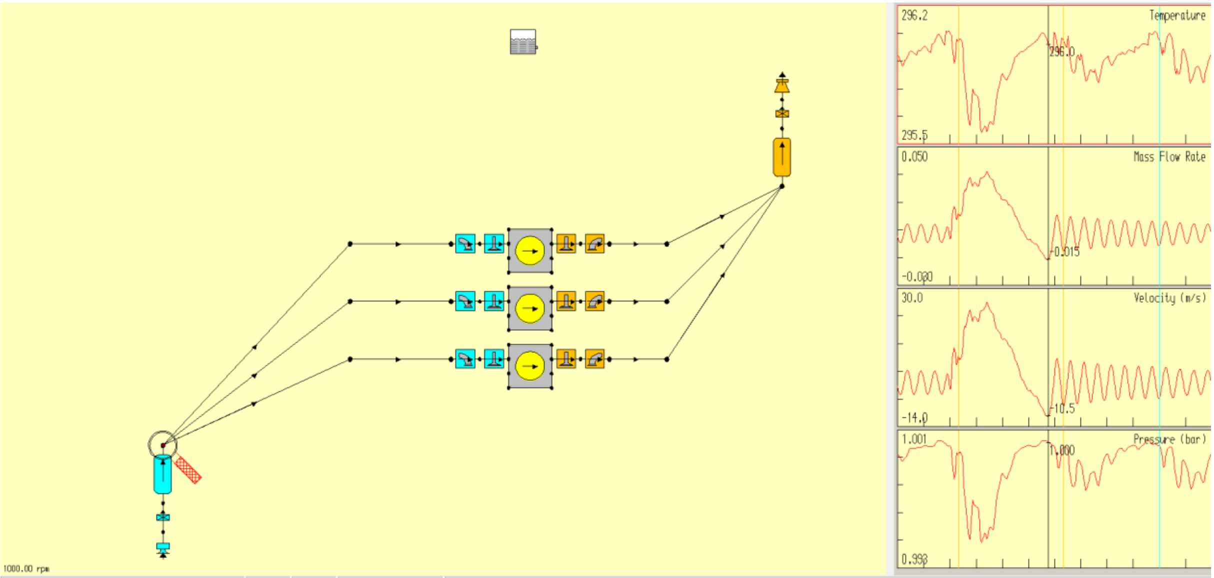


Figure 3-13: Inlet Valve Result Graphs

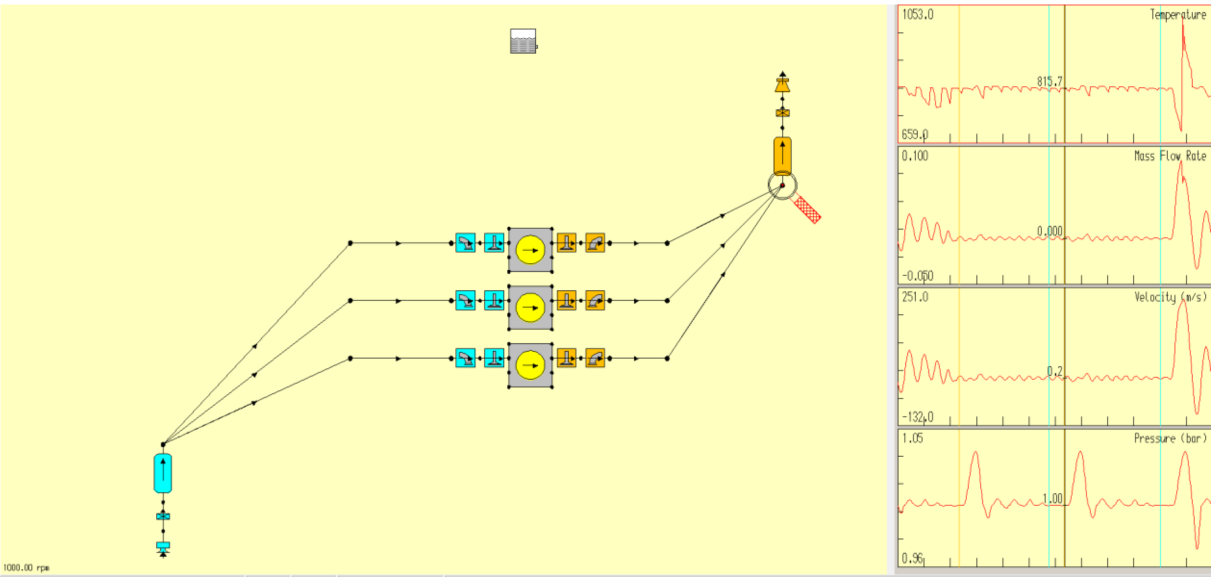


Figure 3-14: Exhaust Valve Result Graphs

Along with the other results, performance summary is also accessable in Lotus Engine Simulation.

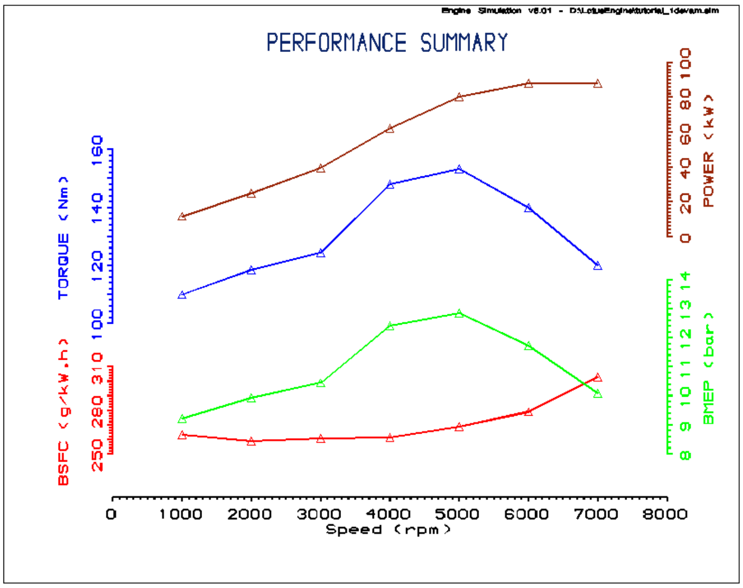


Figure 3-15: Engine Performance Summary in Lotus Engine Simulation

3.4.3 Ricardo WAVE

In Ricardo WAVE, result table section is easy to read and well categorized. It does not have unnecessary information and it is very direct. Data is presented in a table unlike the other two software which is given within the text.

Default Summary Data - dizel3silindir_WAVE_Solver_1.sum, Data for engine

Name Data for engine

Results Table

1	Engine speed	rpm	999	1999	3000	4000
2	Case	-	1	2	3	4
3	Subcase	-	0	0	0	0
4	Case title	-	Case 1	Case 2	Case 3	Case 4
5	Stoichiometric A/F	-	14.5569	14.5569	14.5569	14.5569
6	Trapped A/F	-	24.9089	20.8735	18.0177	15.7865
7	Mass Airflow	kg/hr	21.0507	44.5502	71.9576	97.4557
8	Pseudo-volumetric effici	kg/hr/rpm	0.0210507	0.0222751	0.0239858	0.0243639
9	Auxiliary Power	kW	0	0	0	0
10	BMEP	bar	2.2489	2.87899	3.33894	3.28247
11	Brake Power	kW	2.40823	6.1659	10.7265	14.0601
12	Brake specific CO emissi	g/kW/hr	0	0	0	0
13	BSFC	kg/kW/hr	0.593956	0.572643	0.598381	0.701343
14	Brake specific unburned	g/kW/hr	0.136877	0.137093	0.15053	0.179105
15	Brake specific NO2 emiss	g/kW/hr	0	0	0	0
16	Charging efficiency	-	0.801902	0.835542	0.886175	0.908702
17	Delivered efficiency	-	0.473778	0.505053	0.551389	0.568879
18	Total delivered efficien	-	0.473788	0.505046	0.551377	0.568874
19	Combined Displacement	m^3	0.00128502	0.00128502	0.00128502	0.00128502
20	Brake thermal engine eff	%	14.0367	14.5591	13.9329	11.8875
21	EGR	-	2.09225e-005	-1.35521e-005	-2.09312e-005	-7.61481e-006
22	FMEP	bar	0.974959	1.22569	1.49404	1.75285
23	Friction Energy Loss	%	6.0853	6.19837	6.2344	6.34794
24	Friction torque	N*m	9.96981	12.5338	15.2778	17.9244

OK Apply Cancel Help

Figure 3-16: Ricardo WAVE Result Table of the Engine Block

In addition to that, the graph section is practical and orderly. Desired data are easy to get by using the well categorized plot section which has many sub-sections. Ricardo WAVE gives the option to select components to get data of the selected components individually.

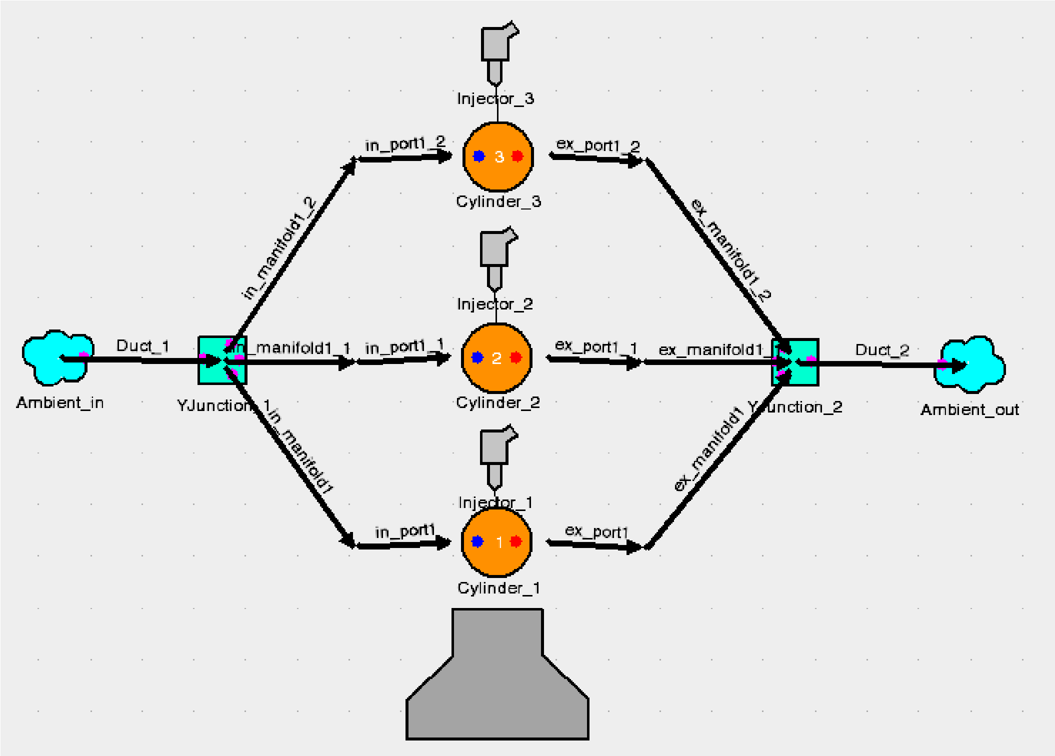


Figure 3-17: Ricardo WAVE Presentation of the Model in Solver Section

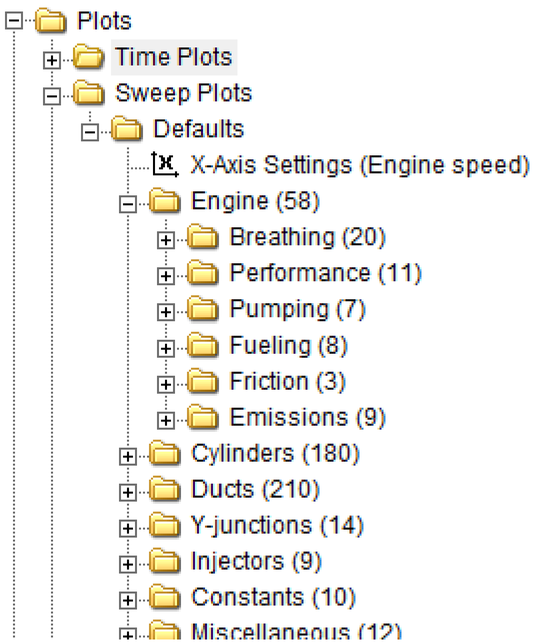


Figure 3-18: Ricardo WAVE Plot Section

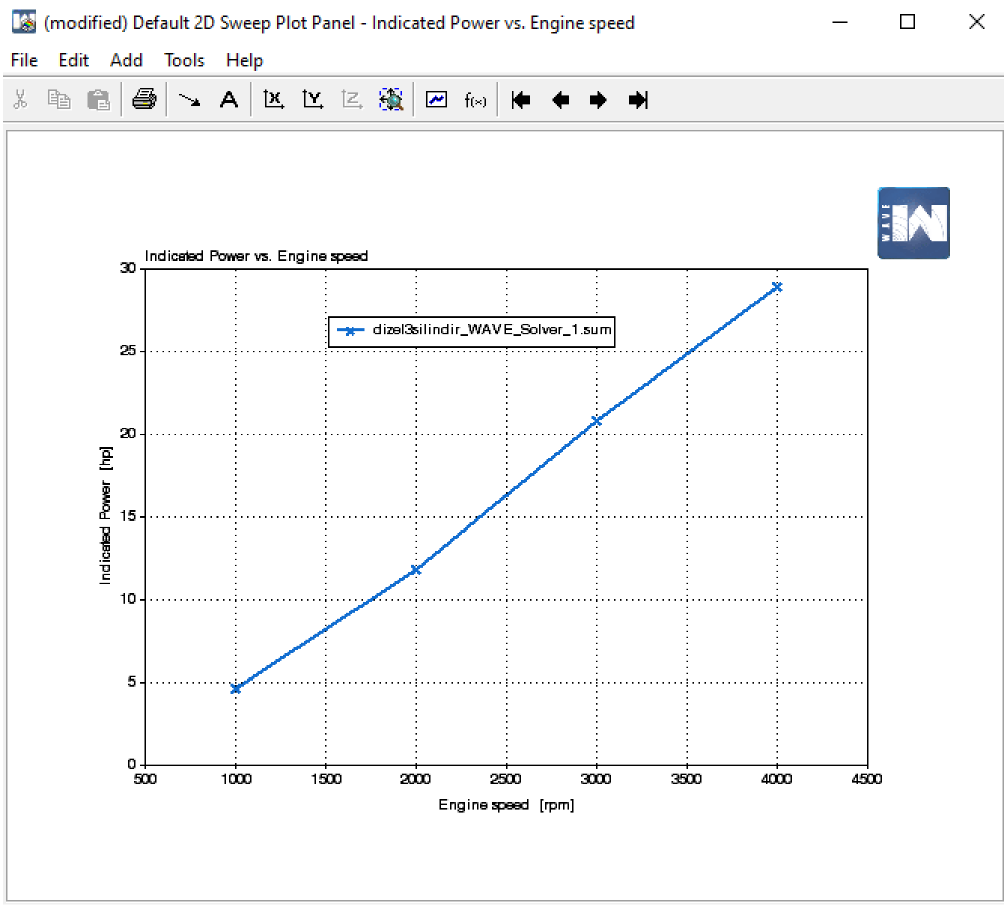


Figure 3-19: Engine Performance Summary in Ricardo WAVE

4 ENGINE SELECTION

1.2 TDI-CR 3-cylinder diesel engine was selected and will be simulated in Ricardo WAVE to be able to see the importance of the technologies that are currently being used in diesel engine market. Turbocharger is the technology that affects the performance the most. Therefore, the chosen engine will be simulated with two different models which are with turbocharger and without turbocharger. The differences will be plotted. The general information, engine specifications, and DPF problems that most of the diesel cars have will be given below.

4.1 General Information

The 1.2 TDI-CR 3-cylinder diesel engine appeared first in 2009 as the new engine for the VW Polo. The 1.2 TDI is derived by cutting one cylinder of the four-cylinder 1.6 TDI-CR engine. The cast iron cylinder block is similar to the 1.6-liter version, but without one cylinder it is shorter and lighter. The cylinder bore and stroke are the same in both 1.2 liter and 1.6 liter versions. The engine got balancing shaft driven by chain from the crankshaft which has counterweights for the first and third cylinder. Balancing shaft is placed under the crankshaft in the crankcase. The cylinder head is made of aluminum alloy, and it has four valves per cylinder (two intake and two exhaust valves), intake and exhaust overhead camshafts (DOHC). Like 1.6 liter, four-cylinder version, the 1.2 TDI timing belt drives exhaust camshaft as well as the high-pressure fuel pump, and the intake camshaft drives by gear from exhaust camshaft. The engine has hydraulic tappets/lifters. The engine is equipped with a control system, variable geometry intake manifold, variable geometry turbocharger, diesel particulate filter and low-temperature EGR system. The detailed technical parameters of this engine are given in the table below.

4.2 Engine Specifications

Cylinder block material	Cast Iron
Cylinder head material	Aluminum
Fuel type	Diesel
Fuel system	Common Rail
Configuration	Inline
Number of cylinders	3
Valves per cylinder	4
Valvetrain layout	DOHC
Bore, mm	79.5
Stroke, mm	80.5
Displacement, cc	1199
Type of internal combustion engine	Four-stroke, turbocharged
Compression Ratio	16.5:1
Power, hp	75/4200 (55kW)
Torque, lb-ft	132 lb-ft (180 Nm)/2,000
Firing order	1-2-3

Table 4-1: Engine Specifications of the 1.2 TDI-CR

4.3 DPF Problems

A diesel particulate filter (DPF) is a filter that captures and stores exhaust soot in order to reduce emissions from diesel cars. DPF is a must for all new diesel cars since the Euro 5 emission standards were introduced in 2009. Nowadays, there are two types of DPF in diesel cars:

- active regeneration
- passive regeneration

Active regeneration uses the engine control unit to sense when the filter is getting clogged with soot and injects extra fuel into the engine to raise the temperature of the exhaust, triggering regeneration. Passive regeneration occurs when the car is running at speed on long motorway journeys which allows the exhaust temperature to increase to a higher level and cleanly burn off the excess soot in the filter. However, in order the passive regeneration to work, the vehicle must be travelling at least 15 minutes and/or 10 miles, to reach at an exhaust temperature which can burn the exhaust soot.

4.4 Results

The engine that was chosen before will be simulated in this section with using WAVE software with the parameters given in Table 4.1. Two different design will be used to get a better understanding on how the turbocharger affects the engine performance and efficiencies. The simulations will be run with five different RPM values, from 1000 to 5000, to see the different results and how the turbocharger behaves. The model of the engine is given below.

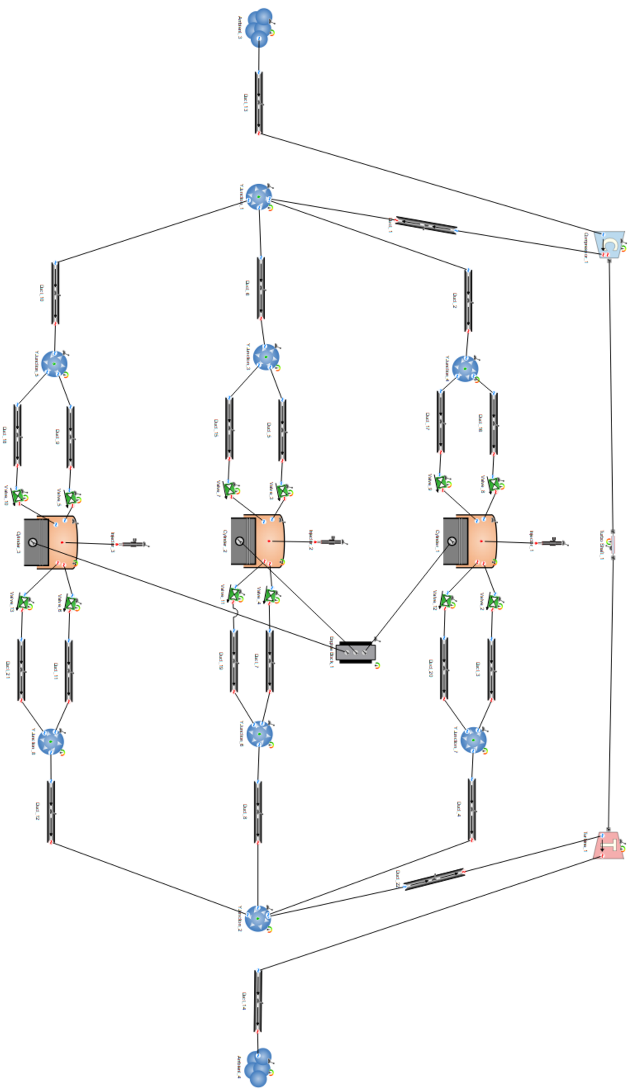


Figure 4-1: 3-cylinder turbocharged diesel engine modelled in WAVE

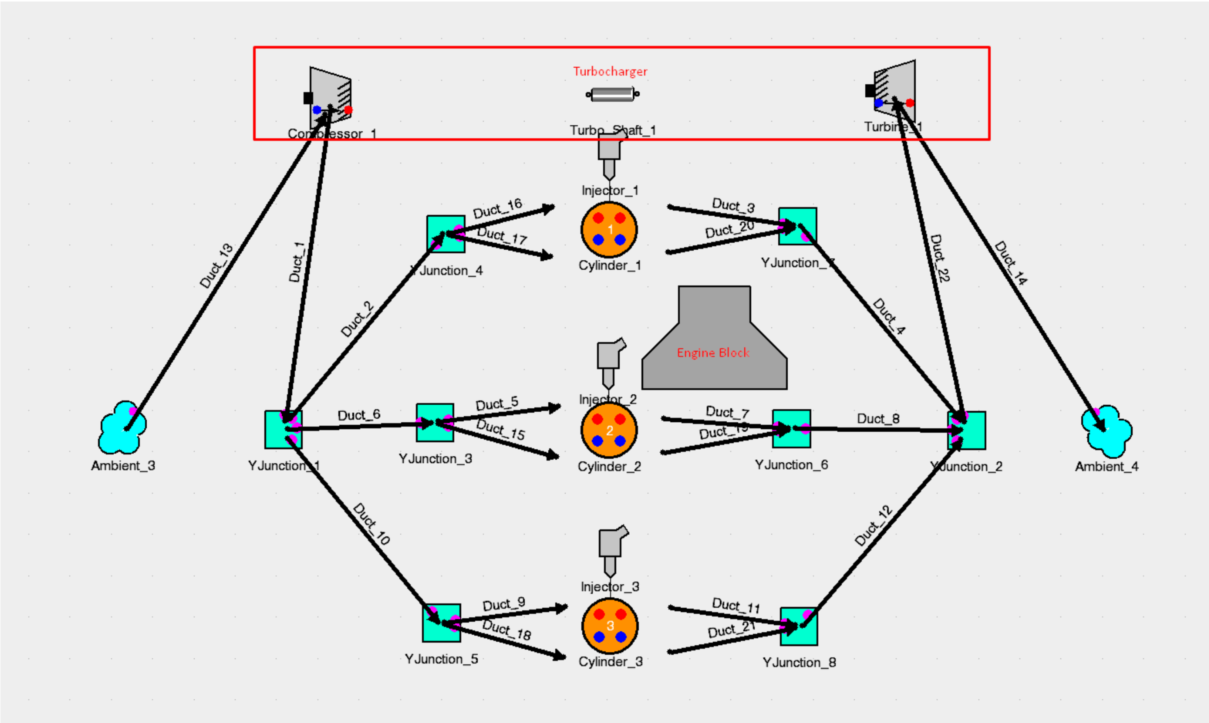


Figure 4-2: Ricardo WAVE Presentation of the Model in Solver Section

The simulation results of the modelled engine is given in figures 4-3, 4-4, 4-5 below.

Default Summary Data - Diesel_3_Cylinder_4_valves_WAVE_Solver_1.sum, Data for engine

Name: Data for engine

Results Table

1	Engine speed	rpm	999	2000	3000	3999	4999
2	Case	-	1	2	3	4	5
3	Subcase	-	0	0	0	0	0
4	Case title	-	Case 1	Case 2	Case 3	Case 4	Case 5
5	Stoichiometric A/F	-	14.5569	14.5569	14.5569	14.5569	14.5569
6	Trapped A/F	-	19.8511	16.597	14.3408	11.0758	9.05612
7	Mass Airflow	kg/hr	42.9591	112.049	216.565	281.272	307.707
8	Pseudo-volumetric effici(	kg/hr/rpm	0.0429593	0.0560243	0.0721884	0.070318	0.0615413
9	Auxiliary Power	kW	0	0	0	0	0
10	BMEP	bar	8.25253	13.491	18.5923	16.3855	12.0852
11	Brake Power	kW	8.1046	26.4985	54.7773	64.3672	59.3428
12	Brake specific CO emissi(	g/kW/hr	0	0	0	0	0
13	BSFC	kg/kW/hr	0.239333	0.236595	0.263448	0.388469	0.572568
14	Brake specific unburned :	g/kW/hr	0.120115	0.118503	0.130926	0.193484	0.280197
15	Brake specific NO2 emiss:	g/kW/hr	0	0	0	0	0
16	Charging efficiency	-	0.834981	0.907918	0.981164	0.98711	0.89514
17	Delivered efficiency	-	0.931566	0.977678	1.02674	1.00252	0.895119
18	Total delivered efficien(	-	0.931538	0.977675	1.02674	1.00252	0.895132
19	Combined Displacement	m^3	0.00117849	0.00117849	0.00117849	0.00117849	0.00117849
20	Brake thermal engine eff:	%	34.8351	35.2382	31.6465	21.4616	14.5611
21	EGR	-	-2.98047e-005	-2.5675e-006	5.68292e-007	1.4932e-006	1.49187e-005
22	FMEP	bar	1.23084	1.61549	2.13224	2.4125	2.61095
23	Friction Energy Loss	%	5.19556	4.21962	3.62934	3.15988	3.14586
24	Friction torque	N*m	11.5431	15.1504	19.9966	22.6249	24.486

OK Apply Cancel Help

Figure 4-3: Data for engine

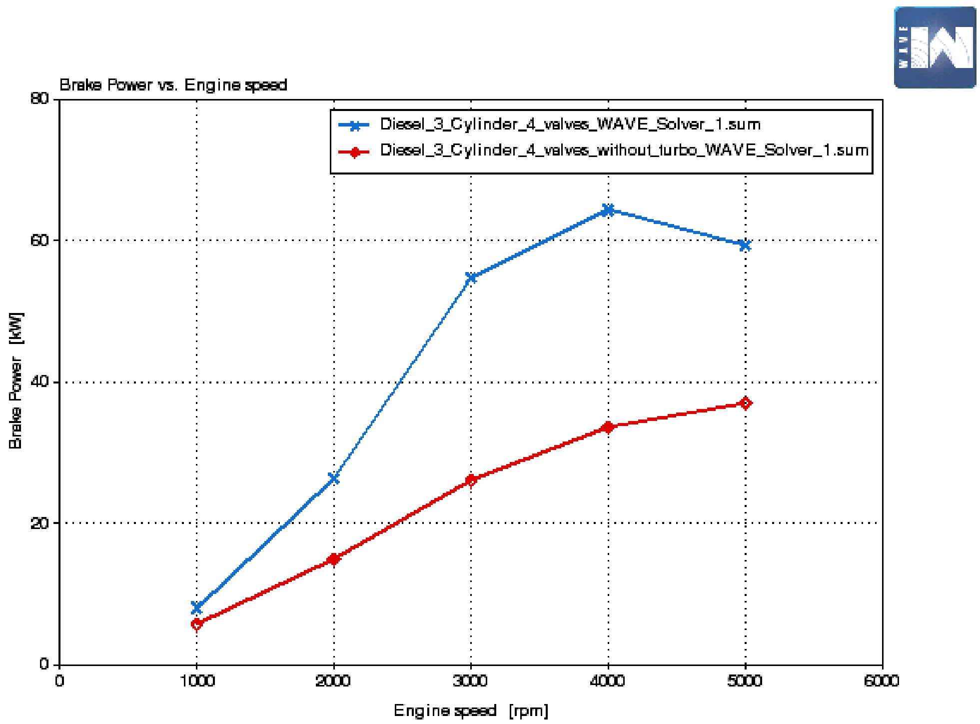


Figure 4-4: Plot of break power-engine speed comparison between 3-cylinder turbocharged diesel engine vs. 3-cylinder diesel engine

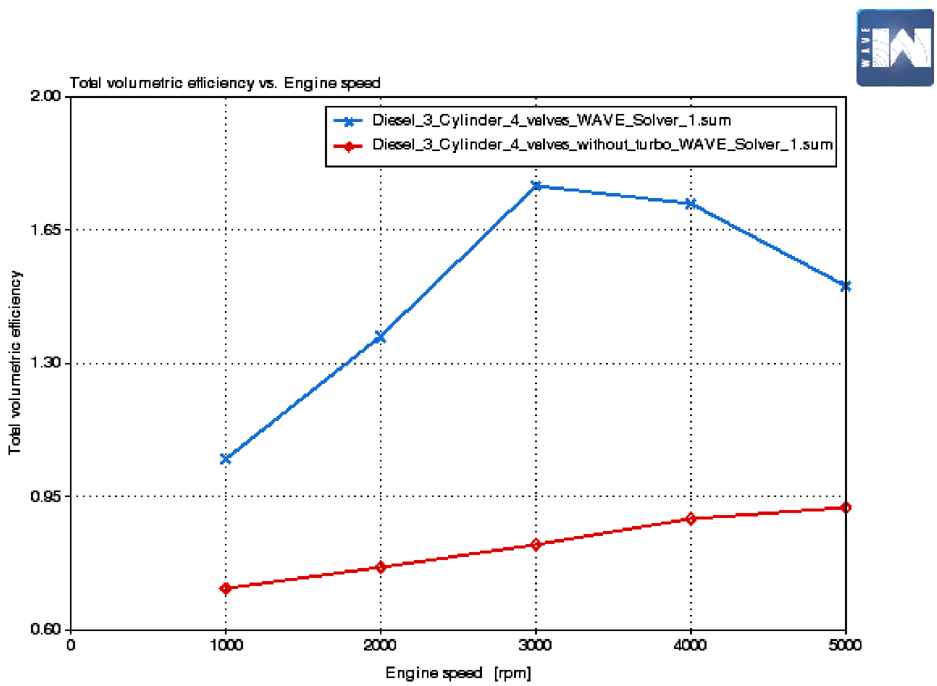


Figure 4-5: Plot of total volumetric efficiency-engine speed comparison between 3-cylinder turbocharged diesel engine vs. 3-cylinder diesel engine

The effect of turbocharger on the engine performance and efficiencies can be easily seen in the simulation graphs. The engine power was substantially increased and the graph peaked at 4000 RPM. The total volumetric efficiency was more than doubled compared to the one from the engine without turbocharger. This engine power from the simulation was within close range of the original power value at 4200 RPM that was given in Table 4.1. Results shows that the simulation was accurate.

5 GOVERNING EQUATIONS

5.1 Single-Zone Model

The simplest approach in engine modelling is to treat the cylinder contents as a solitary fluid or zone. The single-zone model views the burned and unburned gases, residual gases, and unburned HC within the cylinder as an ideal gas with uniform pressure. The Wiebe function can be defined as follows:

$$X_b\,(\theta) = \; 1 - \exp[-a(\frac{\theta_i - \theta_0}{\theta_b})^{k+1}] \tag{1}$$

where $\theta(i)$ is instantaneous crank angle, θ_0 is the crank angle at the start of combustion, θ_b is the burn duration, and a and k are adjustable constants (5 and 2 are commonly used values).

The ideal gas law is defined as:

$$PV = mRT \tag{2}$$

where P is pressure of the ideal gas, V is the volume of the gas, m is the mass of the gas, R is the universal gas constant, and T is the mean gas temperature.

Instantaneous cylinder volume as a function of crank angle is given below:

$$V(\theta) = V_c + \frac{\pi B^2}{4}(l_r + a - s) \tag{3}$$

where B is the cylinder bore, l_r is the connecting rod length, a is the crank radius, s is the instantaneous distance between the crank axis and the piston pin axis, and V_c is the clearance volume.

The clearance volume is given below:

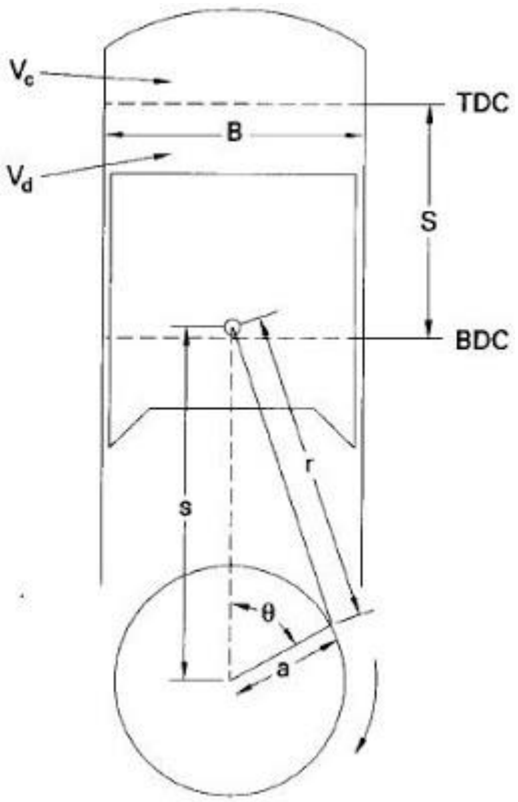
$$V_c = \frac{V_d}{c_r - 1} \tag{4}$$

where V_d is the displaced cylinder volume, and c_r is the compression ratio.

Displacement volume V_d is expressed as:

$$V_d = \frac{\pi}{4} B^2 S \tag{5}$$

where B is bore and S is stroke.



Equation for change in pressure, dP, is shown below:

$$\frac{dP}{d\theta} V + \frac{dV}{d\theta} P = mR \frac{dT}{d\theta} \tag{6}$$

$$\frac{dP}{d\theta} = \left(-\frac{P}{V}\right) \frac{dV}{d\theta} + \frac{P}{T} \left(\frac{dT}{d\theta}\right) \tag{7}$$

The first law of thermodynamics defined as:

$$\Delta U = Q - W \tag{8}$$

where ΔU is the internal energy within the system, Q is the total energy transferred into the system, and W is the work transferred out of the system.

Differentiating the previous equation:

$$\frac{dU}{d\theta} = \frac{dQ}{d\theta} - \frac{dW}{d\theta} = mC_v \left(\frac{dT}{d\theta}\right) \tag{9}$$

where C_v is the specific heat of the gas.

$$\frac{C_v}{R} = \frac{C_v}{C_p - C_v} = \frac{1}{\gamma - 1} \tag{10}$$

Equations (11) and (12) can be used to describe the formation of work and the net heat input:

$$\frac{dW}{d\theta} = P(\frac{dV}{d\theta}) \tag{11}$$

$$\frac{dQ}{d\theta} = \eta_c LHV\left(\frac{dX_b}{d\theta}\right) - \frac{dQ_w}{d\theta} \tag{12}$$

where η_c is the combustion efficiency, LHV is lower heating value of the fuel, $\left(\frac{dX_b}{d\theta}\right)$ is the instantaneous change in mass-fraction burned, and $\frac{dQ_w}{d\theta}$ is the instantaneous change in heat loss to the cylinder walls.

The instantaneous change in temperature is defined as:

$$\frac{dT}{d\theta} = T(\gamma - 1)[\left(\frac{1}{PV}\right)\left(\frac{dQ}{d\theta}\right) - (\frac{1}{V})(\frac{dV}{d\theta})] \tag{13}$$

The change in pressure defined as:

$$\frac{dP}{d\theta} = \left(\frac{-\gamma P}{V}\right)\left(\frac{dV}{d\theta}\right) + \left(\frac{\gamma - 1}{V}\right) Q_{in}\frac{dX_b}{d\theta} + (\gamma - 1)(\frac{1}{V})(\frac{dQ_w}{d\theta}) \tag{14}$$

where γ specific heat ratio, $\frac{dV}{d\theta}$ is instantaneous change in volume, and Q_{in} is heat input which is defined below:

$$Q_{in} = \eta_c LHV\left(\frac{1}{AF_{ac}}\right)(\frac{P}{RT})V_d \tag{15}$$

$$AF_{ac} = \lambda AF_{stoich} \tag{16}$$

where AF_{ac} is actual air/fuel ratio, λ is excess air coefficient, and AF_{stoich} is stoichiometric air/fuel ratio.

5.2 Annand's Heat Transfer Model

Annand's heat transfer model assumed pipe like fluid dynamics to be able to calculate heat transfer coefficient by using the in-cylinder density and Reynolds and Nusselt numbers as functions of time.

With Annand's method, Newton's law of cooling can be broken down as:

$$\delta Q_w = (h_c + h_r)A_w(T - T_w)dt \tag{17}$$

where Q_w is the convective heat loss, h_c is the convective heat transfer coefficient, h_r is the radiative heat transfer coefficient, A_w is the instantaneous heat transfer area, T is the instantaneous bulk gas temperature, and T_w is the cylinder wall temperature.

The convective heat transfer coefficient can be written as:

$$h_c = \frac{k_{gas}Nu}{B} \tag{18}$$

where k_{gas} is the gas thermal conductivity, is Nu the Nusselt number, and B is the cylinder bore. Formulas for gas thermal conductivity and Nusselt number are given below:

$$k_{gas} = 6.1944 \times 10^{-3} + 7.3814 \times 10^{-5}T - 1.2491 \times 10^{-8}T^2 \tag{19}$$

$$Nu = aRe^{0.7} \tag{20}$$

The transition from a laminar into a turbulent boundary layer begins at a certain point. This point is determined by a dimensionless number called the Reynolds number (Re). In internal combustion engines, the Reynolds number can be defined as a function of gas properties and engine cylinder geometry:

$$Re = \frac{\rho_{gas}\overline{S_p}B}{\mu_{gas}} \tag{21}$$

where ρ_{gas} is the instantaneous cylinder gas density, $\overline{S_p}$ is the mean piston velocity, and μ_{gas} is the instantaneous gas viscosity.

The gas density can be defined by rearranging the ideal gas law which is given in equation (2):

$$\rho_{gas} = \frac{P}{R_{air}T} \tag{22}$$

where R_{air} is the fluid-specific gas constant, and is assumed as a value of 287 $\frac{J}{kg \times K}$.

According to Blair, the cylinder gas viscosity can be expressed as:

$$\mu_{gas} = 7.457 \times 10^{-6} + 4.1547 \times 10^{-8}T - 7.4793 \times 10^{-12}T^2 \tag{23}$$

where the instantaneous cylinder temperature must be provided in units of K, and the gas viscosity is output in units of $\frac{kg}{m \times s}$.

Generally, the radiation part would be ignored because its effects are insignificant compared to convection part. Although it is included in this report and is defined as:

$$h_r = 4.25 \times 10^{-9} \left(\frac{T^4 - T_w^4}{T - T_w} \right) \tag{24}$$

5.3 Engine Friction Model

In internal combustion engines, since they are complex machineries, there are a lot of sources for the energy losses such as pumping, mechanical rubbing, and auxiliary device losses. The sum of all these losses are represented as friction mean effective pressure (fmep). These losses increase with the engine speed although not linearly. According to Blair, friction mean effective pressure can be expressed as a function of engine speed with a critical displacement volume value. If the displacement volume of the engine is larger than or equal to 500×10^{-6}, fmep formula can be expressed as:

$$fmep = (100000 + 350 \times L \times N) \times 10^{-3} \tag{25}$$

If the displacement volume of the engine is smaller than 500×10^{-6}, fmep formula is written as below:

$$fmep = (100000 + 100 \times (500 - V_d \times 10^{-6}) + 350 \times L \times N) \times 10^{-3} \tag{26}$$

where L is stroke in meters, N is engine speed in RPM.

5.4 Residual Gas Fraction Model

Using polytropic assumptions during the valve overlap period, the exhaust gas temperature is defined as:

$$T(i) = T_{EVO} (\frac{P_{BDC}}{P_{EVO}})^{\frac{\gamma-1}{\gamma}} \tag{27}$$

where T_{EVO} and P_{EVO} are the temperature and pressure, respectively, at exhaust valve opening (EVO), and P_{BDC} is the pressure at bottom-dead-center (BDC).

5.5 Burned and Unburned Gas Temperature Calculations

In order to determine the burned and unburned masses at BDC, the following three equations are used:

$$m_a = \rho_a V_d \tag{28}$$

$$m_f = \frac{m_a}{AF_{ac}} \tag{29}$$

$$m_c = m_a + m_f \tag{30}$$

where m_a is the mass of air contained within the cylinder, ρ_a is the density of air, m_f is the mass of fuel, and m_c is the total mass contained within the cylinder.

The Wiebe function can be used to divide the combustion chamber gases into unburned and burned masses which can be defined as:

$$m_b(i) = m_b(i-1) + \frac{dX_b}{d\theta}(i)m_c \tag{31}$$

$$m_u(i) = m_u(i-1) + \frac{dX_b}{d\theta}(i)m_c \tag{32}$$

The initial quantity of burned mass is assumed as zero and unburned is defined as 1.

The unburned volume is defined as:

$$V_u(i) = \left(\frac{m_u(i)V_u(i-1)}{m_u(i-1)}\right)\left(\frac{P(i)}{P(i-1)}\right)^{-\frac{1}{\gamma_u(i)}} \tag{33}$$

where $\gamma_u(i)$ is the instantaneous specific heat ratio of the unburned region. With the unburned volume defined, the burned volume is determined by using the relationship:

$$V(i) = V_b(i) + V_u(i) \tag{34}$$

The burned and unburned temperatures are defined as:

$$T_b(i) = \frac{P(i)V_b(i)}{m_b(i)R(i)} \tag{35}$$

$$T_u(i) = \frac{P(i)V_u(i)}{m_u(i)R(i)} \tag{36}$$

where P(i) is the instantaneous bulk pressure and R(i) is the instantaneous, fluid-specific gas constant.

The unburned and burned areas are defined as:

$$A_u(i) = A(i)\left(1 - (X_b(i))^{\frac{1}{2}}\right) \tag{37}$$

$$A_b(i) = A(i)\left(\frac{X_b(i)}{\left(X_b(i)\right)^{\frac{1}{2}}}\right) \tag{38}$$

where ( ) is the instantaneous mass-fraction burned and ( ) is the overall, instantaneous heat transfer area.

The overall, instantaneous heat transfer area is expressed as:

$$A(i) = A_{ch} + \frac{\pi B^2}{2} + \pi B(l + a - s(i)) \tag{39}$$

where A_{ch} is the surface area of the cylinder heat, a is the crank radius, and $s(i)$ is the instantaneous distance between the crank axis and the piston pin axis.

The instantaneous distance between the crank axis and pin axis is defined as:

$$s(i) = a(cos\theta) + (l^2 - a^2 \sin^2 \theta)^{1/2} \tag{40}$$

6 MATLAB APPDESIGNER ENGINE PERFORMANCE SIMULATION MODEL

For the selection of the computer program, MATLAB GUI and MATLAB App Designer were considered in terms of simplicity of interfaces and code parts. The task was to create an app that can run a code which contains real internal combustion engine calculations and give the result of the calculations on a basic and an understandable interface. To achive this, different engine performance simulation softwares such as, Lotus Engine Simulation, Diesel RK, and Ricardo WAVE were researched and analysed. After careful considerations, Ricardo WAVE was chosen as base model for this app because of its user friendly interface. MATLAB App Designer was a better choice than MATLAB GUI because MATLAB GUI was an old version of MATLAB App Designer. Therefore, MATLAB App Designer was chosen to build this app.

6.1 Components of the App

Since the app has over two thousand lines of code, it could be difficult to understand it as a single part. The code was divided into two parts which are calculations/main interface and result app. The code will be examined with following seven parts:

1. Program inputs.
2. Basic interface buttons.
3. The main part.
4. Program outputs
5. Transferring the results to result app.
6. Result app interface
7. Plots and summary table.

6.2 Program Inputs

Interface of the main app can be seen in figure 6-1 when the main app was started for the first time.

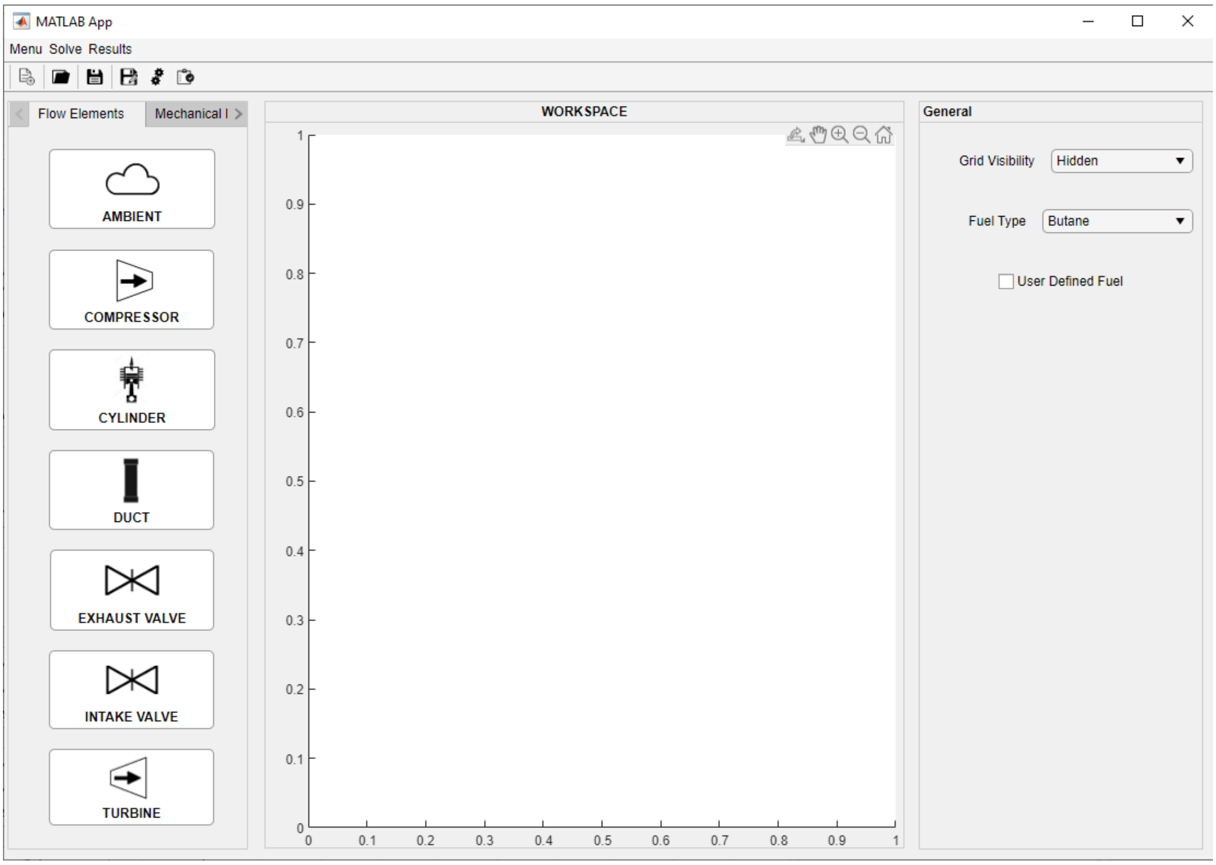


Figure 6-1 Interface of the Main App

The main app contains several number of push buttons and panels. When a push button such as ambient or cylinder is used, a panel that is related to the pushed button will be emerged. At the start the general panel is shown until another button is pushed.

The general panel contains grid visibility, fuel types and an option for user defined fuel type. Grid visibility and fuel types were created by using drop down menu elements. Drop down menu element allows user to select different options which that drop down menu contains. For the grid visibility drop down menu, two options are presented such as visible and hidden. On the other hand fuel type dropdown menu contains ten different fuel types which each one associated with a different lower heating value.

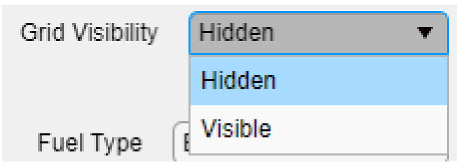


Figure 6-2 Grid Visibility Options

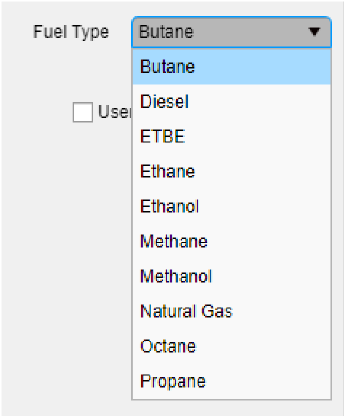


Figure 6-3 Fuel Type Options

```
if strcmp(app.FuelTypeDropDown.Value,'Butane')
    app.LHV = 45277000; %J/kg
elseif strcmp(app.FuelTypeDropDown.Value,'Diesel')
    app.LHV = 42791000; %J/kg
elseif strcmp(app.FuelTypeDropDown.Value,'ETBE')
    app.LHV = 36315000; %J/kg
elseif strcmp(app.FuelTypeDropDown.Value,'Ethane')
    app.LHV = 47611000; %J/kg
elseif strcmp(app.FuelTypeDropDown.Value,'Ethanol')
    app.LHV = 26952000; %J/kg
elseif strcmp(app.FuelTypeDropDown.Value,'Methane')
    app.LHV = 50048000; %J/kg
elseif strcmp(app.FuelTypeDropDown.Value,'Methanol')
    app.LHV = 20094000; %J/kg
elseif strcmp(app.FuelTypeDropDown.Value,'Natural Gas')
    app.LHV = 47141000; %J/kg
elseif strcmp(app.FuelTypeDropDown.Value,'Octane')
    app.LHV = 44786000; %J/kg
elseif strcmp(app.FuelTypeDropDown.Value,'Propane')
    app.LHV = 46330000; %J/kg
end
```

Figure 6-4 Fuel Type Drop Down Value Changed Function

Besides from fuel type drop down menu there is an option for users to define their own lower heating value of the fuel which is created by using a check box. When check box is used, fuel type drop down menu will be disabled and a numeric edit field will be available.

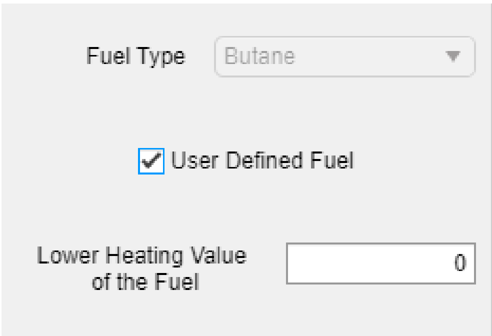


Figure 6-5 User Defined Fuel Check Box

```
userdefinedfuel_value = app.UserDefinedFuelCheckBox.Value;
if userdefinedfuel_value == 0
    app.FuelTypeDropDown.Enable = 'On';
    app.LowerHeatingValueoftheFuelEditField.Visible = 'Off';
    app.LowerHeatingValueoftheFuelLabel.Visible = 'Off';
elseif userdefinedfuel_value == 1
    app.FuelTypeDropDown.Enable = 'Off';
    app.LowerHeatingValueoftheFuelEditField.Visible = 'On';
    app.LowerHeatingValueoftheFuelLabel.Visible = 'On';
    app.LHV = app.LowerHeatingValueoftheFuelEditField.Value;
end
```

Figure 6-6 Code for User Defined Fuel Check Box

The ambient panel consist of two different inputs which are ambient temperature and pressure. Since ambient temperature and pressure directly affect the overall engine performance, user should consider the effects of these two values.

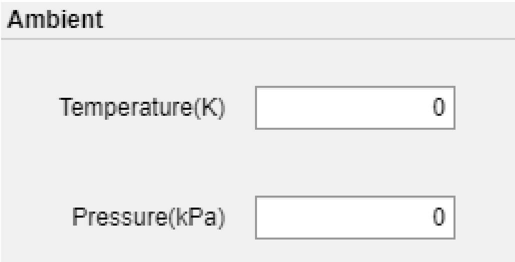


Figure 6-7 Ambient Panel

Cylinder panel contains engine geometry inputs such as, bore, stroke, connecting rod length and compression ratio as well as air/fuel ratio and combustion model. User is expected to enter the air/fuel ratio as its stoichiometric value rather than the real value.

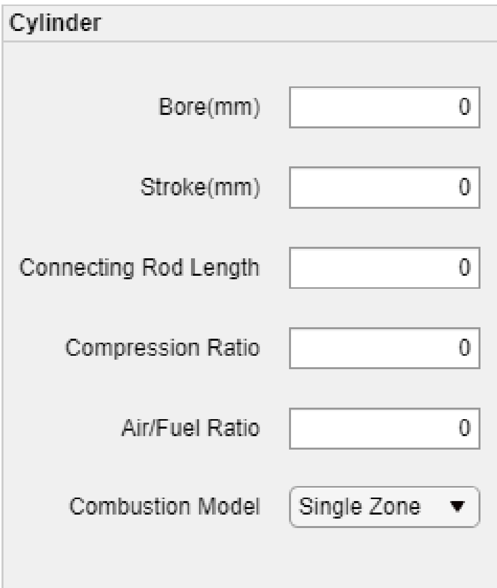


Figure 6-8 Cylinder Panel

The exhaust valve and intake valve panels have two different input values each. The inputs for exhaust valve open and intake valve close should be given carefully because they affect the calculations directly.

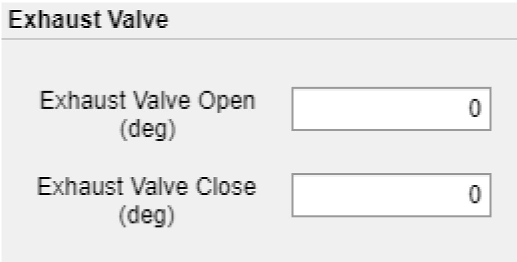


Figure 6-9 Exhaust Valve Panel

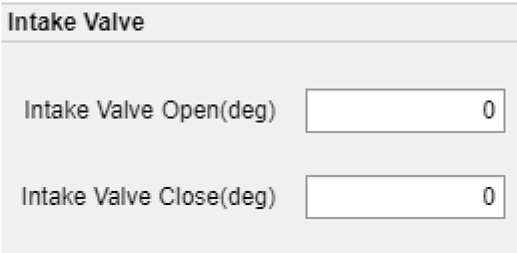


Figure 6-10 Intake Valve Panel

Input push buttons were categorised under two separate sections which are flow and mechanical elements. These two sections will be explained later. In engine block panel, important inputs such as number of cylinders and engine speed were placed. Additionally, different engine speed inputs can be given to the program with using cases drop down menu. This drop down menu allows user to enter up to ten different engine speeds. During the selection of the cases, any engine speed edit field that remains empty will be counted as zero.

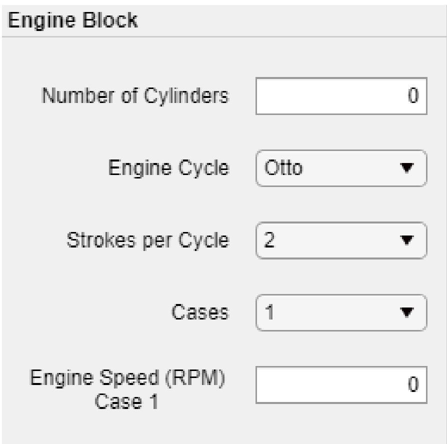


Figure 6-11 Engine Block Panel

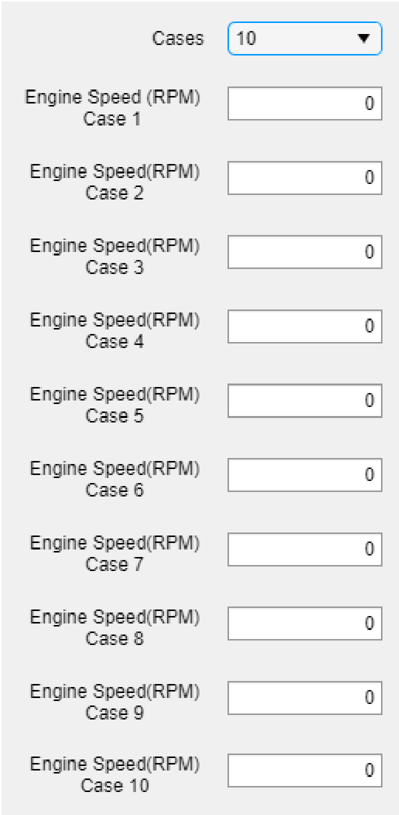


Figure 6-12 Different Cases and Engine Speed Inputs

6.3 Basic Interface Buttons

The main app has several menu buttons consists of new file, open file, save, save as, execute and results icons shown in figure 6-13. Additionally, user can select the desired operation in menu bar above the icons.

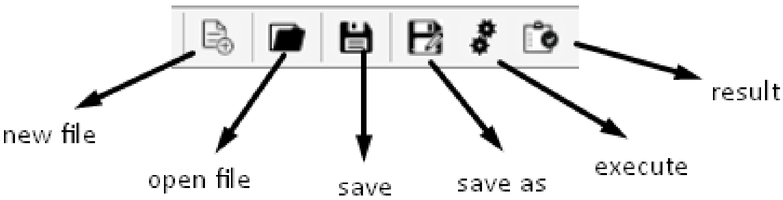


Figure 6-13 Interface Menu Icons

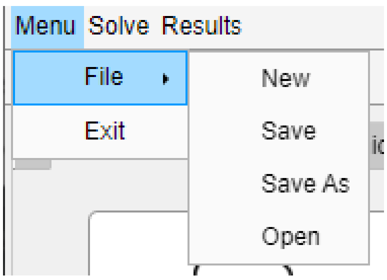


Figure 6-14 Interface Menu Bar

New file button is created with using a ‘Menu Selected Function’ which works as a push button. When user clicked to new file button, an empty version of the main app will be appeared.

Save and save as buttons are similar to each other except for one particular difference which is the option to save the file to a desired place rather than saving it to the file that was selected by the app. In this code, save button uses the folder name “a.mat” when user wants to save the file. Save and save as buttons only save the input values in a MATLAB file with the “.mat” extension. The reason for that is saving the whole app is not recommended since most of it consist of static elements.

```
if isequal(file,0)
    disp('User selected Cancel')
else
    disp(['User selected ', fullfile(path,file)]);
save(file,'temp','pres','fuel','vis','userdef','b','st','rod','comprat','AF','model','evalveopen','evalveclose','ivalveopen','ivalveclose',
    'cases','speed1','speed2','speed3','speed4','speed5','speed6','speed7','speed8','speed9','speed10','cylnum','cycle','spc');
end
```

Figure 6-15 Save As Button Codes

Open file gives the option to user to open a saved file and continue the work. Open file is restricted to open files that were saved in the main app only. Therefore, users should enter their input values and save them first to be able to use the open file button.

```
[file,path] = uigetfile('*.mat');
if isequal(file,0)
    disp('User selected Cancel');
else
    disp(['User selected ', fullfile(path,file)]);
    load(file,'temp','pres','fuel','vis','userdef','b','st','rod','comprat','AF','model','evalveopen','evalveclose','ivalveopen','ivalveclose',
        'cases','speed1','speed2','speed3','speed4','speed5','speed6','speed7','speed8','speed9','speed10','cylnum','cycle','spc');
end
```

Figure 6-16 Open File Codes

Solve and results buttons were the main focus of the app since they had most of the code. A detailed information about solve and result button codes will be given later.

Solve button works with a “Menu Selected Function” which contains the main calculation part of the code. After all the inputs are given, the system can be executed via using solve button. When the solve button is used, a message box with a “System Solved” message written on it will appear on the screen.

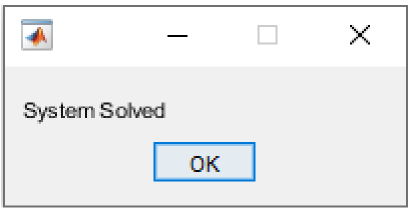


Figure 6-17 Solve Button Message Box

After solving the system, users should use results button to see the results and graphs such as temperature, pressure, brake power and torque etc. When the result button is used, a new app will be appeared and all data from solve button will be transferred to this new app. Then, this data will be used to draw time and swept plots. Time plots contains several graphs such as temperature, pressure and heat release and they were categorised via case number. Any plot that is out of range of the current case number will be appeared as zero line. On the other hand, swept plots category consists of five different subcategories: breathing, performance, pumping, fueling, and friction. Plots under these subcategories are drawn with using the current case number and the related data. At last, there is another button called "Summary Data Tables" that can give all the other results that are not used in graphs in a table based on the current case number.

Exit button is created with using the "closereq" function which works as a "Close Request Function". When it is used, function will immediately close the app and delete the inputs that the current app window contains.

```
% Menu selected function: ExitMenu
function ExitMenuSelected(app, event)
    closereq;
end
```

Figure 6-18 Exit Button Codes

6.4 The Main Part

The majority of the main code was simplified version of the article named "A User-Friendly, Two-Zone Heat Release Model for Predicting Spark-Ignition Engine Performance and Emissions" by Jeremy L. Cuddihy.[1] The main part consists of three parts; engine geometry calculations, pre-allocation matrices and a "for" loop to calculate program outputs as a function of crank angle. This part of the code was started by taking the user entered inputs from the main interface. However not all of the inputs were given by the user. Inputs like combustion duration, crank angle at start of combustion, engine load, number of revolutions per power cycle, and volumetric efficiency correction factor were part of the main code.

```
app.P0 = app.PressurekPaEditField.Value;
app.T0 = app.TemperatureKEditField.Value;
app.B = app.BoremmEditField.Value;
app.L = app.StrokemmEditField.Value;
app.l = app.ConnectingRodLengthEditField.Value;
app.C_r = app.CompressionRatioEditField.Value;
app.RPM1 = app.EngineSpeedRPMCase1EditField.Value;
app.RPM2 = app.EngineSpeedRPMCase2EditField.Value;
app.RPM3 = app.EngineSpeedRPMCase3EditField.Value;
app.RPM4 = app.EngineSpeedRPMCase4EditField.Value;
app.RPM5 = app.EngineSpeedRPMCase5EditField.Value;
app.RPM6 = app.EngineSpeedRPMCase6EditField.Value;
app.RPM7 = app.EngineSpeedRPMCase7EditField.Value;
app.RPM8 = app.EngineSpeedRPMCase8EditField.Value;
app.RPM9 = app.EngineSpeedRPMCase9EditField.Value;
app.RPM10 = app.EngineSpeedRPMCase10EditField.Value;
app.N_cyl = app.NumberofCylindersEditField.Value;
app.AF_ratio_stoich = app.AirFuelRatioEditField.Value;
app.EVO = app.ExhaustValveOpendegEditField.Value;
EVC = app.ExhaustValveClosedegEditField.Value;
IVO = app.IntakeValveOpendegEditField.Value;
IVC = app.IntakeValveClosedegEditField.Value;
```

Figure 6-19 Code for the User Inputs

Engine geometry calculations were performed based on the user defined inputs. Cylinder head surface area was assumed to be two times of the cross sectional piston area due to the complexity of the cylinder head geometry.

```
%_______________________________________________________________________
%Engine Calculations Based On Previous Inputs
%Assumes Average Surface Area In Which Heat Transfer Occurs
A_p = (pi/4)*app.B^2; %Cross Sectional Piston Area [m^2]
A_ch = 2*A_p; %Cylinder Head Surface Area (in chamber)
V_d = app.N_cyl*A_p*app.L; %Displaced Volume Of Engine [m^3]
Nrev1=   app.RPM1/60; Nrev2=   app.RPM2/60; Nrev3=   app.RPM3/60; Nrev4=   app.RPM4/60; Nrev5=   app.RPM5/60;
Nrev6=   app.RPM6/60; Nrev7=   app.RPM7/60; Nrev8=   app.RPM8/60; Nrev9=   app.RPM9/60; Nrev10=   app.RPM10/60; %[rev/s]
Vmean1 = 2*  app.L*Nrev1; Vmean2 = 2*  app.L*Nrev2; Vmean3 = 2*  app.L*Nrev3; Vmean4 = 2*  app.L*Nrev4; Vmean5 = 2*  app.L*Nrev5;
Vmean6 = 2*  app.L*Nrev6; Vmean7 = 2*  app.L*Nrev7; Vmean8 = 2*  app.L*Nrev8; Vmean9 = 2*  app.L*Nrev9; Vmean10 = 2*  app.L*Nrev10;
a = app.L/2; %Calculates Crank Radius (1/2 L)[m]
V_TDC = (V_d/(app.C_r-1))/app.N_cyl; %Calculates Clearance Volume [m^3]
V_BDC = (V_d/app.N_cyl)+V_TDC; %Cyl. Volume At BDC [m^3]
R_ratio=  app.l/a; % Ratio of Connecting Rod Length to Crank Offset
```

Figure 6-20 Engine Geometry Calculations

After engine geometry calculations friction mean effective pressure (fmep) was calculated before pre-allocate array and matrices with using equations (25) and (26) from chapter 5.3. The reason for that was calculating fmep only relates to the displacement volume, stroke, and engine speed (RPM). Because of the minimum values of equations (25) and (26), fmep values were conditioned to be zero if they were smaller than or equal to 100 and 150, respectively.

```
fmep1=(100000+350*app.L*app.RPM1)*10^(-3);
fmep6=(100000+350*app.L*app.RPM6)*10^(-3);
    if fmep1 <= 100
        fmep1 = 0;
    end
```

Figure 6-21 fmep Sample Code

Pre-allocate arrays and matrices were placed after fmep calculations. In MATLAB, pre-allocating arrays and matrices were greatly increasing the efficiency of the program by decreasing the computation time. With pre-allocate arrays and matrices, the initial values of some of the program outputs and key variables such as Reynolds number and Nusselt number were also calculated.

```
%Initial Preallocation Of Matrices
app.V(1:360)=zeros;DV(1:360)=zeros;app.X(1:360)=zeros;DX(1:360)=zeros;
m_b(1:360)=zeros;m_u(1:360)=zeros;A_u(1:360)=zeros;A_b(1:360)=zeros;
```

Figure 6-22 Pre-allocation of Matrices

Fuel inputs & efficiencies and atmospheric inputs were given below. Properties like excess air coefficient, maximum combustion efficiency and cylinder wall temperature were assumed. The purpose for this was to simplify the code.

```
%_______________________________________________________________________________
%Fuel Inputs/Efficiencies
AF_ratio_mol_sotich=14.7; %Molar Air_Fuel Ratio (Stoich)
lambda = .90; %Excess Air Coefficient
AF_ratio_ac = lambda*app.AF_ratio_stoich; %Actual Air Fuel Ratio
AF_ratio_mol=lambda*AF_ratio_mol_sotich;
eta_combmax = .95; %Assumed Maximum Combustion Efficiency
%Predicts Combustion Efficiency (Reference To Blair)
eta_comb=eta_combmax*(-1.6082+4.6509*lambda-2.0764*lambda^2);
%Atmospheric Inputs
P_atm = app.P0*1000;
T_atm = app.T0;
P_BDC = Load*P_atm; %Inlet Pressure[Pa]
R_air = 287; %Gas Constant For Air [J/kg-K]
T_w =350; %Assumed Wall Temperature (Reference Stone)
```

Figure 6-23 Fuel Inputs & Efficiencies and Atmospheric Inputs

After engine geometry calculations and pre-allocating of matrices and arrays, “for” loop of the code was rearranged to be able to calculate program outputs up to ten each. However, different cases of the same outputs were written in static form instead of dynamic form. The reason for that will be explained later.

“For” loop of the main part of the code had a different part than the calculations of outputs which was the calculating an updated version of the gamma by using a method called “Polynomial Method”. The main objective for using this method was to update the gamma value while the temperature and the pressure values were changing. With this, gamma value will be calculated simultaneously with the change of the crank angle value so that the code would gave more accurate results.

```
%Calculates Actual Gamma Value And Returns To Beginning Of Code
gamma_u1(i)=1+R_u1/cv_u1(i);
gamma1(i) = 1 + R1/cv1(i);
```

Figure 6-24 Updating the Gamma Value

Some assumptions were made in this “for” loop to be able to get better results and graphs. For instance, the first gamma value was taken as 1.4 which is the value of a standard day. However, gamma value depends on the state of the gas, it could change when user give another inlet temperature value, normally. Another assumption was the burning of the fuel that was entered the cylinder completely. In real engines, there were always some unburned fuel in the exhaust residual. Calculating that unburned fuel percentage in the exhaust residual was required advance knowledge, it assumed that fuel will be burned completely. Also the ignition delay value was taken as four degree crank angles. In many different sources and books, the ignition delay was stated that it changes engine to engine and defined as a few degree crank angle. Lastly, the corrected value of the inlet temperature was calculated with using residual gas fraction in the combustion chamber but it was assumed to be equal to the user defined inlet temperature for the simpler calculations.

```
if app.theta(i)>theta_0+4

T_b1(i)=  app.P1(i)*V_b1(i)/(m_b(i)*R*1000);
```

Figure 6-25 Ignition Delay as a Value of Four Degree Crank Angle

6.5 Program Outputs

Program outputs were calculated after the "for" loop was ended. Performance terms like brake and indicated mean effective pressure, brake and indicated power, as well as the efficiencies such as mechanical, thermal and fuel conversion efficiencies were calculated in this part. Also, output arrays were created after the calculations were done. These arrays and their role in the program will be explained later.

```
app.bmeps = [bmep1 bmep2 bmep3 bmep4 bmep5 bmep6 bmep7 bmep8 bmep9 bmep10];

app.imeps = [imep1 imep2 imep3 imep4 imep5 imep6 imep7 imep8 imep9 imep10];

app.bsfc = [BSFC1 BSFC2 BSFC3 BSFC4 BSFC5 BSFC6 BSFC7 BSFC8 BSFC9 BSFC10];

app.fmeps = [fmep1 fmep2 fmep3 fmep4 fmep5 fmep6 fmep7 fmep8 fmep9 fmep10];
```

Figure 6-26 Output Arrays

6.6 Transferring the Results to the Result App

Transferring the results to another app were started by building a second app. The objective was to call the second app in the first app by using a "Menu Selected Function" which works as a push button and transferring all the outputs to the second app to create plots and a summary table. The first step was to assign all the outputs as a "property" in the main app. With this, that data could be used in different parts of the first app and even in another apps. There were two different property type in MATLAB App Designer: private and public. Private properties were used to transfer the output data to another functions and parts of the main app. After that, public properties were given to those private properties one by one and that made the outputs transferable through from the main app to the results app. Those public properties were also placed in a public function.

```
methods (Access = public)

    function result(app)
        app.theta = deg;
        app.X = mburned; app.cases_value = rpmcase; app.T_ex = extemp; app.RPMs = speeds; app.T_brake = btorque;
        app.T_ind = itorque; app.Wdot_brake = bpower; app.Wdot_indicated = ipower; app.bmeps = brakemean;
        app.imeps = indmean; app.bsfc = brakespec; app.fmeps = fmean; app.T_f = ftorque; app.airflow = aflow;
        app.fuelflow = fflow; app.networks = worknet; app.F_c = fconeff; app.Meff = mechaeff; app.Teff = teff;
        app.Tburned = burnedtemp; app.Tunburned = unburnedtemp; app.EVO = evopen;
```

Figure 6-27 Public Properties

```
properties (Access = private)
    T_ex; RPMs; T_brake; T_ind; Wdot_brake; Wdot_indicated; bmeps;
    imeps; bsfc; fmeps; T_f; Brakepower; airflow; fuelflow; networks;
    F_c; Meff; Teff; Tburned; Tunburned;
```

Figure 6-28 Private Properties

After rearranging outputs as public properties, another push button was created and all the properties were assigned to that push button. To be able to open a second app, a function called DialogApp" were used with these properties.

```
% Clicked callback: Results
function ResultsClicked(app, event)
    app.DialogApp = Results_Window(app,app.theta,app.cases_value,app.X,app.T_ex,app.RPMs, ...
    app.T_brake,app.T_ind,app.Wdot_brake,app.Wdot_indicated,app.bmeps, ...
    app.imeps,app.bsfc,app.fmeps,app.T_f,app.airflow,app.fuelflow, ...
    app.networks,app.F_c,app.Meff,app.Teff,app.Tburned,app.Tunburned,app.EVO, ...
    ...
```

Figure 6-29 Calling the Result App

Transferring outputs were continued by taking the outputs and assigning them as properties in result app. The first step was to add the data that were transferred from the main app to the result app as input arguments. To be able to transfer the properties properly, that step was a must.

```
% Code that executes after component creation
function startupFcn(app, Bitirme_deneme, deg, rpmcase, mburned, extemp, speeds, btorque, itorque,
```

Figure 6-30 Adding the Properties as Input Arguments in Result App

Then, assigning the input arguments as private properties was the last step. Taking input values under a function called "startupFnc" after assigning new property names for each of them were done. "startupFnc" is a private function that allow the app to read the properties or do certain operation at the start of the app. Therefore, reading the input arguments from the main app with a start-up function were the best choice.

```
% Code that executes after component creation
function startupFcn(app, Bitirme_deneme, deg, rpmcase, mburned, extemp, speeds, btorque, itorque, bpower, ipower,
    app.theta_result = deg; app.fmeaneff = fmean;
    app.massburned = mburned; app.cases = rpmcase; app.exhausttemp = extemp; app.enginespeed = speeds;
    app.braketorque = btorque; app.indtorque = itorque; app.brakepower = bpower; app.indpower = ipower;
    app.bmeaneff = brakemean; app.imeaneff = indmean; app.bspec = brakespec; app.frictiontorque = ftorque;
    app.af = aflow; app.ff = fflow; app.wnet = worknet; app.FC_eff = fconeff; app.ME = mechaeff; app.TE = teff;
    app.T_burned = burnedtemp; app.T_unburned = unburnedtemp; app.exopen = evopen;
```

Figure 6-31 Taking Input Values in Start-up Function

```
properties (Access = private)
    CallingApp;
    massburned; cases; exhausttemp; enginespeed; braketorque; indtorque; brakepower;
    indpower; bmeaneff; imeaneff; bspec; fmeaneff; frictiontorque;
    theta_result; af; ff; wnet; FC_eff; ME; TE; T_burned; T_unburned; exopen;
```

Figure 6-32 Assigning New Property Names to Input Arguments

6.7 Result App Interface

Result app had two different components: a tree and an axes. Tree objective is a component which allows to display hierarchical list of items. Tree objective of result app was divided into three main categories which are time plots, swept plots and summary data table. Time and swept plots had some sub-categories under them and they contained the graphs that were created from the outputs of the main app. On the other hand, summary data table worked as a push button and when used, it created a table which shows the data that was separated by using cases.

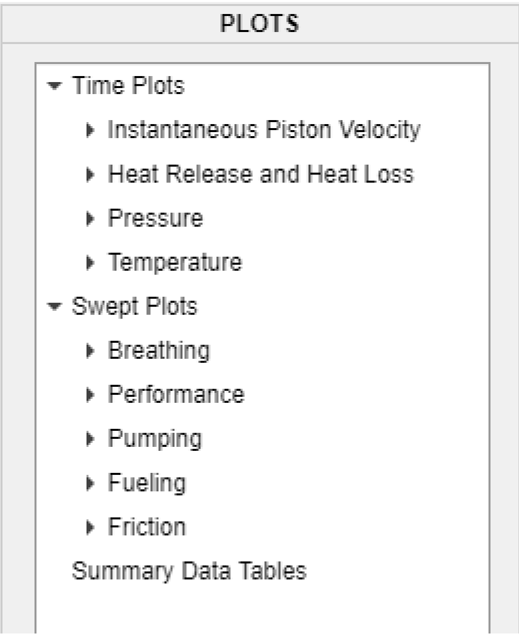


Figure 6-33 Tree Object of Result App

To be able to draw the graphs properly, a function called “strcmp” was used. This function was comparing the words of the tree objective categories or sub-categories and run the code if they are identical to each other. Also, all the graphs were projected to the axes component with their titles and legends. Axis limits were used for every graph to be able to give correct graphs while passing from one graph to another. This prevented the axes component from being stuck with the axis limits of the first graph.

```
% Selection changed function: Tree
function TreeSelectionChanged(app, event)
    selectedNode = app.Tree.SelectedNodes;
    if strcmp(selectedNode.Text,'Case 1-Instantaneous Piston Velocity')
        cla(app.UIAxes);
        plot(app.UIAxes,app.pistonv1);
        title(app.UIAxes,'Instantaneous Piston Velocity Vs. Crank Angle')
        xlabel(app.UIAxes,'Theta[deg]');
        ylabel(app.UIAxes,'Instantaneous Piston Velocity [m/s]');
        xlim(app.UIAxes,[0 360]);
        ylim(app.UIAxes,[app.pistonv1_min app.pistonv1_max]);
        legend(app.UIAxes,'Instantaneous Piston Velocity');
```

Figure 6-34 Plot Codes

Also, for the multiple graphs in one plot, “hold” function were used. This function had a similar functionality with the hold function in MATLAB which holds the data that was written before the function. With that, multiple graphs such as temperature and heat release graphs were written.

```
elseif strcmp(selectedNode.Text,'Case 1-Heat Release')
    cla(app.UIAxes);
    hold(app.UIAxes,'on');
    plot(app.UIAxes,app.h1);
    title(app.UIAxes,'Heat Release and Heat Loss')
    xlabel(app.UIAxes,'Theta[deg]');
    ylabel(app.UIAxes,'[kW]');
    xlim(app.UIAxes,[0 360]);
    plot(app.UIAxes,app.h_dot1);
    legend(app.UIAxes,'Heat Release','Heat Loss')
    ylim(app.UIAxes,[app.h1_min app.h1_max]);
```

Figure 6-35 Hold Function in Plots

Lastly, summary data table were created for users if they want to see the results as numbers. All of the important data was collected from the outputs and put them into a table. “uifigure” and “uitable” functions were used to be able to craft the table in a different window then result app. With using “ColumnEditable” function, all columns were made as non-editable. All of the data from the cases that were not given by user will be shown as “NaN”.

```
elseif strcmp(selectedNode.Text,'Summary Data Tables')
    fig = uifigure('Position',[500 500 760 360]);
    myData = {'Case Title' '-' 'Case 1' 'Case 2' 'Case 3' 'Case 4' 'Case 5' 'Case 6' 'Case 7' 'Case 8' 'Case 9' 'Case 10';
              'Engine Speed' 'rpm' app.enginespeed(1) app.enginespeed(2) app.enginespeed(3) app.enginespeed(4) app.enginespeed(5)
              'Case' '-' 1 2 3 4 5 6 7 8 9 10;
```

Figure 6-36 Summary Data Table Codes

6.8 Plots and Summary Data Table

Transferred data from the main app were used in time plots, swept plots, and summary data table. One plot from each category is given in Appendix-A which includes instantaneous piston velocity, heat release and heat loss, pressure, temperature for the time plots and breathing, performance, pumping, fueling, friction for the swept plots. Time plots were given at 10000 RPM and swept plots are given between 1000-10000 RPM increasing 1000 RPM for each case (ten cases total).Summary data table is given below.

	1	2	3	4	5	6	7	8	9	10	11	12
1	Case Title	-	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8	Case 9	Case 10
2	Engine Speed	rpm	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
3	Case	-	1	2	3	4	5	6	7	8	9	10
4	Mass Flow Rate	kg/s	0.0029	0.0059	0.0088	0.0117	0.0147	0.0176	0.0205	0.0234	0.0264	0.0293
5	Fuel Flow Rate	kg/s	2.1586e-04	4.3168e-04	6.4749e-04	8.6330e-04	0.0011	0.0013	0.0015	0.0017	0.0019	0.0022
6	IMEP	kPa	1.1391e+03	1.1991e+03	1.2284e+03	1.2470e+03	1.2606e+03	1.2714e+03	1.2797e+03	1.2866e+03	1.2925e+03	1.2975e+03
7	FMEP	kPa	168.7600	187.5200	206.2800	225.0400	243.8000	262.5600	281.3200	300.0800	318.8400	337.6000
8	BMEP	kPa	970.3266	1.0115e+03	1.0221e+03	1.0220e+03	1.0168e+03	1.0089e+03	998.4135	986.5491	973.6426	959.9395
9	Indicated Torque	N*m	25.1386	26.4622	27.1100	27.5205	27.8205	28.0591	28.2425	28.3947	28.5239	28.6355
10	Friction Torque	N*m	5.8658	6.3708	6.8082	7.2218	7.6244	8.0209	8.4119	8.7997	9.1852	9.5690
11	Brake Torque	N*m	19.2728	20.0914	20.3018	20.2986	20.1960	20.0382	19.8306	19.5950	19.3386	19.0665
12	Indicated Power	kW	2.6325	5.5422	8.5169	11.5277	14.5668	17.6300	20.7028	23.7879	26.8831	29.9870
13	Brake Power	kW	2.0182	4.2079	6.3780	8.5027	10.5746	12.5904	14.5366	16.4159	18.2262	19.9664
14	BSFC	g/kW*h	385.0374	369.3147	365.4711	365.5184	367.3680	370.2568	374.1271	378.6226	383.6383	389.1119
15	Net Work	J	315.9005	332.5339	340.6744	345.8325	349.6026	352.6009	354.9058	356.8182	358.4415	359.8439
16	Fuel Conversio...	-	0.2065	0.2153	0.2176	0.2175	0.2164	0.2147	0.2125	0.2100	0.2073	0.2043
17	Mechanical Effi...	-	0.8518	0.8436	0.8321	0.8195	0.8066	0.7935	0.7802	0.7668	0.7533	0.7398
18	Thermal Efficie...	-	0.2427	0.2530	0.2557	0.2556	0.2543	0.2524	0.2498	0.2468	0.2436	0.2401

Figure 6-37 Summary Date Table

7 CONCLUSION AND RECOMMENDATIONS

7.1 Thesis Summary

At the beginning of this report, a research about the history, working principles, reasons for the current popularity of the 3-cylinder diesel engines was made. In addition to that, this research was expanded with the information about the technologies that are being used in 3-cylinder diesel engines. In the past, 3-cylinder engines were not preferred because of the low power output and unbalanced design of the engine that causes vibration which leads rough driving sensation. However, with the new technologies such as cylinder deactivation that mentioned earlier, the new generation 3-cylinder engines entered the market more than ever.

After completing the literature survey about the 3-cylinder diesel engines, comparison stage began. In comparison stage, three 1-D engine performance softwares were chosen. The programmes consist of Diesel-RK, Lotus Engine Simulation and WAVE. After comparing these softwares, it concluded that all three programmes have advantages and disadvantages. While all programmes are capable of doing engine performance simulations and calculations, Ricardo WAVE was the most appropriate for the project. The reason for this choice is that the information and documents about Ricardo WAVE are abundant and easily accessable compared to Diesel-RK which has almost none. Although, Lotus Engine Simulation also has documents, Ricardo WAVE has a better help section and additionally, Ricardo WAVE has a lot of examples and tutorials within the programme files.

For the next part of the report, the sample engine selection was made by looking at the 3-cylinder diesel engines that contains turbocharger on their design in the market. To get a better look at the sample engine, two different simulation in WAVE software were made. One of the two designs was contained a turbocharger. The turbocharger has increased the engine performance and efficiencies perceptibly. Results of these two simulations show the importance of the turbocharger in the small diesel engines.

After research phase was done, design and coding parts of the app were started. The purpose of the thesis was to create an app which was capable of doing engine performans calculations and giving related plots of the outputs using a program. This program was chosen as MATLAB App Designer for the design and coding parts of the thesis. The part of the code that was in charge of the calculations were a simplified version of "A User-Friendly, Two-Zone Heat Release Model for Predicting Spark-Ignition Engine Performance and Emissions" thesis by Jeremy L. Cuddihy. Although the code was simplified, it was changed by adding different codes and it was updated to a state which it was capable of doing multiple input and case calculations. As stated above, user interfaces of main and result app were made based on Ricardo WAVE with the exception of drag and drop function which MATLAB App Designer was not capable of creating. Similar to Ricardo WAVE, the thesis app was also able to open a secondary app for the results window which gives performance plots and a summary data table.

7.2 Possible Improvements of the Model

Multiple input codes were written as static type which made the code longer than necessary. The static part could be rewriten as dynamic type. This would considerably shorten the code and reduce the calculation time of the app which give the app better performance. Also multiple input part was written with the assumption that would be maximum number of ten inputs given at any time. Once agian if this part were to be converted to a dynamic type code, any number of inputs could be given by users.

Calculation part of the code was written based on the closed system combustion model calculations. Therefore, there was only half of an engine power cycle visible in plots. If it was written based on the open system combusiton model calculations, a whole engine cycle could be seen in plots.

As mentioned above drag and drop function which Ricardo WAVE uses as a means to create different engine models was not implemented in the thesis. The reason for this was that drag and drop function was not within the capabilities of MATLAB App Designer. Therefore this function can be added to the code when MATLAB App Designer have such operation in the future.

Lastly, the calculation part of the code was only capable of doing single cylinder diesel engine calculations. Multi-cylinder engine calculations require firing order knowledge, distinctly from single cylinder engine calculations. Therefore multi-cylinder engine calculations was not included in the app. With sufficient knowledge multi-cylinder calculations can be added to the app.

8 REFERENCES

[1] Jeremy L. Cuddihy, 2014, A User-Friendly, Two-Zone Heat Release Model for Predicting Spark-Ignition Engine Performance and Emissions

[2] Pulkrabek, W.W., 1997, Engineering Fundamentals of the Internal Combustion Engine, 1st Edition, Pearson, p. 5

[3] Pulkrabek, W.W., 1997, Engineering Fundamentals of the Internal Combustion Engine, 1st Edition, Pearson, p. 303

[4] Ford Media Center, Ford to offer fuel-saving cylinder deactivation tech for 1.0 litre ecoboost; global first for a 3-cylinder engine

[5] Khan, S., Khan, B.Y., Khan, H., Khan, A., 2014, Valvetronic Engine Technology, IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE)

[6] BMWBlog, What Is Valvetronic? BMW’s Innovative Throttle System

[7] Pulkrabek, W.W., 1997, Engineering Fundamentals of the Internal Combustion Engine, 1st Edition, Pearson, p. 272

[8] Pulkrabek, W.W., 1997, Engineering Fundamentals of the Internal Combustion Engine, 1st Edition, Pearson, p. 272

[9] Diesel-RK, What is a Diesel-RK

[10] Lotus Engineering, Getting Started Using Lotus Engine Simulation

[11] Ricardo, Value Through Virtual Engineering

9 APPENDIX-A

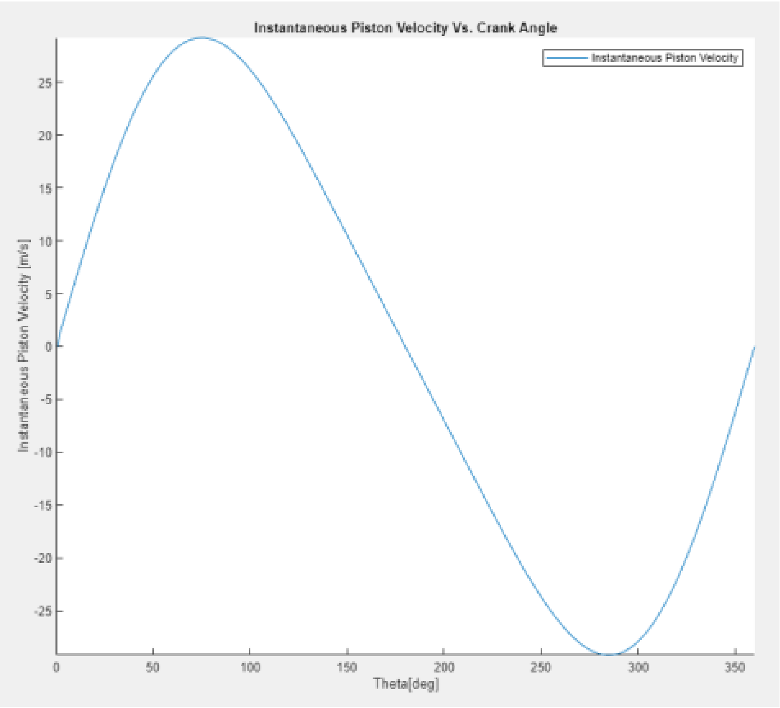


Figure 9-1 Instantaneous Piston Velocity vs. Crank Angle

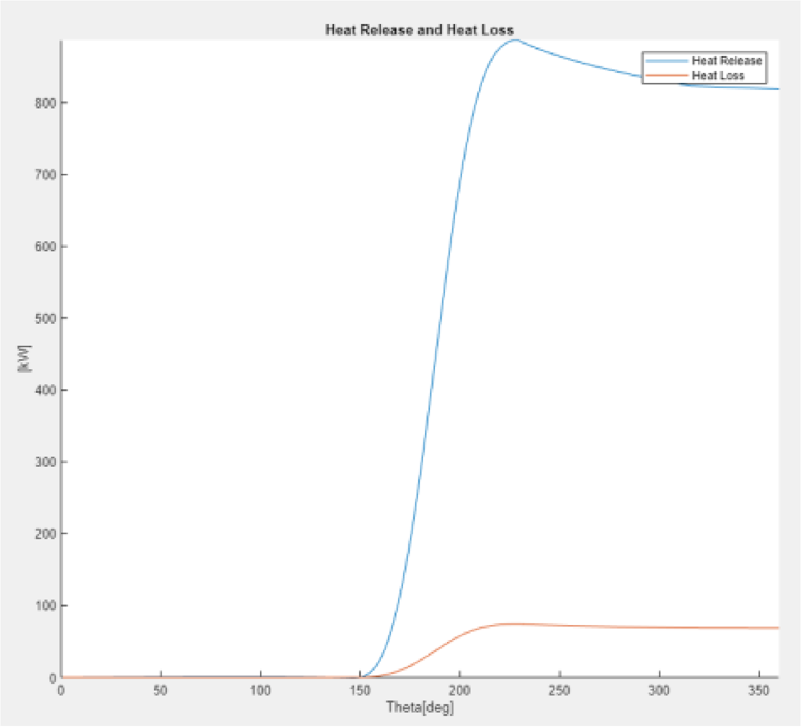


Figure 9-2 Heat Release and Heat Loss vs. Crank Angle

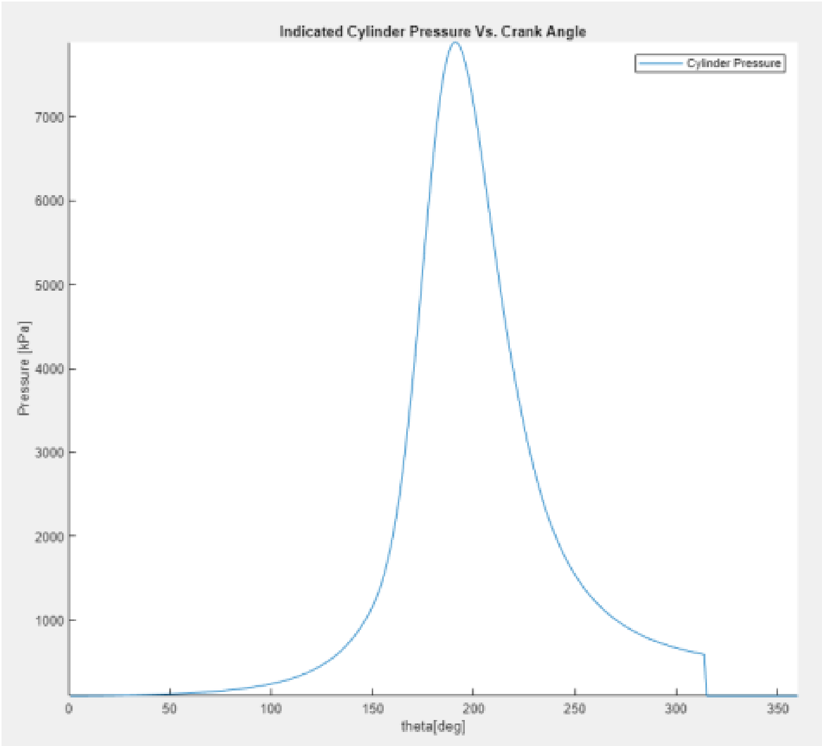


Figure 9-3 Indicated Cylinder Pressure vs. Crank Angle

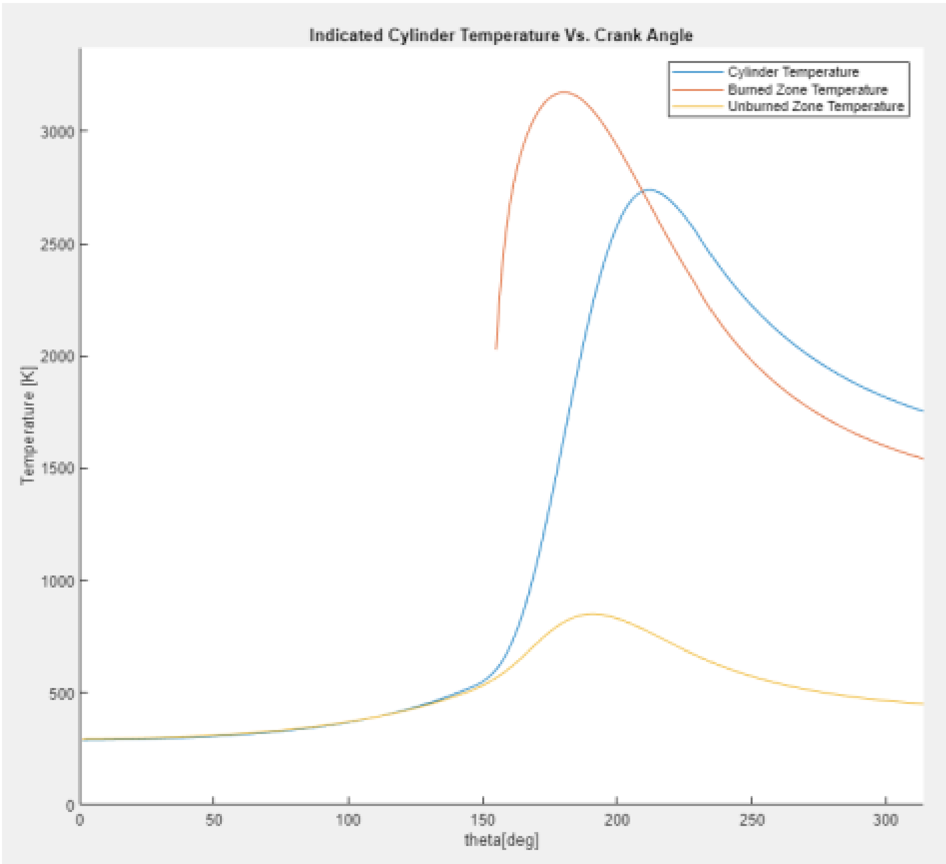


Figure 9-4 Indicated Cylinder Temperature vs. Crank Angle

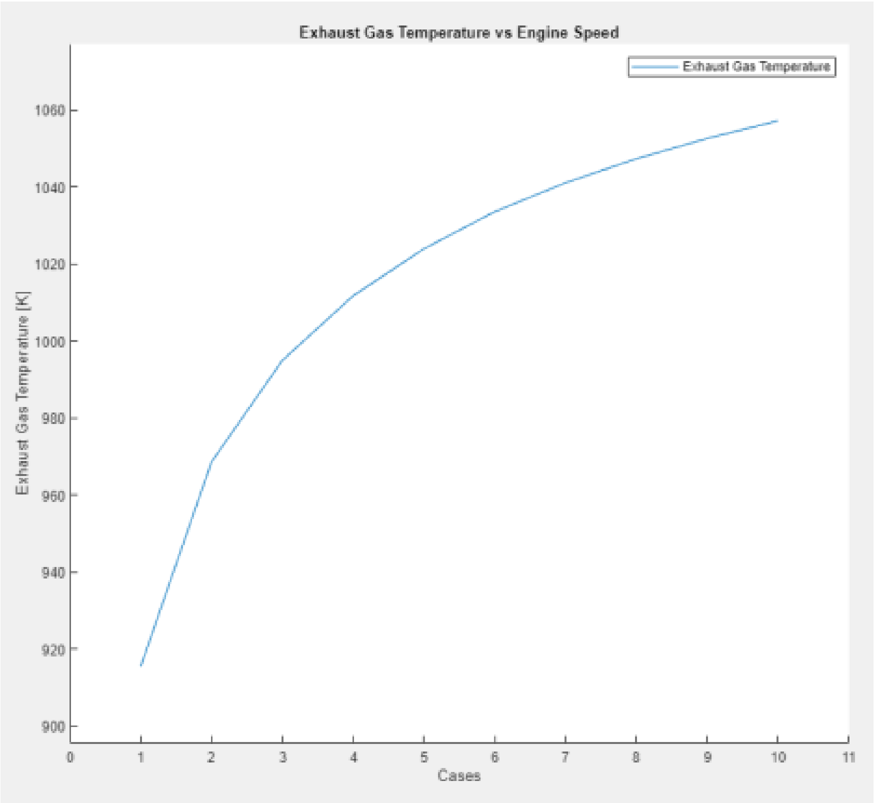


Figure 9-5 Exhaust Gas Temperature vs. Engine Speed

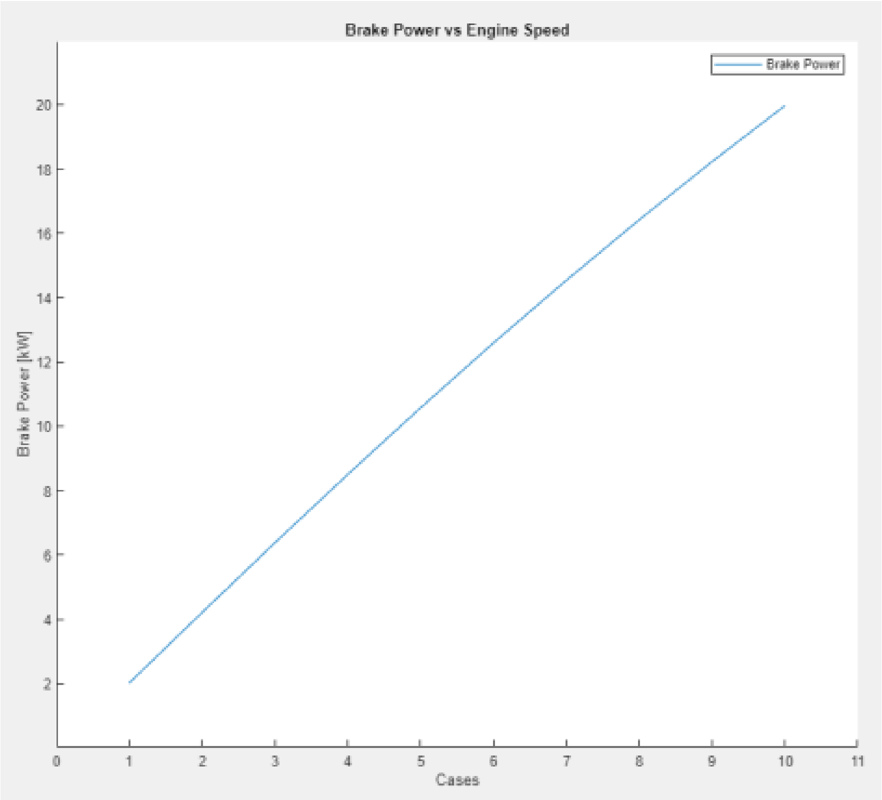


Figure 9-6 Brake Power vs. Engine Speed

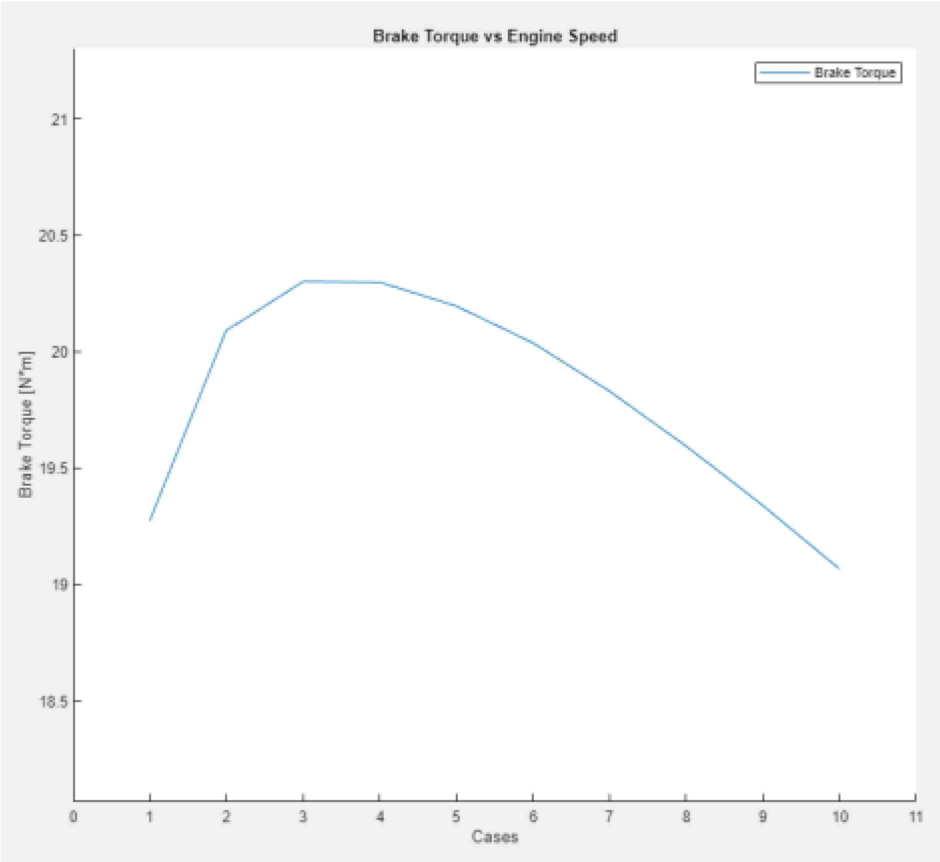


Figure 9-7 Brake Torque vs. Engine Speed

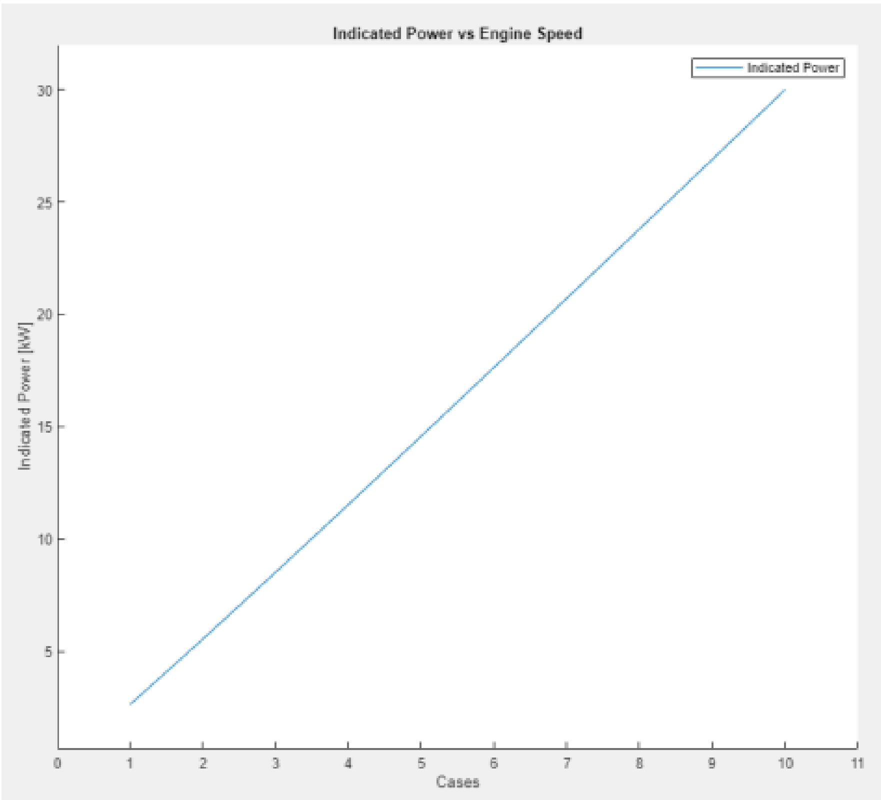


Figure 9-8 Indicated Power vs. Engine Speed

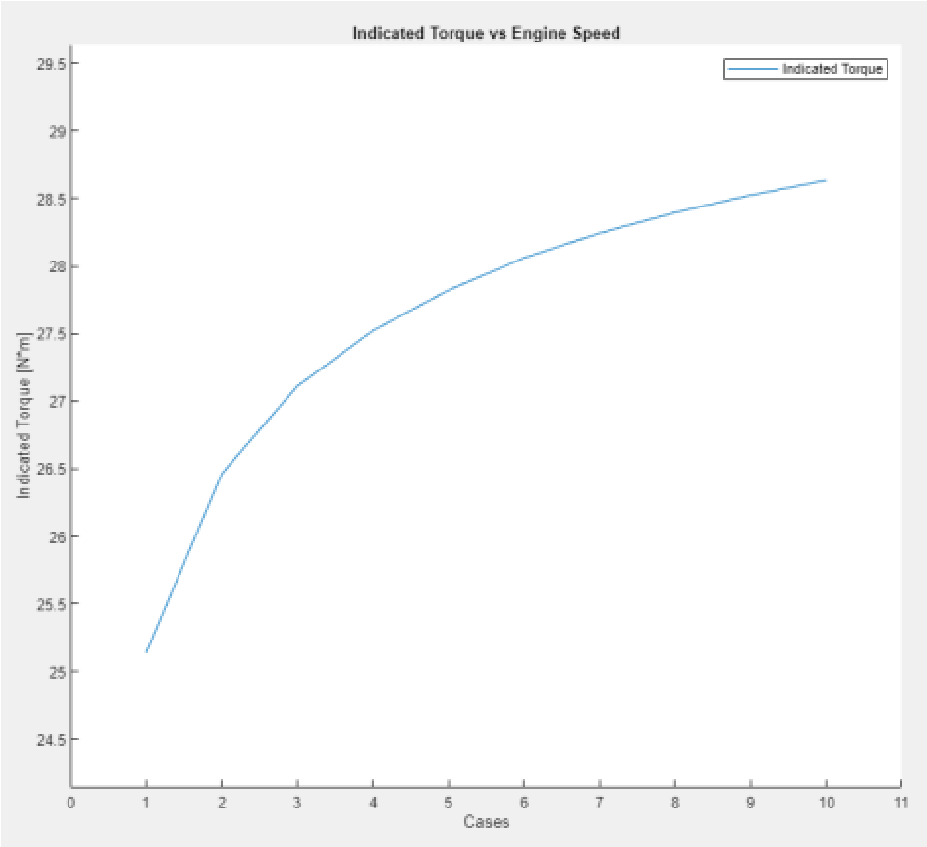


Figure 9-9 Indicated Torque vs. Engine Speed

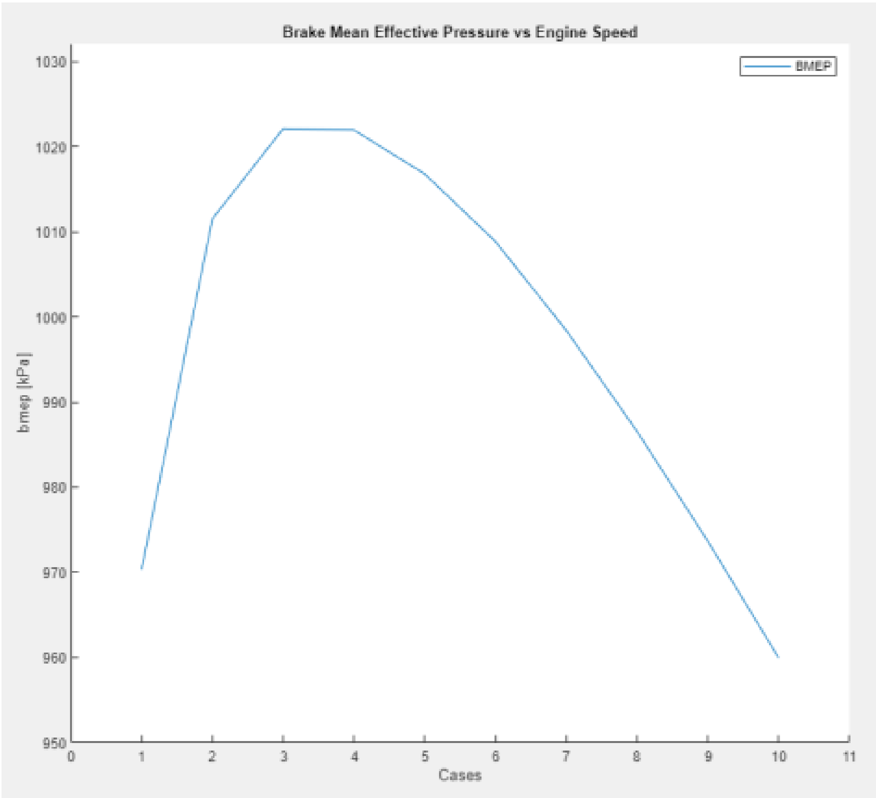


Figure 9-10 Break Mean Effective Pressure vs. Engine Speed

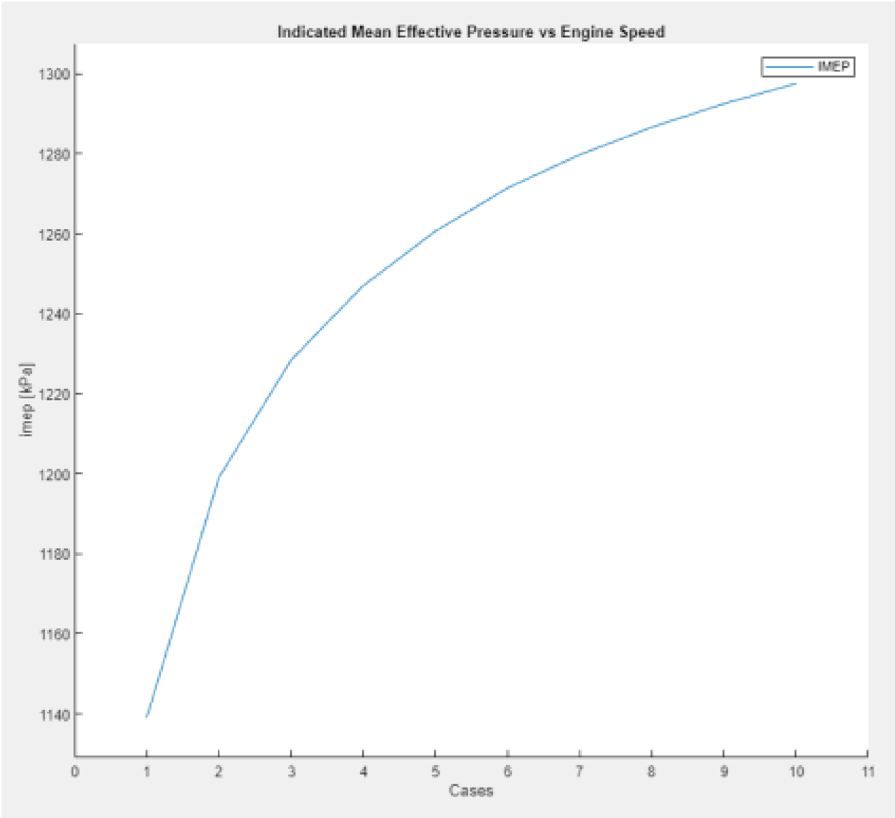


Figure 9-11 Indicated Mean Effective Pressure vs. Engine Speed

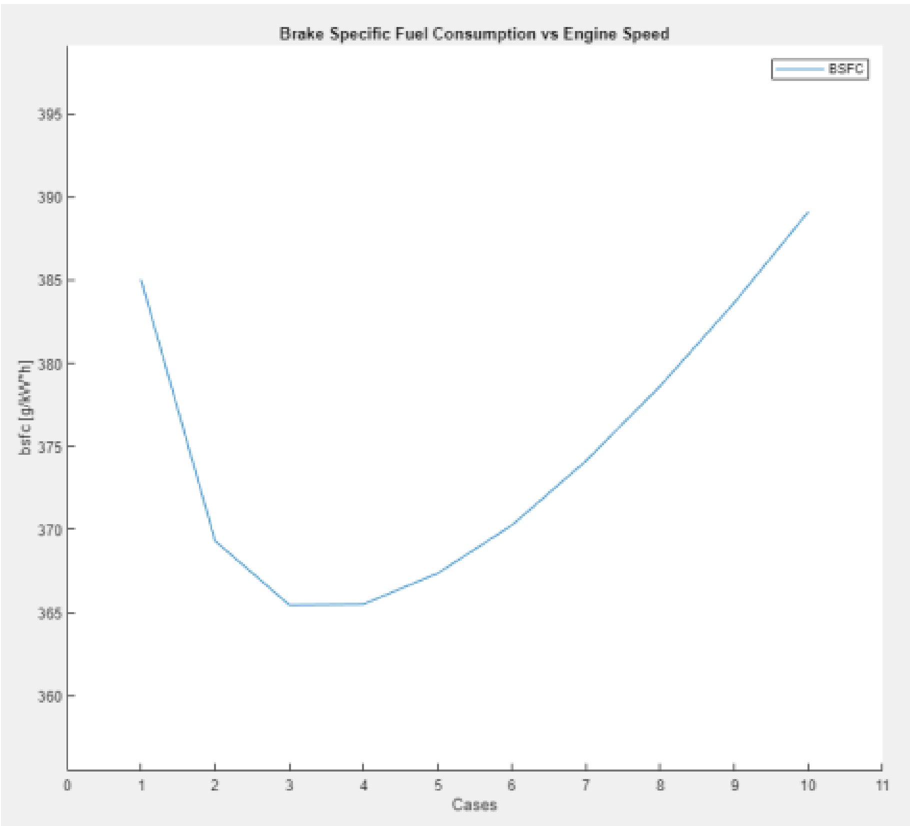


Figure 9-12 Brake Specific Fuel Consumption vs. Engine Speed

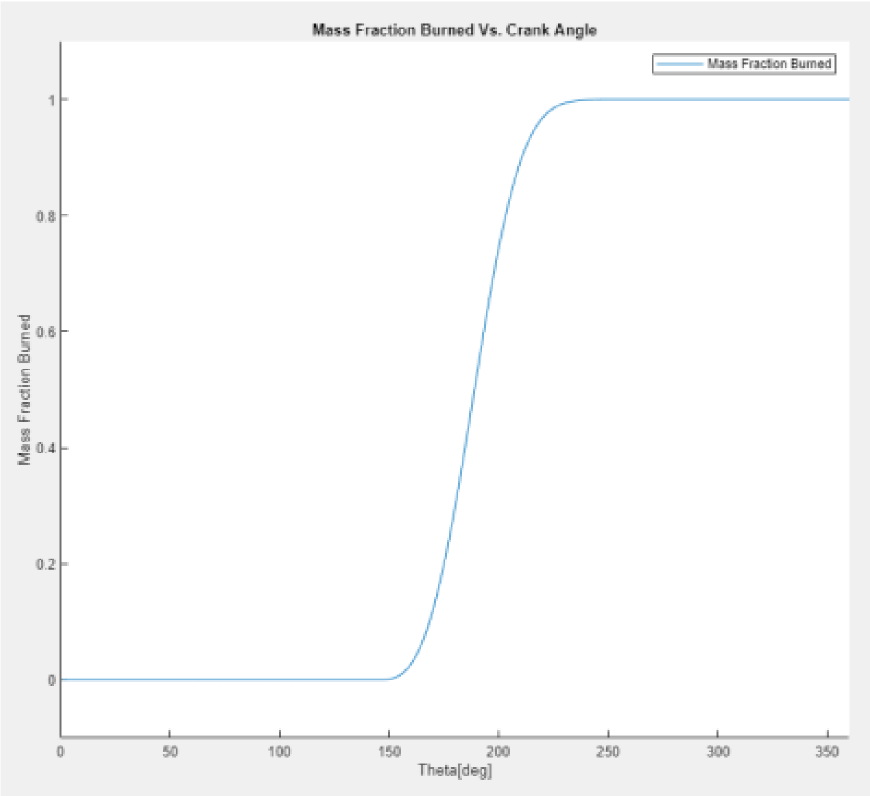


Figure 9-13 Mass Fraction Burned vs. Crank Angle

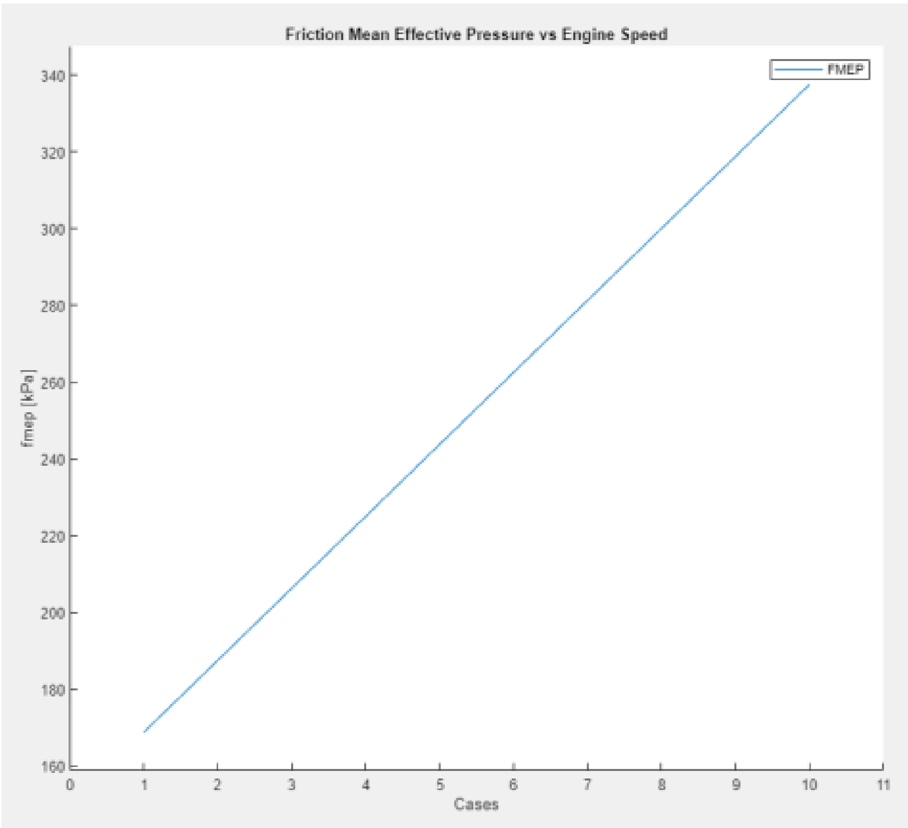


Figure 9-14 Friction Mean effective Pressure vs. Engine Speed

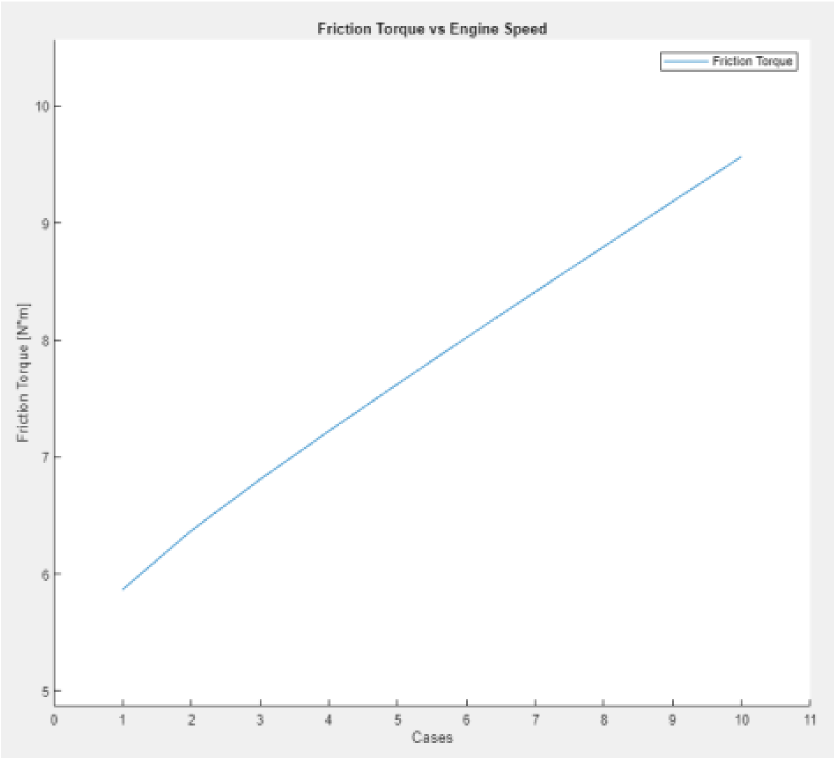


Figure 9-15 Friction Torque vs. Engine Speed