

A User-Friendly, Two-Zone Heat Release Model for Predicting Spark-Ignition  
Engine Performance and Emissions

A Thesis

Presented in Partial Fulfillment of the Requirements for the

Degree of Master of Science

with a

Major in Mechanical Engineering

in the

College of Graduate Studies

University of Idaho

by

Jeremy L. Cuddihy

May 2014

Major Professor: Steven Beyerlein, Ph.D.



## Abstract

A single-zone heat release model was developed in MATLAB to predict engine performance and efficiencies on a Yamaha YZ250F engine. The general single-zone model was expanded to include friction losses as a function of engine speed, combustion efficiencies at different operating points, temperature-dependent thermodynamic properties, and residual gas fractions. The single-zone model was then split into a two zone model using burned and unburned volume fractions. The elevated burned zone temperature provided by the two-zone model was used to predict NO and HC emissions.

Upon completing the heat release model, a Yamaha YZ250F engine was tested on an eddy-current dynamometer. Data points were gathered over multiple throttle positions, between operating speeds of 4500-9000 RPM. In comparing model predictions with actual data, it was found that all power predictions were accurate to within  $\pm 3\%$ , while all BSFC predictions were accurate to within  $\pm 10\%$ . It was theorized that the relative error in BSFC predictions could be minimized by documenting fuel temperatures and updating these within the model. In comparing emissions predictions to experimental data, it was found that all emissions predictions were accurate to within  $\pm 15\%$ .

It's expected that future research will heavily involve this model. With a volumetric efficiency map and desired air-fuel ratio map, this model can be used to construct optimized spark-timing maps, thus reducing down-time when tuning engines. The chemical model derived in this research can be expanded to include alternative fuels and carbon-monoxide emissions predictions. The heat release model can also be modified to predict two-stroke engine performance and emissions.

## Acknowledgments

Throughout this research, I have been the recipient of a great deal of support from friends, family, and colleagues. First and foremost, I would like to thank my fiancée, Bridget, for supporting me throughout graduate school. Without her reassurance and patience, I would not be in the position that I am today.

I would like to thank my major professor, Dr. Steve Beyerlein. Dr. Beyerlein piqued my interest in internal combustion engines when I took ME 433 in the summer of 2012. After taking ME 433, Dr. Beyerlein gave me the opportunity to work for Idaho Engineering Works (IEW), where I have conducted my research. Any time I have needed assistance with my model, Dr. Beyerlein has provided me with insight and support.

I would like to thank Dr. Dan Cordon for keeping an open-door policy with regard to questions about motorcycle engines, wiring, and experimental setups. Dr. Cordon met with me on a weekly basis throughout the course of my research and assisted me with engine troubleshooting. I would also like to thank Rory Lilley and Dillon Savage for spending endless hours troubleshooting engine and wiring issues during testing.

Lastly, I would like to thank the National Institute for Advanced Transportation Technology (NIATT), the Department of Mechanical Engineering, and the Department of Graduate Studies for financially supporting me. Without funding, this research would not have been possible.

## Table of Contents

Authorization to Submit Thesis .....	ii
Abstract .....	iii
Acknowledgments .....	iv
Table of Contents .....	v
List of Tables .....	xi
Chapter 1. Introduction.....	1
1.1 Background and Motivation.....	1
1.2 Thesis Objectives .....	2
1.3 Thesis Organization.....	2
Chapter 2. Literature Review.....	4
2.1 Heat Release Models .....	5
2.2 Convective Heat Transfer Models.....	6
2.3 NO Formation Mechanisms .....	9
2.4 NO Modeling Techniques .....	11
2.5 HC Formation Mechanisms .....	13
2.6 HC Modeling Techniques .....	16
2.7 Model Selection.....	17
Chapter 3. Governing Equations .....	20
3.1 Single-Zone Engine Model .....	20
3.2 Woschni's Heat Transfer Model .....	24
3.3 Annand's Heat Transfer Model.....	26
3.4 Temperature-Dependent Specific Heat Ratios ( $\gamma$ ) .....	28
3.5 Determination of Air-Fuel Ratio .....	28
3.6 Engine Friction Model .....	29

3.7	Residual Gas Fraction Model.....	30
3.8	Two-Zone Engine Model .....	31
3.9	NO Emissions Prediction .....	34
3.10	HC Emissions Prediction .....	39
Chapter 4. MATLAB Engine Simulation Model .....		44
4.1	Components of the MATLAB Model .....	44
4.2	Engine Geometry and Atmospheric Inputs .....	45
4.3	Pre-allocation of Arrays and Matrices .....	47
4.4	Combustion Efficiency and Fuel Inputs.....	47
4.5	Instantaneous Engine and Fluid Properties .....	48
4.6	Valve Simulation.....	49
4.7	Two-zone calculations.....	50
4.8	Plot Statements.....	50
4.9	Emissions Prediction Functions .....	51
Chapter 5. MATLAB Model Validation .....		53
5.1	Model Inputs .....	53
5.2	Weibe Function Validation .....	55
5.3	Pressure Trace Validation .....	56
5.4	Two-Zone Temperature Validation.....	57
5.5	Heat Transfer Validation.....	59
5.6	Validating Emissions Models.....	63
Chapter 6. Experimental Setup.....		66
6.1	Repairing the Dynamometer Engine .....	67
6.2	Mounting the Test-Engine to the Eddy-Current Dynamometer.....	70
6.3	Five-Gas Analyzer Setup and Calibration.....	71

6.4	Calibrating the ECU .....	75
6.5	Setting-up the Fuel Measurement System.....	75
6.6	Equipment Uncertainties .....	77
Chapter 7.	Engine Testing .....	81
7.1	Test Plan.....	81
7.2	Test Results .....	82
Chapter 8.	ME 433 Learning Activity .....	88
8.1	Guided Exploration of the Model.....	88
8.2	Mini-Project Description.....	90
8.3	Optimizing Spark-Timing .....	91
8.4	Generation of BSFC Maps .....	95
8.5	Estimating Engine Emissions.....	99
Chapter 9.	Conclusions and Recommendations .....	103
9.1	Summary of Accomplishments .....	103
9.2	Application of the Model in Future Research .....	104
9.3	Model Improvements .....	105
9.4	Test-Stand Improvements .....	105
References	.....	107
APPENDIX A:	Derivation of the Polynomial Method.....	109
APPENDIX B:	Expanding the Zeldovich Mechanism.....	111
APPENDIX C:	Calculating the Water-Gas Shift Constant .....	113
APPENDIX E:	MATLAB Script.....	115
APPENDIX F:	NO <sub>x</sub> Function .....	125
APPENDIX G:	HC Function .....	128
APPENDIX H:	Correction Factor .....	130

APPENDIX I: Load Function.....	132
APPENDIX J: BSFC and Emissions Mapping Call Function.....	134
APPENDIX K: BSFC and Emissions Mapping Function .....	137
APPENDIX L: Spark- Timing Call Function.....	146
APPENDIX M: Spark- Timing Function .....	148
APPENDIX N: A User’s Guide to Two-Zone Simulations.....	154
APPENDIX O: Optimizing Spark-Timing .....	158
APPENDIX P: BSFC and Emissions Mapping Activity.....	165
APPENDIX Q: Heat Release Class Notes.....	180

## List of Figures

Figure 1. A comparison of heat transfer prediction methods [4].....	9
Figure 2. Pie chart of HC emissions contributions. ....	14
Figure 3. Sources of HC crevice losses [14].....	15
Figure 4. Engine geometry and variables used in modeling. ....	22
Figure 5. Two-zone diagram.....	32
Figure 6. Engine inputs in the MATLAB model. ....	45
Figure 7. MATLAB calculations based on engine inputs.....	46
Figure 8. Fuel inputs and combustion efficiencies. ....	48
Figure 9. The Weibe function and fuel mass within the cylinder. ....	49
Figure 10. The opening and closing of valves. ....	49
Figure 11. Fuel-rich calculations. ....	52
Figure 12. The simulated YZ250F burn profile.....	56
Figure 13. Indicated cylinder pressure.....	57
Figure 14. Burned and unburned temperatures.....	58
Figure 15. Indicated net heat transfer. ....	59
Figure 16. A comparison of Annand's and Woschni's heat transfer predictions. ....	60
Figure 17. The predicted heat transfer coefficients as a function of crank angle. ....	61
Figure 18. Refined curve-fitting of the CO equilibrium constant.....	64
Figure 19. Refined relative error for CO equilibrium constant. ....	64
Figure 20. Refined relative error for CO <sub>2</sub> equilibrium constant. ....	65
Figure 21. Experimental setup. ....	66
Figure 22. YZ250F cylinder head and scored bearings. ....	67
Figure 23. YZ250F cylinder condition. ....	68
Figure 24. Plugged oil pickup screen.....	69
Figure 25. The engine mounted to the eddy-current dynamometer.....	70
Figure 26. Dust filter element. ....	72
Figure 27. Pre-filter element. ....	73
Figure 28. Strainer filter element. ....	73
Figure 29. Horiba gas calibration bottle. ....	74

Figure 30. Max 710 series fuel measurement system.....	76
Figure 31. Diagram of fuel-system.....	77
Figure 32. BSFC map over data in table 14.....	84
Figure 33. Refined BSFC map.....	85
Figure 34. Flow-chart of the MATLAB model's inputs and outputs.....	89
Figure 35. Spark-timing loop.....	92
Figure 36. Spark-advance curve.....	93
Figure 37. Optimal spark-advance at 6,000 RPM.....	94
Figure 38. Optimal spark-advance relative to a change in air-fuel ratio.....	95
Figure 39. The BSFC call function.....	96
Figure 40. Modeling spark –advance.....	97
Figure 41. BSFC plot statements.....	98
Figure 42. BSFC map produced with MATLAB.....	99
Figure 43. NO emissions map for the YZ250F.....	100
Figure 44. HC emissions map for the YZ250F.....	101
Figure 45. NO emissions at 13000 RPM as a function of $\lambda$ .....	102

## List of Tables

Table 1. A Comparison of Heat Release Models [1] .....	6
Table 2. Engine models adopted in this work.....	19
Table 3. Constants for the General HC Equation [3].....	36
Table 4. MATLAB Inputs When Running the YZ250F simulation. ....	54
Table 5. Predicted location of mass-fraction burned percentages .....	55
Table 6. A Comparison of Woschni's and Annand's Heat Transfer Prediction Methods.....	62
Table 7. Indicated outputs using Otto's method .....	62
Table 8. MoTec Inputs.....	75
Table 9. Horiba Display Resolution Uncertainty.....	78
Table 10. Horiba Equipment Errors.....	79
Table 11. Horiba Uncertainty .....	79
Table 12. Borghi and Saveri Eddy-Current Dynamometer Uncertainties[24] .....	79
Table 13. Fuel Measurement System Equipment Uncertainties .....	80
Table 14. Refining BSFC Results.....	82
Table 15. Experimental Data .....	83
Table 16. Comparing Model Outputs with Experimental Data .....	86
Table 17. Yamaha R6 Experimental Data .....	87
Table 18. Comparing Emissions Predictions to Experimental Data.....	87

## Key Variables

<u>Variable</u>	<u>Definition</u>	<u>Units</u>
$A, A_w$	Instantaneous Heat Transfer Area	$m^2$
$AF_{ac}$	Actual Air-Fuel Ratio	unit-less
$AF_{stoich}$	Stoichiometric Air-Fuel Ratio	unit-less
$a$	Crank Radius	m
$B$	Cylinder Bore	m
$C_r$	Compression Ratio	unit-less
$C_v$	Constant Volume Specific Heat	$\frac{J}{K}$
$d_{splug}$	Spark Plug Offset	m
$f, x_r$	Residual Gas Fraction	unit-less
$f_{ox}$	Oxidized Fraction	unit-less
$f_{unburned}$	Unburned Fraction	unit-less
$f_{vapor}$	Vapor Fraction	unit-less
$H$	Henry's Constant	(bar)
$h, h_w, h_c$	Convective Heat Transfer Coefficient	$\frac{W}{m^2 * K}$
$h_r$	Radiative Heat Transfer Coefficient	$\frac{W}{m^2 * K}$
$K_p$	Equilibrium Constant	unit-less
$K_{wgs}$	Water Gas Shift Equilibrium Constant	unit-less
$k, k_{gas}$	Thermal Conductivity	$\frac{W}{m * K}$
$k_{1f}$	Forward Reaction Rate Coefficient	$\frac{cm^3}{gmol * s}$
$L$	Engine Stroke	m
$LHV$	Lower Heating Value	$\frac{J}{kg}$
$l_r$	Connecting Rod Length	m
$N_{cyl}$	Number of Cylinders	unit-less
$N_{tot}$	Total Number Of Increments	unit-less

<b>Nu</b>	Nusselt Number	unit-less
<b>m</b>	Cylinder Mass	kg
<b>m<sub>a</sub></b>	Mass Of Air	kg
<b>m<sub>b</sub></b>	Burned Mass	kg
<b>m<sub>c</sub></b>	Combustion Mass	kg
<b>m<sub>crevice</sub></b>	Crevice Mass	kg
<b>m<sub>f</sub></b>	Mass Of Fuel	kg
<b>m<sub>HC</sub></b>	Mass Of Hydrocarbons	kg
<b>m<sub>oil</sub></b>	Mass Of Oil On Chamber Walls	kg
<b>m<sub>u</sub></b>	Unburned Mass	kg
<b>N</b>	Revolutions Per Second	$\frac{\text{rev}}{\text{s}}$
<b>n</b>	Polytropic Index	unitless
<b>P</b>	Cylinder Pressure	Pa
<b>P<sub>inlet</sub></b>	Inlet Pressure	atm
<b>P<sub>e</sub></b>	Exhaust Pressure	Pa
<b>P<sub>fuel</sub></b>	Partial Pressure Of Absorbed Fuel	atm
<b>P<sub>m</sub></b>	Motored Pressure	Pa
<b>P<sub>peak</sub></b>	Peak Cylinder Pressure	atm
<b>P<sub>r</sub></b>	Reference Pressure	Pa
<b>ΔP<sub>c</sub></b>	Pressure Rise Due to Combustion	Pa
<b>ΔP<sub>v</sub></b>	Pressure Rise Due to Volumetric Expansion	Pa
<b>Q</b>	Heat Energy Transfer	J
<b>Q<sub>ch</sub></b>	Chemical Heat Release	J
<b>Q<sub>w</sub></b>	Heat Transfer To The Wall	J
<b>R</b>	Universal Gas Constant	$\frac{\text{J}}{\text{kg} * \text{K}}$
<b>Re</b>	Reynold's Number	Unit-less
<b>RPM</b>	Revolutions Per Minute	$\frac{\text{rev}}{\text{min}}$
<b>s</b>	Distance Between Crank, Piston Axes	m

$\bar{S}_p$	Mean Piston Speed	$\frac{m}{s}$
$T$	Cylinder Temperature	K
$T_b$	Burned Temperature	K
$T_{BDC}$	Temperature At Bottom Dead Center	K
$T_{corr}$	Corrected Temperature	K
$T_r$	Reference Temperature	K
$T_u$	Unburned Temperature	K
$T_w$	Cylinder Wall Temperature	K
$U$	Internal Energy	J
$V$	Volume	$m^3$
$V_c$	Clearance Volume	$m^3$
$V_d$	Displaced Volume	$m^3$
$V_r$	Reference Volume	$m^3$
$W$	Work Energy Transfer	J
$w$	Average Gas Velocity	$\frac{m}{s}$
$x$	Characteristic Length	m
$x_{fuel}$	Molar Fraction Of Absorbed Fuel	unit-less
$X_b$	Mass Fraction Burned	unit-less
$\alpha$	Fraction Of Dissociation	unit-less
$\delta_{oil}$	Oil Layer Thickness	m
$\varepsilon$	Fuel Hydrogen To Carbon Ratio	unit-less
$\eta_c$	Combustion Efficiency	unit-less
$\gamma$	Specific Heat Ratio	unit-less
$\lambda$	Excess Air Coefficient	unit-less
$\mu$	Viscosity	$\frac{kg}{m * s}$
$\rho$	Gas Density	$\frac{kg}{m^3}$
$\rho_{oil}$	Oil Density	$\frac{kg}{m^3}$

$\theta$	Crank Angle	Degrees [°]
$\theta_b$	Burn Duration	Degrees [°]
$\theta_0$	Spark Advance	Degrees [°]

## Chapter 1. Introduction

### 1.1 Background and Motivation

Each summer, the University of Idaho offers an internal combustion (IC) engines course (ME 433) in which students learn about spark-ignition (SI) and compression-ignition (CI) engines, road load modeling, and numerical engine modeling. During the summer of 2013, a basic version of the MATLAB model from this thesis was incorporated into students' final projects as a means of calculating engine performance at different operating points. Students were required to simulate brake-specific fuel consumption (BSFC) maps and analyze pressure and temperature curves on a four-stroke motorcycle engine.

In the summer of 2014, students will use the most up-to-date version of this model to compare theoretical and analytical data on an engine of their choice. Students will use the finalized MATLAB model to predict nitric oxide (NO) and hydrocarbon (HC) emissions, exhaust gas recirculation (EGR) effects, and combustion and volumetric inefficiencies. With the additions of emissions predictions and EGR, students will be able to observe the relative emissions contribution of add-ons, such as camshaft replacements, and will also be able to see the effect of EGR on NO emissions.

In addition to in-class activities, this model will be useful for the future development of engine and emissions models centered on competition-based vehicles, such as the formula hybrid car. A simple engine performance and emissions model will assist students in evaluating the overall effect of changing performance parts. Instead of using a trial-and-error method of increasing performance, students will have analytical results to justify the

purchasing of new parts. With increasingly stringent emissions rules in these competitions, an emissions prediction model becomes progressively more important.

## **1.2 Thesis Objectives**

This research was performed to further advance the ME 433 course in terms of data acquisition and interpretation of results. Over the past few semesters, a numerical model used to expose students to combustion phenomena has evolved from a simple, ideal gas model to one that has included real fluid properties, friction effects, and emissions predictions. The objectives of this research were:

- 1.) To develop an activity in which students optimize spark timing based on simulation of in-cylinder heat release.
- 2.) To expose students to NO and HC formation mechanisms within the context on an actual engine
- 3.) To obtain, analyze, and compare theoretical as well as experimental brake-specific fuel consumption data for an engine platform in NIATT's Small Engines Research Facility.
- 4.) To illustrate and quantify sources of inefficiency in SI engines.
- 5.) To create a starting point for future engine modeling in NIATT research.

## **1.3 Thesis Organization**

This thesis encompasses nine chapters. Chapter 2 is the literature review, which researches classical and current engine modeling techniques and heat transfer prediction methods. Chapter 3 details the mathematical equations used in single and two-zone models, as well as emissions models. Chapter 4 explains the organization of the MATLAB program, and overviews geometric and atmospheric inputs, as well as model assumptions. Chapter 5 contains analytical validations for model outputs, as well as statistical uncertainties associated with the model. Chapter 6 focuses on the experimental setup and setup

uncertainties. Chapter 7 overviews engine testing, and the corresponding results that were obtained. Chapter 8 outlines in-class activities will use this model to predict four-stroke engine performance. Lastly, chapter 9 presents conclusions that were made based on the current model, and suggests future research surrounding this topic.

## Chapter 2. Literature Review

Engine modeling literature was reviewed to determine the most effective models used in research and in the field. This literature review looked at the most commonly used engine modeling techniques and observed some of the changes that have been made to these models to make them more accurate and applicable to today's engines. In studying combustion literature, it was found that modified versions of the polytropic and apparent heat release models were often used in modeling engine performance; thus, this literature review looked at these models in depth.

Of the changes made to the apparent heat release model, heat transfer effects were researched extensively. This literature review overviewed Woschni's and Annand's empirical heat transfer models and compared the predicted heat transfer coefficient associated with each model. Although it was found that Eichelberg's heat transfer model is still used in some research, it was considered out-of-date for the purpose of this thesis, and wasn't researched to the extent of the other models.

The viability of predicting emissions with a numerical model was also researched. Within the context of a SI, IC engine, NO and HC formation mechanisms and modeling techniques were researched and analyzed for complexity. These models ranged from basic empirical models to highly-involved, multi-zone chemical models. Based on information found in this literature review, decisions were then made on modeling techniques that were used in the numerical, MATLAB model. The subsequent sections will describe the polytropic and apparent heat release engine models, convective heat transfer models that have been used to revise the apparent heat release model, NO and HC formation mechanisms, and decisions that were made based on this literature review.

## 2.1 Heat Release Models

Although the thermodynamic background for engine modeling was established in the 1800's, zone-based models didn't become popular until 1938 when Rassweiler and Withrow constructed a single-zone model that correlated polytropic behavior with measured pressure traces [1]. The Rassweiler and Withrow model assumed that pressure and volume could be related using the polytropic relationship [2]:

$$PV^n = \text{constant} \quad (2.1)$$

where  $P$  was pressure,  $V$  was volume, and  $n$  was the polytropic index. In this model,  $n$  was assumed to be constant, blow-by was ignored, and a known pressure trace was required for functionality [1].

According to Rassweiler and Withrow, the measured change in pressure could be described additively in terms of a combustion pressure rise and a volumetric pressure rise:

$$\Delta P = \Delta P_c + \Delta P_v \quad (2.2)$$

where  $\Delta P_c$  was the change in pressure due to combustion and  $\Delta P_v$  was the pressure rise due to volumetric expansion or contraction. The combustion and volumetric pressure rises were found using polytropic assumptions and the measured pressure trace, and the ratio of accumulated to total pressure change (due to combustion) was then used to extract the burn profile [1]. Extraction of the burn profile required the assumption that the instantaneous combustion pressure rise was proportional to the instantaneous fraction of fuel burned.

In 1967, Krieger and Borman developed a model that combined the first law of thermodynamics and the ideal gas law [1][3]. Krieger and Borman's model included a net heat release term that was defined as [3]:

$$\frac{dQ_n}{d\theta} = \frac{\gamma}{\gamma-1} P \left( \frac{dV}{d\theta} \right) + \frac{1}{\gamma-1} V \left( \frac{dP}{d\theta} \right) \quad (2.3)$$

where  $dQ_n$  was the change in net heat release and  $\gamma$  was the specific heat ratio. With the specific heat ratio equal to the polytropic index, the predicted heat release was equivalent to that of Rassweiler and Withrow's model. Klein [1] showed that careful selection of the specific heat ratio significantly improved the accuracy of burn profile predictions on a specific engine platform, as can be seen in table 1.

**Table 1. A Comparison of Heat Release Models [1]**

<b>Method</b>	<b><math>\theta_{10}</math></b>	<b><math>\theta_{50}</math></b>	<b><math>\theta_{85}</math></b>	<b><math>\theta_b</math></b>
Rassweiler-Withrow	-4.5	9.8	25.2	29.8
Apparent Heat Release	-6.4	11	26.9	33.3

Although careful selection of the ratio of specific heats produced more accurate results than the basic polytropic model, many researchers have chosen to further expand the apparent heat release model to include convective heat losses, thus improving the accuracy of net heat release predictions.

## 2.2 Convective Heat Transfer Models

With 20-35 percent of the total fuel energy being absorbed by the engine's coolant [4], researchers knew that a method of predicting convective heat transfer was paramount to the accuracy of the chosen engine model. Modeling convective heat transfer from the combustion chamber gases to the cylinder walls began with Newton's law of cooling [2]:

$$\frac{dQ_{ht}}{dt} = hA(T - T_w) \quad (2.4)$$

where  $h$  was the convective heat transfer coefficient,  $A$  was the heat transfer area,  $T$  was the cylinder gas temperature, and  $T_w$  was the cylinder wall temperature.

Researchers began looking at known heat transfer correlations in an attempt to calculate the convective heat transfer coefficient in combustion chambers; much of this

research was focused on fluid flow through a pipe. In terms fluid turbulence, flow through a pipe was defined as [4][5]:

$$\mathbf{Nu} = \mathbf{aRe}^{\mathbf{b}} \quad (2.5)$$

where  $\mathbf{Nu}$  was the Nusselt number and  $\mathbf{Re}$  was the Reynolds number. Although researchers differed on the basic assumptions necessary to calculate the heat transfer coefficient, they all began with equation 2.5 and adapted it to IC engines.

In 1963 [4], Annand developed a heat transfer prediction method that's still widely used today. He began by splitting apart equation 2.4, and the Nusselt and Reynolds numbers, respectively, into constituents:

$$\frac{hx}{k} = \mathbf{a} \left( \frac{\rho \bar{S}_p x}{\mu} \right)^{\mathbf{b}} \quad (2.6)$$

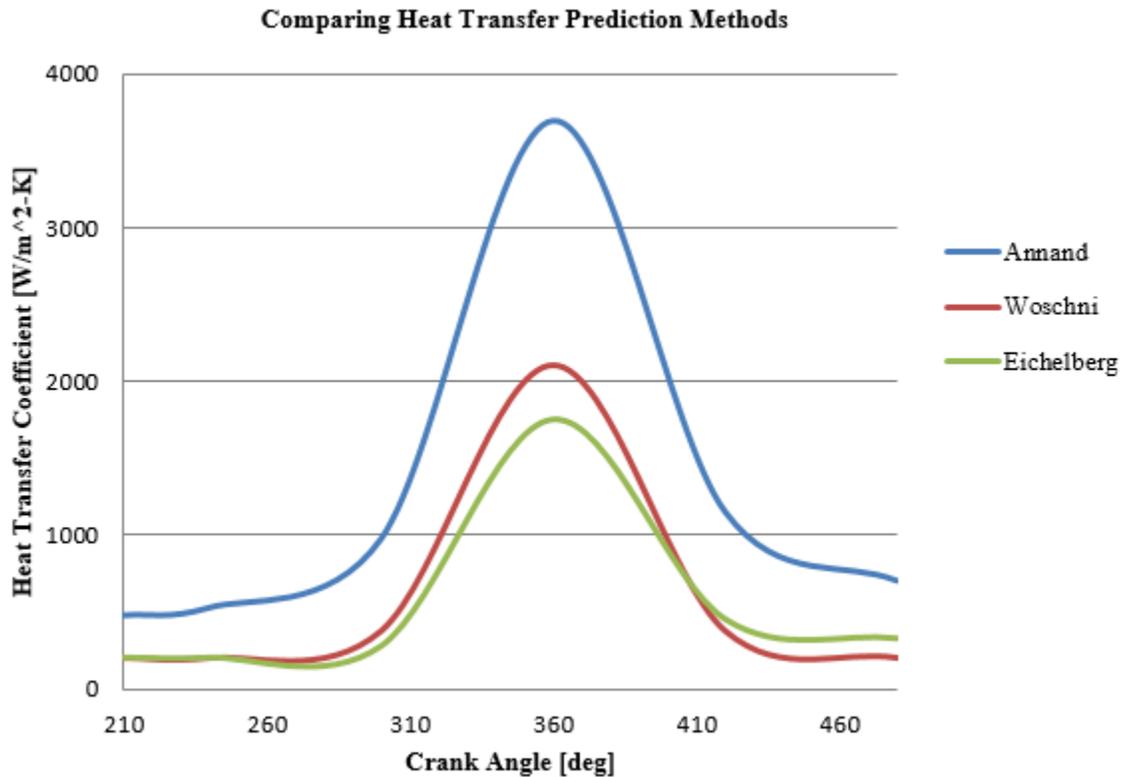
where  $x$  was the characteristic length,  $k$  was the gas thermal conductivity,  $\rho$  was the gas density,  $\bar{S}_p$  was the mean piston speed, and  $\mu$  was the gas viscosity. The characteristic length was defined as being equal to the cylinder bore, and constants  $\mathbf{a}$  and  $\mathbf{b}$  were found experimentally. Annand noticed that the value of these constants varied as a function of cylinder charge motion and engine geometry, but under normal operating conditions,  $\mathbf{a}$  fell between 0.35 and 0.8, and  $\mathbf{b}$  was approximately equal to 0.7 [3]. By assuming that the cylinder gas had properties equivalent to air at the same temperature, Annand was then able to calculate the convective heat transfer coefficient through the use of equation 2.6 [5].

In calculating the overall heat transfer, it should also be noted that Annand incorporated radiative losses. In many systems, it was found that these terms could be ignored. However, some researchers have chosen to include these terms because of increased accuracy in specific IC engine configurations [5].

Like Annand, Woschni broke apart equation 2.4 into constituents and assumed a characteristic length equivalent to the cylinder bore. Although their initial assumptions were similar, Annand and Woschni differed in methodologies of calculating the convective heat transfer coefficient. Woschni also ignored radiative heat transfer altogether.

In calculating the convective heat transfer coefficient, Woschni included a term for the average gas velocity and theorized that the average gas velocity was proportional to the mean piston speed [3]. During combustion and expansion, Woschni took density changes into account, thus adding a motored pressure term and a term representing the change in pressure due to combustion [3]. The average gas velocity was then modified through the use of constants based on the respective combustion period during the four-stroke (or two-stroke) cycle [3].

Instead of using cylinder gas properties as a function of temperature, Woschni developed correlations between the known cylinder temperature and the gas properties. Woschni assumed that the cylinder gas pressure fluctuated ideally and that the gas thermal conductivity changed relative to the in-cylinder temperature, which was then expanded to an experimentally-determined power. Woschni assumed similar behavior when calculating the in-cylinder gas viscosity. A comparison of the predicted heat transfer coefficient using Woschni's, Annand's, and Eichelberg's methods can be found in figure 1 [4].



**Figure 1. A comparison of heat transfer prediction methods [4].**

Figure 1 showed that the heat transfer coefficient varied substantially between the different heat transfer prediction methods. Per the suggestion of Blair [5], and based on the results seen in figure 1, it was expected that Annand's method would be more accurate for small displacement engines. However, since previous research at the University of Idaho hadn't been focused on this topic, it was decided that Woschni's and Annand's methods would be implemented and compared for accuracy.

### **2.3 NO Formation Mechanisms**

Assuming combustion of a nitrogen-free fuel, the formation of NO in SI engines, can be attributed to four primary mechanisms [5][6][7]:

- (1) The Zeldovich mechanism.
- (2) The prompt mechanism.
- (3) The N<sub>2</sub>O -intermediate mechanism.
- (4) The NNH mechanism.

As the air-fuel ratio and engine operating characteristics change, different mechanisms can become accordingly more or less important.

The Zeldovich, or thermal, mechanism is formed from the oxidation of atmospheric nitrogen (N<sub>2</sub>) and dominates in high temperature combustion [6][8]. This mechanism is highly temperature-dependent and varies according to the supplied oxygen (O<sub>2</sub>) concentration during combustion. Since its rate constants are slow, the Zeldovich mechanism is only relevant in circumstances such as those of the IC engine, where the burned-gas temperature is above 1800 (K). The extended Zeldovich mechanism is expressed as [4][5][6][7][8]:



It can be seen in equations 2.7-2.9 that of the two NO<sub>x</sub> species (NO and NO<sub>2</sub>), only NO is contained within the elementary reactions. This is because NO is typically the most dominant nitrogen oxide formed during SI combustion, and further oxidation of NO typically leads to NO<sub>2</sub> in the environment [4].

The prompt mechanism is formed by reactions between N<sub>2</sub> and HC radicals and is important under fuel-rich operating conditions [6][8]. This is because the prompt mechanism requires the collision of HC and N<sub>2</sub>, thus producing a nitrogen-free radical (N) and amine or

cyano compound [7]; these compounds can then further react to form NO. The prompt mechanism is defined as [7][8]:



In examining combustion literature, it was found that the prompt mechanism is typically ignored in modeling IC engine emissions. It was found that the prompt mechanism is only significant in scenarios such as coal diffusion [8].

The N<sub>2</sub>O- intermediate mechanism is important under fuel-lean, low temperature operating conditions, and is relatively insignificant in spark-ignition engines [6][7]. The N<sub>2</sub>O- intermediate mechanism involves the collision between hydrogen (H) and oxygen (O) free-radicals and N<sub>2</sub>O, thus forming NO. The N<sub>2</sub>O-intermediate mechanism is defined as [6]:



Lastly, the NNH mechanism is a reaction between NNH and oxygen (O) free-radicals that produces NO [9][10]. This is a fairly new concept and is probably the least understood of the four mechanisms. The NNH mechanism is defined as [6]:



It's hypothesized that this mechanism is important at low pressures and temperatures [10], and it will be ignored for the purpose of this model.

## 2.4 NO Modeling Techniques

In researching NO formation literature, it was found that the kinetics of SI combustion can be described by 11 chemical reactions [4][11]:





Tinaut and Melgar [11] determined the exhaust composition through an equilibrium model and the rate of reactions 2.17 – 2.27 through chemical kinetics. They found that equations 2.17 – 2.24 were governed by carbon monoxide (CO) kinetics, while equations 2.25 – 2.27 were governed by the Zeldovich mechanism. The forward and reverse reaction rates were then calculated using the Arrhenius equation and constants from a look-up table.

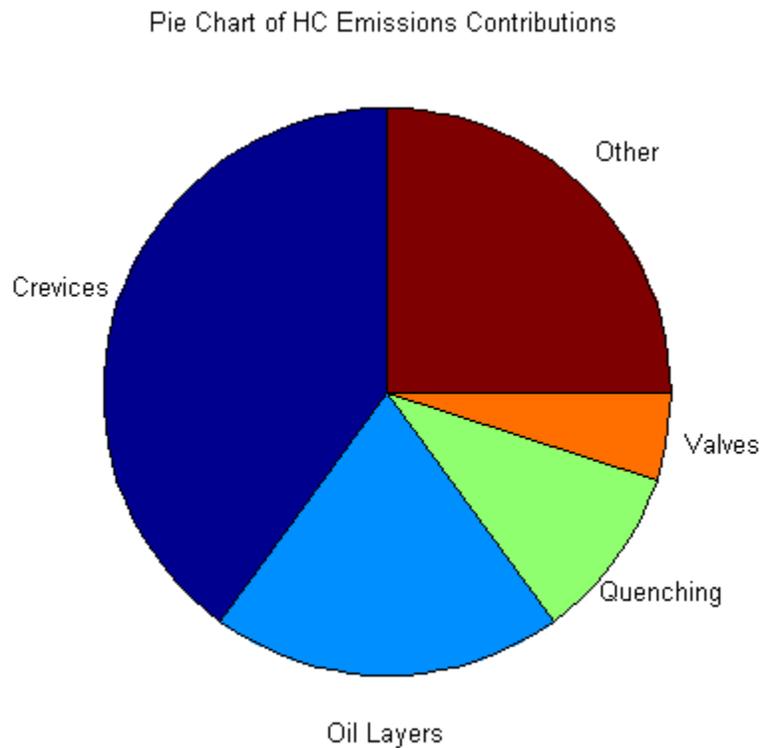
Many other researchers chose to simplify the Zeldovich mechanism using assumptions based on experimental results, which prevented the hassle of having to look up values for 11 reactions. Heywood, Turns, Blair [3][5][6] and others suggested using the assumption that the concentration of N is the lowest of the Zeldovich species. Therefore, the change in N concentration over time was assumed to be zero. They then assumed a decoupling of NO formation processes because NO dominated in post-gas processes. This reduced NO formation to a relatively simple differential equation that required O and N equilibrium concentrations, and a forward reaction rate. With a given burned temperature profile and known residence time, this was a relatively effortless calculation.

## 2.5 HC Formation Mechanisms

The formation of HC emissions is highly dynamic and influenced by many factors. Of the principal HC emissions formation mechanisms, most researchers have focused on [6][12]:

- (1) Flame Quenching due to cylinder crevices.
- (2) Oil layer absorption and desorption.
- (3) Liquid fuel pooling.
- (5) Exhaust Valve Leakage.
- (6) Accumulation in engine deposits.

During an engine cycle, total HC emissions can be broken up percentage-wise as follows [13]: 40 percent due to crevice losses, 20 percent due to oil layers and accumulated engine deposits, 10 percent due to liquid fuel and flame quenching on cylinder walls, and less than 5 percent due to exhaust valve leakage. Figure 2 shows the contribution of each emissions formation mechanisms under normal engine operation.



**Figure 2. Pie chart of HC emissions contributions.**

HC emissions form in crevices through the mechanism of flame quenching, where, in order for flame propagation to occur, a minimum amount of space is required for the flame to pass through. If the flame proceeds through a smaller space, the flame is extinguished. Crevices can be found in numerous areas throughout an internal combustion engine.

However, the main crevice volumes can be designated as [1][14]:

- (1) The volume between the piston rings and liner.
- (2) The volume formed by the head gasket.
- (3) The volume between spark plug threads.
- (4) The space around the pressure transducer (if applicable).

Figure 3 shows the different crevice volumes and their relationship to the combustion chamber.

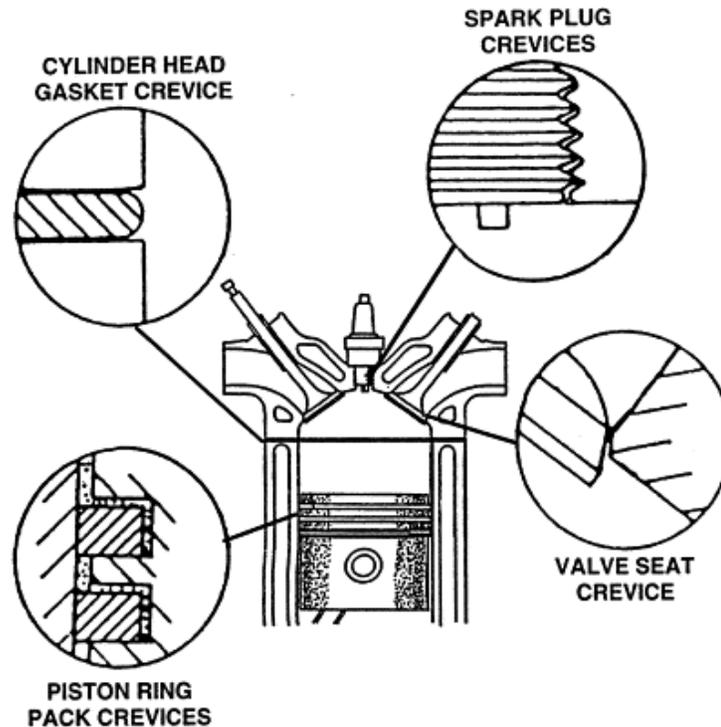


Figure 3. Sources of HC crevice losses [14].

Assuming a fully-warmed engine, the top-land piston ring crevice and piston ring pack serve as the most important HC emissions formation locations [11][12][13][14], where the top-land is the portion of the piston above the upper piston ring. The volume between the piston and cylinder wall above the top piston ring, and any other volume in which quenching may occur on top of the piston comprise the top-land crevice volume.

After crevice formation, oil layers and engine deposits contribute second-most towards engine-out HC emissions. Although no models were found with regards to engine deposits, a physical model representing oil-layer formation was found. Heywood [3][12] described HC emissions from oil layers as an absorption and desorption process. He theorized that fuel vapor absorbed into the oil film on the cylinder walls during the intake and compression cycles, and as the film became saturated, the fuel vapor continued to absorb because of its increasing vapor pressure. With a rapidly-decreasing concentration of fuel

vapor during and after combustion, the imbalance in concentrations then caused desorption from the oil layer.

During the startup and warm-up periods, fuel quenching on cylinder walls, exhaust valve leakage, and liquid fuel formation become increasingly important [11]. During fully-warmed combustion, fuel quenching on cylinder walls, liquid fuel formation, and exhaust valve leakage contribute only small amounts towards engine-out emissions because of secondary combustion and the fact that these mechanisms occur during the intake, or beginning, portion of the cycle. Therefore, these mechanisms were neglected for the purpose of this model.

## **2.6 HC Modeling Techniques**

The most basic HC crevice models have used empirical equations, along with known crevice volumes, to predict the percentage of unburned fuel that bypassed the combustion process. Hamrin and Heywood [12] assumed a crevice temperature equivalent to the coolant temperature and were able to model the mass of HC escaping combustion as a function of peak cylinder pressure, residual gas fraction, exhaust gas recirculation, and engine geometry. Other researchers have chosen to model HC emissions as a function of combustion chamber zones. Tinaut and Melgar [11] treated each crevice volume as a burned and unburned zone. They then modeled HC emissions as a function of mass flow rates and conservation of energy between these zones. With multiple zones and additional equations, this method was far beyond the scope of an entry level HC emissions prediction model.

In modeling oil absorption and desorption and liquid fuel effects, researchers have used numerous empirical equations to predict the effective emissions contributions. Hamrin and Heywood [12] recognized that liquid fuel effects were highly dependent on injector

design and the temperature of surroundings, and therefore chose to neglect these effects in predicting emissions. They then assumed a characteristic diffusion time that was suitably less than the characteristic engine speed [3], and used Henry's law and the fluid properties of engine oil to estimate the fraction of HC absorption and desorption.

After predicting a percentage of fuel that bypassed the combustion process, most researchers then predicted a percentage of post-flame oxidation. Hamrin and Heywood [12] chose to use empirical equations based on experimental data to predict the fraction of exhaust oxidation, where their equations were modified based on a calculated spark advance and change in engine speed due to EGR. Upon combining several curve-fitted equations and incorporating the equivalent spark advances and changes in engine speeds, they were able to predict the percentage of HC post-combustion oxidation. Other post-flame oxidation prediction methods have considered the thermal-fluid effects in the exhaust. Tinaut and Melgar [11] modeled the exhaust gas temperature as a function of wall losses through the exhaust pipe. They treated the exhaust gas as a bulk fluid and calculated the thermal conductivity of this gas, and thus, found the heat flux between the bulk gas and the exhaust pipe, relative to geometric and fluid properties. The residence time of each zone was then modeled relative to the exhaust pipe length and each zone velocity; this led to an equation that was fit to experimental data using a constant multiplier.

## **2.7 Model Selection**

The literature review provided valuable insight towards producing a numerical model. It was decided that the apparent heat release model would be used in the numerical model, and both Woschni's and Annand's heat transfer models would be included and compared for accuracy. It was also decided that the simpler emissions prediction models would be used,

because other models were far too advanced for an entry level numerical simulation. Table 2 shows the decisions that were made and the reasoning behind each decision, with respect to the literature review.

Table 2. Engine models adopted in this work

<u>Model</u>	<u>Included</u>	<u>Reasoning</u>
Polytropic Heat Release Model	No	The polytropic model would be useful to modify the Weibe function based on pressure measurements. Pressure measurements weren't available for this research.
Apparent Heat Release Model	Yes	The apparent heat release model with several modifications will be used.
Woschni's Heat Transfer Model	Yes	It was unknown whether Woschni's or Annand's method would produce more accurate results.
Annand's Heat Transfer Model	Yes	It was unknown whether Woschni's or Annand's method would produce more accurate results.
The Zeldovich Mechanism	Yes	This is the simplest method for predicting NO <sub>x</sub> emissions. The original Zeldovich mechanism will be used; not the expanded mechanism.
The Prompt NO Mechanism	No	This is only applicable in highly fuel rich scenarios. This is typically ignored in IC engine applications.
The NNH Mechanism	No	It is theorized that this is only applicable at low pressures and temperatures. This is not applicable to IC engines.
Hamrin and Heywood's HC Prediction Model	Yes	The empirical relationships associated with this method are easy to apply to a numerical code.
Tinaut and Melgar's Constituent HC Prediction Model	No	This model is much more complicated than Hamrin and Heywood's method. This model requires complex fluid dynamics in the exhaust.
Crevice Formation	Yes	It is relatively easy to measure top-land crevice dimensions. This is a major contributor towards HC emissions.
Oil Absorption and Desorption	Yes	This is relatively simple in terms of concept and modeling. This contributes significantly towards overall HC emissions.
Liquid Fuel Formation	No	Liquid fuel formation varies dynamically during startup periods. This is difficult to model and not significant during normal operation.
Valve Leakage	No	This takes valve geometries into account and is difficult to predict.

## Chapter 3. Governing Equations

After decisions were made with regard to the modeling techniques being used in this research, a single-zone model was developed. The single-zone model formed the basis by which the two-zone model was derived and was used to predict the bulk cylinder gas temperature and pressure, as well as power characteristics of the engine. Heat transfer prediction methods, variable specific heat ratios, and friction models were then included to improve the accuracy of the single-zone model.

Upon completing the single-zone model, the two-zone model was then developed. The two-zone model was used for emissions predictions because it calculated the burned gas temperature, which was elevated in comparison to the bulk, or single-zone, temperature. This led to HC and NO emissions models which, although basic, provided excellent insight towards the addition of emissions associated with engine operating points. The following sections will describe the development of the single and two-zone models, as well as the HC and NO emissions models.

### 3.1 Single-Zone Engine Model

The simplest approach in engine modeling is to treat the cylinder contents as a solitary fluid or zone [1]. The single-zone model views the burned and unburned gases, residual gases, and unburned HC within the cylinder as an ideal gas with uniform pressure. Single-zone models typically use the Weibe function to represent the chemical energy release as a function of crank angle [3]. The Weibe function has a characteristic “S-shape” and is defined as follows:

$$X_{b(\theta)} = 1 - \exp \left[ -a \left( \frac{\theta(i) - \theta_o}{\theta_b} \right)^{k+1} \right] \quad (3.1)$$

where  $\mathbf{a}$  and  $\mathbf{k}$  are adjustable constants (5 and 2 are commonly used values),  $\theta(i)$  is the instantaneous crank angle,  $\theta_o$  is the spark-advance, and  $\theta_b$  is the burn duration. The burn profile is engine-specific, and the constants  $\mathbf{a}$  and  $\mathbf{k}$  can be adjusted to tune a profile to a specific application.

The ideal gas law forms the basis for the single-zone engine model, and is defined as:

$$\mathbf{PV} = \mathbf{mRT} \quad (3.2)$$

where  $\mathbf{P}$  is the pressure of an ideal gas,  $\mathbf{V}$  is the volume of the gas,  $\mathbf{m}$  is the mass of the gas,  $\mathbf{R}$  is the universal gas constant, and  $\mathbf{T}$  is the mean gas temperature. The cylinder gas volume,  $\mathbf{V}$ , in a reciprocating, IC engine can be related to the engine geometry as a function of crank angle [3]:

$$\mathbf{V}(\theta) = \mathbf{V}_c + \frac{\pi \mathbf{B}^2}{4} (\mathbf{l}_r + \mathbf{a} - \mathbf{s}) \quad (3.3)$$

where  $\mathbf{B}$  is the cylinder bore,  $\mathbf{l}_r$  is the connecting rod length,  $\mathbf{a}$  is the crank radius,  $\mathbf{s}$  is the instantaneous distance between the crank axis and the piston pin axis, and  $\mathbf{V}_c$  is the clearance volume. The clearance volume is expressed as:

$$\mathbf{V}_c = \left( \frac{\mathbf{V}_d}{\mathbf{C}_r - 1} \right) \quad (3.4)$$

where  $\mathbf{V}_d$  is the displaced cylinder volume, and  $\mathbf{C}_r$  is the compression ratio. Figure 4 shows a diagram of these figures and their relationship to engine geometry.

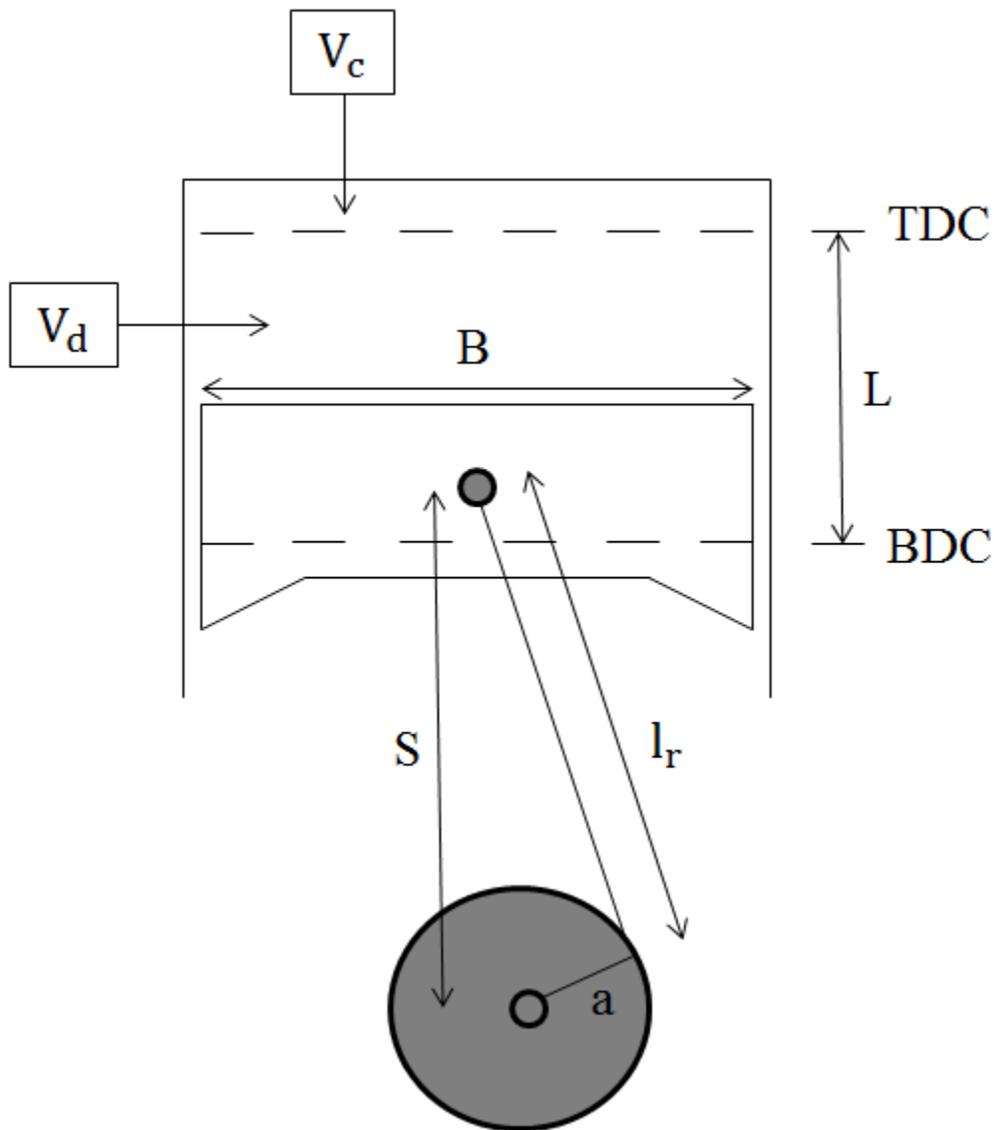


Figure 4. Engine geometry and variables used in modeling.

In differentiating equation 3.2 with respect to  $d\theta$ , equation 3.5 is obtained:

$$\frac{dP}{d\theta} V + P \frac{dV}{d\theta} = mR \frac{dT}{d\theta} \quad (3.5)$$

and upon rearranging equation 3.5 and solving for  $dP$ , expression 3.6 is obtained:

$$\frac{dP}{d\theta} = \left(-\frac{P}{V}\right) \left(\frac{dV}{d\theta}\right) + \left(\frac{P}{T}\right) \left(\frac{dT}{d\theta}\right) \quad (3.6)$$

where  $P$ ,  $V$ , and  $T$  are instantaneous values modeled relative to the engine's crank angle. The same process can be applied to the first law of thermodynamics, which is expressed as:

$$\Delta U = Q - W \quad (3.7)$$

where  $Q$  is the total energy transferred into the system,  $W$  is the work transferred out of the system, and  $\Delta U$  is the change in internal energy within the system. In differentiating equation 3.7 with respect to  $d\theta$ , expression 3.8 can be obtained:

$$\frac{dU}{d\theta} = \frac{dQ}{d\theta} - \frac{dW}{d\theta} = mC_v \left( \frac{dT}{d\theta} \right) \quad (3.8)$$

where  $C_v$  is the specific heat of the combustion chamber gas. Upon dividing the specific heat by the universal gas constant, equation 3.9 can be obtained:

$$\frac{C_v}{R} = \frac{C_p}{C_p - C_v} = \frac{1}{\gamma - 1} \quad (3.9)$$

where  $\gamma$  is the specific heat ratio.

Equations 3.10 and 3.11 can be used to describe the formation of work and the net heat input [15]:

$$\frac{dW}{d\theta} = P \left( \frac{dV}{d\theta} \right) \quad (3.10)$$

$$\frac{dQ}{d\theta} = \eta_c \text{LHV} \left( \frac{dX_b}{d\theta} \right) - \frac{dQ_w}{d\theta} \quad (3.11)$$

where  $\eta_c$  is the combustion efficiency,  $\text{LHV}$  is the lower heating value of the fuel,  $\left( \frac{dX_b}{d\theta} \right)$  is the instantaneous change in mass-fraction burned, and  $\frac{dQ_w}{d\theta}$  is the instantaneous change in heat loss to the cylinder walls. Upon substituting equations 3.9 and 3.10 into equation 3.8, the instantaneous change in temperature is defined as:

$$\frac{dT}{d\theta} = T(\gamma - 1) \left[ \left( \frac{1}{P V} \right) \left( \frac{dQ}{d\theta} \right) - \left( \frac{1}{V} \right) \left( \frac{dV}{d\theta} \right) \right] \quad (3.12)$$

where the gross change in input heat  $\left( \frac{dQ}{d\theta} \right)$  can be found in equation 3.11. With the change in temperature as a function of crank angle designated, the heat input from the fuel can be used to find the change in pressure as a function of crank angle [15]:

$$Q_{in} = \eta_c \text{LHV} \left( \frac{1}{A_{F_{ac}}} \right) \left( \frac{P}{RT} \right) V_d \quad (3.13)$$

where  $AF_{ac}$  is the actual air-fuel ratio, and a method for calculating this can be found in section 3.1.5. Lastly, the change in pressure is defined as:

$$\frac{dP}{d\theta} = \left(\frac{-\gamma P}{V}\right) \left(\frac{dV}{d\theta}\right) + \left(\frac{\gamma-1}{V}\right) Q_{in} \frac{dX_b}{d\theta} + (\gamma - 1) \left(\frac{1}{V}\right) \left(\frac{dQ_w}{d\theta}\right) \quad (3.14)$$

where all of the variables have been defined, thus allowing for the development of a single-zone numerical model.

### 3.2 Woschni's Heat Transfer Model

Woschni's method is a set of empirical equations that predicts the heat transfer coefficient between in-cylinder gasses and walls. The convective losses between in-cylinder gasses and walls can be predicted using Newton's law of cooling [4]:

$$\delta Q_w = h_w A_w (T - T_w) dt \quad (3.15)$$

where  $Q_w$  is the convective heat loss,  $h_w$  is the instantaneous heat transfer coefficient,  $A_w$  is the instantaneous heat transfer area,  $T$  is the instantaneous bulk gas temperature, and  $T_w$  is the cylinder wall temperature. This model assumed a constant cylinder wall temperature of 350[K], per the suggestion of Stone [4].

The convective heat loss equation is time-dependent and must be converted to angular dependency in order to mesh with the single-zone model. A straightforward conversion between unit time and crank angle is[5]:

$$dt = \frac{d\theta}{360 \text{ RPM}} = \frac{d\theta}{360 N} \quad (3.16)$$

where  $RPM$  is the speed of the engine  $\left(\frac{\text{rev}}{\text{min}}\right)$  and  $N$  is the converted speed of the engine  $\left(\frac{\text{rev}}{\text{s}}\right)$ .

By substituting  $dt$  into equation 3.20, the convective heat loss can be modeled as a function of crank angle.

With the heat loss modeled angularly, the convective heat transfer coefficient is now defined as [3]:

$$h_w \left( \frac{W}{m^2 \cdot K} \right) = 3.26 B(m)^{-0.2} P(kPa)^{0.8} T(K)^{-0.55} w \left( \frac{m}{s} \right)^{0.8} \quad (3.17)$$

where  $B$  is the cylinder bore,  $P$  is the instantaneous cylinder pressure, and  $w$  is the average gas velocity.

According to Woschni, the average gas velocity fluctuates as a function of several engine parameters and a reference state, such as the closing of the intake valve [3]. Woschni defined the average gas velocity as:

$$w = \left[ C_1 \bar{S}_p + \frac{C_2 (V_d T_r)}{P_r V_r} (P - P_m) \right] \quad (3.18)$$

where  $C_1$  and  $C_2$  were constants that varied depending on the combustion period,  $\bar{S}_p$  was the mean piston velocity,  $T_r$  was the reference temperature,  $P_r$  was the reference pressure,  $V_r$  was the reference volume, and  $P_m$  was the motored cylinder pressure. The constants  $C_1$  and  $C_2$  were defined as:

$$C_1 = 2.28 \text{ and } C_2 = 0 \quad (3.19)$$

$$C_1 = 2.28 \text{ and } C_2 = 3.24 \times 10^{-3} \quad (3.20)$$

during compression and combustion, respectively. Since the model only calculated engine performance during these periods, details surrounding the gas exchange phases, such as intake and exhaust, were ignored.

In compression and expansion processes, Watson and Janota [4] suggested modeling the motored cylinder pressure as a polytropic process:

$$P_m = P_r \left( \frac{V_r}{V} \right)^n \quad (3.21)$$

where  $V$  was the instantaneous cylinder volume and  $n$  was the polytropic constant. With all of these variables previously expressed, the convective heat transfer coefficient and corresponding heat loss could then be calculated.

### 3.3 Annand's Heat Transfer Model

Annand's and Woschni's heat transfer models differed in the fact that Annand's approach separated the convective and radiative terms [5]. Annand's method solved for the heat transfer coefficient by assuming pipe-like fluid dynamics, and using the in-cylinder density, and Reynolds and Nusselt numbers as functions of time [5].

Using Annand's method, Newton's law of cooling can be broken into convective and radiative terms as follows:

$$\delta Q_w = (h_c + h_r)A_w(T - T_w)dt \quad (3.22)$$

where  $h_c$  is the convective heat transfer coefficient and  $h_r$  is the radiative heat transfer coefficient. The convective heat transfer coefficient can be extracted from the relationship between the Nusselt number and fluid properties[5]:

$$h_c = \frac{k_{gas}Nu}{B} \quad (3.23)$$

where  $k_{gas}$  is the gas thermal conductivity,  $Nu$  is the Nusselt number, and  $B$  is the cylinder bore.

With an iterative solver, the thermal conductivity of the cylinder gas can be modeled using a polynomial curve-fitting of experimental data. Blair [5] suggests using the curve-fitted equation:

$$k_{gas} \left( \frac{W}{m \cdot K} \right) = 6.1944 \times 10^{-3} + 7.3814 \times 10^{-5}T(K) - 1.2491 \times 10^{-8}T(K)^2 \quad (3.24)$$

where  $T$  is the instantaneous cylinder temperature [K], and the thermal conductivity is output in units of  $\left[ \frac{W}{m \cdot K} \right]$ .

The Nusselt number can be described relative to the Reynolds number and the type of engine:

$$Nu = aRe^{0.7} \quad (3.25)$$

where  $a$  is a constant having a value of 0.26 for a two-stroke engine and 0.49 for a four-stroke engine [5], and  $Re$  is the instantaneous Reynolds number. The Reynolds number is expressed as:

$$Re = \frac{\rho_{gas} \bar{S}_p B}{\mu_{gas}} \quad (3.26)$$

where  $\rho_{gas}$  is the instantaneous cylinder gas density,  $\bar{S}_p$  is the mean piston velocity, and  $\mu_{gas}$  is the instantaneous gas viscosity. Since the model assumes ideal gas behavior, the cylinder gas density can be found by rearranging the ideal gas law:

$$\rho_{gas} = \frac{P}{R_{air} T} \quad (3.27)$$

where  $R_{air}$  is the fluid-specific gas constant, and an assumed value of  $287 \left[ \frac{J}{kg \cdot K} \right]$  was used for this variable.

As with the thermal conductivity, the cylinder gas viscosity was modeled using empirical equations. According to Blair [5], the cylinder gas viscosity can be expressed as:

$$\mu_{gas} \left( \frac{kg}{m \cdot s} \right) = 7.457 \times 10^{-6} + 4.1547 \times 10^{-8} T(K) - 7.4793 \times 10^{-12} T(K)^2 \quad (3.28)$$

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

Although the radiative heat transfer coefficient is small [5], it was decided that radiation should be included in considering overall heat losses in the model. The radiative heat transfer coefficient is defined as [5]:

$$h_r \left( \frac{W}{m^2 \cdot K} \right) = 4.25 \times 10^{-9} \left( \frac{T^4 - T_w^4}{T - T_w} \right) \quad (3.29)$$

where the instantaneous cylinder temperature and wall temperature must be provided in units of [K]. With known pressure and temperature traces from the single-zone calculations, Annand's method could then be used to calculate heat losses.

### 3.4 Temperature-Dependent Specific Heat Ratios ( $\gamma$ )

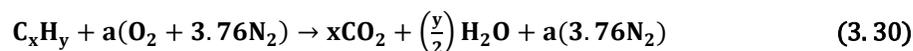
Due to the large temperature gradients in an IC engine, instantaneous specific heat ratios were desired. It was found that numerous specific heat ratio models exist, and the accuracy of these models depended highly on the complexity of the corresponding computer code. It was decided that a curve-fitted, polynomial method would be used to calculate the specific heat ratio as a function of combustion temperature.

The polynomial method was developed by Krieger and Borman for combustion processes such as those involving hydrocarbon fuels [1]. Krieger and Borman's method calculates changes in internal energy through the use of correction factors corresponding to changes in temperature compared to a reference temperature. Through a series of derivations, the specific heat ratio as a function of temperature can be obtained with Krieger and Borman's method, and this derivation can be found in appendix A.

### 3.5 Determination of Air-Fuel Ratio

It was known that the excess air coefficient ( $\lambda$ ) would be available during testing, so the air-fuel ratio was calculated using the  $\lambda$  reading and the balanced, stoichiometric reaction between iso-octane and air. Before defining  $\lambda$  and the air-fuel ratio, a succinct explanation of the chemical reactions involved in SI combustion will be included.

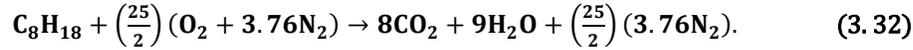
The general, balanced stoichiometric reaction for a HC fuel is defined as [6]:



$$a = x + \frac{y}{4} \quad (3.31)$$

where  $x$  is the number of carbon (C) atoms in the fuel,  $y$  is the number of hydrogen (H) atoms in the fuel, and  $a$  is a constant necessary to balance the equation. By assuming iso-octane

and substituting the equivalent C and H atoms into equation 3.30, the balanced, stoichiometric reaction is expressed as:



Based on the balanced HC reaction, the stoichiometric, gravimetric air-fuel ratio is found to be [6]:

$$\text{AF}_{\text{stoich}} = \frac{4.76(a)}{1} \frac{\text{MW}_{\text{air}}}{\text{MW}_{\text{fuel}}} \quad (3.33)$$

where  $\text{MW}_{\text{air}}$  is the molecular weight of air and  $\text{MW}_{\text{fuel}}$  is the molecular weight of fuel. Upon substituting the molecular weights of air and iso-octane into equation 3.33, the stoichiometric air-fuel ratio is expressed as:

$$\text{AF}_{\text{stoich}} = \frac{4.76\left(\frac{25}{2}\right)}{1} \frac{28.97}{114.2} = 15.09 \quad (3.34)$$

$$\therefore \text{AF}_{\text{ac}} = \lambda \text{AF}_{\text{stoich}} \quad (3.35)$$

which fluctuates based on the  $\lambda$  reading during a specific engine test.

### 3.6 Engine Friction Model

Friction losses vary significantly amongst engines and can be introduced through bearing components, pistons, and engine-driven accessories [3]. Friction losses can be very difficult to model without known engine data and can fluctuate based on engine coolant and oil temperatures, ambient conditions, and engine throttle settings [3]. For this model, rolling bearings were assumed, and a process suggested by Blair [5] was used to estimate friction mean effective pressure (FMEP) losses. This model was implemented so that this model could be used on a variety of theoretical or actual engines to predict performance without limiting the model to a specific application.

Various researchers such as Blair and Heywood [5][3] have used general linear equations to predict FMEP losses as a function of RPM. According to Blair [5], the general linear FMEP loss equation is defined as:

$$\mathbf{fmep} = \mathbf{a} + \mathbf{b(L)(RPM)} \quad (3.36)$$

where **a** and **b** are constants that vary depending on the engine type, **L** is the stroke [m] of the engine, and **RPM** is the engine speed  $\left[\frac{\text{rev}}{\text{min}}\right]$ . For an SI motorcycle engine with rolling bearings, Blair [5] defined the FMEP loss equation as:

$$\mathbf{FMEP(Pa)} = \mathbf{250(L(m))(RPM)}. \quad (3.37)$$

Although these equations provide a starting point for numerical simulations, a curve-fitted profile based on specific engine test data should be used in optimizing model performance.

### 3.7 Residual Gas Fraction Model

The exhaust blow-down process is nearly impossible to model without accounting for intake and exhaust gas dynamics. However, the process can be simulated accurately at low-to-medium engine speeds, where the pressure gradient across the exhaust port isn't excessively large [16]. A method developed by Fox et al. [16] was used to calculate the residual gas fraction due to exhaust back flow during the period of valve overlap.

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

$$\mathbf{T(i)} = \mathbf{T_{EVO} \left(\frac{P_{BDC}}{P_{EVO}}\right)^{\frac{\gamma-1}{\gamma}}} \quad (3.38)$$

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

The residual gas fraction is the mass ratio of residual gas to fuel and air and is defined as [16]:

$$f = \frac{1}{C_r} \left( \frac{P_e}{P_i} \right)^{\frac{1}{\gamma}} \left( \frac{1}{\lambda} \right) \quad (3.39)$$

where  $C_r$  is the compression ratio,  $P_e$  is the exhaust gas pressure, and  $P_i$  is the inlet gas pressure. The residual gas fraction can be used to create a corrected or modified intake temperature:

$$T_{\text{corr}} = f(T(360)) + (1 - f)T_{\text{BDC}} \quad (3.40)$$

where  $T(360)$  is the ending cycle temperature. The corrected temperature can then be returned to the beginning of the model, thus simulating EGR.

### 3.8 Two-Zone Engine Model

Two-zone models are closely related to the equations that were derived in the single-zone model. The bulk-system pressure, mass-fraction burned, and bulk-system volume can be described using equations 3.14, 3.1, and 3.3. However, the two-zone model recognizes a burned and unburned zone, thus predicting heat transfer and emissions more accurately. Figure 5 shows the burned and unburned gas regions and their relationship to the combustion chamber.

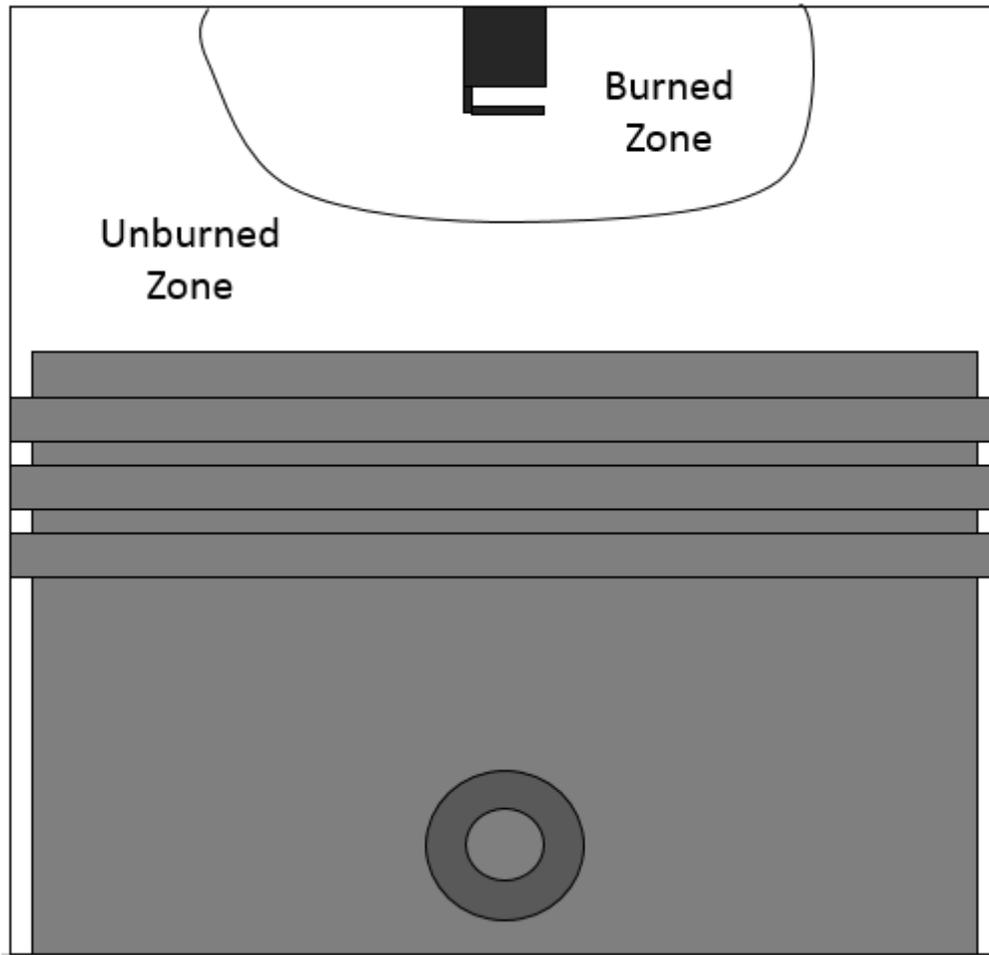


Figure 5. Two-zone diagram.

The construction of a two-zone engine model begins with the modification of the Weibe function to include unburned and burned regions. In order to determine the unburned mass at BDC, the following three relationships are used [5]:

$$\mathbf{m}_a = \rho_a V_d \quad (3.41)$$

$$\mathbf{m}_f = \frac{\mathbf{m}_a}{AF_{ac}} \quad (3.42)$$

$$\mathbf{m}_c = \mathbf{m}_a + \mathbf{m}_f \quad (3.43)$$

where  $\mathbf{m}_a$  is the mass of air contained within the cylinder,  $\rho_a$  is the density of air,  $\mathbf{m}_f$  is the mass of fuel, and  $\mathbf{m}_c$  is the total mass contained within the cylinder.

With a known cylinder mass, the Weibe function can be used to bisect the combustion chamber gases into unburned and burned masses. The burned and unburned masses can be defined in a discretized form as [5]:

$$\mathbf{m}_b(i) = \mathbf{m}_b(i-1) + \frac{dX_b}{d\theta}(i)\mathbf{m}_c \quad (3.44)$$

$$\mathbf{m}_u(i) = \mathbf{m}_u(i-1) - \frac{dX_b}{d\theta}(i)\mathbf{m}_c \quad (3.45)$$

where  $\frac{dX_b}{d\theta}$  is the instantaneous change in mass-fraction burned, and the initial quantity of burned mass is zero.

With calculated unburned and burned masses, the corresponding volumes can be obtained. Blair [5] suggests using the polytropic relations and a known pressure-trace to calculate the unburned and burned volumes. The unburned volume is defined as:

$$V_u(i) = \left( \frac{m_u(i)V_u(i-1)}{m_u(i-1)} \right) \left( \frac{P(i)}{P(i-1)} \right)^{-\frac{1}{\gamma_u(i)}} \quad (3.46)$$

where  $\gamma_u(i)$  is the instantaneous specific heat ratio of the unburned region, which can be calculated using appendix A. With the unburned volume defined, the burned volume is determined by using the relationship [5][15]:

$$V(i) = V_b(i) + V_u(i). \quad (3.47)$$

It was previously stated that two-zone models bisect the single-zone model. In order for the two-zone model to work, the ideal gas assumption must continue to each constituent zone where the burned and unburned temperatures are defined as:

$$T_b(i) = \frac{P(i)V_b(i)}{m_b(i)R(i)} \quad (3.48)$$

$$T_u(i) = \frac{P(i)V_u(i)}{m_u(i)R(i)} \quad (3.49)$$

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

After calculating the two-zone geometric and fluid properties, the instantaneous heat transfer area has to be broken into unburned and burned regions. According to Rakopoulos and Michos [17], the unburned and burned areas are defined as:

$$A_u(i) = A(i) \left( 1 - (X_b(i))^{\frac{1}{2}} \right) \quad (3.50)$$

$$A_b(i) = A(i) \left( \frac{X_b(i)}{(X_b(i))^{\frac{1}{2}}} \right) \quad (3.51)$$

where  $X_b(i)$  is the instantaneous mass-fraction burned and  $A(i)$  is the overall, instantaneous heat transfer area. The overall, instantaneous heat transfer area is expressed as:

$$A(i) = A_{ch} + \frac{\pi B^2}{2} + \pi B(l + a - s(i)) \quad (3.52)$$

where  $A_{ch}$  is the surface area of the cylinder head [ $m^2$ ],  $B$  is the cylinder bore [ $m$ ],  $l$  is the connecting rod length [ $m$ ],  $a$  is the crank radius [ $m$ ], and  $s(i)$  is the instantaneous distance between the crank axis and the piston pin axis [ $m$ ]. For the purpose of this model, the internal surface area of the cylinder head was assumed to be equivalent to twice that of cross-sectional area of the piston. The crank radius is equivalent to one-half of the stroke length, while the instantaneous distance between the crank axis and pin axis is defined as:

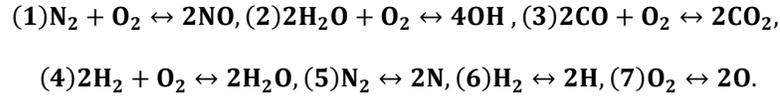
$$s(i) = a(\cos \theta) + (l^2 - a^2 \sin^2 \theta)^{\frac{1}{2}}. \quad (3.53)$$

With updated heat transfer areas, Woschni's and Annands methods can then be refined, thus predicting overall heat transfer more accurately. The burned-zone temperature can then be used for the emissions predictions described in the subsequent sections.

### 3.9 NO Emissions Prediction

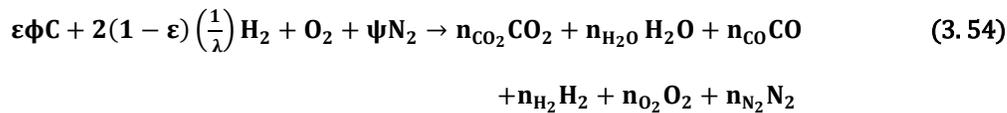
In predicting emissions, the chemical constituents and interactions must first be recognized. According to Rakopoulos [18], the chemical constituents and compounds involved in SI combustion are  $CO_2$ ,  $CO$ ,  $H_2O$ ,  $H_2$ ,  $C_8H_{18}$ ,  $NO$ ,  $OH$ ,  $N$ ,  $H$ , and  $O$ . The

equilibrium concentration of each constituent and compound can be found with atom balances and the following equilibrium reactions:



These reactions are the basis for NO emissions predictions and will be used extensively later in this section.

Atom balances and the operating parameters of an engine can be used to calculate the molar concentrations of many combustion products. The general HC combustion equation is expressed as [3]:



where  $n_i$  is the molar concentration of each respective species per mole of  $\text{O}_2$  reactant,  $\psi$  is the molar nitrogen-to-oxygen ratio of air (3.773), and  $\varepsilon$  is a constant, which defined as:

$$\varepsilon = \frac{4}{4+y} \quad (3.55)$$

where  $y$  is the molar hydrogen-to-carbon ratio of the fuel.

Solving the general HC combustion equation requires a known  $\lambda$  reading and the following three assumptions:

- 1.) CO and  $\text{H}_2$  can be neglected under fuel-lean ( $\lambda \geq 1$ ) operating conditions;
- 2.)  $\text{O}_2$  can be neglected under fuel-rich ( $\lambda \leq 1$ ) operating conditions;
- 3.) The water-gas shift reaction is assumed to be in equilibrium.

Under fuel-lean operating conditions, the general HC combustion equation can be solved using the previous assumptions and equation 3.54. However, under fuel-rich operating conditions, the water-gas shift (WGS) reaction must be used to predict the molar concentrations of  $\text{CO}_2$ ,  $\text{H}_2$ , CO, and  $\text{H}_2\text{O}$  and is defined as [3][6]:



Using the number of constituent moles and rearranging the reaction, the WGS equilibrium constant is defined as:

$$\mathbf{K}_{\text{wgs}} = \frac{n_{\text{H}_2\text{O}}n_{\text{CO}}}{n_{\text{CO}_2}n_{\text{H}_2}} \quad (3.57)$$

which has been curve-fitted to the JANAF [19] databases to obtain the following empirical equation:

$$\mathbf{K}_{\text{wgs}}(\mathbf{T}) = \exp\left(2.743 - \frac{1.761 \times 10^3}{\mathbf{T}(\text{K})} - \frac{1.611 \times 10^6}{\mathbf{T}(\text{K})^2} + \frac{0.2803 \times 10^9}{\mathbf{T}(\text{K})^3}\right) \quad (3.58)$$

where  $\mathbf{T}$  is the burned gas temperature. The equilibrium constant can then be used in conjunction with appendix C and table 3 to calculate the general HC equation constants.

**Table 3. Constants for the General HC Equation [3]**

Species	$\lambda \geq 1$	$\lambda \leq 1$
$\text{CO}_2$	$\varepsilon \left(\frac{1}{\lambda}\right)$	$\varepsilon \left(\frac{1}{\lambda}\right) - c$
$\text{H}_2\text{O}$	$2(1 - \varepsilon) \left(\frac{1}{\lambda}\right)$	$2 \left(1 - \varepsilon \left(\frac{1}{\lambda}\right)\right) + c$
$\text{CO}$	0	c
$\text{H}_2$	0	$2 \left(\left(\frac{1}{\lambda}\right) - 1\right) - c$
$\text{O}_2$	$1 - \left(\frac{1}{\lambda}\right)$	0
$\text{N}_2$	$\psi$	$\psi$
Total: $n_{\text{tot}}$	$(1 - \varepsilon) \left(\frac{1}{\lambda}\right) + 1 + \psi$	$(2 - \varepsilon) \left(\frac{1}{\lambda}\right) + \psi$

By ignoring the prompt and NNH formation mechanisms described in section 2.3, the Zeldovich mechanism can be expanded to calculate a theoretical NO profile. For the purpose of this model, only the general Zeldovich mechanism, equations 2.7 and 2.8, was used. A full outline of the expansion of the Zeldovich mechanism can be viewed in appendix B; however, the resulting term is [6]:

$$\frac{d[\text{NO}]}{dt} = 2k_{1f}[\text{N}_2]_e[\text{O}]_e \quad (3.59)$$

where [i] represents the respective constituent concentration,  $k_{1f}$  represents the forward reaction rate coefficient, and the subscript e represents an equilibrium concentration.

According to Turns [6], the forward reaction rate coefficient fluctuates as a function of combustion chamber temperature and is defined as:

$$k_{1f} \left( \frac{\text{cm}^3}{\text{gmol-s}} \right) = (1.82 * 10^{14}) \exp \left( -\frac{38370}{T(\text{K})} \right) \quad (3.60)$$

where the input temperature is the burned gas temperature ( $T_b$ ). The equilibrium concentration of oxygen is defined as:

$$[\text{O}]_e = \frac{K_o[\text{O}_2]_e^{\frac{1}{2}}}{(\text{R}_u T)^{\frac{1}{2}}} \quad (3.61)$$

where  $\text{R}_u$  is the universal gas constant and  $K_o$  is defined as [3] :

$$K_o \left( \text{Pa}^{\frac{1}{2}} \right) = 3.6 \times 10^3 \exp \left( -\frac{31090}{T} \right) \times (101325)^{\frac{1}{2}} \quad (3.62)$$

which is the equilibrium constant for reaction 7.

Although equations 3.61 and 3.62 seem relatively straightforward, it was found that  $[\text{O}_2]_e$  was difficult to predict in fuel-rich combustion, because of high temperature dissociation. The formation of  $\text{O}_2$  and additional CO in fuel-rich combustion can be attributed to the dissociation of  $\text{CO}_2$ :



where the equilibrium constant can be found by looking up each constituent in the JANAF tables at the given burned gas temperature.

Since the O<sub>2</sub> equilibrium constant was zero at all temperatures, only the CO<sub>2</sub> and CO equilibrium constants had to be calculated. The equilibrium constants and corresponding temperatures were extracted from the JANAF database and fitted to a 4<sup>th</sup> order polynomial using MATLAB's plotting features (a further discussion on this topic can be found in section 5.6). Using equation 3.63 and the 4<sup>th</sup> order polynomials, the overall equilibrium constant is defined as:

$$\log_{10}K_p = K_{p_{CO_2}} - K_{p_{CO}} \quad (3.64)$$

where  $K_p$  is the overall equilibrium constant,  $K_{p_{CO_2}}$  is the equilibrium constant for CO<sub>2</sub>, and  $K_{p_{CO}}$  is the equilibrium constant for CO.

The fraction of dissociation can then be defined relative to the overall equilibrium constant [3]:

$$\frac{1-\alpha}{\alpha\left(\frac{\alpha}{2}\right)^{\frac{1}{2}}}\left(\frac{n_p}{P_p}\right)^{\frac{1}{2}} = K_p \quad (3.65)$$

where  $\alpha$  is the fraction of CO<sub>2</sub> dissociation. The constant  $\frac{n_p}{P_p}$  is the ratio of product moles to product partial-pressures and is expressed as [3]:

$$\frac{n_p}{P_p} = \frac{1.5}{P_{EXH}} \left( \frac{T}{T_{BDC}} \right) \quad (3.66)$$

where  $P_{EXH}$  is the exhaust pressure and  $T_{BDC}$  is the temperature at bottom-dead-center (BDC), after the introduction of EGR. With a known fraction of dissociation, the molar fraction of O<sub>2</sub> is then defined as:

$$x_{O_2} = \frac{\alpha}{1.5} \left( x_{CO_2} \left( \frac{1-\alpha}{n_p} \right) \right) \quad (3.67)$$

where  $x_{\text{CO}_2}$  is found using table 3 and the general HC combustion equation, and  $\left(\frac{1-\alpha}{n_p}\right)$  modifies the original molar fraction of  $\text{CO}_2$  based on dissociation.

### 3.10 HC Emissions Prediction

In modeling HC emissions, the temperature of the gases contained within the crevices can be assumed to be equivalent to that of the coolant. The mass of the crevice gasses is defined as [12]:

$$m_{\text{crevice}} = \frac{pV_{\text{crevice}}MW_{\text{gas}}}{RT_{\text{coolant}}} \quad (3.68)$$

where  $V_{\text{crevice}}$  is the crevice volume,  $MW_{\text{gas}}$  is the molecular weight of the crevice gas,  $R$  is the universal gas constant, and  $T_{\text{coolant}}$  is the coolant temperature. The overall crevice mass coincides with cylinder pressure; therefore, a peak in crevice mass can be found at peak cylinder pressures.

The crevice gases can be broken into unburned gases and fuel vapor, which are defined in a fraction form as [12]:

$$f_{\text{unburned}} = (1 - x_r - \text{EGR}) \quad (3.69)$$

$$f_{\text{vapor}} = \frac{1}{1 + AF_{\text{ac}}} \quad (3.70)$$

where  $x_r$  is the residual gas fraction and  $\text{EGR}$  is the fraction of exhaust gas recirculated.

Hamrin and Heywood [12] have suggested using a modification factor for the offset of the spark-plug in relation to the cylinder center-axis, because side-mounted spark plugs have been shown to affect the fraction of unburned and burned gases within crevices. This modification factor is defined as [12]:

$$f_{\text{mod}} = \left(1 - 0.858 \left(\frac{d_{\text{spark}}(\text{m})}{B(\text{m})}\right)\right) \quad (3.71)$$

where  $d_{\text{spark}}$  is the offset distance of the spark plug from the central axis of the cylinder, and  $B$  is the cylinder bore. As one can see, this term becomes unity when the spark plug is centrally located.

Combining equations 3.68-3.71 with known engine parameters, the crevice emissions index is defined as [12]:

$$SF_{\text{crevice}} = 5443 \left( \frac{P_{\text{peak}}}{\text{IMEP}} \right) \left( \frac{V_{\text{crevice}}}{\frac{V_d}{N_{\text{cyl}}}} \right) \left( \frac{1}{T_{\text{coolant(K)}}} \right) (f_{\text{unburned}})(f_{\text{vapor}})(f_{\text{mod}}) \quad (3.72)$$

where  $P_{\text{peak}}$  is the peak cylinder pressure,  $\text{IMEP}$  is the indicated mean effective pressure, and  $N_{\text{cyl}}$  is the number of cylinders.

In predicting HC emissions formation due to oil layer absorption and desorption, the first step is to quantify the residual layer of oil on the cylinder wall, which is expressed as [12]:

$$m_{\text{oil}} = \rho_{\text{oil}} \pi \delta_{\text{oil}} BS \quad (3.73)$$

where  $m_{\text{oil}}$  is the mass of the oil film,  $\rho_{\text{oil}}$  is the density of the oil film, and  $\delta_{\text{oil}}$  is the thickness of the oil film.. Hamrin and Heywood [12] have suggested using constant density ( $\rho_{\text{oil}} \sim 900 \left[ \frac{\text{kg}}{\text{m}^3} \right]$ ) and constant oil layer thickness ( $\delta_{\text{oil}} \sim 3 [\mu\text{m}]$ ) values.

The mole fraction of absorbed fuel in the oil film on the cylinder liner wall can be predicted using Henry's law:

$$x_{\text{fuel}} = \frac{P_{\text{fuel}}}{H} \quad (3.74)$$

where  $x_{\text{fuel}}$  is the absorbed molar fraction of fuel,  $P_{\text{fuel}}$  is the partial pressure of absorbed fuel, and  $H$  is Henry's constant. Hamrin and Heywood [12] have suggested using the relationship:

$$\bar{H} = H \left( \frac{MW_{\text{oil}}}{MW_{\text{fuel}}} \right) \sim 10^{0.0082 T_{\text{oil(K)}}} \quad (3.75)$$

where  $MW_{oil}$  is the molecular weight of the oil,  $MW_{fuel}$  is the molecular weight of the fuel, and  $T_{oil}$  is the temperature of the oil. For the purpose of this model, the temperature of the oil was assumed to be equivalent to the coolant temperature. Using Henry's law and the mass of the residual oil film, the mass of unburned HC is defined as:

$$m_{HC} = p \left( \frac{1}{AF_{ac}} \right) \left( \frac{MW_{air}}{MW_{HC}} \right) \left( \frac{MW_{HC}}{MW_{oilH}} \right) m_{oil} \quad (3.76)$$

where  $MW_{air}$  is the molecular weight of air, and  $MW_{HC}$  is the molecular weight of the absorbed HC.

With the mass of unburned HC being dependent on the bulk gas pressure, it's obvious that a maximum pressure would produce the highest percentage of unburned HC emissions. Hamrin and Heywood [12] have suggested using an average pressure between the inlet and peak conditions:

$$p(atm) = \frac{P_{inlet(atm)} + P_{inlet(atm)} R_c^\gamma}{2} \quad (3.77)$$

where  $p_{inlet}$  is the inlet pressure,  $R_c$  is the compression ratio, and  $\gamma$  is the specific heat ratio.

In defining the inlet pressure, Hamrin and Heywood [12] developed a fitted equation defined as:

$$P_{inlet(atm)} = 0.09875 + 0.00986IMEP(kPa) \quad (3.78)$$

where the inlet pressure is adjusted relative to spark-timing, temperature, and other operating conditions.

In combining the previously-defined equations, the unburned HC emissions index is expressed as:

$$SF_{wall} = 63024 \left( \frac{1}{IMEP(kPa)} \right) \left( \frac{1}{AF_{ac} (10^{0.0082T_{oil}(K)})^{B(m)}} \right) \left( \frac{P_{inlet(atm)} + P_{inlet(atm)} R_c^\gamma}{2} \right). \quad (3.79)$$

where the  $IMEP$  term comes from a relationship between  $IMEP$  and  $p_{inlet}$  that was found experimentally [12].

Throughout the expansion process, many of the unburned HC are able to oxidize due to the high temperature of the combustion products. The point at which 50% of these unburned HCs continue into the exhaust without oxidizing is known as the threshold of HC oxidation, and corresponds to a crank position of approximately 70° after top-dead center (ATDC)[12]. It should be noted that the oxidation threshold is highly dependent on load and operating parameters and should thus be tailored to a specific platform.

In calculating the fraction of HC oxidation within the cylinder, the log mean average temperature is used to relate the bulk gas temperature and cylinder wall temperatures [12].

At the threshold of HC oxidation the log mean average temperature is expressed as:

$$\bar{T}_{\text{HC}}(\text{K}) = \frac{T(250) - T_{\text{coolant}}}{\ln\left(\frac{T(250)}{T_{\text{coolant}}}\right)} \quad (3.80)$$

where  $T(250)$  is the bulk gas temperature, 70° ATDC. Using empirical equations, the adjusted log mean average temperature at the threshold of HC oxidation is defined as:

$$\bar{T}_{\text{HCadj}}(\text{K}) = 1600 + 0.759\text{IMEP}(\text{kPa}) - 0.000051\text{IMEP}(\text{kPa})^2 \quad (3.81)$$

where the coefficients can be updated according to spark timing, EGR, compression ratio, and coolant temperature based on experimental data obtained by Hamrin and Heywood [12].

Using equations 3.79 and 3.80, the fraction of HC oxidation within the cylinder is expressed as:

$$f_{\text{ox}} = 1 - \left(\frac{P(250)(\text{atm})}{P_{\text{max}}}\right) \left(\frac{\bar{T}_{\text{HC}}}{\bar{T}_{\text{HCadj}}}\right)^3 \quad (3.82)$$

where  $P(250)$  is the bulk gas pressure 70° ATDC,  $P_{\text{max}}$  is the theoretical maximum combustion pressure, and the exponent (3) can be adjusted to fit the fraction of oxidation to measured data.

Of the HC fraction leaving the combustion chamber, a substantial amount oxidizes throughout the exhaust and in the catalytic converter. Empirical equations have been developed by several researchers [12][20] relating HC combustion within the exhaust to engine load, speed, coolant temperature, and spark-advance. The fraction of HC oxidation within the exhaust is defined as:

$$f_{\text{oxexh}} = 0.866 - 0.0000148N \left( \frac{\text{rev}}{\text{s}} \right) - 0.00071\text{IMEP}(\text{kPa}) \quad (3.83)$$

$$-0.00791\text{RELSP} - 0.0000255T_w(\text{K})$$

$$\text{RELSP} = 0.829 \left( \frac{\text{EGR}}{100} \right) \quad (3.84)$$

where  $N$  is the piston speed,  $\text{RELSP}$  is an adjustment factor for the spark advance, and  $\text{EGR}$  is the percentage of exhaust gas recirculated.

With the main emissions indices defined, an overall HC formation mechanism can be expressed as:

$$\text{HC}_{\text{formed}} = \left( \sum_i \text{SF}_i (1 - f_{\text{ox}}) \right) (f_{\text{ox}}) (1 - f_{\text{oxexh}}) \quad (3.85)$$

It should be noted that each scaling factor is summed independently and equation 3.88 can thus be expanded to:

$$\text{HC}_{\text{formed}} = (\text{SF}_{\text{crevice}}(1 - f_{\text{ox}}) + \text{SF}_{\text{wall}}(1 - f_{\text{ox}}))(f_{\text{ox}}) * (1 - f_{\text{oxexh}}) \quad (3.86)$$

$$\therefore \text{M}_{\text{HCformed}} = \text{HC}_{\text{formed}} \text{m}_f$$

where  $\text{M}_{\text{HCformed}}$  is the total mass of HC emissions formed during a particular engine cycle relative to the amount of fuel contained within the cylinder.

## **Chapter 4. MATLAB Engine Simulation Model**

In selecting a computer program to execute the demands of a multi-zone model, engineering equation solver (EES) and Matrix Laboratory (MATLAB) were carefully considered. EES was initially tested because of its ability to effectively calculate fluid properties from a built-in database. It was speculated that the accuracy of a zone-based engine model could be moderately improved with EES because of its on-hand fluid properties.

Although EES was capable of working through the most basic single-zone model, EES struggled with iterations and required careful selection of initial guesses. Although EES struggled with iterations, having fluid properties on-hand was useful, and it was found that one could use a call function within MATLAB to access fluid properties in EES. With this in mind for future modifications, a script and set of functions were constructed in MATLAB.

### **4.1 Components of the MATLAB Model**

The bulk of the MATLAB model was set up through the use of a script. The model required many unrelated equations, so functions were only used under certain circumstances.

The MATLAB code was broken into the following sub-sections:

1. Engine geometry and atmospheric inputs.
2. Pre-allocation of arrays and matrices.
3. Fuel inputs and combustions efficiencies.
4. Instantaneous engine and fluid properties.
5. Valve simulation.
6. Two-zone calculations.
7. Plot statements.
8. Emissions prediction functions.

The purpose of these sub-sections and the organization of the MATLAB model will be elucidated in subsequent sections.

## 4.2 Engine Geometry and Atmospheric Inputs

The MATLAB script began with known engine inputs. The bore, stroke, connecting rod length, number of cylinders, compression ratio, and operating characteristics were defined in lines 12-25. Figure 6 displays the engine inputs and the corresponding block of code, with a set of inputs corresponding to the Yamaha YZ250F engine.

```

%Engine Inputs
Load = 1;           %Engine Load (Affects Inlet Pressure)
RPM = 10500;       %Revolutions Per Minute [1/min]
L = (53.6/1000);   %Stroke of Engine [m]
B = (77/1000);     %Bore of Engine [m]
l = .0935;         %Length of Engine Connecting Rod [m]
N_cyl = 1;         %Number of Cylinders [unitless]
C_r = 12.5;        %Compression Ratio [unitless]
N_r = 2;           %Number of Revolutions Per Power Stroke
theta_b = 85;      %Combustion Burn Duration [degrees]
theta_0 = 145;     %Crank Angle At Start of Combustion [degrees]
theta_f = theta_0+theta_b; %Final Comb. Angle [degrees]
IVC = 0;           %Time [degrees] when Intake Valve Closes
EVO = 314;         %Time [degrees] when Exhaust Valve Opens

```

Figure 6. Engine inputs in the MATLAB model.

Based on the previously-defined inputs and constraints, the script then calculated engine geometry parameters. This block of code calculated the cross-sectional area of the piston, the surface area of the cylinder head within the combustion chamber, the displaced cylinder volume, the crank radius, and the clearance volume. As was explained in section 3.8, the surface area of the cylinder head within the combustion chamber was defined as being twice that of the cross-sectional area of the piston; this was due to the complexity of the cylinder head geometry. The script then predicted engine friction losses based on the engine speed

and stroke, and used a function to calculate a volumetric efficiency correction factor based on the engine speed. Figure 7 shows lines 29-56 of the MATLAB script, which calculated engine geometries, friction losses, and volumetric efficiencies.

```

%Engine Calculations Based On Previous Inputs
%Assumes Average Surface Area In Which Heat Transfer Occurs

A_p = (pi/4)*B^2;           %Cross Sectional Piston Area [m^2]
A_ch = 2*A_p;              %Cylinder Head Surface Area (in chamber)
V_d = N_cyl*A_p*L;        %Displaced Volume Of Engine [m^3]
N = RPM/60;                %Converts RPM to RPS [1/s]
S_bar_p = 2*L*N;          %Calculates Mean Piston Speed [m/s]
a = L/2;                   %Calculates Crank Radius (1/2 stroke) [m]
V_TDC = (V_d/(C_r-1))/N_cyl; %Calculates Clearance Volume [m^3]
V_BDC = (V_d/N_cyl)+V_TDC; %Cyl. Volume At BDC [m^3]

%
%-----
%Calculating Losses Due To Friction
%fmep (obtained from Blair) Based On Displacement, RPM

% if V_d>500*10^(-6)
%   fmep=(100000+350*L*RPM)*10^(-3);
% end
% if V_d<500*10^-6
%   fmep=(100000+100*(500-V_d*10^(-6))+350*L*RPM)*10^(-3);
%end
%For Motorcycles, Use "Rolling" Bearings (For Automobiles, Use Previous)
fmep = (250*L*RPM)*10^-3;

%Volumetric Efficiency Correction Factor
CF = correction( Load,RPM );

```

Figure 7. MATLAB calculations based on engine inputs.

The model specified the atmospheric inputs in lines 88-94 and used several of these inputs throughout the main loop to simulate EGR and other physical phenomena.

Atmospheric pressure was reduced to simulate operating conditions in Moscow, ID, and a temperature of 350[K] was chosen to represent the cylinder wall temperature, per the suggestion of Stone [5]. An initial inlet temperature of 290[K] was specified in line 112;

with the initial inlet temperature being placed in the main loop because of EGR, and an if-then statement that corrected the inlet temperature as a function of iterations.

### **4.3 Pre-allocation of Arrays and Matrices**

Through experimentation, it was found that pre-allocating arrays and matrices drastically improved the efficiency of the program. This prevented MATLAB from having to re-size arrays or matrices between iterations, thus decreasing the overall computation time. Pre-allocated arrays and matrices were also used as a means of setting appropriate properties equal to their initial values with the introduction of EGR; where functionality of the EGR simulation required the script to run two times with only the gas temperature and fluid properties changing during the second iteration. Array and matrix pre-allocation was specified in lines 58-70 and 126-175.

### **4.4 Combustion Efficiency and Fuel Inputs**

The fuel, air-fuel ratio, and combustion efficiency inputs were placed in lines 76-86 of the MATLAB script. A lower heating value (LHV) of  $44.6 \left[ \frac{\text{MJ}}{\text{kg}} \right]$  was used per the suggestion of Stone [4], and a maximum combustion efficiency of 0.95 was selected based on intuition. The cycle combustion efficiency was then calculated using empirical equations developed by Blair [5] that took the maximum combustion efficiency and the excess air coefficient into account. The fuel and combustion efficiency inputs can be observed in figure 8.

```

%Fuel Inputs/Efficiencies

AF_ratio_stoich = 15.09;    %Gravimetric Air Fuel Ratio (Stoich)
AF_ratio_mol_stoich=14.7; %Molar Air_Fuel Ratio (Stoich)
lambda = .90;             %Excess Air Coefficient
AF_ratio_ac = lambda*AF_ratio_stoich; %Actual Air Fuel Ratio
AF_ratio_mol=lambda*AF_ratio_mol_stoich;
LHV = 44.6e6;             %Lower Heating Value Of Fuel Mixture [J/kg]
eta_combmax = .95;       %Assumed MAX Comb. Efficiency

%Predicts Combustion Efficiency (Reference To Blair)

eta_comb=eta_combmax*(-1.6082+4.6509*lambda-2.0764*lambda^2);

```

Figure 8. Fuel inputs and combustion efficiencies.

#### 4.5 Instantaneous Engine and Fluid Properties

The instantaneous engine properties were calculated within the main loop of the MATLAB script, which fell between lines 109-374. The main loop was broken into two sub-loops that served different functions. The loop with a specified index ( $k = 1:2$ ) served as the EGR simulation, while the loop with a specified index ( $i = 2:360$ ) calculated instantaneous engine features, which will be discussed in the subsequent paragraphs. Fluid properties, the pre-allocation of arrays, and temperature corrections factors were placed between the first and second sub-loops; that way, all fluid and gas properties were updated as a function of EGR.

In the second sub-loop, the combustion chamber volume, instantaneous heat transfer area and overall heat transfer, Weibe function, and all other instantaneous engine characteristics were calculated. Lines 182-200 calculated geometric properties such as the instantaneous cylinder volume; these lines also calculated viscosity, thermal conductivity, and other instantaneous gas properties. The Weibe function and fuel-mass contained within the cylinder were calculated in lines 208-215, where an if-then statement was used to specify

the burn fraction as being zero until the cycle reached the spark advance. This is illustrated in figure 9.

```

if theta(i)<theta_0
    X(i)=0;
else
    X(i) = 1-exp(-5*((theta(i)-theta_0)/theta_b)^3);
    if theta(i) < theta_f
        M_F(i) = V(theta_0-1)*rho(theta_0-1)/(lambda*AF_ratio_mol);
    end
end
end

```

Figure 9. The Weibe function and fuel mass within the cylinder.

#### 4.6 Valve Simulation

Lines 247-260 simulated the opening and closing of intake and exhaust valves, which was assumed to be instantaneous; that is, the gas dynamics and valve lift profile weren't considered. The opening and closing of the valves can be seen in figure 10.

```

if IVC< theta(i)
    DT(i)=T(i-1)*(gamma(i-1)-1)*((1/(P(i-1)*V(i-1)))*DQ(i)...
        -(1/V(i-1))*DV(i));
    DP(i)=(-P(i-1)/V(i-1))*DV(i)+(P(i-1)/T(i-1))*DT(i);
    P(i) = P(i-1)+DP(i);
end
if EVO < theta(i)
    P(i) = P_atm;
end
if 200 < theta(i)
    if P(i)<=P_atm
        P(i)=P_atm;
    end
end
end

```

Figure 10. The opening and closing of valves.

It can be observed that this script included a statement referencing 200 crank-angle degrees. Upon the opening of the exhaust valves, the cycle pressure was set equal to atmospheric pressure, but at part load, the cycle pressure became negative before the opening of the

valves. To prevent negative cycle pressure, an if-then statement was created for all crank degrees past 200, with 200 degrees being arbitrarily chosen.

#### **4.7 Two-zone calculations**

Lines 258-284 were reserved for two-zone calculations such as the burned and unburned masses, volumes, temperatures, and areas, while much of the rest of the second sub-loop was occupied by specific heat ratio calculations. The two-zone calculations used variables from the single-zone calculations such as the bulk gas pressure and the mass fraction burned to calculate two-zone characteristics. Lines 311-357 used coefficients that were defined in lines 97-105 to calculate the temperature dependent specific heat ratios; where the coefficients were defined outside of the main loop because they were unchanging. Therefore, less information within the main loop resulted in a more efficient simulation.

Line 292 calculated the residual fraction of exhaust gases within the combustion chamber based on polytropic relationships, and line 368 calculated a corrected temperature based on the volumetric ratios of residual and inlet gases. The calculations were placed on different lines, because of the constantly-updating temperatures relative to crank-angle.

The first sub-loop iterated twice, with the first iteration assuming inlet gas properties equivalent to atmospheric properties, while the second iteration used the corrected temperature to update gas properties. This resulted in reduced peak pressures, temperatures, and NO emissions, and thus simulated EGR.

#### **4.8 Plot Statements**

Lines 416-463 were reserved for plot statements. Each plot was sized based on the minimum and maximum variable values, and each plot was given a title appropriate to the variable being plotted. Plots 1 and 2 showed the mass-fraction burned and cylinder volume

as a function of crank angle. Plots 3 and 4 displayed the cylinder pressure and temperature as a function of crank angle. Plot 5 showed the engine power and heat transfer during a cycle, and plot 6 displayed the burned and unburned zone temperatures as a function of crank angle. In predicting emissions and overall engine performance, the plot statements were typically commented out to reduce simulation times.

#### **4.9 Emissions Prediction Functions**

The NO prediction model was set up as a function to reduce wasted space within the MATLAB script. With only one sought after output, it was determined that storage of the NO model within the script was unnecessary. Lines 2-4 of the function were designated as inputs based on the fuel, and would thus need to be changed if alternative fuels were being modeled. Lines 12-37 represented the equations found in table 3 and a quadratic method for solving these, which can be found in appendix C.

Lines 43-60 were designated for fuel-rich combustion and followed the process described in section 3.2.4. Lines 46 and 48 calculated the equilibrium constant for CO<sub>2</sub> and CO, respectively, and line 49 calculated the overall equilibrium constant for CO<sub>2</sub> dissociation. Lines 51-58 were built using MATLAB's symbolic solver and equation 3.67. Figure 11 shows the equilibrium calculations, and the large equation in the center represents equation 3.67 in terms of  $\alpha$ .

```

if 1/lambda>1
  n_prod=1.5;
  z=(T_NO-3.1*10^3)/(1.7*10^3);
  K_p_CO2=1.7*z^8-2.6*z^7-5.8*z^6+7.5*z^5+7.6*z^4-8.4*z^3-.51*z^2-2.1*z+...
  6.8;
  Z=(T_NO-3.1*10^3)/(1.7*10^3);
  K_p_CO=.48*Z^8-.72*Z^7-1.6*Z^6+2.1*Z^4-2.4*Z^3-.13*Z^2-.74*Z+6.4;
  K_P=10^(K_p_CO2-K_p_CO);
  P_p = ((P_EXH/101325)/n_prod)*(T_NO/T_BDC))^(-1);
  ALPHA= (2*P_p)/(3*K_P^2) + ((P_p/K_P^2 - (4*P_p^2)/(3*K_P^4) + ...
    (8*P_p^3)/(27*K_P^6))^2 + ((4*P_p)/(3*K_P^2) - ...
    (4*P_p^2)/(9*K_P^4))^3)^(1/2) + P_p/K_P^2 - (4*P_p^2)/(3*K_P^4) + ...
    (8*P_p^3)/(27*K_P^6))^^(1/3) - ((4*P_p)/(3*K_P^2) - ...
    (4*P_p^2)/(9*K_P^4))/((P_p/K_P^2 - (4*P_p^2)/(3*K_P^4) + ...
    (8*P_p^3)/(27*K_P^6))^2 + ((4*P_p)/(3*K_P^2) - ...
    (4*P_p^2)/(9*K_P^4))^3)^(1/2) + P_p/K_P^2 - (4*P_p^2)/(3*K_P^4) + ...
    (8*P_p^3)/(27*K_P^6))^^(1/3);
  x_O2=(ALPHA/(2*n_prod))*(x_CO2*((1-ALPHA)/n_prod));
end

```

**Figure 11. Fuel-rich calculations.**

The rest of the function calculated other necessary equilibrium constants and molar fractions of constituents, and also predicted the quantitative fraction of NO particles. The residence time for NO formation was calculated in line 75, and the integrated amount of NO was calculated in line 77.

The HC function was much simpler than the NO function, because all variables were calculated based on empirical equations. Lines 2-7 calculated the spark plug offset and crevice volume based on a known crevice height and width. Lines 8-16 calculated the fraction of unburned gases and fuel vapors within the combustion chamber, a modification factor based on the sparkplug offset, and an emissions index based on the aforementioned fractions and modification factor. The temperature and pressure thresholds were calculated in lines 21-26 based on empirical information. Although these values could have just as easily been extracted from the numerical code, it was decided during the troubleshooting period that the empirical methods would be used for consistency.

## **Chapter 5. MATLAB Model Validation**

After setting up the MATLAB script, the corresponding outputs were analyzed and compared to known results to determine their validity. This chapter begins by defining the model inputs that were used to simulate outputs for the Yamaha YZ250F engine. Next, the mass-fraction burned profile was plotted and compared to profiles obtained using the polynomial and apparent heat release models; this verified that the Weibe function was working correctly. The pressure trace and two-zone temperatures were then compared to peak combustion pressures and temperatures to verify that the rest of the MATLAB model was accurate. Lastly, the emissions models were confirmed by comparing fitted equations to experimental data.

### **5.1 Model Inputs**

The MATLAB script required engine geometry and atmospheric inputs before simulations were run. Table 4 shows the necessary inputs and assumptions that were made to simulate engine performance.

Table 4. MATLAB Inputs When Running the YZ250F simulation.

<u>Constraint</u>	<u>Value</u>	<u>Reason For Assumed Value</u>
Load	1	Assuming full throttle.
RPM	10500	Approximate location at which peak power occurs.
Stroke (L)	(53.6/1000)	Stroke is 44.5 mm.
Bore (B)	(77/1000)	Engine bore is 65.5 mm.
Connecting Rod (l)	0.0935	Connecting rod length (m).
Number of Cylinders (N_cyl)	1	N/A
Compression Ratio (C_r)	12.5:1	N/A
Burn Duration (theta_b)	85	Used peak cylinder pressures and temperatures to predict the burn duration.
Spark Advance (theta_0)	145	A sweep of the spark advance showed peak timing between 140 and 145 degrees.
Intake Valve Closing (IVC)	0	Valve overlap was considered non-existent. Pressure began building at bottom-dead-center (BDC).
Exhaust Valve Opening (EVO)	314	Camshaft profiles showed exhaust valve opening 46.5 degrees before bottom-dead-center (BBDC).
Lambda ( $\lambda$ )	0.90	At peak power, $\lambda$ is fairly rich.
Max Combustion Efficiency (eta_combmax)	0.95	This was an assumed peak value. The actual, calculated value fluctuated based on $\lambda$ .
Atmospheric Pressure (P_atm)	101325	The atmospheric pressure was assumed to be that of sea-level (to compare to stock data).
Atmospheric Temperature (T_atm)	290	An assumed temperature to represent spring-like weather in Moscow, ID.
Cylinder Wall Temperature (T_w)	350	This value coincided with suggestions made by Stone.
EGR	0	EGR was turned off during the simulation; the engine didn't have EGR.

Upon running a simulation with the previously-stated inputs and assumptions, the model outputs were evaluated.

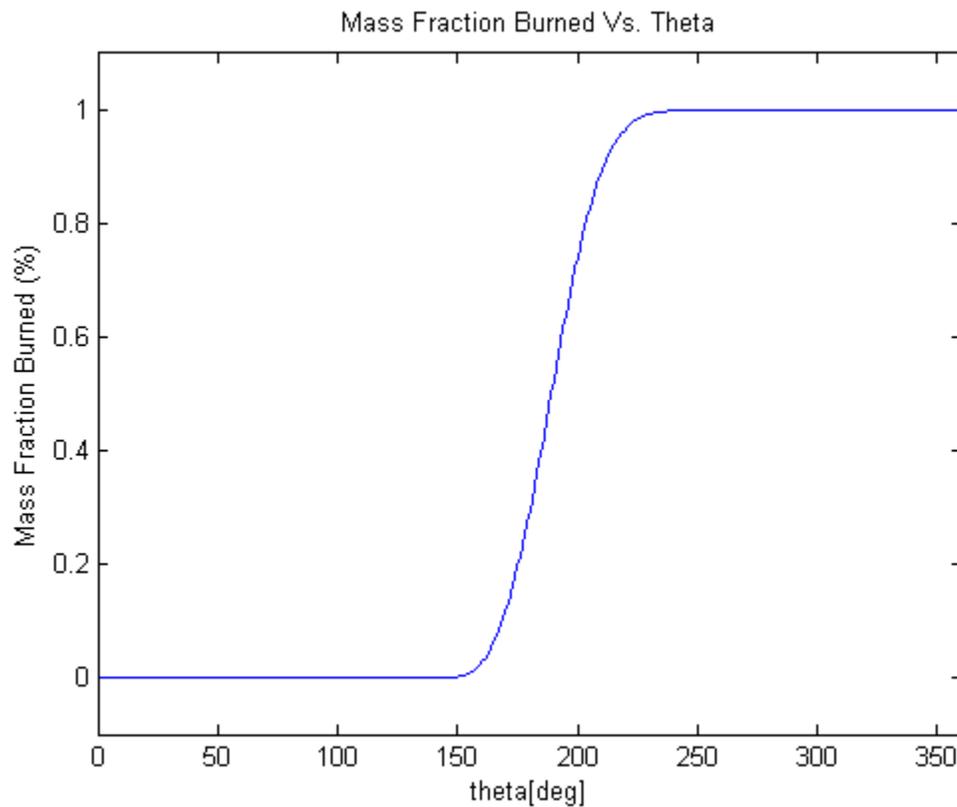
## 5.2 Weibe Function Validation

Since nearly everything in the model depended on the accuracy of the Weibe function, the mass-fraction burned profile was first inspected. The locations of burn-percentages were compared to results obtained using the polynomial and apparent heat release methods, found in the literature review. Although these results were from a completely different engine, the overall shape of the plot and plot centroid were expected to be similar. Table 5 shows the predicted location of mass-fraction burned percentages using the MATLAB model. It should be noted that all angles were referenced to  $0^\circ$ , with this being top-dead-center (TDC).

**Table 5. Predicted location of mass-fraction burned percentages**

<b>Model</b>	<b><math>\theta_{10}</math></b>	<b><math>\theta_{50}</math></b>	<b><math>\theta_{85}</math></b>	<b><math>\theta_{99}</math></b>
Weibe Function	-11	9	27	47

In comparing the MATLAB model to the polynomial and apparent heat release models (referencing Table 1), it was found that the Weibe function accurately predicted the 50% and 85% burn angles, while the 10% and 100% burn angles fluctuated based on the spark advance and burn duration. This showed that the shape of the plot was consistent with the results obtained using the polynomial and apparent heat release models, and varied based on engine parameters. Figure 12 shows the burn-profile as a function of crank angle.

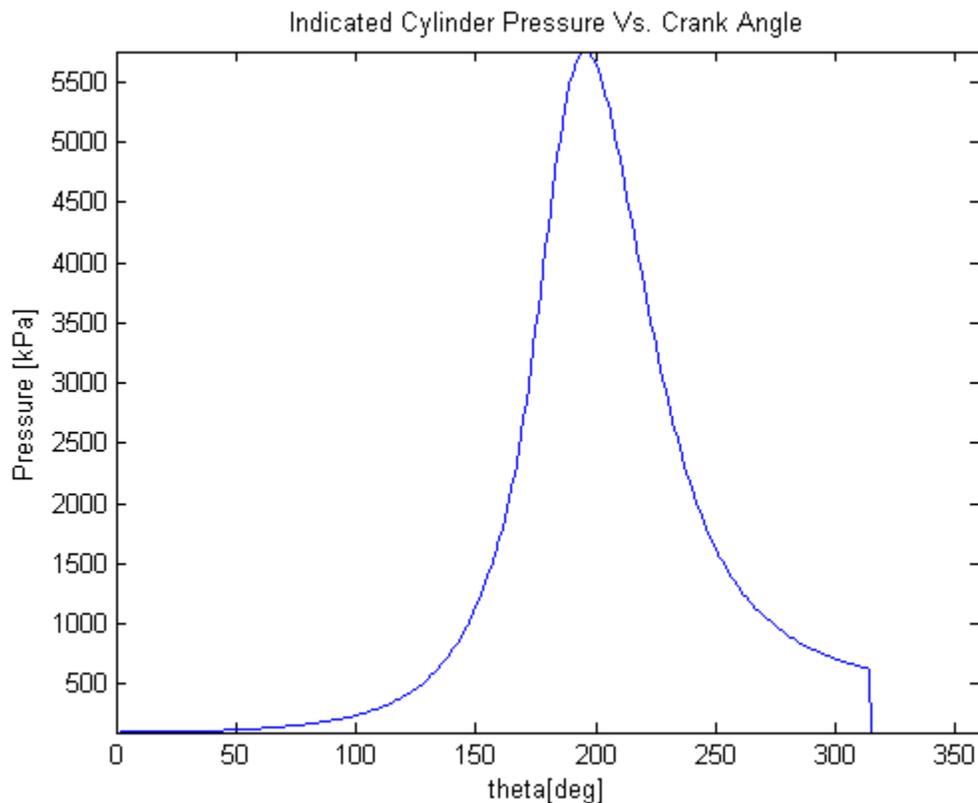


**Figure 12. The simulated YZ250F burn profile.**

In examining the burn-profile as a function of crank angle, it was found that the mass-fraction burned was zero before the spark advance. After reaching the spark advance, the profile abruptly increased, before plateauing at one. In changing the spark advance and burn duration, the plot widened and narrowed, as expected. Based on the provided information, it was decided that the Weibe function was working correctly.

### **5.3 Pressure Trace Validation**

Upon completing an analysis of the burn profile, the overall shape and peak values of the pressure trace were investigated. Figure 13 shows the predicted, indicated cylinder pressure as a function of crank angle.



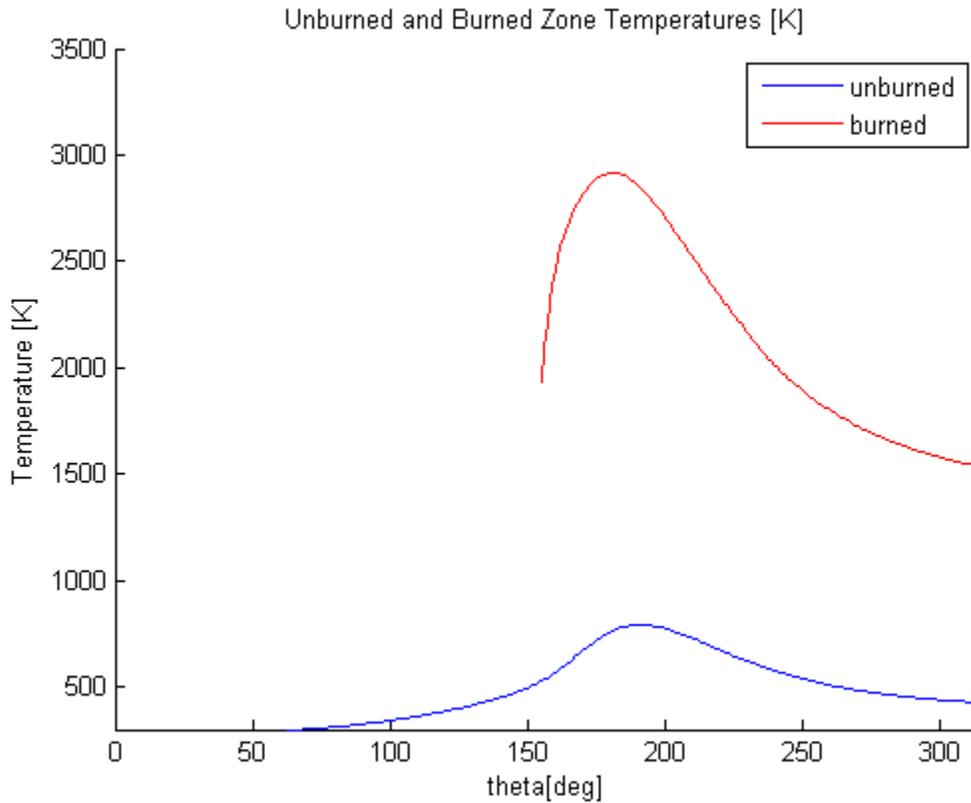
**Figure 13. Indicated cylinder pressure.**

Although the predicted peak cylinder pressure was high, it was determined that this estimate was strongly dependent on the burn duration and combustion efficiency, thus, the pressure trace could be tailored according to measured values. The indicated pressure plot also didn't include friction and other losses; therefore, the actual peak cylinder pressure would be significantly smaller. Based on factory specifications, it was determined that the predicted mean-effective pressure and correlated variables were suitably accurate.

#### **5.4 Two-Zone Temperature Validation**

An analysis of the burned and unburned gas temperatures was desired to verify that the zone-temperatures weren't considerably higher than the adiabatic flame temperature. Although it was expected that the burned-zone temperature would revolve around the

adiabatic flame temperature, it wasn't expected to be more than a few hundred degrees higher. The burned and unburned zone temperatures are displayed in figure 14.

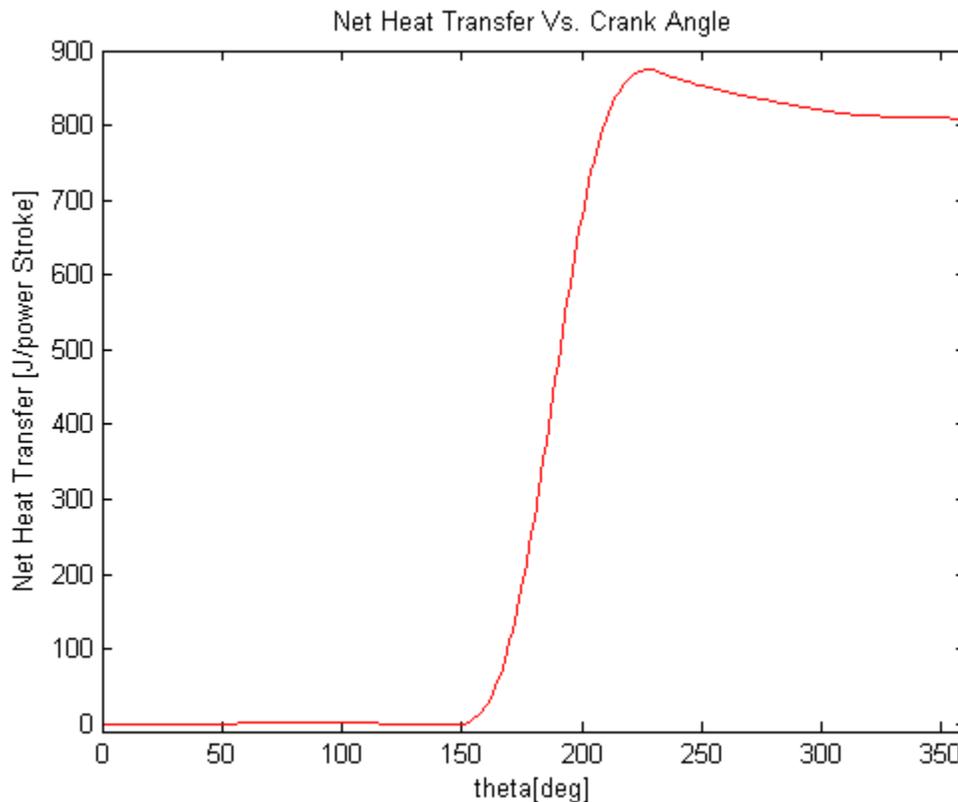


**Figure 14. Burned and unburned temperatures.**

The burned gas temperature ranged from approximately 1500-2700[K] relative to crank-angle. Although the peak burned-gas temperature was high, on average, it was consistent with the adiabatic flame temperature, which ranged from 2300-2400[K], depending on the air-fuel ratio. The unburned gas temperature began at the atmospheric temperature and peaked at around 1000[K], following a motored profile. As the unburned temperature plot approached cycle completion, it moved towards atmospheric pressure, as expected.

## 5.5 Heat Transfer Validation

With the zone-temperatures and pressure traces evaluated for accuracy, the cumulative, net heat transfer was plotted and analyzed for expected physical characteristics. In initial iterations, it was found that the heat transfer area had to be adjusted to compensate for an under-prediction of heat transfer to the combustion chamber walls; in simulating small displacement engines, this was necessary due to their small heat transfer areas. Figure 15 shows the net heat transfer per power stroke.

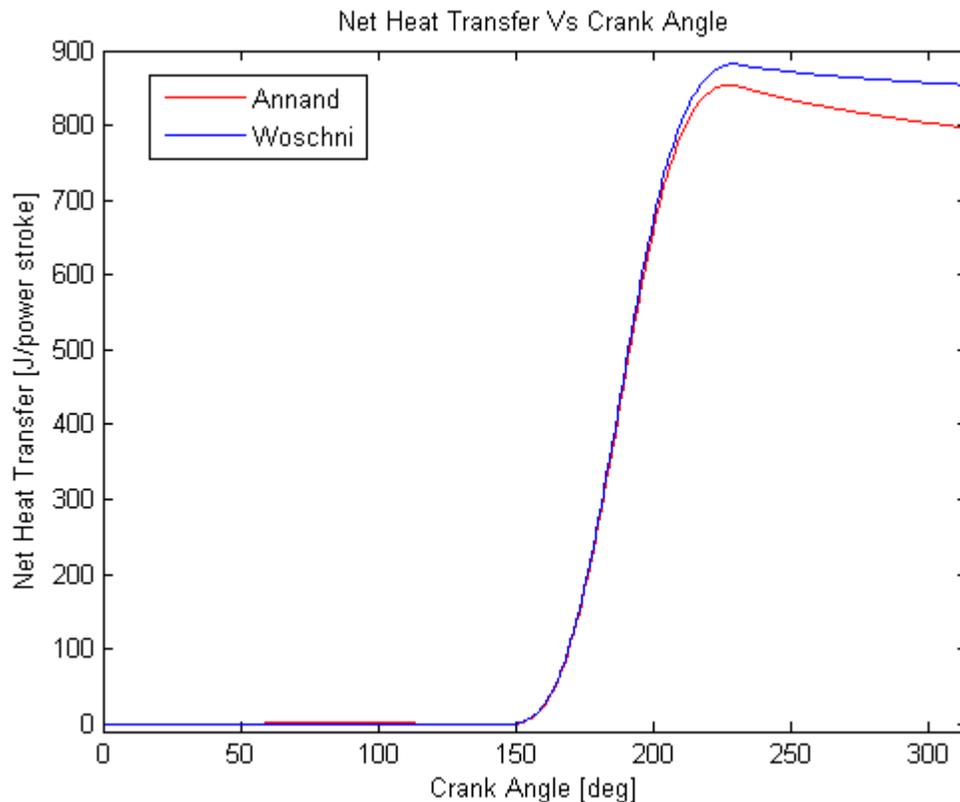


**Figure 15. Indicated net heat transfer.**

The net heat transfer plot showed the combination of fuel heat transferred into the system and convective losses transferred out. Throughout the intake and compression processes, the plot showed minimum convective gains or losses to the walls, but as the air-fuel mixture was ignited, the plot showed a rapid increase in overall heat transfer, thus

representing the input due to fuel. As the fuel burned completely, the plot leveled off and began decreasing; this represented the convective losses due to elevated combustion temperatures and gas movement.

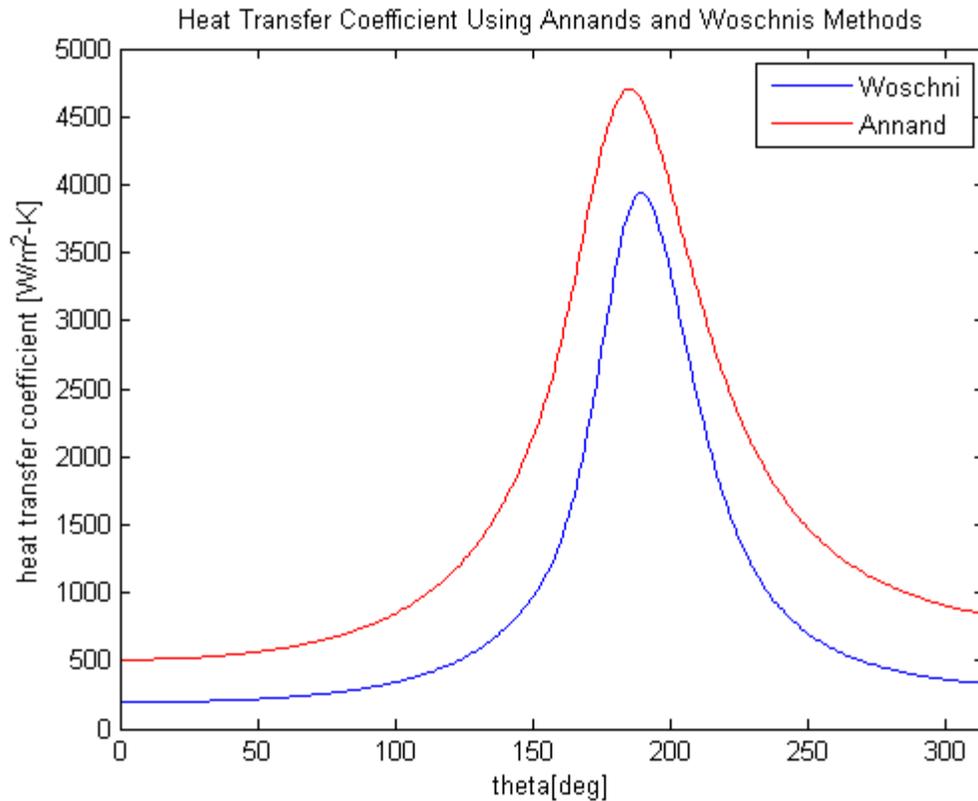
Since the model included two heat transfer prediction methods, these were compared against each other, and against the Otto cycle, for accuracy. Upon running each simulation, the overall heat transfer, heat transfer coefficient, and general outputs were studied. Figure 16 shows the net heat transfer as a function of crank-angle using each prediction method.



**Figure 16. A comparison of Annand's and Woschni's heat transfer predictions.**

It was observed that the plots obtained using Annand's and Woschni's prediction methods followed the same contours until approximately TDC, where the plots diverged for several crank-angle degrees. Otto's method assumed instantaneous heat release, and was bounded

by the lower heating value and the mass of fuel within the cylinder; this was used as a point of reference in comparing actual to idealized heat release models. To further analyze the validity of each model, the heat transfer coefficients were plotted and compared. Figure 17 shows the heat transfer coefficients as a function of crank-angle.



**Figure 17. The predicted heat transfer coefficients as a function of crank angle.**

It was found that Annand's method produced significantly higher heat transfer coefficient predictions throughout the simulated cycle. Although Annand's method predicted considerably larger heat transfer coefficients than those of Woschni's, the predicted power output, and general results were comparable. Table 6 shows the model outputs using Annand's and Woschni's heat transfer predictions, and table 7 shows indicated outputs using Otto's method.

**Table 6. A Comparison of Woschni's and Annand's Heat Transfer Prediction Methods**

<b><u>Variable</u></b>	<b><u>Woschni</u></b>	<b><u>Annand</u></b>	<b><u>Relative Error</u></b>
NO	429.57[PPM]	341.19[PPM]	20.57 [%]
HC	2.47 [%]	2.27 [%]	8.10[%]
Power	24.40 [kW]	22.76 [kW]	6.72[%]
BMEP	1117.20[kPa]	1042.10 [kPa]	6.72[%]
BSFC	334.30 [g/kW-h]	353.51 [g/kW-h]	5.75[%]
Mechanical Efficiency	88.81 [%]	88.81 [%]	0[%]
Combustion Efficiency	85.09 [%]	85.09 [%]	0[%]

**Table 7. Indicated outputs using Otto's method**

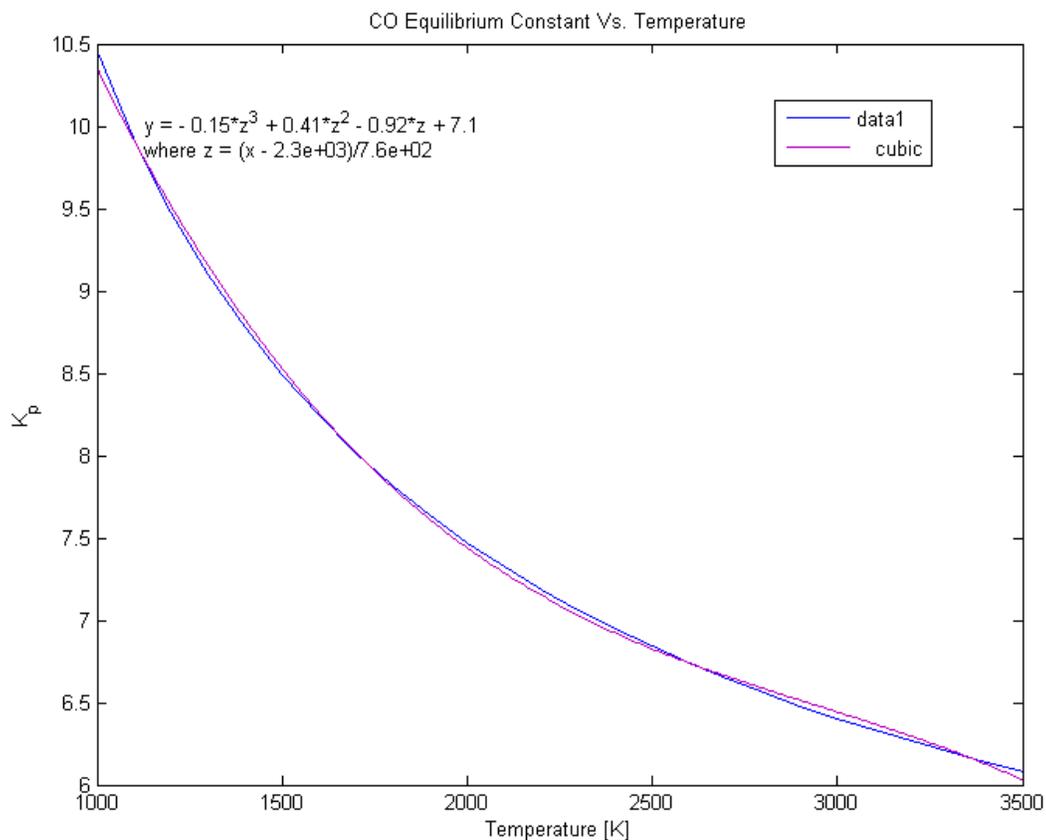
<b><u>Variable</u></b>	<b><u>Output</u></b>
Power	32.71 [kW]
IMEP	1497.70 [kPa]

Although both plots were considerably different with regard to the heat transfer coefficient prediction, most predictions were very similar, as can be seen in table 6. Even though most predictions were similar, significant differences were noticed with regard to emissions predictions; this was attributed to an elevated combustion temperature when using Woschni's method. Since emissions formation and burn-up were highly temperature-dependent, a difference of approximately 100[K] produced a moderately high relative error in reference to NO and HC emissions.

Aside from emissions predictions, it was observed that BMEP and power values predicted by both models were considerably lower than the values obtained using Otto's method (table 7), thus, it was determined that the model was working correctly. Since no significant difference was detected between the two heat transfer prediction methods, it was decided that Annand's method would be used in further simulations based on preliminary comparisons between torque curves (obtained during previous testing) [21] and model outputs.

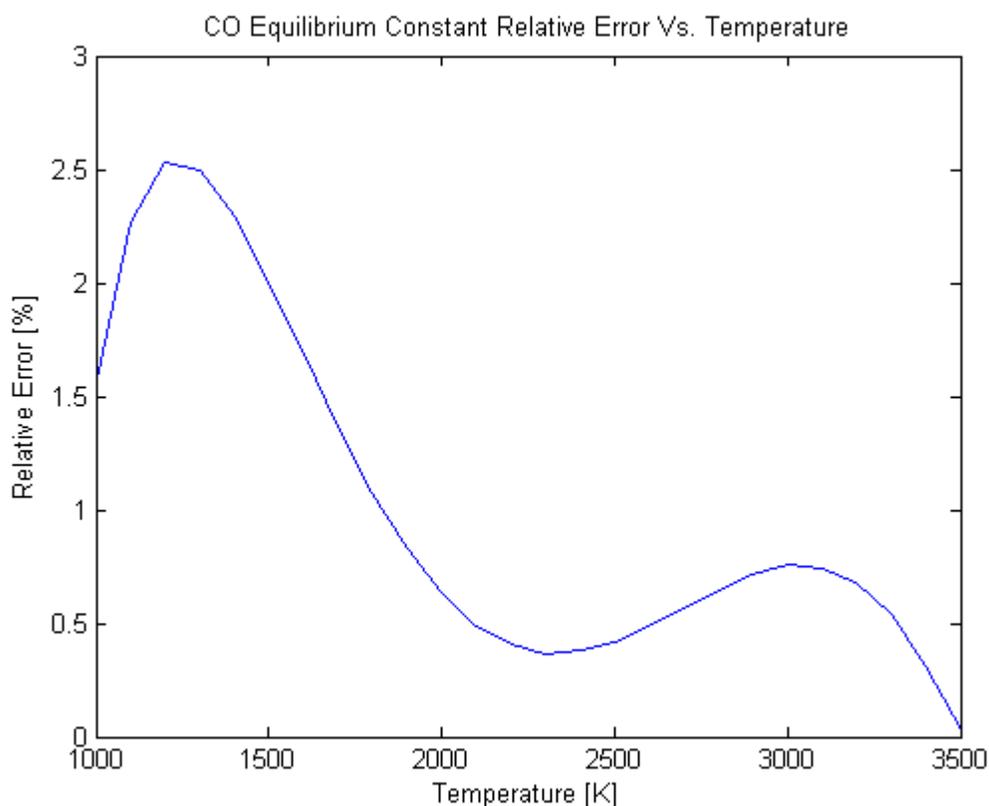
## 5.6 Validating Emissions Models

The emissions models were validated by comparing equilibrium constants to experimental data, where a statistical analysis was performed on the equilibrium constants to verify that the outputs were acceptable. Based on preliminary experimentation, the JANAF data was refined to reduce relative errors and oscillatory behavior. It was decided that the temperature range over which the JANAF data was fitted could be reduced to the normal combustion temperature range; in this case, 1000-3500 [K]. Although this restricted the model to a smaller temperature range, it was expected that burned-zone temperatures would always fall within this range. Figure 18 shows the refined polynomial curve-fitting of CO; it should be noted that the curve-fitting of CO<sub>2</sub> produced similar results.



**Figure 18. Refined curve-fitting of the CO equilibrium constant.**

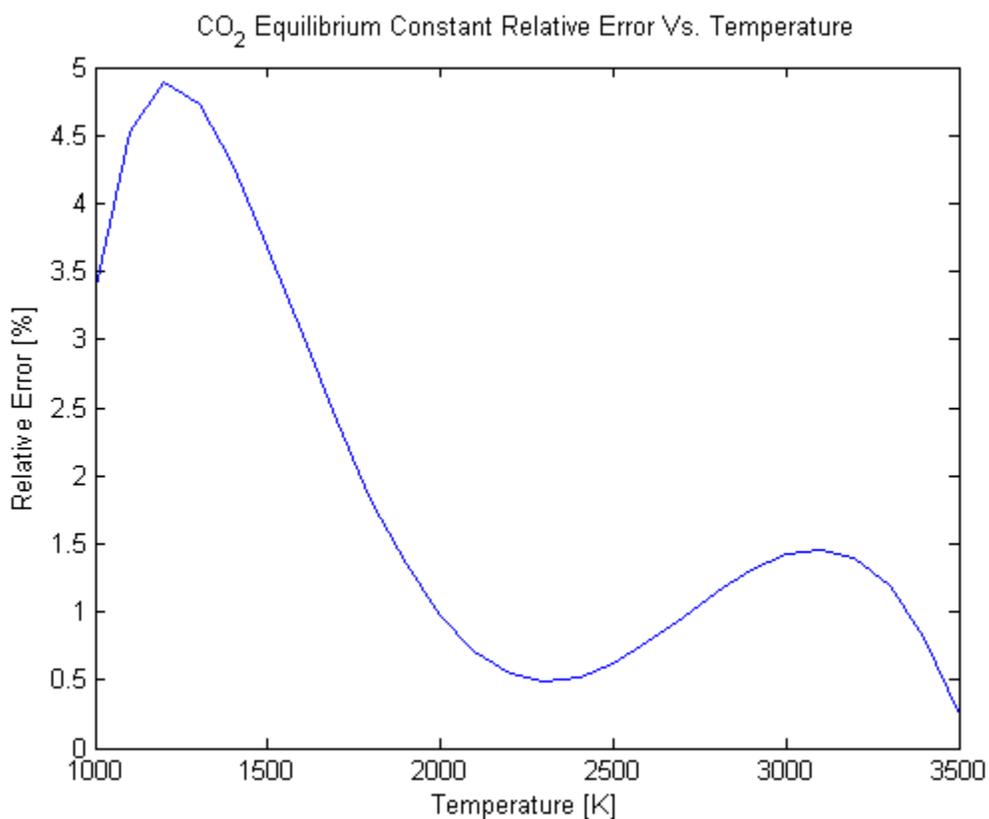
In confirming the accuracy of the equilibrium constants, each set of data was plotted using the basic fitting tools in MATLAB (as described in section 3.9). The equation provided by the MATLAB fitting tools was then compared to the JANAF data with a calculated relative error between the data sets. The relative error of the CO equilibrium constant as a function of temperature can be observed in Figure 19.



**Figure 19. Refined relative error for CO equilibrium constant.**

The relative error associated with the CO equilibrium constant reached a maximum of approximately 2.5% at 1100 [K], and across the 2000-3000 [K] range, the error was less than 1%, which was within the desired accuracy of the model. Although the relative error plot showed oscillations, the magnitudes of these oscillations were small in comparison to normal

burned-gas temperature gradients. Figure 20 shows the relative error of the  $\text{CO}_2$  equilibrium constant as a function of temperature.



**Figure 20. Refined relative error for  $\text{CO}_2$  equilibrium constant.**

Although the  $\text{CO}_2$  equilibrium constant's relative error was higher than that of  $\text{CO}$ , it was below 5% throughout the combustion temperature range. The relative error reached a peak of approximately 4.75% at 1100 [K], and in the 2000-3000 [K] range, the error was below 1.5%. With the relative error and oscillations being small, it was decided that the polynomial equilibrium constant prediction models were sufficiently accurate.

## Chapter 6. Experimental Setup

After testing the model against theoretical data, experimental data was desired to further validate the model. The experimental setup on which the YZ250F engine was tested included an emissions analyzer, a fuel cart, and an eddy current dynamometer. Figure 21 shows the experimental setup that was used to measure emissions, fuel consumption, and engine outputs.

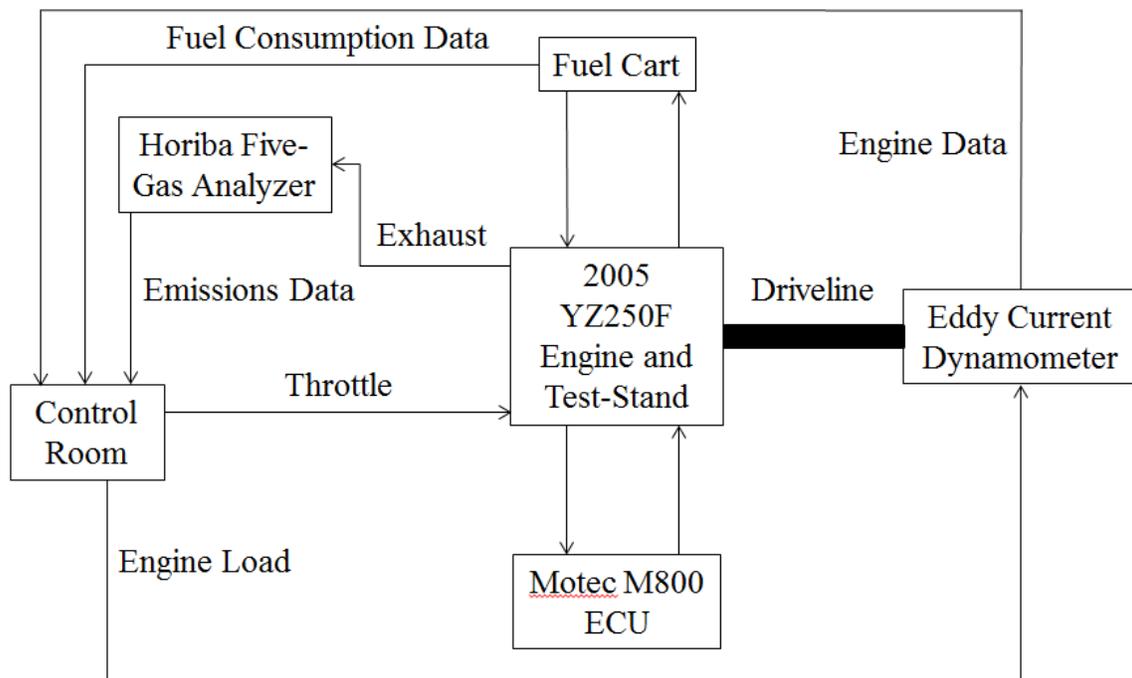


Figure 21. Experimental setup.

The subsequent sections will describe the evaluation and improvements made to the YZ250F engine, the mounting of the test-engine to the eddy-current dynamometer, the setup and calibration of the five-gas analyzer, and the setup and calibration of the fuel measurement system, as well as the uncertainties associated with each component.

## 6.1 Repairing the Dynamometer Engine

Before the YZ250F was tested on the dynamometer, the condition of the engine was evaluated. In a preliminary inspection, the cylinder leak-down was tested by spinning the crankshaft with a socket. It was observed that cylinder didn't build pressure, so the cylinder head, piston, and cylinder walls were further investigated.

In removing the camshafts, it was noticed that there was a small amount of scoring around the point of contact between the journal bearings and the intake and exhaust camshafts. It was found that the camshafts were salvageable; however, the journal bearings on the cylinder head were in poor shape, and the assembly had to be replaced. Figure 22 shows the cylinder head and scored journal bearings.

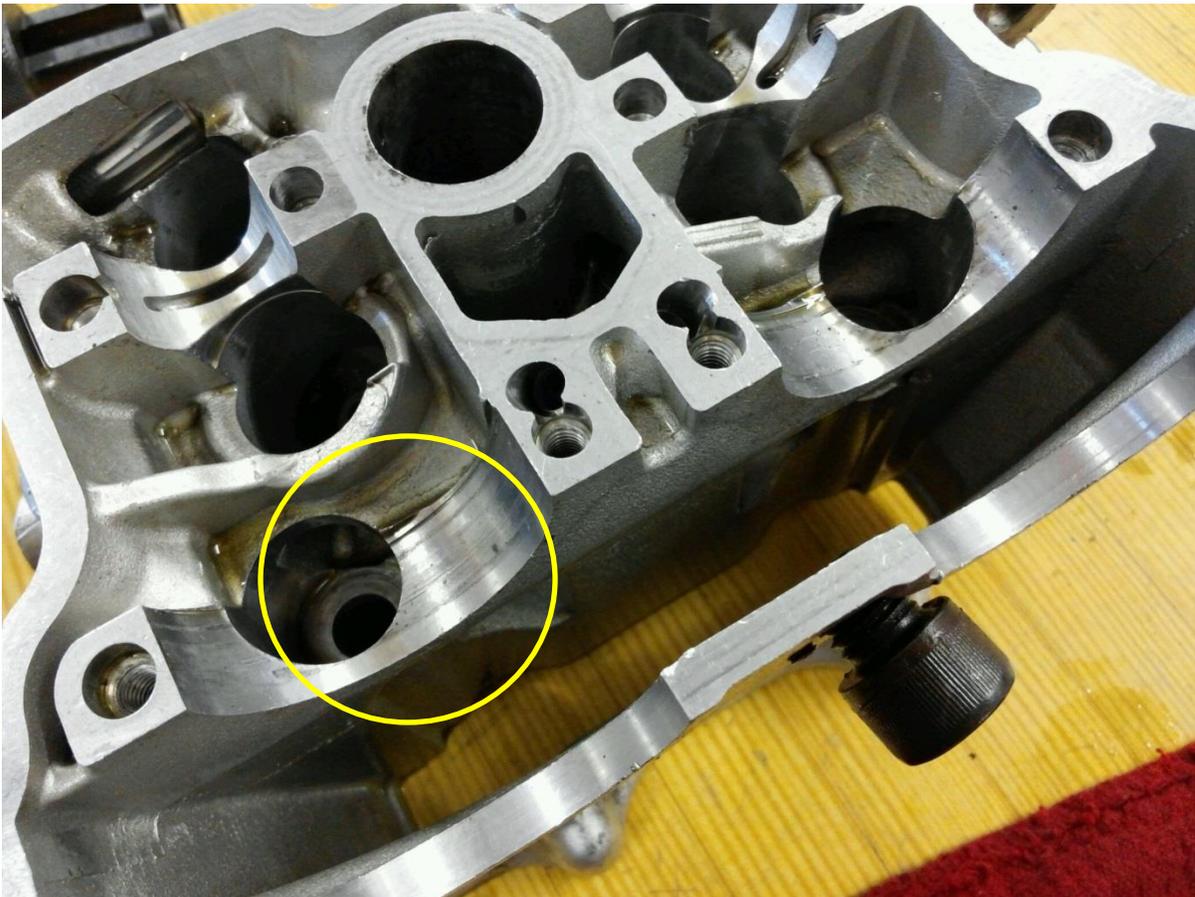


Figure 22. YZ250F cylinder head and scored bearings.

With the cylinder head removed, the cylinder and piston were next inspected for damage. It was found that the cylinder wall showed moderate discoloration and scoring, while the piston was badly scoured, and the piston rings were almost non-existent. An initial attempt was made to hone the cylinder, but it was found that the cylinder was unsalvageable, and since the condition of the piston was poor, no attempt was made to salvage it. Figure 23 shows the discoloration and scouring of the cylinder.



**Figure 23. YZ250F cylinder condition.**

Given the condition of the cylinder head, cylinder, and piston, it was decided that rest of the engine would be taken apart, cleaned, and refurbished. As the case bolts were removed and the case was split apart, it was found that the engine was filled with rust, dirt, and RTV silicone. In order to remove the rust, dirt, and grime from the engine, each component was

placed in a parts washer and scrubbed with a scotch-brite pad, and allowed to dry for several hours. In analyzing the oil pump, it was found that RTV silicone and metal shavings were nearly blocking the oil pickup screen, and it was theorized that this contributed significantly towards the cylinder and piston damage. Figure 24 shows the plugged oil-pickup screen.



**Figure 24. Plugged oil pickup screen.**

Since so much RTV sealant was found on the oil pickup screen, it was decided that ample time would be provided in allowing the new case gasket to dry. Each gasket was allowed to dry overnight, and the engine wasn't filled with oil for several days; this assured that RTV sealant wouldn't mix with the engine oil and cause this problem again. After removing the dirt, grease, and rust from engine components, the engine was put back together, re-sealed, and mounted to the eddy-current dynamometer.

## 6.2 Mounting the Test-Engine to the Eddy-Current Dynamometer

After moving the test-engine to the Small Engines Research Facility, an eddy-current dynamometer was setup for testing. In mounting the test-engine to the dynamometer, the test-stand was held in place by an adjustable fastener on each side of the frame, and the output shaft of the YZ250F was bolted to a driveline. The fasteners used between the driveshaft and the dynamometer were grade 5,  $\frac{1}{2}$ " – 13 bolts, and the required tightening torque was 75 [lbf\*ft]. Medium grade Loctite was used as an extra precaution on all bolts. Aside from mounting the driveshaft to the output shaft of the engine, the eddy-current dynamometer required very little setup time; figure 25 displays the test-engine and stand mounted to the dynamometer.

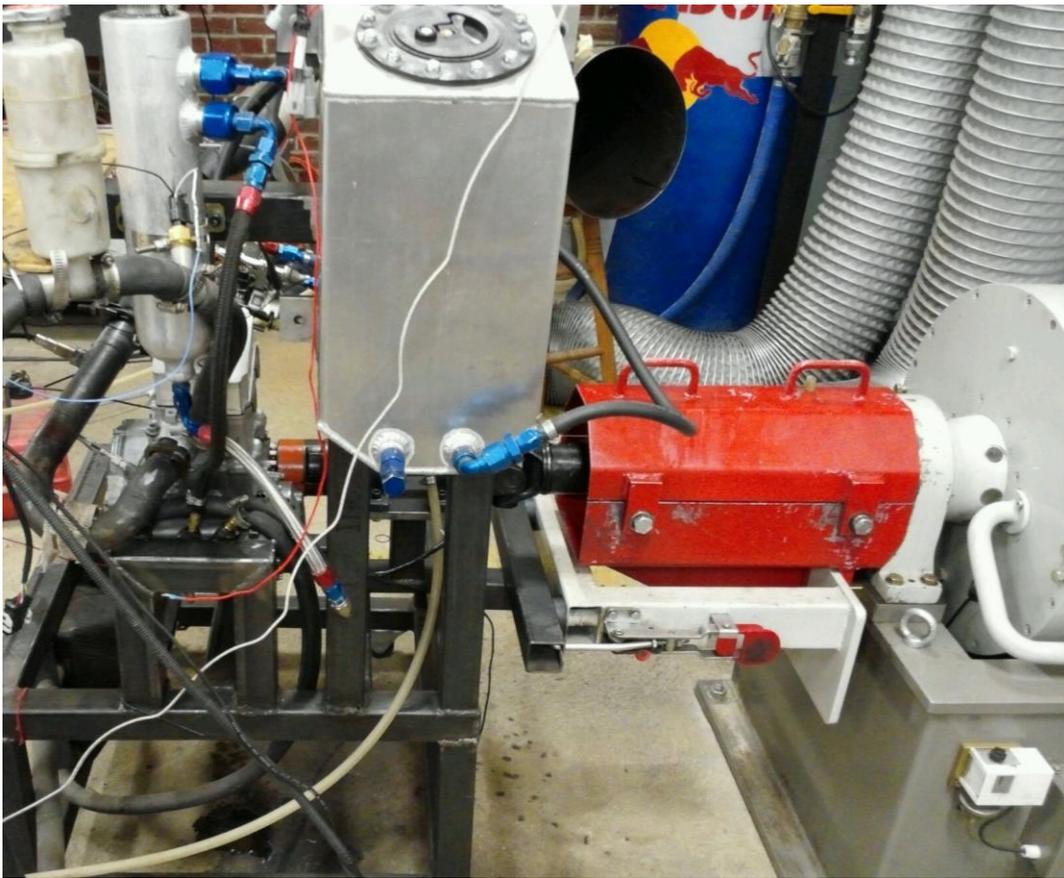


Figure 25. The engine mounted to the eddy-current dynamometer.

### 6.3 Five-Gas Analyzer Setup and Calibration

In gathering emissions data, a Horiba five-gas analyzer was used. Before the gas analyzer was calibrated, several rules with regard to atmospheric and surrounding conditions were followed:

- 1.) The temperature span over which testing could be performed was 0°[C] to 45°[C].
- 2.) The relative humidity during testing was less than 90%.
- 3.) The five-gas analyzer couldn't be exposed to radiative heat sources.

After verifying the aforementioned criteria, the dust, pre-filter, and strainer filter elements were then checked and replaced. The dust and strainer filters were located on the top and back of the Horiba main unit, respectively, while the pre-filter was located in the center of the sampling tube. Upon completing a filter check, the Horiba power was turned on, which was followed by a 300 second warm-up period. During the warm-up period, a leak check was performed to assure adequate sealing. Figures 26-28 show the dust, pre-filter, and strainer filter elements that were replaced.



Figure 26. Dust filter element.



Figure 27. Pre-filter element.



Figure 28. Strainer filter element.

Upon completing a leak test and replacing the filters, the gas analyzer was then calibrated. C<sub>3</sub>H<sub>8</sub>, CO, N<sub>2</sub>, and CO<sub>2</sub> concentrations were specified on the Horiba calibration bottle, while the O<sub>2</sub> concentration was specified based on atmospheric conditions (20.9% by volume). The concentration of each gas was adjusted using the up, down, and side-to-side arrows on the main menu of the Horiba analyzer. Figure 29 shows the specified calibration data on the Horiba calibration bottle.

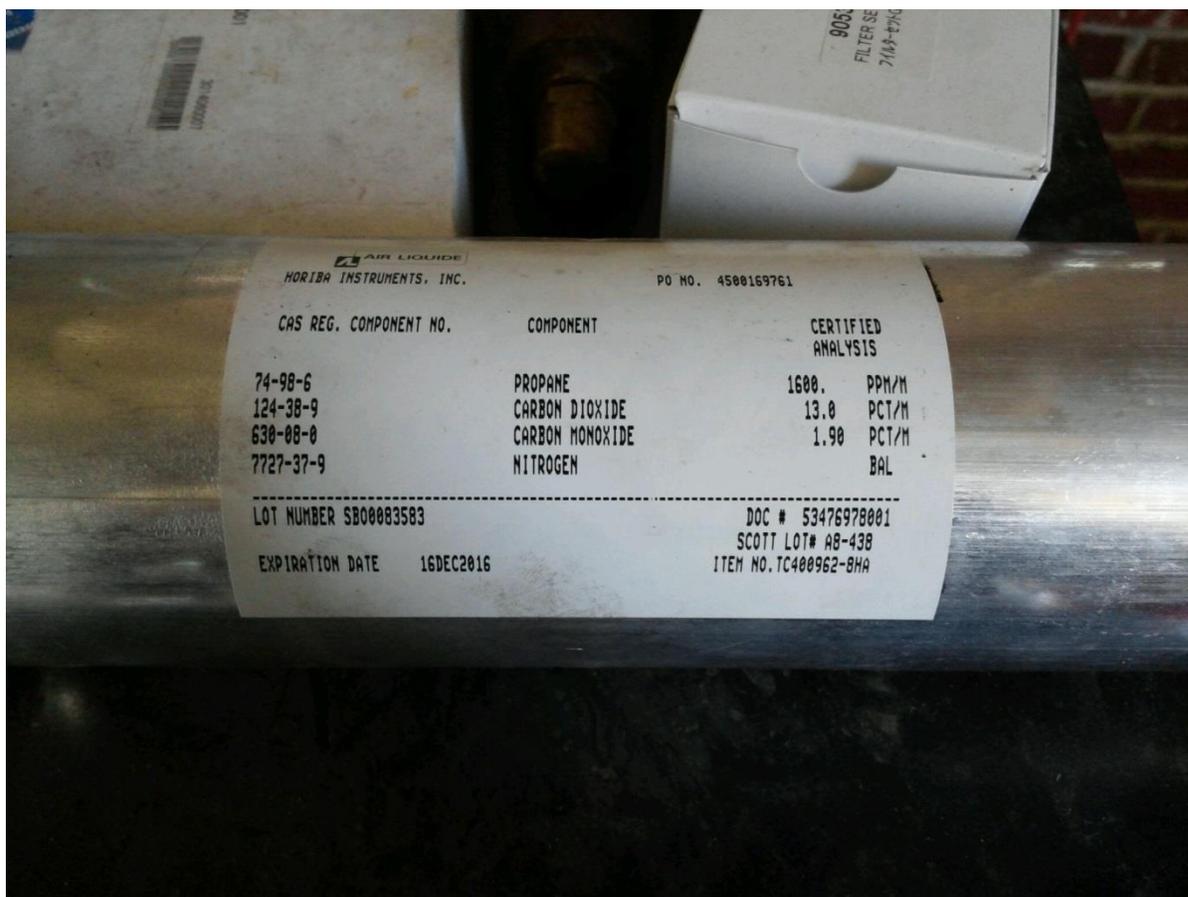


Figure 29. Horiba gas calibration bottle.

After the gases were calibrated, the Horiba sampling tube was placed in the exhaust. A stopwatch was then used to gather data over one-minute intervals, where the data was compiled and averaged in a Microsoft excel worksheet.

## 6.4 Calibrating the ECU

A MoTec M800 ECU was used to adjust fueling and spark parameters in the YZ250F. Since the ECU allowed on-the-fly tuning, optimization of each engine operating point was possible. In order to start the YZ250F, several settings had to be put into MoTec correctly. Table 8 shows these inputs.

**Table 8. MoTec Inputs**

<u>Variable</u>	<u>Angle</u>
CRIPP	293°
Trigger Voltage	1°

With these inputs, tuning the engine required little-more than plugging the ECU into the YZ250F.

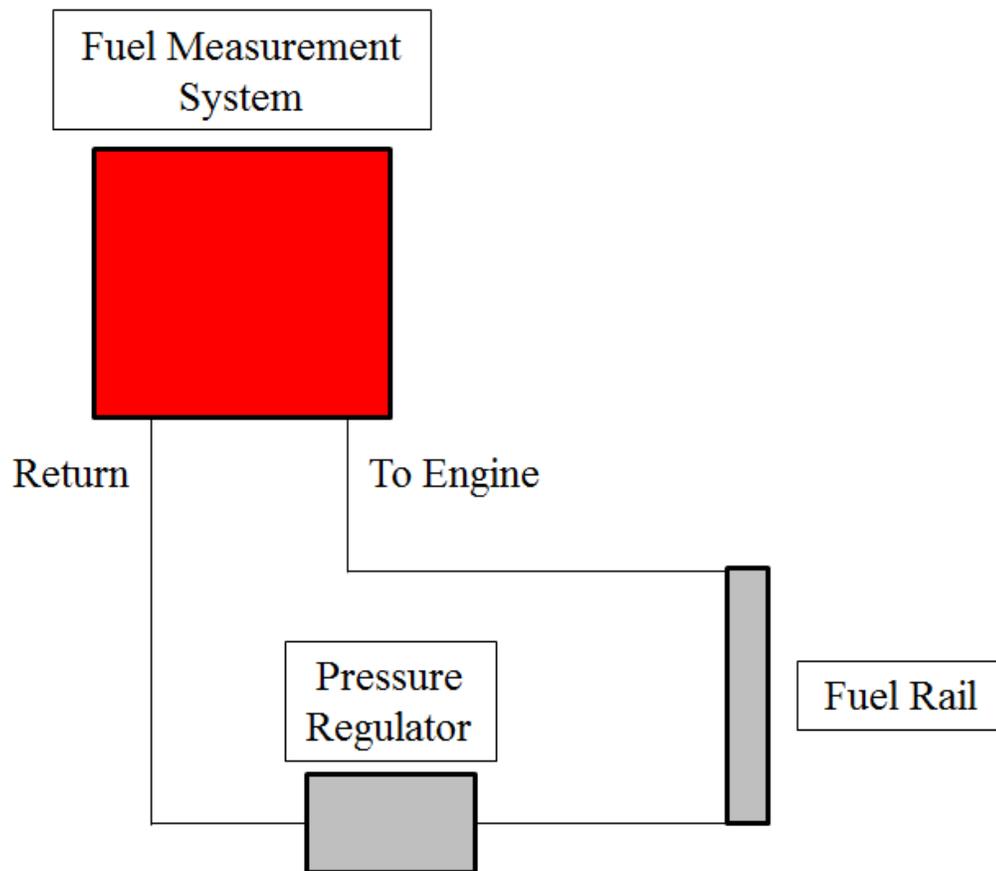
## 6.5 Setting-up the Fuel Measurement System

In order to monitor fuel consumption, a Max 710 series fuel measurement system was used. The measurement system had a built-in fuel pump and tank, and operation of the system required a DC power supply and a timer. In attaching the measurement system to the YZ250F, hoses were routed from the front face of the 710 machine to the inlet and return on the YZ250F's fuel injection system. Figure 30 shows the Max 710 series fuel measurement system and the attached YZ250F.



**Figure 30. Max 710 series fuel measurement system.**

Since the stock YZ250F didn't have a return line on the fuel system, a modified fuel rail was used for testing. Seeing as the modified fuel rail had a return line, it was expected that this would reduce strain on the Max 710 internal fuel pump. It was also found that the return line needed a pressure regulator in order to function correctly. Without a pressure regulator, the modified fuel rail and system failed to pressurize. Figure 31 shows the interactions between the Max fuel system and the YZ250F.



**Figure 31. Diagram of fuel-system.**

The Max 710 series fuel measurement system offered no computer interface or fancy way of timing mass transfer, but instead required the use of a stopwatch. Since the fuel measurement system was timed by hand, a small amount of repeatability error was introduced during experimentation. The equipment uncertainties will be described in the following section.

## **6.6 Equipment Uncertainties**

In referencing the Horiba instruction manual, it was found that the experimental setup had multiple laboratory uncertainties, which varied depending on the gas being measured. In

order to quantify these uncertainties, the root-mean-square (RMS) method was used to combine errors, which was defined as [22]:

$$\mathbf{u}_e = (\mathbf{e}_1^2 + \mathbf{e}_2^2 + \mathbf{e}_k^2)^{\frac{1}{2}} \quad (6.1)$$

where  $\mathbf{u}_e$  was the equipment uncertainty and  $\mathbf{e}_k$  was the equipment error. Since NO and HC emissions measurements were desired, laboratory uncertainties in reference to other chemicals were ignored. However, it was noted that a further chemical analysis would need to include these uncertainties.

In considering the laboratory uncertainties associated with emissions measurements, the Horiba machine precision was examined first. The uncertainty associated with the machine precision was defined as [22]:

$$\mathbf{u}_0 = \pm \left(\frac{1}{2}\right) \text{resolution} \quad (6.2)$$

where the resolution was simply a metric of displayed decimal places. Since the gas analyzer displayed and stored NO and HC emissions on a parts-per-million basis, it was found that the resolution, or interpolation, error was equal to the values displayed in table 9.

**Table 9. Horiba Display Resolution Uncertainty**

<b>Chemical</b>	<b><math>\mathbf{u}_0</math></b>
NO	0.50 [PPM]
HC	0.50 [PPM]

After the machine precision errors were evaluated, other sources of uncertainty were found in the Horiba instruction manual. It was found that the five-gas analyzer had errors associated with drift, linearity, repeatability, warm-up conditions, interference, and voltage fluctuations. Table 10 displays the errors associated with these conditions, and table 11 shows the combined errors, which were found using equation 6.1.

**Table 10. Horiba Equipment Errors**

<b>Source Of Error</b>	<b>NO</b>	<b>HC</b>
Linearity	25 [PPM]	10 [PPM]
Repeatability	N/A	3.3 [PPM]
Warm-Up Conditions	N/A	10 [PPM]
Drift	N/A	10 [PPM]
Interference	N/A	5.0 [PPM]
Voltage Fluctuations	N/A	5.0 [PPM]

**Table 11. Horiba Uncertainty**

<b>Chemical</b>	<b><math>u_e</math></b>
NO	$\pm 25$ [PPM]
HC	$\pm 19$ [PPM]

In measuring engine outputs, a Borghi and Saveri 260-S eddy-current dynamometer was used, which was capable of dissipating up to 191.17 [kW] of power and 610 [N\*m] of torque [23]. In calculating the uncertainty of the eddy-current dynamometer, it was found that the dynamometer strain gage had several uncertainties [24]. Table 12 shows the constituent uncertainties and overall equipment uncertainty.

**Table 12. Borghi and Saveri Eddy-Current Dynamometer Uncertainties[24]**

<b>Source Of Error</b>	<b><math>u_e</math></b>
Linearity	0.050[%]
Repeatability	0.020[%]
Hysteresis	0.030[%]
Overall	$\pm 0.062$ [%]

The uncertainty of the fuel measurement system was specified in the instruction manual. It was found that the accuracy of the Max 710 series fuel measurement system was within 0.75% over normal temperature ranges. However, it was observed that care had to be taken in measuring fuel consumption with regard to the temperature of the fuel. It was found that rising fuel temperatures could introduce errors during testing.

In calculating the resolution error of the fuel measurement system, it was found that the system was capable of displaying three decimal places. Since the system displayed in kilo-grams, the resolution had to be converted to a percentage of uncertainty. Preliminary

testing [21] showed that the minimum amount of fuel consumed under normal testing conditions was on the scale of  $1.2 \left(\frac{\text{kg}}{\text{hr}}\right)$  ( $\sim 0.02 \left(\frac{\text{kg}}{\text{min}}\right)$ ). Therefore, a maximum precision uncertainty was associated with this measurement (percentage-wise, the maximum precision uncertainty was associated with the minimum amount of fuel consumed over a set time-interval). Since one-minute intervals were used during testing, the machine precision uncertainty ( $\pm 0.0005$  (kg)) was divided by the minimum amount of fuel consumed ( $\sim 0.02 \left(\frac{\text{kg}}{\text{min}}\right)$ ) to determine the maximum amount of error due to machine precision. Table 13 shows the uncertainty in the fuel measurement system.

**Table 13. Fuel Measurement System Equipment Uncertainties**

<b>Source Of Uncertainty</b>	<b><math>u_e</math></b>
Resolution	2.5 [%]
Equipment	0.75 [%]
Overall	$\pm 2.61$ [%]

## Chapter 7. Engine Testing

Engine testing was performed in the University of Idaho's Small Engines Research facility (SmERF), and a Windyn data acquisition system was used to collect data at different engine operating points. Before testing began, previously collected data on the YZ250F was evaluated, and a test-plan was developed based on these findings. After a test-plan was developed, fuel-consumption and emissions data points were collected and compared to outputs generated in the MATLAB model. The following sections will describe the experimental test plan and results obtained from testing the YZ250F.

### 7.1 Test Plan

Based on previous results [21], it was decided that many data points would be necessary to effectively resolve BSFC maps on the YZ250F. In previous testing [21], approximately 3-4 loads per RPM-interval (every 1000 RPM) were monitored over 30-second time-intervals. In order to refine these results, it was decided that 7-8 loads per RPM-interval (every 500 RPM) would be monitored over one-minute time-intervals. In previous testing [21], it was also found that human errors were introduced through on-screen torque readings. It was found that these errors could be minimized by utilizing a "steady-state" test and saving torque values in the Windyn program; thus, the variables could then be time-averaged. Table 14 shows the changes that were made in gathering engine data.

Table 14. Refining BSFC Results

<u>Variable</u>	<u>Previous Testing</u>	<u>Current Test</u>
Loads	3-4 [Per RPM Interval]	7-8 [Per RPM Inteval]
RPM-Inteval	Every 1000 RPM	Every 500 RPM
Torque	Read From Screen	Saved In Windyn
Time-Interval	30-second	1-minute

## 7.2 Test Results

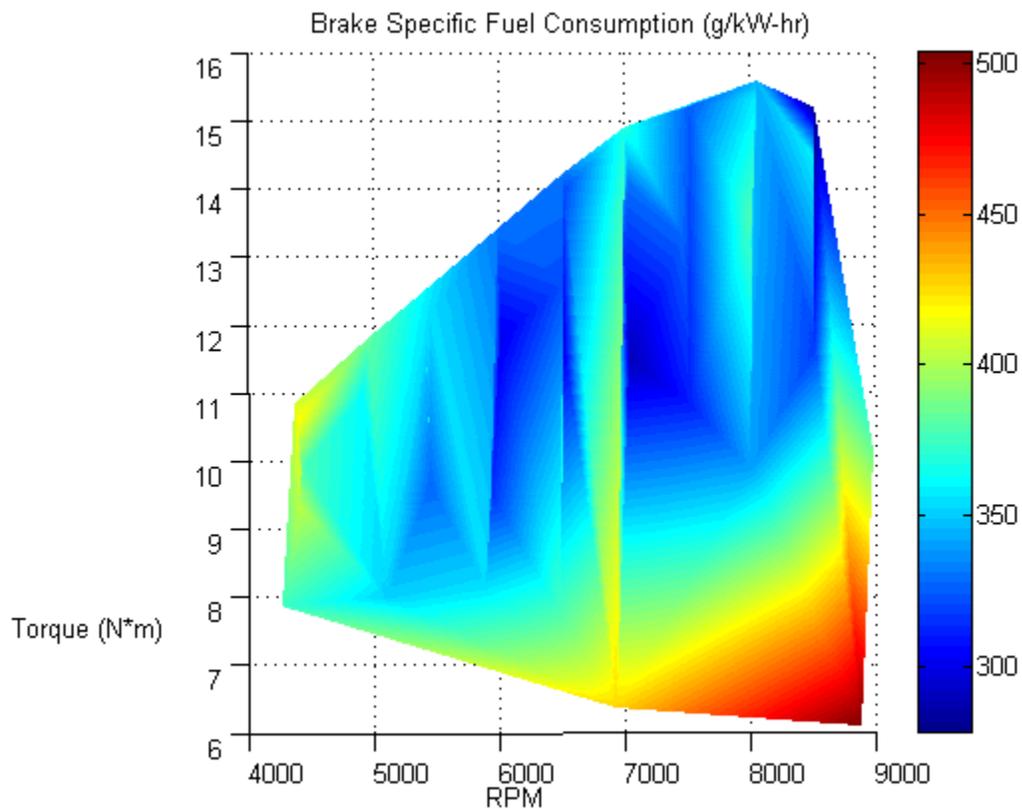
After a test plan was created, Windyn and the eddy-current dynamometer were used to acquire experimental data. During testing, the MoTec ECU was adjusted on-the-fly to meet a  $\lambda$  reading of approximately 0.90, where  $\lambda$  was adjusted based on the spark-advance and injector pulse-width percentage (this adjusted the percentage of fuel injected). Initially, the pulse-width percentage was adjusted to accommodate the desired air-fuel ratio, and the level of vibration associated with the test-engine and corresponding map settings was analyzed. If the test-engine vibrated substantially, the spark-advance was then adjusted to even-out the vibration. When the vibration subsided, the spark-advance was then adjusted in small increments in order to optimize torque.

After adjusting the operating parameters to meet the desired  $\lambda$  reading, the Windyn “steady-state” program was used at a set throttle position. This allowed the dynamometer to hold the engine at a set speed, over a one-minute span, while gathering data points every second. While the “steady-state” program was running, start-stop switches on the fuel measurement system were used to acquire the total amount of fuel used over a one-minute interval. The total amount of consumed fuel and gathered data were then time-averaged and stored in an Excel file for later use. Table 15 displays the gathered data.

Table 15. Experimental Data

Avg. RPM	Power (kW)	Torque (N*m)	TP(%)	lambda	BSFC(g/kW-h)
4279	3.52	7.86	10.0	0.87	374.60
4406	4.32	9.36	22.0	0.86	403.10
4392	4.86	10.58	32.0	0.86	419.41
4369	5.00	10.85	39.5	0.88	408.31
5099	4.27	8.00	13.5	0.9	351.17
4441	4.60	9.90	21.5	0.92	378.01
4910	5.65	10.98	29.5	0.88	361.26
4958	6.11	11.80	55.5	0.83	382.68
5452	5.50	9.63	22.0	0.9	327.51
5424	6.41	11.25	32.0	0.92	346.17
5443	6.64	11.66	39.5	0.92	352.58
5448	6.94	12.20	50.5	0.91	354.72
5432	6.56	11.52	60.5	0.88	384.02
5899	5.07	8.13	16.5	0.90	354.98
5948	5.89	9.49	21.5	0.88	315.73
6014	7.38	11.80	30.5	0.92	300.71
6005	7.83	12.47	38.5	0.93	314.18
5993	8.35	13.42	53.5	0.92	330.47
6493	5.22	7.73	16.5	0.90	379.32
6519	7.98	11.66	29.0	0.93	300.79
6493	8.58	12.61	38.5	0.92	321.85
6520	9.54	13.96	52.5	0.91	320.59
6515	9.77	14.24	63.0	0.88	331.67
6936	4.62	6.37	14.0	0.88	428.26
7043	8.50	11.52	27.0	0.92	289.38
6996	9.62	13.15	42.0	0.94	318.10
7022	10.37	14.10	53.0	0.90	329.95
7009	10.89	14.91	65.5	0.88	368.44
7522	8.50	10.85	24.5	0.91	310.55
7473	10.44	13.29	41.5	0.88	327.59
7516	12.01	15.19	58.0	0.90	324.84
7511	11.63	14.78	50.5	0.90	324.94
8004	8.28	9.90	24.0	0.88	340.69
8026	10.74	12.74	35.5	0.90	346.43
8003	11.41	13.56	47.0	0.88	378.64
8039	12.90	15.32	60.0	0.88	358.12
8052	13.20	15.59	70.0	0.89	345.48
8524	10.37	11.66	34.0	0.90	318.37
8520	13.57	15.19	49.0	0.90	278.52
8500	12.90	14.51	63.0	0.89	358.12
8891	5.59	6.10	16.0	0.91	504.22
8991	9.54	10.17	32.0	0.90	383.45

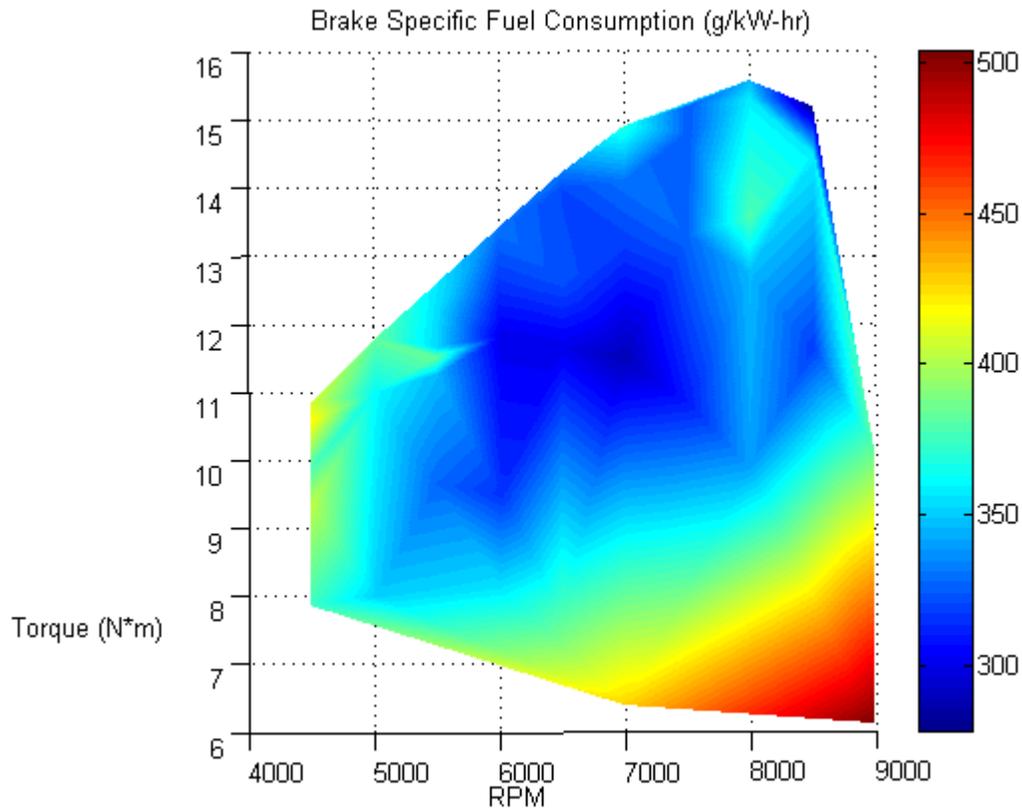
During testing, the oil pump seized and caused catastrophic engine failure, so a complete set of operating points was not obtained. Although the raw data was limited to an engine speed range of 4500-9000 RPM, it was found that the increased number of data points and longer time-span significantly resolved the obtained data. Over the tested range, it was found that the minimum BSFC was approximately  $280\left(\frac{\text{g}}{\text{kW-h}}\right)$ , and it was found that the torque and BSFC curves responded as expected until the oil pump failed. Figure 32 shows the BSFC map from the data displayed in table 15.



**Figure 32. BSFC map over data in table 14.**

In evaluating the BSFC map, it was noticed that the trough of the plot was at the top, right corner; although, more data points would be necessary to fully resolve ideal operating conditions on the YZ250F. In further studying the BSFC map, it was observed that the map showed several bands of discontinuity. It was found that these bands occurred in areas that

lacked sufficient data to linearly interpolate in the MATLAB program (i.e. the band of discontinuity around 7000 RPM lacked data points at low-load operation). In order to refine the discontinuous bands, the target RPM at each interval (instead of the average RPM) was used to create figure 33.



**Figure 33. Refined BSFC map.**

It was found that the target RPM at each interval produced much more desirable results for BSFC mapping. It was expected that the average RPM could be used in a map containing many points; however, the limited data that was gathered in this thesis was insufficient in terms of resolution.

After a BSFC map was created, outputs generated with the MATLAB model were compared to the experimental data. Since the model had trouble predicting low-load outputs,

it was decided that 50%-70% throttle-position data points would be used in matching the model. Table 16 shows a comparison of the chosen data points with the model outputs.

**Table 16. Comparing Model Outputs with Experimental Data**

<b>Source</b>	<b>TP (%)</b>	<b>lambda</b>	<b>RPM</b>	<b>Power (kW)</b>	<b>Relative Error (%)</b>	<b>Torque (N*m)</b>	<b>Relative Error (%)</b>	<b>BSFC [g/kW-h]</b>	<b>Relative Error (%)</b>
Experimental	<b>63</b>	<b>0.90</b>	<b>8500</b>	<b>12.90</b>	<b>2.09</b>	<b>14.51</b>	<b>7.51</b>	<b>358.12</b>	<b>3.80</b>
Model				<b>12.63</b>		<b>13.42</b>		<b>344.52</b>	
Experimental	<b>70</b>	<b>0.89</b>	<b>8052</b>	<b>13.20</b>	<b>0.15</b>	<b>15.59</b>	<b>0.58</b>	<b>345.48</b>	<b>3.28</b>
Model				<b>13.22</b>		<b>15.68</b>		<b>334.14</b>	
Experimental	<b>65.5</b>	<b>0.88</b>	<b>7009</b>	<b>10.89</b>	<b>2.11</b>	<b>14.91</b>	<b>2.62</b>	<b>368.44</b>	<b>6.30</b>
Model				<b>10.66</b>		<b>14.52</b>		<b>345.22</b>	
Experimental	<b>60.5</b>	<b>0.88</b>	<b>5432</b>	<b>6.56</b>	<b>0</b>	<b>11.52</b>	<b>0.09</b>	<b>384.02</b>	<b>5.99</b>
Model				<b>6.56</b>		<b>11.53</b>		<b>407.04</b>	
Experimental	<b>55.5</b>	<b>0.83</b>	<b>4958</b>	<b>6.11</b>	<b>2.29</b>	<b>11.80</b>	<b>2.63</b>	<b>382.68</b>	<b>8.61</b>
Model				<b>5.97</b>		<b>11.49</b>		<b>415.61</b>	

It was found that the model accurately predicted outputs at operating points above 50% throttle, and of the compared data points, the maximum relative error was 8.61%. It was expected that low-load outputs could be greatly refined by creating a volumetric efficiency map and relating this to the numerical model; however, it was also realized that most low-load operating points would be unnecessary to predict, because engines rarely operate under these conditions.

Since the test-engine was out-of-commission, emissions data from a Yamaha R6 were compared to model outputs. Only NO predictions were compared, because the amount of consumed fuel during testing was unknown, and the model-predicted HC emissions were on a percentage basis. Table 17 shows the experimental data that was gathered on the Yamaha R6.

**Table 17. Yamaha R6 Experimental Data**

<b>RPM</b>	<b>TP (%)</b>	<b>Lambda</b>	<b>NO (PPM)</b>
3500	0	0.925	88.41
3500	10	0.925	441.23
3500	20	0.925	481.82
3500	30	0.925	600.69

In studying the R6 data points, it was found that the NO emissions drastically jumped between idle and 10% throttle. Since the 0% throttle (idle) point couldn't be predicted by the model, this wasn't further analyzed. It was also observed that the NO emissions jumped between 20% and 30% throttle. It was hypothesized that the 30% throttle data point was skewed due to ignition problems in the engine, thus causing excess heat and NO readings (the cylinder head and piston were destroyed during testing). A further analysis of the second-by-second data also showed an extreme jump from 544(PPM) to 644(PPM) during this test period. Table 18 shows a comparison of measured and predicted NO data.

**Table 18. Comparing Emissions Predictions to Experimental Data**

<b>RPM</b>	<b>TP (%)</b>	<b>Measured NO (PPM)</b>	<b>Predicted NO (PPM)</b>	<b>Relative Error (%)</b>
3500	10	441.23	441.65	0.10
3500	20	481.82	469.97	2.46
3500	30	600.69	494.39	17.70

A comparison of the predicted and measured data showed a significant difference in NO at 30% throttle. Much of this was attributed to the engine failure and skewed emissions during testing; however, it was also observed that the model was highly sensitive to the point at which frozen exhaust composition was assumed. It was found that a frozen composition at 90% of the peak burned-zone temperature matched the experimental data. Aside from the skewed data point, the rest of the results were very close to what the model predicted. Although more data was necessary to fully validate the emissions model, the preliminary results were promising.

## Chapter 8. ME 433 Learning Activity

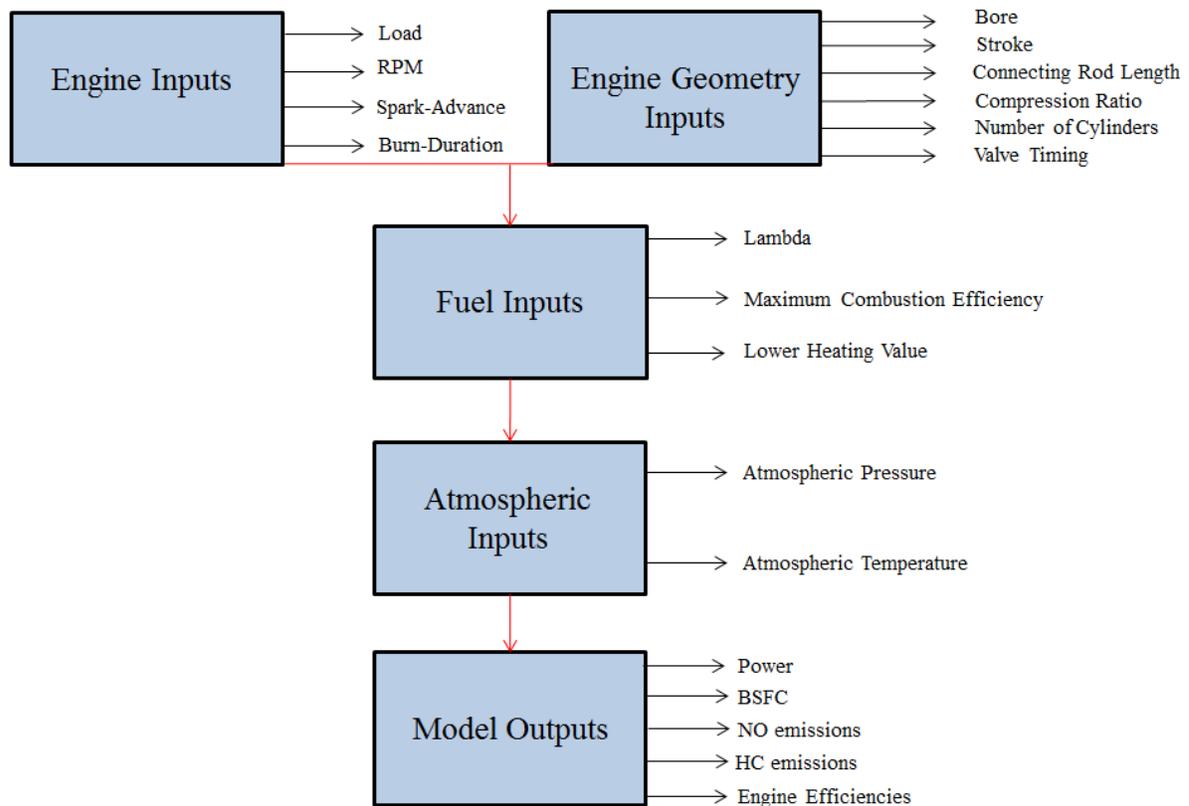
The ME 433 Combustion Engine Systems course concludes with a mini-project intended to bring together engine-sizing parameters, in-cylinder heat release, and prediction of exhaust emissions. The model developed in this thesis is ideally suited for use as a learning tool in this section of the course. Special features of the model which can promote deeper understanding of in-cylinder thermo-physical processes include:

- 1.) The impact of finite fuel heat release on laminar flame speed.
- 2.) The influence of combustion parameters on in-cylinder pressure.
- 3.) The effect of volumetric and combustion efficiencies.
- 4.) The changes in exhaust chemistry at different operating points.

The subsequent sections will discuss the formulation of a class project aimed at meeting these objectives and how this project was performed using the MATLAB model.

### 8.1 Guided Exploration of the Model

An important part of using this model is by understanding its assumptions and functionality. In preparation for the class-project, students will explore the MATLAB model. They will find the blocks of code in which engine, fuel, and atmospheric inputs will be placed. Students will be urged to explore the relationships between variables and the governing equations used in the model (by clicking on a variable in the MATLAB script, all uses of the variable are highlighted on-screen). Figure 34 will be used as a tool to show students the inputs and outputs of the model.



**Figure 34. Flow-chart of the MATLAB model's inputs and outputs.**

In exploring engine inputs, students will discover the purpose of “load”, “spark-advance”, and “valve-timing” in the model. The following questions will be addressed in guiding students through the model.

- 1.) How does the engine load affect model outputs?
- 2.) How does the spark-advance influence the pressure-trace?
- 3.) How does valve-overlap change model predictions?

Students will find that the engine load alters the inlet pressure, thus reducing peak pressure and temperature predictions. They will find that engine load affects every model output, and signifies throttle position. In analyzing spark-timing, students will find that delayed timing will cause secondary-peaks in the pressure-trace. They will also find that the peak pressure and model outputs will be affected substantially. In experimenting with valve-

timing, students will find that valve-overlap directly influences model outputs, because it delays pressure from building. Students will need to consider this in the class project addressed in the following section.

After exploring basic engine inputs, students will focus on atmospheric and fuel inputs. Students will alter the air-fuel ratio by changing “lambda”, and will observe the changes in outputs due to changing atmospheric conditions. The following questions will be used in guiding students through fuel and atmospheric inputs.

- 1.) How does operating fuel-rich as opposed to fuel-lean affect torque?
- 2.) Do atmospheric conditions change model outputs?

In experimenting with the air-fuel ratio, students will find that fuel-rich operating conditions produce more power; however, they will also find that this affects emissions. Students will also find that atmospheric conditions can change model outputs significantly. As atmospheric pressure is reduced, air within the engine becomes less dense, thus reducing the mass of combusted air and fuel.

## **8.2 Mini-Project Description**

After developing an understanding of the MATLAB model, students will select and simulate a 4-stroke, SI engine. Students will focus on naturally aspirated, multi-cylinder engines to avoid the under-prediction of heat transfer. Upon selecting an engine, students will search engine manufacturer’s websites for engine parameters such as the bore, stroke, and connecting rod length, and will put these into the MATLAB model. Aside from inputting geometric constraints, students will also have to modify the MATLAB code to include EGR (if necessary), and the FMEP losses will have to be modified based on the type of engine bearings used. In selecting an engine, it will be suggested that students choose

something that can be validated. Although the MATLAB model predicts accurately, adjustments often need to be made with regard to heat transfer surface area, volumetric efficiency, and spark-timing. Students will adjust these variables and efficiencies based on an expected torque or power curve.

Students should be encouraged to research SAE articles and theses from the University of Idaho. However, Heywood [3] and Blair's [5] books should be sufficient in modeling an engine of their choosing. "Internal Combustion Engine Fundamentals" (pp. 393-395) should be used in the optimization of spark-timing, because it provides excellent insight towards burn durations and timing patterns in reference to engine speed. "Design and Simulation of Four-Stroke Engines" (pp. 534-537) should be used in predicting FMEP losses on different engines, because it provides linear equations for a number of applications.

In meeting the project objectives, students will simulate the engine of their choosing to produce:

1. Optimized spark-timing plots.
2. BSFC maps.
3. Emissions formation maps.

The following sections will describe the suggested procedure by which students should construct these plots, and the physical phenomena that students should be aware of when simulating engine performance. Each section contains results obtained using the YZ250F.

### **8.3 Optimizing Spark-Timing**

Students will be exposed to the effect of laminar flame speed on spark-timing and the change in burn duration relative to engine speed through an optimized spark-timing activity. The optimal spark-timing will be found by varying the spark-advance and observing changes

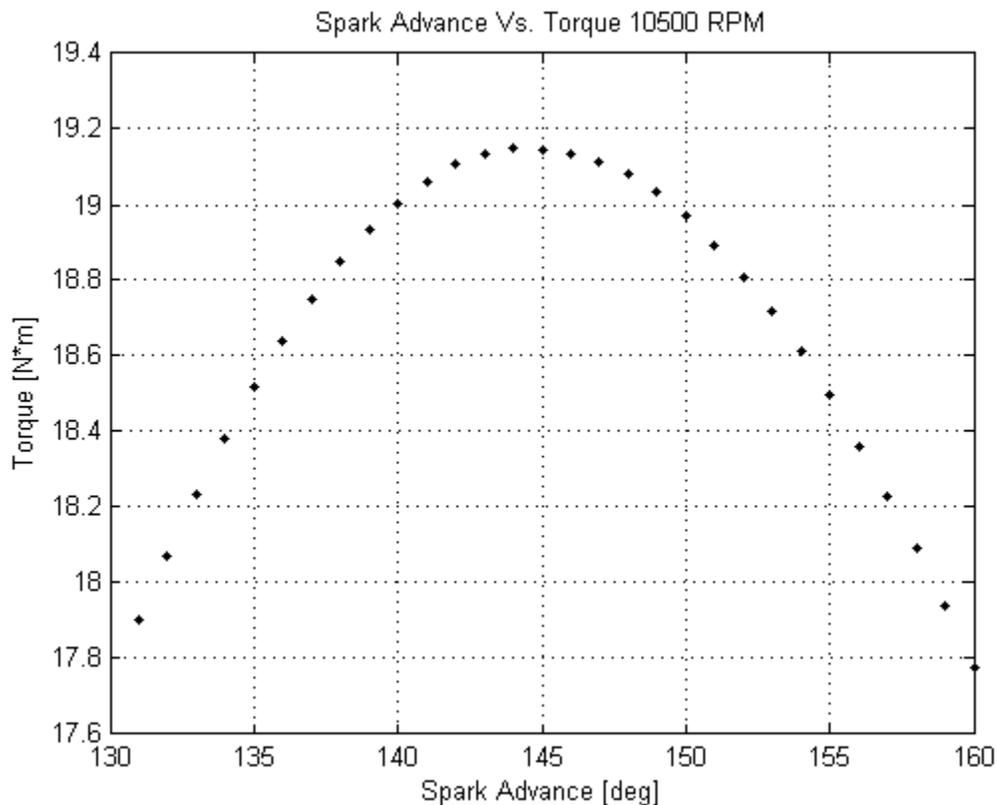
in predicted torque and power. This section will describe suggested activities for students to observe changes in timing relative to engine speed and air-fuel ratio.

Students will optimize spark-timing through the use of a call function. The call-function will be placed around the body of the primary MATLAB script, with the input being the spark advance, and the outputs being brake-power and torque. The call-function will then specify spark-advances and keep track of the resulting power and torque. Figure 35 shows an example loop inside of a call function that specifies spark-advances.

```
for i=1:(theta_fin-(theta_st))
    [W_dot_ac(i),T_ac(i)]=timingfunc(theta_0);
    theta_0=theta_0+1;
    theta_o(i)=theta_0;
end
```

**Figure 35. Spark-timing loop.**

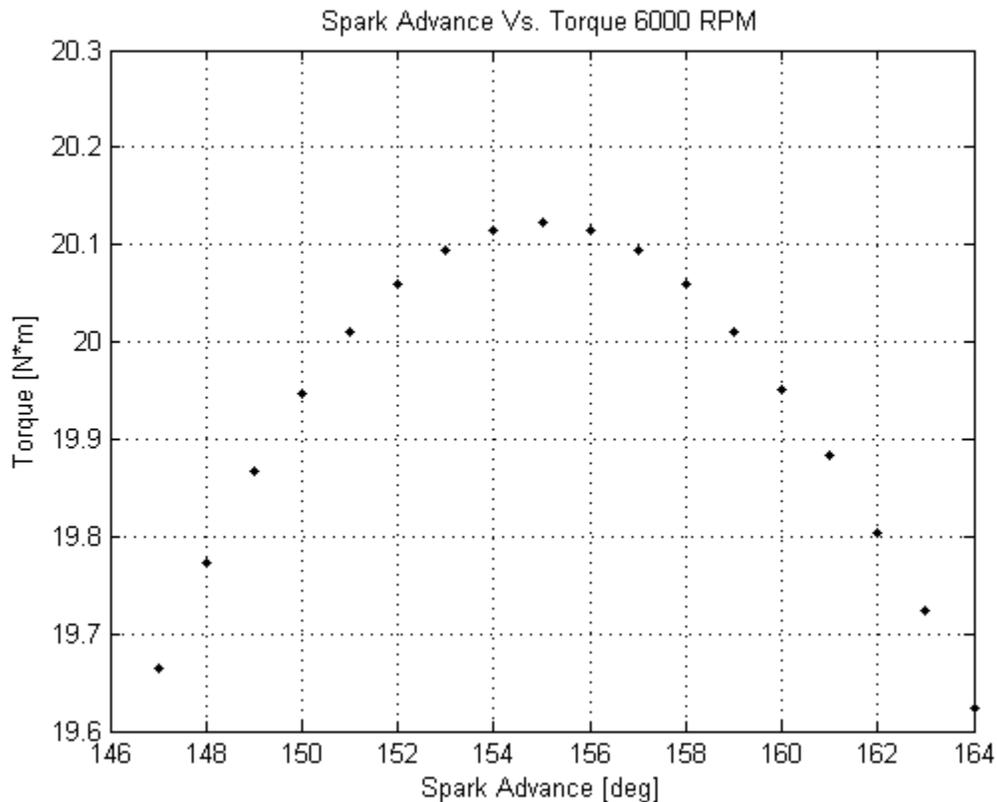
Torque and power predictions will then be stored in an array, where they can be easily accessed for plotting. Grid-lines and data points will be used in the plot statements so-as to simplify the process of reading peak-timing from a plot. Figure 36 shows an example of the peak torque as a function of spark-advance, where a 90% load,  $\lambda$  input of 0.90, and engine speed of 10,500 RPM was used.



**Figure 36. Spark-advance curve.**

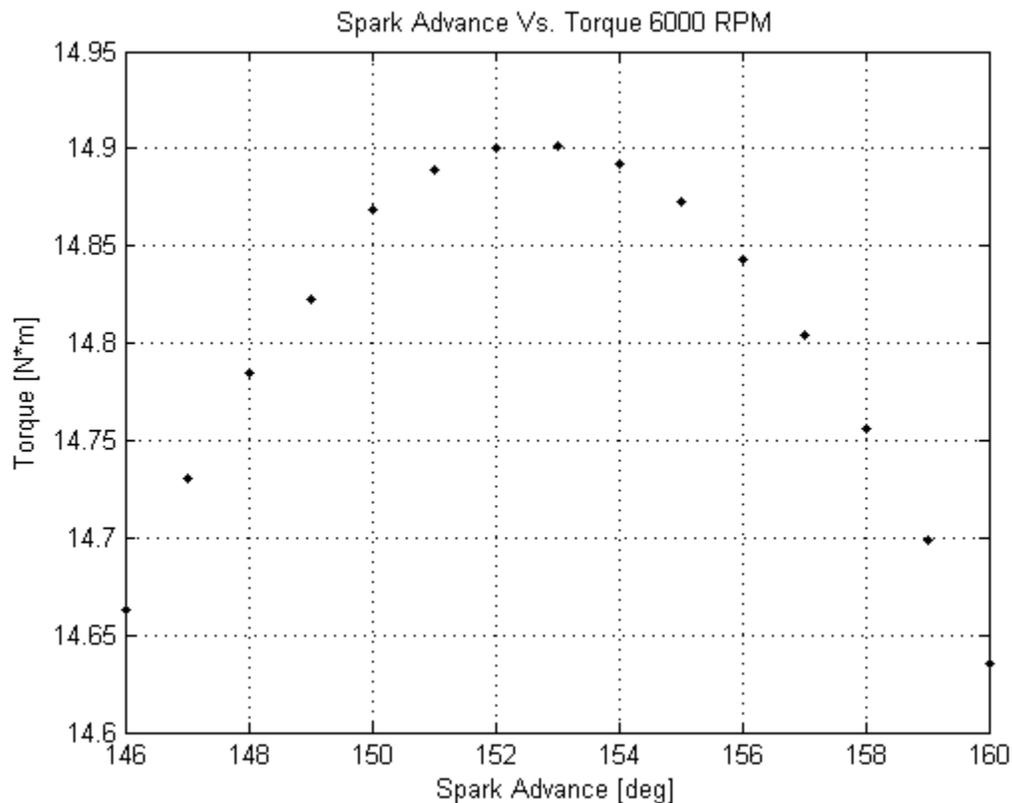
For the purpose of an in-class project, students will need to make assumptions about the burn duration relative to the laminar flame speed and engine speed. Students may also consider the idea of creating a model to calculate the laminar flame speed and burn duration at different operating points; a process by which this can be calculated can be found in Heywood's [3] book. Students should be encouraged to analyze predicted peak cylinder pressures and temperatures, and the effect of the burn duration on these predictions; in this regard, students can tailor the model to known cylinder pressures and temperatures. Students should also find that the burn duration increases from low to high engine speeds, and from fuel-rich to fuel-lean operating conditions (i.e. the laminar flame speed is the fastest in reference to a  $\lambda$  value of approximately 0.9 [3] in iso-octane combustion). Figure 37 displays

an example of the change in spark advance at a lower engine speed, where the burn duration was changed from  $85^\circ$  at 10,500 RPM to  $65^\circ$  at 6,000 RPM, a load of 90% was used, and  $\lambda$  input of 0.90 was used.



**Figure 37. Optimal spark-advance at 6,000 RPM.**

As was explained previously, students should inspect the burn duration in modeling changes in spark advance relative to changes in the air-fuel ratio. By increasing the burn duration relative to a leaner mixture, students should be able to obtain the change in spark-advance relative to a change in the air-fuel ratio. Figure 36 shows an example spark-advance plot at an engine speed of 6,000 RPM, a  $\lambda$  input of 1.1, and burn duration of  $70^\circ$  (this is the same engine speed as was used in figure 38, with a different air-fuel ratio and corresponding burn duration).



**Figure 38. Optimal spark-advance relative to a change in air-fuel ratio.**

Students should find that the change in air-fuel ratio changes the optimal spark-advance by a few degrees, and that this is directly correlated with the change in burn duration.

In conclusion, students should find that the optimal spark-timing is strongly influenced by engine speed, load, and the burn duration. However, they should also realize that the air-fuel ratio cannot be completely detached from the aforementioned operating parameters (i.e. the ECU would provide rich fueling under full loads). Appendix O is a PowerPoint that should be used in guiding students through the spark-advance activities.

#### **8.4 Generation of BSFC Maps**

Students will be exposed to volumetric efficiencies and the shaping of torque profiles and combustion efficiencies relative to engine operating points in a BSFC mapping activity.

In constructing BSFC maps, students should utilize several secondary functions that call the primary MATLAB model. A function with torque and BSFC outputs, and RPM and load inputs, should be placed around the main MATLAB script. A new function should then be constructed that provides the main script with an array of RPM inputs. Lastly, a call-function will be created that feeds a matrix of loads into the previously described functions and runs each simulation. Figure 39 shows a call function that was created to simulate BSFC mapping on the YZ250F.

```

for j = 1:N_load
    Load = Load-.1; %Decreases Load With Each Iteration
    [BSFC,T_ac,N_RPM]=LOAD(Load); %Feeds "Load" Into Load Function
    BSFC=BSFC'; %Transposes Vector
    bsfc(1:N_RPM,j)=BSFC;%Creates Matrix Out of BSFC Vectors
    T_ac=T_ac'; %Transposes Torque Vector
    Torque(1:N_RPM,j)=T_ac'; %Creates Matrix Out of Torque Vectors
end

```

**Figure 39. The BSFC call function.**

In creating BSFC maps, students should be aware of the changes in spark-advance, air-fuel ratios, burn durations, and volumetric efficiencies in a real engine cycle. If these parameters aren't fluctuated correctly, the output BSFC maps won't contain closed contours, and the values won't be correct. Students should start the BSFC mapping activity by creating a linear equation to relate the burn duration to engine speed at a set load. At part load, the spark-advance and burn duration should be related to the full load prediction described previously. Mathematical equations or MATLAB logic can be used to relate these variables.

In modeling the air-fuel ratio, students should use if-then statements to specify fuel-rich and fuel-lean constraints; where the air-fuel ratio should be controlled through the excess air coefficient ( $\lambda$ ). Students may have to research stock ECU maps to correctly configure

their model. However, students should use linear relationships in all cases, because complex equations can be difficult to resolve with the provided number of data points in the model.

Figure 40 displays an example equation that was used to model spark-advance as a function of RPM and load.

```
if Load==1
theta_0=ceil(-.0019*RPM+168.19);
end
if Load<1
theta_0=ceil(-.0019*RPM+168.19)-(10-Load*10);
end
```

**Figure 40. Modeling spark -advance.**

Upon running a simulation with the call function, torque, BSFC, and RPM values will be stored in matrices. The number of RPM and load data points will be specified in the aforementioned functions and the matrices will be sized based on the number of data points (the simulation shown used 12 engine speeds and 7 loads, so the corresponding matrices were sized 12 x 7).

With torque, BSFC, and RPM outputs, the BSFC map will be constructed. The BSFC call function will use the “contourf” and “colorbar” commands to produce a filled, contour plot and corresponding colored key. The “caxis” command will then be used to resolve minimum and maximum contour values. Since the plot will cover a broad range of BSFC values, the colored contours should be restricted using the “caxis” command, accordingly. It will be found that loads of less than 30-40% can disrupt the flow of the plot, because torque values at high and low engine speeds will fall near, or below, zero.

Therefore, engine loads should be restricted to 40% (approximately) and above. Figure 41 shows the plot statements that were used in the example BSFC call function.

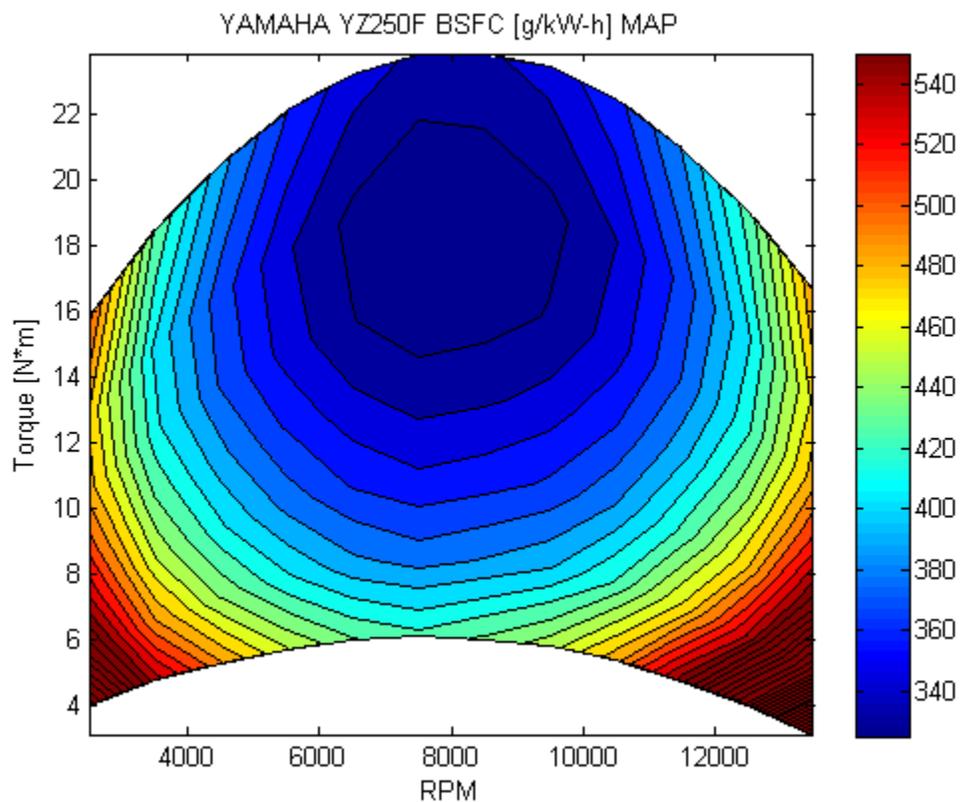
```

%Plot Statements
figure(1)
contourf(RPM,Torque,bsfc,100)
xlabel('RPM')
ylabel('Torque [N*m]')
title('YAMAHA R6 BSFC MAP')
colorbar
colormap jet
caxis([300 425])

```

Figure 41. BSFC plot statements.

The contour plot will take a bit of tailoring to show optimal engine operating points. By using load increments of .1, students will be able to sufficiently resolve peak engine operating conditions. However, using load increments of less than .1 will cause numerical issues within the main script, because engine parameters will be calculated on a one-degree crank angle basis, so the program will attempt to access data points that don't exist. Figure 42 shows an example BSFC map produced in MATLAB.



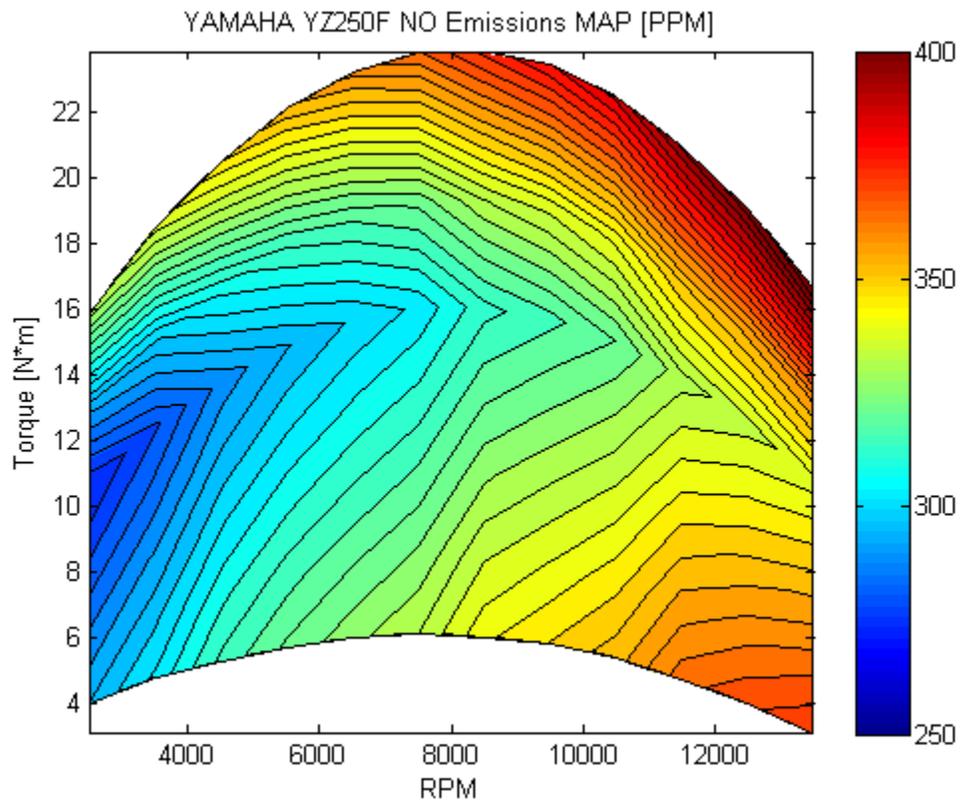
**Figure 42. BSFC map produced with MATLAB.**

Students will be able to obtain a similar plot by assuming volumetric efficiencies, spark-timing as a function of speed and load, and burn durations as a function on operating points. They should tailor the plot to match obtained torque curves, and BSFC values should be scrutinized based on known ranges and the chosen type of engine. Appendix P should be used as a class handout for students in creating BSFC maps in MATLAB.

### **8.5 Estimating Engine Emissions**

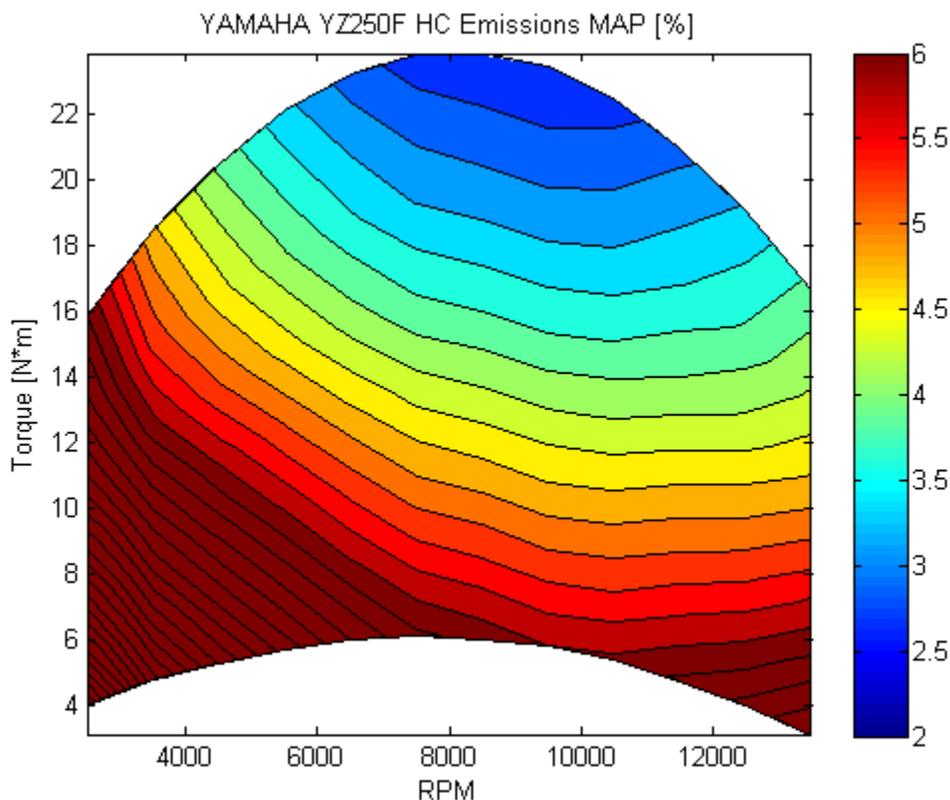
Students will be introduced to emissions formation at different operating points by constructing emissions maps and emissions plots at different air-fuel ratios. Studying NO and HC emissions formation will begin with the development of brake-specific emissions formation (BSEF) maps, since it was found that this was the easiest way to examine formation mechanisms with respect to normal engine operation.

In creating BSEF maps, the primary BSFC function, and BSFC call functions (described in section 8.3) will be used to produce additional NO and HC emissions outputs, where the function surrounding the main MATLAB model will be modified to include these outputs. Once again, matrices will be created within the BSFC call function to store NO and HC outputs as a function of engine speed and load. With stored values at different operating points, the NO and HC emissions maps will then be created using the “contourf” MATLAB command. Figure 43 shows an example NO emissions map.



**Figure 43. NO emissions map for the YZ250F.**

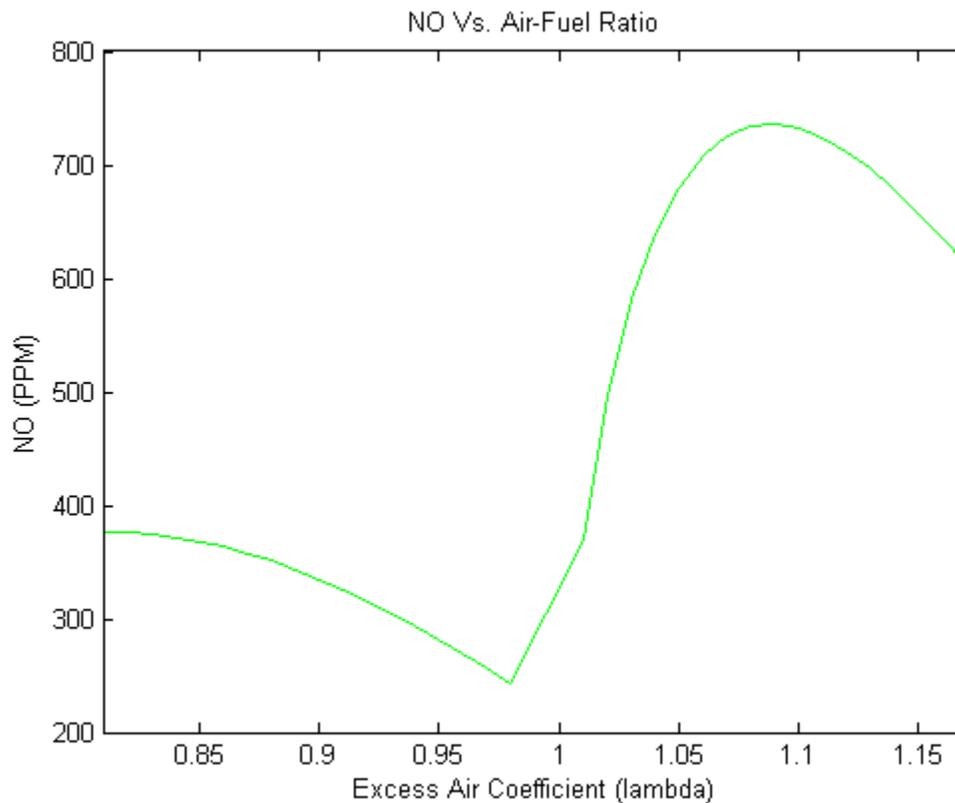
In studying the NO emissions map, it should be found that the highest predicted emissions are associated with full-load engine operation. Students should be able to apply this idea to the fact that NO formation is temperature-dependent, and would thus be produced at the highest levels at peak operating points. At full-load, it should be observed that the engine would attain peak cylinder pressures, and consequently, peak cylinder temperatures. Figure 44 shows an example of the predicted HC emissions at different engine operating points.



**Figure 44. HC emissions map for the YZ250F.**

Students should observe that the HC emissions map clearly defines operating points to avoid when trying to reduce emissions. The low-load, low engine speed region of the map will show low percentages of post-combustion HC burn up, while the high-load, high engine speed region of the map will show high percentages of post-combustion burn up. Students should be able to relate the fact that peak operating points produce the hottest combustion (and exhaust) temperatures, and thus, the lowest HC emissions.

After creating BSEF maps, students will conduct a study in which NO emissions will be compared across different air-fuel ratios, at a set RPM and load. The  $\lambda$  input will be varied from fuel-rich to fuel-lean, and the predicted NO emissions will be documented. Figure 45 shows an example of predicted NO emissions as a function of  $\lambda$ .



**Figure 45. NO emissions at 13000 RPM as a function of  $\lambda$ .**

Students should find that peak NO emissions correlate with a  $\lambda$  reading of approximately 1.08, and a similar maximum should be found on the rich side of things. They will find that fuel-rich NO emissions are on average much lower than fuel-lean emissions. Students should be urged to review NO formation in their textbook and notes in determining the effectiveness of their prediction model. At this point, students should also be able to analyze the effects of increasing or decreasing EGR, if applicable. Appendix P will be used as a class handout in reference to emissions modeling.

## **Chapter 9. Conclusions and Recommendations**

### **9.1 Summary of Accomplishments**

Single and two-zone heat release models were developed in MATLAB. Since a pressure trace wasn't available, the single-zone model was used to predict a pressure trace, where the two-zone model then bisected the combustion chamber. The single-zone model was used to predict engine outputs such as torque and specific-fuel consumption, while the two-zone model was used to predict a burned-zone temperature. Using the burned-zone temperature, sub-functions were created to calculate NO and HC emissions.

The MATLAB model was validated using theoretical and experimental Data. Back-of-the-envelope calculations, such as the adiabatic flame temperature and the power generated in an Otto Cycle, were used as a point of reference and validation. After the model was checked for accuracy using back-of-the-envelope calculations, experimental results were gathered on two engines. Although the results were limited due to mechanical failures, it was found that the model was highly accurate across most engine operating points.

After the model was validated, several class activities were created for the purpose of exposing students to engine phenomena and the process of numerically modeling engine performance. It was found that this model could be used extensively in exposing students to fuel and emissions mapping, heat release, sources of inefficiency, and spark-timing in the context of a real engine. Introductory and advanced class-projects were created for ME 433 (IC Engines).

It was also found that this research created an excellent starting point for future work. During testing, it was found that the model could be used extensively in creating baseline maps with a more complete set of data (i.e. a volumetric efficiency map and desired air-fuel

ratio map). This could save significant time in tuning an engine. The MATLAB model could be expanded to cover a broad range of topics, as will be described in the following sections.

## **9.2 Application of the Model in Future Research**

It was found that the model could be used in simulating baseline maps. This could save an enormous amount of time in tuning an engine, especially when little is known about the engine. With an air-fuel ratio and volumetric efficiency map, spark-timing could be optimized, thus minimizing wear-and-tear on the engine and dynamometer equipment. Much research could be directed towards refining the model and using it for the development of engine maps.

It was previously mentioned that empirical models could be refined based on a particular application. This could cover numerous categories including fluid dynamics through the intake and exhaust, fuel pooling during startup periods, gas dynamics through the valves, and engine friction as a function of RPM, just to name a few. Any one category could be studied and numerically modeled, and updated within the MATLAB model, thus producing more accurate results for a particular application.

The model could also grow significantly in regard to HC emissions predictions. Curve-fitted equations could be developed to relate HC oxidation within the exhaust to engine load, speed, coolant temperature, and spark advance. Although specific models can be found in SAE articles and in areas of research, these are really only applicable to particular engine designs and geometries. A refined model would be very useful for competition-based projects that require emissions testing.

### 9.3 Model Improvements

It was discovered that a graphical user interface (GUI) could be created in MATLAB. A GUI would make the model much simpler, because users wouldn't have to worry about accidentally deleting a line of code and having the model no longer function. This would also specify the exact inputs necessary to make the simulation function without students having to dig through hundreds of lines of MATLAB script.

Although the MATLAB model included variable specific heats and updated fluid properties, all of these variables were estimated empirically, which can produce error in a complex computer code. It was discovered that an interface could be created between MATLAB and EES, which could be used to access real fluid properties during the combustion cycle. This would reduce the overall complexity of the MATLAB model and would simplify atmospheric inputs. The purely MATLAB model and the new model could then be compared for accuracy.

The model also has room for growth in predicting engine performance with alternative fuels. As of now, the model only includes predictions for SI, iso-octane engines, but this could be expanded to include alternative fuels and engine configurations. Updating the SI simulation to include alcohols and nitrogen fuels would be a matter of changing a few chemical formulas; however, a CI model would involve much more work.

### 9.4 Test-Stand Improvements

The stand-stand that holds the YZ250F has significant room for improvement. First of all, the YZ250F is joined to the driveline on the dynamometer through the use of a Lovejoy coupler. The rubber padding on the couple has the tendency of wearing out because of the oscillatory and violent nature of the dynamometer. The wear-and-tear on the coupler

could be diminished by removing the parts between the driveline and the output shaft of the engine.

The table on which all electrical components are mounted also needs to be replaced. The current design vibrates during engine operation, and allows electrical connections to wear against each other and come loose. A sturdier platform would prevent hours of troubleshooting when electrical connections come loose.

## References

- [1] M. Klein, "A specific heat ratio model and compression ratio estimation," Univ., Linköping, 2004.
- [2] M. J. Moran, H. N. Shapiro, D. D. Boettner, and M. B. Bailey, *Fundamentals of Engineering Thermodynamics*, 7 edition. Wiley, 2010.
- [3] J. Heywood, *Internal Combustion Engine Fundamentals*. Tata Mcgraw Hill Education, 2011.
- [4] R. Stone, *Introduction to internal combustion engines*. Warrendale, PA: Society of Automotive Engineers, 1999.
- [5] G. P. Blair, *Design and Simulation of Four Stroke Engines [R-186]*. Society of Automotive Engineers Inc, 1999.
- [6] S. Turns, *An Introduction to Combustion: Concepts and Applications*, 3 edition. McGraw-Hill Science/Engineering/Math, 2011.
- [7] R. R. Raine, C. R. Stone, and J. Gould, "Modeling of nitric oxide formation in spark ignition engines with a multizone burned gas," *Combust. Flame*, vol. 102, no. 3, pp. 241–255, Aug. 1995.
- [8] S. C. Hill and L. Douglas Smoot, "Modeling of nitrogen oxides formation and destruction in combustion systems," *Prog. Energy Combust. Sci.*, vol. 26, no. 4–6, pp. 417–458, Aug. 2000.
- [9] S. J. Klippenstein, L. B. Harding, P. Glarborg, and J. A. Miller, "The role of NNH in NO formation and control," *Combust. Flame*, vol. 158, no. 4, pp. 774–789, Apr. 2011.
- [10] A. A. Konnov, G. Colson, and J. De Ruyck, "The new route forming NO via NNH," *Combust. Flame*, vol. 121, no. 3, pp. 548–550, Mar. 2000.
- [11] F. V. Tinaut, A. Melgar, and A. J. Horrillo, "Utilization of a Quasi-Dimensional Model for Predicting Pollutant Emissions in SI Engines," SAE International, Warrendale, PA, SAE Technical Paper 1999-01-0223, Mar. 1999.
- [12] D. A. Hamrin and J. B. Heywood, "Modeling of Engine-Out Hydrocarbon Emissions for Prototype Production Engines," SAE International, Warrendale, PA, SAE Technical Paper 950984, Feb. 1995.

- [13] K. MIN and W. K. CHENG, "Oxidation of the Piston Crevice Hydrocarbon During the Expansion Process in a Spark Ignition Engine," *Combust. Sci. Technol.*, vol. 106, no. 4–6, pp. 307–326, 1995.
- [14] A. C. Alkidas, "Combustion-chamber crevices: the major source of engine-out hydrocarbon emissions under fully warmed conditions," *Prog. Energy Combust. Sci.*, vol. 25, no. 3, pp. 253–273, Jun. 1999.
- [15] Y. G. Guezennec and W. Hamama, "Two-Zone Heat Release Analysis of Combustion Data and Calibration of Heat Transfer Correlation in an I. C. Engine," SAE International, Warrendale, PA, SAE Technical Paper 1999-01-0218, Mar. 1999.
- [16] J. W. Fox, W. K. Cheng, and J. B. Heywood, "A Model for Predicting Residual Gas Fraction in Spark-Ignition Engines," SAE International, Warrendale, PA, SAE Technical Paper 931025, Mar. 1993.
- [17] C. D. Rakopoulos and C. N. Michos, "Development and validation of a multi-zone combustion model for performance and nitric oxide formation in syngas fueled spark ignition engine," *Energy Convers. Manag.*, vol. 49, no. 10, pp. 2924–2938, Oct. 2008.
- [18] C. D. Rakopoulos, D. C. Rakopoulos, and D. C. Kyritsis, "Development and validation of a comprehensive two-zone model for combustion and emissions formation in a DI diesel engine," *Int. J. Energy Res.*, vol. 27, no. 14, pp. 1221–1249, 2003.
- [19] D. R. Stull, *Janaf Thermochemical Tables 2ND Edition NBS 37*. NATIONAL BUREAU OF STANDARDS.
- [20] K. Drobot, "Hydrocarbon Oxidation in the Exhaust Port and Runner of a Spark Ignition Engine." Department of Mechanical Engineering, Massachusetts Institute of Technology, 1991.
- [21] D. Rinker, *Use of a TK Solver Performance Model in the Design and Testing of a Formula Hybrid Racecar*. University of Idaho, 2013.
- [22] R. S. Figliola and D. E. Beasley, *Theory and Design for Mechanical Measurements*, 4th ed. Wiley, 2006.
- [23] B&S, *Borghi & Saveri Instruction Manual*. Bologna, Italy, 2006.
- [24] A. Fuhrman, *Comparison of the Exhaust Emissions From a Synchronous Charge Trapping Engine and a Rotax E-Tec Engine*. University of Idaho, 2013.

## **APPENDIX A: Derivation of the Polynomial Method**

Table of Coefficients

$a_1$ .692	$a_2$ 39.17e-06	$a_3$ 52.9e-09	$a_4$ -228.62e-13
$a_5$ 227.58e-17	$b_0$ 3049.33	$b_1$ -5.7e-03	$b_2$ -9.5e-05
$b_3$ 21.53e-09	$b_4$ -200.26e-14	$C_u$ 2.32584	$C_r$ 4.186e-03
$d_0$ 10.41066	$d_1$ 7.85125	$d_3$ -3.71257	$e_0$ -15.001e03
$e_1$ -15.838e03	$e_3$ 9.613e03	$f_0$ -1.10329	$f_1$ -.38656
$f_3$ .154226	$f_4$ -14.763	$f_5$ 118.27	$f_6$ 14.503
$r_0$ -.2977	$r_1$ 11.98	$r_2$ -25442	$r_3$ -.4354

Constants as a function of Temperature:

$$A(T) = a_1T + a_2T^2 + \dots + a_5T^5$$

$$B(T) = b_0 + b_1T + \dots + b_4T^4$$

Constants as a function of Lambda (Excess Air Coefficient):

$$D(\lambda) = d_0 + d_1\lambda - 1 + d_3\lambda - 3$$

$$E(T, \lambda) = e_0 + e_1\lambda - 1 + e_3\lambda - 3$$

Constant as a function of Temperature, Pressure, and Lambda:

$$F(T, p, \lambda) = (f_0 + f_1\lambda - 1 + f_3\lambda - 3 + f_4 + f_5\lambda - 1 T)\ln(f_6p)$$

Correction Factors for Internal Energy and the Gas Constant Were Found to Be:

$$u_{\text{corr}}(T, p, \lambda) = C_u \exp(D(\lambda) + E(T, \lambda) + F(T, p, \lambda))$$

$$R_{\text{corr}}(T, p, \lambda) = C_r \exp\left(r_0 \ln(\lambda) + \frac{r_1 + \frac{r_2}{T} + r_3 \ln(f_6P)}{\lambda}\right)$$

The Internal Energy as a function of Temperature, Pressure, and Lambda Was Found to Be:

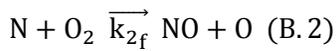
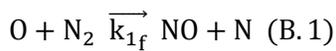
$$u(T, p, \lambda) = A(T) - B(T)\lambda + u_{\text{corr}}(T, p, \lambda)$$

The Ratio of Specific Heats Was Then Found to Be:

$$\gamma = \frac{c_p}{c_v} = 1 + \frac{R}{C_v}$$

## **APPENDIX B: Expanding the Zeldovich Mechanism**

**The General Zeldovich Mechanism Is Defined As:**



**The Reaction for NO and N in Terms of Concentration are Written:**

$$\frac{d[\text{NO}]}{dt} = k_{1f}[\text{O}][\text{N}_2] + k_{2f}[\text{N}][\text{O}_2] \quad (\text{B. 3})$$

$$\frac{d[\text{N}]}{dt} = k_{1f}[\text{O}][\text{N}_2] - k_{2f}[\text{N}][\text{O}_2] \quad (\text{B. 4})$$

**Assume That the Nitrogen Concentration is Constant (equation B.4):**

$$k_{1f}[\text{O}][\text{N}_2] = k_{2f}[\text{N}][\text{O}_2] \quad (\text{B. 4.1})$$

$$\therefore [\text{N}] = \frac{k_{1f}[\text{O}][\text{N}_2]}{k_{2f}[\text{O}_2]} \quad (\text{B. 4.2})$$

**Substitute the Steady State Nitrogen Concentration (equation B.4.2) into B.3:**

$$\frac{d[\text{NO}]}{dt} = k_{1f}[\text{O}][\text{N}_2] + k_{2f}[\text{O}_2] \left( \frac{k_{1f}[\text{O}][\text{N}_2]}{k_{2f}[\text{O}_2]} \right) \quad (\text{B. 5})$$

**Simplify Equation B.5:**

$$\frac{d[\text{NO}]}{dt} = 2k_{1f}[\text{O}]_e[\text{N}_2]_e \quad (\text{B. 5.1})$$

## **APPENDIX C: Calculating the Water-Gas Shift Constant**

**Calculating Constant A**

$$A = K_{\text{wgs}} - 1 \quad (\text{C.1})$$

**Calculating Constant B**

$$B = -K_{\text{wgs}} \left( 2 \left( \left( \frac{1}{\lambda} \right) - 1 \right) + \varepsilon \left( \frac{1}{\lambda} \right) \right) + 2 \left( 1 - \varepsilon \left( \frac{1}{\lambda} \right) \right) \quad (\text{C.2})$$

**Calculating Constant C**

$$C = 2K_{\text{wgs}} \varepsilon \left( \frac{1}{\lambda} \right) \left( \frac{1}{\lambda} - 1 \right) \quad (\text{C.3})$$

**Combining Constants (C.1-C.3)**

$$c = \frac{-B - (B^2 - 4AC)}{2A} \quad (\text{C.4})$$

## **APPENDIX E: MATLAB Script**

```

%University Of Idaho Engine Simulation
%Uses "Two Zone" Combustion Analysis With Variable Specific Heats Ratios
%Only Models The Compression And Expansion Strokes
%

```

---

```

clear all;
close all;
clc;

```

```

%

```

---

```

%Engine Inputs

```

```

Load = 1;           %Engine Load (Affects Inlet Pressure)
RPM = 10500;       %Revolutions Per Minute [1/min]
L = (53.6/1000);   %Stroke of Engine [m]
B = (77/1000);     %Bore of Engine [m]
l = .0935;         %Length of Engine Connecting Rod [m]
N_cyl = 1;         %Number of Cylinders [unitless]
C_r = 12.5;        %Compression Ratio [unitless]
N_r = 2;           %Number of Revolutions Per Power Stroke
theta_b = 85;      %Combustion Burn Duration [degrees]
theta_0 = 145;     %Crank Angle At Start of Combustion [degrees]
theta_f = theta_0+theta_b; %Final Comb. Angle [degrees]
IVC = 0;           %Time [degrees] when Intake Valve Closes
EVO = 314;         %Time [degrees] when Exhaust Valve Opens

```

```

%

```

---

```

%Engine Calculations Based On Previous Inputs

```

```

%Assumes Average Surface Area In Which Heat Transfer Occurs

```

```

A_p = (pi/4)*B^2;   %Cross Sectional Piston Area [m^2]
A_ch = 2*A_p;       %Cylinder Head Surface Area (in chamber)
V_d = N_cyl*A_p*L;  %Displaced Volume Of Engine [m^3]
N = RPM/60;         %Converts RPM to RPS [1/s]
S_bar_p = 2*L*N;    %Calculates Mean Piston Speed [m/s]
a = L/2;            %Calculates Crank Radius (1/2 stroke) [m]
V_TDC = (V_d/(C_r-1))/N_cyl; %Calculates Clearance Volume [m^3]
V_BDC = (V_d/N_cyl)+V_TDC; %Cyl. Volume At BDC [m^3]

```

```

%

```

---

```

%Calculating Losses Due To Friction

```

```

%fmep (obtained from Blair) Based On Displacement, RPM

```

```

% if V_d>500*10^(-6)
%   fmep=(100000+350*L*RPM)*10^(-3);
% end
% if V_d<500*10^-6
%   fmep=(100000+100*(500-V_d*10^(-6))+350*L*RPM)*10^(-3);
%end

```

```

%For Motorcycles, Use "Rolling" Bearings (For Automobiles, Use Previous)

```

```

fmep = (250*L*RPM)*10^-3;

```

```
%Volumetric Efficiency Correction Factor
```

```
CF = correction( Load,RPM );
```

```
%Initial Preallocation Of Matrices (Second Preallocation In Loops Needs To  
%Be Included (Do Not Delete)
```

```
V(1:360)=zeros;DV(1:360)=zeros;rho(1:360)=zeros;mu(1:360)=zeros;  
C_k(1:360)=zeros;C_R(1:360)=zeros;X(1:360)=zeros;M_F(1:360)=zeros;  
DX(1:360)=zeros;Re(1:360)=zeros;Nus(1:360)=zeros;h_g(1:360)=zeros;  
DQ_w(1:360)=zeros;DQ(1:360)=zeros;Q(1:360)=zeros;DT(1:360)=zeros;  
DP(1:360)=zeros;P(1:360)=zeros;T(1:360)=zeros;W_dot(1:360)=zeros;  
W(1:360)=zeros;T_indicated(1:2)=zeros;Q_dot(1:360)=zeros;u(1:360)=zeros;  
du(1:360)=zeros;cv(1:360)=zeros;m_b(1:360)=zeros;m_u(1:360)=zeros;  
V_u(1:360)=zeros;V_b(1:360)=zeros;T_u(1:360)=zeros;T_b(1:360)=zeros;  
A_u(1:360)=zeros;A_b(1:360)=zeros;DT_u(1:360)=zeros;gamma_u(1:360)=zeros;  
u_u(1:360)=zeros;du_u(1:360)=zeros;cv_u(1:360)=zeros;DQ2(1:360)=zeros;  
DQ_w2(1:360)=zeros;Q2(1:360)=zeros;
```

```
%
```

---

```
%Fuel Inputs/Efficiencies
```

```
AF_ratio_stoich = 15.09;    %Gravimetric Air Fuel Ratio (Stoich)  
AF_ratio_mol_sotich=14.7; %Molar Air_Fuel Ratio (Stoich)  
lambda = .90;            %Excess Air Coefficient  
AF_ratio_ac = lambda*AF_ratio_stoich; %Actual Air Fuel Ratio  
AF_ratio_mol=lambda*AF_ratio_mol_sotich;  
LHV = 44.6e6;            %Lower Heating Value Of Fuel Mixture [J/kg]  
eta_combmax = .95;      %Assumed MAX COmb. Efficiency
```

```
%Predicts Combustion Efficiency (Reference To Blair)
```

```
eta_comb=eta_combmax*(-1.6082+4.6509*lambda-2.0764*lambda^2);
```

```
%Atmospheric Inputs
```

```
P_atm = 101325;  
T_atm = 290;  
P_BDC = Load*P_atm;    %Inlet Pressure[Pa] Moscow,ID  
R_air = 287;            %Gas Constant For Air [J/kg-K]  
gamma(1:360) = 1.4;    %Preallocate Gamma Array (sets initial value)  
T_w =350;              %Assumed Wall Temperature (Reference Stone)
```

```
%
```

---

```
%Polynomials Used To Calculate Gamma As A Function Of RPM
```

```
a_1 = .692;    a_2 = 39.17e-06;    a_3 = 52.9e-09; a_4 = -228.62e-13;  
a_5 = 277.58e-17;b_0 = 3049.33;  b_1 = -5.7e-02; b_2 = -9.5e-05;  
b_3 = 21.53e-09;b_4 = -200.26e-14;c_u = 2.32584;  c_r = 4.186e-03;  
d_0 = 10.41066; d_1 = 7.85125;  d_3 = -3.71257;e_0 = -15.001e03;  
e_1 = -15.838e03;  e_3 = 9.613e03;f_0 = -.10329;  f_1 = -.38656;  
f_3 = .154226;  f_4 = -14.763;  f_5 = 118.27;  f_6 = 14.503;  
r_0 = -.29777;  r_1 = 11.98;  r_2 = -25442;  r_3 = -.4354;
```

```
%
```

---

```
R=R_air/1000;
```

```

for k = 1:2
%Corrects Temperature Based On Exhaust Gas Residuals
if k==1
    T_BDC = T_atm;           %Assumed Inlet Temperature [K]
else
    T_BDC=T_corr;
end

%Calculate Mass of Air In Cylinder/ Mass Of Fuel Based On AFR
rho_a = P_BDC/(R_air*T_BDC); %Air Density kg/m^3
m_a = rho_a*V_d;           %Mass of Air In Cylinder [kg]
m_f = m_a/AF_ratio_ac;    %Mass Of Fuel In Cylinder [kg]
m_c = m_a+m_f;           %Mass In Cylinder

%Specifying Initial Conditions For Loops
%DV,DX,etc. Are Relative To Change In Theta (i.e. DV/Dtheta)

theta(1:360)=zeros; %Starting Crank Angle [deg]
V(1:360)=zeros; %Preallocate Volume Array
V(1)=V_BDC; %Starting Combustion Chamber Volume [m^3]
DV(1:360) = zeros; %Preallocate Change In Volume Array
DV(1) = 0; %Specifying Initial Change In Volume [m^3]
P(1:360)=P_BDC; %Preallocate Pressure Array
DP(1:360) = zeros; %Specifying Initial Change In Pressure
T(1:360)=zeros; %Preallocate Temperature Array
T(1) = T_BDC; %Inlet Temperature [K]
T_u(1)=T_BDC; %Initial Unburned Temperature[K]
DT(1:360) = zeros; %Specifying Initial Change In Temperature
DT_u(1:360)=zeros; %Preallocate Change In Unburned Temperature
gamma(1)=1.4; %Initial Gamma Input
gamma_u(1)=1.4; %Initial Gamma Input
X(1:360) = 0; %Preallocate Mass Burn Array
DX(1:360) = zeros; %Preallocate Change In Mass Burn Fraction [unitless]
DQ(1:360) = zeros; %Preallocate Heat Release Array
DQ2(1:360)=zeros; %Preallocate Two Zone Heat Release Array
Q(1:360)=zeros; %Preallocate Heat Array
Q2(1:360)=zeros; %Preallocate 2 zone Heat Array
M_F(1:360) = 0; %Preallocate Mass In Combustion Chamber Array
rho(1:360) = zeros; %Preallocates Ideal Gas Law array
rho(1) = P(1)/(R_air*T(1)); %Initial Value Ideal Gas Array
mu(1:360)=zeros; %Preallocate Viscosity Array
mu(1)=7.457*10^(-6)+4.1547*10^(-8)*T_BDC-7.4793*10^(-12)*T_BDC^(2);
C_k(1:360)=zeros; %Preallocate Thermal COnductivity Array
C_k(1) = 6.1944*10^(-3)+7.3814*10^(-5)*T_BDC-1.2491*10^(-8)*T_BDC^(2);
C_R(1:360) = zeros; %Preallocate Radiation Coefficient Array
C_R(1) = 4.25*10^(-09)*((T(1)^4-T_w^4)/(T(1)-T_w)); %Initial Rad. Coeff
Re(1:360)=zeros; %Preallocate Reynolds Value Array
Re(1)=rho(1)*S_bar_p*B/mu(1); %Initial Reynolds Value
Nus(1:360)=zeros; %Preallocating Nusselt Number Array
Nus(1)=.49*Re(1)^(.7); %Initial Nusselt Number
h_g(1:360)=zeros; %Preallocate Heat Transfer Coefficient Array
h_g(1)=C_k(1)*Nus(1)/B; %Initial Heat Transfer Coefficient
s(1:360)=zeros; %Preallocates Distance Crank/Piston Axes Array
s(1) = -a*cosd(theta(1))+sqrt(1^2 - a^2*sind(theta(1))^2); %Initial Val.
W(1:360) = zeros; %Preallocate Work Array

```

```

W_dot(1:360) = zeros; %Preallocate Power Array
T_indicated(1:360) = zeros; %Preallocate Torque Array
Q_dot(1:360) = zeros; %Preallocate Heat Transfer Array
u(1:360) = zeros; %Preallocate Internal Energy Array
du(1:360) = zeros; %Preallocates Change In Internal Energy Array
cv(1:360) = zeros; %Preallocates Heat Capacity Array
DQ_w(1:360)=zeros; %Preallocate Convective Heat Loss Array
DQ_w2(1:360)=zeros; %Preallocate Convective Heat Loss Array 2 zone
m_b(1:360)= zeros; %Preallocate mass burned array
m_u(1:360)=m_c; %Preallocate unburned mass array
V_u(1:360)=zeros; %Preallocate unburned Volume Array
V_u(1) = V(1); %Initial Unburned Volume

%
theta=1:360;

for i = 2:360

    %Specifies Distance Between Crank/Piston Axes As A Function Of theta
    s = -a*cosd(theta(i))+sqrt(l^2 - a^2*sind(theta(i))^2);
    %Specifies Volume As A Function Of Crank Angle
    V(i) = V_TDC + ((pi/4)*B^2)*(l + a - s);
    %Specifies Change In Volume As A Function Of Crank Angle
    DV(i) = V(i)-V(i-1);
    %Calculates Density As A Function Of Crank Angle
    rho(i) = P(i-1)/(R_air*T(i-1));
    %Calculates Viscosity As A Function Of Temperature
    mu(i)=7.457*10^(-6)+4.1547*10^(-8)*T(i-1)-7.4793*10^(-12)*T(i-1)^(2);
    %Calculating Instantaneous Thermal Conductivity of Cylinder Gas
    C_k(i) = 6.1944*10^(-3)+7.3814*10^(-5)*T(i-1)-1.2491*10^(-8)*T(i-
1)^(2);
    %Calculating The Radiation Heat Transfer Coefficient
    C_R(i) = 4.25*10^(-09)*((T(i-1)^4-T_w^4)/(T(i-1)-T_w));
    %Instantaneous Surface Area (For Heat Transfer)
    A = A_ch + A_p + pi*B*(l+a-s);
    if i<=2
        A_u=A;
    end

%
    %Specifies Mass Fraction Burn As A Function Of Crank Angle (Weibe
Fcn.)
    %Also Specifies Mass Of Fuel In Combustion Chamber As A Function Of
%Theta

    if theta(i)<theta_0
        X(i)=0;
    else
        X(i) = 1-exp(-5*((theta(i)-theta_0)/theta_b)^3);
        if theta(i) < theta_f
            M_F(i) = V(theta_0-1)*rho(theta_0-1)/(lambda*AF_ratio_mol);
        end
    end
end

```

```

%
%-----
%Specifies Change In Mass Fraction Burn As A Function Of Crank Angle
DX(i) = X(i) - X(i-1);

%
%-----
%Incorporating The Annand Method To Predict Heat Transfer
%Calculating Reynolds Number
Re(i)=rho(i)*S_bar_p*B/mu(i);
%Calculating Nusselt Number (constant=.26 two stroke, .49 4 stroke)
Nus(i)=.49*Re(i)^(.7);
%Calculating Heat Transfer Coefficient Using Annand Method
h_g(i)=C_k(i)*Nus(i)/B;
%Calculates Convective Losses Into Wall As A Function Of Crank Angle
DQ_w(i) = (h_g(i)+C_R(i))*A*(T(i-1)-T_w)*(60/(360*RPM));
%Calculates Change In Heat Transfer (total) As A Function Of Crank
%Angle
DQ(i) = eta_comb*LHV*M_F(i)*DX(i)-DQ_w(i);
%Calculates Total Heat Transfer (Per Cycle)
Q(i) = Q(i-1)+DQ(i);

%
%-----
%Specifies Pressure and Temperature Increases Between Intake Valve
%Closing and Exhaust Valve Opening
if IVC< theta(i)
    DT(i)=T(i-1)*(gamma(i-1)-1)*((1/(P(i-1)*V(i-1)))*DQ(i)...
        -(1/V(i-1))*DV(i));
    DP(i)=(-P(i-1)/V(i-1))*DV(i)+(P(i-1)/T(i-1))*DT(i);
    P(i) = P(i-1)+DP(i);
end
if EVO < theta(i)
    P(i) = P_atm;
end
if 200 < theta(i)
    if P(i)<=P_atm
        P(i)=P_atm;
    end
end
end
%
%-----
%Calculate Burned, Unburned Mass Fractions
m_b(i) = m_b(i-1)+DX(i)*m_c;    %Burned Mass
m_u(i) = m_u(i-1)-DX(i)*m_c;    %Unburned Mass
%Calculating Burned, Unburned Volumes
if theta(i)<=theta_0
    V_u(i)=N_cyl*V(i);
end
if theta(i)>theta_0
V_u(i)=(m_u(i)*V_u(i-1))/m_u(i-1)*(P(i)/P(i-1))^(1/gamma_u(i-1));
end
V_b(i)=N_cyl*V(i)-V_u(i);
if V_b(i)<0
    V_b(i)=0;
end
end

```

```

%Calculating Burned, Unburned Temperatures
T_u(i)=P(i)*V_u(i)/(m_u(i)*R*1000);
if theta(i) <= theta_0+4
    T_b(i)=0;
end
if theta(i)>theta_0+4
    T_b(i)=P(i)*V_b(i)/(m_b(i)*R*1000);
end

%Calculate Unburned, Burned Areas Based On Volume Ratio
A_u(i)=A*(1-sqrt(X(i)));
A_b(i)=A*(X(i)/sqrt(X(i)));
DT_u(i)=T_u(i)-T_u(i-1);

%
%Returns Temperature Values To Beginning Of Loop
%Assumes Temperature Drops Back To ATM Temp After Exhaust Is Extracted
T(i) = T(i-1)+DT(i);
%Calculate The Residual Gas Fraction
%Assume A Polytropic Constant Of 1.3
R_frac = (1/C_r)*(P_BDC/P_atm)^(1/1.3)*(1/lambda);
%Calculates Cylinder Work [J] As A Function Of Crank Angle
%Treats Atmospheric Pressure As Reference State
W(i) = W(i-1)+(P(i)-P_atm)*DV(i);
%Calculates Power [kW] As A Function Of Crank Angle
W_dot(i)=(N_cyl*W(i)*N/N_r)/1000;
%Indicated Mean Effective Pressure
imep = CF*W_dot(360)*N_r*1000/(V_d*1000*N);
%Calculates Torque[N*m] As A Function Of Crank Angle
T_indicated(i) = (W_dot(i)*1000)/(2*pi*N);
%Calculates Heat Loss [kW] As A Function Of Crank Angle
Q_dot(i) = (N_cyl*Q(i)*N/N_r)/1000;

%
% The Following Section Of Code Calculates An Updated Value Of Gamma
% Using The "Polynomial Method" Developed By Krieger-Borman
% User Of This Code Must Be Careful Because Accuracy Of This Method
% Drops As The Fuel Mixture Becomes Increasingly Rich

%Calculates A,B Factors For Following Block Of Code
A_t = a_1*T(i)+a_2*T(i)^2+a_3*T(i)^3+a_4*T(i)^4+a_5*T(i)^5;
A_tu = a_1*T_u(i)+a_2*T_u(i)^2+a_3*T_u(i)^3+a_4*T_u(i)^4+a_5*T_u(i)^5;
B_t = b_0+b_1*T(i)+b_2*T(i)^2+b_3*T(i)^3+b_4*T(i)^4;
B_tu = b_0+b_1*T_u(i)+b_2*T_u(i)^2+b_3*T_u(i)^3+b_4*T_u(i)^4;
%Calculates Factor "D" As A Function Of lambda
D_lambda = d_0 + d_1*lambda^(-1)+ d_3*lambda^(-3);
%Calculates Factor "F" As A Function Of Temperature,lambda
E_TLambda = (e_0 + e_1*lambda^(-1)+ e_3*lambda^(-3))/T(i);
E_TLambdau = (e_0 + e_1*lambda^(-1)+ e_3*lambda^(-3))/T_u(i);
F_TPLambda = (f_0 + f_1*lambda^(-1) + f_3*lambda^(-3) + ...
    ((f_4 + f_5*lambda^(-1))/T(i)))*log(f_6*P(i));
F_TPLambdau = (f_0 + f_1*lambda^(-1) + f_3*lambda^(-3) + ...
    ((f_4 + f_5*lambda^(-1))/T_u(i)))*log(f_6*P(i));
%Calculates Correction Factor For Internal Energy
u_corr = c_u*exp(D_lambda +E_TLambda + F_TPLambda);

```

```

u_corr_u=c_u*exp(D_lambda +E_TLambdau + F_TPLambdau);
%Calculates Internal Energy As A Function Of Crank Angle
u(i) = A_t - B_t/lambda + u_corr;
u_u(i) = A_tu - B_tu/lambda + u_corr_u;
%Calculates Change In Internal Energy
du(i) = u(i) - u(i-1);
du_u(i) = u_u(i) - u_u(i-1);
%Calculates Heat Capacity "C_v" As A Function Of Crank Angle
cv(i) = du(i)/DT(i);
cv_u(i)=du_u(i)/DT_u(i);
%Calculates Correction Factor For "R" Value As A Function Of Crank
%Angle
R_corr = c_r*exp(r_0*log(lambda) + (r_1+r_2/T(i) + ...
    r_3*log(f_6*P(i)))/lambda);
R_corr_u = c_r*exp(r_0*log(lambda) + (r_1+r_2/T_u(i-1) + ...
    r_3*log(f_6*P(i)))/lambda);
%Calculates Actual "R" Value
R = .287 + .020/lambda + R_corr;
R_u = .287 + .020/lambda + R_corr_u;
%Calculates Actual Gamma Value And Returns To Beginning Of Code
gamma_u(i)=1+R_u/cv_u(i);
gamma(i) = 1 + R/cv(i);
    if gamma(i)<1.2
        gamma(i)=1.4;
        gamma_u(i)=1.4;
    end

    if theta(i)>=EVO
        gamma(i)=1.4;
        gamma_u(i)=1.4;
    end

%


---


%Calculate Temperature Of Exhaust Based On Polytropic Relations
if EVO < theta(i)
T(i)=T(EVO)*(P_BDC/P(EVO))^( (gamma(i)-1)/gamma(i));
T_b(i)=T_b(EVO)*(P_BDC/P(EVO))^( (gamma(i)-1)/gamma(i));
end
end
%Calculates A Corrected Inlet Temperature Based On EGR
%T_corr = R_frac*T(360)+(1-R_frac)*T_BDC;
T_corr = T_BDC;
end
%


---


%Specified Outputs (On Matlab Screen)
W_dot_indicated=W_dot(360);
bmep = imep-fmep;
W_dot_ac = (bmep*V_d*1000*N/(N_r*1000));
T_ac = W_dot_ac/(2*pi*N*10^(-3));

%Calculated Mechanical Efficiency (Based On Previous Inputs)
eta_m = bmep/imep; %Calculates Mechanical Efficiency

```

```

%
%Calculates Brake Specific Fuel Consumption
m_ta = P_BDC*V_d/(R_air*T_BDC);           %Calculate Trapped Air In Cylinder
eta_v = CF*((m_ta)/(rho_a*V_d));          %Corrected Volumetric Efficiency
m_dot_f = N_cyl*M_F(theta_0)*(N/N_r);    %Mass Flow Rate Of Fuel
m_dot_a = AF_ratio_ac*m_dot_f;           %Mass Flow Rate Of Air
BSFC = (m_dot_f*1000*3600)/(W_dot_ac);    %BSFC [g/kW*h]
eta_f = 3600/(BSFC*(LHV*10^(-6)));        %Fuel Conversion Efficiency

%Calculate Emissions

T_NO=.875*max(T_b);                       %Calculate Avg. Burn Temp
P_NO=max(P);                              %Assuming Pressure is peak
P_EXH=(P(EVO)+P_atm)/2;                   %Calculating Exhaust Press.
[ PPM_NO ] = NOX( T_NO,P_atm,lambda,P_NO,T_BDC,P_BDC,P_EXH)
P_peak = max(P);                          %Peak Pressure
disp('Percentage of Fuel Mass Reaching Exhaust')
[ HC ] = hydrocarbons(
R_frac,AF_ratio_ac,B,P_peak,imep,C_r,V_d,N_cyl,T_w,N );
HC =100*HC
%
%Specifies Conditions For Minimum and Maximum Plot Values
v_min = min(V); v_max = max(V);
p_min = min(P); p_max = max(P);
w_min = min(W_dot); w_max = max(W_dot);
T_min = min(T); T_max = max(T);
Q_min = min(Q_dot); Q_max = max(Q_dot);
Tmin = min(T_indicated); Tmax = max(T_indicated);

%
%Plot Statements

% figure(1)
% plot(theta,X)
% title('Mass Fraction Burned Vs. Theta')
% xlabel('theta[deg]')
% ylabel('Mass Fraction Burned (%)')
% axis([0 360 -.1 1.1])
%
% figure(2)
% plot(theta,V)
% title('Volume Vs. Crank Angle')
% xlabel('theta[deg]')
% ylabel('Volume [m^3]')
% axis([0 360 v_min v_max])
%
% figure(3)
% plot(theta,P/1000)
% title('Indicated Cylinder Pressure Vs. Crank Angle')
% xlabel('theta[deg]')
% ylabel('Pressure [kPa]')
% axis([0 360 p_min/1000 p_max/1000])

```

```
%  
% figure(4)  
% plot(theta,T)  
% title('Cylinder Temperature Vs. Crank Angle')  
% xlabel('theta[deg]')  
% ylabel('Temperature [K]')  
% axis([0 360 T_min T_max])  
%  
% figure(5)  
% title('Power and Heat Transfer')  
% plot(theta,W_dot,'g')  
% hold on;  
% plot(theta,Q_dot,'r')  
% legend Power HX  
% xlabel('theta[deg]')  
% ylabel('kW')  
% axis([1 360 -50 300])  
%  
% figure(6)  
% %plot(theta,T,'g')  
% xlabel('theta[deg]')  
% ylabel('Temperature [K]')  
% title('Unburned and Burned Zone Temperatures [K]')  
% hold on;  
% plot(theta(1:EVO),T_u(1:EVO),'b')  
% plot(theta(theta_0+10:EVO),T_b(theta_0+10:EVO),'r')  
% legend unburned burned  
% axis([0 EVO 300 3500])
```

**APPENDIX F: NO<sub>x</sub> Function**

```

function [ PPM_NO ] = NOX( T_NO,P_atm,lambda,P_NO,T_BDC,P_BDC,P_EXH)
R_u=8315;           %Universal Gas Constant
psi=3.773;         %Molar N/O ratio
y=18/8;           %Molar H/C ratio (Using Iso-Octane)
epsilon=4/(4+y);   %y is the molar H/C ratio

%Calculate Equilibrium Constant At Given Temperature (Water Gas Shift)
K_wgs=exp(2.743-1.761*10^3/T_NO-1.611*10^6/(T_NO^2)+.2803*10^9/(T_NO^3));

%Atom Balance Based On Excess Air Coefficient

if 1/lambda<1
    n_CO2=epsilon*(1/lambda);
    n_H2O=2*(1-epsilon)*(1/lambda);
    n_CO=0;
    n_H2=0;
    n_O2=1-(1/lambda);
    n_N2=psi;
    n_b= (1-epsilon)*(1/lambda)+1+psi;
end

if 1/lambda>=1
    A=(K_wgs-1);
    B=-K_wgs*(2*((1/lambda)-1)+epsilon*(1/lambda))+2*(1-
epsilon*(1/lambda));
    C=2*K_wgs*epsilon*(1/lambda)*((1/lambda)-1);
    %Watch Quadratic Equation, Moles Must Be Positive!
    c=(-B-sqrt(B^2-4*A*C))/(2*A);
    n_CO2=epsilon*(1/lambda)-c;
    n_H2O=2*(1-epsilon*(1/lambda))+c;
    n_CO=c;
    n_H2=2*((1/lambda)-1)-c;
    n_O2=0;
    n_N2=psi;
    n_b = (2-epsilon)*(1/lambda)+psi;
end

%
-

%Calculate Molar Fractions Of Each Constituent Element
x_CO2=n_CO2/n_b;    x_H2O=n_H2O/n_b;    x_CO=n_CO/n_b;    x_H2=n_H2/n_b;
x_O2=n_O2/n_b;    x_N2_e=n_N2/n_b;

if 1/lambda>1
    n_prod=1.5;
    z=(T_NO-3.1*10^3)/(1.7*10^3);
    K_p_CO2=1.7*z^8-2.6*z^7-5.8*z^6+7.5*z^5+7.6*z^4-8.4*z^3-.51*z^2-
2.1*z+...
    6.8;
    Z=(T_NO-3.1*10^3)/(1.7*10^3);
    K_p_CO=.48*Z^8-.72*Z^7-1.6*Z^6+2.1*Z^4-2.4*Z^3-.13*Z^2-.74*Z+6.4;
    K_P=10^(K_p_CO2-K_p_CO);
    P_p = ((P_EXH/101325)/n_prod)*(T_NO/T_BDC)^(-1);
    ALPHA= (2*P_p)/(3*K_P^2) + (((P_p/K_P^2) - (4*P_p^2)/(3*K_P^4) + ...

```

```

(8*P_p^3)/(27*K_P^6))^2 + ((4*P_p)/(3*K_P^2) - ...
(4*P_p^2)/(9*K_P^4))^3)^(1/2) + P_p/K_P^2 - (4*P_p^2)/(3*K_P^4)
+ ...
(8*P_p^3)/(27*K_P^6))^(1/3) - ((4*P_p)/(3*K_P^2) - ...
(4*P_p^2)/(9*K_P^4))/(((P_p/K_P^2 - (4*P_p^2)/(3*K_P^4) + ...
(8*P_p^3)/(27*K_P^6))^2 + ((4*P_p)/(3*K_P^2) - ...
(4*P_p^2)/(9*K_P^4))^3)^(1/2) + P_p/K_P^2 - (4*P_p^2)/(3*K_P^4) +
...
(8*P_p^3)/(27*K_P^6))^(1/3);
x_O2=(ALPHA/(2*n_prod))*(x_CO2*((1-ALPHA)/n_prod));
end

%Calculate Equilibrium Concentrations
X_O2_e=x_O2*P_atm/(R_u*T_NO);
%
-----
%Equilibrium Constant For O2 to Oxygen Reaction
Kp_7=3.6*10^3*exp(-31090/T_NO)*(101325^(1/2)); %Pa^(1/2)
x_O_e= (Kp_7*X_O2_e^(1/2))/((R_u*T_NO)^(1/2))/(P_atm/(R_u*T_NO));
%kmol/m^3
%The Forward Reaction Rate Constant (m^3/kmol-s)
k_1f=1.82*10^11*exp(-38370/T_NO);
%Calculate Change in NO Concentration as Function of Time
dNOdt=2*k_1f*x_O_e*x_N2_e*P_atm/(R_u*T_NO);
%Calculate residence time
t_NO=(8*10^(-16)*T_NO*exp(58300/T_NO))/(P_NO/101325)^(1/2);
%Calculate NO PPM
PPM_NO = dNOdt*t_NO*10^6;
end

```

## **APPENDIX G: HC Function**

```

function [ HC ] = hydrocarbons(
R_frac,AF_ratio_ac,B,P_peak,imep,C_r,V_d,N_cyl,T_w,N )
%Offset Of Spark Plug From Central Axis of Cylinder
d_splug = 0;
%Calculate Crevice Volume
h_crevice = (3/1000); %Crevice Height (m)
gap = (1.5/1000); %Crevice Width
V_crevice = (pi/4)*B^2*h_crevice - (pi/4)*(B-2*gap)^2*h_crevice;
%Calculate Unburned Fraction
f_unburned = (1-R_frac);
%Calculate Fuel Vapor
f_vapor = 1/(1+AF_ratio_ac);
%Modification Factor Based On Spark Plug
f_mod = (1-.858*(d_splug/B));
%Crevice Emissions Index
SF_crevice = 5443*(P_peak/imep)*(V_crevice/(V_d/N_cyl))*(1/T_w)*...
f_unburned*f_vapor*f_mod;
%Oil Layer Predictions
P_i = .09875+.00986*imep;
P_ideal = (P_i+P_i*C_r^1.4)/2;
SF_wall = 63024*(1/imep)*(1/(AF_ratio_ac*10^(.0082*T_w)*B))*P_ideal;
%The Threshold of HC Oxidation
P_70 = .209+.0102*imep;
T_70 = 1600+.759*imep-.00051*imep^2;
T_HC = (T_70-T_w)/log(T_70/T_w);
T_HC_adj = 1600+.759*imep-.000051*imep^2;
%Fraction of Cylinder Oxidation
f_ox = 1-(P_70/P_ideal)*(T_HC/T_HC_adj)^3;
RELSP = .829*R_frac/100;
f_ox_ex = .866-.0000146*N-.00007*imep-.007918*RELSP-.0000255*T_w;
HC = (SF_crevice*(1-f_ox)+SF_wall*(1-f_ox))*(f_ox)*(1-f_ox_ex);
end

```

**APPENDIX H: Correction Factor**

```
function [ CF ] = correction( Load,RPM )
if Load<=1
    %CF = (-3*10^(-9))*RPM^2+5*10^(-5)*RPM+.7088;
    CF=-8*10^(-9)*RPM^2+.000135*RPM+.31944;
    if Load<=.9
        CF=-8*10^(-9)*RPM^2+.000135*RPM+.31944;
        %CF = CF-(1-Load)/4;
    end
end

end

end
```

**APPENDIX I: Load Function**

```
function [BSFC,T_ac,N_RPM,PPM_NO,HC,eta_f]=LOAD2(Load)

RPM = 1500;          %Sets Starting Pt. For Loop
N_RPM = 12;         %Fifteen RPM Data Sets
BSFC(1:N_RPM)=zeros; %Preallocate Array
T_ac(1:N_RPM)=zeros; %Preallocate Array
PPM_NO(1:N_RPM)=zeros;
for i = 1:N_RPM
RPM = RPM+1000;
[BSFC(i),T_ac(i),PPM_NO(i),HC(i),eta_f(i)]=BSFCAnnand(RPM,Load);
end
end
```

## **APPENDIX J: BSFC and Emissions Mapping Call Function**

```

clear all;
close all;
clc;
%


---


Load = 1.1;           %Start High, Decreased By .1 Each Iteration
N_load = 7;          %Six Loads
% bsfc(1:N_RPM,1:N_load)=zeros;
% Torque(1:N_RPM,1:N_load)=zeros;

%


---


for j = 1:N_load
Load = Load-.1;      %Decreases Load With Each Iteration
[BSFC,T_ac,N_RPM,PPM_NO,HC,eta_f]=LOAD2(Load); %Feeds "Load" Into Load
Function
BSFC=BSFC';         %Transposes Vector
bsfc(1:N_RPM,j)=BSFC;%Creates Matrix Out of BSFC Vectors
T_ac=T_ac';         %Transposes Torque Vector
Torque(1:N_RPM,j)=T_ac'; %Creates Matrix Out of Torque Vectors
PPM_NO=PPM_NO';
NO_x(1:N_RPM,j)=PPM_NO;
HC=HC';
hydrocarbon(1:N_RPM,j)=HC;
eta_f=eta_f';
ETA_f(1:N_RPM,j)=eta_f;
end

%


---


%Create RPM Matrix
RPM(1:N_RPM,1:N_load)=zeros;
RPM(1,1)=2500;
for i = 2:N_RPM
RPM(i,1) = RPM(i-1,1)+1000;
end
for j = 2:N_load
RPM(1:N_RPM,j)=RPM(1:N_RPM,j-1);
end

%


---


%Plot Statements
figure(1)
contourf(RPM,Torque,bsfc,45)
xlabel('RPM')
ylabel('Torque [N*m]')
title('YAMAHA YZ250F BSFC [g/kW-h] MAP')
colorbar
colormap jet
caxis([300 500])

figure(2)
contourf(RPM,Torque,NO_x,35)
xlabel('RPM')
ylabel('Torque [N*m]')

```

```
title('YAMAHA YZ250F NO Emissions MAP [PPM]')
colorbar
caxis([325 475])

figure(3)
contourf(RPM,Torque,hydrocarbon,35)
xlabel('RPM')
ylabel('Torque [N*m]')
title('YAMAHA YZ250F HC Emissions MAP [%]')
colorbar
caxis([2 6])

figure(4)
contourf(RPM,Torque,ETA_f,35)
xlabel('RPM')
ylabel('Torque [N*m]')
title('YAMAHA YZ250F HC Fuel Efficiency Map [%]')
colorbar
```

## **APPENDIX K: BSFC and Emissions Mapping Function**

```

function [BSFC, T_ac, PPM_NO, HC, eta_f] = BSFCAnnand (RPM, Load)

%University Of Idaho Engine Simulation
%Uses "Two Zone" Combustion Analysis With Variable Specific Heats Ratios
%Only Models The Compression And Expansion Strokes
%
-----

% clear all;
% close all;
% clc;

%
-----

%Engine Inputs
%Load = 1;           %Engine Load (Affects Inlet Pressure)
%RPM = 16500;       %Revolutions Per Minute [1/min]
L = (53.6/1000);    %Stroke of Engine [m]
B = (77/1000);     %Bore of Engine [m]
l = .0935;         %Length of Engine Connecting Rod [m]
N_cyl = 1;         %Number of Cylinders [unitless]
C_r = 12.5;        %Compression Ratio [unitless]
N_r = 2;           %Number of Revolutions Per Power Stroke
%theta_b = 85;     %Combustion Burn Duration [degrees]
%theta_0 = 145;    %Crank Angle At Start of Combustion [degrees]

if Load==1
theta_0=ceil(-.0013*RPM+154.82);
theta_b=ceil(.0038*RPM+40);
end
if Load<1
theta_0=ceil(-.0013*RPM+154.82)-(10-Load*10);
theta_b=ceil(.0038*RPM+40)-(10-Load*10);
end

theta_f = theta_0+theta_b; %Final Comb. Angle [degrees]
IVC = 0;           %Time [degrees] when Intake Valve Closes
EVO = 335;         %Time [degrees] when Exhaust Valve Opens
%
-----

%Engine Calculations Based On Previous Inputs
%Assumes Average Surface Area In Which Heat Transfer Occurs

A_p = (pi/4)*B^2;  %Cross Sectional Piston Area [m^2]
A_ch = 2*A_p;      %Cylinder Head Surface Area (in chamber)
V_d = N_cyl*A_p*L; %Displaced Volume Of Engine [m^3]
N = RPM/60;        %Converts RPM to RPS [1/s]
S_bar_p = 2*L*N;   %Calculates Mean Piston Speed [m/s]
a = L/2;           %Calculates Crank Radius (1/2 stroke) [m]
V_TDC = (V_d/(C_r-1))/N_cyl; %Calculates Clearance Volume [m^3]

```

```

V_BDC = (V_d/N_cyl)+V_TDC;           %Cyl. Volume At BDC [m^3]

%
-----

%Calculating Losses Due To Friction
%fmep (obtained from Blair) Based On Displacement, RPM
%
% if V_d>500*10^(-6)
%     fmep=(100000+350*L*RPM)*10^(-3);
% end
% if V_d<500*10^-6
%     fmep=(100000+100*(500-V_d*10^(-6))+350*L*RPM)*10^(-3);
% end

%For Motorcycles, Use "Rolling" Bearings (For Automobiles, Use Previous)
fmep = (250*L*RPM)*10^-3;

%Volumetric Efficiency Correction Factor
CF = correction( Load,RPM );

%Initial Preallocation Of Matrices (Second Preallocation In Loops Needs To
%Be Included (Do Not Delete)
V(1:360)=zeros;DV(1:360)=zeros;rho(1:360)=zeros;mu(1:360)=zeros;
C_k(1:360)=zeros;C_R(1:360)=zeros;X(1:360)=zeros;M_F(1:360)=zeros;
DX(1:360)=zeros;Re(1:360)=zeros;Nus(1:360)=zeros;h_g(1:360)=zeros;
DQ_w(1:360)=zeros;DQ(1:360)=zeros;Q(1:360)=zeros;DT(1:360)=zeros;
DP(1:360)=zeros;P(1:360)=zeros;T(1:360)=zeros;W_dot(1:360)=zeros;
W(1:360)=zeros;T_indicated(1:2)=zeros;Q_dot(1:360)=zeros;u(1:360)=zeros;
du(1:360)=zeros;cv(1:360)=zeros;m_b(1:360)=zeros;m_u(1:360)=zeros;
V_u(1:360)=zeros;V_b(1:360)=zeros;T_u(1:360)=zeros;T_b(1:360)=zeros;
A_u(1:360)=zeros;A_b(1:360)=zeros;DT_u(1:360)=zeros;gamma_u(1:360)=zeros;
u_u(1:360)=zeros;du_u(1:360)=zeros;cv_u(1:360)=zeros;DQ2(1:360)=zeros;
DQ_w2(1:360)=zeros;Q2(1:360)=zeros;

%
-----

%Fuel Inputs/Efficiencies

AF_ratio_stoich = 15.09;           %Theoretical Air Fuel Ratio (gravimetric)
% lambda = .85;                   %Excess Air Coefficient
if Load ==1
    lambda = .85;
end
if Load==.9
    lambda=.925;
end
if Load==.8
    lambda=.95;
end
if Load<.8
    lambda=.95;
end
end

```

```

AF_ratio_ac = lambda*AF_ratio_stoich; %Actual Air Fuel Ratio
AF_ratio_mol_sotich=14.7; %Molar Air_Fuel Ratio (Stoich)
AF_ratio_mol=lambda*AF_ratio_mol_sotich;
LHV = 44.6e6; %Lower Heating Value Of Fuel Mixture [J/kg]
eta_combmax = .95; %Assumed MAX COmb. Efficiency

```

```
%Predicts Combustion Efficiency (Reference To Blair)
```

```
eta_comb=eta_combmax*(-1.6082+4.6509*lambda-2.0764*lambda^2);
```

```
%Atmospheric Inputs
```

```

P_atm = 101325;
T_atm = 290;
P_BDC = Load*P_atm; %Inlet Pressure[Pa] Moscow,ID
R_air = 287; %Gas Constant For Air [J/kg-K]
gamma(1:360) = 1.4; %Preallocate Gamma Array (sets initial value)
T_w =350; %Assumed Wall Temperature (Reference Stone)
%
```

```
%Polynomials Used To Calculate Gamma As A Function Of RPM
```

```

a_1 = .692; a_2 = 39.17e-06; a_3 = 52.9e-09; a_4 = -228.62e-13;
a_5 = 277.58e-17;b_0 = 3049.33; b_1 = -5.7e-02; b_2 = -9.5e-05;
b_3 = 21.53e-09;b_4 = -200.26e-14;c_u = 2.32584; c_r = 4.186e-03;
d_0 = 10.41066; d_1 = 7.85125; d_3 = -3.71257;e_0 = -15.001e03;
e_1 = -15.838e03; e_3 = 9.613e03;f_0 = -.10329; f_1 = -.38656;
f_3 = .154226; f_4 = -14.763; f_5 = 118.27; f_6 = 14.503;
r_0 = -.2977; r_1 = 11.98; r_2 = -25442; r_3 = -.4354;

```

```
%
```

```
R=R_air/1000;
```

```
for k = 1:2
```

```
%Corrects Temperature Based On Exhaust Gas Residuals
```

```
if k==1
```

```
T_BDC = T_atm; %Assumed Inlet Temperature [K]
```

```
else
```

```
T_BDC=T_corr;
```

```
end
```

```
%Calculate Mass of Air In Cylinder/ Mass Of Fuel Based On AFR
```

```

rho_a = P_BDC/(R_air*T_BDC); %Air Density kg/m^3
m_a = rho_a*V_d; %Mass of Air In Cylinder [kg]
m_f = m_a/AF_ratio_ac; %Mass Of Fuel In Cylinder [kg]
m_c = m_a+m_f; %Mass In Cylinder

```

```
%Specifying Initial Conditions For Loops
```

```
%DV,DX,etc. Are Relative To Change In Theta (i.e. DV/Dtheta)
```

```
theta(1:360)=zeros; %Starting Crank Angle [deg]
```

```
V(1:360)=zeros; %Preallocate Volume Array
```

```
V(1)=V_BDC; %Starting Combustion Chamber Volume [m^3]
```

```
DV(1:360) = zeros; %Preallocate Change In Volume Array
```

```

DV(1) = 0; %Specifying Initial Change In Volume [m^3]
P(1:360)=P_BDC; %Preallocate Pressure Array
DP(1:360) = zeros; %Specifying Initial Change In Pressure
T(1:360)=zeros; %Preallocate Temperature Array
T(1) = T_BDC; %Inlet Temperature [K]
T_u(1)=T_BDC; %Initial Unburned Temperature[K]
DT(1:360) = zeros; %Specifying Initial Change In Temperature
DT_u(1:360)=zeros; %Preallocate Change In Unburned Temperature
gamma(1)=1.4; %Initial Gamma Input
gamma_u(1)=1.4; %Initial Gamma Input
X(1:360) = 0; %Preallocate Mass Burn Array
DX(1:360) = zeros; %Preallocate Change In Mass Burn Fraction [unitless]
DQ(1:360) = zeros; %Preallocate Heat Release Array
DQ2(1:360)=zeros; %Preallocate Two Zone Heat Release Array
Q(1:360)=zeros; %Preallocate Heat Array
Q2(1:360)=zeros; %Preallocate 2 zone Heat Array
M_F(1:360) = 0; %Preallocate Mass In Comubstion Chamber Array
rho(1:360) = zeros; %Preallocates Ideal Gas Law array
rho(1) = P(1)/(R_air*T(1)); %Initial Value Ideal Gas Array
mu(1:360)=zeros; %Preallocate Viscosity Array
mu(1)=7.457*10^(-6)+4.1547*10^(-8)*T_BDC-7.4793*10^(-12)*T_BDC^(2);
C_k(1:360)=zeros; %Preallocate Thermal COnductivity Array
C_k(1) = 6.1944*10^(-3)+7.3814*10^(-5)*T_BDC-1.2491*10^(-8)*T_BDC^(2);
C_R(1:360) = zeros; %Preallocate Radiation Coefficient Array
C_R(1) = 4.25*10^(-09)*((T(1)^4-T_w^4)/(T(1)-T_w)); %Initial Rad. Coeff
Re(1:360)=zeros; %Preallocate Reynolds Value Array
Re(1)=rho(1)*S_bar_p*B/mu(1); %Initial Reynolds Value
Nus(1:360)=zeros; %Preallocating Nusselt Number Array
Nus(1)=.49*Re(1)^(.7); %Initial Nusselt Number
h_g(1:360)=zeros; %Preallocate Heat Transfer Coefficient Array
h_g(1)=C_k(1)*Nus(1)/B; %Initial Heat Transfer Coefficient
s(1:360)=zeros; %Preallocates Distance Crank/Piston Axes Array
s(1) = -a*cosd(theta(1))+sqrt(1^2 - a^2*sind(theta(1))^2);%Initial Val.
W(1:360) = zeros; %Preallocate Work Array
W_dot(1:360) = zeros; %Preallocate Power Array
T_indicated(1:360) = zeros; %Preallocate Torque Array
Q_dot(1:360) = zeros; %Preallocate Heat Transfer Array
u(1:360) = zeros; %Preallocate Internal Energy Array
du(1:360) = zeros; %Preallocates Change In Internal Energy Array
cv(1:360) = zeros; %Preallocates Heat Capacity Array
DQ_w(1:360)=zeros; %Preallocate Convective Heat Loss Array
DQ_w2(1:360)=zeros; %Preallocate Convective Heat Loss Array 2 zone
m_b(1:360)= zeros; %Preallocate mass burned array
m_u(1:360)=m_c; %Preallocate unburned mass array
V_u(1:360)=zeros; %Preallocate unburned Volume Array
V_u(1) = V(1); %Initial Unburned Volume

```

```

%
```

---

```

theta=1:360;
```

```

for i = 2:360
```

```

    %Specifies Distance Between Crank/Piston Axes As A Function Of theta
    s = -a*cosd(theta(i))+sqrt(1^2 - a^2*sind(theta(i))^2);
    %Specifies Volume As A Function Of Crank Angle

```

```

V(i) = V_TDC + ((pi/4)*B^2)*(l + a - s);
%Specifies Change In Volume As A Function Of Crank Angle
DV(i) = V(i)-V(i-1);
%Calculates Density As A Function Of Crank Angle
rho(i) = P(i-1)/(R_air*T(i-1));
%Calculates Viscosity As A Function Of Temperature
mu(i)=7.457*10^(-6)+4.1547*10^(-8)*T(i-1)-7.4793*10^(-12)*T(i-1)^(2);
%Calculating Instantaneous Thermal Conductivity of Cylinder Gas
C_k(i) = 6.1944*10^(-3)+7.3814*10^(-5)*T(i-1)-1.2491*10^(-8)*T(i-
1)^(2);
%Calculating The Radiation Heat Transfer Coefficient
C_R(i) = 4.25*10^(-09)*((T(i-1)^4-T_w^4)/(T(i-1)-T_w));
%Instantaneous Surface Area (For Heat Transfer)
A = A_ch + A_p + pi*B*(l+a-s);
if i<=2
    A_u=A;
end

%
%
%Specifies Mass Fraction Burn As A Function Of Crank Angle (Weibe
Fcn.)
%Also Specifies Mass Of Fuel In Combustion Chamber As A Function Of
%Theta

    if theta(i)<theta_0
        X(i)=0;
    else
        X(i) = 1-exp(-5*((theta(i)-theta_0)/theta_b)^3);
        if theta(i) < theta_f
            M_F(i) = V(theta_0-1)*rho(theta_0-1)/(lambda*AF_ratio_mol);
        end
    end

%
%
%Specifies Change In Mass Fraction Burn As A Function Of Crank Angle
DX(i) = X(i) - X(i-1);

%
%
%Incorporating The Annand Method To Predict Heat Transfer
%Calculating Reynolds Number
Re(i)=rho(i)*S_bar_p*B/mu(i);
%Calculating Nusselt Number (constant=.26 two stroke, .49 4 stroke)
Nus(i)=.49*Re(i)^(.7);
%Calculating Heat Transfer Coefficient Using Annand Method
h_g(i)=C_k(i)*Nus(i)/B;
%Calculates Convective Losses Into Wall As A Function Of Crank Angle
DQ_w(i) = (h_g(i)+C_R(i))*A*(T(i-1)-T_w)*(60/(360*RPM));
%Calculates Change In Heat Transfer (total) As A Function Of Crank
%Angle
DQ(i) = eta_comb*LHV*M_F(i)*DX(i)-DQ_w(i);
%Calculates Total Heat Transfer (Per Cycle)
Q(i) = Q(i-1)+DQ(i);

```

```

%
%-----
%Specifies Pressure and Temperature Increases Between Intake Valve
%Closing and Exhaust Valve Opening
if IVC< theta(i)
    DT(i)=T(i-1)*(gamma(i-1)-1)*((1/(P(i-1)*V(i-1)))*DQ(i)...
        -(1/V(i-1))*DV(i));
    DP(i)=(-P(i-1)/V(i-1))*DV(i)+(P(i-1)/T(i-1))*DT(i);
    P(i) = P(i-1)+DP(i);
end
if EVO < theta(i)
    P(i) = P_atm;
end
if 200 < theta(i)
    if P(i)<=P_atm
        P(i)=P_atm;
    end
end
%
%-----
%Calculate Burned, Unburned Mass Fractions
m_b(i) = m_b(i-1)+DX(i)*m_c;    %Burned Mass
m_u(i) = m_u(i-1)-DX(i)*m_c;    %Unburned Mass
%Calculating Burned, Unburned Volumes
if theta(i)<=theta_0
    V_u(i)=N_cyl*V(i);
end
if theta(i)>theta_0
    V_u(i)=(m_u(i)*V_u(i-1))/m_u(i-1)*(P(i)/P(i-1))^(1/gamma_u(i-1));
end
V_b(i)=N_cyl*V(i)-V_u(i);
if V_b(i)<0
    V_b(i)=0;
end
%Calculating Burned, Unburned Temperatures
T_u(i)=P(i)*V_u(i)/(m_u(i)*R*1000);
if theta(i) <= theta_0+4
    T_b(i)=0;
end
if theta(i)>theta_0+4
    T_b(i)=P(i)*V_b(i)/(m_b(i)*R*1000);
end

%Calculate Unburned, Burned Areas Based On Volume Ratio
A_u(i)=A*(1-sqrt(X(i)));
A_b(i)=A*(X(i)/sqrt(X(i)));
DT_u(i)=T_u(i)-T_u(i-1);

%
%-----
%Returns Temperature Values To Beginning Of Loop
%Assumes Temperature Drops Back To ATM Temp After Exhaust Is Extracted
T(i) = T(i-1)+DT(i);
%Calculate The Residual Gas Fraction
%Assume A Polytropic Constant Of 1.3
R_frac = (1/C_r)*(P_BDC/P_atm)^(1/1.3)*(1/lambda);

```

```

%Calculates Cylinder Work [J] As A Function Of Crank Angle
%Treats Atmospheric Pressure As Reference State
W(i) = W(i-1)+(P(i)-P_atm)*DV(i);
%Calculates Power [kW] As A Function Of Crank Angle
W_dot(i)=(N_cyl*W(i)*N/N_r)/1000;
%Indicated Mean Effective Pressure
imep = CF*W_dot(360)*N_r*1000/(V_d*1000*N);
%Calculates Torque[N*m] As A Function Of Crank Angle
T_indicated(i) = (W_dot(i)*1000)/(2*pi*N);
%Calculates Heat Loss [kW] As A Function Of Crank Angle
Q_dot(i) = (N_cyl*Q(i)*N/N_r)/1000;

%
%-----
% The Following Section Of Code Calculates An Updated Value Of Gamma
% Using The "Polynomial Method" Developed By Krieger-Borman
% User Of This Code Must Be Careful Because Accuracy Of This Method
% Drops As The Fuel Mixture Becomes Increasingly Rich

%Calculates A,B Factors For Following Block Of Code
A_t = a_1*T(i)+a_2*T(i)^2+a_3*T(i)^3+a_4*T(i)^4+a_5*T(i)^5;
A_tu = a_1*T_u(i)+a_2*T_u(i)^2+a_3*T_u(i)^3+a_4*T_u(i)^4+a_5*T_u(i)^5;
B_t = b_0+b_1*T(i)+b_2*T(i)^2+b_3*T(i)^3+b_4*T(i)^4;
B_tu = b_0+b_1*T_u(i)+b_2*T_u(i)^2+b_3*T_u(i)^3+b_4*T_u(i)^4;
%Calculates Factor "D" As A Function Of lambda
D_lambda = d_0 + d_1*lambda^(-1)+ d_3*lambda^(-3);
%Calculates Factor "F" As A Function Of Temperature,lambda
E_TLambda = (e_0 + e_1*lambda^(-1)+ e_3*lambda^(-3))/T(i);
E_TLambdau = (e_0 + e_1*lambda^(-1)+ e_3*lambda^(-3))/T_u(i);
F_TPLambda = (f_0 + f_1*lambda^(-1) + f_3*lambda^(-3) + ...
    ((f_4 + f_5*lambda^(-1))/T(i))*log(f_6*P(i)));
F_TPLambdau = (f_0 + f_1*lambda^(-1) + f_3*lambda^(-3) + ...
    ((f_4 + f_5*lambda^(-1))/T_u(i))*log(f_6*P(i)));
%Calculates Correction Factor For Internal Energy
u_corr = c_u*exp(D_lambda +E_TLambda + F_TPLambda);
u_corr_u=c_u*exp(D_lambda +E_TLambdau + F_TPLambdau);
%Calculates Internal Energy As A Function Of Crank Angle
u(i) = A_t - B_t/lambda + u_corr;
u_u(i) = A_tu - B_tu/lambda + u_corr_u;
%Calculates Change In Internal Energy
du(i) = u(i) - u(i-1);
du_u(i) = u_u(i) - u_u(i-1);
%Calculates Heat Capacity "C_v" As A Function Of Crank Angle
cv(i) = du(i)/DT(i);
cv_u(i)=du_u(i)/DT_u(i);
%Calculates Correction Factor For "R" Value As A Function Of Crank
%Angle
R_corr = c_r*exp(r_0*log(lambda) + (r_1+r_2/T(i) + ...
    r_3*log(f_6*P(i)))/lambda);
R_corr_u = c_r*exp(r_0*log(lambda) + (r_1+r_2/T_u(i-1) + ...
    r_3*log(f_6*P(i)))/lambda);
%Calculates Actual "R" Value
R = .287 + .020/lambda + R_corr;
R_u = .287 + .020/lambda + R_corr_u;
%Calculates Actual Gamma Value And Returns To Beginning Of Code
gamma_u(i)=1+R_u/cv_u(i);
gamma(i) = 1 + R/cv(i);

```

```

    if gamma(i) < 1.2
        gamma(i) = 1.4;
        gamma_u(i) = 1.4;
    end

    if theta(i) >= EVO
        gamma(i) = 1.4;
        gamma_u(i) = 1.4;
    end

%
%-----

%Calculate Temperature Of Exhaust Based On Polytropic Relations
if EVO < theta(i)
    T(i) = T(EVO) * (P_BDC/P(EVO)) ^ ((gamma(i)-1)/gamma(i));
    T_b(i) = T_b(EVO) * (P_BDC/P(EVO)) ^ ((gamma(i)-1)/gamma(i));
end
end

%Calculates A Corrected Inlet Temperature Based On EGR
%T_corr = R_frac*T(360)+(1-R_frac)*T_BDC;
T_corr = T_BDC;
end
%
%-----

%Specified Outputs (On Matlab Screen)
W_dot_indicated = W_dot(360);
bmep = imep - fmep;
W_dot_ac = (bmep * V_d * 1000 * N / (N_r * 1000));
T_ac = W_dot_ac / (2 * pi * N * 10 ^ (-3));

%Calculated Mechanical Efficiency (Based On Previous Inputs)
eta_m = bmep / imep; %Calculates Mechanical Efficiency

%
%-----

%Calculates Brake Specific Fuel Consumption
m_ta = P_BDC * V_d / (R_air * T_BDC); %Calculate Trapped Air In Cylinder
eta_v = CF * ((m_ta) / (rho_a * V_d)); %Corrected Volumetric Efficiency
m_dot_f = N_cyl * M_F(theta_0) * (N / N_r); %Mass Flow Rate Of Fuel
m_dot_a = AF_ratio_ac * m_dot_f; %Mass Flow Rate Of Air
BSFC = (m_dot_f * 1000 * 3600) / (W_dot_ac); %BSFC [g/kW*h]
eta_f = 3600 / (BSFC * (LHV * 10 ^ (-6))); %Fuel Conversion Efficiency

%Calculate Emissions
T_NO = .875 * max(T_b); %Calculate Avg. Burn Temp
P_NO = max(P); %Assuming Pressure is peak
P_EXH = (P(EVO) + P_atm) / 2; %Calculating Exhaust Press.
[ PPM_NO ] = NOX(T_NO, P_atm, lambda, P_NO, T_BDC, P_BDC, P_EXH);
P_peak = max(P); %Peak Pressure
%disp('Percentage of Fuel Mass Reaching Exhaust')
[ HC ] = hydrocarbons(
R_frac, AF_ratio_ac, B, P_peak, imep, C_r, V_d, N_cyl, T_w, N );
HC = 100 * HC;

end

```

## **APPENDIX L: Spark- Timing Call Function**

```

clear all;
close all;
clc;

%Set Spark Angle Bounds
theta_st = 144;
theta_fin=160;
%Preallocate w
W_dot_ac(1:theta_st-theta_fin)=zeros;
T_ac(1:theta_st-theta_fin)=zeros;
%Changes Spark Angle As A Function Of I
theta_0=theta_st;
theta_o(1)=theta_0;
for i=1:(theta_fin-(theta_st))
    [W_dot_ac(i),T_ac(i)]=timingfunc(theta_0);
    theta_0=theta_0+1;
    theta_o(i)=theta_0;
end
%Plots Spark Advance and Corresponding Power Output
figure(1)
plot(theta_o,W_dot_ac,'k.')
grid on;
title('Spark Advance Vs. Power Output')
xlabel('Spark Advance [deg]')
ylabel('Power [kW]')

figure(2)
plot(theta_o,T_ac,'k.')
grid on;
title('Spark Advance Vs. Torque 6000 RPM')
xlabel('Spark Advance [deg]')
ylabel('Torque [N*m]')

```

## **APPENDIX M: Spark- Timing Function**

```

function [W_dot_ac,T_ac]=timingfunc(theta_0)

%University Of Idaho Engine Simulation
%Optimizes Spark Advance Vs. Power Output
%Engine Inputs
Load = .9;           %Engine Load (Affects Inlet Pressure)
RPM = 6000;         %Revolutions Per Minute [1/min]
L = (53.6/1000);    %Stroke of Engine [m]
B = (77/1000);      %Bore of Engine [m]
l = .0935;          %Length of Engine Connecting Rod [m]
N_cyl = 1;          %Number of Cylinders [unitless]
C_r = 12.5;         %Compression Ratio [unitless]
N_r = 2;            %Number of Revolutions Per Power Stroke
theta_b = 70;       %Combustion Burn Duration [degrees]
%theta_0 = 145;     %Crank Angle At Start of Combustion [degrees]
theta_f = theta_0+theta_b; %Final Comb. Angle [degrees]
IVC = 0;            %Time [degrees] when Intake Valve Closes
EVO = 314;          %Time [degrees] when Exhaust Valve Opens
%Engine Calculations Based On Previous Inputs
%Assumes Average Surface Area In Which Heat Transfer Occurs
A_p = (pi/4)*B^2;   %Cross Sectional Piston Area [m^2]
A_ch = A_p;         %Cylinder Head Surface Area (in chamber)
V_d = N_cyl*A_p*L;  %Displaced Volume Of Engine [m^3]
N = RPM/60;         %Converts RPM to RPS [1/s]
S_bar_p = 2*L*N;    %Calculates Mean Piston Speed [m/s]
a = L/2;            %Calculates Crank Radius (1/2 stroke) [m]
V_TDC = (V_d/(C_r-1))/N_cyl; %Calculates Clearance Volume [m^3]
V_BDC = (V_d/N_cyl)+V_TDC; %Cyl. Volume At BDC [m^3]
%Calculating Losses Due To Friction
%fmep (obtained from Blair) Based On Displacement, RPM
% if V_d>500*10^(-6)
%   fmep=(100000+350*L*RPM)*10^(-3);
% end
% if V_d<500*10^-6
%   fmep=(100000+100*(500-V_d*10^(-6))+350*L*RPM)*10^(-3);
% end
%Initial Preallocation Of Matrices (Second Preallocation In Loops Needs To
%Be Included (Do Not Delete)
V(1:360)=zeros;DV(1:360)=zeros;rho(1:360)=zeros;mu(1:360)=zeros;
C_k(1:360)=zeros;C_R(1:360)=zeros;X(1:360)=zeros;M_F(1:360)=zeros;
DX(1:360)=zeros;Re(1:360)=zeros;Nus(1:360)=zeros;h_g(1:360)=zeros;
DQ_w(1:360)=zeros;DQ(1:360)=zeros;Q(1:360)=zeros;DT(1:360)=zeros;
DP(1:360)=zeros;P(1:360)=zeros;T(1:360)=zeros;W_dot(1:360)=zeros;
W(1:360)=zeros;T_indicated(1:2)=zeros;Q_dot(1:360)=zeros;u(1:360)=zeros;
du(1:360)=zeros;cv(1:360)=zeros;m_b(1:360)=zeros;m_u(1:360)=zeros;
V_u(1:360)=zeros;V_b(1:360)=zeros;T_u(1:360)=zeros;T_b(1:360)=zeros;
A_u(1:360)=zeros;A_b(1:360)=zeros;DT_u(1:360)=zeros;gamma_u(1:360)=zeros;
u_u(1:360)=zeros;du_u(1:360)=zeros;cv_u(1:360)=zeros;DQ2(1:360)=zeros;
DQ_w2(1:360)=zeros;DQ2(1:360)=zeros;
%For Motorcycles, Use "Rolling" Bearings (For Automobiles, Use Previous)
fmep = (250*L*RPM)*10^-3;

%Volumetric Efficiency Correction Factor
CF = correction( Load,RPM );

```

```

%Fuel Inputs/Efficiencies
AF_ratio_stoich = 15.09; %Gravimetric Air Fuel Ratio (Stoich)
AF_ratio_mol_sotich=14.7; %Molar Air Fuel Ratio (Stoich)
lambda = 1.1; %Excess Air Coefficient
AF_ratio_ac = lambda*AF_ratio_stoich; %Actual Air Fuel Ratio
AF_ratio_mol=lambda*AF_ratio_mol_sotich;
LHV = 44.6e6; %Lower Heating Value Of Fuel Mixture [J/kg]
eta_combmax = .95; %Assumed MAX COmb. Efficiency
%Predicts Combustion Efficiency (Reference To Blair)
eta_comb=eta_combmax*(-1.6082+4.6509*lambda-2.0764*lambda^2);
%Atmospheric Inputs
P_atm = 101300;
P_BDC = Load*P_atm; %Inlet Pressure[Pa] Moscow, ID
R_air = 287; %Gas Constant For Air [J/kg-K]
gamma(1:360) = 1.4; %Preallocate Gamma Array (sets initial value)
T_w =350; %Assumed Wall Temperature (Reference Stone)
%Polynomials Used To Calculate Gamma As A Function Of RPM
a_1 = .692; a_2 = 39.17e-06; a_3 = 52.9e-09; a_4 = -228.62e-13;
a_5 = 277.58e-17;b_0 = 3049.33; b_1 = -5.7e-02; b_2 = -9.5e-05;
b_3 = 21.53e-09;b_4 = -200.26e-14;c_u = 2.32584; c_r = 4.186e-03;
d_0 = 10.41066; d_1 = 7.85125; d_3 = -3.71257;e_0 = -15.001e03;
e_1 = -15.838e03; e_3 = 9.613e03;f_0 = -.10329; f_1 = -.38656;
f_3 = .154226; f_4 = -14.763; f_5 = 118.27; f_6 = 14.503;
r_0 = -.2977; r_1 = 11.98; r_2 = -25442; r_3 = -.4354;
R=R_air/1000;
for k = 1:2
%Corrects Temperature Based On Exhaust Gas Residuals
if k<2
T_BDC = 290; %Assumed Inlet Temperature [K]
else
T_BDC=T_corr;
end
%Calculate Mass of Air In Cylinder/ Mass Of Fuel Based On AFR
rho_a = P_atm/(R_air*T_BDC); %Air Density kg/m^3
m_a = rho_a*V_d; %Mass of Air In Cylinder [kg]
m_f = m_a/AF_ratio_ac; %Mass Of Fuel In Cylinder [kg]
m_c = m_a+m_f; %Mass In Cylinder
%Specifying Initial Conditions For Loops
%DV,DX,etc. Are Relative To Change In Theta (i.e. DV/Dtheta)
theta(1:360)=zeros; V(1:360)=zeros;V(1)=V_BDC;DV(1:360) = zeros;DV(1) = 0;
P(1:360)=P_BDC;DP(1:360) =zeros;T(1:360)=zeros;T(1) = T_BDC;T_u(1)=T_BDC;
DT(1:360) = zeros; DT_u(1:360)=zeros;gamma(1)=1.4;gamma_u(1)=1.4;
X(1:360) = 0;DX(1:360) = zeros;DQ(1:360) = zeros;DQ2(1:360)=zeros;
Q(1:360)=zeros;Q2(1:360)=zeros;M_F(1:360) = 0;rho(1:360) = zeros;
rho(1) = P(1)/(R_air*T(1));mu(1:360)=zeros;
mu(1)=7.457*10^(-6)+4.1547*10^(-8)*T_BDC-7.4793*10^(-12)*T_BDC^(2);
C_k(1:360)=zeros; %Preallocate Thermal COnductivity Array
C_k(1) = 6.1944*10^(-3)+7.3814*10^(-5)*T_BDC-1.2491*10^(-8)*T_BDC^(2);
C_R(1:360) = zeros;
C_R(1) = 4.25*10^(-09)*((T(1)^4-T_w^4)/(T(1)-T_w)); %Initial Rad. Coeff
Re(1:360)=zeros; Re(1)=rho(1)*S_bar_p*B/mu(1);Nus(1:360)=zeros;
Nus(1)=.49*Re(1)^(.7);h_g(1:360)=zeros; h_g(1)=C_k(1)*Nus(1)/B;
s(1:360)=zeros;
s(1) = -a*cosd(theta(1))+sqrt(1^2 - a^2*sind(theta(1))^2);%Initial Val.
W(1:360) = zeros;W_dot(1:360) = zeros; T_indicated(1:360) = zeros;
Q_dot(1:360) = zeros; u(1:360) = zeros;

```

```

du(1:360) = zeros;cv(1:360) = zeros;DQ_w(1:360)=zeros;DQ_w2(1:360)=zeros;
m_b(1:360)= zeros;m_u(1:360)=m_c; V_u(1:360)=zeros;V_u(1) = V(1);
theta=1:360;
for i = 2:360
    %Specifies Distance Between Crank/Piston Axes As A Function Of theta
    s = -a*cosd(theta(i))+sqrt(l^2 - a^2*sind(theta(i))^2);
    %Specifies Volume As A Function Of Crank Angle
    V(i) = V_TDC +((pi/4)*B^2)*(l + a - s);
    %Specifies Change In Volume As A Function Of Crank Angle
    DV(i) = V(i)-V(i-1);
    %Calculates Density As A Function Of Crank Angle
    rho(i) = P(i-1)/(R_air*T(i-1));
    %Calculates Viscosity As A Function Of Temperature
    mu(i)=7.457*10^(-6)+4.1547*10^(-8)*T(i-1)-7.4793*10^(-12)*T(i-1)^(2);
    %Calculating Instantaneous Thermal Conductivity of Cylinder Gas
    C_k(i) = 6.1944*10^(-3)+7.3814*10^(-5)*T(i-1)-1.2491*10^(-8)*T(i-
1)^(2);
    %Calculating The Radiation Heat Transfer Coefficient
    C_R(i) = 4.25*10^(-09)*((T(i-1)^4-T_w^4)/(T(i-1)-T_w));
    %Instantaneous Surface Area (For Heat Transfer)
    A = A_ch + A_p + pi*B*(l+a-s);
    %Specifies Mass Fraction Burn As A Function Of Crank Angle (Weibe
Fcn.)
    if theta(i)<theta_0
        X(i)=0;
    else
        X(i) = 1-exp(-5*((theta(i)-theta_0)/theta_b)^3);
        if theta(i) < theta_f
            M_F(i) = V(theta_0-1)*rho(theta_0-1)/(lambda*AF_ratio_mol);
        end
    end
    DX(i) = X(i) - X(i-1);
    %Interpolate Between Data Points
    if theta(i) < theta_f
        dvmf=V(floor(theta_0-1))-V(floor(theta_0));
        dth=theta_0-floor(theta_0);
        vmf=V(floor(theta_0-1))+dvmf*dth;
        drho=rho(floor(theta_0-1))-rho(floor(theta_0));
        rhomf=rho(floor(theta_0-1))+drho*dth;
        M_F(i) = vmf*rhomf/(lambda*AF_ratio_ac);
    end
    %Incorporating The Annand Method To Predict Heat Transfer
    %Calculating Reynolds Number
    Re(i)=rho(i)*S_bar_p*B/mu(i);
    %Calculating Nusselt Number (constant=.26 two stroke, .49 4 stroke)
    Nus(i)=.49*Re(i)^(.7);
    %Calculating Heat Transfer Coefficient Using Annand Method
    h_g(i)=C_k(i)*Nus(i)/B;
    %Calculates Convective Losses Into Wall As A Function Of Crank Angle
    DQ_w(i) = (h_g(i)+C_R(i))*A*(T(i-1)-T_w)*(60/(360*RPM));
    DQ_w2(i) = ((h_g(i)+C_R(i))*A_b(i-1)/N_cyl*(T_b(i-1)-T_w)...
        +(h_g(i)+C_R(i))*A_u(i-1)/N_cyl*(T_u(i-1)-T_w))*(60/(360*RPM));
    %Calculates Change In Heat Transfer (total) As A Function Of Crank
    %Angle
    DQ(i) = eta_comb*LHV*M_F(i)*DX(i)-DQ_w(i);
    DQ2(i) = eta_comb*LHV*M_F(i)*DX(i)-DQ_w2(i);

```

```

%Calculates Total Heat Transfer (Per Cycle)
Q(i) = Q(i-1)+DQ(i);
Q2(i) = Q2(i-1)+DQ2(i);
%Specifies Pressure and Temperature Increases Between Intake Valve
%Closing and Exhaust Valve Opening
if IVC< theta(i)
    DT(i)=T(i-1)*(gamma(i-1)-1)*((1/(P(i-1)*V(i-1)))*DQ(i)...
        -(1/V(i-1))*DV(i));
    DP(i)=(-P(i-1)/V(i-1))*DV(i)+(P(i-1)/T(i-1))*DT(i);
    P(i) = P(i-1)+DP(i);
end
if EVO < theta(i)
    P(i) = P_atm;
end
if 200 < theta(i)
    if P(i)<=P_atm
        P(i)=P_atm;
    end
end
%Returns Temperature Values To Beginning Of Loop
%Assumes Temperature Drops Back To ATM Temp After Exhaust Is Extracted
T(i) = T(i-1)+DT(i);
%Calculate The Residual Gas Fraction
%Assume A Polytropic Constant Of 1.3
R_frac = (1/C_r)*(P_BDC/P_atm)^(1/1.3)*(1/lambda);
%Calculates Cylinder Work [J] As A Function Of Crank Angle
%Treats Atmospheric Pressure As Reference State
W(i) = W(i-1)+(P(i)-P_atm)*DV(i);
%Calculates Power [kW] As A Function Of Crank Angle
W_dot(i)=(N_cyl*W(i)*N/N_r)/1000;
%Indicated Mean Effective Pressure
imep = CF*W_dot(360)*N_r*1000/(V_d*1000*N);
%Calculates Torque[N*m] As A Function Of Crank Angle
T_indicated(i) = (W_dot(i)*1000)/(2*pi*N);
%Calculates Heat Loss [kW] As A Function Of Crank Angle
Q_dot(i) = (N_cyl*Q(i)*N/N_r)/1000;
%Calculates A,B Factors For Following Block Of Code
A_t = a_1*T(i)+a_2*T(i)^2+a_3*T(i)^3+a_4*T(i)^4+a_5*T(i)^5;
A_tu = a_1*T_u(i)+a_2*T_u(i)^2+a_3*T_u(i)^3+a_4*T_u(i)^4+a_5*T_u(i)^5;
B_t = b_0+b_1*T(i)+b_2*T(i)^2+b_3*T(i)^3+b_4*T(i)^4;
B_tu = b_0+b_1*T_u(i)+b_2*T_u(i)^2+b_3*T_u(i)^3+b_4*T_u(i)^4;
%Calculates Factor "D" As A Function Of lambda
D_lambda = d_0 + d_1*lambda^(-1)+ d_3*lambda^(-3);
%Calculates Factor "F" As A Function Of Temperature,lambda
E_TLambda = (e_0 + e_1*lambda^(-1)+ e_3*lambda^(-3))/T(i);
E_TLambdau = (e_0 + e_1*lambda^(-1)+ e_3*lambda^(-3))/T_u(i);
F_TPLambda = (f_0 + f_1*lambda^(-1) + f_3*lambda^(-3) + ...
    ((f_4 + f_5*lambda^(-1))/T(i))*log(f_6*P(i)));
F_TPLambdau = (f_0 + f_1*lambda^(-1) + f_3*lambda^(-3) + ...
    ((f_4 + f_5*lambda^(-1))/T_u(i))*log(f_6*P(i)));
%Calculates Correction Factor For Internal Energy
u_corr = c_u*exp(D_lambda +E_TLambda + F_TPLambda);
u_corr_u=c_u*exp(D_lambda +E_TLambdau + F_TPLambdau);
%Calculates Internal Energy As A Function Of Crank Angle
u(i) = A_t - B_t/lambda + u_corr;
u_u(i) = A_tu - B_tu/lambda + u_corr_u;

```

```

%Calculates Change In Internal Energy
du(i) = u(i) - u(i-1);
du_u(i) = u_u(i) - u_u(i-1);
%Calculates Heat Capacity "C_v" As A Function Of Crank Angle
cv(i) = du(i)/DT(i);
cv_u(i)=du_u(i)/DT_u(i);
%Calculates Correction Factor For "R" Value As A Function Of Crank
%Angle
R_corr = c_r*exp(r_0*log(lambda) + (r_1+r_2/T(i) + ...
    r_3*log(f_6*P(i)))/lambda);
R_corr_u = c_r*exp(r_0*log(lambda) + (r_1+r_2/T_u(i-1) + ...
    r_3*log(f_6*P(i)))/lambda);
%Calculates Actual "R" Value
R = .287 + .020/lambda + R_corr;
R_u = .287 + .020/lambda + R_corr_u;
%Calculates Actual Gamma Value And Returns To Beginning Of Code
gamma_u(i)=1+R_u/cv_u(i);
gamma(i) = 1 + R/cv(i);
    if gamma(i)<1.2
        gamma(i)=1.4;
        gamma_u(i)=1.4;
    end
if theta(i)>=EVO
    gamma(i)=1.4;
    gamma_u(i)=1.4;
end
%Calculate Temperature Of Exhaust Based On Polytropic Relations
if EVO < theta(i)
T(i)=T(EVO)*(P_BDC/P(EVO))^( (gamma(i)-1)/gamma(i));
T_b(i)=T_b(EVO)*(P_BDC/P(EVO))^( (gamma(i)-1)/gamma(i));
end
end
%Calculates A Corrected Inlet Temperature Based On EGR
%T_corr = R_frac*T(360)+(1-R_frac)*T_BDC;
T_corr=T_BDC;
end
%Specified Outputs Actual Power and Torque
W_dot_indicated=W_dot(360);
bmep = imep-fmep;
W_dot_ac = (bmep*V_d*1000*N/(N_r*1000));
T_ac = W_dot_ac/(2*pi*N*10^(-3));
%Calculated Mechanical Efficiency (Based On Previous Inputs)
eta_m = bmep/imep; %Calculates Mechanical Efficiency
%
%Calculates Brake Specific Fuel Consumption
m_ta = P_BDC*V_d/(R_air*T_BDC); %Calculate Trapped Air In Cylinder
eta_v = CF*((m_ta)/(rho_a*V_d)); %Corrected Volumetric Efficiency
m_dot_f = N_cyl*M_F(theta_0)*(N/N_r); %Mass Flow Rate Of Fuel
m_dot_a = AF_ratio_ac*m_dot_f; %Mass Flow Rate Of Air
BSFC = (m_dot_f*1000*3600)/(W_dot_ac); %BSFC [g/kW*h]
eta_f = 3600/(BSFC*(LHV*10^(-6))); %Fuel Conversion Efficiency
end

```

**APPENDIX N: A User's Guide to Two-Zone Simulations**

## 1.) Engine Inputs and Spark Advance

```

%Engine Inputs
Load = 1;           %Engine Load (Affects Inlet Pressure)
RPM = 13000;       %Revolutions Per Minute [1/min]
L = (44.5/1000);   %Stroke of Engine [m]
B = (65.5/1000);   %Bore of Engine [m]
l = .0905;         %Length of Engine Connecting Rod [m]
N_cyl = 4;         %Number of Cylinders [unitless]
C_r = 12.4;        %Compression Ratio [unitless]
N_r = 2;           %Number of Revolutions Per Power Stroke
theta_b = 85;      %Combustion Burn Duration [degrees]
theta_0 = 145;     %Crank Angle At Start of Combustion [degrees]
theta_f = theta_0+theta_b; %Final Comb. Angle [degrees]
IVC = 0;          %Time [degrees] when Intake Valve Closes
EVO = 335;        %Time [degrees] when Exhaust Valve Opens

```

*For engine inputs, reference the yellow block of code. For spark advance and valves, reference the blue block.*

### Engine Inputs

- Q. How would you use the model to simulate load?
- A. Adjust “load” in the engine inputs (0=no throttle, 1= full throttle)
- Q. In simulating engine performance, what geometric constraints must be put into the model?
- A. The stroke, bore, connecting rod length, number of cylinders, and compression ratio (L, B, and l, N\_cyl, and C\_r) must be specified for a particular engine.

### Spark Advance and Valves

- Q. How does the burn duration (theta\_b) manipulate model outputs?
- A. The burn duration (theta\_b) specifies the width of the burn profile. This can be used to tailor the pressure trace.
- Q. How can spark-advance (theta\_0) be optimized?
- A. Spark-advance (theta\_0) can be optimized relative to power or torque. This model specifies 180 degrees as top-dead-center.
- Q. How can a camshaft be adapted to the model?
- A. IVC and EVO represent the closing of the intake valve and the opening of the exhaust valve. This simulates the action of a camshaft without taking gas dynamics and valve profiles into account.

## 2.) Fuel Inputs

```
AF_ratio_stoich = 15.09;    %Theoretical Air Fuel Ratio
lambda = .95;             %Excess Air Coefficient
AF_ratio_ac = lambda*AF_ratio_stoich; %Actual Air Fuel Ratio
LHV = 44.6e6;             %Lower Heating Value Of Fuel Mixture [J/kg]
eta_combmax = .98;       %Assumed MAX COmb. Efficiency
```

*The necessary fuel inputs are highlighted in yellow.*

- Q. How can one adjust the air-fuel ratio in the model?
- A. The excess-air-coefficient (lambda) must be changed within the model. The actual air-fuel-ratio is calculated from a stoichiometric balance and lambda.
- Q. How is combustion efficiency adjusted in the model?
- A. The maximum combustion efficiency is used to calculate the actual combustion efficiency in the model. This can be adjusted according to a specific engine.
- Q. Can alternative fuels be used in this model?
- A. The model can calculate outputs with alternative fuels. However, all chemical balances and inputs are in terms of iso-octane. Changing all of these inputs would require some time.

## 3.) Atmospheric Inputs

```
%Atmospheric Inputs
P_atm = 101325;
T_atm = 290;
P_BDC = Load*P_atm;    %Inlet Pressure[Pa] Moscow,ID
R_air = 287;           %Gas Constant For Air [J/kg-K]
gamma(1:360) = 1.4;    %Preallocate Gamma Array (sets initial value)
T_w = 350;             %Assumed Wall Temperature (Reference Stone)
```

*The atmospheric inputs are highlighted in yellow.*

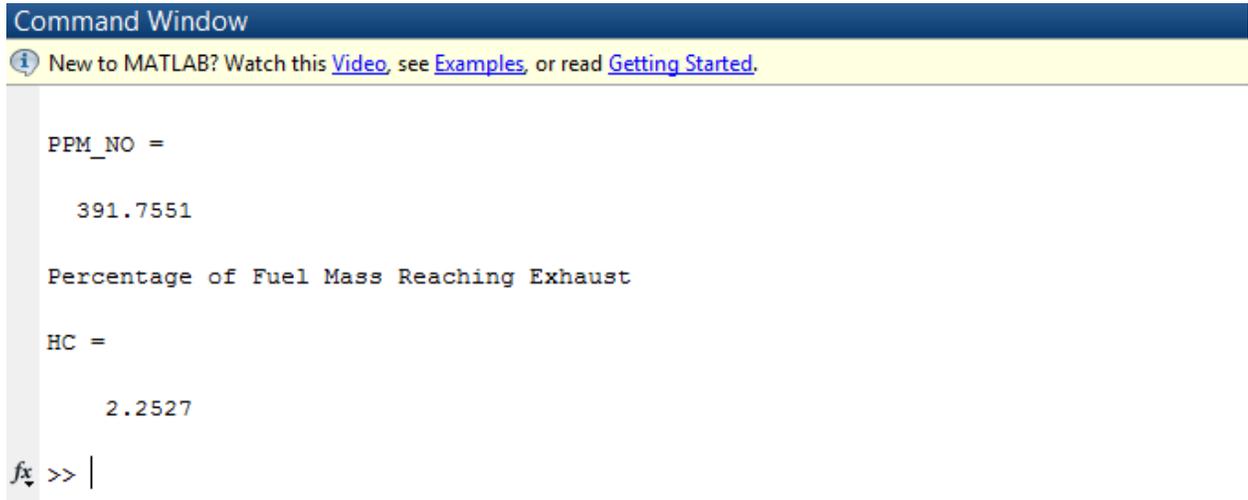
- Q. Do atmospheric inputs change model outputs?
- A. The atmospheric temperature and pressure have to be adjusted according to the weather. This can affect predictions by approximately 5-10%.

## 4.) Running a Simulation

- Q. What m-files are necessary to run a simulation?
- A. Four m-files are necessary to run a simulation (YZ250FTwoZoneAnnand.m, NOX.m, correction.m, and hydrocarbons.m). These files should be saved in the same directory in order to run the simulation (i.e. a flash drive).
- Q. After inputting geometric and atmospheric constraints, how is a simulation run?
- A. Open YZ250FTwoZoneAnnand.m and press “run” in the editor window.

## 5.) Model Outputs

- The amount of NO and hydrocarbons will be displayed in the MATLAB command window. NO will be displayed in particles-per-million, and hydrocarbons will be displayed in a percentage of input fuel. An image of the command window can be seen below.



The screenshot shows the MATLAB Command Window interface. At the top, there is a blue header bar with the text "Command Window". Below this is a yellow banner with an information icon and the text "New to MATLAB? Watch this [Video](#), see [Examples](#), or read [Getting Started](#)." The main area of the window is white and contains the following text:

```

PPM_NO =

    391.7551

Percentage of Fuel Mass Reaching Exhaust

HC =

    2.2527

fx >> |

```

- All simulated variables will be stored to the right of the command window in the “workspace”. Everything from brake-specific fuel consumption (BSFC) to fmep losses can be found in this variables list.
- The model will also produce 6 plots. These range from the zone-temperatures to the pressure trace. These plots can be commented out in the last few lines of YZ250FTwoZoneAnnand.m if desired.

## **APPENDIX O: Optimizing Spark-Timing**

# Spark-Timing Optimization Activity

## Learning Objectives

1. Understand how the burn duration changes relative to the air-fuel ratio and engine speed.
2. Describe how the burn duration is influenced by laminar flame speed.
3. Identify how timing is influenced by engine speed and load.

## Governing Equations

### 1. Weibe Function (constants 3 and 5 are used)

```
X(i) = 1-exp(-5*((theta(i)-theta_0)/theta_b)^3);
```

### 2. Annand's Heat Transfer Prediction Method

```
%Calculates Convective Losses Into Wall As A Function Of Crank Angle
DQ_w(i) = (h_g(i)+C_R(i))*A*(T(i-1)-T_w)*(60/(360*RPM));
%Calculates Change In Heat Transfer (total) As A Function Of Crank
%Angle
DQ(i) = eta_comb*LHV*M_F(i)*DX(i)-DQ_w(i);
```

### 3. Apparent Heat Release Model

```
DT(i)=T(i-1)*(gamma(i-1)-1)*((1/(P(i-1)*V(i-1)))*DQ(i)...
-(1/V(i-1))*DV(i));
DP(i)=(-P(i-1)/V(i-1))*DV(i)+(P(i-1)/T(i-1))*DT(i);
P(i) = P(i-1)+DP(i);
```

- Choose to optimize spark-timing relative to torque or power.
- Place the main MATLAB model within a function (be sure to save the function as the correct name i.e. “timingfunc”)

```
function [W_dot_ac,T_ac]=timingfunc(theta_0)
%PLACE MATLAB MODEL HERE
end
```

The “function” statement says to input “theta\_0” values and output “W\_dot\_ac” and “T\_ac” values. Theta\_0 is the spark advance, and W\_dot\_ac and T\_ac are power and torque values, respectively.

- Create a script that calls the function.

```
clear all;
close all;
clc;

%Set Spark Angle Bounds
theta_st = 144;
theta_fin=160;
%Preallocate w
W_dot_ac(1:theta_st-theta_fin)=zeros;
T_ac(1:theta_st-theta_fin)=zeros;
%Changes Spark Angle As A Function Of I
theta_0=theta_st;
theta_o(1)=theta_0;
for i=1:(theta_fin-(theta_st))
    [W_dot_ac(i),T_ac(i)]=timingfunc(theta_0);
    theta_0=theta_0+1;
    theta_o(i)=theta_0;
end
```

Theta\_st and theta\_fin specify the range over which timing is optimized. A “for” loop is used to specify each angle over the specified range. Notice that the function is called on the first line inside of the “for” loop.

- Specify plotting statements in the call script

```
figure(1)
plot(theta_o,W_dot_ac,'k.')
grid on;
title('Spark Advance Vs. Power Output')
xlabel('Spark Advance [deg]')
ylabel('Power [kW]')

figure(2)
plot(theta_o,T_ac,'k.')
grid on;
title('Spark Advance Vs. Torque 6000 RPM')
xlabel('Spark Advance [deg]')
ylabel('Torque [N*m]')
```

These plotting statements create plots relative to torque and power outputs.

- Specify engine inputs in the main model (shown below).
- Comment-out the spark-advance, “clear all”, “close all”, and “clc” within the main model.
- Assume a burn duration based on the critical thinking questions.

```
%Engine Inputs
Load = .9;           %Engine Load (Affects Inlet Pressure)
RPM = 6000;         %Revolutions Per Minute [1/min]
L = (53.6/1000);    %Stroke of Engine [m]
B = (77/1000);      %Bore of Engine [m]
l = .0935;          %Length of Engine Connecting Rod [m]
N_cyl = 1;          %Number of Cylinders [unitless]
C_r = 12.5;         %Compression Ratio [unitless]
N_r = 2;            %Number of Revolutions Per Power Stroke
theta_b = 70;       %Combustion Burn Duration [degrees]
%theta_0 = 145;     %Crank Angle At Start of Combustion [degrees]
theta_f = theta_0+theta_b; %Final Comb. Angle [degrees]
IVC = 0;            %Time [degrees] when Intake Valve Closes
EVO = 314;          %Time [degrees] when Exhaust Valve Opens
```

## **APPENDIX P: BSFC and Emissions Mapping Activity**

# BSFC and Emissions Maps

## Learning Objectives

1. Define volumetric efficiency. Describe how volumetric efficiency changes as a function of engine speed.
2. Identify how the air-fuel ratio changes at different engine operating points.
3. Categorize the mechanisms that cause NO and HC emissions formation.

## Governing Equations

### 1. Brake-Specific Fuel Consumption

$$\text{BSFC} = (\dot{m}_f \cdot 1000 \cdot 3600) / (\dot{W}_{ac}); \quad \% \text{BSFC} \text{ [g/kW}\cdot\text{h]}$$

### 2. Water-Gas Shift Equilibrium Constant

*%Calculate Equilibrium Constant At Given Temperature (Water Gas Shift)*

$$K_{wgs} = \exp(2.743 - 1.761 \cdot 10^3 / T_{NO} - 1.611 \cdot 10^6 / (T_{NO}^2) + .2803 \cdot 10^9 / (T_{NO}^3));$$

### 3. NO Concentration

*%Calculate Change in NO Concentration as Function of Time*

$$dNO/dt = 2 \cdot k_1 \cdot x_{O_e} \cdot x_{N_2_e} \cdot P_{BDC} / (R_u \cdot T_{NO});$$

#### 4.) NO Residence Time

```
%Calculate residence time
t_NO=(8*10^(-16)*T_NO*exp(58300/T_NO))/(P_NO/101325)^(1/2);
```

#### 5.) HC Emissions Index Due To Crevices

```
SF_crevice = 5443*(P_peak/imep)*(V_crevice/(V_d/N_cyl))*(1/T_w)*...
f_unburned*f_vapor*f_mod;
```

#### 6.) HC Oil Absorption Emissions Index

```
SF_wall = 63024*(1/imep)*(1/((AF_ratio_ac/15.09)*...
14.7*10^(.0082*T_w)*B))*P_ideal;
```

- Use the provided BSFC call function to predict BSFC and emissions maps.
- Open the main MATLAB program “BSFCAnnand” and make sure that “Load”, “RPM”, “theta\_b”, and “theta\_0” are commented-out in the engine inputs block. Input the selected engine parameters in this block (i.e. bore, stroke, connecting rod length, etc.)

```
%Engine Inputs
%Load = 1;           %Engine Load (Affects Inlet Pressure)
%RPM = 16500;       %Revolutions Per Minute [1/min]
L = (53.6/1000);    %Stroke of Engine [m]
B = (77/1000);      %Bore of Engine [m]
l = .0935;          %Length of Engine Connecting Rod [m]
N_cyl = 1;          %Number of Cylinders [unitless]
C_r = 12.5;         %Compression Ratio [unitless]
N_r = 2;            %Number of Revolutions Per Power Stroke
%theta_b = 85;      %Combustion Burn Duration [degrees]
%theta_0 = 145;     %Crank Angle At Start of Combustion [degrees]
```

- Create spark-advance and burn-duration equations relative to engine speed and engine load. Keep in mind that the air-fuel ratio changes as a function of load as well.

```
if Load==1
theta_0=ceil(-.0013*RPM+154.82);
theta_b=ceil(.0038*RPM+40);
end
if Load<1
theta_0=ceil(-.0013*RPM+154.82)-(10-Load*10);
theta_b=ceil(.0038*RPM+40)-(10-Load*10);
end
```

General equations were created by inputting RPM and spark-advance values in an array in Microsoft Excel. The burn duration and spark-advance should be carefully selected based on the air-fuel ratio at each load and engine speed.

- Adjust friction losses relative to the chosen engine (lines 56-67 in BSFCAnnand).

```

%Calculating Losses Due To Friction
%fmep (obtained from Blair) Based On Displacement, RPM
%
% if V_d>500*10^(-6)
%   fmep=(100000+350*L*RPM)*10^(-3);
% end
% if V_d<500*10^-6
%   fmep=(100000+100*(500-V_d*10^(-6))+350*L*RPM)*10^(-3);
% end

%For Motorcycles, Use "Rolling" Bearings (For Automobiles, Use Previous)
fmep = (250*L*RPM)*10^-3;

```

The shown fmep losses are linear equations that can be found in “Design and Simulation of Four-Stroke Engines”. These should be adjusted based on the type of engine. General four-stroke engines should use the commented-out statements, while race engines should use the statement shown.

- Adjust the if-then statements (lines 92-103 of BSFCAnnand) relating the air-fuel ratio to engine load. Remember, lambda is the excess air coefficient.

```
AF_ratio_stoich = 15.09; %Theoretical Air Fuel Ratio (gravimet
% lambda = .85; %Excess Air Coefficient
if Load ==1
    lambda = .85;
end
if Load==.9
    lambda=.925;
end
if Load==.8
    lambda=.95;
end
if Load<.8
    lambda=.95;
end
```

- If the chosen engine has EGR, this will need to be activated in the main script (lines 395-397 in BSFCAnnand).

```
%Calculates A Corrected Inlet Temperature Based On EGR  
%T_corr = R_frac*T(360)+(1-R_frac)*T_BDC;  
T_corr = T_BDC;
```

If the engine has EGR, the boxed line should be commented, and the line above should be uncommented. This adjusts the combustion chamber temperature based on a residual gas fraction.

- Open the function “LOAD2”
- Set RPM at the desired starting point (iterations start 1000 RPM past this)

```
function [BSFC,T_ac,N_RPM,PPM_NO,HC,eta_f]=LOAD2 (Load)
    RPM = 1500;           %Sets Starting Pt. For Loop
    N_RPM = 12;          %Fifteen RPM Data Sets
    BSFC(1:N_RPM)=zeros; %Preallocate Array
    T_ac(1:N_RPM)=zeros; %Preallocate Array
    PPM_NO(1:N_RPM)=zeros;
    for i = 1:N_RPM
        RPM = RPM+1000;
        [BSFC(i),T_ac(i),PPM_NO(i),HC(i),eta_f(i)]=BSFCAnnand(RPM,Load);
    end
end
```

With the RPM set at 1500, calculations will begin at 2500 RPM because of how the “for” loop is set up.

- Set the number of RPM intervals relative to the desired maximum RPM.

```
function [BSFC,T_ac,N_RPM,PPM_NO,HC,eta_f]=LOAD2(Load)

RPM = 1500;      %Sets Starting Pt. For Loop
N_RPM = 12;     %Fifteen RPM Data Sets
BSFC(1:N_RPM)=zeros; %Preallocate Array
T_ac(1:N_RPM)=zeros; %Preallocate Array
PPM_NO(1:N_RPM)=zeros;
for i = 1:N_RPM
    RPM = RPM+1000;
    [BSFC(i),T_ac(i),PPM_NO(i),HC(i),eta_f(i)]=BSFCAnnand(RPM,Load);
end
end
```

Since the RPM loop starts at 2500,  
the loops will continue until 13500  
RPM.

- Open “BSFCcall2” and adjust engine load parameters (if desired).
- Don’t attempt to create loads in increments of less than .1 (this causes issues in the main numerical program). Don’t attempt to decrease the load below 30%. This will result in negative torques and in-operable conditions.

```
Load = 1.1;           %Start High, Decreased By .1 Each Iteration
N_load = 7;          %Six Loads
% bsfc(1:N_RPM,1:N_load)=zeros;
% Torque(1:N_RPM,1:N_load)=zeros;
```

Keep “Load” at 1.1. This initializes the “for” loop. N\_load sets the number of load operating points. The current setting allows loads from 100% to 40%. This was bounded as shown to prevent non-existent operating points. It’s suggested to leave these variables as shown.

- Open “correction” and adjust the volumetric efficiency according to the chosen engine.

```
function [ CF ] = correction( Load,RPM )  
if Load<=1  
    %CF = (-3*10^(-9))*RPM^2+5*10^(-5)*RPM+.7088;  
    CF=-8*10^(-9)*RPM^2+.000135*RPM+.31944;  
    if Load<=.9  
        CF=-8*10^(-9)*RPM^2+.000135*RPM+.31944;  
        %CF = CF-(1-Load)/4;  
    end  
end  
end
```

The volumetric efficiency can be adjusted based on the chosen type of engine and the output torque curves. The shown correction factor was created in excel by plotting efficiencies as a function of RPM. The volumetric efficiency can also be adjusted relative to load using an if-then statement (this code didn't adjust relative to load)

- Run a simulation using BSFCcall2. The program will generate emissions and BSFC maps.
- Adjust the plot statements to sufficiently resolve the maps on your chosen engine.

```
figure(1)
contourf(RPM,Torque,bsfc,45)
xlabel('RPM')
ylabel('Torque [N*m]')
title('YAMAHA YZ250F BSFC [g/kW-h] MAP')
colorbar
colormap jet
caxis([325 550])
```

Contourf creates a plot using filled contours (the number 45 represents the number of levels). Colorbar inserts a bsfc key to the right of the contour plot. Caxis adjusts the minimum and maximum BSFC contours (this will need to be adjusted relative to the chosen engine).

- Analyze the results based on the governing equations and critical thinking questions. Do the maps make sense?
- If the results don't make sense, make sure all engine inputs are correct (i.e. number of cylinders, bore, stroke, etc.)

**APPENDIX Q: Heat Release Class Notes**

# Heat Release Modeling

## 1.) Differentiate the Ideal Gas Law With Respect to Crank-Angle

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

## 2.) Rearrange To Find Instantaneous Change in Pressure, Numerically Integrate To Find Pressure

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

$$P(i) = P(i-1) + \frac{dP(i)}{d\theta}$$

## 5.) First Law Of Thermodynamics

$$\frac{dU}{d\theta} = \frac{dQ}{d\theta} - \frac{dW}{d\theta} \rightarrow m_{\text{dot}} C_p \left(\frac{dT}{d\theta}\right) = \frac{dQ}{d\theta} + P \frac{dV}{d\theta}$$

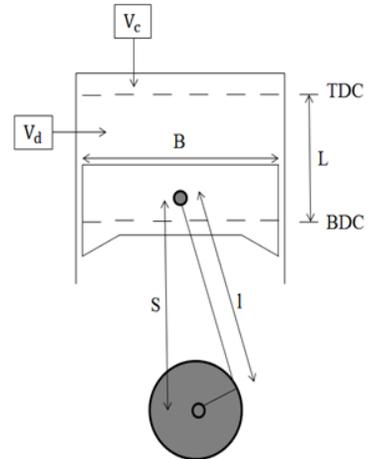
### 3.) Find the Instantaneous Cylinder Volume

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

$$s = a \cos(\theta) + (l^2 - a^2 \sin^2 \theta)^{\frac{1}{2}}$$

$$\frac{dV}{d\theta} = V(i) - V(i-1)$$

$l$  = connecting rod length  
 $L$  = stroke  
 $a = \frac{L}{2}$



**6.) Rearrange to Find Instantaneous Change In Temperature, Integrate To Find Temperature**

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

$$T(i) = T(i - 1) + \frac{dT}{d\theta}(i)$$

**7.) Gross Heat Transfer As A Function Of Crank-Angle**

$$\frac{dQ}{d\theta} = \text{LHV} \frac{dX_b}{d\theta} - \frac{dQ_w}{d\theta}$$

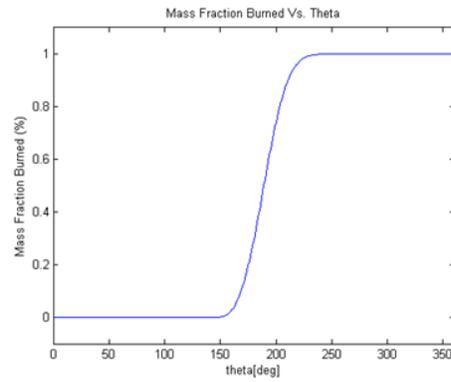
LHV = Lower Heating Value

### 8.) The Weibe Function

$$X_b = 1 - \exp\left(-A \left(\frac{(\theta - \theta_0)}{\theta_b}\right)^m\right)$$

$A = 5, m = 2$  (for initial iterations)

$$\frac{dX_b}{d\theta} = X_b(i) - X_b(i - 1)$$



**9.) Predict Convective Wall Losses Using Newton's Law of Cooling**

$$\frac{dQ_w}{dt} = h_w A_w (T - T_w)$$

$h_w$  = convective heat transfer coefficient

$A_w$  = Instantaneous HX area

$T$  = Combustion Gas Temperature

$T_w$  = Cylinder Wall Temp. (Assume 350 [K])

**10.) Change Differential Equation To Crank-Angle Basis**

$$dt = \frac{d\theta}{360 \text{ RPM}} \frac{60}{1} = \frac{d\theta}{360 N}$$

$$N = \text{engine speed} \left[ \frac{\text{rev}}{\text{s}} \right]$$

**10.) Predict Heat Transfer Coefficient Using Woschni's Method**

$$h_w = 3.26B^{-.2}P^{.8}T^{-.55}w^{.8}$$

B = Cylinder Bore

w = average gas velocity

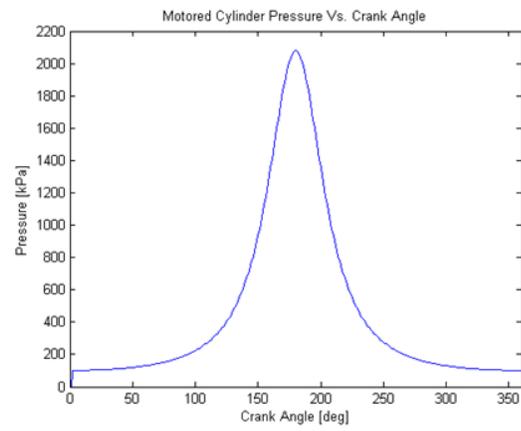
$$w = \left[ C_1 \bar{S}_p + \frac{C_2 (V_d T_r)}{P_r V_r} (P - P_m) \right]$$

$T_r, P_r, V_r$  = Reference Values (@ BDC)

$\bar{S}_p$  = mean piston velocity

$C_1 = 2.28$  and  $C_2 = 0$  (Compression)  
 $C_1 = 2.28$  and  $C_2 = 3.24 \times 10^{-3}$  (Combustion)

$P_m$  = motored cylinder pressure



$$P_m = P_r \left( \frac{V_r}{V} \right)^n$$

$n$  = polytropic index (assume 1.3)