Implementation of an Integrative MATLAB Engine Model as a Final Project in an Internal Combustion Engines Course

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

Theron White

Major Professor: Steven Beyerlein, Ph.D.

Committee Members: Daniel Cordon, Ph.D.; Edwin Odom, Ph.D.

Department Administrator: Steven Beyerlein, Ph.D.

May 2016

Authorization to Submit Thesis

This Thesis of Theron O. White, submitted for the degree of Master of Science with a major in Mechanical Engineering and titled “Implementation of an Integrative MATLAB Engine Model as a Final Project in an Internal Combustion Engines Course," has been reviewed in final form. Permission, as indicated by the signatures and dates below, is now granted to submit final copies to the College of Graduate Studies for approval.

Major Professor ________________________________Date___________________
Steven Beyerlein, Ph.D.

Committee Members: ________________________________Date___________________
Daniel Cordon, Ph.D.

________________________________Date___________________
Edwin Odom, Ph.D.

Department Administrator: ________________________________Date___________________
Steven Beyerlein, Ph.D.

Abstract

There is insufficient time within a single technical elective to learn principles of internal combustion engine operation as well as specialized simulation tools such as GT Suite or Kiva. A number of authors have recognized this constraint, and they have structured their internal combustion engine text around use of programming languages such as FORTRAN, C++, and MATLAB®. This paper reports on how the capabilities of MATLAB® have been synergized with learning activities and homework assignments to set the stage for a successful final engine simulation project. The MATLAB® code involved in this effort can accept basic input parameters such as bore, stroke, compression ratio, spark advance, throttle position, RPM, air/fuel equivalence ratio, and volumetric efficiency and output power and torque using the Wiebe function and bulk temperature. In addition to power and torque predictions, the model described here uses a two-zone heat release model to predict brake specific fuel consumption as well as volumetric emissions. An engine-specific volumetric efficiency map is suggested to ensure accurate results at all throttle positions. Students are pleasantly surprised at how well their customized MATLAB® model is able to match published performance data from a diversity of applications: mopeds, racing motorcycles, sedans, and SUVs. The model has been used successfully by students for simulating engine results from 50 cm^3 to 5 liters. These simulations produced power and torque results very close to the actual outputs of the engine based on factory engine data. These simulations also yield brake specific fuel consumption and brake specific emissions maps that agree with empirical data. A positive impact on professional skill development is demonstrated through a series of course surveys.

Acknowledgements

I would like to thank Dr. Steve Beyerlein. He initially interested me in internal combustion through the ME 433 IC engines course and then furthered my knowledge and interest by giving me the opportunity to work with an advanced engine model. He was also instrumental in keeping the project moving with his encouragement, insight, and support.

I would also like to thank the department of Mechanical Engineering and Idaho Engineering Works for their financial support. Without funding none of this would have been possible.

Table of Contents

List of Figures

List of Tables

Key Variables

Variable	Definition	Units
A, A_w	Instantaneous Heat Transfer Area	m^2
AF_{ac}	Actual Air-Fuel Ratio	unit-less
AF_{stoich}	Stoichometric 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
Nu	Nusselt Number	unit-less
m	Cylinder Mass	kg

m_a	Mass of Air	kg
m_b	Burned Mass	kg
m_c	Combustion Mass	kg
$m_{crevice}$	Crevice Mass	kg
m_f	Mass of Fuel	kg
m_{HC}	Mass of Hydrocarbons	kg
m_{oil}	Mass of Oil on Chamber Wass	kg
m_u	Unburned Mass	kg
N	Revolutions Per Second	$\frac{rev}{s}$
n	Polytropic Index	unit-less
P	Cylinder Pressure	Pa
P_{inlet}	Inlet Pressure	atm
P_e	Exhaust Pressure	Pa
P_{fuel}	Partial Pressure of Absorbed Fuel	atm
P_m	Mtored Pressure	Pa
P_{peak}	Peak Cylinder Pressure	atm
P_r	Reference Pressure	Pa
ΔP_c	Pressure Rise due to Combustion	Pa
ΔP_v	Pressure Rise due to Volumetric Expansion	Pa
Q	Heat Energy Transfer	J
Q_{ch}	Chemical Heat Release	J
Q_w	Heat Trasnfer to the Wall	J
R	Universal Gas Constant	$\frac{J}{kg * K}$
Re	Reynolds Number	unit-less
RPM	Revolutions Per Minute	$\frac{rev}{min}$
s	Distance Between Crank, Piston Axes	m
$\overline{S_p}$	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 Absored Fuel	unit-less
X_b	Mass Fraction Burned	unit-less
α	Fraction of Dissociation	unit-less
δ_{oil}	Oil Layer Thickness	m
ε	Fuel Hydrogen to Carbon Ratio	unit-less
η_c	Combustion Efficiency	unit-less
γ	Specific Heat Ratio	unit-less
λ	Excess Air Coefficient	unit-less
μ	Viscosity	$\frac{kg}{m * s}$
ρ	Gas Density	$\frac{kg}{m^3}$
ρ_{oil}	Oil Density	$\frac{kg}{m^3}$
θ	Crank Angle	Degrees
θ_b	Burn Duration	Degrees
θ_o	Spark Advance	Degrees

Chapter 1. Introduction

An internal combustion engine systems course is a popular technical elective subject that provides opportunity for cultivating skills in thermal systems modeling [1,2]. At the University of Idaho, this elective is taken by 4th year Mechanical Engineering majors as well as by a number of graduate students just beginning their Master's program. A first course in thermodynamics and a first course in heat transfer are the pre-requisites. An underlying goal of the course is to extend many of the analysis techniques featured in introductory thermodynamics courses that otherwise yield system performance estimates for gas cycles that have large margins of error, increasing students' confidence in applying thermo-fluid concepts. In doing so, it is important to recognize that working knowledge about a broad array of thermophysical processes inside internal combustion engines along with considerable grit in computer modeling is required to produce engine simulations that mimic real-world performance.

This thesis explores the implementation of a customizable MATLAB® simulation capable of modeling authentic engine performance as a take-home, final project in the aforementioned internal combustion engine technical elective [3]. This culminating project spans the final two weeks of the course and is intended to synthesize understanding of thermodynamic principles, air induction, fuel-air mixing, in-cylinder heat release phenomena, pollutant formation mechanisms, frictional losses, heat rejection, and interpretation of engine performance data.

The engine model used in this final project has been slowly improved, with more functionality being added every year. Adding additional functionality to the MATLAB code will allow students to predict brake specific emissions. This will allow students to further optimize their engine models by minimizing the emissions produced for the fuel burned.

The purpose of designing this model was to allow students in the ME 433 course to be able to combine everything they'd learned in the course into an easy-to-use model that will allow students to perform quick adjustments to their engine parameters for optimization. Since the equations individually take significant time to program into an equation solver and are difficult to solve by hand, the advantages of having a single package for students to use is apparent.

This thesis will be separated into eight chapters. Chapter 2 will be a review of the previous work done on the engine model and better documentation of the code for an external audience. Chapter 3 will have details on the modifications done to the engine model and the significance of those modifications. Chapter 4 will introduce the goals of the internal combustion engine course, leading into the implementation of the final project. Chapter 5 contains details of the implementation of the engine model. Chapter 6 describes the final project of the ME 433 course and covers the details of the results produced by students. Chapter 7 contains assessments of the overall effectiveness of the course and mini-project. Chapter 8 contains the conclusions drawn from the course and an overview of future work.

Chapter 2. Analysis of Previous Engine Model

The work done for this thesis was based off an engine model built by Jeremy Cuddihy in 2014 [3]. This model uses a single zone bulk temperature function to predict power and torque outputs, and a two-zone heat transfer model to accurately predict emissions. A two-zone heat-release model is considered as a tool for incorporating effects of valve timing, spark timing, fuel enrichment, leading to estimation of realistic engine performance and emissions maps. From the 1990’s through 2010, a more basic version of the model was supplied to students as a FORTRAN executable and inputs could be changed using batch files. This version required students to express inputs using non-dimensional parameters and make manual adjustments for volumetric efficiency and engine friction within an EXCEL spreadsheet. It did not include any emissions predictions. The following is an improved layout of the governing equations for the current engine model, based on peer review feedback from the 2016 SAE technical paper [4].

2.1 Equation Development of Engine Model

The simplest approach in engine modeling is to treat the cylinder contents as a solitary fluid or zone [5]. The ideal gas equation forms the basis for a single-zone engine model:

$$PV = mRT \quad (1)$$

where P is pressure, V is volume, m is mass, R is the ideal gas constant, and T is temperature. Upon differentiating equation (1) with respect to the change in engine crank angle (dθ) equation (2) is obtained:

$$\frac{d}{d\theta}(PV) = \frac{d}{d\theta}(mRT) \quad (2)$$

Upon using the chain rule and rearranging, equation (3) is obtained [1]:

$$\frac{dP}{d\theta} = \left(-\frac{P}{V}\right)\left(\frac{dV}{d\theta}\right) + \left(\frac{P}{T}\right)\left(\frac{dT}{d\theta}\right) \quad (3)$$

where P, V, and T are the instantaneous pressure, volume, and temperature, respectively.

The first law of thermodynamics is expressed as:

$$\Delta U = Q - W \quad (4)$$

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

Upon differentiating equation (4) with respect to dθ, equation (5) is obtained:

$$\frac{dU}{d\theta} = \frac{dQ}{d\theta} - \frac{dW}{d\theta} = mC_v\left(\frac{dT}{d\theta}\right) \quad (5)$$

where C_v is the specific heat of combustion chamber gasses. Upon dividing the specific heat by the universal gas constant, equation (6) is obtained:

$$\frac{C_v}{R} = \frac{C_v}{C_p - C_v} = \frac{1}{\gamma - 1} \qquad (6)$$

where γ is the specific heat ratio. Equations (7) and (8) can be used to describe the formation of work and the net heat input [6]:

$$\frac{dW}{d\theta} = P\left(\frac{dV}{d\theta}\right) \quad (7)$$

$$\frac{dQ}{d\theta} = \eta_c LHV\left(\frac{dX_b}{d\theta}\right) - \frac{dQ_w}{d\theta} \qquad (8)$$

where $\left(\frac{dQ}{d\theta}\right)$ is the gross change in input heat, η_c is the combustion efficiency, 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. The instantaneous change in mass-fraction burned is defined using the Weibe function as follows [3]:

$$X_{b(\theta)} = 1 - exp\left[-a\left(\frac{\theta - \theta_o}{\theta_b}\right)^{k+1}\right] \qquad (9)$$

where a and k are adjustable constants, θ is the instantaneous crank angle, θ_o is the spark-advance, and θ_b is the burn duration. Upon substituting equations (6) and (7) into equation (5), the instantaneous change in temperature is defined as:

$$\frac{dT}{d\theta} = T(\gamma - 1)\left[\left(\frac{1}{PV}\right)\left(\frac{dQ}{d\theta}\right) - \left(\frac{1}{V}\right)\left(\frac{dV}{d\theta}\right)\right] \qquad (10)$$

With the change in temperature as a function of crank angle defined, the heat input from the fuel can be used to find the change in pressure as a function of crank angle [6]:

$$Q_{in} = \eta_c LHV\left(\frac{1}{AF_{ac}}\right)\left(\frac{P}{RT}\right)V_d \quad (11)$$

where AF_{ac} is the actual air-fuel ratio.Substituting equations (8) and (9) into equation (3) and rearranging results in:

$$\frac{dP}{d\theta} = \left(\frac{-\gamma P}{V}\right)\left(\frac{dV}{d\theta}\right) + \left(\frac{\gamma-1}{V}\right)Q_{in}\frac{dX_b}{d\theta} + (\gamma - 1)\left(\frac{1}{V}\right)\left(\frac{dQ_w}{d\theta}\right) \qquad (12)$$

This can be combined with Woschni’s or Annand’s heat transfer prediction method to complete the single-zone 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 [7]. A method developed by Fox et al. [7] 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:

$$T(\theta) = T_{EVO}\left(\frac{P_{BDC}}{P_{EVO}}\right)^{\frac{\gamma-1}{\gamma}} \quad (13)$$

where T_{EVO} and P_{EVO} are the exhaust temperature and pressure, respectively, upon opening the exhaust valve, and P_{BDC} is the cylinder 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 [8]:

$$f = \frac{1}{C_r}\left(\frac{P_e}{P_i}\right)^{\frac{1}{\gamma}}\left(\frac{1}{\lambda}\right) \quad (14)$$

where C_r is the compression ratio, P_e is the exhaust gas pressure, P_i is the inlet gas pressure, and λ is the air/fuel equivalence ratio. The residual gas fraction can be used to create a corrected or modified intake temperature [7]:

$$T_{corr} = T_{CE} * f + T_{BDC} * (1 - f) \quad (15)$$

where T_{CE} is the ending cycle temperature. The corrected temperature is then returned as an input to the single-zone model, thus correcting the inlet temperature and simulating EGR.

Two-zone models are closely related to the equations derived in the single-zone model. The two-zone model bisects the combustion chamber into a burned and unburned zone, thereby increasing the accuracy of heat transfer and emissions predictions. With a known cylinder mass, the Weibe function can be used to split combustion chamber contents into unburned and burned masses [9]:

$$m_b(\theta) = m_b(\theta - 1) + \frac{dX_b}{d\theta}(\theta)m_c \quad (16)$$

$$m_u(\theta) = m_u(\theta - 1) - \frac{dX_b}{d\theta}(\theta)m_c \quad (17)$$

where m_c is the total mass of air and fuel within the cylinder, m_u is the instantaneous unburned mass within the cylinder, and m_b is the instantaneous burned mass within the cylinder. With known unburned and burned masses, the corresponding volumes can be obtained. Blair [9] suggests using polytropic relationship to calculate the unburned cylinder volume [9]:

$$V_u(\theta) = \left(\frac{m_u(\theta)V_u(\theta-1)}{m_u(\theta-1)}\right)\left(\frac{P(\theta)}{P(\theta-1)}\right)^{-\frac{1}{\gamma_u(\theta)}} \qquad (18)$$

where $\gamma_u(\theta)$ is the specific heat ratio of the unburned region. With the unburned volume defined, the burned volume is determined using the relationship [6][9]:

$$V(\theta) = V_b(\theta) + V_u(\theta) \qquad (19)$$

where $V(\theta)$ is the instantaneous total cylinder volume, which is calculated from the engine geometry and crank angle. The ideal-gas assumption then carries to each constituent zone, where the burned and unburned temperatures are defined as:

$$T_b(\theta) = \frac{P(\theta)V_b(\theta)}{m_b(\theta)R} \qquad (20)$$

$$T_u(\theta) = \frac{P(\theta)V_u(\theta)}{m_u(\theta)R(\theta)} \qquad (21)$$

where $P(\theta)$ is the bulk cylinder pressure and R is the specific gas constant for air. Lastly, the heat transfer surface area is bisected into unburned and burned regions [6]:

$$A_u(\theta) = A(\theta)\left(1 - \left(X_b(\theta)\right)^{\frac{1}{2}}\right) \qquad (22)$$

$$A_b(\theta) = A(\theta)\left(\frac{X_b(\theta)}{\left(X_b(\theta)\right)^{\frac{1}{2}}}\right) \qquad (23)$$

where $A(\theta)$ is the overall heat transfer surface area, which is calculated from the engine geometry and crank angle.

The burned zone temperature (equation 20) is used to predict NO emissions because it is elevated in comparison to the bulk, or single-zone, temperature. This elevated temperature is more accurate in describing NO formation at the threshold of the burned and unburned regions in the combustion chamber. The basis for the NO emissions model is the following chemical reactions where reactions R.1 – R.8 are governed by carbon monoxide kinetics and reactions R.9-R.11 are governed by the general Zeldovich mechanism[8][10]:

$$H_2 + O \leftrightarrow OH + H, \quad \text{(R.1)}$$

$$OH + O \leftrightarrow O_2 + H, \quad \text{(R.2)}$$

$$OH + H_2 \leftrightarrow H_2O + H, \quad \text{(R.3)}$$

$$\mathrm{OH} + \mathrm{OH} \leftrightarrow \mathrm{H_2O} + \mathrm{O}, \quad \text{(R.4)}$$

$$CO + OH \leftrightarrow CO_2 + H, \quad \text{(R.5)}$$

$$H_2O + M \leftrightarrow OH + H + M, \quad \text{(R.6)}$$

$$H_2 + M \leftrightarrow H + H + M, \quad \text{(R.7)}$$

$$O_2 + M \leftrightarrow O + O + M, \quad \text{(R.8)}$$

$$N + O_2 \leftrightarrow NO + O, \quad \text{(R.9)}$$

$$N + NO \leftrightarrow N_2 + O, \quad \text{(R.10)}$$

$$N + OH \leftrightarrow NO + H \quad \text{(R.11)}$$

The general Zeldovich mechanism dominates in high temperature combustion and is expressed as the elementary reactions [8][9][11][12][13]:

$$O + N_2 \overrightarrow{k_{1_f}} NO + N, \quad \text{(R.12)}$$

$$N + O_2 \overrightarrow{k_{2_f}} NO + O. \quad \text{(R.13)}$$

where $\mathrm{k}_{1_\mathrm{f}}$ and $\mathrm{k}_{2_\mathrm{f}}$ represent the forward reaction rate coefficients in each constituent chemical reaction. Reorganizing and expressing reactions 12 and 13 results in:

$$\frac{d[NO]}{dt} = k_{1_f}[O][N_2] + k_{2_f}[N][O_2] \quad (24)$$

$$\frac{d[N]}{dt} = k_{1_f}[O][N_2] - k_{2_f}[N][O_2] \qquad (25)$$

where [i] represents the respective constituent concentration. By assuming that that the change in concentration of Nitrogen over time is zero, reactions (12) and (13) are decoupled, thereby allowing the concentration of NO as a function of time to be expressed as [11]:

$$\frac{d[NO]}{dt} = 2k_{1_f}[N_2]_e[O]_e \qquad (26)$$

where the subscript e represents an equilibrium concentration. According to Turns [11], the forward reaction rate is defined as:

$$k_{1_f}\left(\frac{cm^3}{gmol-s}\right) = (1.82 * 10^{14})\, exp\left(-\frac{38370}{T_b(K)}\right) \qquad (27)$$

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

$$[O]_e = \frac{K_o[O_2]_e^{\frac{1}{2}}}{(R_uT)^{\frac{1}{2}}} \qquad (28)$$

where R_u is the universal gas constant and K_o the equilibrium constant for the dissociation of oxygen and is defined as [1] :

$$K_o\left(Pa^{\frac{1}{2}}\right) = 3.6\ x\ 10^3 exp\left(-\frac{31090}{T_b}\right) x\ (101325)^{\frac{1}{2}} \qquad (29)$$

The most basic HC crevice models use empirical equations, along with known crevice volumes, to predict the percentage of unburned fuel that bypassed the combustion process. Hamrin and Heywood modeled the mass of HC escaping combustion as a function of peak cylinder pressure, residual gas fraction, exhaust gas recirculation, and engine geometry.

The mass of the crevice gasses is defined as [14]:

$$m_{crev} = \frac{P*V_{crev}*MW_{gas}}{R*T_{coolant}} \quad (30)$$

where V_{crev} is the crevice volume and is defined by the engine geometry, MW_{gas} is the molecular weight of the crevice gas, and $T_{coolant}$ is the coolant temperature (K), which can be assumed to be equivalent to that of the coolant.

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

$$f_{unburned} = 1 - f - EGR \quad (31)$$

$$f_{vapor} = \frac{1}{1+AF_{ac}} \quad (32)$$

where EGR is the percentage of recirculated exhaust gas.

A modification factor is used 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:

$$f_{mod} = 1 - 0.858\left(\frac{d_{splug}}{B}\right) \quad (33)$$

where d_{splug} is the offset distance (m) of the spark plug from the central axis of the cylinder, and B is the cylinder bore (m).

Combining equations (25)-(28) with known engine parameters, the crevice emissions index is defined as [14]:

$$\mathrm{SF_{crevice}} = 5443\left(\frac{\mathrm{P_{peak}}}{\mathrm{IMEP}}\right)\left(\frac{\frac{\mathrm{V_{crev}}}{\mathrm{V_d}}}{\mathrm{N_{cyl}}}\right)\left(\frac{1}{\mathrm{T_{coolant}}}\right)(\mathrm{f_{unburned}})(\mathrm{f_{vapor}})(\mathrm{f_{mod}}) \quad (34)$$

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

Assuming a constant oil density and residual layer thickness, and utilizing Henry's law, the HC emissions index due to oil layer absorption and desorption is defined as [14]:

$$SF_{wall} = \frac{63024\left(\frac{1}{IMEP}\right)\left(\frac{1}{AF_{ac}\left(10^{0.0082*T_{oil}}\right)B}\right)\left(P_{inlet}+P_{inlet}*R_c^{\gamma}\right)}{2} \quad (35)$$

where P_{inlet} is the inlet pressure (atm), T_{oil} is the oil temperature and R_c is the compression ratio.

In determining 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. The fraction of HC oxidation within the cylinder is defined as:

$$f_{ox} = 1 - \left(\frac{P(70°ATDC)}{P_{max}}\right)\left(\frac{T_{HC}}{T_{HC_{adj}}}\right)^3 \quad (36)$$

where T_{HC} is the log mean average temperature, P_{max} is the maximum cylinder pressure, and $T_{HC_{adj}}$ is the adjusted log mean average temperature. Hamrin and Heywood [14] have developed curve-fitted equations that define the adjusted log mean average temperature.

Using the emissions indices, an overall HC formation mechanism is defined as [14]:

$$HC_{formed} = \left(SF_{crevice} * (1 - f_{ox}) + SF_{wall} * (1 - f_{ox})\right) * f_{ox} * \left(1 - f_{ox_{exh}}\right) \quad (37)$$

where HC_{formed} is expressed as a percentage of the input fuel mass and $f_{ox_{exh}}$ is the fraction of HC oxidation within the exhaust. Hamrin and Heywood [14] have developed curve-fitted equations that define the fraction of hydrocarbons oxidized within the exhaust.

Chapter 3. Expanded Emissions Functionality

3.1 Brake Specific Emissions

The first modification to the MATLAB engine model was to convert the hydrocarbon and nitrogen oxide emissions data from a volumetric basis to a brake specific basis. Brake specific emissions are a calculation of how efficiently an engine is producing pollutants based on its power generation. Calculation of this value allows engines of different sizes and speeds to be accurately compared.

For NO, this was done by taking the molar constituents used in the volumetric NO function and converting them to a mass basis:

$$m_{const} = x_{const} * MM_{const} \quad (38)$$

where m_{const} is the mass of the constituent in the exhaust, x_{const} is the molar fraction of the constituent, and MM_{const} is the molar mass of the constituent. After constituents were converted to a mass basis, the total mass of the constituents was added together:

$$m_{all} = \sum m_{const} \quad (39)$$

Once the total mass was determined, the mass fraction of NO was found by dividing the mass of NO in the exhaust by the total mass of all constituents:

$$w_{NO} = m_{NO}/m_{all} \quad (40)$$

where $w_{NO}$is the mass fraction of NO in the exhaust. The function output the mass fraction of NO to the main script of the code. The mass flow rate of NO was then calculated by multiplying the mass fraction of NO found in the BSNOX function by the total mass flow rate into the system:

$$\dot{m}_{NO} = w_{NO} * \left(\dot{m}_a + \dot{m}_f\right) \quad (41)$$

where $\dot{m}_{NO}$ is the mass flow rate of NO, $\dot{m}_a$ is the mass flow rate of air, and $\dot{m}_f$ is the mass flow rate of fuel. After the mass flow rate of NO was calculated, it was converted from kilograms to grams and multiplied by 3600 seconds to find the total mass that would be produced in one hour. This was divided by the work rate of the engine to find the brake specific NO:

$$BSNOX = \frac{\dot{m}_{NO} * 1000^{g}/_{kg} * 3600\ ^{s}/_{hr}}{\dot{W}} \qquad (42)$$

Since the program output for hydrocarbons was a percentage of fuel mass passing through the engine to the exhaust unburned, it was a simple conversion to mass flow rate of hydrocarbons:

$$\dot{m}_{HC} = HC * \dot{m}_f \qquad (43)$$

where HC is the percent of fuel mass reaching the exhaust. After calculating the mass flow of unburned hydrocarbons, the same equation was used again to convert the output to a brake specific basis:

$$BSHC = \frac{\dot{m}_{HC} * 1000^{g}/_{kg} * 3600\ ^{s}/_{hr}}{\dot{W}} \qquad (44)$$

3.2 Process for Obtaining Emissions Maps

In order to construct an emissions map, several data points must be recorded. A minimum of fifteen data points is recommended to get accurate emissions maps. The first step is to determine the RPM operating range of the engine selected. After setting low and high bounds for RPM, use the low and high bounds as well as three engine speeds between them. In addition to engine speeds, three throttle positions should be used. The recommended throttle positions for emissions maps are 30%, 60%, and 100% throttle. After determining the matrix of engine speeds and throttles, values are then input into the engine model one at a time, with the results being recorded in an excel spreadsheet. Figure 1 shows the input of values, and figures 2 and 3 show the output and recording of emissions, fuel consumption, and torque and power data.

```
%Engine Inputs
Load = 1;        %Engine Load (Affects Inlet Pressure)
RPM = 11700;         %Revolutions Per Minute [1/min]
L = (55/1000);     %Stroke of Engine [m]
B = (76/1000);     %Bore of Engine [m]
l = .1065;           %Length of Engine Connecting Rod [m]
N_cyl = 4;           %Number of Cylinders [unitless]
C_r = 12.7;          %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 1: Engine inputs from engine model code

```
Command Window
  W_dot_ac =
     96.1814
  T_ac =
     78.5013
  PPM_NO =
    472.0590
  Percentage of Fuel Mass Reaching Exhaust
  HC =
      2.8816
  BSNOX =
      2.7821
  BSHC =
     10.8692
fx >>
```

Figure 2: Outputs of engine model

	A	B	C	D	E	F	G	H	I
1	RPM	Power	Torque	BSFC	NO	HC Emissi	Load	BSNOX	BSHC
2	1000	1.135	10.84	819.41	308.76	9.42	0.3	3.954	77.179
3	3000	5.139	16.36	543.12	395.285	6.306	0.3	3.345	34.247
4	5000	10.165	19.41	457.63	430.28	5.252	0.3	3.0764	24.036
5	7000	14.882	20.301	437.61	451.48	4.8166	0.3	3.087	21.0775
6	9000	17.9	18.99	467.76	466.52	4.726	0.3	3.409	22.108

Figure 3: Example of excel table used for data plotting

After all data points have been recorded in the excel spreadsheet, users graph the data using the fuction BSFCcode. This function allows the data to be plotted in a 3-D contour plot with shading. Data is plotted by drawing points from the excel file. The default arrangement has engine speed, in RPM, set as the x variable and engine torque, in N*m, set as the y variable. The z variable is selected as one of the five other data outputs which are BSFC, NO emissions, HC emissions, BSNOx emissions, and BSHC emissions. Labels are changed as needed and the plot is generated. Figure 4 shows an excerpt from the BSFCcode function:

```
% YZ250f Data analysis
% 5/1/2014

clear all
close all
clc

data=xlsread('Theo Data');
x = data(:,1);
y = data(:,3);
z = data(:,6);

plot3(x,y,z,'.-')
tri = delaunay(x,y);
h = trisurf(tri, x, y, z);
shading interp
colormap jet
colorbar
title('Kawasaki ZX10R HC EMISSIONS MAP (%)')
xlabel('RPM')
```

Figure 4: BSFCcode function used for plotting data

3.3 Comparison of Emissions Maps

After plotting the emissions data into emissions maps, comparisons could be made between the volumetric and brake specific emissions. Figures 5 and 6 show an example of a volumetric NO map and a brake specific NO map for a Kawasaki ZX10R engine:

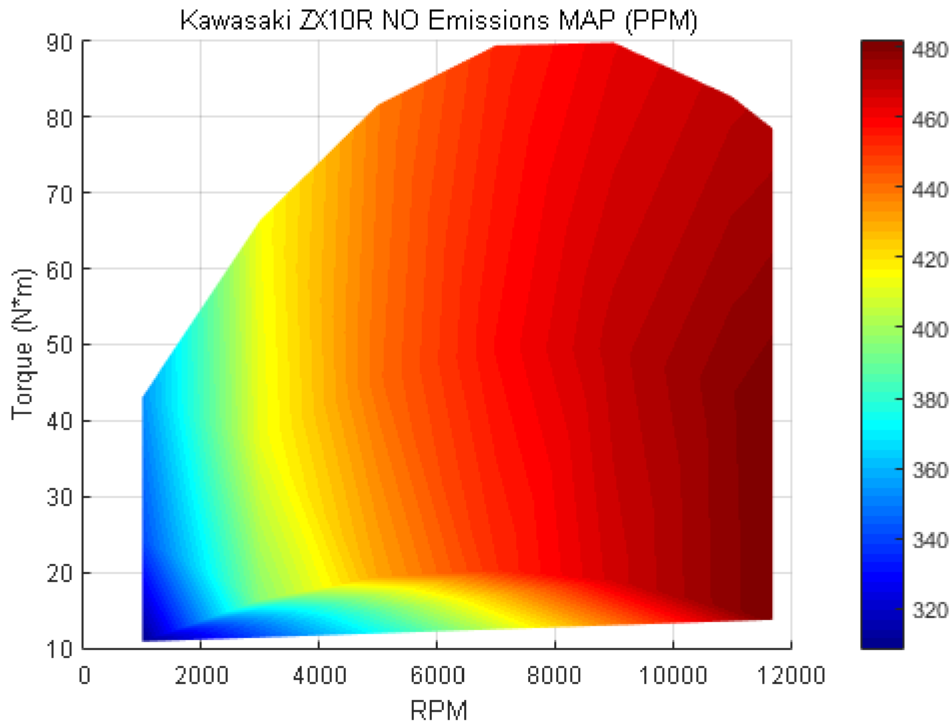


Figure 5: Predicted NO emissions map for 1000 cm^3 motorcycle engine

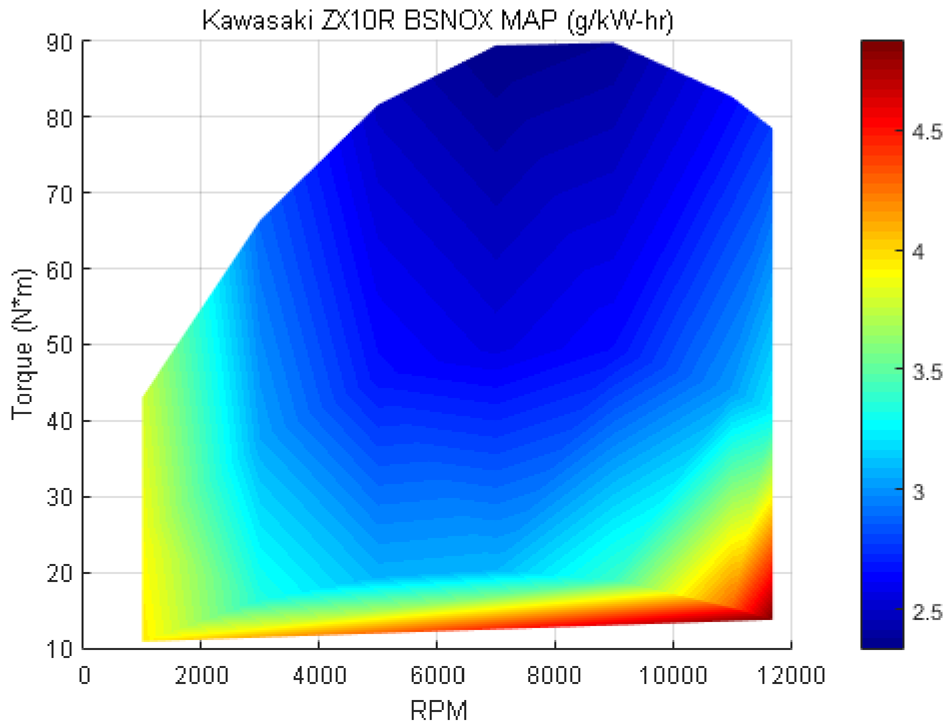


Figure 6: Predicted brake specific NO emissions map for 1000 cm^3 motorcycle engine

The volumetric emissions map shows that NO levels get progressively higher as RPM and throttle position increase, which is expected due to the higher temperatures reached at high engine speeds. The higher temperatures are caused due to there being less time for heat to be lost through cylinder walls. The brake specific map, however, shows that there is a valley in the top third of the map where the amount of NO produced per unit of energy is the lowest. When comparing HC emissions to brake specific HC emissions, the maps are far more similar as can be seen in figures 7 and 8. This is due to the nature of HC emissions in this model, being mostly unburned fuel. However, this is still useful when comparing two different engines because it will still normalize the results.

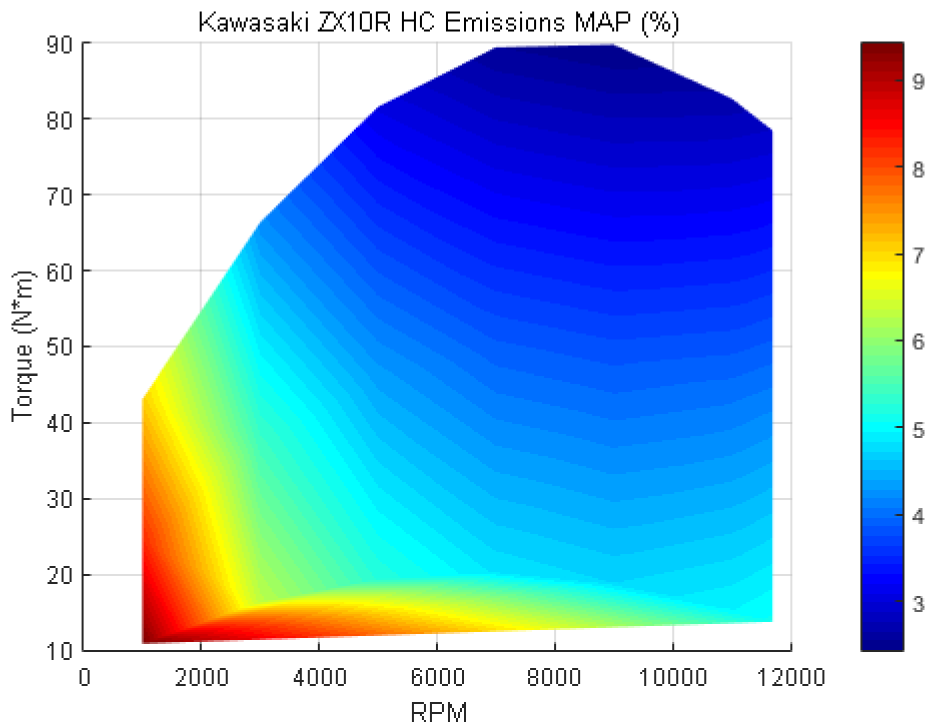


Figure 7: Predicted hydrocarbon emissions of 1000cm^3 motorcycle engine

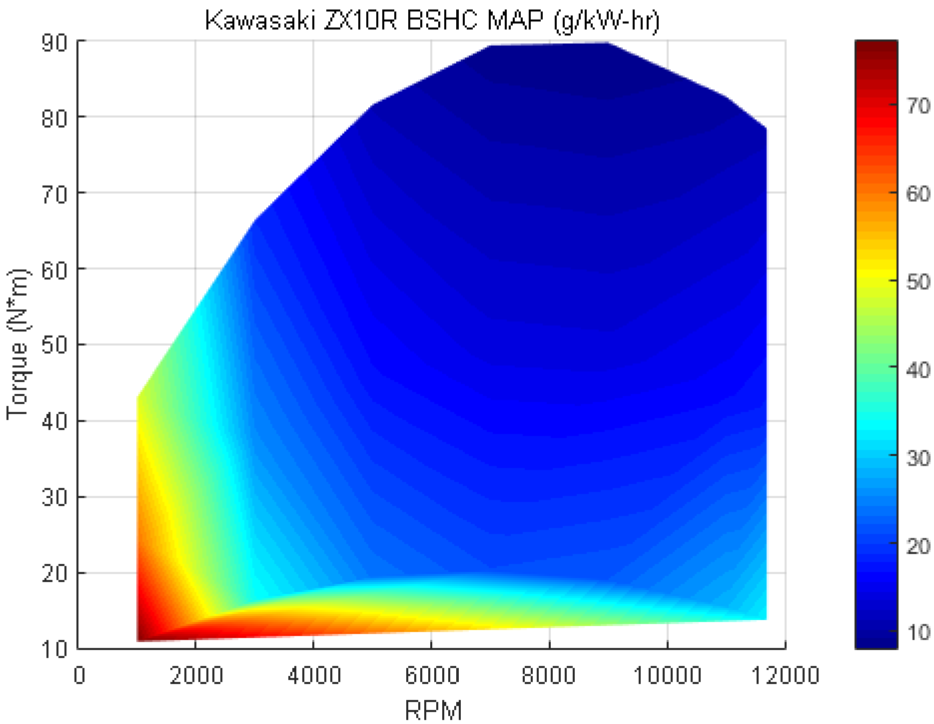


Figure 8: Predicted brake specific hydrocarbon emissions of 1000cm^3 motorcycle engine

Chapter 4: Course Overview

The design of the internal combustion engine technical elective at the University of Idaho uses cooperative learning activities as well as a continuous improvement mindset [15, 16]. This involves regular use of student-generated and instructor-generated assessment reports as well as a quantitative problem solving rubric [17, 18]. The learning outcomes for this course are given in Table 1.

Table 1. ME 433 IC Engines Course Learning Outcomes

1	Become conversant with common engine components and test equipment.
2	Apply the first and second laws of thermodynamics along with balanced chemical reactions to estimate the performance of ideal air cycles.
3	Specify and interpret engine design parameters as well as performance data for a variety of internal combustion engine systems.
4	Predict heats of reaction and concentrations of exhaust species using equilibrium thermodynamics for common and alternative fuels.
5	Predict instantaneous heat release and associated work output as well as pollutant formation for real fuel-air cycles.
6	Complete a literature review, computer modeling project, or an engine lab project on an aspect of combustion engine technology and communicate discoveries in a 5 page technical paper.

The course begins with a one-week segment dedicated to reviewing prerequisite knowledge and skills, culminating with a take-home exam that assesses mastery of the items inventoried in Table 2.

Table 2. ME 433 IC Engines Course Prerequisites

1	Graphing Functions with mathematics software.
2	Documenting engineering calculations so that others can follow your reasoning.
3	Solving Systems of ordinary differential equations that describe dynamics systems.
4	Finding roots of scalar and vector equations that describe engineering phenomena.
5	Performing atom balances for simple chemical reactions.
6	Using first and second laws of thermodynamics to explain engineering systems.
7	Representing and interpreting thermodynamic processes on PV and TS diagrams.
8	Calculating quantities associated with single mode conduction or convection heat transfer

Learning activities that comprise the bulk of the course address the topics listed in Table 3. These learning activities are derived from a process oriented teaching/learning pedagogy that has evolved within the chemical education community over the last ten years [19]. This curriculum design method is known as Process Oriented Guided Inquiry Learning (POGIL) and it is built around thoughtfully constructed concept models that are then investigated in cooperative learning teams using a structured sequence of critical thinking questions [20].

Table 3. ME 433 IC Engines Course Topics

1	Reviewing of chemical, thermodynamic, and fluid fundamentals.
2	Representing processes and cycles on PV and TS diagrams.
3	Identifying and explaining typical engine components.
4	Making distinctions between SI and CI engines from the standpoint of load control, air/fuel mixture control`, ignition control, and pollutant formation.
5	Describing purpose and function of engine control systems, MBT timing, carburetion, fuel injection hardware, and emissions abatement technology.
6	Calculating brake power, brake torque, arbitrary efficiency, and specific fuel consumption.
7	Relating engine design specifications and performance parameters, including different service classifications.
8	Using rules of thumb to select and size engines.
9	Determining road load power and simulating acceleration events.
10	Estimating mean effective pressures (brake, indicated, friction etc).
11	Balancing chemical reaction equations for common combustion scenarios.
12	Classifying hydrocarbon fuels, measuring heating values, computing heats of reaction, estimating flame temperatures, and determining chemical equilibrium.
13	Becoming familiar with issues associated with alternative fuel production and emissions.
14	Simulating engine cycles on the computer, including finite combustion time, cylinder heat transfer, crevice effects, and valve opening/closing.
15	Analyzing performance data from an engine on a test-stand and a vehicle on a chassis dynamometer.

Course learning activities are divided into a multi-week segment on engine design and operating parameters and then a multi-week segment on chemical thermodynamics. Cumulative exams with short calculations and constructed responses are given at the end of each of these segments. The final quarter of the course is dedicated to heat release modeling and customization of a generic MATLAB® model to simulate performance of a specific SI engine selected by each student. This serves as an integrative exercise in which students acquire data from technical literature, select appropriate model inputs required to approximate engine performance, obtain quantitative results, and discuss findings in terms of performance metrics that are independent of engine size (i.e. mean piston speed, brake mean effective pressure, brake specific fuel consumption, and engine emissions).

Eight professional behaviors were targeted in learning activities surrounding this course. These were derived from a profile of an engineering professional that was based on an extensive survey of practitioners inside and outside of academia [21, 22]. Specific learning skills were paired with each of these behaviors and a subset of these skills were targeted in each learning activity. The professional behaviors and supporting life-long learning skills (in parentheses) include:

Locates and consults relevant literature to answer technical questions (inquiring, identifying knowledge gaps).

Questions manufacturer claims and popular beliefs by using knowledge and tools from the course to make informed decisions and recommendations (identifying assumptions, analyzing sources).

Uses engineering concepts and analysis tools when performing diagnostics on engines/vehicles (reasoning with theory, diagramming, validating results).

Documents engineering work for future use in a professional manner (communicating results).

Aware of and concerned about significance of global energy problem and its relationship to next generation vehicle design (preparing for professional practice).

Seeks collaboration, when appropriate, to solve difficult problems (initiating interaction, sharing knowledge).

Values self-directed learning as a value-added source of personal development (being proactive, seeking assessment); and

Periodically reflects on experiences, events, products, and processes to capture and reuse lessons learned (assessing performance, being non-judgmental, generalizing solutions).

Course learning activities follow a common format that includes an orientation that connects the activity with overall course outcomes, specific learning objectives that can be measured at the end of the activity, targeted learning skills that are emphasized in facilitating the activity, supporting models (data tables, code segments, and diagrams), and a series of critical thinking questions as well as short calculations. Class begins with a 15-20 minute mini-lecture followed by large-class orientation and instructions. For the next 30-45 minutes students address cooperative learning tasks in teams of 2-4. The final 10-15 minutes of the class is devoted to reporting major findings and addressing unanswered questions. Typically the reporting generates insights and lively discussion relevant to upcoming homework assignments and mid-term exam segments. All course lecture materials and learning activities is publicly available and can be accessed via the course website [23].

Engine and vehicle performance modeling are major course themes. The initial thermodynamics review stresses visualization of different processes on P-v and T-s diagrams as well as estimating cycle performance with properties calculated from different equations of state. This is followed by introduction of basic equations for engine power, engine torque, and engine fuel consumption that include typical parameters that scale across different sized engines.

$$\dot{W} = n_{cyl} * \left(\pi * B^2 * \frac{L}{4}\right) * \frac{bmep}{nr} * N \quad (45)$$

$$\mathrm{S_p} = 2 * \mathrm{L} * \mathrm{N} \quad (46)$$

$$\mathrm{T} = \frac{\dot{\mathrm{W}}}{2*\pi*\mathrm{N}} \quad (47)$$

$$\mathrm{bsfc} = \eta_a * \dot{m}_f * \frac{\mathrm{HV}}{\dot{\mathrm{W}}} \quad (48)$$

where,

$\dot{W}$ = brake power

n_{cyl} = number of cylinders

B = bore

L = stroke

Bmep = brake mean effective pressure

N = engine speed

n_r = number of revolutions per power stroke

S_p = mean piston speed

T = brake torque

bsfc = brake specific fuel consumption

η_a = arbitrary engine efficiency

$\dot{m}_f$= fuel flow rate

HV = fuel heating value

One of the course resources is a sheet with typical parameters for different engine types, but which span fairly broad ranges of engine sizes. These parameters include bore/stroke ratio, mean piston speed, brake mean effective pressure, volumetric efficiency, mechanical efficiency, and brake specific fuel consumption. In early homework assignments, students compare these typical parameters with ones derived from data reported in engine manufacturer literature. Students are instructed to pick engines of personal interest, search for, and then evaluate relevant data sources. Students then solve the inverse problem of sizing a particular type of engine that is required to meet specific performance requirements.

Practice with differential equation solvers available in MATLAB® is acquired through production and analysis of vehicle velocity and position trajectories on a dragstrip. A longer, more complicated equation is then introduced for engine brake power output that includes finite combustion, effects of heat transfer, effects of fluid mechanics, and effects of frictional losses. This equation is given below. Terms in the first parentheses relate to engine thermochemistry, the terms in the second parentheses relate to engine fluid mechanics, and the terms in the third parentheses related to engine kinematics. Each of these phenomena is studied in detail as the course unfolds. Attention is also given to equilibrium chemistry and rate-controlled reactions associated with pollutant formation.

$$\dot{W} = \left(\eta_m * \eta_t * \eta_c * HV * \frac{F}{A}\right) * \left(\eta_v * \frac{P_i}{R_a * T_i}\right) * \left(\pi * B^2 * L * n_{cyl} * \frac{N}{4 * n_r}\right) \quad (49)$$

where the previously undefined parameters are:

η_m = mechanical efficiency

η_t = thermal efficiency

η_c = combustion efficiency

$\frac{F}{A}$ = fuel to air ratio on a mass basis

η_v = volumetric efficiency

P_i = intake pressure

R_a = gas constant for air

T_i = intake temperature

Chapter 5. Engine Model Implementation

The following order of calculation was done with MATLAB® Release 2014b:

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 Emissions predictions functions.
8 Plot Statements.

The script was initiated with known inputs, such as the bore, stroke, connecting rod length, number of cylinders, compression ratio, and fuel inputs. These inputs were followed by the operating characteristics of the engine, such as the combustion burn duration, spark-advance, and intake and exhaust valve timings. These inputs and operating characteristics served as the basis for the MATLAB® model and allowed all engine geometry parameters, such as the cross-sectional area of the piston, the displaced cylinder volume, clearance volume, and instantaneous heat transfer surface area to be calculated. Engine friction losses were modeled linearly as a function of engine speed and stroke, and volumetric efficiency losses were modeled as a correction factor to inlet pressure, based on experimental data from similar engines.

It was found that pre-allocating vectors drastically improved the efficiency of the MATLAB® script. This prevented the program from re-sizing vectors between iterations, thereby reducing computation time. Pre-allocated arrays and matrices were also used as a means of setting initial values while simulating exhaust gas recirculation (EGR). EGR required the script to run two times with only the gas temperature and fluid properties changing during the second iteration. It was found that updating fluid properties between iterations allowed for more accurate fluid and gas property predictions, thereby increasing the accuracy of the model.

The opening and closing of intake and exhaust valves was assumed to be instantaneous; that is, the gas dynamics and valve lift profile weren't considered. The timing of intake and exhaust valves was adjusted in crank angle degrees relative to top-dead center. Upon intake valve closure, the model assumed an instantaneous spike in cylinder pressure, and upon the opening of the exhaust valve, the model assumed an instantaneous release of combustion gases. Over the crank-angle degrees between intake valve closure and the exhaust valve opening, two-zone calculations such as the burned and unburned masses, volumes, temperatures, and areas were calculated using instantaneous thermal properties. The two-zone calculations used outputs from the single-zone calculations such as the bulk gas pressure and the mass fraction burned.

It was determined that storage of the NO model within the script was unnecessary, so a MATLAB ® function was developed for this purpose. The bulk of this function was reserved for fuel-rich combustion and calculating the associated equilibrium constants for CO2 and CO using the water-gas shift reaction, while the rest of the function focused on predicting the quantitative fraction of NO particles and determining the residence time for NO formation. Methods described by Heywood [1] were used to predict the NO residence time and the equilibrium constants for the water-gas shift reaction. Lastly, plots and diagrams were developed using outputs from the model. Respective plots were generated to validate the

Weibe function, cylinder pressure and temperature, engine power, and heat release. Figure 9 shows the model inputs and outputs.

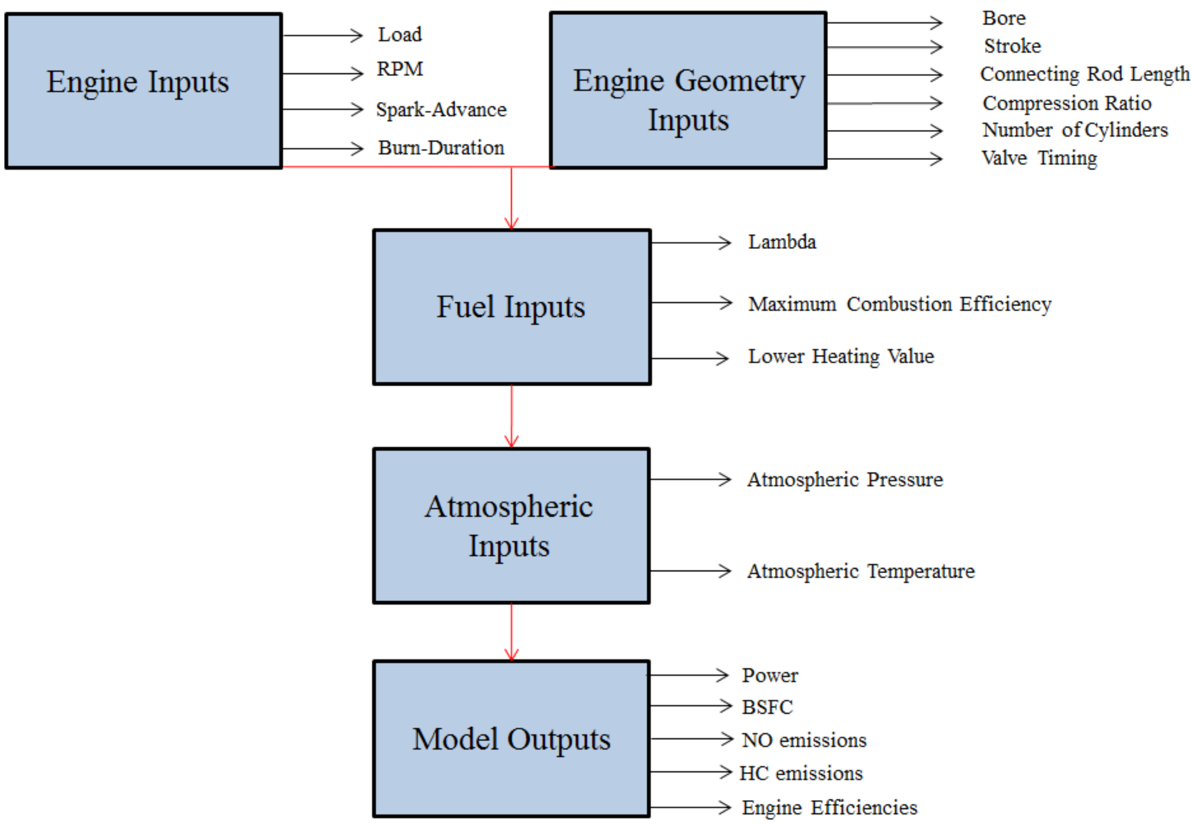


Figure 9: Flow-chart of the MATLAB® model's inputs and outputs

Chapter 6. Engine Modeling Mini-Project

6.1 Mini-Project Description

The final project of ME 433 is designed to bring together all of the elements taught across the course into a usable model. This includes working knowledge of mechanisms as well as assumptions associated with the Wiebe function, bulk temperature heat release, air/fuel ratio maps, volumetric efficiency variations, friction losses, and wall heat losses. It also exposes students to a two-zone burn model used to predict the formation of NOx, the use of the water-gas shift reaction to predict carbon monoxide levels in rich running conditions, and the use of oil adsorption and crevice quenching to estimate the amount of unburned hydrocarbons in the exhaust.

Students select a 4-stroke, spark ignition, naturally aspirated engine that has factory or otherwise available power and torque data. Students are given a MATLAB® code and attend an instructional activity in the computer lab where they learn about the operation and modification of the code. This includes questions about the governing equations used in different parts of the code. The data for YZ250 engine used in the verification study is used in this learning activity. Most students are able to understand and effectively operate the MATLAB® model after this 60 minute hands-on session.

As part of their final project, students are expected to complete a variety of set-up tasks and data collection/analysis tasks. The four set-up tasks and their weighting in grading within the final project are:

Engine Specifications. Locate data on your engine bore, stroke, connecting rod length, number of cylinders, compression ratio, and valve timing (10 points)

Lambda Tuning. Based on the chosen engine, construct a custom air-fuel ratio map over your engine's operating range. You might be able to find an aftermarket map, or you might have to make assumptions based on the type of engine. Explain your assumptions. (10 points)

Volumetric Efficiency Tuning. Based on the chosen engine, construct a customized volumetric efficiency profile and update this within the volumetric efficiency function. Explain your assumptions. (10 points)

MBT Timing. Determine optimal spark-timing for a matrix of 16 to 25 points that covers the operating range of your engine. Plot and discuss bmep versus optimal spark-timing for a sample mid-range operating point (10 points).

Next, students create a matrix of operating points for their engines. At each operating point RPM, Torque, BSFC, NO emissions, and HC emissions were recorded in an Excel file that was in turn read by a standard plot generator. Students were then expected to successfully complete the following data generation/analysis tasks. The weighting of each task in grading the project is also given.

Performance Curves. Plot and discuss power as well as torque curves. If available, compare these with power and torque curves reported for your engine. (10 points)

Fuel Economy. Plot and discuss the BSFC map for your engine. (10 points)

Engine Emissions. Plot and discuss NO and HC emissions maps for your engine. (10 points)

Assessment of Learning. Write a one page essay about your three most valuable lessons learned as well as your two most burning questions about engine heat release modeling as a result of this assignment. (15 points)

Assessment of Work Products. Write a one-page essay about the two greatest strengths and two greatest improvements exhibited in this project. State why the strengths are valuable in professional practice and how they were produced. Identify the added-value of each improvement and outline an action plan for adopting these. (15 points)

6.2 Mini-Project Predictions

Two different case studies are presented here, with displacements of 50 cm^3 and 1 liter. The input parameters for each engine are listed in table 4.

Table 4: List of inputs used for mini-project engines

Variable	50cc	1000cc
Load	.3-1	.3-1
RPM	4200-7600	1000-11,700
Spark-Advance (deg)	26	25
Burn Duration (deg)	55	85
Bore (mm)	36	76
Stroke (mm)	41.8	55
Connecting Rod Length (mm)	48.8	106.75
Compression Ratio	8	12.7
Number of Cylinders	1	4
Intake Valve Closes (deg)	0	0
Exhaust Valve Opens (deg)	314	314
Lambda	1.01	.9
Combustion Efficiency	.99	.95
Lower Heating Value (J/kg)	44.6E+6	44.6E+6
Atmospheric Pressure (Pa)	101325	101325
Atmospheric Temperature (K)	290	290

The smallest example from the class is the modeling of a 50 cm^3 4-stroke engine from a small scooter. The performance was mapped over a range of engine speed from 4200 to 7600 rpm. Spark timing was set to 26 degrees before top dead center. Volumetric efficiency was modeled as a quadratic equation that changed as a function of RPM. Lower values at low speed were assumed to result from charge heating and backflow. Maximum values at intermediate speed were assumed to correspond to tuned operation. Decreasing values at high speed were assumed to result from flow friction and choking. After modification, torque and power vs. RPM were plotted in graphs and then compared to actual values for the engine shown in figures 10 and 11:

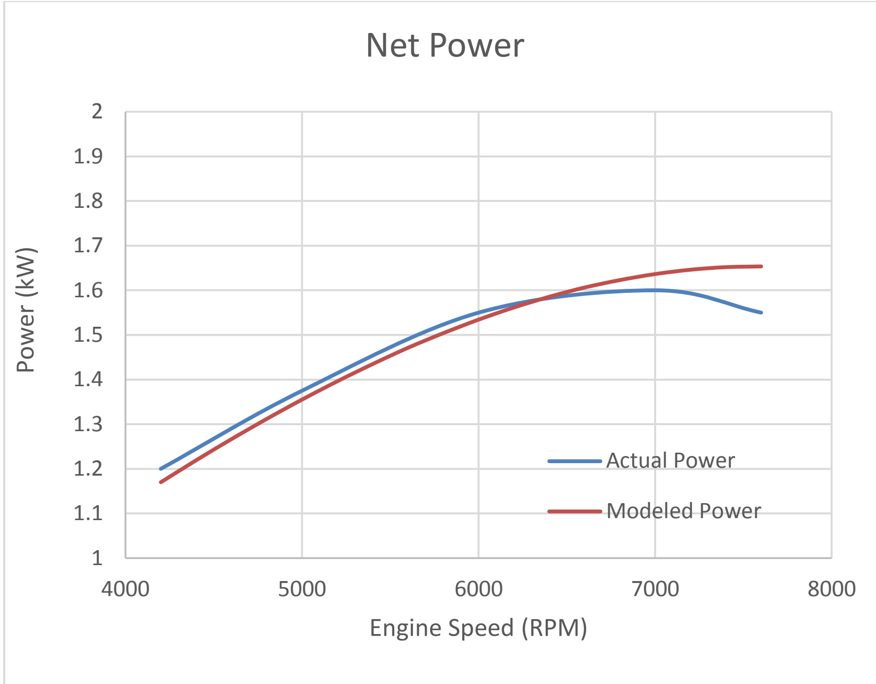


Figure 10: Predicted and actual power curves of 50 cm^3 scooter engine

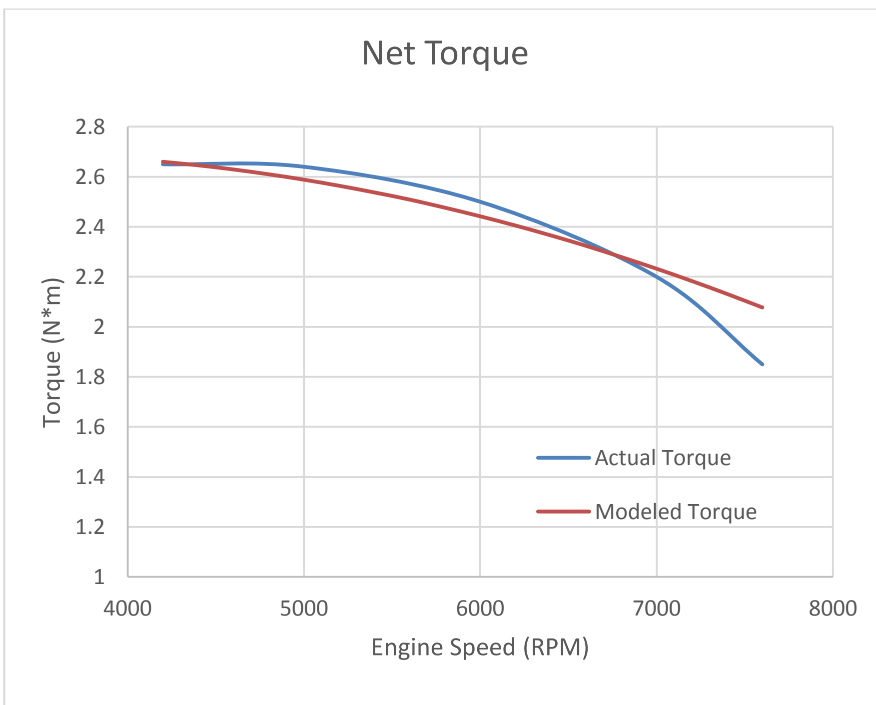


Figure 11: Predicted and actual torque curves of 50 cm^3 scooter engine

The predicted values from the model are within 3% of the actual values, with the largest discrepancy being approximately 6% at the high end of the RPM range. Further refinement of the volumetric efficiency would increase accuracy over the range, but for a first pass model the results would be accurate enough to draw conclusions about engine performance. BSFC, NOx and HC were also predicted, all of which were higher than expected. The HC emissions were high due to the high surface area to volume ratio of the engine and NOx emissions were high due to the large amount of oxygen left over after combustion. BSFC was higher due to the large amount of unburned fuel leaving the engine.

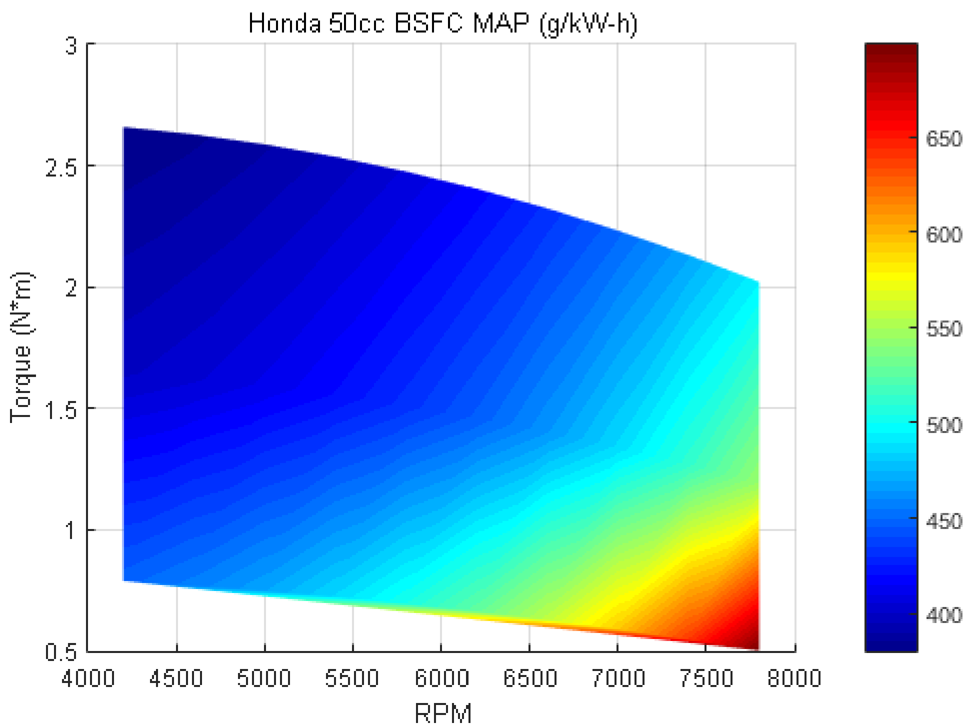


Figure 12: Predicted brake specific fuel consupmtion map of 50 cm^3 scooter engine

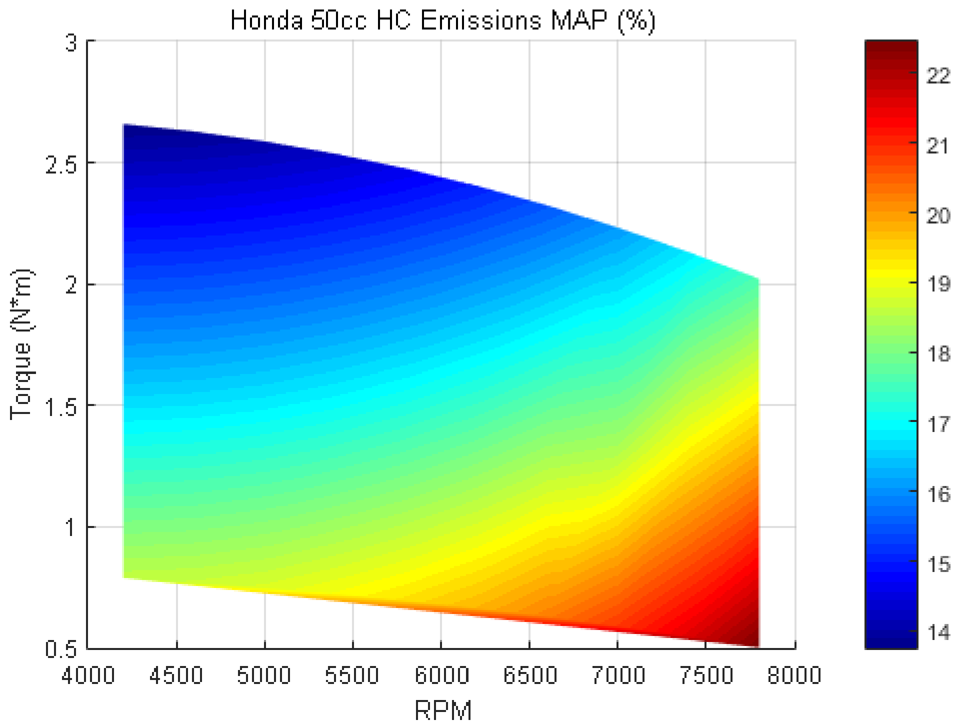


Figure 13: Predicted HC emissions for 50 cm^3 engine

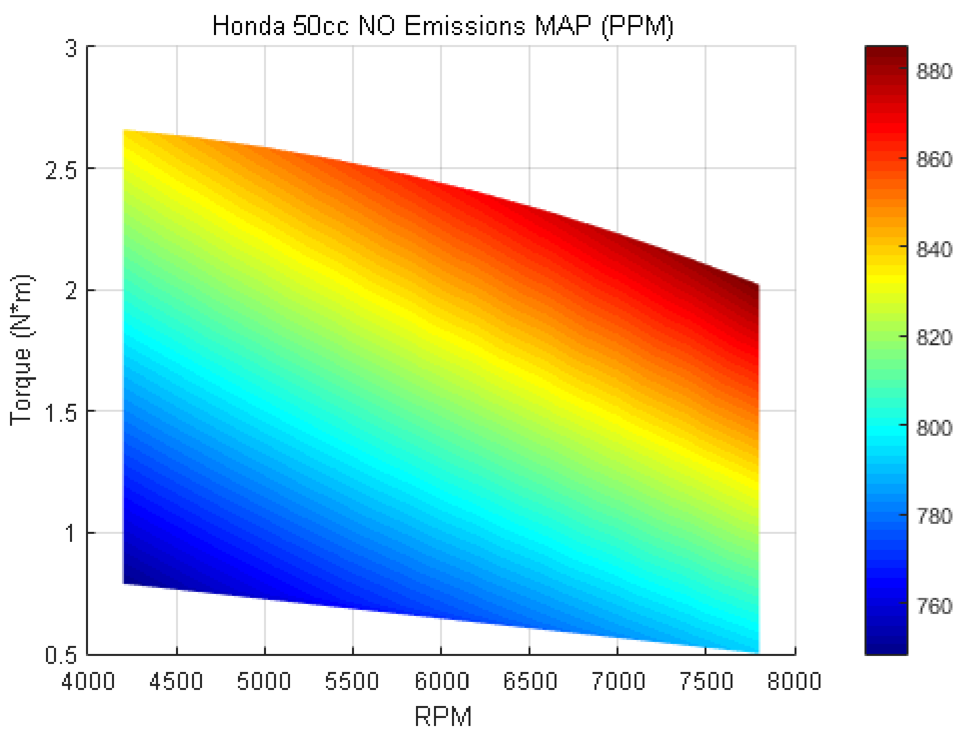


Figure 14: Predicted NO emissions for 50 cm^3 engine

The second example is the model of a 1000 cm^3 4-cylinder engine from a performance motorcycle. Performance was mapped over a range from 1000 to 11,700 rpm. Spark timing was optimized to 25 degrees before top dead center over the ideal range for maximum torque between 6000-9600 rpm. Volumetric efficiency was again modeled as a quadratic equation changing as a function of RPM. Torque and power were graphed over the range:

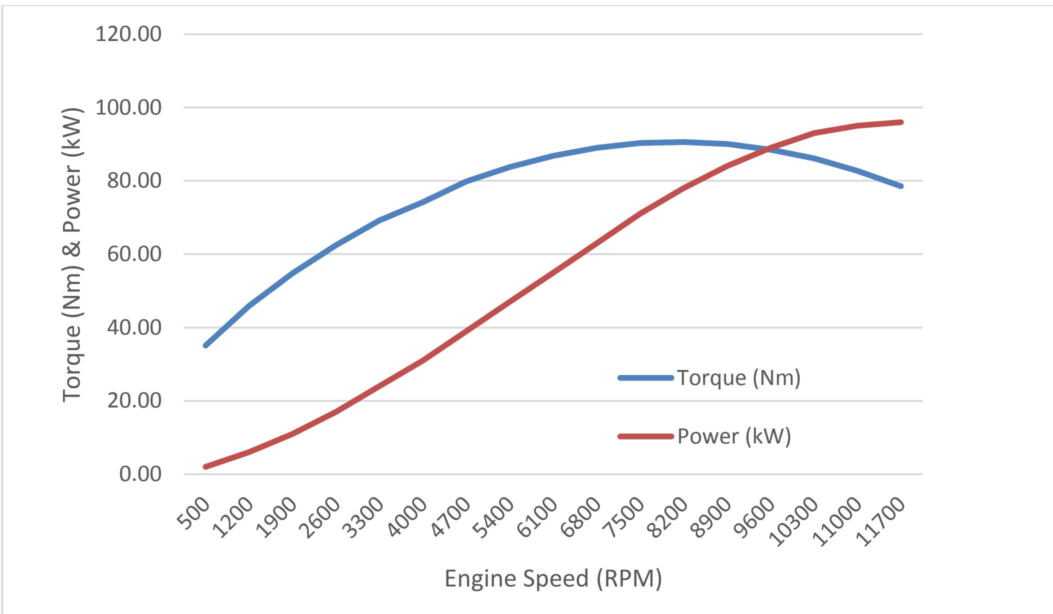


Figure 15: Predicted torque and power of 1000 cm^3 mtorcycle engine.

As shown in figure 15, the maximum torque value of 90.6 N-M occurs at 8600 rpm for this tune, and is similar to the factory specified maximum torque value of 99.5 N-M for this engine. However, the factory maximum torque point occurs at 11,000 RPM. This indicates that the volumetric efficiency curve and spark timings may need further optimization to match factory stock data. The power predictions indicate a similar conclusion, with the maximum predicted power of 96 kW at 11,700 rpm falling short of the factory specified value of 119.3 kW at the same engine speed. Fuel and emissions data were also predicted and graphed for this engine, as shown in figures 11, 12, and 13.

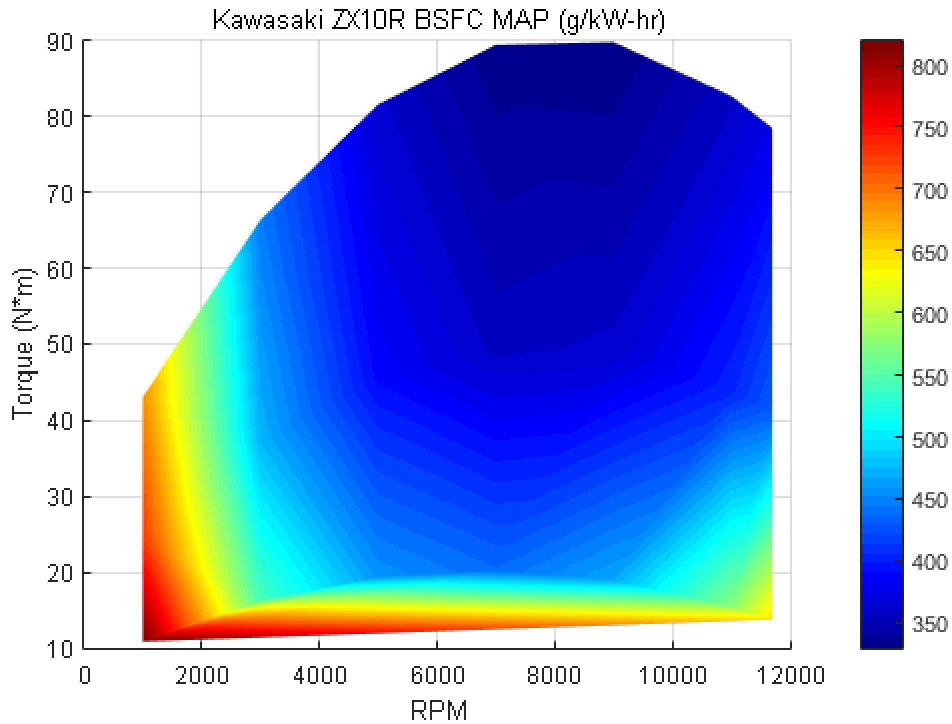


Figure 16: BSFC map for 1000 cm^3 engine

The BSFC map shows maximum efficiency in the upper third of the RPM range at open throttle conditions. This is to be expected from a high-performance motorcycle engine.

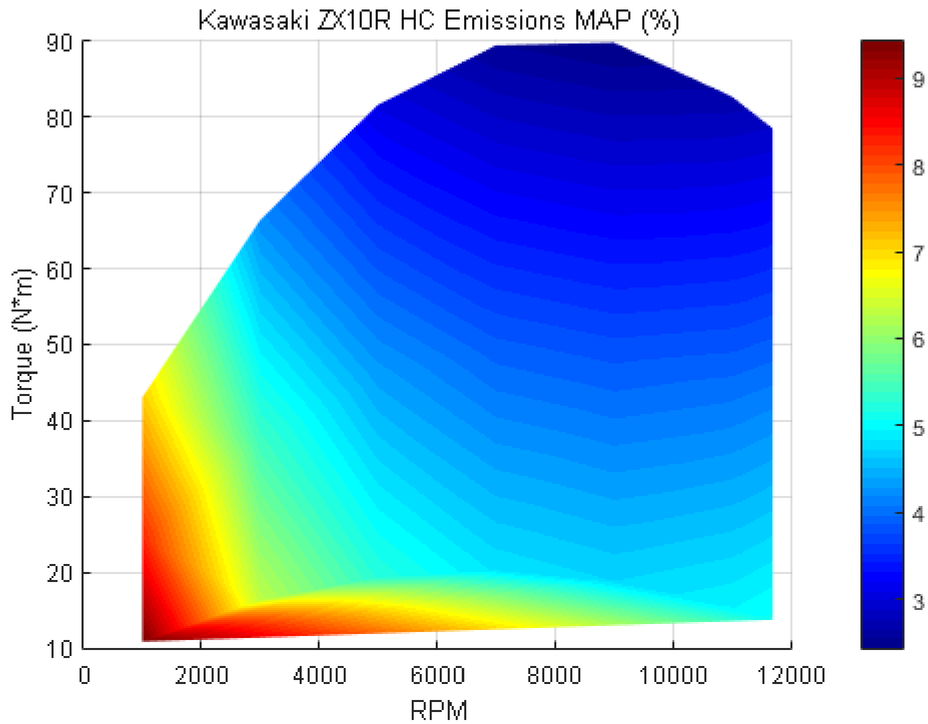


Figure 17:HC emissions map for 1000 cm^3 engine

The predicted HC emissions map shows that the engine will burn most of the fuel input at high load and RPM conditions, but at lower RPM and load the emissions increase.

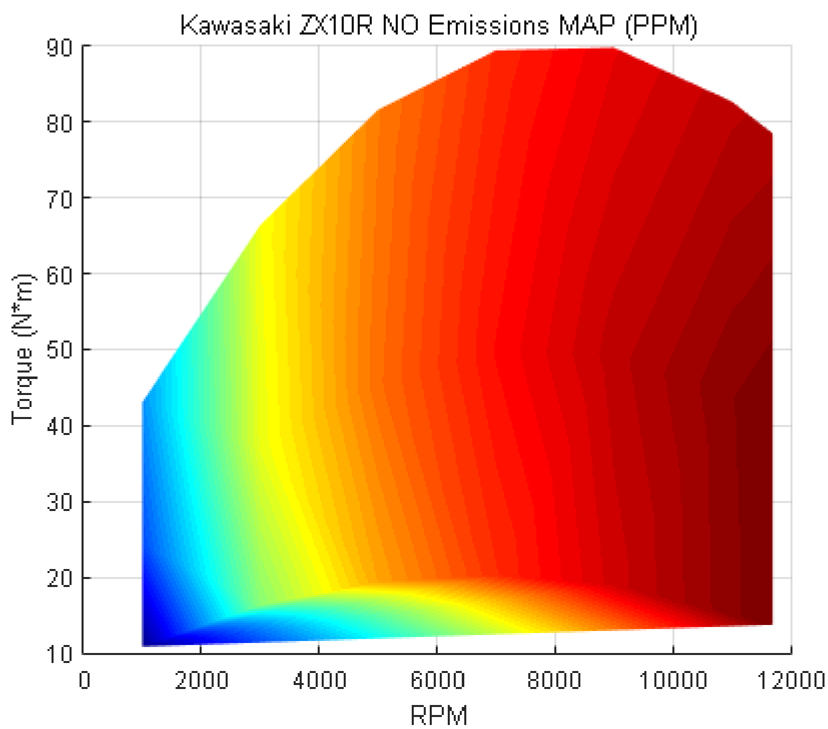


Figure 18: NO emissions map for 1000 cm^3 engine

The NO emissions map shows that NO is predicted to increase at higher RPM and higher load. This is expected, as temperature increases at higher loads and RPM due to decreased time for heat to be lost through cylinder walls.

6.3 Online Resources for Mini-Project

There are several resources available to students for use during the course of the mini-project. They are available on the University of Idaho Mindworks site under the IC engines class. The resources used in the mini-project include copies of the engine code, which is broken up into the main code, labeled YZ250 Engine Model, and three functions labeled Hydrocarbon Function, NO Function, and Volumetric Efficiency Fcn. These four matlab files are necessary to begin the final project. There is also a MATLAB script for the plotting of brake specific fuel consumption and engine emissions as soon as data has been collected, with an example file for use as a resource. Additional resources are useful in operation of the code. These include the final project description, which contains a step-by-step instruction set to begin using the MATLAB engine model, Jeremy Cuddihy's thesis, which has detailed instructions as well as technical information on the workings of the code, and a power point containing tips on optimizing the spark timing of the engine simulation. This is all shown in figure 19:

Wed, 7/29 - Session 25 Exam #3 Recap Engine Modeling Project	Jeremy's Thesis Presentation (first part of lecture) Engine Model Governing Equations (second part of lecture) Jeremy's Thesis MatLAB Engine Code: YZ250 Engine Model (MatLAB) Hydrocarbon Function (MatLAB) NO Function (MatLAB) Volumetric Efficiency Fcn (MatLab) MatLAB Plots: BSFC Plotting Data File Engine Information: YZ250 Specifications YZ250 Dyno Curves R6 Air-Fuel Ratio Map	Final Project Spark Timing Tips

Figure 19: Resources available for use on the engine modeling mini-project.

Chapter 7. Assessment of Mini-Project and Course Learning

7.1 Mini-Project Assessment

As shown in table 5, the class average was very good across almost all grading criteria for the final project. The areas that showed significant difficulty were lambda tuning, volumetric efficiency tuning, and engine emissions. Lambda tuning included spark timing optimization, and many students had difficulty with this. Many optimized the spark timing and lambda for a single operating point and used this over the entire range of their engine instead of finding the optimal conditions at each point. This, in turn, caused problems in performance curves. Volumetric efficiency tuning was an area of difficulty because students needed to construct their own volumetric efficiency curve. Many opted to use the curve built into the program, which was optimized for a single cylinder motorcycle engine instead of creating their own. This was okay for some students, as they used motorcycle engines for their final project choice, but others who chose automobile engines found that their results were less accurate as a result. The last point of difficulty was in producing NO and HC emissions maps. The largest issue with the construction of emissions maps was a lack of data points, with some students producing only a wide-open throttle curve to compare against factory torque and power curves.

Table 5: Assessment of student performance on mini project

PROJECT COMPONENT	Class Average
Engine Specifications	**97**
Lambda Tuning	**77**
Volumetric Efficiency Tuning	**81**
MBT Timing	**83**
Performance Curves	**80**
Fuel Economy	**83**
Engine Emissions	**79**
Assessment of Learning	**83**
Assessment of Work Product	**83**

Overall, the modeling project was very well-received by students. In post project assessments, students indicated that the model is an excellent way to show practical applications for the topics covered in the course. They were impressed with the high accuracy and ease of use of the model and the opportunity to choose their own engine to model increased student commitment to the project.

7.2 Course Assessment

Prior to the final project students were asked to rate a number of course components as to their usefulness in helping them learn the content of the course. The results (average and standard deviation) from this survey are shown in Table 6. A score of 1 was designated 'marginally beneficial'; a score of 3 was 'moderately beneficial'; and a score of 5 was 'highly beneficial. The results show students relying on a diverse set of mutually supportive, active learning methods.

Table 6: Student assessment of learning methods within the course

Course Activity	Class Average	Standard Deviation
Taking notes during lecture	3.9	1.3
Asking questions in class	3.7	0.9
Studying handouts and web resources	3.8	1.0
Reading the text	3.0	1.2
Classroom learning activities (with partner)	4.3	1.0
Working homework problems	4.6	0.9
Discussing homework problems in class	4.3	1.0
Interacting with the instructor	4.1	1.0
Experimenting with computer models	3.9	1.0
Working with a partner (outside of class)	4.0	1.1
Taking exams	3.5	1.3

Students were also asked to comment on how their professional behaviors had changed over the course. Tables 7 and 8 show perceived changes in professional behavior resulting from this course. Table 7 contains data prior to the introduction of the MATLAB ® based final project when the project was conducted with compiled FORTRAN code along with batch files. Table 8 contains data after the second implementation of the MATLAB ® based final project. Both course offerings included two students who were enrolled through the Engineering Outreach program, participating at a distance using pre-encoded sessions that contained break points where they could work through course activities. Ratings submitted by Engineering Outreach students were consistent with those shared by in-class students. The results of this survey imply somewhat higher levels of perceived transformation in professional behaviors targeted in the 2015 course. These changes cannot be entirely attributed to the final project, as they also reflect refinements to course learning activities. However, student comments associated with the survey lend support to the role of MATLAB® modeling exercises in the 2015 survey.

Table 7: Perceived improvement in professional behavior over duration of course (2010 – 17 students)

Professional Behavior (targeted for this course)	**Significantly Improved**	**Slightly Improved**	**No Change**
Locates and consults relevant literature to answer technical questions	12%	53%	35%
Questions manufacturer claims and popular beliefs by using knowledge and tools from the course	18%	64%	18%
Uses engineering concepts and modeling tools to analyze engine performance	59%	24%	18%
Documents work for future use by self and others	0%	53%	47%
Aware of how next generation vehicle technology can impact the global energy environment	12%	47%	41%
Seeks collaboration, when appropriate, solve difficult problems	12%	29%	59%
Values self-direct learning as a source of personal and professional development	12%	53%	35%
Periodically reflects on experiences, events, work products, and solution processes to extract lessons learned	0%	65%	35%

Table 8: Perceived improvement in professional behavior over duration of course (2015 – 15 students)

Professional Behavior (targeted for this course)	Significantly Improved	Slightly Improved	No Change
Locates and consults relevant literature to answer technical questions	33%	60%	7%
Questions manufacturer claims and popular beliefs by using knowledge and tools from the course	53%	33%	14%
Uses engineering concepts and modeling tools to analyze engine performance	66%	27%	7%
Documents work for future use by self and others	47%	33%	20%
Aware of how next generation vehicle technology can impact the global energy environment	27%	53%	20%
Seeks collaboration, when appropriate, solve difficult problems	27%	60%	13%
Values self-direct learning as a source of personal and professional development	33%	60%	7%
Periodically reflects on experiences, events, work products, and solution processes to extract lessons learned	47%	53%	0%

7.3 Student Feedback

Students disclosed that doing calculations on engines of personal interest was a great motivation to get into the technical literature and acquire engine design as well as operating data. In particular, a number of students identified www.sae.org as a great starting point for this exploration. The sheet of typical parameters for different engine types was identified as a touchstone for evaluating the results of personal engine calculations as well as claims in the media. Connecting thermodynamic diagrams with governing equations used throughout the course and conducting parametric studies across a range of different input variables was also cited as a great tool for learning. Exemplary homework submissions were circulated for students to study and emulate after each submission. This was felt to be very motivational in increasing the quality of personal documentation in subsequent homework submissions. The cooperative learning component of the class was appreciated and this was familiar from many other courses in our curriculum. The instructor emphasis on validating and justifying results was also felt to be a good prompt for doing reflective writing as part of most calculations and simulation exercises. The integration of engineering tools such as EES and MATLAB® was viewed as an avenue for doing authentic design work aligned with workplace practices.

Chapter 8. Conclusions

8.1 Conclusions

The course design reported here was found to be effective in teaching internal combustion engine fundamentals as well as engine performance modeling. The interactive nature of the class and including in-class activities allowed students to get help from each other and instructors on abstract concepts that are often difficult to learn from a Powerpoint-driven lecture accompanied by homework problems that focus only on single engine processes. The small, interactive nature also allowed the entire course to take place in an 8 week summer term. The culminating engine modeling project required students to synthesize their understanding of internal combustion engine systems as well as reflect on their professional growth. The additional brake specific emissions modeling capabilities added through the course of this research will allow future students another parameter to optimize their engines for, as well as allowing engines across a range of displacements to be directly compared to each other.

8.2 Future Work

The engine modeling project has given us insight into ways we can improve our MATLAB® model as well increase undergraduate student engagement in engine simulation. One of the largest and most sought after model refinements would be expanding the simulation to forced induction engine performance. This will add value because the automotive industry is currently beginning to transition from large-displacement naturally-aspirated engines to smaller displacement turbocharged engines to increase part-load efficiency without sacrificing maximum power and torque. Another improvement includes modifying the code to allow for late intake valve closing so the model can be applied to engines running the Miller or Atkinson cycles. The automotive industry is beginning to favor higher efficiency cycles over power density, the Atkinson cycle is being used more often, particularly in the application of hybrid gasoline-electric vehicles.

References

[1] J. Heywood, *Internal Combustion Engine Fundamentals*. Tata Mcgraw Hill Education, 2011.

[2] C. Ferguson, and A. Kirkpatrick, *Internal Combustion Engines: Applied Thermosciences.* Third Edition. John Wiley & Sons, 2016.

[3] Cuddihy, Jeremy. “A User-Friendly, Two-Zone Heat Release Model for Predicting Spark-Ignition Engine Performance and Emission”, MS Thesis, University of Idaho, 2014.

[4] Cuddihy, J., Beyerlein, S., White, T., and Cordon, D., "MATLAB® Modeling of an IC Engine as a Capstone Learning Experience in a Combustion Engines Course," SAE Technical Paper 2016-01-0173, 2016, doi:10.4271/2016-01-0173.

[5] M. Klein, “A specific heat ratio model and compression ratio estimation,” Univ., Linköping, 2004.

[6] 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, 1999.

[7] 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.

[8] R. Stone, *Introduction to internal combustion engines.* Warrendale, PA: Society of Automotive Engineers, 1999.

[9] G. P. Blair, *Design and Simulation of Four Stroke Engines [R-186]*. Society of Automotive Engineers Inc, 1999.

[10] 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, 1999.

[11] S. Turns, *An Introduction to Combustion: Concepts and Applications*, 3 edition. McGraw-Hill Science/Engineering/Math, 2011.

[12] 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, 1995.

[13] 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, 2000.

[14] 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.

[15] Hanson, D. and Heller, D. Learning Communities as a Strategy for Success, *International Journal for Process Education*, 1(1), 19-24, 2009.

[16] Baehr, M. and Beyerlein, S. "Overview of Assessment", in *Faculty Guidebook: A Comprehensive Tool for Improving Faculty Performance*, Pacific Crest, 437-440, 2007.

[17] Wasserman, J., and Beyerlein, S. "SII Method of Assessment Reporting", in *Faculty Guidebook: A Comprehensive Tool for Improving Faculty Performance*, Pacific Crest, 455-456, 2007.

[18] McCormack, J., Williams, B., Noren, K., Beyerlein, S., Cordon, D., and Morgan, J., Development and Usability of a Generic Quantitative Problem-Solving Rubric for Student Learning, *International Journal for Process Education*, 4(1), 29-41, 2012.

[19] www.pogil.org, Process Oriented Guided Inquiry Learning, accessed 12/31/15.

[20] Hanson, D. *Instructor's Guide for Process Oriented Guided Inquiry Learning,* Pacific Crest, 2006.

[21] Davis, D., Beyerlein, S., and Davis, I., "Development and Use of an Engineer Profile", Proceedings of the American Society for Engineering Education Conference, 2007.

[22] Apple, D., Beyerlein, S., Leise, C., and Baehr, M., "Classification of Learning Skills", in *Faculty Guidebook: A Comprehensive Tool for Improving Faculty Performance*, Pacific Crest, 43-46, 2007.

[23] ME 433 Internal Combustion Engine Systems www.webpages.uidaho.edu/mindworks/ic_engines.htm, accessed 12/27/2015.

Appendices

Appendix A: 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 = .3;          %Engine Load (Affects Inlet Pressure)
RPM = 7800;           %Revolutions Per Minute [1/min]
L = (36/1000);       %Stroke of Engine [m]
B = (41.8/1000);       %Bore of Engine [m]
l = (48.8/1000);             %Length of Engine Connecting Rod [m]
N_cyl = 1;             %Number of Cylinders [unitless]
C_r = 8;            %Compression Ratio [unitless]
N_r = 2;               %Number of Revolutions Per Power Stroke
theta_b = 55;          %Combustion Burn Duration [degrees]
theta_0 = 108;         %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 = .99;              %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 = -.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(l^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 Suface 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=.90*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

%Calculate brake specific emissions

[ w_NO ] = BSNOX( T_NO,P_atm,lambda,P_NO,T_BDC,P_BDC,P_EXH);
m_dot_NO = w_NO*(m_dot_a+m_dot_f);
BSNOX = (m_dot_NO*1000*3600)/(W_dot_ac) % [g/kW-h]

m_dot_HC=HC/100*m_dot_f;
BSHC=(m_dot_HC*1000*3600)/(W_dot_ac) % [g/kW-h]

%___________________________________________________________________________

%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('Bulk-Gas 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 B: BSNOx Function

```
function [ w_NO ] = BSNOX( 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-2.3*10^3)/(7.6*10^2);
    K_p_CO2 = -.55*z^3+1.5*z^2-3*z+9.1;
    Z=(T_NO-2.3*10^3)/(7.6*10^2);
    K_p_CO = -.15*Z^3+.41*Z^2-.92*Z+7.1;
    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));
    %x_CO=x_CO+(ALPHA/n_prod);
end

%Calculate Equilibrium Concentrations
X_O2_e=x_O2*P_BDC/(R_u*T_NO);
%___________________________________________________________________________

%Equilibrium Constant For O2 to Oxygen Reaction
Kp_7=3.6*10^3*exp(-31090/T_NO)*318.3; %318.3 converts atm^(1/2) to pa^(1/2)
x_O_e= (Kp_7*X_O2_e^(1/2))/((R_u*T_NO)^(1/2))/(P_BDC/(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_BDC/(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;

%___________________________________________________________________________

%Converting to mass fractions of constituents:

m_CO2=x_CO2*(44.01);    m_H2O=x_H2O*(18.015);    m_CO=x_CO*(28.01);
m_H2=x_H2*(2.01);  m_O2=x_O2*32;    m_N2_e=x_N2_e*28;

%Adding mass fractions of all constituents:

m_all=m_CO2+m_H2O+m_CO+m_H2+m_O2+m_N2_e;

%Calculating mass fraction of NO:

w_NO=dNOdt*t_NO*30.01/m_all;

end
```

Appendix C: NOx 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-2.3*10^3)/(7.6*10^2);
    K_p_CO2 = -.55*z^3+1.5*z^2-3*z+9.1;
    Z=(T_NO-2.3*10^3)/(7.6*10^2);
    K_p_CO = -.15*Z^3+.41*Z^2-.92*Z+7.1;
    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));
    %x_CO=x_CO+(ALPHA/n_prod);
end

%Calculate Equilibrium Concentrations
X_O2_e=x_O2*P_BDC/(R_u*T_NO);
%___________________________________________________________________________

%Equilibrium Constant For O2 to Oxygen Reaction
Kp_7=3.6*10^3*exp(-31090/T_NO)*318.3; %318.3 converts atm^(1/2) to pa^(1/2)
x_O_e= (Kp_7*X_O2_e^(1/2))/((R_u*T_NO)^(1/2))/(P_BDC/(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_BDC/(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 D: 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/15.09)*14.7);
%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/15.09)*...
    14.7*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 E: Correction Factor Function

```
function [ CF ] = correction( Load,RPM )
if Load<=1
   %CF = (-3*10^(-9))*RPM^2+5*10^(-5)*RPM+.7088;
   CF=-7.9*10^(-9)*RPM^2+.00005*RPM+.73;
   if Load<=.9
       CF=-8*10^(-9)*RPM^2+.00005*RPM+.73;
       %CF = CF-(1-Load)/4;
   end
end

end
```

Appendix F: Instructions of ME 433 Mini-Project

In this project, you will simulate a 4-stroke (preferably multi-cylinder), spark-ignition, naturally aspirated engine using a MATLAB two-zone heat release model. You will choose an engine about which the bore, stroke, connecting rod length, and other manufacturer's information can be found. Based on the engine chosen, you will modify the volumetric efficiency and engine friction as a function of engine speed. You will also assign air-fuel ratios to different engine operating points and determine MBT timing. You will be expected to relate your findings to typical engine design and operating parameters that we have explored throughout this course.

A set of MATLAB files is provided to assist you in this effort. All of these can be found on the Mindworks website. Make sure to save all MATLAB files in the same directory or folder, and do not change the name of the files (these are referenced throughout the main code). Before getting started, review lecture material on laminar flame speed, EGR, and engine modeling equations. These are summarized in lecture 25.

1. Put your engine's bore, stroke, connecting rod length, number of cylinders, compression ratio, and predicted valve timing into the model. Exhaust valve timing can be found on performance camshaft websites (remember, this model runs from 0 to 360 degrees, with 180 degrees being TDC).

2. If your engine has EGR, comment-out line 369 of the two-zone script, and uncomment line 368. This corrects the inlet temperature based on EGR. Note: In order to see the effects of EGR on NO, the timing and burn duration have to be adjusted (see pages 836-841 in the textbook).

3. Based on the chosen engine, construct a customized volumetric efficiency profile and update this within the volumetric efficiency function (provided on the Mindworks website). This should be as simple as creating a few volumetric efficiency data points (relative to engine speed) and curve-fitting these points (in Excel or Matlab).

4. Based on the chosen engine, modify the linear friction equation. The friction equation provided in the two-zone script (line 53) is adjusted for motorcycle operation. Reference your textbook or Blair's textbook (cited in Jeremy's thesis) for help adjusting this. Site any external sources that you use.

5. Based on the chosen engine, construct a custom air-fuel ratio map over your engine's operating range. You might be able to find an aftermarket map, or you might have to make assumptions based on the type of engine. This is for the purpose of step 6; don't get too obsessive about this. Site any external sources that you use.

6. Run your MATLAB code using pre-assigned speeds, loads, air-fuel ratios (by changing lambda, line 78), and burn-durations. The burn-duration will need to be adjusted relative to engine speeds and air-fuel ratios (pages 390-395 and 827-829 from the textbook describe changes in burn duration and spark-timing, respectively). Adjusting burn duration is extremely important to getting accurate results, so spend a little time on it.

7. Find the MBT spark-timing for each speed and load point. Record your observed imep and MBT setting. You should have at least 15 points. Suggested point distribution includes 5 different RPM settings and 3 throttle positions at each RPM. This will ensure accurate BSFC and emissions maps.

8. At each operating point, store RPM, Torque, BSFC, NO emissions, and HC emissions in an Excel file. Each output (i.e. RPM, Torque, etc.) should be stored in an individual column of the Excel file. Make sure that RPM, Torque, and BSFC values are stored in the first 3 columns of the Excel file. Save the variable as "data". The file "JeremyData" is provided as an example (make sure to save the Excel file in the same directory as the BSFC file).

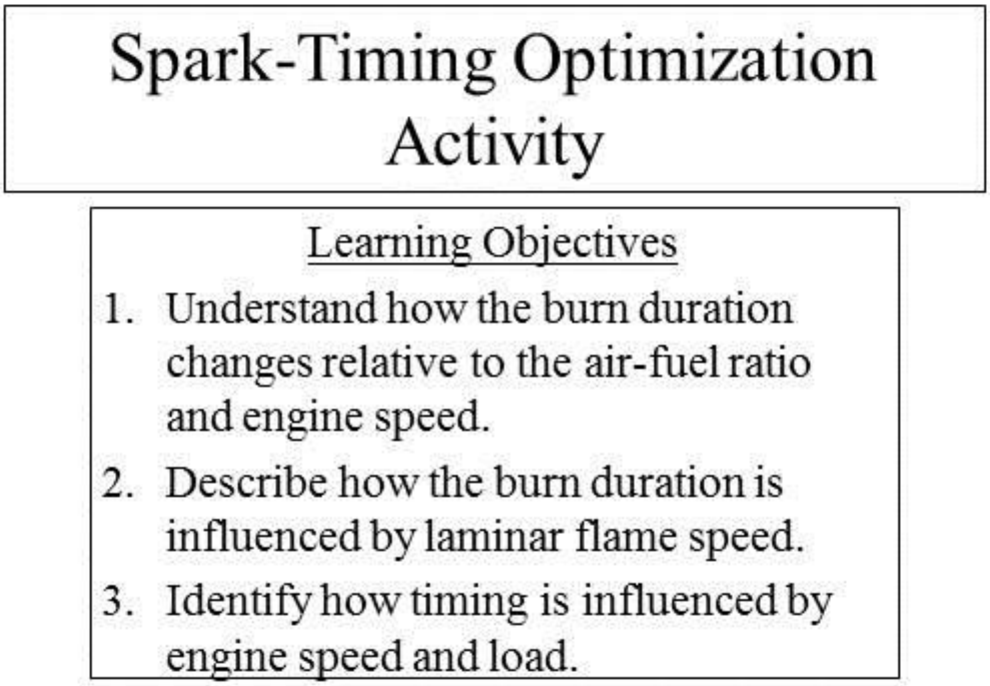
9. In the "BSFCcode.m" file, update line 8 to "`jdata=xlsread('data');`"

10. By running the "BSFCcode.m" file, a BSFC map will be produced. Construct NO and HC emissions maps by adding additional lines and plots to the "BSFCcode.m" file (i.e. NO and HC emissions should already be stored in the Excel file, simply plot these in the same way as RPM, Torque, and BSFC values were plotted to create BSFC maps).

You will need to turn in:

- Documentation of all equations and variables used (i.e. BSFC, torque, etc.)
- Even if your engine doesn't have it, discuss how EGR affects combustion pressures, temperatures, and NO production.
- Plot and discussion of spark-timing at a few operating points
- Plot and discussion of power and torque curves (vs. speed) compared with actual power and torque (vs. speed)
- Plot and paragraph discussion of BSFC and BSemissions plots for your engine.
- Compose a ½ page reflection on your two most valuable lessons learned about engine heat release modeling through this assignment, along with aspects of the simulation that you would like to learn more about/refine.
- Rate your performance on this exam using the problem solving rubric. Write a paragraph on the two greatest strengths exhibited in your engineering problem solving/documentation, why these are valuable, and why they should be sustained. Write a second paragraph on your two greatest areas for improvement in your engineering problem solving/documentation along with an action plan how to make each improvement.

Appendix G: Spark Timing Tips



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
```