A New Internal Combustion Engine Configuration - Opposed Piston with Crank Offset

A Dissertation submitted by

Mr Ray Malpress

in fulfilment of the requirements of

Bachelor of Engineering (Mechanical)

November, 2007
Abstract

The performance of a new engine configuration is assessed. The engine type is unique – details of similar engines have not been found in the open literature. The primary goal of this new engine design is to improve engine efficiency. It consists of two opposed pistons in a single cylinder controlled by two synchronously timed crankshafts at opposite ends of the cylinder. It makes use of crank offset to create the required piston motion aimed at engine efficiency improvements through thermodynamic performance gains. In particular, the engine employs full expansion. It also features a greater rate of volume change after combustion than a conventional 4-stroke engine for the same crank speed. The engine is a piston ported, spark ignition petrol engine.

Thermodynamic and friction modelling using Matlab predicted net efficiencies in the order of 38%. Solid modelling and Finite Element Analysis were employed to build a prototype engine. Several facets of the engine build process resulted in the prototype differing from the design specifications. The original Matlab model was used to recalculate the predicted engine performance based on the prototype specifications. Deficiencies and errors in the original Matlab model were revealed by testing of the prototype and the data obtained allowed the original Matlab model to be reviewed. Modelling parameters used in the Matlab model were subsequently re-evaluated allowing the Matlab model to be adjusted to reflect the performance of the prototype.

Some constructive results were obtained with regard to the performance of the Matlab model. To date the engine has not been able to ‘power’ itself. The engine has overwhelmingly high friction relative to the original Matlab model predictions. When motored, the engine can maintain a consistent burn and the thermodynamic cycle delivers about half the originally predicted torque, which ‘unloads’ the powering motor. This torque represents about 40% of the torque required to motor the engine as measured in the prototype testing.
Future work could address the thermodynamic deficiencies and reduce friction, but the engine as designed and constructed shows little potential as a viable engine. However the Matlab model and thermodynamic cycle have positive attributes as quantified by the engine test. Minor changes to the model structure and appropriate specification of parameters determined from the prototype test allowed the model to accurately reflect the performance of the prototype. The main feature of the thermodynamic cycle, full expansion was confirmed by the model to produce the extra work predicted relative to a conventional cycle, thereby allowing for improved efficiency if that work was obtained without excessive addition friction losses. The Matlab model and thermodynamic cycle may have future applications.
Acknowledgements

Assistance in the completion of this project is thankfully acknowledged.

Mr Charlie Weeding of DC Racing for the provision of specialist engine machining and technical advice.

Mr Ian Gebbett of Condamine drilling for the use of workshop machines.

Mr. Brian Aston and Mr. Chris Galligan of the USQ mechanical workshop for the fabrication of the engine crankshafts.

Assoc. Prof. David Buttsworth for enthusiastic support and encouragement.
University of Southern Queensland
Faculty of Engineering and Surveying

ENG4111 Research Project Part 1 & ENG4112 Research Project Part 2

Limitations of Use

The Council of the University of Southern Queensland, its Faculty of Engineering and Surveying, and the staff of the University of Southern Queensland, do not accept any responsibility for the truth, accuracy or completeness of material contained within or associated with this dissertation.

Persons using all or any part of this material do so at their own risk, and not at the risk of the Council of the University of Southern Queensland, its Faculty of Engineering and Surveying or the staff of the University of Southern Queensland.

This dissertation reports an educational exercise and has no purpose or validity beyond this exercise. The sole purpose of the course "Project and Dissertation" is to contribute to the overall education within the student’s chosen degree programme. This document, the associated hardware, software, drawings, and other material set out in the associated appendices should not be used for any other purpose: if they are so used, it is entirely at the risk of the user.

Professor Frank Bullen
Dean
Faculty of Engineering and Surveying
Certification

I certify that the ideas, designs and experimental work, results, analysis and conclusions set out in this dissertation are entirely my own effort, except where otherwise indicated and acknowledged.

I also certify that the work is original and has not been previously submitted for assessment in any other course or institution, except where specifically stated.

Ray Malpress

Student Number: 0038170152

[Signature]

[25 October 2007]

Date
# Table of Contents

**ABSTRACT**

**ACKNOWLEDGEMENTS**

**CERTIFICATION**

**LIST OF FIGURES**

**LIST OF TABLES**

**LIST OF APPENDICES**

**LIST OF ACRONYMS AND ABBREVIATIONS**

**CHAPTER 1 THE INVESTIGATION DEFINED**

1.1 Overview

1.2 Introduction

1.2.1 Project Motivation

1.3 Research Objectives

1.4 Model Refinement

1.5 Conclusions

**CHAPTER 2 LITERATURE AND RESOURCES REVIEW**

2.1 Introduction

2.2 Background

2.2.1 The Engine Concept

2.2.2 Opposed Pistons

2.2.3 Crank Offset

2.2.4 Full Expansion

2.2.5 100% Scavenging

2.3 Applications of Thermodynamics to I.C.E.
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.4 I.C.E. Thermodynamic Modelling</td>
<td>15</td>
</tr>
<tr>
<td>2.5 Friction Analysis</td>
<td>16</td>
</tr>
<tr>
<td>2.6 Computational modelling techniques and applications</td>
<td>16</td>
</tr>
<tr>
<td>2.7 Solid Modelling and FEA</td>
<td>17</td>
</tr>
<tr>
<td>2.8 Conclusion</td>
<td>17</td>
</tr>
<tr>
<td><strong>CHAPTER 3 PROJECT METHODOLOGY AND ENGINE DESIGN</strong></td>
<td>18</td>
</tr>
<tr>
<td>3.1 Introduction</td>
<td>18</td>
</tr>
<tr>
<td>3.2 Software Modelling</td>
<td>19</td>
</tr>
<tr>
<td>3.2.1 Overview</td>
<td>19</td>
</tr>
<tr>
<td>3.2.2 Thermodynamic Model</td>
<td>19</td>
</tr>
<tr>
<td>3.2.3 GUI Input</td>
<td>19</td>
</tr>
<tr>
<td>3.2.4 Features of the GUI output</td>
<td>24</td>
</tr>
<tr>
<td>3.2.5 Piston Position, Velocity and Acceleration</td>
<td>25</td>
</tr>
<tr>
<td>3.3 Friction and Losses Model</td>
<td>27</td>
</tr>
<tr>
<td>3.4 Optimisation</td>
<td>30</td>
</tr>
<tr>
<td>3.5 Matlab Model Plots</td>
<td>32</td>
</tr>
<tr>
<td>3.6 Solid Modelling in ProE</td>
<td>34</td>
</tr>
<tr>
<td>3.7 Engine Configuration and Features</td>
<td>35</td>
</tr>
<tr>
<td>3.8 Sizing and Material Selection</td>
<td>36</td>
</tr>
<tr>
<td>3.8.1 Crankcases</td>
<td>37</td>
</tr>
<tr>
<td>3.8.2 Cylinder (Barrel)</td>
<td>37</td>
</tr>
<tr>
<td>3.8.3 Crankshafts</td>
<td>37</td>
</tr>
<tr>
<td>3.8.4 Pistons</td>
<td>37</td>
</tr>
<tr>
<td>3.9 Interference and Function</td>
<td>39</td>
</tr>
<tr>
<td>3.10 Engine Concept Display</td>
<td>40</td>
</tr>
<tr>
<td>3.11 Drawings and specifications</td>
<td>40</td>
</tr>
<tr>
<td>3.12 ProE Models as a Basis for FEA</td>
<td>41</td>
</tr>
</tbody>
</table>
3.13 FEA Analysis 41
3.14 Engine Components 45
3.15 Prototype Build 45
3.16 Engine Component Features Specific to the Opposed Piston Configuration 47
  3.16.1 Piston Rings 47
  3.16.2 Inclined Crank Arm 48
  3.16.3 Gudgeon Centre to Piston Crown Height 48
  3.16.4 Cylinder-Crankshaft Interference 48
  3.16.5 Inner Main Crank Bearing 49
3.17 Ancillary Components 49
  3.17.1 Induction Reed Valve 50
3.18 Important Assembly Notes 51
3.19 Conclusion 52

CHAPTER 4 PROTOTYPE MANIPULATION 53
4.1 Introduction 53
4.2 Engine Starting Problems 53
  4.2.1 Cranking Speed 53
  4.2.2 Engine Preheat 54
  4.2.3 Engine Motoring 54
  4.2.4 Mixture Control 55
  4.2.5 Ignition Options 56
  4.2.6 Combustion Chamber Shape 56
  4.2.7 Compression Pressure Achieved in the Prototype 58
  4.2.8 Blow-by Compensation 59
  4.2.9 Improved Ignition 62
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1 Introduction</td>
<td>93</td>
</tr>
<tr>
<td>7.2 The Matlab Model Performance</td>
<td>93</td>
</tr>
<tr>
<td>7.3 Discussion</td>
<td>94</td>
</tr>
<tr>
<td>7.4 Potential Further Applications</td>
<td>96</td>
</tr>
<tr>
<td>7.5 Summary of Chapter 7 Conclusions</td>
<td>97</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>98</td>
</tr>
</tbody>
</table>
List of Figures

Figure 1.1 – p-v diagram of theoretical full expansion cycle ........................................ 2
Figure 3.1 – Matlab Gui Input .................................................................................. 20
Figure 3.2 – Matlab GUI displaying outputs ............................................................. 22
Figure 3.3 – GUI output ......................................................................................... 23
Figure 3.4 – Piston motion geometry ...................................................................... 26
Figure 3.5 – Forces Analysis for friction calculations .............................................. 28
Figure 3.6 – Striebeck diagram .............................................................................. 29
Figure 3.7 – Comparison of the original Matlab model prediction of engine efficiency .............................................................................................................. 31
Figure 3.8 – Plot of both connecting rod inclinations ............................................. 32
Figure 3.9 – Plot of lower piston friction for the original Matlab Model optimised engine configuration .......................................................... 33
Figure 3.10 – ProE solid model of Engine Assembly ................................................. 35
Figure 3.11 – Connection rod loads and orientation ................................................. 42
Figure 3.12 – Ansys contour stress plot of crankshaft ............................................. 43
Figure 3.13 – Ansys deformation plot for the piston ............................................... 44
Figure 3.14 – Displacement plot of cylinder under belt tension bending ................ 45
Figure 3.15 – Induction reed valve ......................................................................... 50
Figure 4.1 – Piston combustion chamber modifications .......................................... 58
Figure 4.2 – compression pressure gauge fitted during crankcase pump pressure tests ............................................................................................................... 59
Figure 4.3 – Crank case cover with induction reed valve ........................................ 61
Figure 4.4 – Ignition prongs .................................................................................... 62
Figure 5.1 – Torque measurement using a spring balance ........................................ 66
Figure 5.2 – Prototype in test configuration ............................................................... 66
Figure 5.3 – Pitot tube upstream of carburettor ....................................................... 68
Figure 5.4 – Inclined tube manometer ..................................................................... 68
Figure 5.5 – Fuel flow measured in vertical tube ...................................................... 68
Figure 5.6 – Digital multimeter (with pulse meter capability) and stop watch .......... 69
Figure 5.7 – Engine compression pressure gauge .................................................. 69
Figure 5.8 – Plot of motored compression pressures of Table 5-4 ............................ 75
Figure 5.9 – Plot of compression pressures – induction devices removed ............... 76
Figure 6.1 – Possible elevated blow-by mechanism........................................ 82
Figure 6.2 – b – Matlab model predicted engine pressure with modified blow-by constant .................................................................................................................................. 83
Figure 6.3 – b – Modified Matlab model including spark–plug port volume........... 85
Figure 6.4 – b – Revised Matlab model engine pressure prediction ...................... 87
List of Tables

Table 3-1 – Comparative properties of possible piston materials ......................... 38
Table 3-2 – Prototype specifications ................................................................. 46
Table 4-1 – Engine compression pressure with elevated induction pressure ........... 60
Table 4-2 – Engine compression pressure resulting from crankcase pump .......... 61
Table 5-1 – Motoring test data. Data from three tests performed over two days are
presented. ........................................................................................................ 70
Table 5-2 – Motoring tests – derived results. Results from three tests performed over
two days are presented. ................................................................................ 73
Table 5-3 – Comparison of simulation results and prototype test results .......... 74
Table 5-4 – Motored engine compression pressure without crankcase pump ....... 75
Table 5-5 – Compression pressures – induction devices removed ...................... 76
Table 5-6 – Component friction predicted by the original Matlab model ........... 78
Table 6-1 – Matlab model features reviewed .................................................. 80
Table 6-2 – Comparison of internal thermodynamic work and maximum pressure for
the original and revised Matlab models ......................................................... 87
List of Appendices

Appendix A – Project Specification ................................................................. 100
Appendix B – Matlab scripts, structure chart ................................................. 101
Appendix C – Matlab model plot scripts and representative plots .................. 185
Appendix D – Engine Components ................................................................. 195
Appendix E – Solid Model Images and Drawings of Engine Components ......... 198
Appendix F – Engine Component Photographs .............................................. 208
List of Acronyms and Abbreviations

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.C.R.</td>
<td>Actual Compression Ratio (includes spark plug port volume)</td>
</tr>
<tr>
<td>A.E.R.</td>
<td>Actual Expansion Ratio</td>
</tr>
<tr>
<td>ASM</td>
<td>American Society for Metals</td>
</tr>
<tr>
<td>ASPO</td>
<td>Association for the Study of Peak Oil</td>
</tr>
<tr>
<td>C.R. (CR)</td>
<td>Compression Ratio</td>
</tr>
<tr>
<td>C.R.Ign</td>
<td>Compression Ratio at ignition</td>
</tr>
<tr>
<td>CCR</td>
<td>Constant Compression Ratio</td>
</tr>
<tr>
<td>CEP</td>
<td>Constant Engine Pressure</td>
</tr>
<tr>
<td>Conrod</td>
<td>Connecting rod</td>
</tr>
<tr>
<td>E.V.D.R.</td>
<td>Effective Variable Displacement Ratio</td>
</tr>
<tr>
<td>eff</td>
<td>efficiency</td>
</tr>
<tr>
<td>ESP</td>
<td>Engine Simulation Program</td>
</tr>
<tr>
<td>FEA</td>
<td>Finite Element Analysis (Computer Application)</td>
</tr>
<tr>
<td>GHG</td>
<td>Green House Gas</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>I.C.E.</td>
<td>Internal Combustion Engine</td>
</tr>
<tr>
<td>int</td>
<td>internal</td>
</tr>
<tr>
<td>NO\textsubscript{X}</td>
<td>Nitrous Oxide compounds</td>
</tr>
<tr>
<td>ode</td>
<td>Ordinary Differential Equation</td>
</tr>
<tr>
<td>P.C.</td>
<td>Piston Clash</td>
</tr>
<tr>
<td>ProE</td>
<td>ProEngineer solid modelling application</td>
</tr>
<tr>
<td>SAE</td>
<td>Society of Automotive Engineers</td>
</tr>
<tr>
<td>WOT</td>
<td>Wide open throttle</td>
</tr>
</tbody>
</table>
Chapter 1 The Investigation Defined

1.1 Overview

The broad aim of the project was to assess the performance of a new engine configuration. This engine type is unique, entirely conceived of by the author and has not previously been reported in the open literature. It consists of two opposed pistons in a single cylinder controlled by two crankshafts at opposite ends of the cylinder. It makes use of crank offset to create the required piston motion aimed at engine efficiency improvements through thermodynamic performance gains. In particular, the engine employs full expansion, 100 percent exhaust scavenging with no residual gas and a higher rate of volume change in the power stroke compared to a conventional engine for the same engine speed. The engine is a piston ported, spark ignition, petrol engine.

The task entailed the application of existing modelling techniques and the creation of new techniques to conceive, assess, build and test a prototype engine for the express purpose of investigating the potential for improving the efficiency of internal combustion engines.

A particular goal of the project is to assess the viability of the engine design presented and its potential applications.

1.2 Introduction

This project is motivated by the desire to identify techniques to improve engine efficiency. Theoretical efficiency limits are far higher than those achieved by conventional engines so there should be reasonable prospects for engine efficiency improvements. Typically engine efficiency is quoted at wide open throttle (WOT)
where the engine is operating at peak pressures and consequently peak thermal efficiency. For petrol engines, the WOT efficiency achieved is typically in the low to mid 30% range but the theoretical thermal efficiency of the Otto cycle is well over 50% for a compression ratio tolerated by petrol engines. For example, the theoretical thermal efficiency of an Otto cycle at a compression ratio of 10:1 is 61% (Cengel and Boles [1]).

The expected thermodynamic cycle efficiency improvements for the new engine are based on changes in the cycle form relative to that employed in conventional engines. Figure 1.1 shows additions to the conventional Otto cycle that reflect a cycle that represents full expansion. The additional area under the curve represents the idealised increase in work per cycle. Some of that additional work is offset by added pumping losses. The project aims to identify the net advantage of the use of the cycle using full expansion.

![Figure 1.1 – p-v diagram of theoretical full expansion cycle.](image)

The following equations describe the theoretical advantage of full expansion.
From Figure 1.1

the expansion ratio, \( ER = \frac{v_4}{v_3} \) \hfill (1)

and the compression ratio, \( CR = \frac{v_1}{v_2} \) \hfill (2)

From first principles,

efficiency, \( \eta = 1 - \frac{Q_{\text{out}}}{Q_{\text{in}}} \quad (Q = \text{heat}) \) \hfill (3)

for \( m = \text{mass}, \)
\( c_p = \text{specific heat at constant pressure} \)
and \( c_v = \text{specific heat at constant volume} \)

\[ \eta = 1 - \frac{mc_p(T_4 - T_1)}{mc_v(T_3 - T_2)} \] \hfill (4)

From the ideal gas law,

\[ \frac{T_4}{T_1} = \frac{v_4}{v_1} \] \hfill (5)

giving

\[ T_4 = \left( \frac{ER}{CR} \right) \times T_1 \quad (v_3 = v_2) \] \hfill (6)

From the isentropic compression and expansion,

\[ T_3 = T_4 \times (ER)^{(\lambda-1)} \] \hfill (7)

\[ T_2 = T_1 \times (CR)^{(\lambda-1)} \] \hfill (8)

therefore,

\[ T_3 = \left( \frac{ER}{CR} \right) \times T \times (ER)^{(\lambda-1)} \] \hfill (9)
In comparison to the theoretical ideal Otto cycle efficiency of 61% for a compression ratio of 10:1, the full expansion theoretical cycle produces an efficiency of 69.5% for a compression ratio of 10:1, an expansion ratio of 30:1 and a ratio of specific heats of 1.4.

### 1.2.1 Project Motivation

An increasing awareness of the consequences of energy use is emerging as a contemporary social issue. It is reflected in Global Warming concerns as a consequence of Green House Gas (GHG) emissions. Internal combustion engine emissions are a significant contributor. The Australian Green House Office [2] reports that transport contributed 14% of 2005 GHG emissions in Australia. The vast majority of this was from internal combustion engine powered vehicles. 50% of total emissions in that year were reported from stationary power generation to which internal combustion engines also contribute. These figures indicate that at least one sixth of the GHG produced in Australia is from internal combustion engines.

Many government agency and corporate web sites present the sustainability of internal combustion engine fuels in varied lights. OPEC [3] countries report having
added reserves in the period 2000-2005 which exceeded the cumulative production up to that time. ASPO Australia [4] reports Australian oil reserves have peaked and that it ‘serves as a microcosm of a world entering the peak oil era’. Many media and web publications reflect varied opinion about the peak oil phenomenon. Most dispute arises about the extent of yet undiscovered reserves. Consistently though, the tone is that pressure is developing on the supply side resulting in expected increases in oil prices in the near future.

In spite of the mounting pressures building against the use of internal combustion engines, they dominate the automotive field. This reflects their inherent suitability based on their many favourable traits, suggesting that the prolific use of internal combustion engines will continue. Consequently, any improvements in engine efficiency will become a requirement via supply and demand economic principles and through regulatory control of emissions through the political system.

Engine efficiency is becoming a more prominent factor in automobile manufacturer’s decision making. Ford Motor Company released a report [5] in 2006 outlining its approach to environmental concerns. It referred to several technological developments associated with improved efficiency and reduced emissions. Ford’s CEO, Bill Ford is quoted to say, “We are more convinced than ever that our long-term success depends on how our Company addresses issues such as climate change, energy security, …. noise and innovative use of renewable resources and materials”. Significant focus over recent years has lead to various technological advances in engine control resulting in efficiency gains. Features such as variable valve timing and its associated control have appeared recently in production vehicles. The preface of Variable Valve Actuation 2000 [6] refers to the predicted outcome of current variable valve timing technology to be camless valve actuation leading to efficiency improvements. In general however, little attention has been paid to engine configuration alternatives although several concepts have been developed to various extents in the past. Saab [7] researched a variable displacement engine using mechanical means to change the compression ratio. Australia’s Ralph Sarich invented an engine configuration alternative, the orbital engine. The Power House
Museum [8] has a site referring to the orbital engine development timelines and outcomes.

The opposed piston engine introduced in this work is an alternative engine configuration that, through its unique engine geometry attempts to address certain deficiencies of the thermodynamic cycle as employed in conventional engines. Specifically, the new configuration adopts a full expansion that aims to extract the proportion of the energy still available in a conventional engine when the exhaust valve opens. At WOT this represents approximately 20% of the total work by the thermodynamic cycle in a conventional engine. Furthermore, via crankshaft offset (in which the crankshaft centreline is displaced from the cylinder centreline), the piston motion used in the opposed piston engine creates a faster rate of change of the engine volume after combustion thereby reducing the gas temperature in a shorter time than for an equivalent conventional engine. This is intended to reduce conductive heat losses during the power stroke, improving thermodynamic efficiency. This earlier reduction in temperature is expected to allow the engine to burn a pure charge without an increase in NO\textsubscript{X} production. All these features are potential benefits of the new engine.

The opposed piston engine concept is investigated in this work through thermodynamic and friction simulations, finite element analysis of the main engine components and experiments performed with a prototype configuration.

### 1.3 Research Objectives

Broadly, the project explores an alternative approach to improving internal combustion engine efficiency. The particular goal of this research was to assess the viability of the proposed engine design. It extends to addressing the techniques and tools used for engine efficiency research.
Efficiency gains can be achieved by two main processes

- Improvement in thermodynamic cycle efficiency and
- Reduction in losses.

The thermodynamic model used is described in Section 3.2.2. The project Matlab simulation model employs an existing engine simulation program written by Ferguson [9]. The original software by Ferguson was earlier transcribed into Matlab code by Butsworth [10].

The program considers the following features of thermodynamic performance,

- Equilibrium combustion products calculation based on temperature and pressure using the simplified approach of Olikara and Borman [11].
- Heat loss from the working gas using a user-defined heat transfer coefficient.
- Burn time set by an input constant following the approach of Ferguson [9].
- Gas loss via blow-by estimated using a first order system approximation with a user-defined rate-constant.

The losses model created specifically for this application features,

- Ring friction
- Piston friction including dynamic effect and reaction loads
- Friction based on Stribeck theory
- Pumping losses
- Bearing and belt losses

Achievement of the prescribed goals for the project required the following additional processes,

- A GUI suitable for the large number of input variables and which allows quick feedback to the user of input changes
- Modelling of the proposed engine design in Solid Modelling software to create required build details
- Modelling of the proposed design in FEA software to confirm part integrity
- Building a prototype and associated test equipment
- Testing of the prototype to obtain performance data
- Review of modelling in light of test results
- Analysis and recommendations

The details of the processes used above are given in Section 3.2.

1.4 Model Refinement

This project produced experimental results from the testing of the prototype engine. The results broadly showed that some significant deficiencies existed in the original Matlab model used to design and predict the performance of the engine. Chapter 6 details the changes in the Matlab model using information obtained from the prototype test. Chapter 3 reports on the approach to the task of modelling the engine using information that was available at the time that modelling was initially performed. Subsequent information from the prototype test allowed several of the modelling assumptions and input parameters to be revised to more accurately reflect the measured performance of the engine. Chapter 6 reports on the changes to the Matlab model. Chapter 7 addresses the validity of the initial assumptions and parameters used in the model and assesses the appropriateness of the modelling initially used. In early chapters, the model is described at various stages of development. In all references to the Matlab model outputs, a reference is made to the modelling (pre-experimental or post-experimental). If no reference to the model version is made, the aspect referred to does not alter between models.
1.5 Conclusions

This project aims to assess the performance of an original engine configuration. It extends to exploring the techniques and tools required for that purpose.

The performance data obtained from the prototype test will allow an assessment of the viability of the engine configuration concept and provide feedback that will allow the validity of the modelling techniques used and/or improvements to the modelling assumptions and specifications.

The following chapters address the processes of the project methodology and outcomes.
2.1 Introduction

This chapter reports relevant information obtained from a literature review of previous work that used the same or similar techniques as those defined in Chapter 1. The four main mechanisms contributing to the new engine configuration are:

- opposed piston;
- crank offset;
- full expansion; and
- 100% scavenging;

and the literature review initially proceeds under these headings.

In addition, the modelling techniques needed in the design approach were reviewed. The computer modelling formed the base from which subsequent design decisions were made. This part of the project was substantial in content and significance. In this regard, the literature review identifies existing information in the areas of

- I.C.E. thermodynamic modelling;
- friction and losses modelling; and
- computational modelling techniques and applications.

The results of the literature review for each topic is addressed in the following sections.
2.2 Background

2.2.1 The Engine Concept

A literature review revealed no similar engine has been presented in the open literature to date. Web, journal and text searches failed to find a similar engine. A significant number of engine concepts were revealed by the searches and varied from refinements to the conventional 4-stroke theme to radical ideas seen only in literature form. The number of arrangements of the major components of an engine, referred to as the configuration, is essentially limitless. A web site that displays common engine configurations is published by diracdelta.com.uk [13]. Radical configurations that proclaim extreme performance levels in power to weight and/or efficiency can be found on the web. One sample is the ‘Massive Yet Tiny Engine’ [14] which claims a power to weight ratio 40 time that of a conventional 4-stroke. Many concepts were only seen in literature form, but many others have been built and are in service in one form or another.

Searches based on engine efficiency revealed that several techniques have been employed to increase engine efficiency with varying degrees of success and varying side effects. Some radical improvements to engine efficiency have been claimed by inventors such as Malcolm Beare [15]. He has named his alternative porting arrangement, the ‘6-stroke’ engine. That same term has been used by other inventors claiming to make use of the heat exhausted in a conventional engine. Bruce Crower [16] claims to have proven that by adding an extra revolution to a convention 4-stroke engine by modifying the valve timing, the energy of the exhaust heat can be returned to useful work by injecting water. Commonly, a reduction in power to weight is referred to as the primary counter-effect of attempts to significantly improve engine efficiency.

The Toyota [17] Prius uses a modified conventional light vehicle four cylinder engine operating at a higher compression ratio than a conventional engine. A
reduced inducted volume at WOT is achieved through valve timing. In a conventional engine, a reduced volume results from throttling and directly reduces the engine pressure before ignition and throughout the cycle relative to the pressure for a full inducted volume. The increased compression ratio in the Prius engine returns the pressure to an equivalent full inducted volume pressure and the expansion stroke partially reflects the full expansion theory described in Section 2.2.4. The engine subsequently produces about 60% of the power of the original unmodified engine. The power to weight ratio is reduced and the engine efficiency is increased. The power train uses high technology control and electric drives that result in net vehicle efficiency about 40% higher than equivalent sized vehicles. The combination of battery stored energy delivered by the electric motor simultaneously providing drive with the IC engine provides the prius with acceleration characteristics similar to a similar sized conventional light passenger vehicle.

Other engine configurations have become viable automobile engines. The Wankel Rotary engine has been thoroughly developed by Mazda [18]. It produces very impressive power to weight figures but is notoriously inefficient.

### 2.2.2 Opposed Pistons

Opposed piston engines appear frequently as an alternative engine design where the pistons essentially move in opposition. Hugo Junkers [19] invented an opposed piston engine in the late 19th century. His basic concept has been employed in many applications with a trend towards very large capacity installations. The application of such a configuration has found a niche as a large diesel engine. The most common installations are marine.
2.2.3 Crank Offset

In literature searches, Crank Offset is often referring to the angular displacement of one crank arm relative to others in a multicylinder engine, especially ‘V’ configurations. It is also referred to in the meaning used in this project, that is the displacement of the centreline of the cylinder from the crankshaft centreline. Academic research in this area has been carried out in the past. In all instances found, the application referred to a friction reduction technique. This is achieved by aligning the crankshaft so as to have the connecting rod more parallel to the cylinder through the high pressure portion of the power stroke. Myung-Rae Cho et al [20] showed that reduced normal reaction force of the piston on the cylinder wall reduced engine friction in certain scenarios. This technique has the opposite effect for heat loss reduction, maintaining the temperature difference between the burnt gas and cylinder wall for a longer time.

2.2.4 Full Expansion

The only reference to full expansion in literature as a topic of previous investigation was restricted to the Toyota Prius engine modification in Section 2.2.1.

2.2.5 100% Scavenging

No specific reference to complete scavenging was found in the literature review. Scavenging in two and four stroke engines is addressed in detail in all engine references. The significant point revealed is that residual gas, the gas left from the previous cycle due to incomplete scavenging, is used to ‘dilute’ the incoming charge. This reduces the maximum temperature reached during burning and subsequently reduces the total volume of nitrous-oxide compounds (NO\textsubscript{X}) produced. These
compounds contribute to one form of engine emission pollutants, in particular smog as described in Thermodynamics, an Engineering Approach [1].

In two stroke engines in particular, no control over inducted gas being exhausted without being burnt is possible over the complete engine speed range. Although it is often thought that the burning of oil in two stroke engines has lead to their demise due to emissions control legislation, it is the exhausting of unburnt hydrocarbons that is responsible, as described in Michigan State University research [21].

Four stroke engines are able to control the exhausting of unburnt gas via appropriate valve timing and is one of the many advances in modern engine technology. In conventional engines, the exhaust port is closed well before inducted gases have an opportunity to escape resulting in a necessarily significant volume of residual gas. Ferguson [9] applies residual gas analysis in his engine simulation program and this in turn is adopted by the Matlab model.

2.3 Applications of Thermodynamics to I.C.E.


Compression ratio is limited by fuel characteristics and properties and engine materials mechanical properties. In petrol engines, compression ratio is limited by detonation.

The form of the cycle, as represented by Figure 1.1, is able to be achieved by this engine design concept. A full expansion of the combustion gases is employed. In this
context, “full expansion” means expanding the combustion gases to local atmospheric pressure and for this to occur, the swept volume of the power stroke must be larger than the swept volume of the induction stroke. This uses the ‘extra’ energy that is expelled from a conventional engine when the exhaust port is opened while substantial pressure still remains in the engine. Full expansion has the capacity to add about 20% to the thermal work done in the engine cycle at WOT.

Full expansion is not employed as standard practice in conventional engines because they inherently induct the same volume as they expand. That allows for a larger power to be produced from that larger inducted volume and results in a higher power to weight ratio. That is probably the optimum configuration for a motor vehicle where the engine weight is a contributor to the overall efficiency of the vehicle. This project is orientated towards optimising engine efficiency; the anticipated decrease in power to weight ratio that accompanies the new configuration would limit the engine to stationary applications in the first instance.

2.4 I.C.E. Thermodynamic Modelling

Several avenues were revealed for a source of a thermodynamic model in the literature review. Undoubtedly, sophisticated software is employed by engine manufacturers researching this field. The author required a program that could be used in the Matlab application that the rest of the computer model employed. Several references to an ‘Engine Simulation Program - ESP’ were found. The Stanford Engine Simulation Program [23] and an associated text by Lumley [22] are examples. The author chose an existing engine simulation program written by Ferguson [9]. The original software has been transcribed into Matlab code by David Buttsworth [10] of USQ.
2.5 Friction Analysis

Several references were found for the use of the Stribeck Diagram (see Section 3.3) as the basis of a friction regime analysis, although no specific reference to its use in a computer model was found. The friction analysis uses the Matlab model to calculate the reaction loads on the piston during the cycle and applies the Stribeck Diagram to produce the resultant friction forces. The total friction losses from bearing, pumping and ancillary loads are estimated based on proportions of losses defined for existing engine configurations in relevant texts. Ferguson [9] and Stone [12] provided data for this purpose.

2.6 Computational modelling techniques and applications

Ertas and Jones [24] describe the basics of modelling which were built upon in this project. To readily reach an optimum configuration for any desired output, the main Matlab script sets up a GUI interface. This technique allows the many required engine specifications to be altered in an intuitive fashion resulting in a faster and better understanding of the effect of changes to input specifications and their relationship with output data, in particular, engine net efficiency.

Matlab was used as the model input through the GUI, for the model calculations and output. It consists of a suite of files called from the GUI using a set of predefined input data files to establish an optimum configuration for the criteria required (see Section 3.4). Subsequent to the information obtained from the prototype test, the Matlab model was modified. This allowed the model to be run in alternative configurations by input options at the start of the main GUI file. The modifications allow the program to run with the original input assumptions and under the modifications that allow the model to correlate with the prototype test results.
2.7 Solid Modelling and FEA

The project did not attempt to assess the suitability or effectiveness of the two software applications used for the solid modelling and FEA analysis of the proposed engine. ProEnginer and Ansys were used as they are the applications available from the University’s resources. For all intents and purposes they achieve the same result as any contemporary equivalent.

2.8 Conclusion

The literature review established a complete basis for proceeding with the design. It exposed no specific source of an equivalent idea or design.

Several features of the necessary project methodology are based in existing techniques, computer programs and established practices. This project combines and expands on them to the degree required to produce a conclusion about the original design concept and the computer modelling used to establish the design criteria.

The Matlab based modelling and subsequent solid modelling and FEA are detailed in Chapter 3.
Chapter 3 Project Methodology and Engine Design

3.1 Introduction

The overall project task requires a succession of subtasks to be completed in order. This sequence was broadly:

- Create a computer model to predict the engine performance and specifications.
- Solid model the engine to produce part drawings, confirm functionality, provide geometry for FEA.
- FEA analysis to confirm component life.
- Source materials, components and workshop services.
- Employ USQ workshop facilities to fabricate some components.
- Fabricate remaining components.
- Assemble and install with dynamometer on test bench.
- Produce prototype performance data from engine test.
- Compare computer model performance specifications with test results.
- Assess and report.

The BEng undergraduate course has provided all of the associated exposure to allow the above integrated process to be completed. Prior experience of the author enabled him to fabricate the majority of the engine components. The USQ workshop fabricated the components outside the capability of the equipment to which the author had access. Familiarity of the author with common engine components allowed ready recognition of alternatives for off-the-shelf items and for fabrication techniques.
3.2 Software Modelling

3.2.1 Overview

The primary computer model used to analyse the engine is written in Matlab. This allows the initial concept to be refined and specified. It performs several functions including, physical parameter analysis, thermodynamic analysis, friction and other losses analysis, optimisation and visualisation of the cycle. The input is achieved through a created GUI, allowing fast and effective changes of input specification with immediate user feedback.

3.2.2 Thermodynamic Model

The primary performance characteristic of internal combustion engines are governed by the fundamental physics of thermodynamics. The project employs an existing engine simulation program written by Ferguson [9]. The original software has been transcribed into Matlab code by Buttsworth [10].

The main script calls the appropriate files of the Matlab engine simulation program where required. Several parameters used in the original Matlab engine simulation program are redefined to reflect the specific idiosyncrasies of the new engine configuration. The thermodynamic model outputs the net work done in one cycle and is the starting point for the efficiency analysis.

3.2.3 GUI Input

To readily reach an optimum configuration for any desired output, the main Matlab script sets up a GUI interface. This technique allows the many required engine
specifications to be altered in an intuitive fashion resulting in a faster and better understanding of the effect of changes to input specifications and their relationship with output data, in particular, engine net efficiency. Sets of default input data specifications are assembled as files commencing with ‘enginespec…’. The main file, ‘slider_bar_input’ requires a default set of data from an input data specifications file. That file is called on line 31 of ‘slider_bar_input’. To reset the default data used, one can alter the file name called on line 31.

Figure 3.1 – Matlab Gui Input
The mouse pointer is displayed as cross-hairs showing that Matlab ‘ginput’ is active. The eleven input specifications are named with their ranges and defaults displayed.

To follow the function of the Matlab program, see the Matlab script ‘slider_bar_input’ in Appendix B for details. A structure chart assists at the beginning of Appendix B. Comments are included to assist in following the Matlab code. Figure 3.1 shows the GUI input Matlab figure. The Matlab function ‘ginput’ takes the mouse pointer position (the crosshairs) click, calculates its relative position on the scale bar for the particular engine specification chosen. It recalculates the displayed value of the specification and repositions the indicator ‘circle’ to confirm the new value. It initiates a new calculation of the Matlab Engine Simulation Model.
Figure 3.1 shows the default values on the right as set in the script code. The ‘Exit’ and ‘Current Spec’ buttons close the window and calculate the current specifications respectively. It shows the minimum and maximum values of the range and the names set for each specification.

The next stage of the Matlab model is initiated by the ‘ginput’ mouse click or the ‘Current Spec’ button. The program takes the list of input specifications as displayed and calculates the engines component position using first principle geometry (see Section 3.2.5). Together with data from ‘enginedata’, the geometry gives the required input for the engine simulation section of the program which gives the engine thermodynamic cycle outputs. ‘slider_bar_input’ calls several files including files used by the engine simulation program in its original form. Many of the variables used between the files are made available to each file as global variables. At the beginning of Appendix B, a broad outline of the file connectivity is shown in a Structure Chart. The variables names are self explanatory.

The program uses the Matlab function ‘ode45’ to achieve the required integration in the engine simulation program. The program steps through the cycle in one-degree-of-crankshaft-rotation increments. It generates engine pressure and other thermodynamic properties for the cycle at each increment. These outputs are used by ‘slider_bar_input’ to calculate the connecting rod loads and resulting cylinder-to-piston reaction loads that are used to calculated piston friction losses.

Other losses are calculated in various files and assembled in ‘slider_bar_input’ to be displayed in the GUI. The primary objective is the engine efficiency, which is calculated from the engine simulation program outputs adjusted by losses. Figure 3.2 shows the GUI after the program has completed calculations and returned outputs.
Figure 3.2 – Matlab GUI displaying outputs

The right hand half of the screen displays outputs in numerical form for the simulation run from the last input specifications. The position plots for each piston are shown as the blue and black plots. Other features are described in Section 3.2.4

The right hand side of the screen displays the outputs. With practice, the user can intuitively learn what input changes produce what output changes. The primary goal is maximum efficiency. A variation to any input specification changes the output. Some changes will result in impossible combinations. For example reducing the ‘Crank to Crank’ dimension will bring the crankshafts and consequently the pistons closer together. If other specifications are not compliant, the piston position lines on the output plot will overlap meaning the pistons will contact and interfere with each other. The output side of the screen is enlarged in Figure 3.3.
Figure 3.3 – GUI output

The GUI output screen is described in Section 3.2.4. The two plots represent each piston position through the cycle. All numerical values required to assess the performance of the input specifications are displayed.

At the bottom of the image, ‘P.C.’ referring to Piston Clash is the piston separation in millimetres at their closest point immediately before the exhaust port closes. The Matlab script serves as a tool and is not ‘debugged’ to the extent that it can accommodate any inputs and achieve physically achievable outputs. It does not for example require P.C. to be positive. Other specifications can be set that would also produce impossible combinations. The user is required to analyse the output to assess its viability.
The ODE solvers used in the engine simulation program can not always complete their calculations if the inputs exceed specific limits. Errors may result from the integration process having ill-defined boundaries or the input specifications attempted result in an unsolvable set of equations. The consequences for the program are that either the program is put into an endless loop or it takes an impractical length of time to produce results. It can also produce error symbols in the results displays. Once again, the user is required to assess the appropriateness of the data.

The ODE solver can also display an error message in the Matlab workspace indicating that the results may not be accurate because of technical limitations in the function. This is also not evident on the GUI display and requires user assessment.

### 3.2.4 Features of the GUI output

The prime purpose of the GUI is a graphical display of the piston motion, but it also shows various required numeric outputs. The following refer to Figure 3.3,

- In the top left hand corner, the specification file used in shown.
- In the top right hand corner, the general specifications for the simulation that are defined in the Matlab file, ‘enginedata’ are shown. They are independent of the specification input by the GUI (initially read from an ‘enginespec…’ file).
- Inducted volume, compressed volume and expanded volume are shown as numeric values and as positions in the cycle by vertical dotted lines.
- The ‘burn start’ (ignition) position is shown as a vertical dotted line.
- The following cycle features are displayed;
  - C.R.Ign (Compression Ratio at ignition).
• Exhaust port height (the exhaust port upper surface position relative to the piston positions) It is used to position the piston plots to commence at the close of the exhaust port.

The primary output data is displayed in the centre of Figure 3.3

• Int WORK (internal work per cycle in joules). Includes induction and exhaust work.
• Int Eff (Internal Efficiency). Calculated from the engine simulation program Outputs modified by induction and exhaust work per cycle.
• Net WORK (= internal work – losses).
• Net EFF (Net Efficiency). Calculated from the energy required to produce the internal work at the internal efficiency divided into the net work.

The four phases of the cycle, Induction, Compression, Power and Exhaust are also illustrated in Figure 3.3 to depict their positions in the cycle.

### 3.2.5 Piston Position, Velocity and Acceleration

Piston position is required throughout the Matlab program, which defines the path of the pistons allowing the position of other engine features including the spark plug ports and the exhaust port to be specified relative to the pistons. The engine simulation program requires specification of the piston dynamics in order to calculate engine volume and its rate of change. The piston position is used to calculate piston speed and acceleration used to determine fiction losses. The piston position is calculated in the Matlab program using the geometry of Figure 3.4.
The piston height is the critical parameter governing the linear motion of the piston. From geometric considerations (Figure 3.4), instantaneous piston height can be expressed as a function of crank angle as,

\[ h = T \sin \theta + \sqrt{r^2 - (T \cos \theta - s)^2} \]  

(12)

Figure 3.4 – Piston motion geometry

In the Matlab program, the position of the pistons is the position of the gudgeon (wrist) pin centre-line. In the solid model (and prototype), the crankshaft separation has the distance from the top of each piston to the gudgeon pin added to the Matlab ‘Crank to Crank’ specification.

The velocity and acceleration of the pistons is calculated where required in the Matlab program by the ‘diff’ technique. The positions are calculated at the defined increment of one degree of crankshaft rotation and the ‘diff’ function in Matlab returns the difference between successive positions. Dividing by the time for one degree of rotation gives a reasonable approximation of piston velocity at that point.
Repeating the ‘diff’ process on the velocity gives the acceleration. For a piston position array of length 361 (one degree increments and returning to the start position), this differencing process results in a velocity array of length 360 and an acceleration array of length 359. The velocity and acceleration arrays are returned to length of 361 using the ‘interp’ function.

### 3.3 Friction and Losses Model

The net efficiency is the internal thermodynamic work done by the engine less the losses due to friction, pumping and the ancillaries load divided into the net energy available in the fuel burnt. The friction includes the piston, ring and bearing friction. Pumping losses define the work required to transport the inducted gas into the engine and expel the burnt gas. Ancillary losses are restricted to the synchronous belt drive required to maintain the engines two crankshafts at constant relative speed.

The friction losses model is also implemented in Matlab. No appropriate existing software could be sourced for this purpose and the model used is based on fundamental first principle physics of the loads and reactions within the engine on a dynamic basis at each one degree of crank rotation. Dynamic effects from the angular momentum of the connecting rods are not considered and this seems reasonable because relative to the pistons in motion, the connecting rods are very light. Their dynamic effect was determined to be of little influence. A more thorough analysis would be achieved with the inclusion of the dynamic effect of the connecting rod and might be warranted in future use of the Matlab model. Figure 3.5 shows the force analysis used in the friction model.
The engine pressure is produced by the thermodynamic model. $\theta$ and the piston acceleration, $a_{\text{piston}}$ are produced by the Matlab piston motion model. Resolving the forces in the axial and normal directions gives,

\begin{align*}
T \sin \theta + F - pA &= m_{\text{piston}} a_{\text{piston}} \\
N + T \cos \theta &= 0
\end{align*} \quad (13)

Also,

\begin{align*}
F &= \mu N \\
(15)
\end{align*}

Combining these equations produces, for the piston moving down,

\begin{align*}
T &= \frac{ma + pA}{\left(-\frac{\sin \theta}{\mu \cos \theta} + 1\right)} \\
&= ma + pA \quad (16)
\end{align*}

and for the piston moving up,

\begin{align*}
T &= \frac{ma + pA}{\left(-\frac{\sin \theta}{\mu \cos \theta} - 1\right)} \\
&= ma + pA \quad (17)
\end{align*}
The script written for the friction losses model uses the model in Figure 3.5 in an iterative approach in Matlab based on published coefficient of friction data.

Several references were found for the use of the Striebeck Diagram as the basis of the friction regime analysis. Figure 3.6 shows a Striebeck diagram used in Introduction to Internal Combustion Engines [12]

\[
\frac{\mu v}{p} = \text{Sommerfield Number } \times 10^4
\]

where \( \mu \) = viscosity
\( v \) = sliding velocity
\( p \) = bearing pressure

**Figure 3.6 – Striebeck diagram**

Engine lubrication regimes defined in the Striebeck diagram. The diagram reflects the general nature of engine friction. Only the piston skirts operate in full hydrodynamic conditions in conventional engines referred to here. Adapted from Stone [12]

The plot in Figure 3.6 was encoded in Matlab by separating it into three subplots. The two straight sections were referenced by ‘polyfit’ and ‘polyval’ of degree one. The curved section was referenced by a ‘spline’ approximation.

The Matlab file, ‘pistfriction’ uses an iterative approach to calculate the piston friction losses. An initial estimate is calculated by setting the coefficient of friction to 0.01. Using the engine pressure from the engine simulation program together with the piston acceleration and connecting rod angle, an initial normal reaction force is
calculated according to Figure 3.6. The Sommerfield number required to return a coefficient of friction from the Stribeck diagram is then calculated. Depending on the direction of the piston (away from or towards the crank) and the calculated normal reaction, the program determines a temporary coefficient of friction and returns to the second iteration. By trial and error, only five iterations were necessary to achieve friction forces within 0.1% of those achieved with a large number of iterations. Since the file calculates these friction values on every initiation of the program by the input GUI, reduced calculation time was useful in reducing the total time between input changes and display of results.

The value of viscosity used equated to the general data for SAE 50 oil at 120°C. The engine inducts oil and operates with 2-stroke oil in the crank cases.

The total friction work per cycle is calculated from the sum of the instantaneous friction force multiplied by the piston displacement increment for each piston over one crankshaft rotation.

### 3.4 Optimisation

The primary goal of the project is to produce an engine with improved efficiency over conventional engines. Any one set of engine specifications can be assessed with the Matlab engine simulation program model. An optimisation feature allows the engine simulation program model to be looped with a succession of input specification files for varying sized engines. These files were assembled by trial and error and are listed in Appendix B, followed by a sample of the engine input specification files, ‘enginespecoptsizeschosen6’ and the specification file reflecting the dimensions achieved in the prototype, ‘enginespecprototype’. The various specifications all maintain the same bore size and ignition timing. They are pre-run in the engine simulation program to trim the specifications to produce the same maximum engine pressure. The comparison based on a constant maximum engine pressure is performed with Matlab script, ‘compile_data_CEP’. An alternative
comparison is done using a constant effective compression ratio in ‘compile_data_CCR’. These optimisation simulations were performed with the original Matlab model. They were not repeated with the revised (corrected) Matlab model. The optimisation script collates the data and displays a plot of thermal efficiency and net engine efficiency.

Figure 3.7 shows the Matlab graphical display of the optimisation script. It shows that an optimum efficiency is predicted for the specification numbered 6.2 in the original Matlab model. The general trend of increasing thermal efficiency is shown in the top plot. The increase is the result of a lower surface area relative to the volume displaced by the engine reducing thermal losses. It is also aided by an increase in piston speed for a larger engine. The comparison is based on a constant bore size. The net efficiency plot reveals that at small specification settings (low Spec number), the thermal efficiency is poor and at larger specification settings the friction losses dominate.

![Efficiency Vs Inducted Volume for Various Eng Specs](image)

**Figure 3.7 – Comparison of the original Matlab model prediction of engine efficiency**
The optimisation program showed that configurations in a broad range exhibit the potential for optimum efficiency.

### 3.5 Matlab Model Plots

The Matlab model is used to display various output data as plots. These are used as visualisation aids for verification of program outputs, decision making and presentation of data. Appendix C contains the list of plot scripts. The plots scripts require that ‘slider_bar_input’ is run first and a set of output data generated by the GUI.

![Figure 3.8 – Plot of both connecting rod inclinations.](image)

As an example of a visualisation plot, Figure 3.8 shows the plot of the connecting rod inclinations over one crankshaft revolution. The angles shown for the connecting rod inclination are relative to a conventional Cartesian reference plain with the cylinder oriented at 90°. The lower crankshaft has a slightly shorter connecting rod
and consequently shows a slightly higher maximum inclination (same offset). The plot shows that the inclination is extreme relative to a conventional engine and is expected to present the major drawback to the engine configuration. The friction model presented in Section 3.3 predicts that the reaction loads and resulting friction are acceptable.

Figure 3.9 – Plot of lower piston friction for the original Matlab Model optimised engine configuration.

Figure 3.9 shows some interesting features. It is the plot of piston friction (lower) showing the areas where the three friction regimes of the Strubeck diagram apply. The plot reveals that the piston friction is rarely in the hydrodynamic range. Either the speed is low or the load is high for the majority of the cycle. The vertical lines at approximately 160° and 300° indicate the positions where the lower piston changes direction. The friction is negligible for the cycle from 330° to 150° where the model predicts essentially zero relative pressure on the piston. This reveals that the friction model predicts the normal reactions due to accelerations are low compared to the reactions created by the engine pressure when the connecting rod is inclined. The
graph dips just prior to the second “Connecting rod parallel to axis” line, showing that at high piston acceleration and connecting rod inclination, the piston friction is apparent. The part of the cycle showing essentially zero friction results from the very low net piston force through the lower swing of the crankshaft. The ‘mixed friction’ zone displayed at about 60° results from the hydrodynamic friction on the induction stroke slipping into mixed friction at the maximum connecting rod inclination. Because the crank offset is such that the connecting rod is at maximum inclination at the end of the power stroke, even though the pressure has dropped to a small fraction of the maximum during the cycle, the very large connecting rod angle produces a very large normal reaction as the piston is decelerated.

3.6 Solid Modelling in ProE

The specifications obtained from the Matlab model are used to construct a complete solid model of the engines major components.

The first benefit of this process came when the selected optimum configuration proved to be impossible to construct. The interference resulting from the low connecting rod/crank throw ratio resulted in too little room for sufficient material in the cylinder walls at the extreme piston position to adequately support the piston. No combination of crank throw configurations allowed for a suitable arrangement using a simple crank and connecting rod. Consequently, the engine simulation program was employed to reconstruct the engine specifications with an increased connecting rod length. The resulting net efficiency necessarily decreased as it had been previously optimised. The loss in efficiency modelled as approximately 2.5% net or 7% of the optimum efficiency. These changes were readily achieved by the established GUI.

Several ProE features are employed including part mass calculations. The ProE model gives an impressive image of the engine, including the ‘mechanism’ feature that displays the model working. Figure 3.10 is the engine solid model assembly
showing the cylinder and lower case as transparent parts revealing the engine internal components.

Figure 3.10 – ProE solid model of Engine Assembly

3.7 Engine Configuration and Features

At this stage, the details for the engine components and their specification became necessary. The exhaust port position was established by the Matlab model. An original intention to arrange the induction through a poppet valve in the lower cylinder was abandoned due to its complexity and replaced by a reed valve controlled induction port in the cylinder wall. The main bearings were chosen as deep groove ball bearings with the outer bearings having an integral seal. Needle roller bearings used in a production 2-stroke mower engine were selected for the connecting rods. The crankshaft is supported on one side only allowing the big end roller bearing to be assembled with a screw-in crank pin from the same production
engine. The crank timing is achieved with a synchronous belt drive and idler tensioner. Carburation and ignition was arranged as required from off the shelf and available components. Two spark plugs firing simultaneously are employed. Relative to a conventional engine, the longest burn path is increased in the opposed piston engine because of the necessity to have the spark plugs outside the width of the piston. This suggested that the burn time would be significantly longer than for a conventional engine. To reduce the burn time and give the opposed piston engine a better chance of completing burn before the thermodynamics of the expanding gas impeded combustion, the inclusion of two spark plugs at near opposite positions in the cylinder wall was seen to be advantageous. It also allows the engine to be operated with either plug firing which would give some information about the burn characteristics from the prototype test.

### 3.8 Sizing and Material Selection

As the primary dimensions set by the Matlab model define the various parts in ProE, required features and fabrication techniques become apparent. The two modelling techniques interact. For example, the Matlab model uses the weight of the piston in the friction losses model. An initial estimate of that weight can be confirmed once the piston size is determined from the ProE model using its ‘Analyse’ function. This weight is returned to the Matlab model, which generates a revised net efficiency. The data presented so far from the Matlab model is based on the ProE predicted piston and connecting rod weight. In the final analysis, the actual piston weight measured from the prototype pistons were returned to the Matlab model.

As the features of the ProE model developed, techniques for fabrication were decided. The technique and material selection go hand in hand. Selection justifications for the major components are listed in Sections 3.8.1 to 3.8.4. Details of the modelled components are shown in Appendix E.
3.8.1 Crankcases

The crankcases are a complex shapes but can be turned in a lathe if some feature can be welded. This promoted the choice of mild steel as the material for the prototype crankcases.

3.8.2 Cylinder (Barrel)

The cylinder needs considerable machining if made from a single billet or considerable cutting, machining and welding if built up from several components. A single piece of hollow cast iron was sourced for the cylinder which provided a suitable bearing surface for the piston and easy machinability for the extensive machining required. Its thermal and physical properties are suitable as evidenced in its extensive use for light engine cylinders.

3.8.3 Crankshafts

The Crankshafts are a complex shape. Before their material selection was made, it was confirmed that the workshop facilities were able to machine the shape designed. The use of 4140 steel was confirmed when the workshop reported that it would have no effect on the production of the crankshafts relative to a softer, lower strength steel.

3.8.4 Pistons

ASM online handbook [25] originally confirmed that aluminium alloy 4032 has been developed as the preferred material for internal combustion engine pistons. It shows
properties of high thermal conductivity with moderate coefficient of thermal expansion and moderate strength. Aluminium 4032 could not be sourced in bar stock suitable to machine the pistons and casting was considered impractical in this instant. In choosing a substitute material, 4140 steel, mild steel and aluminium 5068 were assessed for general suitability. The following table displays the general properties of each.

<table>
<thead>
<tr>
<th>Property</th>
<th>Alum 4032 (reference)</th>
<th>4140</th>
<th>mild steel</th>
<th>Alum 5068</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coeff of Thermal Expansion (µm/m °C)</td>
<td>20</td>
<td>13</td>
<td>13</td>
<td>24</td>
</tr>
<tr>
<td>Thermal Conductivity (W/m.K @ 20°C)</td>
<td>155</td>
<td>43</td>
<td>50</td>
<td>127</td>
</tr>
<tr>
<td>Hardness</td>
<td>120 HB</td>
<td>200 HB</td>
<td>120 HB</td>
<td>80 HB</td>
</tr>
<tr>
<td>Specific Heat (J/kg.K)</td>
<td>864</td>
<td>470</td>
<td>500</td>
<td>900</td>
</tr>
<tr>
<td>Tensile Strength (MPa @ 200°C)</td>
<td>90</td>
<td>700</td>
<td>450</td>
<td>152</td>
</tr>
<tr>
<td>Density (g/cc)</td>
<td>2.68</td>
<td>7.75</td>
<td>7.70</td>
<td>2.66</td>
</tr>
</tbody>
</table>

Table 3-1 – Comparative properties of possible piston materials


The first choice for a replacement material for aluminium 4032 was aluminium 5068. Its properties are not significantly different from 4032. Because the pistons were to be machined, the structure supporting the gudgeon journal in conventional pistons were not able to be created. Consequently, the fatigue characteristic of aluminium in the situation was not able to be confidently predicted. 4140 steel was chosen even though it has an inferior combination of thermal properties. Because of its high strength, the piston could be constructed with very thin walls. The deformation that resulted when modelled in ProE allowed for a higher fatigue life than the aluminium model. The weight of the piston modelled in ProE was less than a version of an aluminium piston that reflected the dimensions of the piston from the
mower engine from which other engine components were sourced. The machining process was slow and difficult and the tolerances in the prototype piston were moderately attained and the piston weighed slightly more than the ProE prediction. As explained in Section 6.2.1, the choice of 4140 steel pistons could have contributed to increased blow-by. The following calculation was used to choose a piston clearance.

The coefficient of thermal expansion for cast iron is similar to 4140 steel. Assuming a temperature difference between the piston and cylinder of 300°C, then the relative expansion of the 63.3mm piston will be,

\[
300 \times 61.3 \times 13 \times 10^{-6} = 0.24 \text{mm}
\]

The pistons were machined with a clearance of 0.5mm at the crown and 0.1mm at the skirt.

### 3.9 Interference and Function

The biggest advantage from ProE modelling comes in the ability to define the interference of moving parts. The engine’s necessary dimensions require compromises on spacing of parts relative to the configuration used in conventional engines. The primary difference is the length of the connecting rod. That requires the bottom of the piston to move further into the crank case than in a conventional engine. Therefore the cylinder must extend far enough towards the crankshaft to allow the piston to be sufficiently supported. This creates interference between the crankshaft and the cylinder. ProE allows the interference of this type to be assessed and compromises made. The connecting rod length also required the crank arm to be inclined so as to allow the cylinder sleeve to be sufficiently long to support the piston at its nearest point to the crankshaft. This inherently increases the bending moment in the crankshaft. This is exacerbated by the extra width required to include sufficient weight in the counterbalance lobe at the restricted diameter of the case.
That case diameter is a compromise between the counter-balance diameter, the piston height and the exhaust port position. The final crankshaft configuration was a best judgement compromise between part properties and function. The crankshaft form can be seen in Figure 3.12

### 3.10 Engine Concept Display

The ProE mechanism model makes an impressive display for presentation purposes. It allows for the visualisation and confirmation of the engines performance features including the position at which the exhaust port closes relative to the position of the opposing piston. It generates an image that allows for easy interpretation of the full expansion feature of the engine and the faster than conventional engine expansion immediately after combustion.

The ProE model proved the Matlab model geometric analysis to be correct. When the dimensions set from the Matlab model were applied to the ProE mechanism model, the pistons are seen to approach minimum separation as the exhaust port closes as was intended and as was predicted in the Matlab GUI.

### 3.11 Drawings and specifications

Appendix E shows the drawings required for the fabrication of the various components produced in ProE. Only the drawings required to produce the crankshaft were completed to a standard enabling an independent workshop to fabricate the part. All the other drawings were produced to standard that provided sufficient information for the author to fabricate the components.

The major components are also shown in Appendix E as 3D models in various orientations allowing features and function to be seen.
3.12 ProE Models as a Basis for FEA

The ProE models are directly exported to Ansys for FEA analysis. Only the critically loaded components were analysed by FEA.

3.13 FEA Analysis

The two complicated parts of the engine that differ substantially from a conventional engine and therefore require stress analysis are the pistons and cranks. Each piston is essentially the same, one uses three compression rings and one uses two. This was necessary to maintain at least one ring seal as the pistons moved past the spark plug port during compression and 2 ring seals for the combustion pressure. The two crankshafts are the same, except for a slightly different throw.

The Matlab model was used to define the loads and the ProE models define the geometry. Ansys is used to assess the maximum stress and in particular, the predicted fatigue life. The prototype is designed to operate for sufficient time to produce usable performance data.

Figure 3.11 shows another plot produced by the Matlab model. This plot gives the relative angle between the connecting rod and the crank throw and assuming essentially no friction at the big-end (needle roller bearing), will define the load direction for the Ansys analysis. The plot positions the connecting rod load, also determined in the Matlab model, so as to assess the load to apply in Ansys.
The Ansys analysis displays various physical properties of concern including stress, deformation and fatigue. Figure 3.12 displays a Von-Mises stress plot of the crankshaft loaded as per the analysed loads of Figure 3.11
The pistons were also analysed in Ansys. As revealed in Section 3.9, several compromises were required to fabricate the engine. With regard to the pistons, the unavailability of a suitable aluminium wrought bar required returning to ProE and Ansys and re-designing the pistons from an available material. 4140 steel was chosen even though it has material properties which disfavours its use in conventional engines. It allowed the crown and skirt sections to be reduced and consequently reduce the piston weight. ProE and Ansys were used in an iterative process to refine the piston form. Figure 3.13 shows the deformation plot for the piston. This plot uses the gudgeon pin bearing surfaces and pressure on the crown for the loads. The piston is loaded to the maximum pressure achieved in Matlab model of 6MPa. The fabrication technique excluded the formation of interior ribs and the plot was used to check if the piston deformation due to the unsupported bearing sections would cause the piston to interfere in the cylinder under operating loads. The maximum diameter increase determine by the Ansys analysis was in the order of 10% of the piston clearance used.
On assembly of the engine, it was noticed that the belt tension would induce bending in the cylinder. Ansys was employed to check whether the induced bending would result in deflection in the cylinder that would create interference with the pistons. Figure 3.14 shows the resulting deformation plot for the cylinder loaded at the mounting bolt holes under a load equivalent to the maximum engine pressure torque acting as tension in the belt. The local flange deformation was not considered. The Ansys analysis showed that the cylinder deformation from bending was less than 0.005mm over the length of a piston, resulting in no interference with the pistons.
3.14 Engine Components

The engine components were sourced in various ways. Appendix D lists the engine components, whether they were fabricated, purchased or modified existing components. It lists the material used for each component or the manufacturer’s code for the components purchased.

3.15 Prototype Build

Some components were available as manufactured parts including the connecting rod bearings, the main bearings, the piston rings and the crank pins. Parts and their
source are listed in Appendix D. The majority of the components needed to be fabricated. The USQ workshop was employed to machine the crankshafts. All the other components have been fabricated by the author.

The specifications supplied for the crankshafts were not met during its fabrication. The crankshaft throws, which control the relative positions of the pistons at each end of the cylinder, were outside the tolerance specified. Table 3-2 below shows the intended dimensions and the actual dimensions of the prototype for the critical engine components.

<table>
<thead>
<tr>
<th>Crankshaft Throw</th>
<th>Dimension</th>
<th>Intended Dimension</th>
<th>Actual Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upper Crankshaft</td>
<td>Throw</td>
<td>26.2±0.05</td>
<td>26.3</td>
</tr>
<tr>
<td>Upper Case</td>
<td>Offset</td>
<td>7.0</td>
<td>7.5</td>
</tr>
<tr>
<td>Upper Case</td>
<td>Centre-line/face</td>
<td>57.0</td>
<td>56.9</td>
</tr>
<tr>
<td>Upper Conrod</td>
<td>Centre/Centre</td>
<td>44.5</td>
<td>44.5</td>
</tr>
<tr>
<td>Upper Piston</td>
<td>Pin Centre-line/face</td>
<td>62.0</td>
<td>61.9</td>
</tr>
<tr>
<td>Lower Crankshaft</td>
<td>Throw</td>
<td>26.0±0.05</td>
<td>25.5</td>
</tr>
<tr>
<td>Lower Case</td>
<td>Offset</td>
<td>7.0</td>
<td>7.5</td>
</tr>
<tr>
<td>Lower Case</td>
<td>Centre-line/face</td>
<td>57.0</td>
<td>57.0</td>
</tr>
<tr>
<td>Lower Conrod</td>
<td>Centre/Centre</td>
<td>41.5</td>
<td>41.5</td>
</tr>
<tr>
<td>Lower Piston</td>
<td>Pin Centre-line/face</td>
<td>62.0</td>
<td>62.1</td>
</tr>
<tr>
<td>Barrel (Cylinder)</td>
<td>Face/Face</td>
<td>97.0</td>
<td>97.0</td>
</tr>
<tr>
<td>Exhaust Port</td>
<td>Height</td>
<td>16.7</td>
<td>17.6</td>
</tr>
</tbody>
</table>

Table 3-2 – Prototype dimensions

Various dimensions interact and can be compensated for during assembly. The measured dimensions of the prototype were returned to the Matlab model to produce the expected efficiency of the configuration. The crankshaft throw discrepancies resulted in an inability to achieve the intended compression ratio. In a conventional engine the compression ratio could be altered by reducing the compressed volume. In this engine the relative positions of the pistons at the exhaust end of the cylinder
restricts that option. The engine was assembled as the fabricated components allowed.

### 3.16 Engine Component Features Specific to the Opposed Piston Configuration

Details of the engine components form is able to be seen in the drawings and images in Appendix E. Several features of the engine component differ in detail from a conventional engine, but have the same general form and function. Those features are,

- Position of the rings on the pistons;
- Inclined crank arm;
- Large gudgeon centre to piston crown height;
- Lower portions of the cylinder removed to accommodate the motion of the crank arm; and
- Large inner main crank bearing.

#### 3.16.1 Piston Rings

The rings are arranged on the pistons so at least one ring is in complete contact with the cylinder wall at all times during the compression and power strokes as the pistons move past the sparkplug ports. The lower piston has three rings with the top (inner) ring included to improve the compression seal immediately after the intake port is closed. The lower two rings are never exposed to the spark plug port. The lower piston should have a similar seal potential to a conventional engine.

The upper piston rings are farther from the crown of the piston then in a conventional engine. This allows both rings to be in complete contact with the
cylinder wall at ignition and for the majority of the power stroke. The rings on the upper piston are separated by more than the diameter of the spark plug port so that as the rings move past the port, one of the two rings maintains the ring seal.

3.16.2 Inclined Crank Arm

The crankshaft has the big end bearing pin farther from the crankshaft main bearing in the axial direction than a conventional engine. This is necessary to allow the very short connecting rod to function. In a conventional engine, a longer connecting rod allows the main bearing and big end bearing pin to be closely separated axially because the piston does not encroach on the crank throw space. The crank arm in the opposed piston engine is inclined to allow the cylinder wall to penetrate farther into the crankcase thereby supporting the piston at its closest approach to the crankshaft.

3.16.3 Gudgeon Centre to Piston Crown Height

The large distance between the gudgeon centre and piston crown is required to get the crown of the piston out of the vicinity of the crankcase to allow the exhaust port to open in a functional position. The short connecting rods cause this effect.

3.16.4 Cylinder-Crankshaft Interference

The short connecting rods result in the piston moving close to the crankshaft. To support the piston at its closest approach to the crankshaft, the cylinder is formed to extend to the crankshaft centreline. The part of the cylinder that would consequently interfere with the crankshaft is removed. Sufficient support for the pistons is maintained by the larger gudgeon centre to piston crown height.
### 3.16.5 Inner Main Crank Bearing

The consequence of the large axial distance between the inner crankshaft bearing and the big end pin is a large induced bending moment in the crankshaft as described in Section 3.9. To accommodate the load the crankshaft is necessarily larger in diameter at the inner bearing than in a conventional engine. Consequently the inner bearing is large.

### 3.17 Ancillary Components

Several components necessary to complete the engine to the extent that would allow the test, were fabricated by hand as required. For example the two case covers are similar in appearance but can not be interchanged. On disassembly, the components should be marked so as to be reassembled in the same positions.

The crankshafts had counter-balances are attached by welding. The size of the counter balances were restricted by the necessarily short connecting rod. As described earlier in Section 3.9, the crankcase diameter was subsequently limited resulting in a compromise between the counter-weigh width, crankshaft strength and balance effects. A complete balance cannot be achieved as is the case with all single cylinder conventional engines. A reasonable estimate of the compromise counterbalance was confirmed by the engine test. Up to the 1500 rpm achieved in the test, the engine showed no sign of excessive vibration.
3.17.1 Induction Reed Valve

The engine configuration requires the induction phase to occur immediately after the exhaust port is closed by the lower piston. This was initially intended to be achieved by a poppet valve in either one of the two piston crowns operated by crankshaft position. To avoid the complexity of that mechanism, an induction port was added to the cylinder. Because this port is exposed during the last stages of the power stroke, a one way valve is required to exclude burnt gas from being exhausted into the induction port. This was achieved with a small reed valve fabricated to reduce the volume exposed to burnt gas during the cycle. Figure 3.15 shows photographs of the various views of the fabricated reed valve.

Figure 3.15 – Induction reed valve
3.18 Important Assembly Notes

The following should be noted:

- The piston rings are fabricated from a Victor Power Torque 160 2-stroke mower engine rings. Their thickness, depth and end relief have been modified to match the ring grooves of each piston. The rings will not assemble correctly if interchanged between ring grooves. Mark each ring on disassembly.

- The rings are fragile (cast iron). On assembly, position each ring in its proper groove and gently insert the piston into the cylinder with the gudgeon pin, connecting rod and clips assembled. Manipulate the rings one at a time until their ends fall freely each side of the groove pin. Slowly push the piston into the cylinder without any undue force. If the rings are deflected too far they will break. Once the piston is in the bore so that the rings are no longer visible, rotate the piston to its operating position to prevent the rings ends from catching in any of the cylinder ports.

- A spacer is fitted between the crankshaft inner bearing and its spigot face on both cases. This aligns the big end and connecting rod so no contact occurs with the cylinder. These spacers are specific to each case and must be installed in their current orientation.

- Once each crank is positioned in its case (requires a soft drift) and the lower piston (exhaust end) is in the cylinder with its connecting rod attached, slide the cylinder into the case manipulating the connecting rod free of the crank. Before the retaining studs extend past the cylinder flange insert the exhaust collector and gasket over the exhaust port. Before the Cylinder face meets the case face, start the four retaining nuts on the studs. Ease the case and cylinder together by tightening the retaining nuts. Line up the connecting rod big end and the crank throw thread and insert the crank pin. A special tool was made to fit the crank pin. IMPORTANT: use an impact wrench to tighten the crank pin. A standard ½ inch impact on medium torque is sufficient. Rotate the crank to ensure no interference exists.

- Repeat the process for the other piston and case.
3.19 Conclusion

A thorough modelling process was employed to obtain the required specifications to proceed with the prototype build. The Matlab model predicted engine performance that justified the action to build a prototype. The Matlab model re-evaluated the predicted performance of the prototype after its as-fabricated dimensions differed from the design specifications. The prototype build was successful, enabling testing. Chapter 4 reports on difficulties with initial attempts to start the engine and subsequent engine test configuration modifications.
Chapter 4 Prototype Manipulation

4.1 Introduction

No major impediments to the engine build were encountered. Deviations from modelled specified dimensions resulted in the prototype not accurately reflecting the intended design specifications (see Section 3.15). The achieved prototype dimensions were input into the Matlab model which returned new expected performance data. The model showed that even though the engine specifications reduced the maximum engine pressure and predicted efficiency, the engine should still operate. This chapter reports on the difficulties encountered in starting the engine and the techniques employed in attempting to overcome these problems.

4.2 Engine Starting Problems

4.2.1 Cranking Speed

The first attempt to start the engine was unsuccessful. The engine was initially fitted with a modified 2-stroke mower pull starter which produced about three revolutions of motion from a single pull. The high friction of the engine assembly relative to its angular momentum resulted in a small number of starting cycles at a low speed in any one attempt. Infrequent ignition and burning resulted.

To address the cranking speed problem, a motorcycle starter motor was fitted to replace the pull starter. Some progress was achieved in that more frequent burning resulted but the engine displayed no potential to run under its own power.
Further barriers to progressing became evident and these are summarised below.

- The pistons rings are conventional 2-stroke rings and were not intended or capable of excluding crankcase oil from the combustion chamber. Without being able to get the engine spinning for sufficient time to bed the rings and without sufficiently frequent ignition and burning, oil accumulated in the combustion chamber and frequently bridged the spark gap.
- The inducted gas flow was very low and the carburettor metering of the mixture at those flows could not be confirmed. It could not be determined if the poor ignition and burn was a result of a poor mixture.
- It could not be determined whether a single ignition and burn had some dynamic effect on the next inducted charge causing the infrequent burn.

### 4.2.2 Engine Preheat

The next attempt to improve the potential for the engine to run was to preheat the engine with a conventional domestic fan heater. During this testing, the engine reached temperatures above 100°C (as observed with water droplet tests). Engine preheating caused more frequent and complete burning to occur as indicated by increased exhaust noise and an increase in cranking speed relative to the unheated condition. However, the increased burn was not sufficient for the engine to power itself.

### 4.2.3 Engine Motoring

The next impediment to engine starting that was addressed was the limited time over which the starter motor could be operated without the risk of exceeding its duty cycle. Initially an adapter was made to power the engine at the output shaft by a
small impact wrench. The net torque transmitted by the impact wrench was insufficient to crank the engine alone, but it did increase the speed at which the starter motor could crank the engine. Finally, a variable speed electric drill was fitted at the drive coupling on the engine output shaft. It provided ample cranking torque and was able to motor the engine to above 800 rpm.

At this stage, the engine still showed no positive signs of consistent ignition and burn. Some burning took place as evidenced by exhaust noise and smoke, but after prolonged motoring, no significant temperature rise was evident in the cylinder, suggesting that the burning was either very infrequent or very incomplete.

4.2.4 Mixture Control

The mixture control to this point was achieved by choking the carburettor. To assess if this was responsible for the lack of performance, the carburettor was modified to allow the main mixture to be varied over a large range.

The adjusted mixture had a perceivable effect - the exhaust note changed with changes to the main jet setting, recognised as resembling the change in exhaust note with mixture change that result in a conventional engine. More ignitions occurred at the optimum setting, but the engine still showed no sign of consistent ignition and burn or of powering itself at any mixture setting.

After the main jet adjustor was fitted, changes to the exhaust note were encouraging and confirmed that mixture had contributed to the lack of progress to that time. While using the choke to attempt to control the mixture, the exhaust note suggested that the carburettor metering of the inducted charge varied considerably as a result of an ignition event. In a conventional engine/carburettor arrangement, the choking technique normally has an effect on the inducted mixture which has a relationship to the inducted flow rate. For the very low flow rates involved in the testing up to that time, it is apparent that this flow effect on mixture was excessive while using the
choking technique and prevented a combustible mixture being inducted for any appreciable length of time.

### 4.2.5 Ignition Options

The engine was initially fitted with remote ignition timing adjustment (manual). The ignition used a standard electronic module and a magnetic pickup activated by a prong fitted to the upper case synchronous pulley. The ignition was initially set manually by positioning the prong at the pickup for the required piston position. Once the engine was being motored, the ignition timing could be checked using a strobe. This resulted in recognising that the spark was occurring at the retreating magnetic field from the pickup, not at the approaching field. This resulted in ignition timing at approximately 40 degrees later than the optimum modelled ignition timing and approximately 15 degrees after the pistons were at their minimum separation. Reversing the pickup connections to the module shifted the ignition point to the position initially expected.

This once again improved the ignition and burn but no combination of ignition timing and mixture resulted in a continual ignition and no significant heat was generated in the engine from burn.

### 4.2.6 Combustion Chamber Shape

The existence of a combustible mixture was confirmed by adjusting it through the full range until it exceeded combustible bounds in both the lean and rich directions. Ignition was confirmed to be in a position expected to produce burn. However, to this point the engine was still not showing any sign of powering itself.
Because burn was not occurring to any significant degree, the next attempted to improve the combustion was by adding a specific combustion chamber to the engine. A thin disc shaped combustion chamber results from the opposed piston engine design concept because complete scavenging is achieved as the pistons close the exhaust port. Therefore the piston crowns must be flat (or a shape that leaves no volume between the pistons when they are placed in contact with each other in the opposed orientation). Consequently, at ignition, the burn has to proceed across a thin disc like combustion chamber formed between the two flat piston crowns. Burn is necessarily initiated from the spark plugs at the edge of the thin disk-like combustion chamber. Because the spark plug ports were kept to as small a volume as possible to allow as large a separation as possible between the pistons at ignition, the combustion chamber formed had no resemblance to a conventional combustion chamber. The squish effect during the later stages of compression result in very high speed gas leaving the disc-shaped volume between the pistons and entering the volume formed around the tip of the spark plugs. Both pistons are also moving simultaneously past the spark plug port. This complex motion and large surface area to volume ratio was anticipated to cause burn problems. To establish whether a combustion chamber shape closer to that of a conventional engine would improve burn, physical changes to the engine were made.

The combustion chamber additions were most readily achieved by machining into the piston crowns. Figure 4.1 below shows a photograph of the upper piston with the U-shaped cavities added as a combustion chamber. Only the upper piston was modified as the lower piston’s upper ring was too close to the piston crown to allow any material to be removed from that piston. The crankshaft throw dimension errors had restricted the compression ratio to 4.9:1 determined by the Matlab model. The machined grooves added 0.6 cm$^3$ to the compressed volume. The Matlab model has the spark plug port volume input which is used in the thermodynamic analysis. Adding 0.6cm$^3$ to the spark plug port input specification allowed the Matlab model to predict the effect of adding the combustion chambers to the engine specifications. The additional volume reduced the compression ratio to 4.1:1.
The addition of the grooves in the upper piston improved the engine burn and at this stage of development, a consistent burn was achieved, but prolonged motoring with this burn occurring resulted in only modest temperature rises in the cylinder. The cylinder temperature reached only a small fraction of the temperature of a convention engine when operating.

### 4.2.7 Compression Pressure Achieved in the Prototype

To further investigate the cause of the marginal engine performance, a measurement of compression pressure was performed. Figure 4.2 shows a pressure gauge that was adapted to a fitting that could be screwed into a spark plug port. The fitting was made with a small non-return valve in the tip so that the pressure indicated would be similar if not the same as when a spark plug was fitted. The engine was motored in its operating configuration without ignition. The test confirmed that blow-by was a significant problem. Details of these compression tests are presented in Section 5.7.
Motoring the engine in the configuration created at this stage resulted in a compression pressure of only 125kPa (gauge) at 400 rpm rising to 200 kPa at 800 rpm. These figures were significantly lower than the Matlab model predicted and were judged to be a potential cause of the poor ignition and burn. At this stage it was expected that even if the burn characteristics could be improved, the very low compression pressures would result in very low thermodynamic cycle work. To improve the potential for the engine to operate, at this stage it was envisaged that improving the inducted charge volume would result in an improved potential for the engine to run from both the extra energy of a larger inducted charge and the improved burn that would result from the higher compression pressure from the extra inducted charge.

4.2.8 Blow-by Compensation

Because the prototype engine dimensions were able to be measured and they confirmed the Matlab predicted dimensions and because the engine showed a lack of performance, that reduced performance was speculated to be in part a result of a poor charge volume being inducted.
To assess the potential for improved induction volumes using crankcase pressure, the engine motoring pressure readings were taken with a regulated air pressure over the induction devices. Table 4-1 shows the results for pressurising pressures of 200 and 300kPa.

<table>
<thead>
<tr>
<th>Pre-induction Supply Line Pressure (kPa gauge)</th>
<th>Motoring Speed (rpm)</th>
<th>Pressure (kPa)</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>600</td>
<td>155</td>
</tr>
<tr>
<td>200</td>
<td>700</td>
<td>260</td>
</tr>
<tr>
<td>300</td>
<td>500</td>
<td>250</td>
</tr>
<tr>
<td>300</td>
<td>600</td>
<td>300</td>
</tr>
</tbody>
</table>

Table 4-1 – Engine compression pressure with elevated induction pressure

The results in Table 4-1 suggested that pre-charging could be beneficial, because motored compression pressure are measurably higher than without pre-charging (see Section 4.2.7). The lower then expected compression pressures shown in Table 4-1 suggest that the pre-charge pressure was decreased at the induction port due to insufficient flow in the supply line. For example in Table 4-1, the supply line pressure of 300kPa (gauge) with the engine motored at 600rpm, the compression pressure is only measured at 300kPa (gauge). Because the engine is shown to increase the pressure during compression, the supply line pressure at the induction port must necessarily be less then the regulated pressure of 300kPa (gauge). The pre-charge pressure was supplied by a workshop compressor with the pressure regulated at the tank. The results in Table 4-1 show that the required flow to the engine has resulted in a pressure drop in the supply line.

In an attempt to increase the charge volume and obtain in the engine the predicted compression pressures for minimum blow-by, a crankcase pump was fitted. The inducted gas was first drawn into the lower crank case and then pumped to the cylinder inlet port via a double reed valve arrangement in the crank case cover as depicted in Figure 4.3.
With the crankcase pump connected to the induction devices, Table 4-2 shows the resulting compression pressures.

<table>
<thead>
<tr>
<th>Motoring Speed (rpm)</th>
<th>Pressure (kPa gauge)</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>130</td>
</tr>
<tr>
<td>500</td>
<td>150</td>
</tr>
<tr>
<td>600</td>
<td>150</td>
</tr>
<tr>
<td>700</td>
<td>175</td>
</tr>
</tbody>
</table>

Table 4-2 – Engine compression pressure resulting from crankcase pump

The crank case pump accumulated approximately 25 kPa when not connected to the induction port. The pressure was available as a pulse when the piston moved towards the crankcase. Since this pulse lagged the induction pulse, no significant increase in inducted volumes resulted.

Motoring the engine with the crankcase pump connected had little effect on the engine compression pressure.
4.2.9 Improved Ignition

There were ample indications that the engine was not burning the induced charge completely. Because the burn had become reasonably consistent through the addition of the groove combustion chambers in the upper pistons, it was concluded that the burn must be generally initiated but that the flame did not propagate completely or consistently through the charge. To assess whether several ignition sparks over the expected combustion time would result in an increased burn, the single ignition pickup prong was replaced with three prongs. This resulted in a significant increase in burn, evidenced by a significant rise in cylinder temperature from prolonged operation. The number of prongs was then increased to five. Figure 4.4 shows the three prong plates used. The five prong plate was manually distorted after fabrication because it became apparent that the ignition module could not respond to the prongs at a closer separation as detected by the pulse meter (see Section 5.3). The five prong plate resulted in an increase in burn and the engine quickly reached temperatures comparable to conventional engines when operating under moderate load. The engine now significantly unloaded the motoring drill but was still not able to power itself.

![Figure 4.4 – Ignition prongs](image)
The inconsistent burn problem persisted until it was addressed by the addition of multiple sparks during the expected burn period. Of all the modifications employed, the addition of multiple sparks made the most significant improvement to the burn.

The engine was built with two spark plug ports on approximately opposite sides of the cylinder. The initial intention was to provide an option to assess the effect of the spark initiating burn from either one side or both sides of the narrow disk shaped combustion chamber. The length of the burn path was recognised as a potential impediment to efficiency.

All reference to ignition and burn are for both spark plugs firing at the same time. The reason multiple sparks resulted in a more complete burn could be related to the speed and direction of the gas flow as both pistons move past the spark plug ports. Prior to minimum separation of the pistons in the compressed positions the speed of the gas being ‘squished’ into the spark plug ports could exceed the flame speed in the charge. The very small combustion volume with intruding edges and complex fluid motion is likely responsible. A very poor burn was achieved in many configurations of ignition timing, mixture, engine speed and engine temperature prior to the addition of multiple sparks. As the overall burn is improved by subsequent sparks, it is clear that there is a flame propagation problem in the prototype configuration. The prospect for a complete burn is low even with multiple sparks because the probability of a combustible pocket of gas existing at the spark plug ports diminishes with each subsequent spark-combustion event.

4.2.10 Start Aid Always Required

The engine has very poor burn characteristics when cold even with all the previously mentioned modifications. A commercially available ether aerosol sprayed into the induction is required for the first several starting cycles. The readily combustible mixture created provides sufficient heat in the engine to then allow the standard
petrol mixture to burn. Engine preheating will also allow the petrol mixture to combust, but takes considerably longer than the use of the ether aerosol.

4.3 Conclusion

The prototype revealed that several characteristics of the engine outside those addressed by the Matlab model were significant influences over the performance of the engine. Manipulation of the prototype by various means improved the prototype’s burn characteristics, but failed to overcome what became apparent as inherent deficiencies in the configuration. To ascertain which performance criteria were contributing to the difference between the prototype performance and the Matlab model predictions, explanation was sort using engine performance measurement as described in the Chapter 5.
Chapter 5 Engine Performance Measurement

5.1 Introduction

Initial testing with the prototype revealed that the engine performance was significantly different from the Matlab model predicted performance. Initial ignition and burn problems were addressed to some degree as described in Chapter 4. Measurement of certain key performance parameters was the next step in identifying the causes of the discrepancies between the Prototype and Matlab model performance. This chapter gives detail of the measurement equipment, process and results. The test measurements are for the engine returned to the specifications obtained in the initial prototype build, but the use of five sparks over the combustion period was retained. The piston crowns were returned to their original shape by adding brazing material to the machined grooves described in Section 4.2.6

5.2 Flow and Torque Measurement

Even with the use of five sparks at each spark plug, which caused the largest improvement in burn of any of the manipulations described in Chapter 4, the engine still was unable to power itself. However, assessment of the engine performance could be made by measuring the reduction in the input power of the motoring electric drill when the engine ignition was switched on. The engine speed (and drill speed) was measured with a pulse meter (see Section 5.3). The drill power or work per cycle could be determined by measuring the input torque. A spring balance was employed to measure that torque. Figure 5.1 shows a photograph of the arrangement. Measurements on the inducted gas stream would determine the air flow and measurement of the fuel flow rate could then determine the mixture.
Figure 5.1 – Torque measurement using a spring balance

The photograph in Figure 5.2 show the engine in its test configuration with associated measuring equipment.
5.3 Engine Test Equipment

The following components of the test equipment were acquired or fabricated to suit the engine configuration:

- The inducted air flow was measured in the tube upstream of the carburettor using a Pitot tube coupled to an inclined, ethanol-filled manometer (Figure 5.3 and Figure 5.4).
- The fuel flow rate was measured by timing the fuel level drop in a vertical tube with internal diameter resulting in 55 mm/cm$^3$ (Figure 5.5).
- The motor torque measurement was via a spring balance over a 0.180 m arm (see Figure 5.1).
- A stop watch (Figure 5.6) was used for timing of the fuel flow rate.
- An automotive accessories pulse meter (Figure 5.6) was used to measure the engine speed by measuring the ignition pulses from the ignition module. A dial gave options for different numbers of cylinders in conventional multicylinder 4-stroke engines. The displayed speed was corrected by a calculation based on the number of prongs on the pick-up trigger, $N_{\text{pick-up}}$ and the dial setting on the meter, $N_{\text{dial}}$. Division by two accounted for the meter being configured to measure the pulses of a 4-stroke cycle, that is two revolutions per cycle.

$$\text{engine speed} = \frac{N_{\text{dial}}}{2 \times N_{\text{pick-up}}}$$  \hspace{1cm} (19)

- A compression pressure gauge (Figure 5.7) was arranged to screw into one of the spark plug ports. It was fitted with a small non-return valve in the tip to allow the compressed volume to be maintained while the pressure measurements were taken.
Figure 5.3 – Pitot tube upstream of carburettor

Figure 5.4 – Inclined tube manometer

Figure 5.5 – Fuel flow measured in vertical tube
5.4 Engine Performance Data

Table 5-1 shows data taken after the engine had operated for a total of approximately one hour in various conditions. Initial motoring torques were higher than those indicated in Table 5-1. This was mostly due to the running in of all engine components, especially the bearings and rings.
The data was measured to an accuracy that gives a general representation of the contributing effects. No error analysis was performed on the data. Future testing is planned to accurately specify the performance shown in Table 5-1.

### Table 5-1 – Motoring test data. Data from three tests performed over two days are presented.

<table>
<thead>
<tr>
<th>Test Date</th>
<th>Engine cond.</th>
<th>Upper case oil</th>
<th>Ignit.</th>
<th>Angle</th>
<th>Density (kg/m³)</th>
<th>L(mm)</th>
<th>Air Temp (°C)</th>
<th>Engine speed (kg)</th>
<th>Spring (mm)</th>
<th>Radius (mm)</th>
<th>∆h (mm)</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8-Aug am</td>
<td>Cold &lt;10 Low No 5 790 12 18 1050 3 0.18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hot High No 5 790 10 18 920 3.6 0.18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hot High Yes 5 790 6 18 1440 2.1 0.18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9-Aug pm</td>
<td>Cold Low No 3 790 20 20 1115 2.2 0.18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hot Low Yes 3 790 24 20 1400 1.5 0.18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5.5 Data Manipulation Calculations

5.5.1 Inducted Air Flow Rate

The deduction of the Pitot tube dynamic pressure from the inclined tube manometer with reservoir is routine and would be covered in most basic fluid mechanics texts (for example, Introduction to Fluid Mechanics [26]). The following equation was entered in Microsoft Excel to produce the results shown in Table 5-2 for the Pitot tube dynamic pressure (Pitot $\Delta p$). $\rho_i$ is the density of the manometer liquid.

$$\Delta p = \rho_i g L \left( \sin \theta + \left( \frac{d}{D} \right)^2 \right)$$  (20)
The dynamic pressure in the inducted flow stream is given by

\[ \Delta p = \text{dynamic pressure} = \frac{1}{2} \rho V^2 \]  

(21)

where \( \rho \) and \( V \) are the density and velocity of the flowing stream giving

\[ V = \sqrt{\frac{2\Delta p}{\rho}} \]  

(22)

A velocity profile correction factor of 0.9 was employed to produce an average flow stream velocity of

\[ V = \sqrt{\frac{2\Delta p}{\rho}} \times 0.9 \]  

(23)

The density of air was taken as an average, \( \rho = 1.2 \text{ kg/m}^3 \)

The induction tube diameter at the Pitot tube was 27.0mm giving the induction flow rate of

\[ Q_{\text{air}} = V \times \pi \left( 0.0135 \right)^2 \times 10^{-6} \text{ (cc/s)} \]  

(24)

dividing by the revolution/sec gives the inducted volume per cycle

\[ \forall_{\text{air/cycle}} = \frac{Q_{\text{air}}}{\left( \frac{\text{rpm}}{60} \right)} \]  

(25)

multiplying by the air density gives the inducted mass per cycle

\[ m_{\text{air/cycle}} = \rho \forall_{\text{air/cycle}} \]  

(26)

Torque is given by

\[ T = F_s \times R \]  

(27)
The fuel flow rate was measured by timing the fuel level drop in a vertical tube with internal diameter resulting in 55 mm/cm³, giving the mass of fuel per cycle as

\[ m_{\text{fuel/cycle}} = \frac{\Delta h}{55 \times \rho_{\text{fuel}}} \times \frac{\text{rpm}}{60} \times \rho_{\text{fuel}} \quad \left( \rho_{\text{fuel}} \text{ taken as } 0.72 \text{ (kg/m}^3\text{)} \right) \]  

(28)

The air/fuel ratio is given by

\[ \frac{m_{\text{air/cycle}}}{m_{\text{fuel/cycle}}} \]  

(29)

5.5.2 Thermodynamic Work

With the current prototype testing, the power drill dynamometer actually powers the engine at all times, but the thermodynamic work output of the engine can still be deducted from the average total work input calculated from the input torque for the engine being motored with ignition off less the work measured with the ignition on for each test. This represents the thermodynamic work if the friction losses are the same in both cases. This technique is used in conventional engine assessments based on the assertion that the friction losses are not significantly influenced by engine pressure.

\[ \text{Nett Work in per cycle } = T \times 2\pi \]  

(30)

\[ \text{Nett Thermodynamic Work per cycle } = \left( \text{average work in (ignition off) for the test} \right) - \left( \text{work in (ignition on)} \right) \]  

(31)
Table 5-2 – Motoring tests – derived results. Results from three tests performed over two days are presented.

Table 5-2 shows parameter values derived from the data presented in Table 5-1. The mixtures used in each test were set to the mixture that produced the highest engine speed with the ignition on. The closeness of the calculated air to fuel ratio (Table 5-2) to that of a typical optimum mixture indicates that the accuracy of the fuel and air flow measuring devices are acceptable. This implies the air flow rate results are probably reliable and thus the inducted volume per cycle is also reliably reported.
5.6 Comparison of Prototype Performance and Matlab Model Prediction

Table 5-3 provides a comparison of the Matlab model simulated results for the tested prototype configuration and the experimental results.

<table>
<thead>
<tr>
<th></th>
<th>Simulation</th>
<th>Prototype Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal Work (Thermo Work) (J)</td>
<td>23.5</td>
<td>11.5 (average)</td>
</tr>
<tr>
<td>Internal Efficiency (Thermo Eff) (%)</td>
<td>31.4</td>
<td></td>
</tr>
<tr>
<td>Net Work (J)</td>
<td>20.3</td>
<td>-19.7</td>
</tr>
<tr>
<td>Net Efficiency (%)</td>
<td>27.1</td>
<td></td>
</tr>
<tr>
<td>Inducted Volume (cm$^3$)</td>
<td>27.3</td>
<td>94 (average)</td>
</tr>
</tbody>
</table>

Table 5-3 – Comparison of simulation results and prototype test results

This engine configuration inherently requires a very large connecting rod inclination to produce the required piston motion. That large connecting rod angle preoccupied the considerations for friction losses because it was anticipated that the reaction loads would result in large piston friction loads. Data from Section 2.5 suggests that this concern was warranted and also suggests that the difference in friction loss between the motored engine and the engine without ignition and burn might be more significant than would be the case in a conventional engine. Consequently, the 11.5 J of thermodynamic work calculated from the engine test is expected to be conservative. In spite of this, there still remains a significant difference between the modelled thermodynamic work and the estimated thermodynamic work from the test.

The most revealing consequence of the test results is that independent of whether the thermodynamic cycle could or was making the 12 J of work shown to be in deficit, the friction losses are overwhelmingly large and show no correlation to the modelled friction losses. If the engine could be coaxed into doubling its thermodynamic work output, it could still not power itself, let alone produce any output power.
5.7 Compression Pressures

With the test pressure gauge fitted to one of the spark plug ports, Table 5-4 shows pressures recorded for the engine being motored in the configuration that produced the data of Chapter 5 (ignition off). Engine speeds were measured with a pulse meter connected to the ignition pickup.

<table>
<thead>
<tr>
<th>Motoring Speed (rpm)</th>
<th>Pressure (kPa)</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>120</td>
</tr>
<tr>
<td>500</td>
<td>150</td>
</tr>
<tr>
<td>600</td>
<td>150</td>
</tr>
<tr>
<td>700</td>
<td>180</td>
</tr>
<tr>
<td>800</td>
<td>200</td>
</tr>
<tr>
<td>860</td>
<td>225</td>
</tr>
</tbody>
</table>

Table 5-4 – Motored engine compression pressure without crankcase pump

Figure 5.8 shows a plot of the motored compression pressures listed in Table 5-4. The near linear relationship between the compression pressure and the engine speed together with the steep slope of the plot relative to that for a conventional engine.
strongly suggests that blow-by was very significant – in a conventional engine, the peak motoring pressure is a weak function of engine speed compared with Figure 5.8. An assessment of the cause of this excessive blow-by is given in Section 6.2.1

To confirm that the induction devices did not restrict the inducted volume, the test was repeated with the induction port open at the cylinder. The results are shown in Table 5-5 and Figure 5.9

<table>
<thead>
<tr>
<th>Cylinder open at the induction port</th>
<th>Motoring Speed (rpm)</th>
<th>Pressure (kPa)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>400</td>
<td>125</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>140</td>
</tr>
<tr>
<td></td>
<td>600</td>
<td>160</td>
</tr>
<tr>
<td></td>
<td>700</td>
<td>180</td>
</tr>
<tr>
<td></td>
<td>800</td>
<td>200</td>
</tr>
</tbody>
</table>

Table 5-5 – Compression pressures – induction devices removed

Tests represented by Figure 5.8 and Figure 5.9 confirmed that the induction devices were not impeding inducted volumes (at least up to 800 rpm). These tests also revealed that the compression pressure was significantly lower than the Matlab
model indicated. Inspection of the rings where they were visible at the spark plug port showed that moderate bedding had already occurred up to this point, but that degeneration in the rings sufficient to explain the excessive blow-by was not evident.

### 5.8 Component Friction

The prototype tests revealed that the Matlab model predicted thermodynamic work per cycle was not achieved by the prototype. Further investigation is required to assess the causes of thermodynamic work deficiencies. The tests revealed that the engine friction was much higher than the Matlab model predicted. One test was performed to identify where the model friction and prototype friction differ. The lower crankshaft was motored in its case without any other components attached. The motoring torque when adjusted to match the test speeds was measured as approximately 0.9 Nm, which represented about one third of the total motoring friction torque of the prototype test. This was for one crankshaft alone. The engine was further disassembled and the outer main bearings were discovered to be very ‘tight’, easily responsible for the majority of the measured friction on the crankshaft. Table 5-6 shows a comparison of the measured and Matlab model predicted friction.

<table>
<thead>
<tr>
<th>Engine Part</th>
<th>Total Measured input Work average for Tests 2 &amp; 3 @ 1100rpm ignition off. (J/cycle)</th>
<th>Measured Friction</th>
<th>Modelled Friction (J/cycle)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complete Engine</td>
<td>28</td>
<td></td>
<td>3.3</td>
</tr>
<tr>
<td>Bearings (crankshafts)</td>
<td>11</td>
<td></td>
<td>0.6</td>
</tr>
<tr>
<td>Bearings (other), Belt</td>
<td>0.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Piston Friction</td>
<td>0.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ring Friction</td>
<td>1.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pumping Losses</td>
<td>0.5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The engine motoring tests were performed with the throttle open. The pumping losses measured are necessarily higher than the modelled pumping losses, because the engine works against the sub-atmospheric pressure during the expansion stroke when the engine is not firing. The model lumped the losses for all the bearings and the belt together using a generic proportion of the total thermodynamic work based on data from conventional engines. If the crankshaft is allocated half that model losses figure, the test showed that model estimate was in error by a factor of
\[ \frac{11}{0.5 \times 0.6} \approx 35. \]
All other losses summed together were in error in the Matlab model by a factor of
\[ \frac{28 - 11}{3.3 - (0.5 \times 0.6)} \approx 6. \]

5.9 Conclusions

The measurement of the prototype performance revealed that the Matlab model predicted performance was not achieved by the prototype. The prototype could not achieve the thermodynamic cycle performance predicted by the model. The Matlab model friction analysis was shown to grossly underestimate the friction losses in the prototype as constructed. Chapter 6 reflects on the Matlab model initial assumptions and input parameters and reports on the additions to the Matlab model that allows it to reflect the measured prototype performance.
Chapter 6 Revised Matlab Model and Discussion

6.1 Introduction

Measurement of the prototype engine performance detailed in Chapter 5 showed that the original Matlab model did not correlate well with the prototype performance. This chapter addresses the failings of several of the features of the Matlab model and where appropriate re-evaluates the input parameters and assumptions of the model. In doing so, it also reports on the possible mechanisms responsible for the performance achieved by the prototype in relation to the differences with the Matlab model.

6.2 Features Reviewed in the Matlab Model

The Matlab model used to calculate the engine performance consists of basically two parts. The first calculates the thermodynamic work returned from the cycle based on the engine simulation portion of the program. The second addresses the friction work used by the engine. The following features of the Matlab simulation model were reviewed.

<table>
<thead>
<tr>
<th>Model Feature</th>
<th>Original included in the Matlab Model</th>
<th>Justification for Review</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blow-by</td>
<td>Yes</td>
<td>Significantly lower measured compression pressure relative to modelled pressure</td>
</tr>
<tr>
<td>Spark plug port volume</td>
<td>Yes</td>
<td>Manipulation of the model by reducing the piston separation at maximum compression showed compression pressures far higher then possible if the model was accurately including the spark plug port volumes.</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----</td>
<td>---------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Ring Friction</td>
<td>Yes</td>
<td>All friction showed no correlation with the original Matlab model</td>
</tr>
<tr>
<td>Piston friction</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Bearing friction</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Belt friction</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>

Table 6-1 – Matlab model features reviewed

Review of the parameters controlling these features of the Matlab model was achieved by changing the values originally set in the engine specifications data file, ‘enginedata’. The Matlab files of Appendix B are the modified versions that reflect the changes prompted by the prototype test results. To allow the original Matlab program to operate, the original code is included as an option prompted in the main file, ‘slider_bar_input’. Selecting 'ORIGINAL', 'CORRECTED THERMO-CYLE' or 'CORRECTED THERMO-CYCLE & FRICTION' from the screen prompt is used to include or exclude the modified parameters at the initiation of the main Matlab script. An addition to the script also allows the engine speed for the simulation to be input.

### 6.2.1 Blow-by

The near linear relation between the compression pressure and the engine speed and the slope of the plot shown in Figure 5.8 strongly suggests that blow-by was very significant. The arrangement of the rings described in Section 3.16.1 was a necessary consequence of the pistons passing the spark plug ports. The spark plug ports exposed the upper cylinder upper ring directly to the compression pressure at mid
compression stroke. Ring end gap was probably excessive, motivated by an expectation that the rings would be exposed to higher than normal temperatures because of the use of steel as the piston material. Steel was used because appropriate aluminium wrought bar suitable to machine the pistons was not available. This could have also resulted in a poorer seal between the ring and the hard steel groove land than would be achieved with a softer aluminium piston. Significantly, the compressed volume is exposed to ring seals on two sides unlike a conventional engine. The piston diameter is also very large relative to the compressed volume. This exposes more ring seal length to the engine pressure than in a conventional engine. In total, the blow-by would have significantly contributed to a reduction in the net charge available to the thermodynamic cycle.

The Matlab model considers gas loss via blow-by. It is estimated using a first order system approximation with a user-defined rate-constant. The original model doubled the blow-by resulting from the constant used in the original Ferguson [9] engine simulation program. The original decision to double that blow-by quantity was in error based on the large ring seal length relative to compressed volume of the opposed piston engine configuration. The added gas loss resulting from the motion of the pistons past the spark plug port and the mechanical features of the ring/piston interface mentioned in the previous paragraph all suggest that the blow-by rate constant would better reflect the performance of the prototype if modified by a factor several times larger than the original estimate.

Another aspect of the opposed piston configuration that could contribute to the increased blow-by is the fact that one of the pistons is retreating from the compressed volume during the compression stroke. The ring friction in that scenario would tend to reduce the pressure between the ring and the ring groove land because the gas pressure force and the friction force oppose each other. This could result in excessive blow-by around the ring as described in Figure 6.1.
Figure 6.1 – Possible elevated blow-by mechanism

If the ring friction force is in the same order of magnitude as the pressure force, the ring is able to ‘float’ in its groove and no positive seal is possible between the ring and the piston. In a conventional engine, the pressure is high (post combustion) when the friction force and pressure force oppose each other as in Figure 6.1, eliminating this problem.

The Matlab model addresses blow-by in the ‘Rates’ functions called by the ‘ahrind’ function. The Ferguson [9] simulation model used three ‘Rates’ functions for the three phases of the cycle, compression, combustion and expansion. The Matlab model used for the opposed piston engine simulation needed to address the change in volume as the spark plug ports were exposed. It consequently required more ‘Rates’ functions. ‘RatesComp1’ is an example of those functions. The blow-by variable, Cblowby is called in line 30:

\[
\text{mass} = \text{mass1} \times (\exp(-\text{Cblowby} \times (\theta - \theta1)/\omega))^{2};
\]

A reasonable estimate of the value for Cblowby can be obtained from the model by trial and error until the model produces the compression pressure achieved in the prototype. The original value of 0.8 (s\(^{-1}\)) was altered to 8.0 (s\(^{-1}\)) in the modified format. This resulting in the 98% of the inducted mass used for combustion at 1000
rpm in the original version being reduced to 83%. Figure 6.2 represents a comparison between the engine pressures resulting from the original value of the blow-by coefficient (part a of the figure) and the modified value of the blow-by coefficient (part b). All other simulation parameters in Figure 6.2-b match those of Figure 6.2-a and no other model modifications are employed. The prototype engine specifications are used and modelled at 800 rpm to reflect the motored prototype measured performance.

Figure 6.2 – a - Original Matlab model predicted engine pressure

Figure 6.2 – b – Matlab model predicted engine pressure with modified blow-by constant
The erroneous blow-by constant resulted in a significant overestimate of thermodynamic work in the original Matlab model.

6.2.2 Spark Plug Port Volume

The intention of the original model was to include the effect of the spark plug ports being exposed during the compression stroke. However, logic errors resulted in a critical conditional statement never being met so that the pressure at the end of the compression phase did not actually accommodate the extra volume of the spark plug ports. This resulted in an artificially high pressure before the integration of the combustion equations commenced. The integration of the governing combustion equations did actually include the spark plug volume, but the error introduced during the compression phase created too high initial conditions for the combustion phase and produced erroneous results. Figure 6.3 shows the comparison between the engine pressures of the corrected version of the ‘ahrind’ file that correctly includes the spark plug port volumes in the compression phase against the results with the original erroneous file with all other simulation parameters matching and no other model modifications are employed. The prototype engine specifications as manufactured are used and modelled at 1400 rpm to reflect the prototype speed with ignition on during the tests.
Figure 6.3 – a – Original Matlab model predicted engine pressure

Figure 6.3 – b – Modified Matlab model including spark-plug port volume

Figure 6.3-b shows a discontinuity at about 105° corresponding to crank angle where the spark plug ports are first exposed. The resulting maximum pressure achieved in the engine is significantly less than when the model only included the spark plug
port volumes from the burn phase onwards. The result was an overestimate of thermodynamic work in the original Matlab model.

### 6.2.3 Blow-by and Spark Plug Port Volume – Combined Effects

When both the adjusted blow-by rate and the corrected spark plug port volume calculation were included in the Matlab simulation, the resulting reduction in maximum engine pressure is significant. Figure 6.4 shows the effect on the prototype specifications run at 1400 rpm for both of these thermodynamic model changes.

![Engine Pressure Vs Crank Angle after E.C.](image)

**Figure 6.4 – a** - Original Matlab model predicted engine pressure
Figure 6.4 – b – Revised Matlab model engine pressure prediction

Figure 6.4 b shows the extreme sub-atmospheric pressure at the opening of the exhaust port resulting in extra pumping work during the cycle. Table 6-2 displays the comparison between the original and revised thermodynamic cycles reflected in two output criteria.

<table>
<thead>
<tr>
<th>Thermodynamic Cycle Model Version</th>
<th>Internal Work (J)</th>
<th>Maximum Engine Pressure (kPa)</th>
</tr>
</thead>
<tbody>
<tr>
<td>As applied to original analysis</td>
<td>22.7</td>
<td>2000</td>
</tr>
<tr>
<td>Modified blow-by constant with spark plug port volume included in compression model</td>
<td>15.4</td>
<td>1100</td>
</tr>
</tbody>
</table>

Table 6-2 – Comparison of internal thermodynamic work and maximum pressure for the original and revised Matlab models

The results displayed in Table 6-2 show that the Matlab model can readily reflect the performance measured from the prototype and this suggests that the model is basically sound. The overestimates in the original modelling parameters produced an
unrealistically high thermodynamic cycle work that was used to justify continuing with the prototype build. On reflection a higher estimate of the blow-by rate constant was warranted, but a reasonable value could not be ascertained without the experimental results obtained from the prototype test.

### 6.2.4 Ring Friction

The original Matlab model estimated the ring friction for each piston, but failed to include the total number of rings for the passive effect of ring tension and only applied the dynamic effect of engine pressure to the top ring of each piston. Blow-by would have resulted in higher ring friction due to higher pressures on outer rings than initially estimated by the model. The result being an underestimate of ring friction in the original model. Certainly the magnitude of the discrepancy between the measured engine friction and the friction predicted by the model (see Section 5.6) leaves scope to speculate that the ring friction is underestimated by the model. Further testing of the prototype in the future to compare engine friction with and without the rings fitted would provide some basis for readdressing the ring friction model. Without further analysis via prototype testing, refining the ring friction model is not possible. However, the author judges that the ring friction model is not a significant contributor to the poor correlation between the Matlab model predicted friction and the prototype measured friction. This judgement is based on the low percentage of total friction that is attributed to rings in conventional engines and the large proportion that other identified friction contributed to the prototype measured friction (see Section 6.2.5 to 6.2.7).

### 6.2.5 Piston Friction

This engine configuration inherently requires a very large connecting rod inclination to produce the required piston motion. That large connecting rod angle preoccupied
the considerations for friction losses because it was anticipated that the reaction loads would result in large piston friction loads. Table 5-3 and Figure 3.9 confirm this concern was warranted. The large connecting rod inclinations also mean that as a result of higher engine pressure from ignition and burn, the piston normal reaction force and consequently friction would be subject to a greater increase than the effects in a convention engine with low connecting rod inclination. Consequently, the thermodynamic work calculated from the engine test would be conservative because it relies on the engine friction remaining constant between the ignition off and ignition on motored torques.

In the original modelling it was assumed that the largest potential for error in the prediction of the friction losses would come from the high normal reaction loads due to the high connecting rod inclination angles. Consequently this was modelled thoroughly (see Section 2.5), but no direct assessment of its accuracy is possible from the test results to this time. The total discrepancy between the measured engine friction and the friction predicted by the model (see Section 5.8) leaves scope to speculate that the piston friction is underestimated by the model. However, on inspection of the engine components after approximately two hours of engine operation in various conditions there were no indications of excessive wear in any engine components. An initial oil change revealed some metallic debris in the crankcases after the initial attempts to start the engine. Subsequent inspection of the crankcase oil shows no additional metallic debris and no oil discolouration. The cylinder hone marks are consistent with those of a conventional engine after initial running in. It should be remembered that the engine has operated at very low engine pressure loads due to low compression ratio and excessive blow-by. Therefore, the piston friction model is not likely to be contributing to the error in the Matlab friction model.
6.2.6 Bearing Friction

As reported in Section 5.8, each pair of main crankshaft bearings are contributing about one third of the total engine motoring torque at 1100rpm with the throttle open. That is, the crankshaft bearings account for two thirds of the total engine motoring torque at 1100 rpm. This stands out as the single most deficient facet of the model. The author investigated bearing losses for the original Matlab model using SKF Interactive Engineering Catalogue [28]. However, the lack of specific reference data led to that bearing friction model option being discarding.

The crankshaft bearing sizes were similar to those of an engine that matched the piston size and consequently the anticipated maximum piston forces. The original Matlab model used a generic proportion of total engine losses as the criteria for the net bearing losses but applied it to the total thermodynamic output and not to the thermodynamic output from an engine using similar sized components. That proportion was calculated from published data in Ferguson [9] and Stone [12]. Conventional engines using components of the size used in the concept engine develop about 3 kW for the speed at which the prototype was tested. The thermodynamic cycle model indicated that the prototype would produce about 23 J of work at 2000 rpm or about 750W. This deficiency of a factor of four was a significant error and is the biggest contributor to the underestimate in total friction.

An idler tensioner was also required in the belt drive, and this was not originally included in the model. The omission of the belt tensioner bearing was an oversight that reduced the ability of the Matlab model to predict the engine performance. The idler pulley is 30 tooth and the crank pulleys are 36 tooth resulting in the idler bearing operating at a higher speed than the crankshafts magnifying the need to include that bearing.

On inspection of the main crankshaft and idler bearings, the preload due to bearing fit was responsible for a large variation in the bearing resistance. The crankshaft outer bearings were excessively tight without being fitted to the crankcases. This was
assumed to be a design feature in the original application for which they were intended that would prolong the outer crankshaft bearing’s life. The rolling element bearings in the connecting rods showed no resistance.

Attention to spigot dimensions and the choice of larger tolerance bearings would result in reduced friction in the prototype. For future work on the prototype, the author recommends that the rolling element ball bearings be assessed and optimised to reduce friction. The author judges that the majority of the current bearing friction, and hence the majority of the current engine friction overall, could be removed with appropriate work on the main crankshaft bearings and idler bearing.

6.2.7 Belt Friction

The synchronous belt drive was selected using the Gates selection software DesignFlex2K. Gates [27] report that equivalent drives can operate at 98% efficiency. The original Matlab model was again in error in that it applied the belt efficiency to the thermodynamic cycle output and not to the total work to which the belt was exposed which includes the friction work of the engine in addition to the thermodynamic work. The inherent fluctuating load of the belt drive could also reduce the claimed efficiency. Belt preload would also have an effect on the belt drive efficiency. Further testing of the prototype could identify where reduction in losses for the belt drive could be obtained. The original Matlab model underestimated belt drive losses.

6.3 Conclusion

The original Matlab model had the capacity to accurately predict the performance of many aspects of the opposed piston engine. All the chosen parameters and modelling techniques proved to create a model which predicted performance characteristics for
the engine significantly higher than those achieved in the prototype. The thermodynamic model was able to be modified to more accurately reflect the thermodynamic performance of the prototype. The model is unable to predict what proportion of the inducted charge is burnt in the cycle, therefore cannot accurately predict the thermodynamic work for the cycle where incomplete burn occurs. The friction losses model significantly underestimated the total engine friction. The Matlab model has the potential to accurately reflect those friction losses if suitable empirical data can be obtained from the prototype. The following chapter addresses the appropriateness of the original Modelling decisions and their implications.
Chapter 7 Conclusions

7.1 Introduction

The original goal of the project was achieved in that the opposed piston engine concept was assessed from the performance of a prototype. The computer modelling used to originally determine expected performance criteria of the engine was shown to be deficient in the choice of modelling parameters but is assessed as having captured the essential physics such that reliable results are produced when the simulations are modified as described in Chapter 6.

7.2 The Matlab Model Performance

It should be remembered that the prototype tested had specifications which produced potential performance figures significantly lower than the optimum identified from the simulations in the original Matlab model that reflected the original thermodynamic cycle concept. A succession of changes to the engine specifications was required because of the following:

- The original optimum specification obtained from the Matlab model was unable to be physically constructed due to limitations of the moving parts. The changes required to enable the construction of the engine necessarily detracted from the initially predicted thermodynamic performance figures.
- Manufacturing errors reduced the physical performance specifications of the prototype engine. In particular, the designed compression ratio was not achieved. The specifications achieved in the prototype again detracted from the performance figures predicted by the original Matlab model.
The prototype test revealed significant deficiencies in the Matlab model as addressed in Chapter 6. The main features of the Model exposed as in error by the test were:

- The thermodynamic performance was impeded by excess blow-by and the unconventional shape of the combustion chamber together with complex gas motion resulting from the motion of both pistons at ignition resulted in less than complete burn.
- The engine friction was significantly higher than the Matlab modelled friction

7.3 Discussion

The prototype testing to date showed that the design concept, although capable of replicating the intended thermodynamic cycle, is unlikely to produce a functional engine with the efficiency initially predicted by the Matlab model. The Matlab model was revised from the prototype test (see Chapter 6) which allowed specification of modified parameters and minor changes to the model structure. The revised Matlab model indicates that a functional engine of the design presented is possible, but could not operate at the efficiency initially predicted. The revised model predicts the engine would also operate at an efficiency lower than a conventional engine.

Modification to the Matlab model described in Chapter 6 result in thermodynamic cycle outputs that are in general agreement with the measured thermodynamic cycle output from the prototype test. Further testing is required to identify a more precise value of the thermodynamic cycle performance in the prototype. In particular, modification to the engine to achieve the compression ratio designed for the prototype is anticipated to improve test results.

The general soundness of the Matlab model once modified by information obtained from the prototype tests gives confidence in its ability to accurately predict the performance of the original design thermodynamic cycle. Because the model shows
improved thermodynamic efficiency for the full expansion cycle in its optimum configuration relative to a conventional cycle, the thermodynamic cycle full expansion aspect of the original design is assessed as being valid.

The inducted volume per cycle on the prototype test was far higher than the engine’s simulated or geometrically-identified induction displacement (Table 5-3). The intended design had a simulated exhaust opening pressure of 90 kPa (absolute), which is the pressure at the end of the power stroke. However, the fabrication errors in the tested prototype resulted in a substantially sub-atmospheric simulated exhaust opening pressure of 50 kPa (absolute). Once the excessive blow-by and incomplete burn (see Section 4.2.9) achieved in the prototype test is taken into account, the expanding gas in the power stroke cannot fill the expanded volume of the engine and the engine consequently inducts more new charge into the cylinder while the engine is expanding past the induction port. The charge inducted during the expansion stroke accounts for the measured inducted volume. The original modelling showed that a higher than atmospheric pressure would be maintained over the induction port during the power stroke, thereby keeping the reed valve at the induction port closed and separating the two parts of the cycle. A volume around three times the induction displacement is inducted per cycle in the prototype test (Table 5-3). From the measured volume of inducted gas (see Section 5.4), it can be concluded that the burnt gas expands to atmospheric pressure during the power stroke in a volume similar to the induction displacement. This means that the engine has extremely poor ring seal characteristic and is likely to be burning less than the full charge. This easily accounts for the low thermodynamic cycle work values obtained in the prototype tests.

At present the thermodynamic model shows some promise as a tool for simulating variations to conventional engine thermodynamic cycles. The thermodynamic cycle concept originally conceived has potential, but the prototype test shows that the physical concept used to achieve the thermodynamic cycle does not make use of that cycle in an optimal fashion.
7.4 Potential Further Applications

The creation of a positive induction valve as originally conceived (see Section 3.7) will allow assessment of the completeness of burn via exhaust gas analysis as no new charge will be inducted during the power and/or exhaust stroke. This will allow the proportion of gas burnt in the cycle to be experimentally determined leading to clarification of the burn characteristics in the configuration. Consequently, separation of the blow-by effect and the burn completeness would be obtained and allow for better analysis of possible thermodynamic cycle performance and improvement. Subsequently, an accurate evaluation of the thermodynamic model would result.

Further work and analysis on the engine is anticipated with the goal of addressing the friction and in particular addressing friction errors in the model. Several features of the prototype were exposed as contributing excessive friction (see Chapter 6). Technical work on reducing the friction in those features will allow for a more accurate determination of performance of other aspects of the Matlab model. A good potential exists for the Matlab model to be modified to the extent that it could accurately predict the performance of the prototype under any conditions. Confirmation of the validity of the Matlab model will only be achieved if the model can reflect the performance of the prototype in various configurations. A new set of specifications with sufficiently varied performance characteristics will require the production of new components. For example, changing the piston mass will result in a change in piston friction due to altered normal reaction forces created from a different acceleration force. The Matlab model validity would be confirmed if it can successfully reflect the changed performance in the engine with such changed components.
7.5 Summary of Chapter 7 Conclusions

The Matlab modelling used to predict the performance of the concept engine was deficient in several facets. Not all those aspects were foreseeable and were only evident with information from the prototype test. The project methodology was shown to be complete and allowed for valid conclusions about the original design concept and the modelling employed to justify and produce the prototype. Future work could address the thermodynamic deficiencies and reduce friction, but the engine currently shows little potential as a viable design alternative to conventional engines. The concept engine could form the basis for further research into the advantages of the thermodynamic cycle it was created to employ. The Matlab model and thermodynamic cycle have positive attributes as quantified by the engine test. They may have other future applications.
References


[8] http://www.powerhousemuseum.com/australia_innovates/?behaviour=view_article&Section\_id=1020&article\_id=10041


Appendix A – Project Specification

University of Southern Queensland

FACULTY OF ENGINEERING AND SURVEYING

ENG4111/4112 Research Project
PROJECT SPECIFICATION

FOR: Ray Malpress

TOPIC: A new internal combustion engine configuration -
opposed piston with crank offset

SUPERVISOR: Assoc Prof David Buttsworth

SPONSORSHIP: USQ, FOES

PROJECT AIM: The project aims to assess the performance of an original internal combustion engine configuration which is markedly different from conventional engines. This new engine employs complete expansion and other mechanical innovations which are anticipated to improve efficiency.

PROGRAMME: Issue A, 26 March 2007

1. Research past applications of crank offset, thermodynamics, friction losses and computer simulation of efficiency of IC engines.
2. Write new and/or make use of existing Matlab engine simulation programmes to calculate the efficiency of IC engines including losses.
3. Configure a suitable user interface to allow easy variation of specifications to optimise engine efficiency.
4. Model the engine in ProE.
5. Analyse the stress in critical components with Ansys.
6. Fabricate the engine to a standard that will allow an appraisal of the predicted efficiency.
7. Test the engine on a dynamometer to produce performance curves.
8. Appraise the correlation or otherwise between predicted efficiency and measured efficiency.
9. Appraise the potential for the engine to be used in automotive and other applications.
A function softcopy of the Matlab model is contained on the submitted Dissertation CD.

Matlab engine simulation program – Structure Chart
Matlab Model Files

The full list of files used in the Matlab model follows.

Files with the same or similar name as files in the following list of Matlab files for the engine simulation program by Buttsworth [10] contain some of or all the same code as the original Buttsworth engine simulation model.

A function softcopy of the Matlab model is contained on the submitted Dissertation CD.
The Matlab files are printed below.

ahrind.m

function ahrind(splugport)
    global thetas thetab omega...
    heattransferlaw hcu hcb...
    Tw theta1 Vtdc Vbdc mass1...
    p1 T1 V1 minV maxinV maxexV...
    cylindervolume crankoffset conrod...
    crankthrow uppercrankoffset upperconrod...
    uppercrankthrow cranktocrank exportheight...
    uppercranklag thetacorrected maxinvolindex...
    maxexpandvolindex thetaignition upstroke downstroke...
    b stroke eps r Cblowby f fueltype airscheme phi...
    compressedvolindex plotindex portopenindex...
    upper_pist_positionmm lower_pist_positionmm splugvolume...
    thetaaddsplug Vtdc dtheta
    thetaaddsplugindex=max(find(upper_pist_positionmm(1:compressedvolindex)<max(lower_pist_positionmm)-splugport));
    thetaaddsplug=thetaaddsplugindex/180*pi;
    global thetas thetab omega...
    heattransferlaw hcu hcb...
    Tw theta1 Vtdc Vbdc mass1...
    p1 T1 V1 minV maxinV maxexV...
    cylindervolume crankoffset conrod...
    crankthrow uppercrankoffset upperconrod...
    uppercrankthrow cranktocrank exportheight...
    uppercranklag thetacorrected maxinvolindex...
    maxexpandvolindex thetaignition upstroke downstroke...
    b stroke eps r Cblowby f fueltype airscheme phi...
    compressedvolindex plotindex portopenindex...
    upper_pist_positionmm lower_pist_positionmm splugvolume...
    thetaaddsplug Vtdc dtheta
    thetaaddsplugindex=max(find(upper_pist_positionmm(1:compressedvolindex)<max(lower_pist_positionmm)-splugport));
    thetaaddsplug=thetaaddsplugindex/180*pi;
    %
    thetaaddsplug=thetaaddsplugindex/180*pi;
    thetaaddsplugindex=max(find(upper_pist_positionmm(1:compressedvolindex)<max(lower_pist_positionmm)-splugport));
%load the engine parameters and initial conditions
    enginedata
    switch heattransferlaw
        case 'Woschni'
            if (abs(hcu) > 10)|(abs(hcb) > 10),
                warning('Woschni model with weighting factor > 10')
            end
    end
    % integration parameters
    dtheta=1*pi/360;
    options=odeset('RelTol',1e-3);
    % integration during first stage of compression phase
    disp(['integrating over the compression phase']);
    if maxinvolindex/180*pi<thetaaddsplug
        [thetacomp,pTuWQlHl]=ode45('RatesCompO1',...
        [maxinvolindex/180*pi:dtheta:thetaaddsplug],[p1 T1 0 0 0],options);
    % specification of initial conditions at start of combustion phase
    % b - beginning of combustion
pb1 = interp1(thetacomp, pTuWQlHl(:,1), thetaaddsplug);
Tub1 = interp1(thetacomp, pTuWQlHl(:,2), thetaaddsplug);
Wb1 = interp1(thetacomp, pTuWQlHl(:,3), thetaaddsplug);
Qlb1 = interp1(thetacomp, pTuWQlHl(:,4), thetaaddsplug);
Hlb1 = interp1(thetacomp, pTuWQlHl(:,5), thetaaddsplug);

[thetacomp2,p2TuWQlHl] = ode45('RatesCompO2', ...
[thetaaddsplug: dtheta: thetaignition], [pb1 Tub1 Wb1 Qlb1 Hlb1], options);

pb = interp1(thetacomp2, p2TuWQlHl(:,1), thetaignition);
Tub = interp1(thetacomp2, p2TuWQlHl(:,2), thetaignition);
Tbb = Tadiabatic(pb, Tub, phi, f, fueltype, airscheme);
Wb = interp1(thetacomp2, p2TuWQlHl(:,3), thetaignition);
Qlb= interp1(thetacomp2, p2TuWQlHl(:,4), thetaignition);
Hlb= interp1(thetacomp2, p2TuWQlHl(:,5), thetaignition);

else
[thetacomp, pTuWQlHl] = ode45('RatesCompO2', ...
[thetaaddsplug: dtheta: thetaignition], [p1 T1 0 0 0], options);

% specification of initial conditions at start of combustion phase
% b - beginning of combustion
pb= interp1(thetacomp, pTuWQlHl(:,1), thetaignition);
Tub= interp1(thetacomp, pTuWQlHl(:,2), thetaignition);
Tbb= Tadiabatic(pb, Tub, phi, f, fueltype, airscheme);
Wb= interp1(thetacomp, pTuWQlHl(:,3), thetaignition);
Qlb= interp1(thetacomp, pTuWQlHl(:,4), thetaignition);
Hlb= interp1(thetacomp, pTuWQlHl(:,5), thetaignition);

end

% integration during combustion phase
disp(['integrating over the combustion phase']);
[thetacomb, pTbTuWQlHl] = ode45('RatesComb', ...
[thetaignition: dtheta: thetaignition+theta], [pb Tub Wb Qlb Hlb], options);

% specification of initial conditions at start of expansion phase
% e - end of combustion / start of expansion
pe= interp1(thetacomb, pTbTuWQlHl(:,1), thetaignition+theta);
Tbe= interp1(thetacomb, pTbTuWQlHl(:,2), thetaignition+theta);
We= interp1(thetacomb, pTbTuWQlHl(:,3), thetaignition+theta);
Qle= interp1(thetacomb, pTbTuWQlHl(:,4), thetaignition+theta);
Hle= interp1(thetacomb, pTbTuWQlHl(:,5), thetaignition+theta);

% integration during expansion phase
disp(['integrating over the expansion phase']);
[thetexp, pTbWQlHl] = ode45('RatesExp', ...
[thetaignition+theta: dtheta: portopenindex/180*pi], [pe Tbe We Qle Hle], options);

% error checks
mass4 = mass1 * exp(-Cblowby*2*pi/omega);
p4 = interp1(thetaexp, pTbWQlHl(:,1), portopenindex/180*pi);
T4 = interp1(thetaexp, pTbWQlHl(:,2), portopenindex/180*pi);
W4 = interp1(thetaexp, pTbWQlHl(:,3), portopenindex/180*pi);
Q4 = interp1(thetaexp, pTbWQlHl(:,4), portopenindex/180*pi);
H4 = interp1(thetaexp, pTbWQlHl(:,5), portopenindex/180*pi);

[f4, u4, v4, s4, Y4, cp4, dlvlT4, dlvlp4] = ...
  farg(p4, T4, phi, 1, fueltype, airscheme);
U4 = u4 * mass4;
error1 = 1 - v4 * mass4 / maxexV;
error2 = 1 - W4 / (U4-U1+Q4+H4);

% indicated mean effective pressure and thermal efficiency
imep = W4 / (pi*b*2/4 * max(crankthrow, uppercrankthrow)*2); % imep not accurately defined for this configuration
eta = W4 / mass1 * (1 + phi * 0.06548 * (1 - f) / phi / 0.06548 * (1 - f)) / 47870 / 1e3;
% calculate the heat flux in W/m^2
if isempty(thetacomp2)
    qcomp=calcq(thetacomp,pTuWQlHl,'comp'); % compression
    qcombu=calcq(thetacomb,pTbTuWQlHl,'combu'); % combustion-unburned zone
    qcombb=calcq(thetacomb,pTbTuWQlHl,'combb'); % combustion-burned zone
    qexp=calcq(thetaexp,pTbWQlHl,'exp'); % expansion
else
    qcomp=calcq(thetacomp,pTuWQlHl,'comp'); % compression
    qcomp2=calcq(thetacomp,p2TuWQlHl,'comp'); % compression
    qcombu=calcq(thetacomb,pTbTuWQlHl,'combu'); % combustion-unburned zone
    qcombb=calcq(thetacomb,pTbTuWQlHl,'combb'); % combustion-burned zone
    qexp=calcq(thetaexp,pTbWQlHl,'exp'); % expansion
end

timefinish=cputime;
timetaken=timefinish-timestart;

% save all data

clear ahrind.mat

save ahrind.mat

clear

ahrind2.m

function ahrind2(splugport)

% Script file to determine the performance of a fuel inducted engine
% based on a (user-specified) arbitrary heat release profile as a
% function of crank angle.
% Method closely follows that of:
% ********************************************************************
% input:
% enginedata.m - this is another script file that defines all of the
% relevant engine parameters and operating conditions.
% output:
% ahrind.mat - this file contains all of the variables. For plotting
% the results, see the example script file plotresults.m
% ********************************************************************

timestart=cputime;

global thetas thetab omega...
heattransferlaw hcu hcb...
Tw theta1 Vtdc Vbdc mass1...
p1 T1 V1 minV maxinV maxexV...
cylindervolume crankoffset conrod...
crankthrow uppercrankoffset upperconrod...
uppercrankthrow cranktocrank exportheight...
uppercranklag thetacorrected maxinvolindex...
maxexpandvolindex thetaiagnition upstroke downstroke...
b stroke eps r Cblowby f fueltype airscheme phi...
compressedvolindex plotindex portopenindex...
upper_pist_positionmm lower_pist_positionmm splugvolume...
thetaaddsplug Vtdc dtheta

% determine when spark plug ports are exposed

thetaaddsplugindex=max(find(upper_pist_positionmm(1:compressedvolindex)<max(lower_pist_positionmm)-splugport));
thetaaddsplug=thetaaddsplugindex/180*pi;
clear uppercranklag

% load the engine parameters and initial conditions
enginedata
switch heattransferlaw
    case 'Woschni'
        if (abs(hcu) > 10)||(abs(hcb) > 10),
            warning('Woschni model with weighting factor > 10')
        end
    end

% integration parameters
    dtheta=1*pi/360;
    options=odeset('RelTol',1e-3);
% integration during first stage of compression phase
disp(['integrating over the compression phase']);

    if maxinvolindex/180*pi<thetaaddsplug
        [thetacomp,pTuWQlHl]=ode45('RatesComp1', ...
            [maxinvolindex/180*pi:dtheta:thetaaddsplug],[p1 T1 0 0 0].options);
    end

% specification of initial conditions at start of combustion phase
    pb=interp1(thetacomp,pTuWQlHl(:,1),thetaaddsplug);
    Tub=interp1(thetacomp,pTuWQlHl(:,2),thetaaddsplug);
    Wb=interp1(thetacomp,pTuWQlHl(:,3),thetaaddsplug);
    Qlb=interp1(thetacomp,pTuWQlHl(:,4),thetaaddsplug);
    Hlb=interp1(thetacomp,pTuWQlHl(:,5),thetaaddsplug);

    [thetacomp2,p2TuWQlHl]=ode45('RatesComp2', ...
            [thetaaddsplug:dtheta:thetaaddsplug+dtheta],[pb Tub Qlb Hlb].options);

% specification of initial conditions at start of next compression phase
    pb=interp1(thetacomp2,p2TuWQlHl(:,1),thetaaddsplug+dtheta);
    Tub=interp1(thetacomp2,p2TuWQlHl(:,2),thetaaddsplug+dtheta);
    Wb=interp1(thetacomp2,p2TuWQlHl(:,3),thetaaddsplug+dtheta);
    Qlb=interp1(thetacomp2,p2TuWQlHl(:,4),thetaaddsplug+dtheta);
    Hlb=interp1(thetacomp2,p2TuWQlHl(:,5),thetaaddsplug+dtheta);

    [thetacomp3,p3TuWQlHl]=ode45('RatesComp3', ...
            [thetaaddsplug+dtheta:thetaignition],[pb Tub Wb Qlb Hlb].options);

% specification of initial conditions at start of next compression phase
    pb=interp1(thetacomp3,p3TuWQlHl(:,1),thetaignition);
    Tub=interp1(thetacomp3,p3TuWQlHl(:,2),thetaignition);
    Tbb=Tadiabatic(pb,Tub,phi,f,fueltype,airscheme);
    Wb=interp1(thetacomp3,p3TuWQlHl(:,3),thetaignition);
    Qlb=interp1(thetacomp3,p3TuWQlHl(:,4),thetaignition);
    Hlb=interp1(thetacomp3,p3TuWQlHl(:,5),thetaignition);

else
    [thetacomp,pTuWQlHl]=ode45('RatesComp3', ...
            [thetaaddsplug:dtheta:thetaignition],[p1 T1 0 0 0].options);
    thetacomp
% specification of initial conditions at start of combustion phase
    pb=interp1(thetacomp,pTuWQlHl(:,1),thetaignition);
    Tub=interp1(thetacomp,pTuWQlHl(:,2),thetaignition);
    Tbb=Tadiabatic(pb,Tub,phi,f,fueltype,airscheme);
    Wb=interp1(thetacomp,pTuWQlHl(:,3),thetaignition);
    Qlb=interp1(thetacomp,pTuWQlHl(:,4),thetaignition);
    Hlb=interp1(thetacomp,pTuWQlHl(:,5),thetaignition);

    thetacomp3=[ ];
end

% integration during combustion phase
disp(['integrating over the combustion phase']);
    [thetacomp,pTbTuWQlHl]=ode45('RatesComb', ...
% specification of initial conditions at start of expansion phase
% e - end of combustion / start of expansion
pe=interp1(thetacomb,pTbTuWQlHl(:,1),thetaignition+thetab);
Tbe=interp1(thetacomb,pTbTuWQlHl(:,2),thetaignition+thetab);
We=interp1(thetacomb,pTbTuWQlHl(:,4),thetaignition+thetab);
Qle=interp1(thetacomb,pTbTuWQlHl(:,5),thetaignition+thetab);
Hle=interp1(thetacomb,pTbTuWQlHl(:,6),thetaignition+thetab);
% integration during expansion phase
\[\text{disp('integrating over the expansion phase')};\]
[thetaexp,pTbWQlHl]=ode45('RatesExp', ...
[thetaignition+thetab:dtheta:portopenindex/180*pi],[pe Tbe We Qle Hle],options);
% error checks
mass4=mass1*exp(-Cblowby*2*pi/omega);
p4=interp1(thetaexp,pTbWQlHl(:,1),portopenindex/180*pi);
T4=interp1(thetaexp,pTbWQlHl(:,2),portopenindex/180*pi);
W4=interp1(thetaexp,pTbWQlHl(:,3),portopenindex/180*pi);
Ql4=interp1(thetaexp,pTbWQlHl(:,4),portopenindex/180*pi);
Hl4=interp1(thetaexp,pTbWQlHl(:,5),portopenindex/180*pi);
[h4,ae,v4,s4,Y4,cp4,dlvlT4,dlvlp4]= ...
\(farg(p4,T4,phi,1,fueltype,airscheme);\)
U4=u4*mass4;
error1=1-v4*mass4/maxexV;
error2=1+W4/(U4-U1+Ql4+Hl4);
% indicated mean effective pressure and thermal efficiency
imep=W4/(pi*b^2/4*max(crankthrow,uppercrankthrow)*2); %imep not accurately defined for this
configuration
\(\text{eta}=W4/\text{mass}_1*(1+\phi^0.06548*(1-\phi)/\phi/0.06548*(1-\phi)/47870/1e3;\)
% calculate the heat flux in W/m^2
if isempty(thetacomp2)
qcomp=calcq(thetacomp,pTuWQlHl,'comp'); % compression
qcombu=calcq(thetacomp,pTuWQlHl,'combu'); % combustion-unburned zone
qcombb=calcq(thetacomp,pTuWQlHl,'combb'); % combustion-burned zone
qexp=calcq(thetaexp,pTuWQlHl,'exp'); % expansion
else
qcomp=calcq(thetacomp,pTuWQlHl,'comp'); % compression
qcomp2=calcq(thetacomp2,p2TuWQlHl,'comp'); % compression
qcomp3=calcq(thetacomp3,p3TuWQlHl,'comp'); % compression
qcombb=calcq(thetacomp,pTuWQlHl,'combb'); % combustion-burned zone
qexp=calcq(thetaexp,pTuWQlHl,'exp'); % expansion
end
timefinish=cputime;
timetaken=timefinish-timestart;
% save all data
clear ahrind.mat
save ahrind.mat
%clear

airdata.m

function A=airdata(scheme);
% A=airdata(scheme)
% Routine to specify the thermodynamic properties of air and
% combustion products.
% Data taken from:
% Calculation of Complex Chemical Equilibrium Composition, Rocket

********************************************************************
input:
scheme switch:
`GMcB_low' - Gordon and McBride 300 < T < 1000 K
`GMcB_hi' - Gordon and McBride 1000 < T < 5000 K
`Chemkin_low' - Chemkin 300 < T < 1000 K
`Chemkin_hi' - Chemkin 1000 < T < 5000 K
output:
A - matrix of polynomial coefficients for cp/R, h/RT, and s/R of the form h/RT=a1+a2*T/2+a3*T^2/3+a4*T^3/4+a5*T^4/5+a6/T (for example) where T is expressed in K
columns 1 to 7 are coefficients a1 to a7, and
rows 1 to 10 are species CO2 H2O N2 O2 CO H2 H O OH and NO

switch scheme
case 'GMcB_low'
A=
[0.24007797E+01 0.87350957E-02 -0.66070878E-05 0.20021861E-08 ...
0.63274039E-15 -0.48377527E+05 0.96951457E+01 0.40701275E+01 -0.11084499E-02 0.41521180E-05 -0.29637404E-08 ...]
case 'GMcB_hi'
A=
[0.44608041E+01 0.30981719E-02 -0.12392571E-05 0.22741325E-09 ...
-0.15525954E-13 -0.48961442E+05 -0.98635982E+00 0.27167633E+01 0.29462878E+01 -0.16381665E-02 0.24210316E-06 -0.16028432E-08 ...]
case 'Chemkin_low'
A=
[0.02275724E+02 0.09922072E-01 -0.10409113E-04 0.06866686E-07 ...
-0.65223555E-14 -0.90586184E+03 0.61615148E+01 0.7361826E-03 -0.19652282E-06 0.36219535E+01 0.142145228E+00 0.36497155E+01]
case 'Chemkin_hi'
A=
[0.02275724E+02 0.09922072E-01 -0.10409113E-04 0.06866686E-07 ...
-0.65223555E-14 -0.90586184E+03 0.61615148E+01 0.7361826E-03 -0.19652282E-06 0.36219535E+01 0.142145228E+00 0.36497155E+01]
calcq.m

function q=calcq(theta,pTarray,phase);
%
% calculation of the heat flux (W/m^2) from the data generated
% by ahrind.
% theta is an array of crank angles
% pTarray is the corresponding array of pressure, Temperature,
% Work, etc data as generated by running arhind.m
% phase is a switch indicating the part of the cycle:
% 'comp' - compression phase
% 'combu' - combustion phase, unburned gas zone
% 'combb' - combustion phase, burned gas zone
% 'exp' - expansion phase
global b stroke eps r ...
omega ...
heattransferlaw hcu hcb ...
Tw Vdce Vbde ...
p1 T1 V1
switch phase
case 'comp'
p=qTarray(:,1);
end
T=pTarray(:,2);
hc=hcu;
C2=0;
case 'combu'
p=pTarray(:,1);
T=pTarray(:,3);
hc=hcu;
C2=3.24e-3;
case 'comb'
p=pTarray(:,1);
T=pTarray(:,2);
hc=hcb;
C2=3.24e-3;
case 'exp'
p=pTarray(:,1);
T=pTarray(:,2);
hc=hcb;
C2=3.24e-3;
case 'constant'
hcoeff=hc;
case 'Woschni'
V=Vtdc*(1+(r-1)/2*(1-cos(theta)+ ... 
1/eps*(1-eps^2*sin(theta).^2).^0.5));
upmean=omega*stroke/pi; % mean piston velocity
Vs=Vbdc-Vtdc;
k=1.3;
C1=2.28;
pm=p1*(V1./V).^k; % motoring pressure
hcoeff=hc*130*b^(-0.2)*T.^(-0.53).*(p/100e3).^(0.8).* ...
(C1*upmean+C2*Vs*T1/p1/V1*(p-pm)).^(0.8);
end
q=hcoeff.*(T-Tw);

compile_data_CCR.m

%M-file to plot friction work, nett eff and thermal eff vs inducted vol for 
%all engine specs

warning off all

specdata=zeros(110,6);
compare=zeros(3,360);

[version,uppercrankoffset,upperconrod,uppercrankthrow,crankoffset,...
 conrod,crankthrow,cranktorcrank,exportheight,uppercranklag,...
 uppercrankoffsrange,upperconrodrange,uppercrankthrowrange,...
 crankoffsrange,conrodrange,crankthrowrange,cranktorcrankrange,...
 exportheightrange,uppercranklagrange,thetasdeg,thetabdeg,...
 thetasdegrange,thetabdegrange,splugport]=enginespecCCR3;
[neteff,thermaleff,inductedvol,frictionwork,version,maxpress,ACR,StNoupper3,...
 StNolower3,dSupperdt3,d2Supperdt3,dSlowerdt3,d2Slowerdt3,upperpistfrictionvar3,lowerpistfrictionvar3]=...
 set_input_data(version,uppercrankoffset,upperconrod,uppercrankthrow,crankoffset,...
 conrod,crankthrow,cranktorcrank,exportheight,uppercranklag,...
 uppercrankoffsrange,upperconrodrange,uppercrankthrowrange,...
 crankoffsrange,conrodrange,crankthrowrange,cranktorcrankrange,...
 exportheightrange,uppercranklagrange,thetasdeg,thetabdeg,...
 thetasdegrange,thetabdegrange,splugport);
L=length(version);
version=str2num(version(L-3:L));
specdata(30,:)=[neteff,thermaleff,inductedvol,frictionwork,version,maxpress];
StNoowerd4_dSupperdt4_5, d2Supperdt24_5, dSlowerd4_5, d2Slowerd24_5, upperpistfrictionvar4_5, lowerpistfrictionvar4_5]=...
set_input_data(version, uppercrankoffset, upperconrod, uppercrankthrow, crankoffset, uppercrankoffsetrange, upperconrodrange, uppercrankthrowrange, crankoffsetrange, conrodrange, crankthrowrange, cranktocrankrange, thetasdegrange, thetabdegrange, splugport)=enginespecCCR4_5;

StNoowerd5, dSupperdt5, d2Supperdt25, dSlowerd5, d2Slowerd25, upperpistfrictionvar5, lowerpistfrictionvar5]=...
set_input_data(version, uppercrankoffset, upperconrod, uppercrankthrow, crankoffset, uppercrankoffsetrange, upperconrodrange, uppercrankthrowrange, crankoffsetrange, conrodrange, crankthrowrange, cranktocrankrange, thetasdegrange, thetabdegrange, splugport)=enginespecCCR5;

StNoowerd5, dSupperdt5, d2Supperdt25, dSlowerd5, d2Slowerd25, upperpistfrictionvar5, lowerpistfrictionvar5]=...
set_input_data(version, uppercrankoffset, upperconrod, uppercrankthrow, crankoffset, uppercrankoffsetrange, upperconrodrange, uppercrankthrowrange, crankoffsetrange, conrodrange, crankthrowrange, cranktocrankrange, thetasdegrange, thetabdegrange, splugport)=enginespecCCR6;
StNolower6,dSupperd6_2,d2Supperdt6_2,dSlowerd6_2,d2Slowerdt6_2,upperpistfrictionvar6_2,lowerpistfrictionvar6_2]=...
set_input_data(version,uppercrankoffset,upperconrod,uppercrankthrow,crankoffset,....
conrod,crankthrow,crankto crank,exportheight,uppercranklag,...
uppercrankoffsetrange, upperconrodrange, uppercrankthrowrange,....
crankoffsetrange, conrodrange, crankthrowrange, crankto crankrange,....
exportheightrange, uppercranklagrange, thetasdeg, thetabdeg,....
theetasdegrange, thetabdegrange, splugport);
L=length(version);
version=str2num(version(L-3:L));
specdata(60,:)=[neteff, thermaleff, inductedvol, frictionwork, version, maxpress];

[version, uppercrankoffset, upperconrod, uppercrankthrow, crankoffset,....
conrod, crankthrow, crankto crank, exportheight, uppercranklag,...
uppercrankoffsetrange, upperconrodrange, uppercrankthrowrange,....
crankoffsetrange, conrodrange, crankthrowrange, crankto crankrange,....
exportheightrange, uppercranklagrange, thetasdeg, thetabdeg,....
theetasdegrange, thetabdegrange, splugport]=enginespecCCR6_2;
[neteff, thermaleff, inductedvol, frictionwork, version, maxpress, ACR, StNoupper6_2,....
StNolower6_2, dSupperdt6_2, d2Supperdt6_2, dSlowerd6_2, d2Slowerdt6_2, upperpistfrictionvar6_2, lowerpistf
ictionvar6_2]=...
set_input_data(version, uppercrankoffset, upperconrod, uppercrankthrow, crankoffset,....
conrod, crankthrow, crankto crank, exportheight, uppercranklag,...
uppercrankoffsetrange, upperconrodrange, uppercrankthrowrange,....
crankoffsetrange, conrodrange, crankthrowrange, crankto crankrange,....
exportheightrange, uppercranklagrange, thetasdeg, thetabdeg,....
theetasdegrange, thetabdegrange, splugport);
L=length(version);
version=str2num(version(L-3:L));
specdata(62,:)=[neteff, thermaleff, inductedvol, frictionwork, version, maxpress];

[version, uppercrankoffset, upperconrod, uppercrankthrow, crankoffset,....
conrod, crankthrow, crankto crank, exportheight, uppercranklag,...
uppercrankoffsetrange, upperconrodrange, uppercrankthrowrange,....
crankoffsetrange, conrodrange, crankthrowrange, crankto crankrange,....
exportheightrange, uppercranklagrange, thetasdeg, thetabdeg,....
theetasdegrange, thetabdegrange, splugport]=enginespecCCR6_5;
[neteff, thermaleff, inductedvol, frictionwork, version, maxpress, ACR, StNoupper6_5,....
StNolower6_5, dSupperdt6_5, d2Supperdt6_5, dSlowerd6_5, d2Slowerdt6_5, upperpistfrictionvar6_5, lowerpistf
ictionvar6_5]=...
set_input_data(version, uppercrankoffset, upperconrod, uppercrankthrow, crankoffset,....
conrod, crankthrow, crankto crank, exportheight, uppercranklag,...
uppercrankoffsetrange, upperconrodrange, uppercrankthrowrange,....
crankoffsetrange, conrodrange, crankthrowrange, crankto crankrange,....
exportheightrange, uppercranklagrange, thetasdeg, thetabdeg,....
theetasdegrange, thetabdegrange, splugport);
L=length(version);
version=str2num(version(L-3:L));
specdata(65,:)=[neteff, thermaleff, inductedvol, frictionwork, version, maxpress];

[version, uppercrankoffset, upperconrod, uppercrankthrow, crankoffset,....
conrod, crankthrow, crankto crank, exportheight, uppercranklag,...
uppercrankoffsetrange, upperconrodrange, uppercrankthrowrange,....
crankoffsetrange, conrodrange, crankthrowrange, crankto crankrange,....
exportheightrange, uppercranklagrange, thetasdeg, thetabdeg,....
theetasdegrange, thetabdegrange, splugport]=enginespecCCR7;
[neteff, thermaleff, inductedvol, frictionwork, version, maxpress, ACR, StNoupper7,....
StNolower7, dSupperdt7, d2Supperdt7, dSlowerdt7, d2Slowerdt7, upperpistfrictionvar7, lowerpistfrictionvar7]=...
set_input_data(version, uppercrankoffset, upperconrod, uppercrankthrow, crankoffset,....
conrod, crankthrow, crankto crank, exportheight, uppercranklag,...
uppercrankoffsetrange, upperconrodrange, uppercrankthrowrange,....
crankoffsetrange, conrodrange, crankthrowrange, crankto crankrange,....
exportheightrange, uppercranklagrange, thetasdeg, thetabdeg,....
thetasdegrange,thetabdegrange,spugport);
L=length(version);
version=str2num(version(L-3:L));
specdata(70,:)=[neteff,thermaleff,inductedvol,frictionwork,version,maxpress];

L=length(version);
version=str2num(version(L-3:L));
specdata(80,:)=[neteff,thermaleff,inductedvol,frictionwork,version,maxpress,ACR,StNoupper8_5,StNolower8_5,dSupperdt8_5,d2Supperdt8_5,dSlowerdt8_5,d2Slowerdt8_5,upperpistfrictionvar8_5,lowerpistfrictionvar8_5];

L=length(version);
version=str2num(version(L-3:L));
specdata(85,:)=specdata(70,:);

L=length(version);
version=str2num(version(L-3:L));
specdata(90,:)=specdata(80,:);
crankoffsetrange, conrodrange, crankthrowrange, crankto crankrange,...
exportheightrange, uppercranklagrange, thetasdeg, thetabdeg,...
thesasdegrange, thetabdegrange, splugport] = enginespecCCR9_2;

[neteff, thermal eff, inductedvol, frictionwork, version, maxpress, ACR, StNoupper9_2, ...]
StNolower9_2, dSupperdt9_2, d2Supperdt29_2, dSlowerdt9_2, d2Slowerdt29_2, upperpistfrictionvar9_2, lowerpistfrictionvar9_2] =
set_input_data(version, uppercrankoffset, upperconrod, uppercrankthrow, crankoffset,...
conrod, crankthrow, crankto crank, exportheight, uppercranklag,...
uppercrankoffsetrange, upperconrodrange, uppercrankthrowrange,...
crankoffsetrange, conrodrange, crankthrowrange, crankto crankrange,...
exportheightrange, uppercranklagrange, thetasdeg, thetabdeg,...
thesasdegrange, thetabdegrange, splugport);
L = length(version);
version = str2num(version(L-3:L));
specdata(92,:) = [neteff, thermal eff, inductedvol, frictionwork, version, maxpress];

[version, uppercrankoffset, upperconrod, uppercrankthrow, crankoffset,...
conrod, crankthrow, crankto crank, exportheight, uppercranklag,...
uppercrankoffsetrange, upperconrodrange, uppercrankthrowrange,...
crankoffsetrange, conrodrange, crankthrowrange, crankto crankrange,...
exportheightrange, uppercranklagrange, thetasdeg, thetabdeg,...
thesasdegrange, thetabdegrange, splugport] = enginespecCCR9_5;
[neteff, thermal eff, inductedvol, frictionwork, version, maxpress, ACR, StNoupper9_5, ...
StNolower9_5, dSupperdt9_5, d2Supperdt29_5, dSlowerdt9_5, d2Slowerdt29_5, upperpistfrictionvar9_5, lowerpistfrictionvar9_5] =
set_input_data(version, uppercrankoffset, upperconrod, uppercrankthrow, crankoffset,...
conrod, crankthrow, crankto crank, exportheight, uppercranklag,...
uppercrankoffsetrange, upperconrodrange, uppercrankthrowrange,...
crankoffsetrange, conrodrange, crankthrowrange, crankto crankrange,...
exportheightrange, uppercranklagrange, thetasdeg, thetabdeg,...
thesasdegrange, thetabdegrange, splugport);
L = length(version);
version = str2num(version(L-3:L));
specdata(95,:) = [neteff, thermal eff, inductedvol, frictionwork, version, maxpress];

[version, uppercrankoffset, upperconrod, uppercrankthrow, crankoffset,...
conrod, crankthrow, crankto crank, exportheight, uppercranklag,...
uppercrankoffsetrange, upperconrodrange, uppercrankthrowrange,...
crankoffsetrange, conrodrange, crankthrowrange, crankto crankrange,...
exportheightrange, uppercranklagrange, thetasdeg, thetabdeg,...
thesasdegrange, thetabdegrange, splugport] = enginespecCCR10;
[neteff, thermal eff, inductedvol, frictionwork, version, maxpress, ACR, StNoupper10, ...
StNolower10, dSupperdt10, d2Supperdt210, dSlowerdt10, d2Slowerdt210, upperpistfrictionvar10, lowerpistfrictionvar10] =
set_input_data(version, uppercrankoffset, upperconrod, uppercrankthrow, crankoffset,...
conrod, crankthrow, crankto crank, exportheight, uppercranklag,...
uppercrankoffsetrange, upperconrodrange, uppercrankthrowrange,...
crankoffsetrange, conrodrange, crankthrowrange, crankto crankrange,...
exportheightrange, uppercranklagrange, thetasdeg, thetabdeg,...
thesasdegrange, thetabdegrange, splugport);
L = length(version);
version = str2num(version(L-3:L));
specdata(100,:) = [neteff, thermal eff, inductedvol, frictionwork, version, maxpress];
StNolower 11,d Supper dt 11,d2 Supper dt 211,d Suppler dt 11,d2 Slower dt 211, upper pist friction var 11, lower pist friction var 11 = ...

set_input_data(version, upper crank offset, upper con rod, upper crank throw, crank offset, ...
con rod, crank throw, crank to crank, export height, upper crank lag, ....
upper crank offset range, upper con rod range, upper crank throw range, ...
crank offset range, con rod range, crank throw range, crank to crank range, ....
export height range, upper crank lag range, theta deg, theta deg, ....
theta deg range, theta deg range, splug port);

L = length(version);
version = str2num(version(L-3:L));
specdata(110,:) = [net eff, thermal eff, inducted vol, friction work, version, max press];

activ rows = find(specdata(:,1)~=0);
specdata = specdata(activ rows,:);

net eff = specdata(:,1)*100;
inducted vol = specdata(:,3);
thermal eff = specdata(:,2)*100;
friction work = specdata(:,4);
version = specdata(:,5);
max press = specdata(:,6);

plot(inducted vol, net eff, 'o', inducted vol, thermal eff, 's')
title ('Efficiency Vs Inducted Volume for Various Eng Specs')
legend ('Net Eff', 'Thermal Eff')

coeffs = polyfit(inducted vol, net eff, 4);
xpoints = inducted vol(1):0.1:inducted vol(end);
hold on
polyvaldata = polyval(coeffs, xpoints);
plot(xpoints, polyvaldata, 'g');
coeffs2 = polyfit(inducted vol, thermal eff, 2);
hold on
polyval2data = polyval(coeffs2, xpoints);

xlabel ('Inducted Volume (cc)')
ylabel ('Efficiency (%)')

axis ([min(inducted vol)-15 max(inducted vol)+10 ...
min(net eff)-5 max(thermal eff+5)]);

coeffs = polyfit(inducted vol, net eff, 4);
xpoints = inducted vol(1):0.1:inducted vol(end);
hold on
polyvaldata = polyval(coeffs, xpoints);
plot(xpoints, polyvaldata, 'g');
coeffs2 = polyfit(inducted vol, thermal eff, 2);
hold on
polyval2data = polyval(coeffs2, xpoints);
plot(xpoints,polyval2data,'Color',[0,0.5,0.5]);

% splinedata=spline(eff_disp_data(:,2),eff_disp_data(:,1),xpoints);
% plot (xpoints,splinedata,'r')

[m,n] = size(specdata);

for counter=1:m;
    if counter==1
        speclabel=sprintf('Spec%4.1f',version(counter));
        presslabel=sprintf('Press %4.1f',maxpress(counter));
        text(inductedvol(counter),neteff(counter)+0.5,speclabel,'Color',[0,.5,0.5],'Rotation',[90])
        text(inductedvol(counter)+0.2,neteff(counter)-0.5,presslabel,'Color',[0,0,1],'Rotation',[-60])
    else
        speclabel=sprintf('%4.1f',version(counter));
        presslabel=sprintf('%4.1f',maxpress(counter));
        text(inductedvol(counter),neteff(counter)+0.5,speclabel,'Color',[0,.5,0.5],'Rotation',[90])
        text(inductedvol(counter)+0.2,neteff(counter)-0.5,presslabel,'Color',[0,0,1],'Rotation',[-60])
    end
end

%original constant max press data
% eff Disp_data= [34.9,28.2,7.9,2.4,2.0,23.0;...
%                 41.3,42.3,9.2,6.0,6.2,45.2;...
%                 43.5,53.7,9.7,6.5,6.9,0.0;...
%                 43.7,74.0,10.1,6.9,7.2,0.7;...
%                 44.1,78.3,10.3,8.0,7.9,7.3;...
%                 44.5,83.1,10.5,8.5,8.2,7.5;...
%                 44.9,86.5,10.7,9.0,8.8,9.7;...
%                 44.8,91.0,10.8,3.0,9.2,2.0;...
%                 44.8,94.3,10.9,3.2,9.5,6.3;...
%                 42.9,103.5,10.9,3.2,11.0,116.4;]

figure

plot(inductedvol,frictionwork,'d')
title ('Friction Work Vs Inducted Volume for Various Eng Specs')
xlabel ('Inducted Volume (cc)')
ylabel ('Work (J)')

coeffs2=polyfit(inductedvol,frictionwork,2);
hold on
polyval2data=polyval(coeffs2,xpoints);
plot(xpoints,polyval2data,'Color',[0,0.5,0.5]);
axis([min(inductedvol)-15 max(inductedvol)+10 ...
     min(frictionwork)-5 max(frictionwork+5)]);

text (22,max(frictionwork)+2.0,'Bore' )
text (40,max(frictionwork)+2.0,'= 61.3mm')
text (22,max(frictionwork)+0.5,'E.O.Press')
text (40,max(frictionwork)+0.5,'= 1 atm')
text (22,max(frictionwork)-1,'Const Compression')
text (22,max(frictionwork)-2.5,'Ratio')
text (40,max(frictionwork)-2.5,'= 10.0')
text (22,max(frictionwork)-4,'Const Burn Start')
text (40,max(frictionwork)-5.5,'= -27 deg')
for counter=1:m;
    if counter==1
        speclabel=sprintf('Spec%4.1f',version(counter));
        text(inductedvol(counter),frictionwork(counter)+0.5,speclabel,'Color',[0,.5,0.5], 'Rotation',[90])
    else
        speclabel=sprintf('%4.1f',version(counter));
        text(inductedvol(counter),frictionwork(counter)+0.5,speclabel,'Color',[0,.5,0.5], 'Rotation',[90])
    end
end

compile_data_CEP.m

% M-file to plot friction work, nett eff and thermal eff vs inducted vol for
% all engine specs

warning off all

specdata=zeros(110,6);
compare=zeros(3,360);

[version,uppercrankoffset,upperconrod,uppercrankthrow,crankoffset,....
 conrod,crankthrow,cranktocrank,exportheight,uppercranklag,....
 uppercrankoffsetrange,uppercronoordrange,uppercrankthrowrange,....
 crankoffsetrange,conrodrange,crankthrowrange,cranktocrankrange,....
 exportheightrange,uppercranklagrange,thetasdeg,thetabdeg,....
 thetasdegradge,thetabdegradge,slugport]=enginespecoptsize3;
[neteff,thermaleff,inductedvol,frictionwork,version,maxpress,ACR,StNoupper3,....
 StNolower3,dSupperdt3,d2Supperdt23,dSlowerdt3,d2Slowerdt23,upperpistfrictionvar3,lowerpistfrictionvar3]=...
 set_input_data(version,uppercrankoffset,upperconrod,uppercrankthrow,crankoffset,....
 conrod,crankthrow,cranktocrank,exportheight,uppercranklag,....
 uppercrankoffsetrange,uppercronoordrange,uppercrankthrowrange,....
 crankoffsetrange,conrodrange,crankthrowrange,cranktocrankrange,....
 exportheightrange,uppercranklagrange,thetasdeg,thetabdeg,....
 thetasdegradge,thetabdegradge,slugport]=enginespecoptsize3;

L=length(version);
version=str2num(version(L-3:L));

specdata(30,:)=[neteff,thermaleff,inductedvol,frictionwork,version,ACR];

[version,uppercrankoffset,upperconrod,uppercrankthrow,crankoffset,....
 conrod,crankthrow,cranktocrank,exportheight,uppercranklag,....
 uppercrankoffsetrange,uppercronoordrange,uppercrankthrowrange,....
 crankoffsetrange,conrodrange,crankthrowrange,cranktocrankrange,....
 exportheightrange,uppercranklagrange,thetasdeg,thetabdeg,....
 thetasdegradge,thetabdegradge,slugport]=enginespecoptsize4;

L=length(version);
version=str2num(version(L-3:L));

specdata(40,:)=[neteff,thermaleff,inductedvol,frictionwork,version,ACR];

[version,uppercrankoffset,upperconrod,uppercrankthrow,crankoffset,....
 conrod,crankthrow,cranktocrank,exportheight,uppercranklag,....
 uppercrankoffsetrange,uppercronoordrange,uppercrankthrowrange,....
 crankoffsetrange,conrodrange,crankthrowrange,cranktocrankrange,....
 exportheightrange,uppercranklagrange,thetasdeg,thetabdeg,....
 thetasdegradge,thetabdegradge,slugport]=enginespecoptsize4;

L=length(version);
version=str2num(version(L-3:L));

specdata(50,:)=[neteff,thermaleff,inductedvol,frictionwork,version,ACR];
conrod,crankthrow,cranktocrank,exportheight,uppercranklag,...
uppercrankoffsetrange,upperconrodrange,uppercrankthrowrange,...
crankoffsetrange,conrodrange,crankthrowrange,cranktocrankrange,...
exportheightrange,uppercranklagrange,thetasdeg,thetabdeg,...
thesadegrange,thetabdegrange,splugport]=enginespecoptsize4_5;
[neteff,thermaleff,inductedvol,frictionwork,version,maxpress,ACR,StNoupper4_5,...
StNolower4_5,dSupperdt4_5,d2Supperdt24_5,dSlowerdt4_5,d2Slowerdt24_5,upperpistfrictionvar4_5,lowerpistfrictionvar4_5]=....
set_input_data(version,uppercrankoffset,upperconrod,uppercrankthrow,crankoffset,...
conrod,crankthrow,cranktocrank,exportheight,uppercranklag,...
uppercrankoffsetrange,upperconrodrange,uppercrankthrowrange,...
crankoffsetrange,conrodrange,crankthrowrange,cranktocrankrange,...
exportheightrange,uppercranklagrange,thetasdeg,thetabdeg,...
thesadegrange,thetabdegrange,splugport);
L=length(version);
version=str2num(version(L-3:L));
specdata(45,:)=[neteff,thermaleff,inductedvol,frictionwork,version,ACR];

[version,uppercrankoffset,upperconrod,uppercrankthrow,crankoffset,...
conrod,crankthrow,cranktocrank,exportheight,uppercranklag,...
uppercrankoffsetrange,upperconrodrange,uppercrankthrowrange,...
crankoffsetrange,conrodrange,crankthrowrange,cranktocrankrange,...
exportheightrange,uppercranklagrange,thetasdeg,thetabdeg,...
thesadegrange,thetabdegrange,splugport]=enginespecoptsize5;
[neteff,thermaleff,inductedvol,frictionwork,version,maxpress,ACR,StNoupper5,...
StNolower5,dSupperdt5,d2Supperdt25,dSlowerdt5,d2Slowerdt25,upperpistfrictionvar5,lowerpistfrictionvar5]=....
set_input_data(version,uppercrankoffset,upperconrod,uppercrankthrow,crankoffset,...
conrod,crankthrow,cranktocrank,exportheight,uppercranklag,...
uppercrankoffsetrange,upperconrodrange,uppercrankthrowrange,...
crankoffsetrange,conrodrange,crankthrowrange,cranktocrankrange,...
exportheightrange,uppercranklagrange,thetasdeg,thetabdeg,...
thesadegrange,thetabdegrange,splugport);
L=length(version);
version=str2num(version(L-3:L));
specdata(50,:)=[neteff,thermaleff,inductedvol,frictionwork,version,ACR];

[version,uppercrankoffset,upperconrod,uppercrankthrow,crankoffset,...
conrod,crankthrow,cranktocrank,exportheight,uppercranklag,...
uppercrankoffsetrange,upperconrodrange,uppercrankthrowrange,...
crankoffsetrange,conrodrange,crankthrowrange,cranktocrankrange,...
exportheightrange,uppercranklagrange,thetasdeg,thetabdeg,...
thesadegrange,thetabdegrange,splugport]=enginespecoptsize6;
[neteff,thermaleff,inductedvol,frictionwork,version,maxpress,ACR,StNoupper6,...
StNolower6,dSupperdt6,d2Supperdt26,dSlowerdt6,d2Slowerdt26,upperpistfrictionvar6,lowerpistfrictionvar6]=....
set_input_data(version,uppercrankoffset,upperconrod,uppercrankthrow,crankoffset,...
conrod,crankthrow,cranktocrank,exportheight,uppercranklag,...
uppercrankoffsetrange,upperconrodrange,uppercrankthrowrange,...
crankoffsetrange,conrodrange,crankthrowrange,cranktocrankrange,...
exportheightrange,uppercranklagrange,thetasdeg,thetabdeg,...
thesadegrange,thetabdegrange,splugport);
L=length(version);
version=str2num(version(L-3:L));
specdata(60,:)=[neteff,thermaleff,inductedvol,frictionwork,version,ACR];

[version,uppercrankoffset,upperconrod,uppercrankthrow,crankoffset,...
conrod,crankthrow,cranktocrank,exportheight,uppercranklag,...
uppercrankoffsetrange,upperconrodrange,uppercrankthrowrange,...
crankoffsetrange,conrodrange,crankthrowrange,cranktocrankrange,...
exportheightrange,uppercranklagrange,thetasdeg,thetabdeg,...
thesadegrange,thetabdegrange,splugport]=enginespecoptsize6_2;
StNolower6_2,dSupperdt6_2,d2Supperdt26_2,dSlowerdt6_2,d2Slowerdt26_2,upperpistfrictionvar6_2,lowerpistfrictionvar6_2]=...  
set_input_data(version, uppercrankoffset, upperconrod, uppercrankthrow, crankoffset,....
conrod, crankthrow, cranktocrank, exportheight, uppercranklag,....
uppercrankoffsetrange, upperconrodrange, uppercrankthrowrange,....
crakoffsetrange, conrodrange, crankthrowrange, cranktocrankrange,....
exportheightrange, uppercranklagrange, thetasdeg, thetabdeg,....
theetasdegrange, thetabdegrange, spugport);
L=length(version);
version=str2num(version(L-3:L));
specdata(62,:)=[neteff, thermaleff, inductedvol, frictionwork, version, ACR];

[version, uppercrankoffset, upperconrod, uppercrankthrow, crankoffset,....
conrod, crankthrow, cranktocrank, exportheight, uppercranklag,....
uppercrankoffsetrange, upperconrodrange, uppercrankthrowrange,....
crakoffsetrange, conrodrange, crankthrowrange, cranktocrankrange,....
exportheightrange, uppercranklagrange, thetasdeg, thetabdeg,....
theetasdegrange, thetabdegrange, spugport]=enginespecoptsize6_5;
[neteff, thermaleff, inductedvol, frictionwork, version, maxpress, ACR, StNoupper6_5,....
StNolower6_5, dSupperdt6_5, d2Supperdt26_5, dSlowerdt6_5, d2Slowerdt26_5, upperpistfrictionvar6_5, lowerpistfrictionvar6_5]=...  
set_input_data(version, uppercrankoffset, upperconrod, uppercrankthrow, crankoffset,....
conrod, crankthrow, cranktocrank, exportheight, uppercranklag,....
uppercrankoffsetrange, upperconrodrange, uppercrankthrowrange,....
crakoffsetrange, conrodrange, crankthrowrange, cranktocrankrange,....
exportheightrange, uppercranklagrange, thetasdeg, thetabdeg,....
theetasdegrange, thetabdegrange, spugport);
L=length(version);
version=str2num(version(L-3:L));
specdata(65,:)=[neteff, thermaleff, inductedvol, frictionwork, version, ACR];

[version, uppercrankoffset, upperconrod, uppercrankthrow, crankoffset,....
conrod, crankthrow, cranktocrank, exportheight, uppercranklag,....
uppercrankoffsetrange, upperconrodrange, uppercrankthrowrange,....
crakoffsetrange, conrodrange, crankthrowrange, cranktocrankrange,....
exportheightrange, uppercranklagrange, thetasdeg, thetabdeg,....
theetasdegrange, thetabdegrange, spugport]=enginespecoptsize7;
[neteff, thermaleff, inductedvol, frictionwork, version, maxpress, ACR, StNoupper7,....
StNolower7,dSupperdt7,d2Supperdt27, dSlowerdt7,d2Slowerdt27, upperpistfrictionvar7, lowerpistfrictionvar7]=...  
set_input_data(version, uppercrankoffset, upperconrod, uppercrankthrow, crankoffset,....
conrod, crankthrow, cranktocrank, exportheight, uppercranklag,....
uppercrankoffsetrange, upperconrodrange, uppercrankthrowrange,....
crakoffsetrange, conrodrange, crankthrowrange, cranktocrankrange,....
exportheightrange, uppercranklagrange, thetasdeg, thetabdeg,....
theetasdegrange, thetabdegrange, spugport);
L=length(version);
version=str2num(version(L-3:L));
specdata(70,:)=[neteff, thermaleff, inductedvol, frictionwork, version, ACR];

[version, uppercrankoffset, upperconrod, uppercrankthrow, crankoffset,....
conrod, crankthrow, cranktocrank, exportheight, uppercranklag,....
uppercrankoffsetrange, upperconrodrange, uppercrankthrowrange,....
crakoffsetrange, conrodrange, crankthrowrange, cranktocrankrange,....
exportheightrange, uppercranklagrange, thetasdeg, thetabdeg,....
theetasdegrange, thetabdegrange, spugport]=enginespecoptsize8;
[neteff, thermaleff, inductedvol, frictionwork, version, maxpress, ACR, StNoupper8,....
StNolower8,dSupperdt8,d2Supperdt28, dSlowerdt8,d2Slowerdt28, upperpistfrictionvar8, lowerpistfrictionvar8]=...  
set_input_data(version, uppercrankoffset, upperconrod, uppercrankthrow, crankoffset,....
conrod, crankthrow, cranktocrank, exportheight, uppercranklag,....
uppercrankoffsetrange, upperconrodrange, uppercrankthrowrange,....
crakoffsetrange, conrodrange, crankthrowrange, cranktocrankrange,....
exportheightrange, uppercranklagrange, thetasdeg, thetabdeg,....
L=length(version);
version=str2num(version(L-3:L));
specdata(80,:)=[neteff,thermaleff,inductedvol,frictionwork,version,ACR];

[version,uppercrankoffset,upperconrod,uppercrankthrow,crankoffset,...
 conrod,crankthrow,cranktocrank,exportheight,uppercranklag,...
 uppercrankoffsetrange,upperconrodrange,uppercrankthrowrange,...
 crankoffsetsetrange,conrodrange,crankthrowrange,cranktocrankrange,...
 exportheightrange,uppercranklagrange,thetasdeg,thetabdeg,...
 thetasdegrange,thetabdegrange,splugport]=enginespecoptsize8_5;
[neteff,thermaleff,inductedvol,frictionwork,version,ACR,StNoupper8_5,...
 StNolower8_5,d2Supperdt8_5,d2Slowerdt8_5,d2Slowerdt28_5,upperpistfrictionvar8_5,lowerpistfrictionvar8_5]=...  
 set_input_data(version,uppercrankoffset,upperconrod,uppercrankthrow,crankoffset,...
 conrod,crankthrow,cranktocrank,exportheight,uppercranklag,...
 uppercrankoffsetrange,upperconrodrange,uppercrankthrowrange,...
 crankoffsetsetrange,conrodrange,crankthrowrange,cranktocrankrange,...
 exportheightrange,uppercranklagrange,thetasdeg,thetabdeg,...
 thetasdegrange,thetabdegrange,splugport]=enginespecoptsize9;
[neteff,thermaleff,inductedvol,frictionwork,version,ACR,StNoupper9,...
 StNolower9,d2Supperdt9,d2Slowerdt9,d2Slowerdt29,upperpistfrictionvar9,lowerpistfrictionvar9]=...  
 set_input_data(version,uppercrankoffset,upperconrod,uppercrankthrow,crankoffset,...
 conrod,crankthrow,cranktocrank,exportheight,uppercranklag,...
 uppercrankoffsetrange,upperconrodrange,uppercrankthrowrange,...
 crankoffsetsetrange,conrodrange,crankthrowrange,cranktocrankrange,...
 exportheightrange,uppercranklagrange,thetasdeg,thetabdeg,...
 thetasdegrange,thetabdegrange,splugport]=enginespecoptsize9_2;
[neteff,thermaleff,inductedvol,frictionwork,version,ACR,StNoupper9_2,...
 StNolower9_2,d2Supperdt9_2,d2Slowerdt9_2,d2Slowerdt29_2,upperpistfrictionvar9_2,lowerpistfrictionvar9_2]=...  
 set_input_data(version,uppercrankoffset,upperconrod,uppercrankthrow,crankoffset,...
 conrod,crankthrow,cranktocrank,exportheight,uppercranklag,...
 uppercrankoffsetrange,upperconrodrange,uppercrankthrowrange,...
 crankoffsetsetrange,conrodrange,crankthrowrange,cranktocrankrange,...
 exportheightrange,uppercranklagrange,thetasdeg,thetabdeg,...
 thetasdegrange,thetabdegrange,splugport]=enginespecoptsize9_2_2;
[neteff,thermaleff,inductedvol,frictionwork,version,ACR,StNoupper9_2_2,...
 StNolower9_2_2,d2Supperdt9_2_2,d2Slowerdt9_2_2,d2Slowerdt29_2_2,upperpistfrictionvar9_2_2,lowerpistfrictionvar9_2_2]=...  
 set_input_data(version,uppercrankoffset,upperconrod,uppercrankthrow,crankoffset,...
 conrod,crankthrow,cranktocrank,exportheight,uppercranklag,...
 uppercrankoffsetrange,upperconrodrange,uppercrankthrowrange,...
 crankoffsetsetrange,conrodrange,crankthrowrange,cranktocrankrange,...
 exportheightrange,uppercranklagrange,thetasdeg,thetabdeg,...
 thetasdegrange,thetabdegrange,splugport]=enginespecoptsize9_2_2_2;
[neteff,thermaleff,inductedvol,frictionwork,version,ACR,StNoupper9_2_2_2,...
 StNolower9_2_2_2,d2Supperdt9_2_2_2,d2Slowerdt9_2_2_2,d2Slowerdt29_2_2_2,upperpistfrictionvar9_2_2_2,lowerpistfrictionvar9_2_2_2]=...  
 set_input_data(version,uppercrankoffset,upperconrod,uppercrankthrow,crankoffset,...
 conrod,crankthrow,cranktocrank,exportheight,uppercranklag,...
 uppercrankoffsetrange,upperconrodrange,uppercrankthrowrange,...
 crankoffsetsetrange,conrodrange,crankthrowrange,cranktocrankrange,...
 exportheightrange,uppercranklagrange,thetasdeg,thetabdeg,...
 thetasdegrange,thetabdegrange,splugport]=enginespecoptsize9_2_2_2_2;
[neteff,thermaleff,inductedvol,frictionwork,version,ACR,StNoupper9_2_2_2_2,...
 StNolower9_2_2_2_2,d2Supperdt9_2_2_2_2,d2Slowerdt9_2_2_2_2,d2Slowerdt29_2_2_2_2,upperpistfrictionvar9_2_2_2_2,lowerpistfrictionvar9_2_2_2_2]=...  
 set_input_data(version,uppercrankoffset,upperconrod,uppercrankthrow,crankoffset,...
 conrod,crankthrow,cranktocrank,exportheight,uppercranklag,...
 uppercrankoffsetrange,upperconrodrange,uppercrankthrowrange,...
 crankoffsetsetrange,conrodrange,crankthrowrange,cranktocrankrange,...
 exportheightrange,uppercranklagrange,thetasdeg,thetabdeg,...
 thetasdegrange,thetabdegrange,splugport]=enginespecoptsize9_2_2_2_2_2;
uppercrankoffsetrange, upperconrodrange, uppercrankthrowrange, ...
crankoffsetrange, conrodrange, crankthrowrange, cranktocrankrange, ...
exportheightrange, uppercranklagrange, thetadegree, thetabdegree, ...
thesasdegreerrange, thetabdegreerrange, splugrange, enginespecoptsize9.5;
[neteff, thermal, inducedvol, frictionwork, version, maxpress, ACR, StNupper9.5,...
StNlower9.5, dSupperdt9.5, d2Supperdt29.5, dSlowerdt9.5, d2Slowerdt29.5, upperpistfrictionvar9.5, upperpistfrictionvar9.5] = ... set_input_data(version, uppercrankoffset, upperconrod, uppercrankthrow, crankoffset, ...
crankthrow, cranktocrank, exportheight, uppercranklag, ...
uppercrankoffsetrange, upperconrodrange, uppercrankthrowrange, ...
crankoffsetrange, conrodrange, crankthrowrange, cranktocrankrange, ...
exportheightrange, uppercranklagrange, thetadegree, thetabdegree, ...
thesasdegreerrange, thetabdegreerrange, splugrange);
L = length(version);
version = str2num(version(L - 3:L));
specdata(95:,:) = [neteff, thermal, inducedvol, frictionwork, version, ACR];

[version, uppercrankoffset, upperconrod, uppercrankthrow, crankoffset, ...
crankthrow, cranktocrank, exportheight, uppercranklag, ...
uppercrankoffsetrange, upperconrodrange, uppercrankthrowrange, ...
crankoffsetrange, conrodrange, crankthrowrange, cranktocrankrange, ...
exportheightrange, uppercranklagrange, thetadegree, thetabdegree, ...
thesasdegreerrange, thetabdegreerrange, splugrange, enginespecoptsize10;
[neteff, thermal, inducedvol, frictionwork, version, maxpress, ACR, StNupper10,...
StNlower10, dSupperdt10, d2Supperdt210, dSlowerdt10, d2Slowerdt210, upperpistfrictionvar10, lowerpistfrictionvar10] = ... set_input_data(version, uppercrankoffset, upperconrod, uppercrankthrow, crankoffset, ...
crankthrow, cranktocrank, exportheight, uppercranklag, ...
uppercrankoffsetrange, upperconrodrange, uppercrankthrowrange, ...
crankoffsetrange, conrodrange, crankthrowrange, cranktocrankrange, ...
exportheightrange, uppercranklagrange, thetadegree, thetabdegree, ...
thesasdegreerrange, thetabdegreerrange, splugrange);
L = length(version);
version = str2num(version(L - 3:L));
specdata(100,:) = [neteff, thermal, inducedvol, frictionwork, version, ACR];

[version, uppercrankoffset, upperconrod, uppercrankthrow, crankoffset, ...
crankthrow, cranktocrank, exportheight, uppercranklag, ...
uppercrankoffsetrange, upperconrodrange, uppercrankthrowrange, ...
crankoffsetrange, conrodrange, crankthrowrange, cranktocrankrange, ...
exportheightrange, uppercranklagrange, thetadegree, thetabdegree, ...
thesasdegreerrange, thetabdegreerrange, splugrange, enginespecoptsize11;
[neteff, thermal, inducedvol, frictionwork, version, maxpress, ACR, StNupper11,...
StNlower11, dSupperdt11, d2Supperdt211, dSlowerdt11, d2Slowerdt211, upperpistfrictionvar11, lowerpistfrictionvar11] = ... set_input_data(version, uppercrankoffset, upperconrod, uppercrankthrow, crankoffset, ...
crankthrow, cranktocrank, exportheight, uppercranklag, ...
uppercrankoffsetrange, upperconrodrange, uppercrankthrowrange, ...
crankoffsetrange, conrodrange, crankthrowrange, cranktocrankrange, ...
exportheightrange, uppercranklagrange, thetadegree, thetabdegree, ...
thesasdegreerrange, thetabdegreerrange, splugrange);
L = length(version);
version = str2num(version(L - 3:L));
specdata(110,:) = [neteff, thermal, inducedvol, frictionwork, version, ACR];

activerows = find(specdata(:,1) == 0);
specdata = specdata(activerows,:);
neteff = specdata(:,1) * 100;
inducedvol = specdata(:,3);
\[ \text{thermaleff} = \text{specdata}(:,2) \times 100; \]
\[ \text{frictionwork} = \text{specdata}(:,4); \]
\[ \text{version} = \text{specdata}(:,5); \]
\[ \text{ACR} = \text{specdata}(:,6); \]

\text{plot(inductedvol,neteff,'o',inductedvol,thermaleff,'s')}
\text{title('Efficiency Vs Inducted Volume for Various Eng Specs')}
\text{legend('Nett Eff','Thermal Eff')}

\text{text(22, max(thermaleff)+4.0,'Bore')}
\text{text(40, max(thermaleff)+4.0, '=' 61.3mm')}
\text{text(22, max(thermaleff)+3.0, 'E.O.Press')}
\text{text(40, max(thermaleff)+3.0, '=' 1 atm')}
\text{text(22, max(thermaleff)+2.0, 'Const Maximum')}
\text{text(22, max(thermaleff)+1.0, 'Pressure')}
\text{text(40, max(thermaleff)+1.0, '=' 6.5 MPa')}
\text{text(22, max(thermaleff)+0.0, 'Const Burn Start')}
\text{text(40, max(thermaleff)-1.0, '=' -27 deg')}

\text{text(50,35, 'A.C.R. = Actual Compression Ratio','Color',[0,0,1])}
\text{text(50,33.8, '(Max Inducted Vol/Min Compressed Vol)','Color',[0,0,1])}

\text{xlabel('Inducted Volume (cc)')}
\text{ylabel('Efficiency (%)')}
\text{axis([\text{min(inductedvol)}-15 \text{max(inductedvol)}+10 ... \text{min(thermaleff)}-5 \text{max(thermaleff)+5})];}

\text{coeffs=polyfit(inductedvol,neteff,4);
 xpoints=inductedvol(1):0.1:inductedvol(end);
 hold on
 polyvaldata=polyval(coeffs,xpoints);
 plot(xpoints,polyvaldata,'g');}

\text{coeffs2=polyfit(inductedvol,thermaleff,2);
 hold on
 polyval2data=polyval(coeffs2,xpoints);
 plot(xpoints,polyval2data,'Color',[0,0.5,0.5]);}

\text{[m,n] = size(specdata);}

\text{for counter=1:m;}
\quad \text{if counter==1}
\quad \quad \text{speclabel=sprintf('Spec%4.1f',version(counter));}
\quad \quad \text{ACRlabel=sprintf('A.C.R. %4.1f',ACR(counter));}
\quad \quad \text{text(inductedvol(counter),neteff(counter)+0.5,speclabel,'Color',[0,0.5,0.5],'Rotation',[90])}
\quad \quad \text{text(inductedvol(counter)+0.2,neteff(counter)-0.5,ACRlabel,'Color',[0,0,1],'Rotation',[-60])}
\quad \text{else}
\quad \quad \text{speclabel=sprintf('%4.1f',version(counter));}
\quad \quad \text{ACRlabel=sprintf('%4.1f',ACR(counter));}
\quad \quad \text{text(inductedvol(counter),neteff(counter)+0.5,speclabel,'Color',[0,0.5,0.5],'Rotation',[90])}
\quad \quad \text{text(inductedvol(counter)+0.2,neteff(counter)-0.5,ACRlabel,'Color',[0,0,1],'Rotation',[-60])}
\text{end}

122
% original constant max press data
% eff_disp_data = [34.9, 28.2, 7.9, 2.4, 2.0, 23.0;...
                  41.3, 42.3, 9.2, 2.6, 4.0, 39.0;...
                  42.5, 53.7, 9.7, 2.8, 5.0, 50.4;...
                  43.3, 58.9, 9.8, 2.9, 6.0, 57.6;...
                  43.4, 64.5, 10.1, 2.6, 6.2, 60.5;...
                  43.5, 69.3, 10.1, 6.5, 69.0;...
                  43.7, 74.0, 10.1, 7.0, 72.7;...
                  44.1, 78.3, 10.2, 7.0, 89.7;...
                  44.5, 83.1, 10.6, 2.9, 82.7;...
                  44.9, 86.5, 10.7, 9.0, 88.9;...
                  44.8, 91.0, 10.8, 9.2, 92.2;...
                  44.8, 94.3, 10.7, 9.5, 96.3;...
                  44.0, 103.5, 10.8, 10.0, 105.7;...
                  42.9, 108.3, 10.9, 3.2, 116.4];

figure
plot(inductedvol,frictionwork,'d')
title ('Friction Work Vs Inducted Volume for Various Eng Specs')
xlabel ('Inducted Volume (cc)')
ylabel ('Work (J)')
coeffs2=polyfit(inductedvol,frictionwork,2);
hold on
polyval2data=polyval(coeffs2,xpoints);
plot(xpoints,polyval2data,'Color',[0,0.5,0.5]);
axis([min(inductedvol)-15 max(inductedvol)+10 ...
     min(frictionwork)-5 max(frictionwork+5)]);
text (22,max(frictionwork)+2.0,'Bore' )
text (40,max(frictionwork)+2.0,'= 61.3mm')
text (22,max(frictionwork)+0.5,'E.O.Press')
text (40,max(frictionwork)+0.5,'= 1 atm')
text (22,max(frictionwork)-1,'Const Maximum')
text (22,max(frictionwork)-2.5,'Const Burn Start')
text (22,max(frictionwork)-5.5,'= -27 deg')

for counter=1:m;
    if counter==1
        speclabel=sprintf('Spec%4.1f',version(counter));
        text(inductedvol(counter),frictionwork(counter)+0.5,speclabel,'Color',[0,.5,0.5],'Rotation',[90])
    else
        speclabel=sprintf('%4.1f',version(counter));
        text(inductedvol(counter),frictionwork(counter)+0.5,speclabel,'Color',[0,.5,0.5],'Rotation',[90])
    end
end

% calculate conventional piston speed
dtheta=pi/180;
\[
\theta = \pi/2; \quad d\theta = 5 \pi/2;
\]
\[
\text{stroke} = 0.05;
\]
\[
\text{throw} = \text{stroke}/2
\]
\[
\text{conrod} = 0.07;
\]
\[
\text{RPM} = 5000;
\]
\[
\omega = \text{RPM} \pi/30;
\]
\[
\text{pistpos} = \sin(\theta) \text{throw} + \sqrt{\text{conrod}^2 - (\cos(\theta) \text{throw})^2}^2
\]
\[
\text{pistvel} = \frac{\text{diff(\text{pistpos})}}{d\theta/\omega};
\]
\[
\text{pistacc} = \frac{\text{diff(\text{pistvel})}}{d\theta/\omega}
\]
\[
\text{subplot}(3,1,1)
\]
\[
\text{plot}(1:361, \text{pistpos})
\]
\[
\text{title('pistpos')}
\]
\[
\text{maxpos} = \text{sprintf('max Vel = \%4.1f', \text{max(\text{pistpos})})};
\]
\[
\text{text}(1,1, \text{maxpos})
\]
\[
\text{subplot}(3,1,2)
\]
\[
\text{plot}(1:360, \text{pistvel})
\]
\[
\text{title('pistvel')}
\]
\[
\text{maxvel} = \text{sprintf('max Vel = \%4.1f', \text{max(\text{pistvel})})};
\]
\[
\text{text}(1,1, \text{maxvel})
\]
\[
\text{subplot}(3,1,3)
\]
\[
\text{plot}(1:359, \text{pistacc})
\]
\[
\text{title('pistacc')}
\]
\[
\text{maxacc} = \text{sprintf('max Acc = \%4.1f', \text{max(\text{pistacc})})};
\]
\[
\text{text}(1,1, \text{maxacc})
\]

ccp.m

function [h,u,v,s,Y,cp,dlvlT,dlvlp]=ecp(p,T,phi,fueltype,airscheme,Yguess);

% Routine to determine the equilibrium state of combustion products.
% Method closely follows that of:
% which uses the method described by:
% Calculating Properties of Equilibrium Combustion Products with
% Some Applications to I.C. Engines", SAE Paper 750468.
%
% % input:
% % p, T, phi - pressure (Pa), temperature (K), and equivalence ratio
% % fueltype - 'gasoline', 'diesel', etc - see fueldata.m for full list
% % airscheme - 'GMcB' (Gordon and McBride) or 'Chemkin'
% % Yguess - (optional) initial estimate for mole fractions of the
%
% % species CO2 H2O N2 O2 CO H2 O OH and NO
%
% % output:
% % h - enthalpy (J/kg), u - internal energy (J/kg),
% % v - specific volume (m^3/kg), s - entropy (J/kgK),
% % Y - mole fractions of 10 species, cp - specific heat (J/kgK),
% % dlvIT - partial derivative of log(v) wrt log(T)
% % dlvlp - partial derivative of log(v) wrt log(p)
%
[alpha,beta,gamma,delta,Afuel]=fueldata(fueltype);
switch airscheme
  case 'GMcB'
    A0=airdata('GMcB_hi');
  case 'Chemkin'
    A0=airdata('Chemkin_hi');
end
% Equilibrium constant data from Olikara and Borman via Ferguson
Kp=[0.432168E+00 -0.112464E+05 0.267269E+01 -0.745744E-04 0.242484E-08]
0.310805E+00 -0.129540E+05 0.321779E+01 -0.738336E-04 0.344645E-08
-0.141784E+00 -0.213308E+04 0.853461E+00 0.355015E-04 -0.310227E-08
0.150879E-01 -0.470959E+04 0.646096E+00 0.272805E-05 -0.154444E-08
-0.752364E+00 0.124210E+05 -0.260286E+01 0.259556E-03 -0.162687E-07
-0.415302E-02 0.148627E+05 -0.475746E+01 0.124699E-03 -0.900227E-08;
MinMol=1e-25;
tol=3e-12;
Ru=8314.34; % J/kmol.K
M=[44.01 18.02 32.000 28.008 32.000 16 17.009 30.004]'; % kg/kmol

% check if solid carbon will form
eps=0.210/(alpha+0.25*beta-0.5*gamma);
if phi>(0.210/eps/(0.5*alpha-0.5*gamma))
    error('phi too high - c(s) and other species will form');
end
if nargin==5 % no Yguess so estimate the composition using farg
    [h,u,v,s,Y,cp,dlvlT,dlvlp]=farg(p,T,phi,1,fueltype,airscheme);
    Y(7:10)=ones(4,1)*MinMol; % since farg only returns first 6 species
    if Y(6) 1 Tol
        phi=phi*(1+tol*sign(phi-1));
    end
    i=find(Y<MinMol);
    Y(i)=ones(length(i),1)*MinMol;
    DY3to6=2*tol*ones(4,1);
end
MaxIter=500;
MaxVal=max(abs(DY3to6));
Iter=0;
DoneSome=0;
while (Iter<MaxIter)&((MaxVal>tol)|(DoneSome<1))
    Iter=Iter+1;
    if Iter>2,
        DoneSome=1;
    end
D76=0.5*c(1)/sqrt(Y(6));
D84=0.5*c(2)/sqrt(Y(4));
D94=0.5*c(3)*sqrt(Y(6)/Y(4));
D96=0.5*c(3)*sqrt(Y(4)/Y(6));
D103=0.5*c(4)*sqrt(Y(4)/Y(3));
D104=0.5*c(4)*sqrt(Y(3)/Y(4));
D24=0.5*c(5)*sqrt(Y(4));
D26=c(5)*sqrt(Y(4));
D14=0.5*c(6)*sqrt(Y(4));
D15=c(6)*sqrt(Y(4));
A(1,1)=1+D103;
A(1,2)=D14+D24+1+D84+D104+D94;
A(1,3)=D15+1;
A(1,4)=D26+1+D76+D96;
A(2,1)=0;
A(2,2)=2*D24+D94-d(1)*D14;
A(2,3)=-d(1)*D15-d(1);
A(2,4)=2*D26+2+D76+D96;
A(3,1)=D103;
A(3,2)=2*D14+D24+2+D84+D94+D104-d(2)*D14;
A(3,3)=2*D15+1-d(2)*D15-d(2);
A(3,4)=D26+D96;
A(4,1)=2+D103;
A(4,2)=D104-d(3)*D14;
A(4,3)=-d(3)*D15-d(3);
A(4,4)=0;
% A
B(1)=-(sum(Y)-1);
B(2)=-(2*Y(2)+2*Y(6)+Y(7)+Y(9)-d(1)*Y(1)-d(1)*Y(5));
B(3)=-(2*Y(1)+Y(2)+2*Y(4)+Y(5)+Y(8)+Y(9)+Y(10)-d(2)*Y(1)-d(2)*Y(2)*Y(5));
B(4)=-(2*Y(3)+Y(10)-d(3)*Y(1)-d(3)*Y(5));
invA=inv(A);
DY3to6=invA*B;
MaxVal=max(abs(DY3to6));
Y(3:6)=Y(3:6)+DY3to6/10;
if find(Y<MinMol);
Y(i)=ones(length(i),1)*MinMol;
Y(7)=(1)*sqrt(Y(6));
Y(8)=(2)*sqrt(Y(4));
Y(9)=(3)*sqrt(Y(4)*Y(6));
Y(10)=(4)*sqrt(Y(4)*Y(3));
Y(2)=(5)*sqrt(Y(4))*Y(6);
Y(1)=(6)*sqrt(Y(4))*Y(5);
end
if Iter>=MaxIter
warning('convergence failure in composition loop');
end
TdKdT=[1/T -1/T^2 1 2*T];
dKdT=2.302585*K.*(Kp(:,[1 2 4 5])*TdKdT);
dcdT(1)=dKdT(1)/sqrt(patm);
dcdT(2)=dKdT(2)/sqrt(patm);
dcdT(3)=dKdT(3);
dcdT(4)=dKdT(4);
dcdT(5)=dKdT(5)*sqrt(patm);
dcdT(6)=dKdT(6)*sqrt(patm);
dcdp(1)=-0.5*c(1)/p;
dcdp(2)=-0.5*c(2)/p;
dcdp(5)=-0.5*c(5)/p;
dcdp(6)=-0.5*c(6)/p;
x1=Y(1)/c(6);
x2=Y(2)/c(5);
x7=Y(7)/c(1);
x8=Y(8)/c(2);
x9=Y(9)/c(3);
x10=Y(10)/c(4);
dfdT(1)=dcdT(6)*x1+dcdT(5)*x2+dcdT(1)*x7+dcdT(2)*x8+ ...
dcdT(3)*x9+dcdT(4)*x10;
dfdT(2)=2*dcdT(5)*x2+dcdT(1)*x7+dcdT(3)*x9-d(1)*dcdT(6)*x1;
dfdT(3)=2*dcdT(6)*x1+dcdT(5)*x2+dcdT(2)*x8+dcdT(3)*x9+ ...
dcdT(4)*x10-d(2)*dcdT(6)*x1;
dfdT(4)=dcdT(4)*x10-d(3)*dcdT(6)*x1;
dfdp(1)=dcdp(6)*x1+dcdp(5)*x2+dcdp(1)*x7+dcdp(2)*x8;
dfdp(2)=2*dcdp(5)*x2+dcdp(1)*x7-d(1)*dcdp(6)*x1;
dfdp(3)=2*dcdp(6)*x1+dcdp(5)*x2+dcdp(2)*x8-d(2)*dcdp(6)*x1;
dfdp(4)=d(3)*dcdp(6)*x1;
B=-dfdT;
DYdT(3:6)=invA*B;
DYdT(1)=sqrt(Y(4))*Y(5)*dcdT(6)+D14+dYdT(4)+D15*dYdT(5);
DYdT(2)=sqrt(Y(4))*Y(6)*dcdT(5)+D24*dYdT(4)+D26*dYdT(6);
DYdT(7)=sqrt(Y(6)*dcdT(1)+D76*dYdT(6);
DYdT(8)=sqrt(Y(4)*dcdT(2)+D84*dYdT(4);
126
dYdT(9)=\sqrt{Y(4)\cdot Y(6)}\cdot dcdT(3)+D94\cdot dYdT(4)+D96\cdot dYdT(6);
dYdT(10)=\sqrt{Y(4)\cdot Y(3)}\cdot dcdT(4)+D104\cdot dYdT(4)+D103\cdot dYdT(3);

B=-dfdp;
dYdp(3:6)=invA*B;
dYdp(1)=\sqrt{Y(4)\cdot Y(5)}\cdot dcdp(6)+D14\cdot dYdp(4)+D15\cdot dYdp(5);
dYdp(2)=\sqrt{Y(4)\cdot Y(6)}\cdot dcdp(5)+D24\cdot dYdp(4)+D26\cdot dYdp(6);
dYdp(7)=\sqrt{Y(6)}\cdot dcdp(2)+D76\cdot dYdp(6);
dYdp(8)=\sqrt{Y(4)\cdot Y(5)}\cdot dcdp(1)+D76\cdot dYdp(6);
dYdp(9)=D94\cdot dYdp(4)+D96\cdot dYdp(6);
dYdp(10)=D104\cdot dYdp(4)+D103\cdot dYdp(3);

% calculate thermodynamic properties
Tcp0=[1 T T^2 T^3 T^4];
Th0=[1 T/2 T^2/3 T^3/4 T^4/5 1/T];
Ts0=[log(T) T T^2/2 T^3/3 T^4/4 1];
cp0=A0(:,1:5)*Tcp0;
h0=A0(:,1:6)*Th0;
s0=A0(:,[1:5 7])*Ts0;

% Y(1) and Y(2) reevaluated
Y(1)=(2*Y(3)+Y(10))/d(3)-Y(5);
Y(2)=(d(1)/d(3)*(2*Y(3)+Y(10))-2*Y(6)-Y(7)-Y(9))/2;
i=find(Y<MinMol);
Y(i)=ones(length(i),1)*MinMol;

% properties of mixture
h=sum(h0.*Y);
cp=sum(Y.*cp0+h0.*dYdT*T);
MW=sum(Y.*M);
MT=sum(dYdT.*M);
Mp=sum(dYdp.*M);
R=Ru/MW;
v=R*T/p;
cp=R*(cp-h*T*MT/MW);
dvT=1-max(-T*MT/MW,0);
dvlp=-1-max(p*Mp/MW,0);
h=R*T*h;
s=R*(-log(patm)+s);
u=h-R*T;

enginedata.m

% enginedata.m
%
% Script file used by the function ahrind.m to
% define the engine properties and initial conditions
% ***** engine geometry ******************************************************

global b stroke eps r Cblowby f fueltype airscheme phi ...

% Volume remaining in spark plug thread after apark plug inserted (m^3)
splugvolume=1.6E-6; % (m^3) !!!!(original=0.8*10^-6: added 1.6 for piston chamber, 0.8 for pist clearance)!
muoil=1.5*1e-2;%oil viscosity (Ns/m^2)
mu=0.055; %coeff pof boundary layer friction
%engine crankcase volume (m^3)
volcase=7*10^-4;
r=10; % compression ratio -- set actual compression ratio A.C.R to 10
% Vtdc=pi/4*b^2*stroke/(r-1); % volume at TDC -- replaced with minV
% Vbdc=pi/4*b^2*stroke+Vtdc; % volume at BDC -- replaced with maxinV and
% maxexV
Vtdc=cylindervolume(compressedvolindex);
% ***** engine thermofluids parameters ***************************************
% Cblowby=8.0; % piston blowby constant (s^-1) (0.8 as standard)
f=0.05; % total residual fraction i.e. after spark plug ports exposed
fRC1=0.1; % total residual fraction i.e. after spark plug ports exposed , fRC1=0.025;
fueltype='gasoline';
airschema='GMcB';
phi=1.0; % equivalence ratio
% thetas=-20/180*pi; % start of burn relative to min compressed vol created in slider-bar-input
% RPM=1400;
omega=RPM*pi/30; % engine speed in rad/s
heattransferlaw='constant'; % 'constant', or 'Woschni'
hcu=500; % unburned zone heat transfer coefficient/weighting
hcb=500; % burned zone heat transfer coefficient/weighting
Tw=420; % engine surface temperature
% ***** initial conditions: *******************************************************
p1=100e3;
T1=370; %initial inducted gas temp
%theta1=-pi; theta1 created in 'pistonclash'
%V1=Vbdc;
[h1,a1,v1,s1,Y1,cp1,dlvlT1,dlvlp1]=farg(p1,T1,phi,f,fueltype,airschema);

masscase=volcase/v1;
mass1=maxinV/v1;
U1=u1*mass1;
% masslowerpist=0.300; % as for 60mm stroke
% massupperpist=0.260;
masslowerpist=0.450;% revert to above values of piston mass for general comparison of engine specs
massupperpist=0.450;% use measured values to assess comparison with prototype

enginespecoptsizechosen6.m - the engine specifications determined from the optimisation process to produce the maximum efficiency. This file is representative of all engine specification files.
% enter upper con rod length in mm
upperconrod=50.5/1000; %33.3
% enter upper crank throw in mm
uppercrankthrow=37.1/1000; %23
% enter lower crank offset in mm
crankoffset=-9.2/1000; %0
% enter lower con rod length in mm
conrod=48.0/1000; %32.5
% enter lower crank throw in mm
crankthrow=36.5/1000; %23
% enter crank to crank vertical height
cranktocrank=98.5/1000; %64
% enter exhaust and inlet port opening height
exportheight=12.5/1000; %8
% enter crank lag in degrees
uppercranklag=6.8; %
% enter burn start in degrees
thetasdeg=-25.5; %
% enter burn duration in degrees
thetabdeg=60; %
% Set range for slider-bars

uppercrankoffsetrange=[-40:0.2:0];
upperconrodrange=[40:0.2:80];
uppercrankthrowrange=[25:0.2:50];
crankoffsetrange=[-40:0.2:0];
conrodrange=[40:0.2:80];
crankthrowrange=[25:0.2:50];
cranktocrankrange=[80:0.2:100];
exportheightrange=[0:0.2:40];
uppercranklagrange=[-10:0.2:10];
thesasdegrange=[-50:0.02:-0];
thetabdegrange=[20:0.02:100];

enginespecprototype.m - the specifications achieved in the prototype and are the engine specifications used for the engine tests.

% Enter Engine configuration specifications

function [version,uppercrankoffset,upperconrod,uppercrankthrow,crankoffset,...
    conrod,crankthrow,cranktocrank,exportheight,uppercranklag,...
    uppercrankoffsetrange,upperconrodrange,uppercrankthrowrange,...
    crankoffsetrange,conrodrange,crankthrowrange,cranktocrankrange,...
    exportheightrange,uppercranklagrange,thetasdeg,thetabdeg,...
    thetasdegrange,thetabdegrange,splugport]=enginespecoptsizeselected4

version=sprintf('Prot 1 measured');
splugport=8;

% enter upper crank offset in mm
uppercrankoffset=-7.5/1000; %0
% enter upper con rod length in mm
upperconrod=44.5/1000; %33.3
% enter upper crank throw in mm
uppercrankthrow=26.3/1000; %23
% enter lower crank offset in mm
crankoffset=-7.4/1000; %0
% enter lower con rod length in mm
conrod=41.5/1000; %32.5
% enter lower crank throw in mm
crankthrow=25.5/1000;  %20
% enter crank to crank vertical height
cranktocrank=86.9/1000;  %64
% enter exhaust and inlet port opening height
exportheight=17.2/1000; %8
% enter crank lag in degrees
uppercranklag=0;  %
% enter burn start in degrees
thetasdeg=23;  %
% enter burn duration in degrees
thetabdeg=60;  %

% Set range for slider-bars
uppercrankoffsetrange=[-10:0.2:0];
upperconrodrange=[40:0.2:50];
uppercrankthrowrange=[20:0.2:30];
crankoffsetrange=[-10:0.2:0];
conrodrange=[40:0.2:50];
crankthrowrange=[20:0.2:30];
cranktocrankrange=[80:0.2:90];
exportheightrange=[0:0.2:20];
uppercranklagrange=[-10:0.2:10];
thesasdegrange=[-50:0.02:-10];
theatadegrange=[20:0.02:100];

farg.m

function [h,u,v,s,Y,cp,dlvlT,dlvlp]=farg(p,T,phi,f,fueltype,airscheme);
%
% [h,u,v,s,Y,cp,dlvlT,dlvlp]=farg(p,T,phi,f,fueltype,airscheme)
%
% Routine to determine the state of mixtures of fuel, air
% and residual combustion products at low temperatures.
% Method closely follows that of:
% who uses the results of:
% 2. Hires, S.D., Ekchian, A., Heywood, J.B., Tabaczynski, R.J., and
% Wall, J.C., 1976, "Performance and NOx Emissions Modeling of a Jet
% Ignition Pre-Chamber Stratified Charge Engine", SAE Trans., Vol 85,
% Paper 760161.
%********************************************************************
% input:
% p,T,phi - pressure (Pa), temperature (K), and equivalence ratio
% f - residual mass fraction; set f=0 if no combustion products
% are present and f=1 if only combustion products are present
% fueltype - 'gasoline', 'diesel', etc - see fueldata.m for full list
% airscheme - 'GMcB' (Gordon and McBride) or 'Chemkin'
% output:
% h - enthalpy (J/kg), u - internal energy (J/kg),
% v - specific volume (m^3/kg), s - entropy (J/kgK),
% Y - mole fractions of 6 species: CO2, H2O, N2, O2, CO, and H2,
% cp - specific heat (J/kgK),
% dlvIT - partial derivative of log(v) wrt log(T)
% dlvlp - partial derivative of log(v) wrt log(T)
%********************************************************************
[alpha,beta,gamma,delta,Afuel]=fueldata(fueltype);
switch airscheme
    case 'GMcB'
        A=airdata('GMcB_low');
    case 'Chemkin'
        A=airdata('Chemkin_low');
    end
Ru=8314.34; \% J/kmolK

\text{table}=[-1 1 0 0 1 -1];
\text{M}=[44.01 18.02 28.008 32.000 28.01 2.018]; \% kg/kmol
\text{MinMol}=1e-25;
\text{dlvT}=1; \text{dlvL}=1;
\text{eps}=0.210/(\text{alpha}+0.25*\beta-0.5*\gamma);
\text{if} \ \phi <= 1.0 \ % \text{stoichiometric or lean}
\text{nu}=[\text{alpha}^*\phi^*\text{eps} \ \text{beta}^*\phi^*\text{eps}/2 \ 0.79+\delta^*\text{phi}^*\text{eps}/2 \ ...
0.21^*(1-\phi) \ 0 \ 0]';
\text{dcdT}=0;
\text{else} \ % \text{rich}
z=1000/T;
\text{K}=\exp(2.743+z*(-1.761+z*(-1.611+z*0.2803)));
\text{dKdT}=\text{K}*(-1.761+z*(-3.222+z*0.8409))/1000;
a=1-K;
b=0.42-\phi^*\text{eps}*(2*\text{alpha}^*\gamma)+\text{K}^*(0.42^*(\phi-1)+\text{alpha}^*\text{phi}^*\text{eps});
c=-0.42*\alpha^*\phi^*\text{eps}^*^*(\phi-1)^*^K;\text{nu}=[(b-\text{sqrt}(b^2-4*a^*c))/2/a;
\text{dcdT}=-\text{dKdT}^*\text{nu}5^*2-\text{nu}5^*0.42^*\phi^*\text{eps}^*^*(2*\text{alpha}^*\gamma)+\text{nu}5 \ ...
0.79+\delta^*\text{phi}^*\text{eps}^/2 \ 0 \ \text{nu}5^*0.42^*(\phi-1)-\text{nu}5^*];
\text{end}

% mole fractions and molecular weight of residual
tmoles=sum(\text{nu});
\text{Y}=[\text{nu}']/\text{tmoles};
\text{Mres}=\sum(\text{Y}.*\text{M});
% mole fractions and molecular weight of fuel-air
\text{fuel}=[\text{eps}^*\phi^*^//(1+\text{eps}^*\phi^)];\text{o}_2=0.21^/\text{(1+}\text{eps}^*\phi^);\text{n}_2=0.79/(1+\text{eps}^*\phi^);\text{Maf}=[\text{fuel}^*^/(1.01^*\text{alpha}^+1.008^*\beta+16.000^*\gamma^+14.01^*\delta)+\ ...
32.04^28.02^*n_2];
% mole fractions of fuel-air-residual gas
\text{Yres}=\I^*(\text{fuel}^*\text{Maf}'/(\text{fuel}^*+\text{Mres}^*\text{Maf}^*^/\text{fuel}));\text{Y}^*\text{Yres};
\text{Yf}^*\text{fuel}^*\text{Maf}'\(\text{Yres})^*\text{Yres};
\text{Y}(3)^*\text{Y}(3)+\text{n}_2^*/(\text{1-Yres})^*\text{Y}(4)^*\text{Y}(4)+\text{n}_2^*\text{Y}(1-Yres);
% component properties
\text{Tcp0}=\begin{bmatrix} 1 & T & T^2 & T^3 & T^4 \end{bmatrix};
\text{Tc0}=\begin{bmatrix} 1 & T & T^2 & T^3 & T^4 \end{bmatrix};
\text{Th0}=\begin{bmatrix} 1 & T & T^2 & T^3 & T^4 & 1/\text{T} \end{bmatrix};
\text{c0}=\begin{bmatrix} 1 & T & T^2 & T^3 & 1/\text{T}^2 \end{bmatrix};
\text{h0}=\begin{bmatrix} 1 & T & T^2 & T^3 & 1/\text{T}^2 \end{bmatrix};
\text{s0}=\begin{bmatrix} 1 & T & T^2 & T^3 & 1/\text{T}^2 \end{bmatrix};
\text{cp0}=\begin{bmatrix} 1 & T & T^2 & T^3 & 1/\text{T}^2 \end{bmatrix};
\text{cp0}=\begin{bmatrix} 1 & T & T^2 & T^3 & 1/\text{T}^2 \end{bmatrix};
\text{h0}=\begin{bmatrix} 1 & T & T^2 & T^3 & 1/\text{T}^2 \end{bmatrix};
\text{s0}=\begin{bmatrix} 1 & T & T^2 & T^3 & 1/\text{T}^2 \end{bmatrix};
% set min value of composition so log calculations work
\text{if Yfuel}<\text{MinMol}
\text{Yfuel}^*\text{MinMol};
\text{end}
\text{Y}=\text{Y}^*\text{Yres};
\text{Yf}^*\text{fuel}^*\text{Maf}'\(\text{Yres})^*\text{Yres};
\text{Y}(3)^*\text{Y}(3)+\text{n}_2^*/(\text{1-Yres})^*\text{Y}(4)^*\text{Y}(4)+\text{n}_2^*\text{Y}(1-Yres);
% component properties
\text{Tcp0}=\begin{bmatrix} 1 & T & T^2 & T^3 & T^4 \end{bmatrix};
\text{Tc0}=\begin{bmatrix} 1 & T & T^2 & T^3 & T^4 \end{bmatrix};
\text{Th0}=\begin{bmatrix} 1 & T & T^2 & T^3 & T^4 & 1/\text{T} \end{bmatrix};
\text{c0}=\begin{bmatrix} 1 & T & T^2 & T^3 & 1/\text{T}^2 \end{bmatrix};
\text{h0}=\begin{bmatrix} 1 & T & T^2 & T^3 & 1/\text{T}^2 \end{bmatrix};
\text{s0}=\begin{bmatrix} 1 & T & T^2 & T^3 & 1/\text{T}^2 \end{bmatrix};
% properties of mixture
\text{h}=\text{hfuel}^*\text{Yfuel}^*\text{sum}(\text{h0}^*\text{Y});
\text{s}=\text{s0}^*\text{log(}\text{Yfuel})^*\text{Yfuel}^*\text{sum}(\text{s0}-\text{log(}\text{Y})).^*\text{Y});
\text{cp}^*\text{cp0}^*\text{Yfuel}^*\text{sum}(\text{cp0}^*^*\text{Y})^*\text{sum}(\text{h0}^*\text{table}^*\text{dcdT}^*\text{Yres}^*\text{tmoles});
\text{MW}^*\text{Maf}^*\text{Yfuel}^*\text{sum}(\text{Y}^*\text{M});
\text{R}^*\text{MW};
\text{h}=\text{R}^*\text{T}^*\text{h};
\text{u}=\text{h}-\text{R}^*\text{T}^*;
\[ v = R \cdot \frac{T}{p}; \]
\[ s = R \cdot (-\log\left(\frac{p}{101.325e3}\right) + s); \]
\[ cp = R \cdot cp; \]

**ferguson.txt**

-180. 698. 0. 1.00 NaN 350. 0. 0.714 0.
-170. 695. 0. 1.01 NaN 353. -0. -1. 0.714 -0.14
-160. 684. 0. 1.04 NaN 357. -1. -2. 0.713 -0.28
-150. 666. 0. 1.08 NaN 362. -3. -4. 0.713 -0.42
-140. 641. 0. 1.14 NaN 369. -4. -5. 0.712 -0.56
-130. 609. 0. 1.23 NaN 377. -10. -5. 0.712 -0.69
-120. 571. 0. 1.35 NaN 386. -15. -6. 0.711 -0.81
-110. 527. 0. 1.50 NaN 398. -21. -6. 0.711 -0.93
-100. 477. 0. 1.72 NaN 413. -29. -7. 0.710 -1.05
-90. 424. 0. 2.01 NaN 430. -39. -7. 0.710 -1.15
-80. 368. 0. 2.43 NaN 451. -67. -7. 0.712 -1.25
-70. 312. 0. 3.02 NaN 476. -67. -6. 0.709 -1.34
-60. 257. 0. 3.91 NaN 507. -21. -6. 0.708 -1.46
-50. 205. 0. 5.24 NaN 544. 109. -4. 0.707 -1.46
-40. 160. 0. 7.28 NaN 588. 137. -2. 0.708 -1.49
-30. 122. 0.017 10.90 2143. 647. -170. -0. 0.707 -1.49
-20. 93. 0.146 20.93 2296. 752. -212. 5. 0.707 -1.45
-10. 76. 0.371 38.59 2439. 863. -262. 14. 0.706 -1.34
0. 70. 0.629 56.28 2514. 936. -289. 27. 0.706 -1.15
10. 76. 0.854 61.31 2497. 952. -253. 41. 0.705 -0.91
20. 93. 0.983 52.13 2400. 916. -152. 57. 0.705 -0.68
30. 122. 1.000 37.80 2248. 26. 73. 0.704 -0.53
40. 160. 1.000 26.80 2091. 23. 94. 0.704 -0.50
50. 205. 1.000 19.40 1948. 21. 57. 0.703 -0.58
60. 257. 1.000 14.51 1822. 22. 104. 0.703 -0.75
70. 312. 1.000 11.24 1714. 22. 134. 0.702 -0.99
80. 368. 1.000 8.99 1621. 22. 172. 0.702 -1.30
90. 424. 1.000 7.42 1541. 22. 214. 0.702 -1.67
100. 477. 1.000 6.29 1472. 22. 262. 0.701 -1.67
110. 527. 1.000 5.47 1412. 22. 319. 0.701 -2.09
120. 571. 1.000 4.86 1360. 22. 383. 0.700 -2.54
130. 609. 1.000 4.40 1315. 22. 455. 0.700 -3.03
140. 641. 1.000 4.05 1276. 24. 530. 0.699 -4.10
150. 666. 1.000 3.79 1241. 25. 616. 0.699 -4.67
160. 684. 1.000 3.60 1212. 25. 712. 0.698 -5.26
170. 695. 1.000 3.47 1186. 25. 818. 0.698 -5.86
180. 698. 1.000 3.39 1165. 25. 925. 0.697 -6.48

**fueldata.m**

function [alpha,beta,gamma,delta,Afuel]=fueldata(fuel);

```
function [alpha,beta,gamma,delta,Afuel]=fueldata(fuel);

function [alpha,beta,gamma,delta,Afuel]=fueldata(fuel)

% Routine to specify the thermodynamic properties of a fuel.
% Data taken from:
% McGraw-Hill; and

% ********************************************************************
% input:
% fuel switch
% from Ferguson: 'gasoline', 'diesel', 'methane', 'methanol',
% 'nitromethane', 'benzene';
```
% from Heywood: 'methane_h', 'propane', 'hexane', 'isooctane_h',
% 'methanol_h', 'ethanol', 'gasoline_h1', gasoline_h2', 'diesel_h';
% from Raine: 'toluene', 'isooctane'.
% output:
% alpha, beta, gamma, delta - number of C, H, O, and N atoms
% Afuel - vector of polynomial coefficients for cp/R, h/RT, and s/R
% of the form h/RT=a1+a2*T/2+a3*T^2/3+a4*T^3/4-a5/T^2+a6/T (for
% example) where T is expressed in K.
% ************************************************************************

% Set values for conversion of Heywood data to nondimensional format
% with T expressed in K
SVal=4.184e3/8.31434;
SVec=SVal*[1e-3 1e-6 1e-9 1e-12 1e3 1 1];
switch fuel
case 'gasoline' % Ferguson
alpha=7; beta=17; gamma=0; delta=0;
Afuel=[4.0652 6.0977E-02 -1.8801E-05 0 0 -3.5880E+04 15.45];
case 'diesel' % Ferguson
alpha=14.4; beta=24.9; gamma=0; delta=0;
Afuel=[7.9710 1.1954E-01 -3.6858E-05 0 0 -1.9385E+04 -1.7879];
case 'methane' % Ferguson
alpha=1; beta=4; gamma=0; delta=0;
Afuel=[1.971324 7.871586E-03 -1.048592E-06 0 0 -9.30422E+03 8.873728];
case 'methanol' % Ferguson
alpha=1; beta=4; gamma=1; delta=0;
Afuel=[1.779819 1.262503E-02 -3.624890E-06 0 0 -2.525420E+04 1.50884E+01];
case 'nitromethane' % Ferguson
alpha=1; beta=3; gamma=2; delta=1;
Afuel=[1.412633 2.087101E-02 -8.142134E-06 0 0 -1.026351E+04 1.917126E+01];
case 'benzene' % Ferguson
alpha=6; beta=6; gamma=0; delta=0;
Afuel=[2.545087 4.79554E-02 -2.03765E-05 0 0 8.782234E+03 3.348825E+01];
case 'toluene' % Raine
alpha=7; beta=8; gamma=0; delta=0;
Afuel=[2.09053 5.654331E-2 -2.350992E-5 0 0 4.331.441411 34.55418257];
case 'isooctane' % Raine
alpha=8; beta=18; gamma=0; delta=0;
Afuel=[6.678E-1 8.398E-2 -3.334E-5 0 0 -3.058E+4 2.351E+1];
case 'methane_h' % Heywood
alpha=1; beta=4; gamma=0; delta=0;
Afuel=[-0.29149 26.327 -10.610 1.5656 0.16573 -18.331 19.9887/SVal].*SVec;
case 'propane' % Heywood
alpha=3; beta=8; gamma=0; delta=0;
Afuel=[6.678E-1 8.398E-2 -3.334E-5 0 0 -3.058E+4 2.351E+1];
case 'hexane' % Heywood
alpha=6; beta=14; gamma=0; delta=0;
Afuel=[-20.777 210.48 -164.125 52.832 0.56635 -39.836 79.5542/SVal].*SVec;
case 'isooctane_h' % Heywood
alpha=8; beta=18; gamma=0; delta=0;
Afuel=[-0.55313 181.62 -97.787 20.402 -0.03095 -60.751 27.2162/SVal].*SVec;
case 'methanol_h' % Heywood
alpha=1; beta=4; gamma=1; delta=0;
Afuel=[-2.7059 44.168 -27.501 7.2193 0.20299 -48.288 31.1406/SVal].*SVec;
case 'ethanol' % Heywood
alpha=2; beta=6; gamma=1; delta=0;
Afuel=[6.990 39.741 -11.926 0 0 -60.214 8.01623/SVal].*SVec;
case 'gasoline_h1' % Heywood
alpha=8.26; beta=15.5; gamma=0; delta=0;
Afuel=[-24.078 256.63 -201.68 64.750 0.5808 -27.562 NaN/SVal].*SVec;
case 'gasoline_h2' % Heywood
alpha=7.76; beta=13.1; gamma=0; delta=0;
Afuel=[-22.501 227.99 -177.26 56.048 0.4845 -17.578 NaN/SVec;
case 'diesel_h' % Heywood
alpha=10.8; beta=18.7; gamma=0; delta=0;
Afuel=[-9.1063 246.97 -143.74 32.329 0.0518 -50.128 NaN/SVec;
lowerconrodinclination.m

function
lowerconrodangle=lowerconrodinclination(lower_pist_positionmm,thetacorrected,crankoffsetmm,conrodmm,crankthrowmm,omega)

% function to calculate the lower con rod inclination, lowerconrodangle in radians.
smallendmm=lower_pist_positionmm;
xbigendmm=cos(thetacorrected)*crankthrowmm+crankoffsetmm;
ybigendmm=sin(thetacorrected)*crankthrowmm;
lowerconrodangle=acos(-xbigendmm/conrodmm);

lowerpistfriction.m

function
[lowerpistfrictwork,dSlowerdt,d2Slowerdt2,lowerconrodangle,conrodtensionlower,normallower,fAlower,...
lowerpistfrictionvar,lowerpistfrictworkall,StNolower]¼...
lowerpistfriction(engpressure,lower_pist_positionmm,thetacorrected,...
lowercrankoffsetmm,lowerconrodmm,lowercrankthrowmm,omega,masslowerpist,b,thetaworkcycleindex,...
downstroke,muoil,mu,lowercranklag,cranktocrankmm)

% Function to determine the side thrust on the lowerer piston by calculating
% the pressure in the cylinder, the con rod inclination and the piston
% acceleration.
% Assumes constant crankshaft speed

casepress=0.8*1e5;

% calc piston velocity
dSlowerdt=diff(lower_pist_positionmm/1000)/(pi/180/omega);
% addelement=find(dSlowerdt==min(dSlowerdt));
% dSlowerdt=[dSlowerdt(1:addelement),min(dSlowerdt),dSlowerdt(addelement+1:end)];
dSlowerdt = spline(1:length(dSlowerdt),dSlowerdt,linspace(1,length(dSlowerdt),360));

% calc piston acceleration
d2Slowerdt2=diff(dSlowerdt2)/(pi/180/omega);
% repeat first element to return vector to original length
d2Slowerdt2 = spline(1:length(d2Slowerdt2),d2Slowerdt2,linspace(1,length(d2Slowerdt2),360));
%d2Slowerdt2=[d2Slowerdt2(1),d2Slowerdt2]

lowerconrodangle=lowerconrodinclination(lower_pist_positionmm,thetacorrected,...
lowercrankoffsetmm,lowerconrodmm,lowercrankthrowmm,omega);

%calc stroke for this application
stroke=(max(lower_pist_positionmm)-min(lower_pist_positionmm))/1000

%masslowerpist=(0.7+stroke/.09*0.3)*masslowerpist
masslowerpist

%Force in y-direction =acc in y-direction*piston mass
%Fy=d2Slowerdt2*masslowerpist;

% Force on piston (Fy) = force in con rod (Fconrod)*sin(lowerconrodangle) -
% pressure in cylinder*area - friction ASSUME friction is hydrodynamic for piston speeds over 1m/s.
% ASSUME boundary friction (metal to metal) for speed < 1m/s
% with coeff of friction =0.05

% As an initial estimate of normal force (to give bearing pressure) use
% mu=0.01 for entire piston motion, therefore

% Loop the friction calc using previously determined Sommerfield numbers n times

n=50;
fA=.001*ones(1,length(lowerconrodangle));
for k=1:n;
    ftemp=fA;
    for counterA=1:length(lowerconrodangle);
        if dSlowerdt(counterA)<0
            lowerpistfrictionvar(counterA)=abs((masslowerpist*d2Slowerdt2(counterA)-(engpressure(counterA)-casepress)*pi*(b/2)^2)/...
                (abs(tan(lowerconrodangle(counterA)))/ftemp(counterA)+1));
        else
            lowerpistfrictionvar(counterA)=-abs((masslowerpist*d2Slowerdt2(counterA)-(engpressure(counterA)-casepress)*pi*(b/2)^2)/...
                (abs(tan(lowerconrodangle(counterA)))/ftemp(counterA)-1));
        end
    end
end

% calculate normal force from coeff friction and cylinder projected area

normallower=lowerpistfrictionvar./ftemp;

% determine Strubeck relation between # and coeff of friction (f) from plot
% use as input into Strubeck calculation

graphStNoR=[-5.8 -3];
graphfR=[-3 -1.3];
StNoRfit=polyfit(graphStNoR,graphfR,1);

graphStNoM=[-6.8 -6.5 -6.41 -6.1 -5.8];
graphfM=[-1.3 -1.5 -2.1 -2.85 -3];
StNoMfit=polyfit(graphStNoM,graphfM,3);

% calc Strubeck number

StNo=muoil*abs(dSlowerdt)./(abs(normallower)/(b*stroke));

% determine corresponding coeff of friction

fR=10.^(polyval((StNoRfit),log10(StNo)));
fM=10.^(polyval((StNoMfit),log10(StNo)));
for counterB=1:length(lowerconrodangle);
    if StNo(counterB)<=10^-6.8 & dSlowerdt(counterB)<=0
        lowerpistfrictionvar(counterB)=abs((masslowerpist*d2Slowerdt2(counterB)-(engpressure(counterB)-casepress)*pi*(b/2)^2)/...
            (abs(tan(lowerconrodangle(counterB)))/(mu+1)));    
    elseif StNo(counterB)<=10^-6.8 & dSlowerdt(counterB)>0
        lowerpistfrictionvar(counterB)=-abs((masslowerpist*d2Slowerdt2(counterB)-(engpressure(counterB)-casepress)*pi*(b/2)^2)/...
            (abs(tan(lowerconrodangle(counterB)))/(mu+1)));
    elseif StNo(counterB)>10^-5.8 & StNo(counterB)>10^-6.8 & dSlowerdt(counterB)<=0
        lowerpistfrictionvar(counterB)=-abs((masslowerpist*d2Slowerdt2(counterB)-(engpressure(counterB)-casepress)*pi*(b/2)^2)/...
            (abs(tan(lowerconrodangle(counterB)))/(mu-1)));
    elseif StNo(counterB)>10^-5.8 & StNo(counterB)>10^-6.8 & dSlowerdt(counterB)>0
        lowerpistfrictionvar(counterB)=abs((masslowerpist*d2Slowerdt2(counterB)-(engpressure(counterB)-casepress)*pi*(b/2)^2)/...
            (abs(tan(lowerconrodangle(counterB)))/(mu-1)));
    elseif StNo(counterB)>10^-5.8 & StNo(counterB)>10^-6.8 & dSlowerdt(counterB)>0
        lowerpistfrictionvar(counterB)=abs((masslowerpist*d2Slowerdt2(counterB)-(engpressure(counterB)-casepress)*pi*(b/2)^2)/...
            (abs(tan(lowerconrodangle(counterB)))/(mu+1)));
    elseif StNo(counterB)<10^-3.5 & StNo(counterB)>10^-5.8 & dSlowerdt(counterB)<10^-6.8 & dSlowerdt(counterB)<=0
        lowerpistfrictionvar(counterB)=abs((masslowerpist*d2Slowerdt2(counterB)-(engpressure(counterB)-casepress)*pi*(b/2)^2)/...
            (abs(tan(lowerconrodangle(counterB)))/(mu+1)));
    elseif StNo(counterB)<10^-3.5 & StNo(counterB)>10^-5.8 & dSlowerdt(counterB)>0
        lowerpistfrictionvar(counterB)=-abs((masslowerpist*d2Slowerdt2(counterB)-(engpressure(counterB)-casepress)*pi*(b/2)^2)/...
            (abs(tan(lowerconrodangle(counterB)))/(mu+1)));
elseif StNo(counterB)<10^-3.5 & StNo(counterB)>10^-5.8 & dSlowerdt(counterB)>0
    lowerpistfrictionvar(counterB)=abs((masslowerpist*d2Slowerdt2(counterB)-(engpressure(counterB)-casepress)*\pi*(b/2)^2)/...}
    (abs(tan(lowerconrodangle(counterB)))/fR(counterB)-1)) ;
    end
end

% Obtain lower piston position increments with 'diff'
for counterC=1:length(lowerconrodangle);
    if StNo(counterC)<=10^-6.8
        normallower(counterC)=lowerpistfrictionvar(counterC)/mu;
        fA(counterC)=mu;
    elseif StNo(counterC)<=10^-5.8
        normallower(counterC)=lowerpistfrictionvar(counterC)/fM(counterC);
        fA(counterC)=fM(counterC);
    else
        normallower(counterC)=lowerpistfrictionvar(counterC)/fR(counterC);
        fA(counterC)=fR(counterC);
    end
end

% Calc conrod compress/tension load (tension=positive)
conrodtensionlower=(-masslowerpist*d2Slowerdt2-(engpressure-casepress)*\pi*(b/2)^2+...}
    lowerpistfrictionvar)/sin(lowerconrodangle);

% normallower./cos(lowerconrodangle);
% lowerpistfrictionvar
lowerringfriction.m

function
[lowerringfrictwork]=lowerringfriction(dSlowerdt,d2Slowerdt2,lower_pist_positionmm,b,muoil,mu,engpressure);

lowerringtension=20;
ringwidth=0.0023;

% determine Striebeck relation between # and coeff of friction (f) from plot
% use as input into Striebeck calculation
graphStNoR=[-5.8 -3];
graphfR=[-3 -1.3];
StNofitR=polyfit(graphStNoR,graphfR,1);

graphStNoM=[-6.8 -6.5 -6.41 -6.1 -5.8];
graphfM=[-1.3 -1.5 -2.1 -2.85 -3];
StNofitM=polyfit(graphStNoM,graphfM,3);

% calc Striebeck number
StNo=muoil*abs(dSlowerdt)./(lowerringtension/(b*pi*ringwidth)+engpressure);
StNo2=muoil*abs(dSlowerdt)./(lowerringtension/(b*pi*ringwidth)+engpressure/2);

%determine corresponding coeff of friction
fR=10.^(polyval((StNofitR),log10(StNo)));
fM=10.^(polyval((StNofitM),log10(StNo)));

% obtain lowerer piston position increments with 'diff'
lower_pist_position_diff=diff(lower_pist_positionmm/1000);
lower_pist_position_diff=spline(1:length(lower_pist_position_diff),lower_pist_position_diff,...
    linspace(1,length(lower_pist_position_diff),360));
for counter=1:length(engpressure);
    if StNo(counter)<=10^-6.8
lowerringfriction(counter)=mu*(lowerringtension+engpressure(counter)*pi*b*ringwidth).*sign(dSlowerdt(counter));
    elseif StNo(counter)<=10^-5.8
lowerringfriction(counter)=fM(counter)*(lowerringtension+engpressure(counter)*pi*b*ringwidth).*sign(dSlowerdt(counter));
    else
lowerringfriction(counter)=fR(counter)*(lowerringtension+engpressure(counter)*pi*b*ringwidth).*sign(dSlowerdt(counter));
    end
end
for counter2=1:length(engpressure);
    if StNo(counter2)<=10^-6.8
lowerringfriction2(counter2)=mu*(lowerringtension+engpressure(counter2)*pi*b*ringwidth).*sign(dSlowerdt(counter2));
    elseif StNo(counter2)<=10^-5.8
lowerringfriction2(counter2)=fM(counter2)*(lowerringtension+engpressure(counter2)*pi*b*ringwidth).*sign(dSlowerdt(counter2));
    else
lowerringfriction2(counter2)=fR(counter2)*(lowerringtension+engpressure(counter2)*pi*b*ringwidth).*sign(dSlowerdt(counter2));
    end
end
lowerringfrictworkall=abs(lowerringfriction).*abs(lower_pist_position_diff);
lowerringfrictworkall2=abs(lowerringfriction2).*abs(lower_pist_position_diff);
lowerringfrictwork=sum(lowerringfrictworkall)+sum(lowerringfrictworkall2);

pistfriction.m

function [pistfrictwork,dSdt,d2Sdt2,conrodangle,conrodtension,normal]=...
pistfriction(engpressure,_pist_positionmm,thetacorrected,...
crankoffsetmm,conrodm,crankthrowmm,omega,masspist,b,thetaworkcycleindex,downstroke,muoil,mu)

% Function to determine the side thrust on the er piston by calculating
% the pressure in the cylinder, the con rod inclination and the piston
% acceleration.
% Assumes constant crankshaft speed
%
% calc piston velocity
dSdt=diff(_pist_positionmm/1000)/(pi/180/omega);
adddelement=find(dSdt==min(dSdt));
dSdt=[dSdt(1:adddelement),min(dSdt),dSdt(adddelement+1:end)];
%
% calc piston acceleration
d2Sdt2=diff(dSdt)/(pi/180/omega);
%
% repeat first and last element to return vector to original length
d2Sdt2=[d2Sdt2(1),d2Sdt2];

conrodangle=conrodinclination(_pist_positionmm,thetacorrected,crankoffsetmm,conrodm,crankthrowmm,omega);
%
% calc stroke for this application
stroke=(max(_pist_positionmm)-min(_pist_positionmm))/1000;
%
% Force in y-direction =acc in y-direction*piston mass
% Fy=d2Sdt2*masspist;
%
% Force on piston (Fy) = force in con rod (Fconrod)*sin(conrodangle) -
% pressure in cylinder*area - friction ASSUME friction is hydrodynamic for piston speeds over 1m/s.
% ASSUME boundary friction (metal to metal) for speed < 1m/s
% with coeff of friction =0.05
%
% As an initial estimate of normal force (to give bearing pressure) use
% mu=0.01 for entire piston motion, therefore
%
% Loop the friction calc using previously determined Sommerfield numbers n
times
%

n=5;

for k=1:n;
    if k==1;
        mutemp=0.01*ones(1,length(conrodangle));
    else mutemp=f;
    end
    for counter=1:length(conrodangle);
        if dSdt(counter)<0
            pistfrictionvar(counter)=(masspist*d2Sdt2(counter)+engpressure(counter)*pi*(b/2)^2)/...
            (sin(conrodangle(counter))/(mutemp(counter)*cos(conrodangle(counter))))-1);
        else
            pistfrictionvar(counter)=(masspist*d2Sdt2(counter)+engpressure(counter)*pi*(b/2)^2)/...
            (sin(conrodangle(counter))/(mutemp(counter)*cos(conrodangle(counter))))+1);
        end
    end
end
% calculate normal force from coeff friction and cylinder projected area
normal=pistfrictionvar./mutemp;

% determine Strubeck relation between # and coeff of friction (f) from plot
% use as input into Strubeck calculation
graphStNo=[-6 -3];
graphf=[-3 -1.3];
StNofit=polyfit(graphStNo,graphf,1);

% calc Strubeck number
StNo=muoil*abs(dSdt)./(abs(normal)/(b*pi*(stroke)));

% determine corresponding coeff of friction
f=10.^(polyval((StNofit),log10(StNo)));
for counter=1:length(conrodangle);
    if StNo(counter)>0 & StNo(counter)<=0.01*1e-4 & dSdt(counter)<=0
        pistfrictionvar(counter)=(masspist*d2Sdt2(counter)+engpressure(counter)*pi*(b/2)^2)/...
            (sin(conrodangle(counter))/(mu*cos(conrodangle(counter))-1));
    elseif StNo(counter)>0 & StNo(counter)<=0.01*1e-4 & dSdt(counter)>0
        pistfrictionvar(counter)=(masspist*d2Sdt2(counter)+engpressure(counter)*pi*(b/2)^2)/...
            (sin(conrodangle(counter))/(mu*cos(conrodangle(counter)))+1);
    elseif StNo(counter)>0.01*1e-4 & dSdt(counter)<=0
        pistfrictionvar(counter)=(masspist*d2Sdt2(counter)+engpressure(counter)*pi*(b/2)^2)/...
            (sin(conrodangle(counter))/(f(counter)*cos(conrodangle(counter)))-1);
    else
        pistfrictionvar(counter)=(masspist*d2Sdt2(counter)+engpressure(counter)*pi*(b/2)^2)/...
            (sin(conrodangle(counter))/(f(counter)*cos(conrodangle(counter)))+1);
    end
end

% obtain piston position increments with 'diff'
_pist_position_diff=diff(_pist_positionmm/1000);
_pist_position_diff=[_pist_position_diff,_pist_position_diff(end)];

% pistfrictworkall=abs(pistfrictionvar).*abs(_pist_position_diff);
% pistfrictwork=sum(pistfrictworkall);
% calc conrod compress/tension load (tension=positive)
for counter=1:length(conrodangle);
    if StNo(counter)<0.01*1e-4
        normal(counter)=pistfrictionvar(counter)/mu;
    else
        normal(counter)=pistfrictionvar(counter)./f(counter);
    end
end
conrodtension=normal./cos(conrodangle);
% pistfrictionvar

% alternative friction load if hyrdraudynamic lubrication is sustained
% throughout the piston motion - assumes the film thickness varies between
% 0.03 and 1.0 micrometres linearly with speed^1.2 i.e. (s/20)^1.2*1e-6
% For speeds under 1m/s use boundary layer friction
% use first principle hydrodynamic theory shear stress=viscosity*du/dy
% assume oil viscosity is for engine surface temperature = 100 deg C , use
% 1*10^-3 Ns/m^2
boundaryindex=find(StNo<0.01*1e-4);  \%| StNo>0.01*1e-4
% assess the piston speed at current Strubeck transition

% dSdtStNo=zeros(1,length(conrodangle));
% dSdtStNo(boundaryindex)=StNo[boundaryindex];
% newStNocutoffindex=min(find(dSdtStNo==0));
% StNocutoffspeed=dSdt(newStNocutoffindex)
StNocutoffspeed=min(dSdt(boundaryindex));

pistfrictioncombined=zeros(1,length(pistfrictionvar));
pistfrictioncombined(boundaryindex)=pistfrictionvar(boundaryindex);
end

% for counter=1:length(pistfrictioncombined);
% if pistfrictioncombined(counter)\= 0
%     pistfrictioncombined(counter)=f(counter)*normal(counter);
% end
% end

pistfrictworkall=abs(pistfrictioncombined).*abs(_pist_position_diff);
pistfrictwork=sum(pistfrictworkall);

pistonclash.m

function [pistonclashmm,maxinvolmm,compressedvolmm,CR,maxexpandvolmm,...
piston_to_pistonmm,thetaignition]=...
pistonclash(upper_pist_positionmm, lower_pist_positionmm,exportheightmm)

global thetas thetab omega
global heattransferlaw hcu hcb
global Tw theta1 Vtdc Vbdc mass1
global p1 T1 V1 minV maxinV maxexV

global cylindervolume crankoffset conrod
global crankthrow uppercrankoffset upperconrod
global uppercrankthrow cranktocrank exportheight
global uppercranklag thetacorrected maxinvolindex
global maxexpandvolindex thetaignition upstroke downstroke
global b stroke eps r Cblowby f fueltype airscheme phi
global compressedvolindex plotindex portopenindex

piston_to_pistonmm=(upper_pist_positionmm-lower_pist_positionmm);
%determine exhaust port opening
portcloseindex=max(find(lower_pist_positionmm < exportheightmm));
portopenindex=10+max(find(lower_pist_positionmm(10:300) > exportheightmm));

if isempty(portcloseindex)
    portcloseindex=180
end

increments=length(upper_pist_positionmm);
pistonclashrange=increments/10;
pistonclashmm=min(piston_to_pistonmm(portcloseindex-pistonclashrange:portcloseindex));
maxinvolmm=max(piston_to_pistonmm(1:increments/3));

maxinvolindex=max(find(piston_to_pistonmm==maxinvolmm));
compressedvolmm=min(piston_to_pistonmm(maxinvolindex:maxinvolindex+180));
compressedvolindex=max(find(piston_to_pistonmm==compressedvolmm));
CR=maxinvolmm/compressedvolmm;
maxexpandvolmm=max(piston_to_pistonmm(1:portcloseindex));
maxexpandvolindex=max(find(piston_to_pistonmm==maxexpandvolmm));
%ER=maxexpandvol/compressedvol;
cylindervolume=(upper_pist_positionmm-lower_pist_positionmm)/1000*(b/2)^2*pi;
maxinV=cylindervolume(maxinvolindex);
minV=cylindervolume(compressedvolindex);
maxexV=cylindervolume(maxexpandvolindex);
upstroke=real((cylindervolume(maxinvolindex)-cylindervolume(compressedvolindex))/(pi*(b/2)^2));
downstroke=real((cylindervolume(maxexpandvolindex)-cylindervolume(compressedvolindex))/(pi*(b/2)^2));
thetaignition=(compressedvolindex+thetas/pi*180)/180*pi; %-35*pi/180; % start of burning relative to plot crank angles
theta1=maxinvolindex/180*pi;
V1=maxinV;
%determine exhaust port opening

plot_both_conrod_inclination.m
% M-file to plot both conrod inclination versus crank angle after exhaust port % close
plot(1:360,upperconrodangle/pi*180,1:360,lowerconrodangle/pi*180)
text(45,20,'Spec')
text(75,20,version)
title ('Conrod Inclination Vs Crank Angle after E.C.')
xlabel ('Crank Angle after E.C. (Deg)')
ylabel ('Inclination (deg (big end as origin reference))')
line ([1 360],[90 90])
line ([1 360],[270 270])
set(gca,'XTick',[-35 90 180 270 360]);
set(gca,'XTickLabel',[-35 90 180 270 360]);
legend('Upper','Lower',0)

plot_both_pist_friction.m
% M-file to plot both piston friction versus crank angle after exhaust port % close
plot(1:360,upperpistfrictionvar,1:360,lowerpistfrictionvar)
title ('Piston Friction Vs Crank Angle after E.C.')
xlabel ('Crank Angle after E.C. (Deg)')
ylabel ('Friction (N)(temporal)')
minvar=[min(upperpistfrictionvar),min(lowerpistfrictionvar)];
maxvar=[max(upperpistfrictionvar),max(lowerpistfrictionvar)];
axis([0 360 -50 100]);
set(gca,'XTick',[0 90 180 270 360]);
set(gca,'XTickLabel',[0 90 180 270 360]);

% M-file to plot both piston acceleration versus crank angle after exhaust port
plot(1:360,d2Supperdt2,1:360,d2Slowerd2)

% M-file to plot both piston velocity versus crank angle after exhaust port
plot(1:360,dSupperdt,1:360,dSlowerdt)
plot_both_StNo.m
% M-file to plot both Sommerfield number versus crank angle after exhaust port
% close
plot(1:360,StNoupper,1:360,StNolower)

plot_both_StNo4.m
% M-file to plot both Sommerfield number under 5e-4 versus crank angle after exhaust port
% close
plot(1:360,StNoupper4,1:360,StNolower4)

plot_both_StNo_10toneg5_8.m
% M-file to plot both Sommerfield number under 10^-5.8 versus crank angle after exhaust port
% close
plot(1:360,StNoupper,1:360,StNolower)
maxStNo=[max(StNoupper),max(StNolower)];
axis([0 360 -0.5e-6 10^-5.8]);
set(gca,'XTick',[0 90 180 270 360]);
set(gca,'XTickLabel',[0 90 180 270 360]);
text(20,0,'Spec')
text(50,0,version)
legend('Upper','Lower',0)

plot_both_StNo_10toneg6_8.m

% M-file to plot both Sommerfield number under 10^-6.8 versus crank angle after exhaust port
% close
plot(1:360,StNoupper,1:360,StNolower)
title ('Sommerfield Number Vs Crank Angle after E.C.')</xlabel ('Crank Angle after E.C. (Deg)')
ylabel ('Sommerfield Number')
minStNos=[min(StNoupper),min(StNolower)];
maxStNo=[max(StNoupper),max(StNolower)];
axis([0 360 -0.05e-6 10^-6.8]);
set(gca,'XTick',[0 90 180 270 360]);
set(gca,'XTickLabel',[0 90 180 270 360]);
text(30,0.8*10^-6.8,'Spec')
text(60,0.8*10^-6.8,version)
legend('Upper','Lower',0)

plot_compare_upperpist_friction_4_7_10.m

% M-file to plot both upper piston friction for three engine specs versus crank angle after exhaust port
% close
plot((1:360)+14,upperpistfrictionvar4,(1:360)+1,upperpistfrictionvar7,1:360,upperpistfrictionvar10)
title ('Upper Piston Friction Vs Crank Angle after E.C.for Various Specs')
xlabel ('Crank Angle after E.C. (Deg)')
ylabel ('Friction (N)(temporal)')
minvar=[min(upperpistfrictionvar4),min(upperpistfrictionvar7),min(upperpistfrictionvar10)];
maxvar=[max(upperpistfrictionvar4),max(upperpistfrictionvar7),max(upperpistfrictionvar10)];
axis([0 360 min(minvar-20) max(maxvar)+20]);
set(gca,'XTick',[0 90 180 270 360]);
set(gca,'XTickLabel',[0 90 180 270 360]);
legend('Spec3','Spec7','Spec10',0)

plot_eff_disp_CCR10_0_M9.m

% eff_disp=[40.3,25.47.6,60;48.5,79,34.4,18]
% Eff=[35.1,38.6,40.3,43.0,46.7,46.1,48.5,47.1];
% ExpVol=[17.9,22.3,24.7,37.0,46.4,59.5,68.6,78.8,83.6];

% M-file to plot the efficiency Vs max Relative Piston Displacement for constant E.V.D.R.
eff_disp_data = [38.4, 35.6, 4.2, 6.3, 0.3, 3.0, 8.4, 6.8, 8.1; ...
39.6, 42.2, 7.9, 2.7, 4.0, 8.5, 47.8, 9.8; ...
39.6, 48.8, 7.1, 2.8, 4.5, 9.9, 18.4; ...
39.5, 55.5, 6.8, 2.8, 5.0, 52.8, 49.4, 15.5; ...
39.8, 62.4, 6.6, 2.8, 6.0, 58.9, 49.3, 16.8; ...
40.0, 65.6, 6.4, 2.9, 6.4, 44.9, 18.4; ...
39.9, 69.7, 6.4, 2.8, 6.5, 66.8, 49.7, 19.2; ...
39.9, 74.0, 6.2, 2.9, 7.0, 72.1, 49.9, 21.0; ...
39.9, 78.0, 6.2, 3.0, 8.0, 78.3, 50.5, 23.5; ...
39.6, 84.1, 6.2, 3.0, 8.5, 84.4, 50.8, 26.6; ...
39.7, 87.6, 6.1, 3.0, 9.0, 89.7, 51.0, 28.0; ...
39.5, 90.0, 6.0, 3.1, 9.2, 94.2, 51.3, 30.3; ...
39.8, 93.9, 6.0, 3.1, 9.5, 97.8, 51.3, 30.7; ...
39.2, 100.9, 5.9, 3.1, 10.0, 106.7, 51.6, 35.5; ...
38.3, 109.6, 5.9, 3.1, 11.0, 116.6, 52.0, 42.6];

plot(eff_disp_data(:,2),eff_disp_data(:,1),’o’,eff Disp_data(:,2),eff DISP_data(:,7),’s’)
title(’Efficiency Vs Inducted Volume for Various Eng Specs’)

legend(’Nett Eff’,’Thermal Eff’)
%original constant max press data
% eff_disp_data= [34.9,28.2,7.9,2.4,2.0,23.0;...
%                 41.3,42.3,9.2,2.6,4.0,39.0;...
%                 42.5,53.7,9.7,2.8,5.0,57.6;...
%                 43.5,69.3,10.1,6.5,69.0;...
%                 43.7,74.0,10.1,9.7,72.7;...
%                 44.1,78.3,10.2,8.0,89.7;...
%                 44.5,83.1,10.6,2.9,7.0,72.7;...
%                 44.9,86.5,10.7,3.0,8.0,92.7;...
%                 44.8,91.0,10.8,3.2,11.0,96.3;...
%                 44.0,103.5,10.9,3.0,11.0,105.7;...
%                 42.9,108.3,10.9,3.2,11.0,116.4];

figure
plot(eff_disp_data(:,2),eff_disp_data(:,8),'d')
title ('Friction Work Vs Inducted Volume for Various Eng Specs')
xlabel ('Inducted Volume (cc)')
ylabel ('Work (J)')

coeffs2=polyfit(eff_disp_data(:,2),eff_disp_data(:,8),2);
hold on
polyval2data=polyval(coeffs2,xpoints);
plot(xpoints,polyval2data,'Color',[0,0.5,0.5]);
axis([min(eff_disp_data(:,2))-15 max(eff_disp_data(:,2))+10 ...
     min(eff_disp_data(:,8))-5 max(eff_disp_data(:,8)+5)]);

for counter=1:m;
    if counter==1
        speclabel=sprintf('Spec%4.1f',eff_disp_data(counter,5));
        text(eff Disp_data(counter,2),eff Disp_data(counter,8)+0.5,speclabel,'Color',[0,0.5,0.5],'Rotation',[90])
    else
        configlabel=sprintf('%4.1f',eff Disp_data(counter,5));
        text(eff Disp_data(counter,2),eff Disp_data(counter,8)+0.5,configlabel,'Color',[0,0.5,0.5],'Rotation',[90])
    end
end
% M-file to plot the efficiency Vs max Relative Piston Displacement for constant E.V.D.R.

eff_disp_data= [37.4,36.5,8.7,2.5,3.0,23.0,46.5,7.9;...
38.9,41.9,9.9,1.2,8.4,4.0,39.1,46.9,9.5;...
39.2,49.1,9.5,2.7,4.5,45.6,47.9,12.1;...
39.3,55.3,9.7,2.8,5.0,53.1,49.1,11.5;...
39.7,62.8,9.2,8.6,5.0,58.7,49.2,16.7;...
40.0,65.5,10.0,2.9,6.2,6.4,50.0,18.4;...
40.0,69.9,10.0,2.8,6.5,66.7,49.7,19.3;...
40.1,73.1,10.1,2.9,7.0,72.5,50.3,21.1;...
40.1,77.7,10.3,3.0,8.0,78.2,50.8,23.6;...
39.9,83.8,10.4,3.0,8.5,84.4,51.1,26.6;...
40.0,87.6,10.6,3.0,9.0,89.3,51.4,28.2;...
39.8,90.2,10.6,3.1,9.2,93.9,51.8,30.5;...
40.1,93.7,10.6,3.1,9.5,97.5,51.8,30.9;...
39.5,101.3,10.7,3.0,10.0,105.9,52.0,35.8;...
38.7,109.6,10.8,3.1,11.0,115.8,52.5,42.7];
%eff,ind vol,CR,EVDR,config,max stroke,thermeff,friction work

plot(eff_disp_data(:,2),eff_disp_data(:,1),'o',eff_disp_data(:,2),eff_disp_data(:,7),'s')
title ('Efficiency Vs Inducted Volume for Various Engine Specs')
legend('Nett Eff','Thermal Eff')

text (22,max(eff_disp_data(:,7))+4.0,'Bore')
text (40,max(eff_disp_data(:,7))+4.0,'= 61.3mm')
text (22,max(eff_disp_data(:,7))+3.0,'E.O.Press')
text (40,max(effDisp_data(:,7))+3.0,'= 1 atm')
text (22,max(effDisp_data(:,7))+2.0,'Const Maximum')
text (22,max(effDisp_data(:,7))+1.0,'Pressure')
text (40,max(effDisp_data(:,7))+1.0,'= 6.5 MPa')
text (22,max(effDisp_data(:,7))+0.0,'Const Burn Start')
text (40,max(effDisp_data(:,7))-1.0,'= -27 deg')
text (50,35,'A.C.R. = Actual Compression Ratio','Color',[0,0,1])
text (50,34,'(Max Inducted Vol / Min Compressed Vol)','Color',[0,0,1])

xlabel ('Inducted Volume (cc)')
ylabel ('Efficiency (%)')

axis([min(effDisp_data(:,2))-15 max(effDisp_data(:,2))+10 min(effDisp_data(:,1))-5 max(effDisp_data(:,1)+5))];
coeffs=polyfit(effDisp_data(:,2),effDisp_data(:,1),4);
epoints=effDisp_data(1,2):0.1:effDisp_data(end,2);
hold on
polyvaldata=polyval(coeffs,xpoints);
plot(xpoints,polyvaldata,'g');
coeffs2 = polyfit(eff_disp_data(:,2), eff_disp_data(:,7), 2);
hold on
polyval2data = polyval(coeffs2, xpoints);
plot(xpoints, polyval2data, 'Color', [0, 0.5, 0.5]);

% splinedata = spline(eff_disp_data(:,2), eff_disp_data(:,1), xpoints);
% plot (xpoints, splinedata,'r')

[m, n] = size(eff_disp_data);

for counter = 1:m;
    if counter == 1
        speclabel = sprintf('Spec%4.1f', eff_disp_data(counter, 5));
        strokelabel = sprintf('A.C.R. %4.1f', eff_disp_data(counter, 3));
        text(eff_disp_data(counter, 2), eff_disp_data(counter, 1) + 0.5, speclabel, 'Color', [0, 0.5, 0.5], 'Rotation', [90])
        text(eff_disp_data(counter, 2) + 0.2, eff_disp_data(counter, 1) - 0.5, strokelabel, 'Color', [0, 0, 1], 'Rotation', [-60])
    else
        configlabel = sprintf('%.1f', eff_disp_data(counter, 5));
        strokelabel = sprintf('%4.1f', eff_disp_data(counter, 3));
        text(eff_disp_data(counter, 2), eff_disp_data(counter, 1) + 0.5, configlabel, 'Color', [0, 0.5, 0.5], 'Rotation', [90])
        text(eff_disp_data(counter, 2) + 0.2, eff_disp_data(counter, 1) - 0.5, strokelabel, 'Color', [0, 0, 1], 'Rotation', [-60])
    end
end

%original constant max press data
% eff_disp_data = [34.9, 28.2, 7.9, 2.4, 2.0, 23.0;...
%                  41.3, 42.3, 9.2, 2.6, 4.0, 39.0;...
%                  42.5, 53.7, 9.7, 2.8, 5.0, 50.4;...
%                  43.3, 58.9, 9.8, 2.9, 6.0, 57.6;...
%                  43.4, 64.5, 10.1, 2.9, 6.2, 62.5;...
%                  43.5, 69.3, 10.1, 3.0, 6.5, 69.0;...
%                  43.7, 74.0, 10.1, 2.9, 7.0, 72.7;...
%                  44.1, 78.3, 10.2, 3.0, 8.0, 79.7;...
%                  44.5, 83.1, 10.6, 2.9, 8.5, 82.7;...
%                  44.9, 86.5, 10.7, 3.0, 9.0, 88.9;...
%                  44.8, 91.0, 10.8, 3.2, 10.0, 105.7;...
%                  44.8, 94.3, 10.7, 3.0, 11.0, 116.4;...
figure
plot(eff_disp_data(:,2), eff_disp_data(:,8), 'd')
title ('Friction Work Vs Inducted Volume for Various Eng Specs')
xlabel ('Inducted Volume (cc)')
ylabel ('Work (J)')

coeffs2 = polyfit(eff_disp_data(:,2), eff_disp_data(:,8), 2);
hold on
polyval2data = polyval(coeffs2, xpoints);
plot(xpoints, polyval2data, 'Color', [0, 0.5, 0.5]);
axis([min(eff_disp_data(:,2)) - 15 max(eff_disp_data(:,2)) + 10 ...
     min(eff_disp_data(:,8)) - 5 max(eff_disp_data(:,8)) + 5]);

text(22, max(eff_disp_data(:,8)) + 2.0, 'Bore')
text(40, max(eff_disp_data(:,8)) + 2.0, '= 61.3mm')
text(22, max(eff_disp_data(:,8)) + 0.5, 'E.O.Press')
text(40, max(eff_disp_data(:,8)) + 0.5, '= 1 atm')
text(22, max(eff_disp_data(:,8)) - 1, 'Const Compression')
text(22, max(eff_disp_data(:,8)) + 2.5, 'Ratio')
text(40, max(eff_disp_data(:,8)) + 2.5, '= 10.0')
text(22, max(eff_disp_data(:,8)) - 4, 'Const Burn Start')
for counter=1:m;
    if counter==1
        speclabel=sprintf('Spec%4.1f',eff_disp_data(counter,5));
        text(eff_disp_data(counter,2),eff_disp_data(counter,8)+0.5,speclabel,'Color',[0,.5,0.5],'Rotation',[90])
    else
        configlabel=sprintf('%4.1f',eff_disp_data(counter,5));
        text(eff_disp_data(counter,2),eff_disp_data(counter,8)+0.5,configlabel,'Color',[0,0.5,0.5],'Rotation',[90])
    end
end

plot_engine_pressure.m

% M-file to plot the engine pressure versus crank angle after exhaust port
% close

plot(1:360,engpressure/1e6)

title ('Engine Pressure Vs Crank Angle after E.C.')
xlabel ('Crank Angle after E.C. (Deg)')
ylabel ('Engine Pressure (MPa)')
axis([0 360 0 ceil(max(engpressure)/1e6)]);
% set(gca,'XTick',[0 90 180 270 360]);
% set(gca,'XTickLabel',[0 90 180 270 360]);

plot_lower_conrod_tension.m

% M-file to plot the lower conrod tension versus crank angle after exhaust port
% close

plot(1:360,conrodtensionlower/1000)

title ('Lower ConRod Tension Vs Crank Angle after E.C.')
xlabel ('Crank Angle after E.C. (Deg)')
ylabel ('Tension (kN)')
axis([0 360 floor(min(conrodtensionlower/1000)) ceil(max(conrodtensionlower/1000))]);
set(gca,'XTick',[0 90 180 270 360]);
set(gca,'XTickLabel',[0 90 180 270 360]);

plot_lower_normal_piston_load.m

% M-file to plot the normal load versus crank angle after exhaust port
% close

plot(1:360,normallower/1e3)

title ('Lower Normal Piston Load Vs Crank Angle after E.C.')
xlabel ('Crank Angle after E.C. (Deg)')
ylabel ('Normal Force (KPa)')
axis([0 360 floor(min(normallower/1e3)) ceil(max(normallower/1e3))]);
set(gca,'XTick',[0 90 180 270 360]);
set(gca,'XTickLabel',[0 90 180 270 360]);
plot_lower_pist_friction.m

% M-file to plot lower piston friction versus crank angle after exhaust port
% close

hold on

title('Piston Friction Vs Crank Angle after E.C.')
xlabel('Crank Angle after E.C. (Deg)')
ylabel('Friction (N)(temporal)')

minvar=[min(lowerpistfrictionvar)];
maxvar=[max(lowerpistfrictionvar)];
axis([0 360 minvar-10 maxvar+130]);
set(gca,'XTick',[0 90 180 270 360]);
set(gca,'XTickLabel',[0 90 180 270 360]);

text(5,(max(maxvar))+100,'Spec')
text(30,(max(maxvar))+100,version)

index=find(lowerconrodangle(1:330)>pi/2);
index=[max(index),min(index)];

line([index(1) index(1)], [minvar maxvar],'Color',[.4,.6,.8],'Linewidth',3)
line([index(2) index(2)], [minvar maxvar],'Color',[.4,.6,.8],'Linewidth',3)

line([88 88], [(max(maxvar))+120 (max(maxvar))+80],'Color',[.4,.6,.8],'Linewidth',3)
text(95,(max(maxvar))+110,'ConRod parallel')
text(95,(max(maxvar))+90,'to cylinder axis')

line([200 200], [(max(maxvar))+120 (max(maxvar))+80],'Color',[0.7,0.7,0],'Linewidth',3)
text(207,(max(maxvar))+110,'Boundary')
text(207,(max(maxvar))+90,'Friction')

line([270 270], [(max(maxvar))+120 (max(maxvar))+80],'Color',[1,0.7,0.7],'Linewidth',3)
text(277,(max(maxvar))+110,'Mixed')
text(277,(max(maxvar))+90,'Friction')

for m=1:2:360
    if StNolower(m)<10^-6.8 &  StNolower(m)==index(1) & StNolower(m)==index(2)
        line([m m], [minvar maxvar],'Color',[0.7,0.7,0],'Linewidth',5)
    elseif StNolower(m)==10^-6.8 & StNolower(m)<10^-5.8...
        & (max(maxvar))+120==index(1) & (max(maxvar))+120==index(2)
        line([m m], [minvar maxvar],'Color',[1,0.7,0.7],'Linewidth',5)
    end
end

plot(1:360,lowerpistfrictionvar)

plot_lower_piston_acc.m

% M-file to plot the lower piston acceleration versus crank angle after exhaust port
% close

plot(1:360,d2Slowerdt2)
title('Lower Piston Acceleration Vs Crank Angle after E.C.')
```matlab
xlabel ('Crank Angle after E.C. (Deg)')
ylabel ('Acceleration (m/s)')
axis([0 360 floor(min(d2Slowerdt2))-250 ceil(max(d2Slowerdt2))+250]);
set(gca,'XTick',[0 90 180 270 360]);
set(gca,'XTickLabel',[0 90 180 270 360]);

plot_lower_piston_vel.m

% M-file to plot the lower piston velocity versus crank angle after exhaust port
% close

plot(1:360,dSlowerdt)
title ('Lower Piston Velocity Vs Crank Angle after E.C.' )

xlabel ('Crank Angle after E.C. (Deg)')
ylabel ('Velocity (m/s)')
axis([0 360 floor(min(dSlowerdt))-2 ceil(max(dSlowerdt))+2]);
set(gca,'XTick',[0 90 180 270 360]);
set(gca,'XTickLabel',[0 90 180 270 360]);

plot_rel_angle_lowerconrod_crank.m

% M-file to plot the relative upper conrod angle to crank angle versus crank angle after exhaust port
% close
plotangle=[1:360];
plot(plotangle,-(lowerconrodangle/pi*180-plotangle-plotindex-90))
title ('Relative Angle Between lower ConRod and Crank Arm Vs Crank Angle after E.C.')

xlabel ('Crank Angle after E.C. (Deg)')
ylabel ('Relative Angle (deg)')
axis([0 360 180 630]);
set(gca,'XTick',[0 90 180 270 360]);
set(gca,'XTickLabel',[0 90 180 270 360]);
set(gca,'YTick',[180 270 360 450 540]);
set(gca,'YTickLabel',[180 270 360 450 540]);
grid on

plot_rel_angle_upperconrod_crank.m

% M-file to plot the relative upper conrod angle to crank angle versus crank angle after exhaust port
% close
plotangle=[1:360];
plot(plotangle,-(upperconrodangle/pi*180-plotangle-plotindex-90))
title ('Relative Angle Between Upper ConRod and Crank Arm Vs Crank Angle after E.C.')

xlabel ('Crank Angle after E.C. (Deg)')
ylabel ('Relative Angle (deg)')
axis([0 360 0 450]);
set(gca,'XTick',[0 90 180 270 360]);
set(gca,'XTickLabel',[0 90 180 270 360]);
set(gca,'YTick',[0 90 180 270 360]);
set(gca,'YTickLabel',[0 90 180 270 360]);
```
grid on

**plot_rel_piston_position.m**

% M-file to plot the relative position of pistons Vs crank angle after exhaust port
% close

plot(1:360,upper_pist_positionmm-lower_pist_positionmm)
title ('Relative Position of pistons Vs Crank Angle after E.C.')
xlabel ('Crank Angle after E.C. (Deg)')
ylabel ('Vertical Displacement (mm)')
axis([0 360 0 ceil(max(upper_pist_positionmm-lower_pist_positionmm)+3))]
set(gca,'XTick',[0 90 180 270 360])
maxdisplacementlabel=sprintf('Max Stroke = %4.1f',max(upper_pist_positionmm-lower_pist_positionmm))
text(200, 5, maxdisplacementlabel)

**plot_upper_conrod_tension.m**

% M-file to plot the upper conrod tension versus crank angle after exhaust port
% close

plot(1:360,conrodtensionupper/1000)
title ('Upper ConRod Tension Vs Crank Angle after E.C.')
xlabel ('Crank Angle after E.C. (Deg)')
ylabel ('Tension (kN)')
axis([0 360 floor(min(conrodtensionupper/1000)) ceil(max(conrodtensionupper/1000))]
set(gca,'XTick',[0 90 180 270 360])

**plot_upper_normal_piston_load.m**

% M-file to plot the upper normal piston load versus crank angle after exhaust port
% close

plot(1:360,normalupper/1e3)
title ('Upper Normal Piston Load Vs Crank Angle after E.C.')
xlabel ('Crank Angle after E.C. (Deg)')
ylabel ('Normal Force (KPa)')
axis([0 360 floor(min(normalupper/1e3)) ceil(max(normalupper/1e3))]
set(gca,'XTick',[0 90 180 270 360])

**plot_upper_piston_acc.m**

% M-file to plot the upper piston acceleration versus crank angle after exhaust port
% close

plot(1:360,d2Supperdt2)
title ('Upper Piston Acceleration Vs Crank Angle after E.C.')
xlabel ('Crank Angle after E.C. (Deg)')
plot_upper_piston_vel.m

% M-file to plot the upper piston velocity versus crank angle after exhaust port
% close

plot(1:360,dSupperdt)

title ('upper Piston Velocity Vs Crank Angle after E.C.'

xlabel ('Crank Angle after E.C. (Deg)')
ylabel ('Velocity (m/s)')

plot_work.m

% M-file to plot the accumulated work versus crank angle after exhaust port
% close

nn=4;

work=[pTuWQlHl(:,nn-1);pTbTuWQlHl((2:end),nn);pTbWQlHl((2:end),nn-1)];

thetaworkcycle=[thetacomp;thetacomb(2:end);thetaexp(2:end)];

thetaworkcycleindex=[maxinvolindex:maxexpandvolindex];

work=interp1(thetaworkcycle,work,thetaworkcycleindex/180*pi);

plot(thetaworkcycleindex,work)

plot_work_cycle_pressure.m

% Script file to plot output from ahrind.m and compare results
% with output listed in:
% load ahrind.mat; % results from ahrind.mat
% load ferguson.txt; % tabulated output from ferguon p178-179

plot(thetacomp*180/pi,pTuWQlHl(1,:)/1e6); hold on;
plot(thetacomp2*180/pi,p2TuWQlHl(1,:)/1e6);
plot(thetacomb*180/pi,pTbTuWQlHl(1,:)/1e6);
plot(thetaexp*180/pi,pTbWQlHl(1,:)/1e6);

axis([thetacomp(1)*180/pi thetaexp(end)*180/pi 0 9]);
plotresults.m

% plotresults.m
%

% Script file to plot output from ahrind.m and compare results
% with output listed in:
load ahrind.mat; % results from ahrind.mat
load ferguson.txt; % tabulated output from ferguson p178-179
% Set some parameters to make figures look attractive
NLW=1; % normal line width
NFS=18; % normal font size
NMS=1; % normal marker size
close all;
figure(1);
subplot(1,2,1)
plot(thetacomp*180/pi,pTuWQlHl(:,1)/1e6); hold on;
plot(thetacomp2*180/pi,p2TuWQlHl(:,1)/1e6);
plot(thetacomp3*180/pi,p3TuWQlHl(:,1)/1e6);
plot(thetacomb*180/pi,pTbTuWQlHl(:,1)/1e6);
plot(thetaexp*180/pi,pTbWQlHl(:,1)/1e6);
axis([thetacomp(1)*180/pi thetaexp(end)*180/pi 0 9]);
set(gca,'FontSize',NFS)
set(gca,'LineWidth',NLW)
set(gca,'XTick',[-180 -90 0 90 180 ]);
set(gca,'XTickLabel',[-180 -90 0 90 180 ]);
xlabel('crank angle (degrees ATC)');
ylabel('pressure (MPa)');
print -deps p_ahr.eps
figure(2);
subplot(1,2,1)
plot(ferguson(:,1),ferguson(:,4)*1e5/1e6,'o'); hold on;
plot(ferguson(:,1),ferguson(:,5),'o'); hold on;
set(gca,'FontSize',NFS)
set(gca,'LineWidth',NLW)
set(gca,'XTick',[-180 -90 0 90 180 ]);
set(gca,'XTickLabel',[-180 -90 0 90 180 ]);
xlabel('crank angle (degrees ATC)');
ylabel('temperature (K)');
legend('burned gas','unburned gas',2);
subplot(1,2,2)
plot(ferguson(:,1),ferguson(:,6),'s');
axis([-180 180 0 9]);
figure(3);
subplot(1,2,1)
%text (200,170,'Cycle Work')
nn=4; % for work plot

m=4; % for work plot
plot(thetacomp*180/pi,pTuWQlHl(:,nn-1)); hold on;
plot(thetacomp2*180/pi,p2TuWQlHl(:,nn-1));
plot(thetacomp3*180/pi,p3TuWQlHl(:,nn-1));
plot(thetacomb*180/pi,pTbTuWQlHl(:,nn));
plot(thetaexp*180/pi,pTbWQlHl(:,nn));
axis([thetacomp(1)*180/pi thetaexp(end)*180/pi min(pTbWQlHl(:,nn)) max(pTbTuWQlHl(:,nn-1))]);
xlabel('Crank Angle (compression and expansion)')
title ('Opposed Piston Engine Cycle Cumulative Work')

subplot(1,2,2)
plot(ferguson(:,1),ferguson(:,7),'o');
axis([-180 180 -300 600]);

% set(gca,'FontSize',NFS)
% set(gca,'LineWidth',NLW)
set(gca,'XTick',[-180 -90 0 90 180]);
set(gca,'XTickLabel',[-180 -90 0 90 180]);
set(gca,'YTick',[-300 -150 0 150 300 450 600]);
set(gca,'YTickLabel',[-300 -150 0 150 300 450 600]);
xlabel('crank angle (degrees ATC)')
ylabel('work (J)')
title ('Reference Engine Cycle Cumulative Work')
print -deps W_ahr.eps

figure(4);
subplot(1,2,1)
nn=5; % for heat transfer plot
plot(thetacomp*180/pi,pTuWQlHl(:,nn-1)); hold on;
plot(thetacomp2*180/pi,p2TuWQlHl(:,nn-1));
plot(thetacomp3*180/pi,p3TuWQlHl(:,nn-1));
plot(thetacomb*180/pi,pTbTuWQlHl(:,nn));
plot(thetaexp*180/pi,pTbWQlHl(:,nn));
axis([thetacomp(1)*180/pi thetaexp(end)*180/pi -5 max(pTbWQlHl(:,nn-1))]);

subplot(1,2,2)
plot(ferguson(:,1),ferguson(:,8),'o');
axis([-180 180 -50 300]);

% set(gca,'FontSize',NFS)
% set(gca,'LineWidth',NLW)
set(gca,'XTick',[-180 -90 0 90 180]);
set(gca,'XTickLabel',[-180 -90 0 90 180]);
xlabel('crank angle (degrees ATC)')
ylabel('heat transfer (J)')
print -deps Ql_ahr.eps

figure(5);
subplot(1,2,1)
nn=6; % for heat leakage plot
plot(thetacomp*180/pi,pTuWQlHl(:,nn-1)); hold on;
plot(thetacomp2*180/pi,p2TuWQlHl(:,nn-1));
plot(thetacomp3*180/pi,p3TuWQlHl(:,nn-1));
plot(thetacomb*180/pi,pTbTuWQlHl(:,nn));
plot(thetaexp*180/pi,pTbWQlHl(:,nn));
axis([thetacomp(1)*180/pi thetaexp(end)*180/pi -5 max(pTbWQlHl(:,nn-1))]);

subplot(1,2,2)
plot(ferguson(:,1),ferguson(:,10),'o');
axis([-180 180 -8 1]);

% set(gca,'FontSize',NFS)
% set(gca,'LineWidth',NLW)
set(gca,'XTick',[-180 -90 0 90 180]);
set(gca,'XTickLabel',[-180 -90 0 90 180]);
xlabel('crank angle (degrees ATC)')
ylabel('heat leakage (J)')
print -deps Hl_ahr.eps
function yprime=RatesComb(theta,y,flag);

% % yprime=RatesComb(theta,y,flag)
% % Function that returns the derivatives of the following 6 variables
% % w.r.t. crank angle (theta) for the combustion phase:
% % 1) pressure; 2) unburned temperature; 3) burned temperature;
% % 4) work; 5) heat transfer; and 6) heat leakage.
% % See Ferguson, C.R., 1986, "Internal Combustion Engines", Wiley,
% % p174.
% global b stroke eps r Cblowby f fueltype airscheme phi ...
% thetas thetab omega ...
% heattransferlaw heu hcb ...
% Tw theta1 Vtdc Vbdc mass1 ...
% p1 T1 V1 minV maxinV maxexV ...
% compressedvolindex plotindex ...
% cylindervolume crankoffset conrod ...
% crankthrow uppercrankoffset upperconrod ...
% uppercrankthrow cranktocrank exportheight ...
% uppercranklag thetacorrected maxinvolindex ...
% maxexpandvolindex thetaignition upstroke downstroke...
% splugvolume

p=y(1);
% if p==0
% p=1e-6;
% elseif p<0
% p=1e-6;
% end
Tb=y(2);
Tu=y(3);
yprime=zeros(6,1);

% mass in cylinder accounting for blowby:
mass=mass1*(exp(-Cblowby*(theta-theta1)/omega))^2;

% volume of cylinder:
V=((cranktocrank + (sin(theta+plotindex*pi/180+pi/2+uppercranklag*pi/180)*uppercrankthrow-....
sqrt(upperconrod^2-(cos(theta+plotindex*pi/180+pi/2+uppercranklag*pi/180)*uppercrankthrow+....
uppercrankoffset)^2))+(sin(theta+plotindex*pi/180+pi/2)*crankthrow+sqrt(conrod^2-....
(com(theta+plotindex*pi/180+pi/2)*crankthrow+crankoffset)^2)))/pi*(b/2)^2+2*plugvolume;

% derivate of volume:
dVdtheta=((((cranktocrank + (sin(theta+plotindex*pi/180+pi/2+0.0001+uppercranklag*pi/180)*....
uppercrankthrow-sqrt(upperconrod^2-(cos(theta+plotindex*pi/180+pi/2+0.0001+uppercranklag*pi/180)*....
crankthrow+sqrt(conrod^2-(cos(theta+plotindex*pi/180+pi/2+0.0001)*crankthrow+crankoffset)^2)))/....
pi*(b/2)^2)-((cranktocrank + (sin(theta+plotindex*pi/180+pi/2-0.0001+uppercranklag*pi/180)*....
crankthrow-sqrt(upperconrod^2-(cos(theta+plotindex*pi/180+pi/2-0.0001+uppercranklag*pi/180)*....
crankthrow+uppercrankoffset)^2))-(sin(theta+plotindex*pi/180+pi/2-0.0001)*crankthrow+crankoffset)^2)))/....
pi*(b/2)^2)/0.0002;

% mass fraction burned and derivative:
x=0.5*(1-cos(pi*(theta-thetaignition)/thetab));
dxdtheta=pi/2/thetab*sin(pi*(theta-thetaignition)/thetab);
if x<0.0001, x=0.0001; end;
if x>0.9999, x=0.9999; end;

switch heattransferlaw
  case 'constant'
    hcoeffu=hcu;
    hcoeffb=hcb;
  case 'Woschni'
    upmean=omega*upstroke/((compressedvolindex-maxinvolindex)/180*pi); % mean piston velocity
    downmean=omega*downstroke/((maxexpandvolindex-compressedvolindex)/180*pi);
    meanps=(upmean+downmean)/2;
    C1=2.28;
    C2=3.24e-3;
    Vs=maxinV-minV;
    k=1.3;
    pm=p1*(V1/V)^k; % motoring pressure
    hcoeffu=hcu*130*b^(-0.2)*Tu^(-0.53)*(p/100e3)^(0.8)* ....
             (C1*meansp+C2*Vs*T1/p1/V1*(p-m))^0.8
    hcoeffb=hcb*130*b^(-0.2)*Tb^(-0.53)*(p/100e3)^0.8 ...
             (C1*meansp+C2*Vs*T1/p1/V1*(p-m))^0.8;
  end

A=1/mass*(dVdthetadt+2*V*Cblowby/omega);
Qconvu=hcoeffu*(pi*b^2/2+4*V/b)*(1-sqrt(x))*(Tu-Tw);
Qconvb=hcoeffb*(pi*b^2/2+4*V/b)*sqrt(x)*(Tu-Tw);
Const1=1/omega/mass;
[hu,uu,vu,s,Y,cpu,dlvlTu,dlvlpu]= ...
  farg(p,Tu,phi,fueltype,airscheme);
[hb,ub,vb,s,Y,cpb,dlvlTb,dlvlpb]= ...
  fcp(p,Tb,phi,fueltype,airscheme);
B=Const1*(vb/cpb*dlvlTb*Qconvb/Tb+ ...
    vu/cpu*dlvlTu*Qconvu/Tu);
C=-(vb-vu)*dxdtheta-vb*dvlTb/(hu-hb)/cpb/Tb*(dxdthetadt- ....
(x-x^2)*Cblowby/omega);
D=x*(vb^2/cpb*Tu*dvlTu^2+vb*p*dvlpu);
E=-(1-x)*(vu*2/cpu*Tu*dvlTu^2+vu/p*dlvpu);
yprime(1)=(A+B+C)/(D+E);
yprime(2)=Const1/cpb*x*Qconvb+vb/cpb*dvlTu*yprime(1) + ...
      (hu-hb)/cpb*(dxdthetadt-x-1-x)*Cblowby/omega);
yprime(3)=Const1/cpu*(1-x)*Qconvu+vu/cpu*dvlTu*yprime(1);
yprime(4)=p*dvlthetadt;
yprime(5)=Const1/mass*(Qconvb+Qconvu);
yprime(6)=Cblowby*mass/omega*((1-x^2)*hu+x^2*hb);
RatesComp.m

function yprime=RatesComp(theta,y,flag);
%
% yprime=RatesComp(theta,y,flag)
% 
% Function that returns the derivatives of the following 5 variables
% w.r.t. crank angle (theta) for the compression phase:
% 1) pressure; 2) unburned temperature;
% 3) work; 4) heat transfer; and 5) heat leakage.
% global b stroke eps r Cblowby f fueltype airscheme phi ...
% thetas thetab omega ...
% heattransferlaw hcu hcb ...
% Tw thetal Vtdc Vbdc mass1 ...
% p1 T1 V1 minV maxinV maxexV ...
% compressedvolindex plotindex ...
% cylindervolume crankoffset conrod ...
% crankthrow uppercrankoffset upperconrod ...
% uppercranklag thetacorrected maxinvolindex ...
% maxexpandvolindex thetaignition upstroke downstroke...
% upper_pist_positionmm, lower_pist_positionmm,splugvolume

p=y(1);
Tu=y(2);
yparme=zeros(5,1);
% mass in cylinder accounting for blowby:
mass=mass1*(exp(-Cblowby*(theta-theta1)/omega))^2;
% volume of cylinder:
% For upper piston positions higher than max(lower piston position), add
% 2*splugvolume

V=(cranktocrank + (sin(theta+plotindex*pi/180+pi/2+uppercranklag*pi/180) + uppercrankthrow...)
  sqrt(upperconrod^2-(cos(theta+plotindex*pi/180+pi/2+uppercranklag*pi/180)+ uppercrankthrow...
  + uppercrankoffset).^2))-(sin(theta+plotindex*pi/180+pi/2)* crankthrow + sqrt(conrod^2-...)
  (cos(theta+plotindex*pi/180+pi/2)* crankthrow + crankoffset).^2)))/pi*(b/2)^2; % derivate of volume:

switch heattransferlaw
  case 'constant'
    hcoeff=hcu;
  case 'Woschni'
    upmean=omega*upstroke/((compressedvolindex-maxinvolindex)/180*pi); % mean piston velocity
    C1=2.28;
  hcoeff=hcu*130*b^(-0.2)*Tu^(-0.53)*(p/100e3)^(0.8)*C1*upmean;
  end

A=1/mass*(dVdtheta=(cranktocrank + (sin(theta+plotindex*pi/180+pi/2+0.0001+uppercranklag*pi/180)...)
  uppercrankthrow-sqrt(upperconrod^2-(cos(theta+plotindex*pi/180+pi/2+0.0001+uppercranklag*pi/180)+
  pi/180+uppercrankthrow+uppercrankoffset).^2))-(sin(theta+plotindex*pi/180+pi/2)+0.0001)*...)
  crankthrow+sqrt(conrod^2-(cos(theta+plotindex*pi/180+pi/2+0.0001)* crankthrow + crankoffset).^2))...)
  pi*(b/2)^2-(cranktocrank + (sin(theta+plotindex*pi/180+pi/2-0.0001+uppercranklag*pi/180)...)
  uppercrankthrow-sqrt(upperconrod^2-(cos(theta+plotindex*pi/180+pi/2-0.0001+uppercranklag*pi/180)+...)
  uppercrankthrow+uppercrankoffset).^2)))-(sin(theta+plotindex*pi/180+pi/2-0.0001)*...)
  crankthrow+crankoffset).^2)))/pi*(b/2)^2)/0.0002;

switch heattransferlaw
  case 'constant'
    hcoeff=hcu;
  case 'Woschni'
    upmean=omega*upstroke/((compressedvolindex-maxinvolindex)/180*pi); % mean piston velocity
    C1=2.28;
  hcoeff=hcu*130*b^(-0.2)*Tu^(-0.53)*(p/100e3)^(0.8)*C1*upmean;
  end

A=1/mass*(dVdtheta+(compressedvolindex-maxinvolindex)/180*pi); % mean piston velocity
    C1=2.28;
  hcoeff=hcu*130*b^(-0.2)*Tu^(-0.53)*(p/100e3)^(0.8)*C1*upmean;
  end

Qconv=hcoeff*(pi*b^2/2+4*V^2/Tu(Tu-Tw);
Const1=1/omega/mass;
[h,u,v,s,Y,cp,dlvlT,dlvlp]=farg(p,Tu,phi,f,fueltype,airscheme);
B=Const1*cp*dlvT*Qconv/Tu; % note typo on p174, eq. 4.76
 RatesComp1.m

function yprime=RatesComp1(theta,y,flag);

yprime=RatesComp(theta,y,flag);

Function that returns the dr derivatives of the following 5 variables w.r.t. crank angle (theta) for the compression phase:

1) pressure; 2) unburned temperature;
3) work; 4) heat transfer; and 5) heat leakage.
See Ferguson, C.R., 1986, "Internal Combustion Engines", Wiley,
p174.

p1 T1 V1 minV maxinV maxexV
compressedvoldindex plotindex

6) Heat transfer law:
Woschni

V=((crankto crank + (sin(theta+plotindex*pi/180+pi/2)*uppercranklag*pi/180)*uppercrank throw...-
sqrt(uppercrankrod^2-(cos(theta+plotindex*pi/180+pi/2)*uppercranklag*pi/180)*uppercrank throw...+
uppercrankoffset,)^2))-(sin(theta+plotindex*pi/180+pi/2)*crankthrow+sqrt(conrod^2-...-
(cos(theta+plotindex*pi/180+pi/2)*crankthrow+crankoffset,)^2))/pi*(b/2)^2;

% derivatie of volume:
dVdtheta=((((crankto crank + (sin(theta+plotindex*pi/180+pi/2+0.0001+uppercranklag*pi/180)*...-
uppercrankthrow-sqrt(uppercrankrod^2-(cos(theta+plotindex*pi/180+pi/2+0.0001+uppercranklag*pi/180)*...-
pi/180)*uppercrankthrow+uppercrankoffset,)^2))-(sin(theta+plotindex*pi/180+pi/2+0.0001)*...-
crankthrow+sqrt(conrod^2-(cos(theta+plotindex*pi/180+pi/2+0.0001)*crank throw+crankoff set,)^2))...-
pi*(b/2)^2)-((((crankto crank + (sin(theta+plotindex*pi/180+pi/2-0.0001+uppercranklag*pi/180)*...-
uppercrankthrow-sqrt(uppercrankrod^2-(cos(theta+plotindex*pi/180+pi/2-0.0001+uppercranklag*pi/180)*...-
pi/180)*uppercrankthrow+uppercrank offset,)^2))-(sin(theta+plotindex*pi/180+pi/2-0.0001)*...-
crankthrow+sqrt(conrod^2-(cos(theta+plotindex*pi/180+pi/2-0.0001)*crankthrow+crankoffset,)^2)))...-
pi*(b/2)^2))/0.0002;

switch heattransferlaw
  case 'constant'
    hcoeff=hcu;
  case 'Woschni'
upmean=omega*upstroke/((compressedvolindex-maxinvolindex)/180*pi); % mean piston velocity
C1=2.28;
hcoeff=hcu*130*b^(-0.2)*Tu^(-0.53)*(p/100e3)^(0.8)*C1*upmean;
end
A=1/mass*(dVdtheta+2*V*Cblowby/omega);
Qconv=hcoeff*(pi*b^2/2+4*V/b)*(Tu-Tw);
Const1=1/omega/mass;
\[ h,u,v,s,Y,cp,dlvlT,dlvlp \] = farg(p,Tu,phi,fRC1,fueltype,airscheme);
B=Const1*v/cp*dvlT*Qconv/\( \text{Tu} \); % note typo on p174, eq. 4.76
C=0;
D=0;
E=v^2/cp/\( \text{Tu} \)*dlvlT*2+v/p*dlvlp;
yprime(1)=(A+B+C)/(D+E);
yprime(2)=Const1/cp*Qconv+v/cp*dvlT*yprime(1);
yprime(3)=p*dVdtheta;
yprime(4)=Qconv/omega;
yprime(5)=Cblowby*mass/omega*h;

RatesComp2.m

function yprime=RatesComp2(theta,y,flag);
%\% yprime=RatesComp(theta,y,flag)
%\%
% Function that returns the derivatives of the following 5 variables
% w.r.t. crank angle (theta) for the compression phase:
% 1) pressure; 2) unburned temperature;
% 3) work; 4) heat transfer; and 5) heat leakage.
% See Ferguson, C.R., 1986, "Internal Combustion Engines", Wiley,
% p174.
global b stroke eps r Cblowby f fueltype airscheme phi ...
thesetas thetab omega ...
heattransferlaw hcu hcb ...
Tw thetab Vtdc Vbdc mass1 ...
p1 T1 V1 minV maxinV maxexV ...
compressedvolindex plotindex ...
cylindervolume crankoffset conrod ...
crankthrow uppercrankoffset upperconrod ...
uppercrankthrow cranktocrank exportheight ...
uppercranklag thetacorrected maxinvolindex ...
maxexpandvolindex thetaignition upstroke downstroke...
fRC1 upper_pist_positionmm lower_pist_positionmm splugvolume...
thetaaddsplug dtheta

p=y(1);
Tu=y(2);
yprime=zeros(5,1);
% mass in cylinder accounting for blowby:
mass=mass1*(exp(-Cblowby*(theta-theta1)/omega))\^2;
% volume of cylinder:
% For upper piston positions higher than max(lower piston position), add
% 2*splugvolume
V=((cranktocrank + (sin(theta+plotindex*pi/180+2+uppercranklag)*pi/180)*upuppercrankthrow-...
sqrt((upperconrod\^2-(cos(theta+plotindex*pi/180+2+uppercranklag)*pi/180)*uppercrankthrow...
+uppercrankoffset).\^2))-(sin(theta+plotindex*pi/180+2+uppercranklag)*sin(theta+plotindex*pi/180+2+uppercranklag)*uppercrankthrow-...
sqrt((upperconrod\^2-(cos(theta+plotindex*pi/180+2+uppercranklag)*sin(theta+plotindex*pi/180+2+uppercranklag)*uppercrankthrow-...
sqrt((upperconrod\^2-(cos(theta+plotindex*pi/180+2+uppercranklag)*sin(theta+plotindex*pi/180+2+uppercranklag)*uppercrankthrow-...
sqrt((upperconrod\^2-(cos(theta+plotindex*pi/180+2+uppercranklag)*sin(theta+plotindex*pi/180+2+uppercranklag)*uppercrankthrow-...
sqrt((upperconrod\^2-(cos(theta+plotindex*pi/180+2+uppercranklag)*sin(theta+plotindex*pi/180+2+uppercranklag)*uppercrankthrow-...
sqrt((upperconrod\^2-(cos(theta+plotindex*pi/180+2+uppercranklag)*sin(theta+plotindex*pi/180+2+uppercranklag)*uppercrankthrow-...

% derivate of volume:
dVdtheta=((\((cranktocrank + (sin(theta+plotindex*pi/180+2+uppercranklag)*pi/180)*upuppercrankthrow-...
sqrt((upperconrod\^2-(cos(theta+plotindex*pi/180+2+uppercranklag)*pi/180)*uppercrankthrow...
+uppercrankoffset).\^2))-(sin(theta+plotindex*pi/180+2+uppercranklag)*sin(theta+plotindex*pi/180+2+uppercranklag)*uppercrankthrow-...
sqrt((upperconrod\^2-(cos(theta+plotindex*pi/180+2+uppercranklag)*sin(theta+plotindex*pi/180+2+uppercranklag)*uppercrankthrow-...
sqrt((upperconrod\^2-(cos(theta+plotindex*pi/180+2+uppercranklag)*sin(theta+plotindex*pi/180+2+uppercranklag)*uppercrankthrow-...
sqrt((upperconrod\^2-(cos(theta+plotindex*pi/180+2+uppercranklag)*sin(theta+plotindex*pi/180+2+uppercranklag)*uppercrankthrow-...
sqrt((upperconrod\^2-(cos(theta+plotindex*pi/180+2+uppercranklag)*sin(theta+plotindex*pi/180+2+uppercranklag)*uppercrankthrow-...
sqrt((upperconrod\^2-(cos(theta+plotindex*pi/180+2+uppercranklag)*sin(theta+plotindex*pi/180+2+uppercranklag)*uppercrankthrow-...

160
\[
\text{crankthrow} + \sqrt{\text{conrod}^2 - (\cos(\theta + \text{plotindex} \cdot \pi/180 + \pi/2 + 0.0001) \cdot \text{crankthrow} + \text{crankoffset})^2}) \] 
\[
\times \pi \cdot (\text{b}/2)^2 - \text{upperconrod} - (\pi \cdot \text{crankthrow} + \text{crankoffset}) \} \times (\sin(\theta + \text{plotindex} \cdot \pi/180 + \pi/2 - 0.0001) \cdot \text{crankthrow} + \text{crankoffset})^2) \) 
\[
\times \pi \cdot (\text{b}/2)^2) / 0.0002 + 2 \cdot \text{splugvolume/dtheta;}
\]

switch heattransferlaw
\{ 
  case 'constant'
    hcoeff = hcu;
  case 'Woschni'
    upmean = omega * upstroke / ((compressedvolindex - maxinvolindex) / 180 * pi); % mean piston velocity
    C1 = 2.28;
    hcoeff = hcu * 130 * b^(-0.2) * Tu^(-0.53) * (p/100e3)^(0.8) * C1 * upmean;
  end
\}

A = 1/mass * (dV/dtheta + 2 * V * Cblowby/omega);
Qconv = hcoeff * ((pi * b^2 / 2 + 4 * V / b) * (Tu - Tw));
Const1 = 1/omega / mass;
[h, u, v, s, cp, dlvIT, dlvlp] = farg(p, Tu, phi, fRC1, fueltype, ashescheme);
B = Const1 * v / cp * dlvIT * Qconv / Tu; % note typo on p174, eq. 4.76
C = 0;
D = 0;
E = v^2 / cp * Tu * dlvIT^2 * v / p * dlvlp;
ypreme(1) = (A + B + C) / (D + E);
ypreme(2) = -Const1 / cp * Qconvv / v / cp * dlvIT * ypreme(1);
ypreme(3) = p * dV/dtheta;
ypreme(4) = Qconv / omega;
ypreme(5) = Cblowby * mass / omega * h;

RatesComp3.m

function yprime = RatesComp3(theta, y, flag);

global b stroke eps r Cblowby f fueltype airscheme phi ... 
  thetas thetab omega ...
  heattransferlaw hcu hcb ...
  Tw theta1 Vtdc Vbdc mass1 ...
  p1 T1 V1 minV maxinV maxexV ...
  compressedvolindex plotindex ...
  cylindervolume crankoffset conrod ...
  crankthrow uppercrankoffset upperconrod ...
  uppercrankthrow cranktocrank exportheight ...
  uppercranklag thetacorrected maxinvolindex ...
  maxexpandvolindex thetaignition upstroke downstroke ...
  fRC1 upper_pist_positionmm lower_pist_positionmm splugvolume ...
  thetaaddsplug dtheta

p = y(1);
Tu = y(2);
yprime = zeros(5, 1);

mass = mass1 * (exp(-Cblowby * (theta - theta1) / omega)) * 2;

161
% For upper piston positions higher than max(lower piston position), add
% 2*splugvolume

\[
V = ((\text{cranktocrank} + (\sin(\theta + \text{plotindex} \cdot \pi/180 + \pi/2 + \text{uppercranklag} \cdot \pi/180) \cdot \text{uppercrankthrow} - \sqrt{\text{upperconrod}^2 - (\cos(\theta + \text{plotindex} \cdot \pi/180 + \pi/2 + \text{uppercranklag} \cdot \pi/180) \cdot \text{uppercrankthrow} + \text{uppercrankoffset})^2}) - (\sin(\theta + \text{plotindex} \cdot \pi/180 + \pi/2) \cdot \text{crankthrow} + \sqrt{\text{conrod}^2 - (\cos(\theta + \text{plotindex} \cdot \pi/180 + \pi/2) \cdot \text{crankthrow} + \text{crankoffset})^2})) \cdot \pi \cdot (b/2)^2 + 2 \cdot \text{splugvolume}
\]

% derivate of volume:

\[
dVd\theta = (((\text{cranktocrank} + (\sin(\theta + \text{plotindex} \cdot \pi/180 + 0.0001 + \text{uppercranklag} \cdot \pi/180) \cdot \text{uppercrankthrow} - \sqrt{\text{upperconrod}^2 - (\cos(\theta + \text{plotindex} \cdot \pi/180 + 0.0001 + \text{uppercranklag} \cdot \pi/180) \cdot \text{uppercrankthrow} + \text{uppercrankoffset})^2})) - (\sin(\theta + \text{plotindex} \cdot \pi/180 + 0.0001) \cdot \text{crankthrow} + \sqrt{\text{conrod}^2 - (\cos(\theta + \text{plotindex} \cdot \pi/180 + 0.0001) \cdot \text{crankthrow} + \text{crankoffset})^2})) \cdot \pi \cdot (b/2)^2) - ((\text{cranktocrank} + (\sin(\theta + \text{plotindex} \cdot \pi/180 + 0.0001 - \text{uppercranklag} \cdot \pi/180) \cdot \text{uppercrankthrow} - \sqrt{\text{upperconrod}^2 - (\cos(\theta + \text{plotindex} \cdot \pi/180 + 0.0001 - \text{uppercranklag} \cdot \pi/180) \cdot \text{uppercrankthrow} + \text{uppercrankoffset})^2})) - (\sin(\theta + \text{plotindex} \cdot \pi/180 + 0.0001 - \text{crankthrow} + \sqrt{\text{conrod}^2 - (\cos(\theta + \text{plotindex} \cdot \pi/180 + 0.0001 - \text{crankthrow} + \text{crankoffset})^2})) \cdot \pi \cdot (b/2)^2))/0.0002;
\]

end

switch heattransferlaw
  case 'constant'
    hcoeff = hcu;
  case 'Woschni'
    upmean = omega * upstroke / ((compressedvolindex - maxinvolindex) / 180 * pi); % mean piston velocity
    C1 = 2.28;
    hcoeff = hcu * 130 * b^(-0.2) * Tu^(-0.53) * (p / 100e3)^(0.8) * C1 * upmean;
  end

A = 1 / mass * (dVdtheta + 2 * V * Cblowby / omega);
Qconv = hcoeff * (pi * b^2 / 2 + 4 * V / b) * (Tu - Tw);
Const1 = 1 / omega / mass;
B = Const1 * v / cp * dvlT * Qconv / Tu; % note typo on p174, eq. 4.76
C = 0;
D = 0;
E = v^2 / cp * Tu * dvlT^2 / v * p * dvlvlp;
yprime(1) = (A + B + C) / (D + E);
yprime(2) = Const1 * v / cp * Qconv * v / cp * dvlT * yprime(1);
yprime(3) = p * dVdtheta;
yprime(4) = Qconv / omega;
yprime(5) = Cblowby * mass / omega * h;

RatesCompO1.m

function yprime = RatesCompO1(theta, y, flag);
    
% yprime = RatesComp(theta, y, flag)

% Function that returns the derivatives of the following 5 variables
% w.r.t. crank angle (theta) for the compression phase:

% 1) pressure; 2) unburned temperature;
% 3) work; 4) heat transfer; and 5) heat leakage.
% See Ferguson, C.R., 1986, "Internal Combustion Engines", Wiley,
% p174.

global b stroke eps r Cblowby f fueltype airscheme phi ... thetas thetab omega ...
heattransferlaw hcu hcb ...
Tw theta1 Vtdc Vbdc massl ...
p1 T1 V1 minV maxinV maxexV ...
compressedvolindex plotindex ...
cylindervolume crankoffset conrod ...
crankthrow uppercrankoffset upperconrod ...
uppercrankthrow cranktocrank exportheight ... 
uppercranklag thetacorrected maxinvolindex ... 
maxexpandvolindex thetaignition upstroke downstroke...
fRC1

\[ p = y(1); \]
\[ T_u = y(2); \]
\[ yprime = zeros(5, 1); \]
\% mass in cylinder accounting for blowby:
\[ mass = mass_1 \times (exp(-C_{\text{blowby}} \times (theta_{-}theta_1)/omega))^2; \]
\% volume of cylinder:
\% For upper piston positions higher than max(lower piston position), add
\% 2*splugvolume

\[ V = ((\text{cranktocrank} + \sin(theta_{+}plotindex \times \pi/180 + pi/2 + uppercranklag \times \pi/180) \times uppercrankthrow... \]
\[ \sqrt{(upperconrod^2 - (cos(theta_{+}plotindex \times \pi/180 + pi/2 + uppercranklag \times \pi/180) \times uppercrankthrow... \]
\[ + uppercrankoffset)^2))) - (\sin(theta_{+}plotindex \times \pi/180 + pi/2) \times crankthrow + \sqrt{conrod^2 -... \]
\[ (cos(theta_{+}plotindex \times \pi/180 + pi/2) \times crankthrow + crankoffset)^2))) \times pi \times (b/2)^2; \]
\% derivate of volume:
\[ dV_dtheta = (((\text{cranktocrank} + \sin(theta_{+}plotindex \times \pi/180 + pi/2 + 0.0001 + uppercranklag \times \pi/180) \times... \]
\[ uppercrankthrow - \sqrt{(upperconrod^2 - (cos(theta_{+}plotindex \times \pi/180 + pi/2 + 0.0001 + uppercranklag \times... \]
\[ pi/180) \times uppercrankthrow + uppercrankoffset)^2))) - (\sin(theta_{+}plotindex \times \pi/180 + pi/2 + 0.0001)^2, ... \]
\[ crankthrow + \sqrt{conrod^2 - (cos(theta_{+}plotindex \times \pi/180 + pi/2 + 0.0001)^2, crankthrow + crankoffset)^2))) \times pi \times (b/2)^2; \]
\]
\% derivate of volume:
\[ dV_dtheta = (((\text{cranktocrank} + \sin(theta_{+}plotindex \times \pi/180 + pi/2 + 0.0001 + uppercranklag \times \pi/180) \times... \]
\[ uppercrankthrow - \sqrt{(upperconrod^2 - (cos(theta_{+}plotindex \times \pi/180 + pi/2 + 0.0001 + uppercranklag \times... \]
\[ pi/180) \times uppercrankthrow + uppercrankoffset)^2))) - (\sin(theta_{+}plotindex \times \pi/180 + pi/2 + 0.0001)^2, ... \]
\[ crankthrow + \sqrt{conrod^2 - (cos(theta_{+}plotindex \times \pi/180 + pi/2 + 0.0001)^2, crankthrow + crankoffset)^2))) \times pi \times (b/2)^2; \]
\]
\% switch heattransferlaw
\% case 'constant'
hcoeff = hcu;
\% case 'Woschni'
upmean = omega \times upstroke \times ((\text{compressedvolindex - maxinvolindex}) \times 180 \times pi); \% mean piston velocity
C1 = 2.28;
\% hcoeff = hcu * (130 * b^4 * C_{\text{blowby}} / omega);
\]
\% Qconv = hcoeff \times (pi \times b^2 / 2 + 4 \times V / b) \times (T_u - T_w);
\% Const1 = 1/omega/mass;
\% [h, u, v, s, y, cp, dlvIT, dlvlp] = farg(p, T_u, phi, fRC1, fueltype, airscheme);
\% B = Const1 \times v \times cp \times dlvIT \times Qconv / T_u; \% note typo on p174, eq. 4.76
\% C = 0;
\% D = 0;
\% E = v^2 \times cp / T_u \times dlvIT \times 2 + v \times p \times dlvlp;
\% yprime(1) = (A + B + C) / (D + E);
\% yprime(2) = -Const1 \times cp \times Qconv + v \times cp \times dlvIT \times yprime(1);
\% yprime(3) = p \times dvthetata;
\% yprime(4) = Qconv / omega;
\% yprime(5) = Cblowby \times mass / omega * h;

RatesCompO2.m

function yprime = RatesCompO2(theta, y, flag);
\%
\% yprime = RatesComp(theta, y, flag);
\%
\% Function that returns the drivatives of the following 5 variables
\% w.r.t. crank angle (theta) for the compression phase:
\%
\% 1) pressure; 2) unburned temperature;
\% 3) work; 4) heat transfer; and 5) heat leakage.
global b stroke eps r Cblowby f fueltype airscheme phi ...
theatas thetab omega ...
heattransferlaw hcu hcb ...
Tw thetal Vte Vbdc mass1 ...
p1 T1 V1 minV maxinV maxexV ...
compressedvolindex plotindex ...
cylindervolume crankoffset conrod ...
crankthrow uppercrankoffset upperconrod ...
uppercrankthrow cranktocrank exphovrheight ...
uppercranklag thetacorrected maxinvolindex ...
maxexpandvolindex thetaignition upstroke downstroke...
upper_pist_positionmm lower_pist_positionmm slugvolume...
thetaaddsplug dtheta

p=y(1);
Tu=y(2);
yprime=zeros(5,1);
% mass in cylinder accounting for blowby:
mass=mass1*(exp(-Cblowby*(theta-theta1)/omega))^2;
% volume of cylinder:
% For upper piston positions higher than max(lower piston position), add
% 2*slugvolume
V=((cranktocrank + (sin(theta+plotindex*pi/180+pi/2+uppercranklag*pi/180)*uppercrankthrow-...
sqrt(upperconrod^2-(cos(theta+plotindex*pi/180+pi/2+uppercranklag*pi/180)*uppercrankthrow...
+uppercrankoffset)^2))-(sin(theta+plotindex*pi/180+pi/2)*crankthrow+sqrt(upperrod^2-...
(cos(theta+plotindex*pi/180+pi/2)*crankthrow+uppercrankoffset)^2)))*pi*(b/2)^2+2*slugvolume;
% derivate of volume:
if theta==thetaaddsplug
  dVdtheta=((((cranktocrank + (sin(theta+plotindex*pi/180+pi/2+0.0001+uppercranklag*pi/180)*uppercrankthrow-...
sqrt(upperconrod^2-(cos(theta+plotindex*pi/180+pi/2+uppercranklag*pi/180)*uppercrankthrow+uppercrankoffset)^2))-(sin(theta+plotindex*pi/180+pi/2-0.0001)*crankthrow+sqrt(upperrod^2-(cos(theta+plotindex*pi/180+pi/2+uppercranklag*pi/180)*uppercrankthrow+uppercrankoffset)^2)))*pi*(b/2)^2+2*slugvolume)/0.0002;
else
  dVdtheta=((((cranktocrank + (sin(theta+plotindex*pi/180+pi/2+0.0001+uppercranklag*pi/180)*uppercrankthrow-...
sqrt(upperconrod^2-(cos(theta+plotindex*pi/180+pi/2+uppercranklag*pi/180)*uppercrankthrow+uppercrankoffset)^2))-(sin(theta+plotindex*pi/180+pi/2-0.0001)*crankthrow+sqrt(upperrod^2-(cos(theta+plotindex*pi/180+pi/2+uppercranklag*pi/180)*uppercrankthrow+uppercrankoffset)^2)))*pi*(b/2)^2+2*slugvolume)/0.0002;
end
switch heattransferlaw
  case 'constant'
    hcoeff=hcu;
  case 'Woschni'
    upmean=omega*upstroke/((compressedvolindex-maxinvolindex)/180*pi); % mean piston velocity
    C1=2.28;
    hcoeff=hcu*130*b^(-0.2)*Tu^(-0.53)*(p/100e3)^(0.8)*C1*upmean;
  end
A=1/mass*(dVdtheta+2*V*Cblowby/omega);
Qconv=hcoeff*(pi*b^2/2)*Tu*(Tu-Tw);
Const1=1/omega/mass;
[h,v,s,Y,cp,dlvlT,dlvlp]=farg(p,Tu,phi,fueltype,airscheme);
\[ B = \text{Const}1 \times \frac{v}{\text{cp}} \times d\text{vlT} \times Q_{\text{conv}} / \text{Tu}; \text{ } \% \text{ note typo on p174, eq. 4.76} \]
\[ C = 0; \]
\[ D = 0; \]
\[ E = \frac{v^2}{\text{cp}} \times \frac{d\text{vlT}^2}{\text{v}} / \text{p} \times d\text{vlp}; \]
\[ \text{yprime}(1) = \frac{(A + B + C)}{(D + E)}; \]
\[ \text{yprime}(2) = -\text{Const}1 \times \frac{Q_{\text{conv}}}{\text{cp}} \times v \times d\text{vlT} \times \text{yprime}(1); \]
\[ \text{yprime}(3) = \text{p} \times d\text{Vdtheta}; \]
\[ \text{yprime}(4) = Q_{\text{conv}} / \omega; \]
\[ \text{yprime}(5) = C_{\text{blowby}} \times \frac{\text{mass}}{\omega} \times h; \]

RatesComprans.m

function yprime = RatesComprans(theta, y, flag);
%
% yprime = RatesComp(theta, y, flag)
% % Function that returns the drivatives of the following 5 variables
% w.r.t. crank angle (theta) for the compression phase:
% % 1) pressure; 2) unburned temperature;
% % 3) work; 4) heat transfer; and 5) heat leakage.
% % See Ferguson, C.R., 1986, "Internal Combustion Engines", Wiley,
% % p174.
% global b stroke eps r Cblowby f fueltype airscheme phi ...
% thetas thetab omega ... heattransferlaw hcu hcb ...
% Tw theta1 Vtdc Vbdc mass1 ... p1 T1 V1 minV maxinV maxexV ...
% compressedvolindex plotindex cylindervolume crankoffset conrod ...
% crankthrow uppercrankoffset upperconrod ...
% uppercrankthrow cranktocrank exportheight ...
% uppercranklag thetacorrected maxinvolindex ...
% maxexpandvolindex thetaignition upstroke downstroke...
% RBC1 upper_pist_positionmm lower_pist_positionmm splugvolume...
% thetaaddsplug dtheta

p = y(1);
Tu = y(2);
% mass in cylinder accounting for blowby:
mass = mass1 * (exp(-Cblowby * (theta - theta1) / omega))^2;
% volume of cylinder:
% For upper piston positions higher than max(lower piston position), add
% % 2*splugvolume

V = ((crankto crank + (sin(theta + plotindex * pi/180 + pi/2) + uppercranklag * pi/180) + uppercrankthrow - ...
\sqrt{(upperconrod * pi^2) - (cos(theta + plotindex * pi/180 + pi/2) + uppercranklag * pi/180) + uppercrankthrow + uppercrankoffset * pi^2)}) - (sin(theta + plotindex * pi/180 + pi/2) + crankthrow + sqrt(conrod * pi^2 - ...
(cos(theta + plotindex * pi/180 + pi/2) + crankthrow + crankoffset * pi^2)) * pi) * (b/2) * 2 + 2 * splugvolume;
% derivate of volume:

dVdthetanu = ((crankto crank + (sin(theta + plotindex * pi/180 + pi/2 + 0.0001) + uppercranklag * pi/180) * ...
uppercrankthrow + sqrt(upperconrod * 2 - (cos(theta + plotindex * pi/180 + pi/2 + 0.0001) + uppercranklag * pi/180) + ...
pi/180) + uppercrankthrow + uppercrankoffset * pi^2)) - (sin(theta + plotindex * pi/180 + pi/2 + 0.0001) + ...
crankthrow + sqrt(conrod * 2 - (cos(theta + plotindex * pi/180 + pi/2 + 0.0001) + crankthrow + crankoffset * pi^2)) * ...
pi) * (b/2) * 2 + ((crankto crank + (sin(theta + plotindex * pi/180 + pi/2 - 0.0001) + uppercranklag * pi/180) + ...
uppercrankthrow + sqrt(upperconrod * 2 - (cos(theta + plotindex * pi/180 + pi/2 - 0.0001) + uppercranklag * pi/180) + ...
uppercrankthrow + uppercrankoffset * pi^2)) - (sin(theta + plotindex * pi/180 + pi/2 = 0.0001) + crankthrow + ...
+ 2 * splugvolume/dtheta;
switch heattransferlaw
    case 'constant'
        hcoeff = hcu;
    case 'Woschni'
        upmean = omega*upstroke/((compressedvolindex-maxinvolindex)/180*pi); % mean piston velocity
        C1 = 2.28;
        hcoeff = hcu*130*b^(-0.2)*Tu^(-0.53)*(p/100e3)^(0.8)*C1*upmean;
    end
A = 1/mass*(dVdtheta+2*V*Cblowby/omega);
Qconv = hcoeff*(pi*b^2/2+4*V/b)*(Tu-Tw);
Const1 = 1/omega/mass;
end
\[ yprime(1) = \frac{A+B+C}{D+E}; \]
\[ yprime(2) = -\frac{C1/cp*Qconv+v/cp*dlvlT*yprime(1)}{v/cp*dlvlT}; \]
\[ yprime(3) = p*dVdtheta; \]
\[ yprime(4) = Qconv/omega; \]
\[ yprime(5) = Cblowby*mass/omega*h; \]

RatesExp.m

function yprime = RatesExp(theta,y,flag);

% Function that returns the derivatives of the following 5 variables
% w.r.t. crank angle (theta) for the expansion phase:
% 1) pressure; 2) unburned temperature;
% 3) work; 4) heat transfer; and 5) heat leakage.
% See Ferguson, C.R., 1986, "Internal Combustion Engines", Wiley,
% p174.

global b stroke eps r Cblowby f fueltype airscheme phi ... 
    thetas thetab omega ...
    heattransferlaw hcu hcb ...
    Tw theta1 Vtdc Vbdc mass1 ...
    p1 T1 V1 minV maxinV maxexV ...
    compressedvolindex plotindex ...
    cylindervolume crankoffset conrod ...
    crankthrow uppercrankoffset upperconrod ...
    uppercrankthrow cranktocrank exportheight ...
    uppercranklag thetacorrected maxinvolindex ...
    maxexpandvolindex thetaignition upstroke downstroke ...
    portopenindex splugvolume

p = y(1);
Tk = y(2);
yprime = zeros(5,1);
% mass in cylinder accounting for blowby:
mass = mass1*(exp(-Cblowby*(theta-theta1)/omega))^
% volume of cylinder:
V = ((cranktocrank + (sin(theta+plotindex*pi/180+pi/2+uppercranklag*pi/180)*uppercrankthrow- ... 
    sqrt(upperconrod^2*(cos(theta+plotindex*pi/180+pi/2+uppercranklag*pi/180)*uppercrankthrow+ ... 
    uppercrankoffset)^2))+(sin(theta+plotindex*pi/180+pi/2+uppercranklag*pi/180)*uppercrankthrow+ ... 
    sqrt(uppercrankoffset^2*(cos(theta+plotindex*pi/180+pi/2+uppercranklag*pi/180)*uppercrankthrow+ ... 
    uppercrankoffset)^2)-(cos(theta+plotindex*pi/180+pi/2+uppercranklag*pi/180)*uppercrankthrow+ ... 
    crankoffset)^2))/pi*(h/2)^2+2*splugvolume;
% derivative of volume:
dVdtheta = ((cranktocrank + (sin(theta+plotindex*pi/180+pi/2+uppercranklag*pi/180)*uppercrankthrow- ... 
    sqrt(upperconrod^2*(cos(theta+plotindex*pi/180+pi/2+uppercranklag*pi/180)*uppercrankthrow+ ... 
    uppercrankoffset)^2))+(sin(theta+plotindex*pi/180+pi/2+uppercranklag*pi/180)*uppercrankthrow+ ... 
    sqrt(uppercrankoffset^2*(cos(theta+plotindex*pi/180+pi/2+uppercranklag*pi/180)*uppercrankthrow+ ... 
    uppercrankoffset)^2)-(cos(theta+plotindex*pi/180+pi/2+uppercranklag*pi/180)*uppercrankthrow+ ... 
    crankoffset)^2))/pi*(h/2)^2+2*splugvolume;

166
uppercrankthrow-sqrt(upperconrod^2-(cos(theta+plotindex*pi/180+pi/2-0.0001+uppercranklag*pi/180)*... uppercrankthrow+uppercrankoffset).^2))-(sin(theta+plotindex*pi/180+pi/2-0.0001)*... crankthrow+sqrt(conrod^2-(cos(theta+plotindex*pi/180+pi/2-0.0001)*crankthrow+crankoffset).^2)))*... pi*(b/2)^2)/0.0002;

switch heattransferlaw
  case 'constant'
    hcoeff=hcb;
  case 'Woschni'
    downmean=omega*downstroke/((maxexpandvolindex-compressedvolindex)/180*pi); % mean piston velocity
    C1=2.28;
    C2=3.24e-3;
    Vs=maxexV-minV;
    k=1.3;
    pm=p1*(V1/V)^k; % motoring pressure
    hcoeff=hcb*130*b^(-0.2)*Tb^(-0.53)*(p/100e3)^0.8*...
    (C1*downmean+C2*Vs*T1/p1/V1*(p-pm))^0.8;
  end

A=1/mass*(dVdtheta+2*V*Cblowby/omega);
Qconv=hcoeff*(pi*b^2/2+4*V/b)*(Tb-Tw);
Const1=1/omega/mass;
if Tb<1000
  [h,u,v,s,Y,cp,dlvlT,dlvlp]=farg(p,Tb,phi,1,fueltype,airscheme);
else
  [h,u,v,s,Y,cp,dlvlT,dlvlp]=ecp(p,Tb,phi,fueltype,airscheme);
end
B=Const1*v/cp*dvlV*T*Qconv/Tb;
C=0;
D=v^2/cp/Tb*dvlV^2+vp*p*dvlV;
E=0;
yprime(1)=(A+B+C)/(D+E);
yprime(2)=Const1/cp*Qconv+v/cp*dvlV*yprime(1);
yprime(3)=p*dVdtheta;
yprime(4)=Qconv/omega;
yprime(5)=Cblowby^2*omega*h;

ringfriction.m

function [ringfrictwork]=ringfriction(dSdt,d2Sdt2,_pist_positionmm,b,muoil,mu,engpressure);

ringtension=20;
ringwidth=0.003;

% determine Strubeck relation between # and coeff of friction (f) from plot
% use as input into Strubeck calculation
graphStNo=[-6 -3];
graphf=[-3 -1.3];
StNofit=polyfit(graphStNo,graphf,1);

%calc Strubeck number
StNo=muoil*abs(dSdt)./(ringtension/(b*pi*ringwidth)+engpressure);

%determine corresponding coeff of friction
f=10.^((polyval((StNofit),log10(StNo))));

% obtain er piston position increments with'diff'
_pist_position_diff=diff(_pist_positionmm/1000);
_pist_position_diff=[_pist_position_diff,_pist_position_diff(end)];

for counter=1:length(engpressure);
  if StNo(counter)<=0.01*1e-4
    ringfriction=mu*(ringtension+engpressure).*sign(dSdt);
  end
end
else
    ringfriction=f(counter)*(ringtension/(b*pi*ringwidth)+engpressure).*sign(dSdt);
end
end

ringfrictworkall=abs(ringfrictioncombined).*abs(_pist_position_diff);
nringfrictwork=sum(ringfrictworkall)

slider_bar_input.m

%slider_bar_input - the simulation main file that establishes the GUI, calls other
%functions and files and displays the outputs.
% type slider_bar_input in the matlab workspace
clc;
clear;
starttime=cputime;

global b stroke eps r Cblowby f fueltype airscheme phi ...
    thetas thetab omega ...
    heattransferlaw hcs hcb ...
    Tw thetal Vtdc Vbdc mass1 ...
    p1 T1 V1 minV maxinV maxexV ...
    compressedvolindex plotindex ...
    cylindervolume crankoffset conrod ...
    crankthrow uppercrankoffset upperconrod ...
    uppercrankthrow cranktocrank exportheight ...
    uppercranklag thetacorrected maxinvolindex ...
    maxexpandvolindex thetaignition upstroke downstroke RPM...
    upper_pist_positionmm lower_pist_positionmm splugvolume muoil mu fRC1

% Set default values for variable dimensions - replace 'enginespec....' on
% line 31 to alter default engine spec setting
[version,uppercrankoffset,upperconrod,uppercrankthrow,crankoffset,...
    conrod,crankthrow,cranktocrank,exportheight,uppercranklag,...
    uppercrankoffsetrange,upperconrodrange,uppercrankthrowrange,...
    crankoffsetrange,conrodrange,crankthrowrange,cranktocrankrange,...
    exportheightrange,uppercranklagrange,thetasdeg,thetabdeg,thetasdegrange,...
    thetabdegrange,splugport]=enginespec prototype; %XXXXXXXXXXXXXXXXXX

%user input for engine simulation speed
RPM = input('Engine Speed in Revs/min :  ');

%choose a simulation option using a menu
k = menu('Choose the Model Version','ORIGINAL','CORRECTED THERMO-CYLE','CORRECTED
    THERMO-CYCLE & FRICTION');

%establish the blow-by constant for the simulation option
Cblowby=[0.8, 8, 8];
Cblowby=Cblowby(k);

%convert engine specs to mm
 crankoffsetmm=crankoffset*1000;
 conrodimm=conrod*1000;
 crankthrowmm=crankthrow*1000;
 uppercrankoffsetmm=uppercrankoffset*1000;
 upperconrodmm=upperconrod*1000;
 uppercrankthrowmm=uppercrankthrow*1000;
 cranktocrankmm=cranktocrank*1000;
 exportheightmm=exportheight*1000;

% Set range for slider-bars
uppercrankoffsetrangelength=uppercrankoffsetrange(end)-uppercrankoffsetrange(1);
upperconrodrangelength=upperconrodrange(end)-upperconrodrange(1);
uppercrankthrowrangelength=uppercrankthrowrange(end)-uppercrankthrowrange(1);
crankoffsetrangelength=crankoffsetrange(end)-crankoffsetrange(1);
conrodrangelength=conrodrange(end)-conrodrange(1);
crankthrowrangelength=crankthrowrange(end)-crankthrowrange(1);
cranktocrankrangelength=cranktocrankrange(end)-cranktocrankrange(1);
exportheightrangelength=exportheightrange(end)-exportheightrange(1);
uppercranklagrangelength=uppercranklagrange(end)-uppercranklagrange(1);
thetasdegrangelength=thetasdegrange(end)-thetasdegrange(1);
thetabdegrangelength=thetabdegrange(end)-thetabdegrange(1);

% Get screen size and set figure size
scrsz = get(0,'ScreenSize');
figure('Position',[20 100 scrsz(3)-40 scrsz(4)-200])

% Plot each slider-bar in the LH figure - slider-bars are plots of red dots
% from x=15-45 and y-dimensions of 2, 4, 6,....., 18
subplot(1,2,1);
plot(crankoffsetrange*30/crankoffsetrangelength+15-crankoffsetrange(1)*30/crankoffsetrangelength,12,'r.'),
text(1,12,'Lower OffSet'),
label1low=sprintf('%3.1f',crankoffsetrange(1)),
label1high=sprintf('%3.1f',crankoffsetrange(end)),
hold on
plot(conrodrange*30/conrodrangelength+15-conrodrange(1)*30/conrodrangelength,10,'r.'),
text(1,10,'Lower ConRod'),
label2low=sprintf('%3.1f',conrodrange(1)),
label2high=sprintf('%3.1f',conrodrange(end)),
plot(crankthrowrange*30/crankthrowrangelength+15-crankthrowrange(1)*30/crankthrowrangelength,8,'r.'),
text(1,8,'Lower Throw'),
label3low=sprintf('%3.1f',crankthrowrange(1)),
label3high=sprintf('%3.1f',crankthrowrange(end)),
plot(uppercrankoffsetrange*30/uppercrankoffsetrangelength+15-uppercrankoffsetrange(1)*30/uppercrankoffsetrangelength,18,'r.'),
text(1,18,'Upper OffSet'),
label4low=sprintf('%3.1f',uppercrankoffsetrange(1)),
label4high=sprintf('%3.1f',uppercrankoffsetrange(end)),
plot(upperconrodrange*30/upperconrodrangelength+15-upperconrodrange(1)*30/upperconrodrangelength,16,'r.'),
text(1,16,'Upper ConRod'),
label5low=sprintf('%3.1f',upperconrodrange(1)),
label5high=sprintf('%3.1f',upperconrodrange(end)),
plot(uppercrankthrowrange*30/uppercrankthrowrangelength+15-uppercrankthrowrange(1)*30/uppercrankthrowrangelength,14,'r.'),
text(1,14,'Upper Throw'),
label6low=sprintf('%3.1f',uppercrankthrowrange(1)),
label6high=sprintf('%3.1f',uppercrankthrowrange(end)),
plot(cranktocrankrange*30/cranktocrankrangelength+15-cranktocrankrange(1)*30/cranktocrankrangelength,6,'r.'),

169
text(1.6,'Crank to Crank')
label7low=sprintf('%3.1f',cranktocrankrange(1));
text(15.5,5.5,label7low)
label7high=sprintf('%3.1f',cranktocrankrange(end));
text(45.5,5.5,label7high)

plot(exportheightrange*30/exportheightrangelength+15-exportheightrange(1)*30/exportheightrangelength,4,'r.'
text(1.4,'Ex-port Height')
label8low=sprintf('%3.1f',exportheightrange(1));
text(15.3,5.5,label8low)
label8high=sprintf('%3.1f',exportheightrange(end));
text(45.3,5.5,label8high)

plot(uppercranklagrange*30/uppercranklagrangelength+15-
uppercranklagrange(1)*30/uppercranklagrangelength,2,'r.'
text(1.2,'Crank Lag(deg)'
label9low=sprintf('%3.1f',uppercranklagrange(1));
text(15.1,5,label9low)
label9high=sprintf('%3.1f',uppercranklagrange(end));
text(45.1,5,label9high)

plot(thetasdegrange*30/thetasdegrangelength+15-thetasdegrange(1)*30/thetasdegrangelength,22,'r.'
text(1.22,'Burn Start(deg)'
label10low=sprintf('%3.1f',thetasdegrange(1));
text(15.2,1.5,label10low)
label10high=sprintf('%3.1f',thetasdegrange(end));
text(45.2,1.5,label10high)

plot(thetabdegrange*30/thetabdegrangelength+15-thetabdegrange(1)*30/thetabdegrangelength,20,'r.'
text(1.20,'Burn Duration')
label11low=sprintf('%3.1f',thetabdegrange(1));
text(15.195,label11low)
label11high=sprintf('%3.1f',thetabdegrange(end));
text(45.195,label11high)

text(12,24.5,'Spec')
text(12,23.5,version)

% Limit slider-bar figure axes
axis([0,45,1,25])

% Plot default values of variable dimensions as 'o'
plot(crankoffsetmm*30/crankoffsetrangelength+15-crankoffsetrange(1)*30/crankoffsetrangelength,12,'o',
conrodmm*30/conrodrangelength+15-conrodrange(1)*30/conrodrangelength,10,'o',
crankthrowmm*30/crankthrowrangelength+15-crankthrowrange(1)*30/crankthrowrangelength,8,'o',
uppercrankoffsetmm*30/uppercrankoffsetrangelength+15-
uppercrankoffsetrange(1)*30/uppercrankoffsetrangelength,18,'o',
upperconrodmm*30/upperconrodrangelength+15-upperconrodrange(1)*30/upperconrodrangelength,16,'o',
uppercrankthrowmm*30/uppercrankthrowrangelength+15-
uppercrankthrowrange(1)*30/uppercrankthrowrangelength,14,'o',
cranktockrangement*30/cranktockrangementrange+15-cranktockrangement(1)*30/cranktockrangementrange,6,'o',
exportheightmm*30/exportheightrangelength+15-exportheightrange(1)*30/exportheightrangelength,4,'o',
uppercranklag*30/uppercranklagrangelength+15-uppercranklagrange(1)*30/uppercranklagrangelength,2,'o',
thetastrange*30/thetastrangelength+15-thetasdegrange(1)*30/thetastrangelength,22,'o',
thetastrange*30/thetastrangelength+15-thetabdegrange(1)*30/thetastrangelength,20,'o')

% remove axis ticks on slider-bar figure
axis off

% Plot the default values of the variable dimensions on each slider-bar
subplot(1,2,1);
floatlabel1=sprintf('%3.1f',crankoffsetmm);
text(crankoffsetmm*30/crankoffsetrangelength+15-
crankoffsetrange(1)*30/crankoffsetrangelength,12.8,floatlabel1)
floatlabel2=sprintf('%3.1f',conrodmm);
text(conrodmm*30/conrodrangelength+15-conrodrange(1)*30/conrodrangelength,10.8,floatlabel2)
floatlabel3 = sprintf('%3.1f', crankthrowmm);
text(crankthrowmm*30/crankthrowrangelength+15-
crankthrowrange(1)*30/crankthrowrangelength,8.8,floatlabel3)
floatlabel4 = sprintf('%3.1f', uppercrankoffsetmm);
text(uppercrankoffsetmm*30/uppercrankoffsetrangelength+15-
uppercrankoffsetrange(1)*30/uppercrankoffsetrangelength,18.8,floatlabel4)
floatlabel5 = sprintf('%3.1f', upperconrodmm);
text(upperconrodmm*30/upperconrodrangetrange+15-
upperconrodrange(1)*30/upperconrodrangetrange,16.8,floatlabel5)
floatlabel6 = sprintf('%3.1f', uppercrankthrowmm);
text(uppercrankthrowmm*30/uppercrankthrowrangelength+15-
uppercrankthrowrange(1)*30/uppercrankthrowrangelength,14.8,floatlabel6)
floatlabel7 = sprintf('%3.1f', cranktocrankmm);
text(cranktocrankmm*30/cranktocrankrangelength+15-
cranktocrankrange(1)*30/cranktocrankrangelength,6.8,floatlabel7)
floatlabel8 = sprintf('%3.1f', exportheightmm);
text(exportheightmm*30/exportheightrangelength+15-
exportheightrange(1)*30/exportheightrangelength,4.8,floatlabel8)
floatlabel9 = sprintf('%3.1f', uppercranklag);
text(uppercranklag*30/uppercranklagrangelength+15-
uppercranklagrange(1)*30/uppercranklagrangelength,2.8,floatlabel9)
floatlabel10 = sprintf('%3.1f', thetasdeg);
text(thetasdeg*30/thetasdegrangelength+15-thetasdegrange(1)*30/thetasdegrangelength,22.8,floatlabel10)
floatlabel11 = sprintf('%3.1f', thetabdeg);
text(thetabdeg*30/thetabdegrangelength+15-thetabdegrange(1)*30/thetabdegrangelength,20.8,floatlabel11)

% set input markers '+' at default position at the LH end of each slider
% and replace the displayed position at each ginput bar
aa = 15;
bb = 12;
a = plot(aa, bb, '+');
% display text at end of bar for changed settings
atext = text(50, 12, sprintf('%3.1f', crankoffsetmm));
set(atext, 'EraseMode', 'xor', 'MarkerSize', 14)
set(atext, 'EraseMode', 'xor')
cc = 15;
dd = 10;
c = plot(cc, dd, '+');
c = text(50, 10, sprintf('%3.1f', conrodmm));
set(c, 'EraseMode', 'xor', 'MarkerSize', 14)
set(c, 'EraseMode', 'xor')
ee = 15;
ff = 8;
e = plot(ee, ff, '+');
e = text(50, 8, sprintf('%3.1f', crankthrowmm));
set(e,'EraseMode','xor','MarkerSize',14)
set(etext,'EraseMode','xor')

gg=15;
hh=18;
g=plot(gg,hh, '+');
gtext=text(50,18,sprintf('%3.1f',uppercrankoffsetmm));
set(g,'EraseMode','xor','MarkerSize',14)
set(gtext,'EraseMode','xor')

ii=15;
jj=16;
 i=plot(ii,jj, '+');
 itext=text(50,16,sprintf('%3.1f',upperconrodmm));
 set(i,'EraseMode','xor','MarkerSize',14)
 set(itext,'EraseMode','xor')

kk=15;
mm=14;
km=plot(kk,mm, '+');
ktext=text(50,14,sprintf('%3.1f',uppercrankthrowmm));
set(km,'EraseMode','xor','MarkerSize',14)
set(ktext,'EraseMode','xor')

nn=15;
 oo=6;
 n=plot(nn,oo, '+');
n text=txtex(50,6,sprintf('%3.1f',cranktocrankmm));
set(n,'EraseMode','xor','MarkerSize',14)
set(n text,'EraseMode','xor')

pp=15;
qq=4;
 p=plot(pp,qq, '+');
p text=txtex(50,4,sprintf('%3.1f',exportheightmm));
set(p,'EraseMode','xor','MarkerSize',14)
set(p text,'EraseMode','xor')

rr=15;
 ss=2;
 r=plot(rr,ss, '+');
r text=txtex(50,2,sprintf('%3.1f',uppercranklag));
set(r,'EraseMode','xor','MarkerSize',14)
set(r text,'EraseMode','xor')

tt=15;
 uu=20;
 t=plot(tt,uu, '+');
t text=txtex(50,20,sprintf('%3.1f',thetabdeg));
set(t,'EraseMode','xor','MarkerSize',14)
set(t text,'EraseMode','xor')

vv=15;
 ww=22;
 v=plot(vv,ww, '+');
v text=txtex(50,22,sprintf('%3.1f',thetasdeg));
set(v,'EraseMode','xor','MarkerSize',14)
set(v text,'EraseMode','xor')

% zero the ginput control variable, 'z'
z=0;

% set the movie frame counter to 1

% create conditional statement that allows the ginput to assess which
% variable is being input
while z==0
    subplot(1,2,1);
    [x,y]=ginput(1);
    if((y>11) & (y<13));
        crankoffsetmm=round(((x+crankoffsetrange(1)*30/crankoffsetrangelength)-
15)/30*crankoffsetrangelength)*10)/10;
        aa=crankoffsetmm*30/crankoffsetrangelength+15-crankoffsetrange(1)*30/crankoffsetrangelength;
        crankoffssetmm=crankoffsetmm*1000;
        set(a,'XData',aa,'YData',bb)
        set(atext,'String',sprintf('%3.1f',crankoffsetmm))
    elseif ((y>9) & (y<11));
        conrodmm=round(((x+conrodrange(1)*30/conrodrangelength)-15)/30*conrodrangelength)*10)/10;
        cc=conrodmm*30/conrodrangelength+15-conrodrange(1)*30/conrodrangelength;
        conrodd=conrodmm/1000;
        set(c,'XData',cc,'YData',dd)
        set(ctext,'String',sprintf('%3.1f',conrodmm))
    elseif ((y>7) & (y<9));
        crankthrowmm=round(((x+crankthrowrange(1)*30/crankthrowrangelength)-
15)/30*crankthrowrangelength)*10)/10;
        ee=crankthrowmm*30/crankthrowrangelength+15-crankthrowrange(1)*30/crankthrowrangelength;
        crankthrow=crankthrowmm/1000;
        set(e,'XData',ee,'YData',ff)
        set(etext,'String',sprintf('%3.1f',crankthrowmm))
    elseif ((y>17) & (y<19));
        uppercrankoffsetmm=round(((x+uppercrankoffsetrange(1)*30/uppercrankoffsetrangelength)-
15)/30*uppercrankoffsetrangelength)*10)/10;
        gg=uppercrankoffsetmm*30/uppercrankoffsetrangelength+15-
        uppercrankoffsetrange(1)*30/uppercrankoffsetrangelength;
        uppercrankoffsset=uppercrankoffsetmm/1000;
        set(g,'XData',gg,'YData',hh)
        set(gtext,'String',sprintf('%3.1f',uppercrankoffsetmm))
    elseif ((y>15) & (y<17));
        upperconrodmm=round(((x+upperconrodrange(1)*30/upperconrodrangelength)-
15)/30*upperconrodrangelength)*10)/10;
        ii=upperconrodmm*30/upperconrodrangelength+15-upperconrodrange(1)*30/upperconrodrangelength;
        upperconrod=upperconrodmm/1000;
        set(i,'XData',ii,'YData',jj)
        set(itext,'String',sprintf('%3.1f',upperconrodmm))
    elseif ((y>13) & (y<15));
        uppercrankthrowmm=round(((x+uppercrankthrowrange(1)*30/uppercrankthrowrangelength)-
15)/30*uppercrankthrowrangelength)*10)/10;
        kk=uppercrankthrowmm*30/uppercrankthrowrangelength+15-
        uppercrankthrowrange(1)*30/uppercrankthrowrangelength;
        uppercrankthrow=uppercrankthrowmm/1000;
        set(km,'XData',kk,'YData',mm)
        set(ktext,'String',sprintf('%3.1f',uppercrankthrowmm))
    elseif ((y>5) & (y<7));
        cranktocrankmm=round(((x+cranktocrankrange(1)*30/cranktocrankrangelength)-
15)/30*cranktocrankrangelength)*10)/10;
        nn=cranktocrankmm*30/cranktocrankrangelength+15-cranktocrankrange(1)*30/cranktocrankrangelength;
        cranktocrank=cranktocrankmm/1000;
        set(n,'XData',nn,'YData',oo)
        set(ntext,'String',sprintf('%3.1f',cranktocrankmm))
    elseif ((y>3) & (y<5));
        exportheightmm=round(((x+exportheightrange(1)*30/exportheightrangelength)-
15)/30*exportheightrangelength)*10)/10;
        pp=exportheightmm*30/exportheightrangelength+15-exportheightrange(1)*30/exportheightrangelength;
        exportheight=exportheightmm/1000;
set(p,'XData',pp,'YData',qq)
set(ptext,'String',sprintf('%3.1f',exportheightmm))

elseif ((y>1) & (y<3));
    uppercranklag=round(((x+uppercranklagrange(1)*30/uppercranklagrangelength)-15)/30*uppercranklagrangelength*10)/10;
    rr=uppercranklag*30/uppercranklagrangelength+15-uppercranklagrange(1)*30/uppercranklagrangelength;
    set(rs,'XData',rr,'YData',ss)
    set(rtext,'String',sprintf('%3.1f',uppercranklag))

elseif ((y>19) & (y<21));
    thetabdeg=round(((x+thetabdegrange(1)*30/thetabdegrangelength)-15)/30*thetabdegrangelength*2)/2;
    tt=thetabdeg*30/thetabdegrangelength+15-thetabdegrange(1)*30/thetabdegrangelength;
    set(tu,'XData',tt,'YData',uu)
    set(ttext,'String',sprintf('%3.1f',thetabdeg))

elseif ((y>21) & (y<23));
    thetasdeg=round(((x+thetasdegrange(1)*30/thetasdegrangelength)-15)/30*thetasdegrangelength*2)/2;
    vv=thetasdeg*30/thetasdegrangelength+15-thetasdegrange(1)*30/thetasdegrangelength;
    set(vw,'XData',vv,'YData',ww)
    set(vtext,'String',sprintf('%3.1f',thetasdeg))

elseif ((y>23) & (y<25) & (x>0) & (x<10))
    z=1;

elseif ((y>23) & (y<25) & (x>35) & (x<45))
    crankoffsetmm=crankoffsetmm;

else
    z=0;
end

% convert burn angle to radians
thetas=thetasdeg/180*pi
thetab=thetabdeg/180*pi

% print the current variable values to the workspace
fprintf('Upper Crank Offset = %5.2f mm 
',uppercrankoffsetmm);
fprintf('Upper Con Rod Length = %5.2f mm 
',upperconrodmm);
fprintf('Upper Crank Throw = %5.2f mm 
',uppercrankthrowmm);
fprintf('Upper Lower Crank Offset = %5.2f mm 
',crankoffsetmm);
fprintf('Upper Lower Con Rod Length = %5.2f mm 
',conrodmm);
fprintf('Upper Lower Crank Throw = %5.2f mm 
',crankthrowmm);
fprintf('Crank to Crank = %5.2f mm 
',cranktocrankmm);
fprintf('Ex-port Height = %5.2f mm 
',exportheightmm);
fprintf('Crank Lag = %5.2f deg 

',uppercranklag);
fprintf('Burn Start = %5.2f deg 

',thetasdeg);
fprintf('Burn Duration = %5.2f deg 

',thetabdeg);

% Plot the positions of the pistons in the RH figure
subplot(1,2,2);
axis on

% Set the number of cycles to display
zz=1;

% Set the range of crank angle for the plot with 360 increments
% t=0.01;
thetaplot=linspace(0,2*pi*zz,360);

% calculate the lower piston position for each crank angle
lower_pist_positionmm=sin(thetaplot+pi/2)*crankthrowmm+sqrt(conrodmm^2-(cos(thetaplot+pi/2)*crankthrowmm+crankoffsetmm)^2);

% find the lower piston closing the exhaust port position to use as the start of the plot
plotindex=max(find(lower_pist_positionmm==max(lower_pist_positionmm))); % find the index of the maximum value
thetacorrected=thetaplot+plotindex*pi/180+pi/2;

% Shift lower_pist_positionmm to start at exhaust port close
lower_pist_positionmm=sin(thetacorrected)*crankthrowmm+sqrt(conrodmm^2-(cos(thetacorrected)*crankthrowmm+crankoffsetmm).^2);

% Plot the lower piston position from the maximum value at zero crank angle
plot(thetaplot*180/pi,lower_pist_positionmm,'k');

% Plot the remaining lower piston position values to complete the piston position for the full range
plot(theta(plotindex:length(height)));

% Set the axis limits to the plotted cycles
axis([0 360*zz 0 cranktocrankmm]);

% Plot a line to show the height of the exhaust port
line([0 360*zz], [exportheightmm exportheightmm],'Color',[.8,.6,.5]);
text(10,exportheightmm-2,'exhaust port height','Color',[.8,.6,.5]);

% Create a 'legend' with respective coloured lines
line([10 20], [cranktocrankmm-1.5 cranktocrankmm-1.5],'Color',[0,0,1]);
text(20,cranktocrankmm-2,'Upper Piston','Color',[0,0,1]);
line([10 20], [cranktocrankmm-3.5 cranktocrankmm-3.5],'Color',[0,0,0]);
text(20,cranktocrankmm-5,'Lower Piston','Color',[0,0,0]);
text(20,cranktocrankmm-8,'Spec');
text(20,cranktocrankmm-11,'version');

% Title the plot
title('Piston Positions');

% Annotate the plot with the engines four 'cycles'
text(210,max(lower_pist_positionmm)-(max(lower_pist_positionmm)-exportheightmm)/4,'POWER','Color',[1,0,0],Rotation=[-60]);
text(250,exportheightmm+2,'EXHAUST','Color',[1,0,0],Rotation=[0]);
text(4,exportheightmm+5,'INDUCTION','Color',[0,0,1],Rotation=[60]);
text(60,max(lower_pist_positionmm)-(max(lower_pist_positionmm)-exportheightmm)/3,'COMPRESSION','Color',[0,0,0],Rotation=[30]);

% Calculate and plot the upper piston position for the crank angle range
upper_pist_positionmm=cranktocrankmm + (sin(thetacorrected+uppercranklag*pi/180)*uppercrankthrowmm+upperconrodmm^2-(cos(thetacorrected+uppercranklag*pi/180)*uppercrankthrowmm+uppercrankoffsetmm).^2);

plot(thetaplot*180/pi,upper_pist_positionmm);

% Display work out on plot
hold off
%input engine data from file enginedata;

% call pistonclash function to return the required values from the current upper and lower piston position vectors
[pistonclashmm,maxinvolmm,compressedvolmm,CR,maxexpandvolmm,piston_to_pistonmm,thetaignition]=pistonclash(upper_pist_positionmm, lower_pist_positionmm,exportheightmm);

% plot a line to show the position of maximum inducted volume
line([maxinvolindex, maxinvolindex], [lower_pist_positionmm(maxinvolindex)-2,... upper_pist_positionmm(maxinvolindex)+2], 'LineStyle','--')

% plot a line to show the position of minimum compressed volume
line([compressedvolindex, compressedvolindex], [lower_pist_positionmm(compressedvolindex)-2,... upper_pist_positionmm(compressedvolindex)+2], 'LineStyle','--')

thetaignitionindex=round(thetaignition*180/pi);
% plot a line to show the position of start of burn (ignition)
line([thetaignitionindex, thetaignitionindex], [lower_pist_positionmm(thetaignitionindex)-2,... upper_pist_positionmm(thetaignitionindex)+2], 'LineStyle','--')

% annotate effective compression ratio at ignition
volatignitemm=piston_to_pistonmm(round(thetaignition/pi*180));
CRIG=maxinvolmm/1000*pi*(b/2)^2/(volatignitemm/1000*pi*(b/2)^2+2*splugvolume);
ignitlabel=sprintf('C.R.Ign%4.1f',CRIG);
text(5, 65/(90/cranktocrankmm), ignitlabel)

% Annotate with minimum piston clearance at the end of exhaust cycle
pclabel=sprintf('P.C.%4.1f',pistonclashmm);
text(200, 2, pclabel)

% Annotate with minimum compressed volume (in units of linear displacement of the pistons)
compressedvollabel=sprintf('=%5.1f',compressedvolmm/1000*pi*(b/2)^2*1E6);
text(180, cranktocrankmm-19/(90/cranktocrankmm), 'Compressed')
text(180, cranktocrankmm-22/(90/cranktocrankmm), 'Volume (cc)')
text(180, cranktocrankmm-25/(90/cranktocrankmm), compressedvollabel)

% Annotate with the compression ratio
CRLabel=sprintf('C.R.=%5.1f',CR);
text(10, cranktocrankmm-16, CRLabel)

% Calculate the actual compression ratio after gas expands into spark plug
ACR=maxinvolmm/1000*pi*(b/2)^2/compressedvolmm/1000*pi*(b/2)^2*splugvolume;
ACRLabel=sprintf('A.C.R.=%5.1f',ACR);
text(10, cranktocrankmm-16/(90/cranktocrankmm), ACRLabel)

% Annotate with the effective inducted volume
inductvollabel=sprintf('=%5.1f',maxinvolmm/1000*pi*(b/2)^2*1E6);
text(maxinvolindex+5, exportheightmm+8/(90/cranktocrankmm), 'Inducted')
text(maxinvolindex+5, exportheightmm+5/(90/cranktocrankmm), 'Volume (cc)')
%text(maxinvolindex+5, 12, 'I.V.Units')
text(maxinvolindex+5, exportheightmm+2/(90/cranktocrankmm), inductvollabel)
% Annotate with the effective expanded volume
expandvollabel=sprintf(’ =%5.1f’,maxexpandvolmm/1000*pi*(b/2)^2*1E6);
text(280,exportheightmm+28/(90/cranktocrankmm),’Expanded’)
text(284,exportheightmm+25/(90/cranktocrankmm),’Volume (cc)’)
%text(4,12,’E.V. Units’)
text(287,exportheightmm+22/(90/cranktocrankmm),expandvollabel)

% Annotate with the effective ratio of expanded volume to inducted volume
AER=maxexpandvolmm/1000*pi*(b/2)^2/(compressedvolmm/1000*pi*(b/2)^2+2*splugvolume);
ERlabel=sprintf(’A.E.R.=%5.1f’,AER);
text(10,cranktocrankmm-19/(90/cranktocrankmm),ERlabel)
effectivevariabledisp=AER/ACR;
EVDlabel=sprintf(’E.V.D.R.=%5.1f’,effectivevariabledisp);
text(10,cranktocrankmm-22/(90/cranktocrankmm),EVDlabel)

%Display burn start and duration and Max engine pressure
burnstartlabel=sprintf(’Burn Start=%5.1f’,thetas/pi*180);
text(200,cranktocrankmm-3/(90/cranktocrankmm),burnstartlabel)
burndurationlabel=sprintf(’Burn Duration=%5.0f’,thetab/pi*180);
text(200,cranktocrankmm-6/(90/cranktocrankmm),burndurationlabel)

% create movie by saving the current figure as ten frames
% for framepause=1:10
% F((framecounter-1)*10+framepause) = getframe;
% end
% framecounter=framecounter+1
% Commence thermo and losses analysis when lower piston closes exhaust port
%analysis_start_index=max(find(lower_pist_positionmm<exportheightmm)) %analysis_start_index is NOT
%offset by plotindex

%Use input menu reference to run original or revised ESP
if k==1
   ahrind(splugport);
   whichvers=1
else
   ahrind2(splugport);
   whichvers=2
end

%load data from output of ESP
load ahrind.mat;
nn=4;
intwork=pTbWQlHl(end,nn-1);
worklabel=sprintf(’Int WORK =%5.1f’,intwork);
text(100, exportheightmm+25/(90/cranktocrankmm), worklabel)
efficiencylabel=sprintf(’Int EFF =%5.3f’,eta);
text(100, exportheightmm+22/(90/cranktocrankmm), efficiencylabel)

%conditional statement to enable full output data from ESP
if isempty(thetacomp2)
   engpressure=[pTuWQiHl(:,1);pTbTuWQiHl((2:end),1);pTbWQiHl((2:end),1)];
   thetaworkcycle=[thetacomp;thetacomb((2:end));thetaexp((2:end))];
else
   engpressure=[pTuWQiHl(:,1);pTbTuWQiHl((2:end),1);pTbWQiHl((2:end),1)];
   thetaworkcycle=[thetacomp;thetacomb((2:end));thetaexp((2:end))];
end
thetaworkcycleindex=[maxinvolindex:maxexpandvolindex];

%create eng pressure data at crank angle indeces
engpressure=interp1(thetaworkcycle,engpressure,thetaworkcycleindex/180*pi);

%Add engine pressure = 80,000 Pa during induction and engine press =120,000 Pa during exhaust
addcompzeros=zeros(1,maxinvolindex-1);
engpressure=[addcompzeros+80000,engpressure];
lengthadd=zeros(1,(360-length(engpressure)));
lengthadd=lengthadd+120000;
engpressure=[engpressure,lengthadd];

%Display RPM and max engine pressure
RPMLabel=sprintf('RPM=%5.0f',RPM);
text(200,cranktocrankmm-9/(90/cranktocrankmm),RPMLabel)

maxpresslabel=sprintf('Max MPa=%5.1f',max(engpressure)/1e6);
text(200,cranktocrankmm-12/(90/cranktocrankmm),maxpresslabel)

call friction function files
[lowerpistfrictwork,dSlowerdt,d2Slowerdt2,lowerconrodangle,conrodtensionlower,normallower,fAlower,...
  lowerpistfrictionvar,lowerpistfrictworkall,StNolower]=... 
  lowerpistfriction(engpressure,lower_pist_positionmm,thetacorrected,... 
  crankoffsetmm,conrodmill,crankthrowmm,omega,lowerpistfrictwork,uppercrankangle,...
  downstroke,muoil,mu,uppercranklag,cranktocrankmm);

[upperpistfrictwork,dSupperdt,d2Supperdt2,upperconrodangle,conrodtensionupper,normalupper,fAupper,...
  upperpistfrictionvar,upperpistfrictworkall,StNoupper]=...
  upperpistfriction(engpressure,upper_pist_positionmm,thetacorrected,...
  uppercrankoffsetmm,upperconroddmm,uppercrankthrowmm,omega,upperpistfrictwork,uppercrankangle,...
  downstroke,muoil,mu,uppercranklag,cranktocrankmm);

upperpistfrictwork;
lowerpistfrictwork;
pistfrictlosses=lowerpistfrictwork+upperpistfrictwork;

[lowerringfrictwork]=loweringfrictionfriction(dSlowerdt,d2Slowerdt2,lower_pist_positionmm,b,muoil,engpressure)
;
[upperringfrictwork]=upperringfrictionfriction(dSupperdt,d2Supperdt2,upper_pist_positionmm,b,muoil,engpressure)
;

upperringfrictwork;
lowerringfrictwork;
ringwork=loweringfrictwork+upperringfrictwork;
pumpwork=maxinvolmm/1000*pi*(b/2)^2*1E6/50;
%belt and bearing friction is proportional to load, they're proportional to
%inducted volume. Estimate belt/bearing work is 2% of engine work
work=[pTuWQlHl(:,nn-1);pTbTuWQlHl((2:end),nn);pTbWQlHl((2:end),nn-1)]';

%use input menu reference to define bearing friction option
bbFfactor=[0.02,0.02,.5];
if k==3
  beltheatingwork=work(end)*bbFfactor(k)+(RPM/60)^2*0.01+(RPM/60)*0.01*.25+10;
else
  beltheatingwork=work(end)*.02;
end

%calculate nett work and eff
network=intwork-pistfrictlosses-ringwork-pumpwork-beltbearingwork;
eteff=network/(intwork/eta);

networklabel=sprintf('Nett WORK %5.1f',network);
text(100,exportheightmm+19/(90/cranktocrankmm),networklabel)
etefficiencylabel=sprintf('Nett EFF %5.3f',neteff);
text(100,exportheightmm+16/(90/cranktocrankmm),netefficiencylabel)

%display engine pressure at exhaust port opening
engexpress=pTbWQlHl(end,1)/1e6;
engexpresslabel=sprintf('E.O.Press(Mpa)=%5.2f',engexpress);
text(100,exportheightmm+13/(90/cranktocrankmm),engexpresslabel)

end
Tadiabatic.m

function Tb=Tadiabatic(p,Tu,phi,f,fueltype,airscheme);
%
% Tb=Tadiabatic(p,Tu,phi,f,fueltype,airscheme)
%
% Routine for calculating the adiabatic flame temperature.
% Method involves iteratively selecting flame temperatures until
% the enthalpy of the combustion products (in equilibrium) matches
% the enthalpy of the initial gas mixture.
% farg.m is used to determine the enthalpy of the unburned mixture,
% and ecp.m is used to determine the enthalpy of the burned gas.
%
MaxIter=50;
Tol=0.00001; % 0.001% allowable error in temperature calculation
Tb=2000; % initial estimate
DeltaT=2*Tol*Tb; % something big
Iter=0;

[h,u,v,Y,cp,dlvT,dlvlp]=farg(p,Tu,phi,f,fueltype,airscheme);
while (Iter<MaxIter)&(abs(DeltaT/Tb)>Tol)
    Iter=Iter+1;
    [hb,u,v,Y,cp,dlvT,dlvlp]=ecp(p,Tb,phi,fueltype,airscheme);
    DeltaT=(hu-hb)/cp;
    Tb=Tb+DeltaT;
end
if Iter>=MaxIter
    warning('convergence failure in adiabatic flame temperature loop');
end

upperconrodinclination.m

function upperconrodangle=upperconrodinclination(upper_pist_positionmm,thetacorrected,...
    uppercrankoffssetmm,uppercrankrodmm,uppercrankthrowmm,omega,uppercranklag,cranktocrankmm)
%
% function to calculate the upper con rod inclination, upperconrodangle in radians.

for counter=1:length(upper_pist_positionmm)
    if xbigendmm(counter)<0 & (smallendmm(counter)-ybigendmm(counter))<0
        upperconrodangle(counter)=acos(xbigendmm(counter)/upperconrodmm(counter))+pi;
    elseif xbigendmm(counter)>0 & (smallendmm(counter)-ybigendmm(counter))<0
        upperconrodangle(counter)=acos(xbigendmm(counter)/upperconrodmm(counter))+pi;
    else
        upperconrodangle(counter)=acos(xbigendmm(counter)/upperconrodmm(counter))+pi;
    end
end
% Function to determine the side thrust on the upper piston by calculating
% the pressure in the cylinder, the con rod inclination and the piston
% acceleration.
% Assumes constant crankshaft speed

casepress=0.8*1e5;

massupperpist

% calc piston velocity
dSupperdt=diff(upper_pist_positionmm/1000)/(pi/180/omega);

% calc piston acceleration
d2Supperdt2=diff(dSupperdt)/(pi/180/omega);

% repeat first element to return vector to original length
d2Supperdt2 = spline(1:length(d2Supperdt2),d2Supperdt2,linspace(1,length(d2Supperdt2),360));

upperconrodangle=upperconrodinclination(upper_pist_positionmm,thetacorrected,...
                   uppercrankoffsetmm,upperconrodmm,uppercrankthrowmm,omega,massupperpist,b,thetaworkcycleindex,...
                   downstroke,muoil,mu,uppercranklag,cranktocrankmm);

% calc stroke for this application
stroke=(max(upper_pist_positionmm)-min(upper_pist_positionmm))/1000;

% Force in y-direction =acc in y-direction * piston mass
% Fy=d2Supperdt2*massupperpist;

% Force on piston (Fy) = force in con rod (Fconrod)*sin(upperconrodangle) -
% pressure in cylinder * area - friction ASSUME friction is hydrodynamic for piston speeds over 1 m/s.
% ASSUME boundary friction (metal to metal) for speed < 1 m/s
% with coeff of friction = 0.05

% As an initial estimate of normal force (to give bearing pressure) use
% mu=0.01 for entire piston motion, therefore

% Loop the friction calc using previously determined Sommerfield numbers n
% times

n=50;
fA=.001*ones(1,length(upperconrodangle));
for k=1:n;
    ftemp=fA;
    for counterA=1:length(upperconrodangle);
        if dSupperdt(counterA)<0
            upperpistfrictionvar(counterA)=abs((massupperpist*d2Supperdt2(counterA)-(engpressure(counterA)-
                                             casepress)*pi*(b/2)^2)/...
                                             (abs(tan(upperconrodangle(counterA)))/ftemp(counterA)+1));
        else
            upperpistfrictionvar(counterA)=-abs((massupperpist*d2Supperdt2(counterA)-(engpressure(counterA)-
                                             casepress)*pi*(b/2)^2)/...
                                             (abs(tan(upperconrodangle(counterA)))/ftemp(counterA)-1));
        end
    end
end
normalupper=upperpistfrictionvar./temp;

% determine Strubeck relation between # and coeff of friction (f) from plot

% use as input into Strubeck calculation

graphStNoR=[-5.8 -3];
graphfR=[ -3 -1.3];
StNofitR=polyfit(graphStNoR,graphfR,1);

graphStNoM=[-6.8 -6.5 -6.41 -6.1 -5.8];
graphfM=[-1.3 -1.5 -2.1 -2.85 -3];
StNofitM=polyfit(graphStNoM,graphfM,3);

% calc Strubeck number
StNo=munoil.*abs(dSupperdt)./(abs(normalupper)/(b*skirt));

% determine corresponding coeff of friction

fR=10.^(polyval((StNofitR),log10(StNo)));
fM=10.^(polyval((StNofitM),log10(StNo)));

for counterB=1:length(upperconrodangle);
    if StNo(counterB)<=10^-6.8 & dSupperdt(counterB)<=0
        upperpistfrictionvar(counterB)=abs((massupperpist*d2Supperdt2(counterB)-(engpressure(counterB)-casepress))*pi*(b/2)^2)/...
            (abs(tan(upperconrodangle(counterB)))/mu+1));
    elseif StNo(counterB)<=10^-6.8 & dSupperdt(counterB)>0
        upperpistfrictionvar(counterB)=-abs((massupperpist*d2Supperdt2(counterB)-(engpressure(counterB)-casepress))*pi*(b/2)^2)/...
            (abs(tan(upperconrodangle(counterB)))/mu-1));
    elseif StNo(counterB)>10^-6.8 & StNo(counterB)>10^-6.8 & dSupperdt(counterB)<=0
        upperpistfrictionvar(counterB)=abs((massupperpist*d2Supperdt2(counterB)-(engpressure(counterB)-casepress))*pi*(b/2)^2)/...
            (abs(tan(upperconrodangle(counterB)))/mu+1));
    elseif StNo(counterB)>10^-6.8 & StNo(counterB)>10^-6.8 & dSupperdt(counterB)>0
        upperpistfrictionvar(counterB)=-abs((massupperpist*d2Supperdt2(counterB)-(engpressure(counterB)-casepress))*pi*(b/2)^2)/...
            (abs(tan(upperconrodangle(counterB)))/mu-1));
    elseif StNo(counterB)<10^-3.5 & StNo(counterB)>10^-5.8 & dSupperdt(counterB)<=0
        upperpistfrictionvar(counterB)=abs((massupperpist*d2Supperdt2(counterB)-(engpressure(counterB)-casepress))*pi*(b/2)^2)/...
            (abs(tan(upperconrodangle(counterB)))/mu+1));
    elseif StNo(counterB)<10^-3.5 & StNo(counterB)>10^-5.8 & dSupperdt(counterB)>0
        upperpistfrictionvar(counterB)=-abs((massupperpist*d2Supperdt2(counterB)-(engpressure(counterB)-casepress))*pi*(b/2)^2)/...
            (abs(tan(upperconrodangle(counterB)))/mu-1));
    elseif StNo(counterB)>10^-3.5 & StNo(counterB)>10^-5.8 & dSupperdt(counterB)<=0
        upperpistfrictionvar(counterB)=abs((massupperpist*d2Supperdt2(counterB)-(engpressure(counterB)-casepress))*pi*(b/2)^2)/...
            (abs(tan(upperconrodangle(counterB)))/mu+1));
    else
        upperpistfrictionvar(counterB)=-abs((massupperpist*d2Supperdt2(counterB)-(engpressure(counterB)-casepress))*pi*(b/2)^2)/...
            (abs(tan(upperconrodangle(counterB)))/mu-1));
    end
end

% obtain upper piston position increments with 'diff'
upper_pist_position_diff=diff(upper_pist_positionmm/1000);
upper_pist_position_diff=spline(1:length(upper_pist_position_diff),upper_pist_position_diff,...
linspace(1,length(upper_pist_position_diff),360));

% upperpistfrictworkall=abs(upperpistfrictionvar).*abs(upper_pist_position_diff);
%
% upperpistfrictwork=sum(upperpistfrictworkall);

% calc conrod compress/tension load (tension=positive)
for counterC=1:length(upperconrodangle);
    if StNo(counterC)<=10^-6.8
        normalupper(counterC)=upperpistfrictionvar(counterC)/mu;
        fA(counterC)=mu;
    elseif StNo(counterC)<=10^-5.8
        normalupper(counterC)=upperpistfrictionvar(counterC)/fM(counterC);
        fA(counterC)=fM(counterC);
    elseif StNo(counterC)<=10^-3.5
        normalupper(counterC)=upperpistfrictionvar(counterC)/fR(counterC);
        fA(counterC)=fR(counterC);
    else
        normalupper(counterC)=upperpistfrictionvar(counterC)/0.027;
        fA(counterC)=0.027;
    end
end

conrodtensionupper=(-massupperpist*d2Supperdt2+(engpressure-casepress)*pi*(b/2)^2+...
    upperpistfrictionvar)./sin(upperconrodangle);

upperpistfrictionvar

end
fAupper=fA;
upperpistfrictionvar;
normalupper;
StNoupper=StNo;
upperpistfrictworkall=abs(upperpistfrictionvar).*abs(upper_pist_position_diff);
upperpistfrictwork=sum(upperpistfrictworkall);

upperringfriction.m

function
[upperringfrictwork]=upperringfriction(dSupperdt,d2Supperdt2,upper_pist_positionmm,b,muoil,mu,engpressure) ;

upperringtension=20;
ringwidth=0.0023;

% determine Stribeck relation between # and coeff of friction (f) from plot
% use as input into Stribeck calculation
graphStNoR=[-5.8 -3];
graphfR=[-3 -1.3];
StNofitR=polyfit(graphStNoR,graphfR,1);

graphStNoM=[-6.8 -6.5 -6.4 -6.1 -5.8];
graphfM=[-1.3 -1.5 -2.1 -2.85 -3];
StNofitM=polyfit(graphStNoM,graphfM,3);

% calc Stribeck number
StNo=muoil*abs(dSupperdt)./(upperringtension/(b*pi*ringwidth)+engpressure);
\[ \text{StNo2} = \mu_{oil} \cdot \frac{\text{abs}(d_{Upper})}{\text{upper ring tension}/(b \cdot \pi \cdot \text{ring width}) + \text{eng pressure}/2}; \]

\% determine corresponding coeff of friction
\[ f_R = 10^\cdot \text{polyval}((\text{StNofitR}), \log10(\text{StNo})); \]
\[ f_M = 10^\cdot \text{polyval}((\text{StNofitM}), \log10(\text{StNo})); \]

\% obtain upperer piston position increments with diff
\[ \text{upper piston position diff} = \text{diff}(\text{upper piston position mm}/1000); \]
\[ \text{upper piston position diff} = \text{bspline}(1 : \text{length}(\text{upper piston position diff}), \text{upper piston position diff}, \ldots \text{limspace}(1, \text{length}(\text{upper piston position diff}), 360)); \]

for counter = 1:length(eng pressure);
    if StNo(counter) <= 10^{-6.8}
        upperringfriction(counter) = \mu \cdot (\text{upper ring tension} + \text{eng pressure}(\text{counter}) \cdot \pi \cdot b \cdot \text{ring width}) \cdot \text{sign}(\text{dUpperdt}(\text{counter}));
    elseif StNo(counter) <= 10^{-5.8}
        upperringfriction(counter) = f_M(\text{counter}) \cdot (\text{upper ring tension} + \text{eng pressure}(\text{counter}) \cdot \pi \cdot b \cdot \text{ring width}) \cdot \text{sign}(\text{dSupperdt}(\text{counter}));
    else
        upperringfriction(counter) = f_R(\text{counter}) \cdot (\text{upper ring tension} + \text{eng pressure}(\text{counter}) \cdot \pi \cdot b \cdot \text{ring width}) \cdot \text{sign}(\text{dSupperdt}(\text{counter}));
    end
end

for counter2 = 1:length(eng pressure);
    if StNo(counter2) <= 10^{-6.8}
        upperringfriction2(counter2) = \mu \cdot (\text{upper ring tension} + \text{eng pressure}(\text{counter2}) \cdot \pi \cdot b \cdot \text{ring width}) \cdot \text{sign}(\text{dSupperdt}(\text{counter2}));
    elseif StNo(counter2) <= 10^{-5.8}
        upperringfriction2(counter2) = f_M(\text{counter2}) \cdot (\text{upper ring tension} + \text{eng pressure}(\text{counter2}) \cdot \pi \cdot b \cdot \text{ring width}) \cdot \text{sign}(\text{dSupperdt}(\text{counter2}));
    else
        upperringfriction2(counter2) = f_R(\text{counter2}) \cdot (\text{upper ring tension} + \text{eng pressure}(\text{counter2}) \cdot \pi \cdot b \cdot \text{ring width}) \cdot \text{sign}(\text{dSupperdt}(\text{counter2}));
    end
end

upperringfrictionall = \text{abs}(\text{upperringfriction}) \cdot \text{abs}(\text{upper piston position diff});
upperringfrictionall2 = \text{abs}(\text{upperringfriction2}) \cdot \text{abs}(\text{upper piston position diff});
upperringfrictionall = \text{sum}(\text{upperringfrictionall}) + \text{sum}(\text{upperringfrictionall2});
List of default engine input specifications files

Files used in optimisation for constant compression ratio (10:1):
enginespecCCR3.m
enginespecCCR4.m
enginespecCCR4_5.m
enginespecCCR5.m
enginespecCCR6.m
enginespecCCR6_2.m
enginespecCCR6_5.m
enginespecCCR7.m
enginespecCCR8.m
enginespecCCR8_5.m
enginespecCCR9.m
enginespecCCR9_2.m
enginespecCCR9_5.m
enginespecCCR10.m
enginespecCCR11.m

Files used in optimisation for constant maximum engine pressure (6MPa):
enginespecoptsize3.m
enginespecoptsize4.m
enginespecoptsize4_5.m
enginespecoptsize5.m
enginespecoptsize6.m
enginespecoptsize6_2.m
enginespecoptsize6_5.m
enginespecoptsize7.m
enginespecoptsize8.m
enginespecoptsize8_5.m
enginespecoptsize9.m
enginespecoptsize9_2.m
enginespecoptsize9_5.m
enginespecoptsize10.m
enginespecoptsize11.m
Appendix C – Matlab model plot scripts and representative plots

A functional softcopy of the following list of Matlab plot files is contained on the submitted Dissertation CD. Printed versions of the plot files are in Appendix B

plot_both_conrod_inclination.m
plot_both_pist_friction.m
plot_both_piston_acc.m
plot_both_piston_vel.m
plot_both_StNo.m
plot_both_StNo_10toneg5_8.m
plot_both_StNo_10toneg6_8.m
plot_both_StNo4.m
plot_compare_upperpist_friction_4_7_10.m
plot_eff_disp_CCR10_0_M9.m
plot_eff_piston_disp.m
plot_engine_pressure.m
plot_lower_conrod_tension.m
plot_lower_normal_piston_load.m
plot_lower_pist_friction.m
plot_lower_piston_acc.m
plot_lower_piston_vel.m
plot_rel_angle_lowerconrod_crank.m
plot_rel_angle_upperconrod_crank.m
plot_rel_piston_position.m
plot_upper_conrod_tension.m
plot_upper_normal_piston_load.m
plot_upper_piston_acc.m
plot_upper_piston_vel.m
plot_work.m
plot_work_cycle_pressure.m
plotresults.m
‘plotresults.m’ plots the output performance specifications for the engine simulation program in comparison to the original engine simulation program outputs which were confirmed to correlate with Ferguson’s [9] experimentally confirmed data. The following plots are for the corrected thermodynamic cycle Matlab model for the prototype as modelled at 2000 RPM as was the original ESP simulation.
Opposed Piston Engine Cycle Cumulative Work

Reference Engine Cycle Cumulative Work
The following are sample plots for the Matlab model using an engine speed of 2000 RPM and the corrected thermodynamic cycle.
Piston Friction Vs Crank Angle after E.C.

Piston Acceleration Vs Crank Angle after E.C.
Lower Normal Piston Load Vs Crank Angle after E.C.

Crank Angle after E.C. (Deg) vs Normal Force (KPa)

Piston Friction Vs Crank Angle after E.C.

Crank Angle after E.C. (Deg) vs Friction (N)(temporal)

Note: the low proportion of the full scale used by the plot reflects the low forces relative the engine's capacity at maximum operating pressure.
The following plots were based on the original Matlab model before the corrections associated with information from the prototype tests. The Matlab model uses an engine speed of 2000 RPM.
Friction Work Vs Inducted Volume for Various Eng Specs

Bore = 61.3mm
E.O. Press = 1 atm
Const Compression Ratio = 10.0
Const Burn Start = -27 deg

Efficiency Vs Inducted Volume for Various Eng Specs

Bore = 61.3mm
E.O. Press = 1 atm
Const Maximum Pressure = 6.5 MPa
Const Burn Start = -27 deg

A.C.R. = Actual Compression Ratio
(Max Inducted Vol / Min Compressed Vol)
Appendix D – Engine Components

The fabricated main components for the engine/test assembly

<table>
<thead>
<tr>
<th>Engine Part</th>
<th>Modelled in ProE</th>
<th>Drawings (Appendix E)</th>
<th>Material</th>
<th>Manufacturer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lower Crank Case</td>
<td>Yes</td>
<td>Functional</td>
<td>Mild Steel</td>
<td>Author</td>
</tr>
<tr>
<td>Lower Crank Shaft</td>
<td>Yes</td>
<td>Workshop Standard</td>
<td>4140 Steel USQ Workshop</td>
<td></td>
</tr>
<tr>
<td>Lower Con Rod</td>
<td>Yes</td>
<td>Functional</td>
<td>Aluminium (5000 series)</td>
<td>Author</td>
</tr>
<tr>
<td>Lower Piston</td>
<td>Yes</td>
<td>Functional</td>
<td>4140 Steel</td>
<td>Author</td>
</tr>
<tr>
<td>Upper Crank Case</td>
<td>Yes</td>
<td>Functional</td>
<td>Mild Steel</td>
<td>Author</td>
</tr>
<tr>
<td>Upper Crank Shaft</td>
<td>Yes</td>
<td>Workshop Standard</td>
<td>4140 Steel USQ Workshop</td>
<td></td>
</tr>
<tr>
<td>Upper Con Rod</td>
<td>Yes</td>
<td>Functional</td>
<td>Aluminium (5000 series)</td>
<td>Author</td>
</tr>
<tr>
<td>Upper Piston</td>
<td>Yes</td>
<td>Functional</td>
<td>4140 Steel</td>
<td>Author</td>
</tr>
<tr>
<td>Barrel (Cylinder)</td>
<td>Yes</td>
<td>Functional</td>
<td>Cast Iron P2</td>
<td>Author</td>
</tr>
</tbody>
</table>

The fabricated ancillary components for the engine/test assembly

<table>
<thead>
<tr>
<th>Engine Part</th>
<th>Modelled in ProE</th>
<th>Photograph (Appendix F)</th>
<th>Material</th>
<th>Manufacturer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exhaust collector</td>
<td>No</td>
<td>No</td>
<td>Mild steel</td>
<td>Author</td>
</tr>
<tr>
<td>Exhaust pipe</td>
<td>No</td>
<td>(see engine photographs)</td>
<td>Mild steel</td>
<td>Author</td>
</tr>
<tr>
<td>Induction reed valve</td>
<td>No</td>
<td>Yes</td>
<td>Aluminium / spring steel</td>
<td>Author</td>
</tr>
<tr>
<td>Engine Part</td>
<td>Material</td>
<td>Source/model</td>
<td>Manufacturer</td>
<td></td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------------------------</td>
<td>---------------------</td>
<td>--------------</td>
<td></td>
</tr>
<tr>
<td>Induction Goose neck tube</td>
<td>No</td>
<td>(see engine photographs)</td>
<td>Author</td>
<td></td>
</tr>
<tr>
<td>Engine frame</td>
<td>No</td>
<td>(see engine photographs)</td>
<td>Author</td>
<td></td>
</tr>
<tr>
<td>Belt guard</td>
<td>No</td>
<td>(see engine photographs)</td>
<td>Author</td>
<td></td>
</tr>
<tr>
<td>Belt tensioner pulley</td>
<td>No</td>
<td>(see engine photographs)</td>
<td>Author</td>
<td></td>
</tr>
<tr>
<td>Tensioner bracket</td>
<td>No</td>
<td>No</td>
<td>Author</td>
<td></td>
</tr>
<tr>
<td>Pick-up prong</td>
<td>No</td>
<td>Yes</td>
<td>Author</td>
<td></td>
</tr>
<tr>
<td>Ignition timing adjuster</td>
<td>No</td>
<td>No</td>
<td>Author</td>
<td></td>
</tr>
<tr>
<td>Crankcase covers</td>
<td>No</td>
<td>(see engine photographs)</td>
<td>Author</td>
<td></td>
</tr>
</tbody>
</table>

The commercially manufactured components acquired for the engine/test assembly

<table>
<thead>
<tr>
<th>Engine Part</th>
<th>Material</th>
<th>Source/model</th>
<th>Manufacturer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Piston gudgeon pins</td>
<td>Centre-less ground</td>
<td>Victa PowerTorque</td>
<td>Victa</td>
</tr>
<tr>
<td>(wrist) pins and</td>
<td>case hardened steel</td>
<td>160</td>
<td></td>
</tr>
<tr>
<td>clips</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conrod Bearings</td>
<td>needle roller bearings</td>
<td>&quot;</td>
<td>Victa</td>
</tr>
<tr>
<td>Crankshaft Pins</td>
<td>hardened steel</td>
<td>&quot;</td>
<td>Victa</td>
</tr>
<tr>
<td>Outer crankshaft</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Victa</td>
</tr>
<tr>
<td>bearings</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Piston rings</td>
<td>Cast iron</td>
<td>&quot;</td>
<td>Victa</td>
</tr>
<tr>
<td>Inner crankshaft</td>
<td>6008</td>
<td>NTN Bearings</td>
<td></td>
</tr>
<tr>
<td>bearings</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Component</td>
<td>Material</td>
<td>Model</td>
<td>Manufacturer</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>--------------</td>
<td>-------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Synchronous belt pulleys</td>
<td>Cast iron</td>
<td>8M36S12</td>
<td>Gates</td>
</tr>
<tr>
<td>Pulley taper lock bushes</td>
<td>Cast iron</td>
<td>1610-20</td>
<td>Gates</td>
</tr>
<tr>
<td>Synchronous Belt</td>
<td>Polychain GT2</td>
<td>8MGT-720-12</td>
<td>Gates</td>
</tr>
<tr>
<td>Carburettor</td>
<td></td>
<td></td>
<td>Villiers</td>
</tr>
<tr>
<td>Ignition module</td>
<td>BIM 024</td>
<td></td>
<td>Bosch</td>
</tr>
<tr>
<td>Ignition Coil</td>
<td>BIC800</td>
<td></td>
<td>Bosch</td>
</tr>
<tr>
<td>Spark plugs</td>
<td>CR5HSB</td>
<td></td>
<td>NGK</td>
</tr>
<tr>
<td>Fuel tank</td>
<td>Polymer</td>
<td>Second hand</td>
<td>Victa</td>
</tr>
<tr>
<td>Electrical wiring, throttle cable/lever, fuel line, fittings, fixtures</td>
<td></td>
<td>As required</td>
<td></td>
</tr>
</tbody>
</table>
Appendix E – Solid Model Images and Drawings of Engine Components

Crankshaft

Counterbalance

Crankshaft Assembly

Lower Piston

Upper Piston

Connecting Rod (upper)
Barrel (Cylinder)

Lower Crankcase
Appendix F – Engine Component Photographs

Engine test assembly – shown here with crankcase pump connected

Upper piston in original form
Upper piston showing combustion ‘print’ after combustion chamber addition

Lower crankcase cover with crankcase pump ports exposed – out reed removed.
Fuel tank and mount showing fuel flow measurement tube

Reed valve – outlet and inlet sides
The three ignition prong plates used

The original prototype engine
Engine view showing 12V starter installed