

DEVELOPMENT OF A HIGH-FIDELITY ENGINE MODELING FRAMEWORK IN  
SIMULINK WITH AUTOMATED COMBUSTION PARAMETER TUNING

by

BRADLEY ADAM THOMPSON

HWAN-SIK YOON, COMMITTEE CHAIR

TIM HASKEW

PAULIUS PUZINAUSKAS

ALEXEY VOLKOV

KEITH WILLIAMS

A DISSERTATION

Submitted in partial fulfillment of the requirements  
for the degree of Doctor of Philosophy  
in the Department of Mechanical Engineering  
in the Graduate School of  
The University of Alabama

TUSCALOOSA, ALABAMA

2017

Copyright Bradley Adam Thompson 2017  
ALL RIGHTS RESERVED

## ABSTRACT

The automotive industry continually seeks to improve performance and fuel efficiency due to increasing fuel costs, consumer demands, and greenhouse gas regulations. With advancements in computer-aided design, engine simulation has become a vital tool for product development and design innovation, and as computation power improves, the ability to optimize designs improves as well. Among the simulation software packages currently available, Matlab/Simulink is widely used for automotive system simulations but does not contain a detailed engine modeling toolbox. To leverage Matlab/Simulink's capabilities, a Simulink-based 1D flow engine modeling architecture is proposed. The architecture allows engine component blocks to be connected in a physically representative manner in the Simulink environment, therefore reducing model build time. Each component model, derived from physical laws, interacts with other models according to block connection.

The presented engine simulation platform includes a semi-predictive spark ignition combustion model that correlates the burn rate to combustion chamber geometry, laminar flame speed, and turbulence. Combustion is represented by a spherical flame propagating from the spark plug. To accurately predict the burn rate, the quasi-dimensional model requires tuning. A method is proposed for fitting turbulence and burn rate parameters across an engine's operating space. The method reduces optimization time by eliminating the intake and exhaust flow models when evaluating the fitness function. Using the proposed method, 12 combustion model parameters were optimized to match cylinder pressure. Optimization and validation results are given for a 2.0 L Mazda Skyactiv-G engine.

## ACKNOWLEDGEMENTS

First, I would like to thank God for giving me discernment, ability, and desire to pursue a career in engineering. I would like to thank my parents for teaching me the way which I should go, and through action, modeled dedication and hard work. I would like to thank my wife Emily—my degree could not be completed without her patience, support and encouragement. I am grateful for and indebted to Dr. Yoon, who has influenced and prepared me for future endeavors. I would like to thank my advisory committee for their time, feedback, and advice. Finally, I would like to thank the EPA for providing test data and information, and Dr. Byungho Lee for mentoring me while interning at the EPA.

## CONTENTS

ABSTRACT .....	ii
ACKNOWLEDGEMENTS .....	iii
LIST OF TABLES .....	ix
LIST OF FIGURES .....	x
CHAPTER 1: INTRODUCTION .....	1
CHAPTER 2: LITERATURE REVIEW .....	4
2.2 Mean-Value and Filling Models .....	5
2.3 One-Dimensional Flow .....	7
2.4 Spark-Ignition Combustion Model .....	8
2.5 Multidimensional Models .....	11
CHAPTER 3: ENGINE 1D FLOW MODEL .....	13
3.1 Physics-Based Approach .....	13
3.2 Thermodynamic Properties .....	13
3.2.1 Equation of State .....	14
3.2.2 Internal Energy and Enthalpy .....	15
3.2.3 Gas Mixtures .....	17

3.2.4	Thermodynamic tables .....	19
3.3	Quasi-One-Dimensional Unsteady Flow .....	20
3.3.1	Conservation Laws.....	20
3.3.2	Spatial Discretization .....	23
3.3.3	Numerical Integration .....	27
3.3.4	Friction Factor.....	28
3.3.5	Pipe Bends .....	30
3.3.6	Heat Transfer .....	34
3.4	Pressure Wave Motion.....	35
3.5	Flow Restrictions and Adjoined Pipes.....	39
3.5.1	Model Setup and Conservation Laws .....	40
3.5.2	Boundary Parameter Relationships.....	41
3.5.3	Solution Overview .....	45
3.5.4	Boundary Mass and Energy Flow Rates.....	50
3.6	Boundary Conditions .....	52
3.6.1	Model Setup and Conservation Laws .....	52
3.6.2	Boundary Parameter Relationships.....	54
3.6.3	Pipe Inflow Constraints.....	57
3.6.4	Pipe Outflow Constraints.....	61

3.6.5	Solution Overview .....	63
3.6.6	Boundary Mass and Energy Flow Rates .....	64
3.7	Flow Junction.....	66
3.7.1	General flow junction .....	66
3.7.2	T-junction and Y-junction.....	69
CHAPTER 4: ENGINE CYLINDER AND DYNAMICS MODELS.....		70
4.1	Crank Dynamics.....	70
4.2	Cylinder Conservation Laws.....	73
4.2.1	Single-Zone and Gas Exchange Period.....	73
4.2.2	Two-Zone Combustion Model.....	76
4.3	Cylinder Charge Motion and Turbulence .....	79
4.3.1	Mean Flow Model.....	79
4.3.2	Turbulence Model.....	85
4.4	Spark-Ignition Combustion Burn Rate .....	90
4.4.1	Wiebe Burn Rate.....	91
4.4.2	Turbulent Entrainment Model.....	92
4.5	Laminar Flame Speed .....	95
4.6	Combustion Chemistry .....	97
4.7	Heat Transfer .....	101

CHAPTER 5: SIMULINK ENGINE MODEL.....	104
5.1 Simulink Architecture Overview and User Interface.....	104
5.1.1 Block Connection.....	105
5.1.2 S-function Interface .....	107
5.1.3 Block Interface.....	109
5.1.4 Model Control Interface.....	112
5.2 S-Function.....	113
5.2.1 Model Representation in S-Function .....	114
5.2.2 Engine Component C++ class.....	114
5.3 Simulation.....	115
5.3.1 Simulation Setup.....	116
5.3.2 S-Function Method Execution .....	119
5.3.3 C++ Method Execution.....	121
5.4 Steady State Save and Restart.....	126
5.5 Example Model.....	128
5.5.1 Simulation Setup.....	128
5.5.2 Results and Discussion .....	130
CHAPTER 6: MODEL TUNING AND VALIDATION.....	132
6.1 Mazda Skyactiv-G® Engine.....	132

6.2 Simulink Model .....	135
6.2.1 Overview .....	135
6.2.2 Control .....	136
6.2.3 Plant Model .....	138
6.3 Combustion Model Tuning .....	142
6.3.1 Optimization Parameters .....	143
6.3.2 Fitness Function .....	147
6.3.3 Optimization Procedure .....	150
6.3.4 Optimization Results .....	154
6.4 Validation .....	160
CHAPTER 7: CONCLUSIONS .....	170
7.1 Summary .....	170
7.2 Future Works .....	173
REFERENCES .....	174
APPENDIX A: CONTROL TABLES .....	184
APPENDIX B: SKYACTIV ENGINE MODEL .....	193

## LIST OF TABLES

Table 3.1: Solution summary for adjoined pipes .....	50
Table 3.2: Solution summary for boundary conditions .....	64
Table 4.1: Laminar flame speed coefficients for methanol, propane, isooctane, and gasoline.....	96
Table 4.2: Coefficients for Woschni heat transfer correlation.....	103
Table 5.1: Engine model component class methods.....	111
Table 5.2: Engine model component class properties .....	112
Table 5.3: “EngineModelComponent” C++ class methods.....	116
Table 6.1: Specifications for 2.0L Mazda Skyactiv-G engine.....	133
Table 6.2: Constant turbulence parameters.....	146
Table 6.3: Combustion model parameters and constraints .....	147
Table 6.4: Initial and optimized combustion model parameters for minimum IMEP <sub>n</sub> error, minimum peak cylinder pressure error, and minimum combined fitness function.....	156
Table 6.5: Fitness values for initial parameters, minimum IMEP <sub>n</sub> error, minimum peak cylinder pressure error, and minimum combined fitness function .....	157

## LIST OF FIGURES

Figure 3.1: Control volume for compressible, unsteady flow .....	21
Figure 3.2: Staggered grid discretization .....	24
Figure 3.3: Circular pipe bend parameters.....	31
Figure 3.4: Round pipe bend loss coefficient $K_b^*$ at $Re = 10^6$ .....	32
Figure 3.5: Bend loss coefficient Reynolds correction factor $C_{Re}$ .....	33
Figure 3.6: Acoustic waves traveling in pipe.....	38
Figure 3.7: Schematic of adjoined pipes with restrictive orifice .....	41
Figure 3.8: Forward (a) and reverse (b) parameter constraints for adjoined pipe boundary (unknowns circled) .....	47
Figure 3.9: Mass and energy flow rate across adjoined pipe boundary .....	52
Figure 3.10: Schematic of 1D pipe boundary condition .....	54
Figure 3.11: Boundary constraints and unknown parameters for (a) subsonic and (b) choked inflow (unknowns circled) .....	59
Figure 3.12: Boundary constraints and unknown parameters for (a) subsonic and (b) choked outflow (unknowns circled) .....	62
Figure 3.13: Mass and energy flow rate across boundary .....	65
Figure 3.14: Junction (a) orientation vectors and (b) general flow representation .....	67
Figure 3.15: General junction characteristic lengths and pressure calculation parameters .....	68
Figure 3.16: (a) T-junction and (b) Y-junction model parameters .....	69
Figure 4.1: Engine slider-crank geometry .....	72
Figure 4.2: Diagram of engine cylinder control volume.....	74
Figure 4.3: Diagram of two-zone combustion model .....	76

Figure 4.4: Simplified cylinder flow model characterized by swirl angular momentum $L_z$ and tumble angular momentum $L_x$ .....	80
Figure 4.5: Charge motion decay functions $\Psi_x$ and $\Psi_z$ derived from 3D Taylor-Green vortex simulations .....	84
Figure 4.6: Images of (a) early flame development, (b and c) flame propagation, and (d) flame termination.....	90
Figure 4.7: Cylinder pressure and Wiebe mass fraction burned profile .....	92
Figure 4.8: Turbulent entrainment model zones with spherical flame front expanding from the spark location.....	94
Figure 5.1: Communication of system variables through block connections.....	105
Figure 5.2: Possible component block connection methodologies: (a) standard input/output, (b) two-way connector, and (c) single input/output connection .....	106
Figure 5.3: Example sensor configurations create virtual connections (dashed lines): (a) pressure sensor, (b) pressure and mass flow rate sensor connected in series, (c) flow subsystem with pressure sensor, and (d) internal view of flow subsystem.....	107
Figure 5.4: (a) Example of a 1D flow model with pressure boundary conditions and sensor, (b) view of “Engine Model Control” subsystem, and (c) view of “Pressure Sensor” subsystem.....	109
Figure 5.5: Mask dialog box for “1D Flow Section” block.....	110
Figure 5.6: Global model control graphical user interface .....	113
Figure 5.7: Simulation setup flowchart.....	118
Figure 5.8: S-function setup and simulation flowchart.....	120
Figure 5.9: EngineModelComponent virtual method execution order for mdlStart.....	123
Figure 5.10: EngineModelComponent virtual method execution order for (a) mdlOutputs, (b) mdlGetTimeofNextVarHit, and (c) mdlUpdate.....	124
Figure 5.11: Simulated pipe parameters and boundary conditions.....	129
Figure 5.12: (a) GT-Power and (b) Simulink block diagrams of the 1D flow model.....	130
Figure 5.13: Simulated inlet and outlet mass flow rates from GT-Power and Simulink.....	131
Figure 6.1: Steady state speed and torque points tested by EPA .....	134
Figure 6.2: Measured BSFC (g/kWh) for 2.0L Mazda Skyactiv-G engine .....	135

Figure 6.3: Simulink model of Skyactiv engine .....	136
Figure 6.4: Controller subsystem in Simulink engine model .....	137
Figure 6.5: Full Simulink model for Skyactiv engine with throttle, intake manifold, cylinders, crank dynamics, and exhaust system .....	139
Figure 6.6: Simplified Simulink model for Skyactiv engine with cylinder and crank dynamics only, requiring port flow to be imported from full engine simulation .....	140
Figure 6.7: Skyactiv engine combustion chamber and spherical flame displayed in Matlab.....	141
Figure 6.8: Spherical flame surface area for Skyactiv engine combustion chamber.....	141
Figure 6.9: Burned volume ratio $V_b/V_{cyl}$ for Skyactiv engine combustion chamber.....	142
Figure 6.10: Tumble number parameterization .....	144
Figure 6.11: Tumble decay function $\Psi_x$ parameters .....	145
Figure 6.12: Fitness function evaluation points.....	148
Figure 6.13 Flowchart for parameter optimization.....	153
Figure 6.14 Pareto optimal and sub-optimal results from multi-objective optimization (not all fitness values are shown in order to view Pareto optimal results).....	155
Figure 6.15: Resulting tumble number cubic spline for minimum $f_{total}$ .....	157
Figure 6.16: Resulting tumble decay function for minimum $f_{total}$ .....	158
Figure 6.17: IMEP <sub>n</sub> error at optimization operating points for minimum $f_{IMEP_n}$ , $f_{peak}$ , and $f_{total}$ fitness functions and initial parameters .....	159
Figure 6.18: Peak cylinder pressure error averaged across cylinders at optimization operating points for minimum $f_{IMEP_n}$ , $f_{peak}$ , and $f_{total}$ fitness functions and initial parameters.....	160
Figure 6.19: Air flow controlled IMEP <sub>n</sub> error at validation and optimization operating points .....	161
Figure 6.20: Air mass flow rate error (derived from measured air mass flow fit) at validation and optimization operating points .....	162
Figure 6.21: Percent air mass flow rate error at simulated engine speed and torque .....	163
Figure 6.22: Peak cylinder pressure error averaged across cylinders at validation and optimization operating points .....	164

Figure 6.23: Percent peak cylinder pressure error averaged across cylinders at simulated engine speed and torque .....	164
Figure 6.24: Measured (10 cycles) and simulated cylinder pressures at 2000 rpm and 130 Nm .....	165
Figure 6.25: Measured BSFC (g/(kW·h)).....	167
Figure 6.26: Simulated BSFC (g/(kW·h)) .....	168
Figure 6.27: IMEP <sub>n</sub> and peak pressure averaged across cylinders as a function of spark advance at 4500 rpm and 0.92 normalized air charge (185 Nm torque at measured spark advance) .....	168
Figure 6.28: Simulated Cylinder 1 pressure using measured spark angle and 3 additional degrees advance at 4500 rpm and 0.92 normalized air charge (185 Nm torque at measured spark advance).....	169
Figure A.1: Spark advance table (deg) fit with RBF model and residuals .....	185
Figure A.2: Intake valve open crank angle table fit with GPM and residuals .....	186
Figure A.3: Exhaust valve close crank angle table (deg) fit with GPM and residuals .....	187
Figure A.4: Relative air-fuel ratio table fit with RBF and residuals.....	188
Figure A.5: First start injection angle table (deg) fit with RBF and residuals.....	189
Figure A.6: Second start injection angle table (deg) fit with RBF and residuals .....	190
Figure A.7: First injection pulse width table (ms) fit with RBF and residuals.....	191
Figure A.8: Second injection pulse width table (ms) fit with RBF and residuals .....	192
Figure B.1: Intake and exhaust valve lift for 2.0 L Mazda Skyactiv engine .....	193
Figure B.2: Intake and exhaust valve effective flow area used in Simulink engine model.....	194

# **CHAPTER 1**

## **INTRODUCTION**

Numerical simulation has become an integral part of automotive research, design, and innovation. With the ability to build virtual models, several design iterations can be executed without the high cost of producing physical prototypes, and further understanding of physical phenomenon can be extracted from simulation. Numerical simulation has benefited many automotive design areas. Among the various areas, engine and drivetrain simulation have received a great deal of attention. Increasing fuel costs and demands to lower greenhouse gas emissions have driven manufacturers to continually innovate, and the automotive research community has invested in engine simulation techniques to meet future demands. Engine simulation allows designers to predict performance gains resulting from changes in engine geometries or control strategies. As a result, designs can be optimized for fuel economy, power, and emissions without collecting extensive experimental data. With the steady advancement in the related technology and available computing power, the impact of engine simulation will increase as well.

Based on the desired outcome and available information, engine models can take on several forms. For simplicity and accuracy, engine performance can be simulated using a lookup table containing engine torque, speed, and fuel consumption characteristics. Building the lookup table requires extensive experimental testing—not a predictive approach. By introducing physical engine parameters and physics-based models, experimental data collection can be reduced, and design concepts can be tested without prototype hardware. On the low end of predictability, mean-value models combine the overall effect of engine flow and combustion

phenomena. Because such models require little physical detail, some of the parameters must be derived from experimental testing, and some characteristics of the engine performance cannot be accurately determined. As an advantage, however, mean-value models require little computation time and are convenient for control applications. For high predictability, multidimensional Computational Fluid Dynamics (CFD) can be employed to simulate engine flow and combustion. Multidimensional CFD models require detailed geometric parameters, which in turn provide detailed performance information without experimental tuning. This highly predictive approach comes at the cost of long simulation times. Therefore, multidimensional CFD models cannot be applied to simulating numerous engine cycles.

In order to accurately simulate engine performance for multiple cycles, the dimensions of physics-based equations need to be reduced. Intake and exhaust systems consist of a network of internal flow pipes, tapered ducts, valves, and junctions. Reducing engine intake and exhaust flow to 1D internal flow greatly reduces computation time and still provides accurate results with minimal experimental tuning. In this approach, the engine cylinder model, represented as 0D control volume, estimates pressure and force provided to the piston. Since the fuel burn rate depends on cylinder turbulence, the 0D cylinder models predict burn rate from an experimentally-fit burn profile or a turbulence correlation. With the reduced dimension approach, multiple engine cycles can be simulated in a time-efficient manner with less experimental tuning than a mean-value model, allowing control algorithms and design concepts to be evaluated without changing the physical system.

Combustion models vary in complexity and can be categorized as burn rate fit or semi-predictive models. Burn rate fit models match the combustion burn rate based on cylinder pressure measurements. Because the burn rate depends on several factors, the models are only

appropriate near the tested operating conditions. Semi-predictive combustion models, on the other hand, model combustion at a wide range of operating conditions. Unlike the burn rate fit model, semi-predictive models include tuning parameters independent of the operating conditions. The parameters can be tuned by matching the cylinder pressure at a wide range of operating conditions. Once tuned, the model can be used for desktop calibrations. In order to react to control inputs, the complexity of the semi-predictive combustion model can be adjusted.

Rapidly assembling an engine model requires a logical and convenient design architecture. Among the simulation software packages currently available, Matlab/Simulink is widely used in the automotive industry and academia. Simulink is a powerful tool for modeling physical systems, designing controllers, and simulating prototypes. However, Simulink-based high-fidelity engine modeling packages are not currently available. Therefore, to simulate vehicle performances in Simulink, a third-party engine simulation software package must be connected to Simulink. Implementing a third-party software increases build time and restricts Simulink's capabilities. To leverage MATLAB/Simulink capabilities and provide model development flexibility, a new engine modeling architecture employing the 1D wave propagation theory in the framework of Matlab/Simulink is presented in this dissertation.

The Simulink engine model includes a semi-predictive combustion model. As a proof of concept, the combustion model was tuned and validated with data provided by the Environmental Protection Agency (EPA). The automated tuning method significantly reduced optimization time by running the full engine model and saving intake and exhaust flow once the model reached steady state. The information from the full engine simulation was then used to mimic the intake and exhaust manifold during optimization, allowing tuning parameters to be tested without simulating the intake and exhaust flow.

## **CHAPTER 2**

### **LITERATURE REVIEW**

Engine modeling and simulation, which can include the entire engine or specific process, has been a major contributor to automotive design and research. By simulating an entire engine, the complex interaction between components and control strategies can be examined. On the other hand, modeling specific processes such as combustion provides a detailed understanding of phenomenon within the engine cylinder. The level of required detail dictates the modeling strategy, but regardless of the modeling approach, the primary objectives remain consistent. Heywood suggests four major objectives of engine modeling [1]:

1. Gain knowledge of an engine as a whole or specific processes through formulating the engine model;
2. Identify key engine design parameters that allow for more rational and therefore less costly experimental efforts;
3. Predict engine behavior for a wide range of designs and operating variables prior to testing costly physical hardware by evaluating trends and tradeoffs, and possibly optimizing the design and control;
4. Provide a logical foundation for innovation—predict the effect of engine design innovations before conducting extensive experimental research.

When deriving a mathematical model that describes or represents a physical system, the assumptions and detail associated with the model dictates the model accuracy. Engine models can be derived exclusively from experimental testing, but in order to predict future performance,

some level of physics-based modeling must be introduced. Depending on the level of detail, accuracy, and model tuning required, physics-based models widely vary in complexity and predictability, but all employ physical parameters. Less predictive models, such as a mean value model, require few physical parameters and little computation time but provide little insight. On the other hand, highly predictive models provide detailed insight to physical phenomenon and require precise engine geometry but require long simulation times. Highly predictive models employ 3D Computational Fluid Dynamics (CFD) to predict in-cylinder and manifold flow characteristics, while filling-and-emptying models lump the manifold into larger volumes and neglects flow momentum. As an alternative, flow in engine ducts are frequently assumed to be 1D, thus requiring less computation time than a 3D CFD model and providing more detail than a filling-and-emptying model.

## **2.2 Mean-Value and Filling Models**

Until the 1970's, internal combustion engines have been primarily tested in steady state conditions—constant torque and crankshaft speed—thus models were only required to describe steady state conditions [2]. Such models could be input/output or physics-based [3]. With increasing demands to improve fuel economy and enforcement of government emission regulations, engine control design required transient behavior to be modeled. Powell formally introduced a simple physics-based model that considered crank dynamics, fuel system, and engine flow [4]. By lumping every cylinder into a single volume and neglecting torque fluctuations within each cycle, Powell could fit engine torque to spark advance, intake mass flow rate, and air-fuel ratio. Several other researchers helped pioneer engine simulation in regard to optimizing control: Delosh [5], Dobner *et al.* [6]–[8], Cho and Hedrick [9], and Moskwa *et al.* [10]–[12].

Later, Hendricks and Sorenson created a modeling architecture termed “mean-value model” [13]. For spark ignition engines, a mean-value models consist of three basic subsystems: fuel dynamics, crankshaft dynamics, and manifold dynamics [2]. The fueling subsystem predicts the time averaged mass flow rate of vaporized fuel entering the cylinder based on an evaporation time constant and fraction of injected fuel deposited to the port walls. The crank dynamics subsystem uses the manifold pressure, friction losses, thermal efficiency, and fuel flow rate to predict torque and rate change in crankshaft speed. The manifold subsystem predicts the total intake air mass flow rate based on engine speed and manifold pressure. Since the introduction of the model, mean-value models have been adapted to simulate turbocharged engines as well [14].

The mean-value model lumps every cylinder into a single set of equations and does not account for air entering each cylinder, requiring flow dynamics to be fit from experimental data using nonlinear regression [2]. To improve model accuracy and reduce model tuning, a “filling-and-emptying” model can be utilized. A filling-and-emptying model represents engine manifolds (or sections of the manifolds) as finite volumes [1]. Each control volume contains a uniform gas, whose thermodynamic states - pressure, temperature, and density - can be derived from conservation of mass and energy (first law of thermodynamics) equations. Cylinder valves, boundaries, throttle valve, and air cleaner are represented as flow restrictions that can be connected to the flow volumes. With manifold volumes and flow restrictions arranged to match the actual engine, fuel and air mass flow rate entering the cylinder can be determined without extensive experimental tuning. However, the improvement in accuracy and reduction in tuning translates into a longer computation time than a mean-value model. The increase in computation time does not allow full engine filling-and-emptying modes to be applied to real-time control, but a simplified model has been used predict real-time exhaust flow [15].

### 2.3 One-Dimensional Flow

Filling-and-emptying models predict manifold thermodynamic states according to conservation of mass and energy laws. Without simulating flow velocity, however, manifold tuning cannot be evaluated, thus requiring a more detailed modeling approach [16]. To predict flow velocity, conservation of momentum must be introduced. Flow in engine ducts (e.g. intake/exhaust ports, runners, and pipes) can be fully described with a 3D flow field. However, since majority of the gas flows collinearly with the pipe, the solution can be simplified to 1D by assuming a uniform average flow velocity across the cross section. Coupling 1D conservation of momentum with the conservation of mass and energy equations results in three nonlinear partial differential equations. For a constant cross-section area pipe, the conservation laws result in the 1D version of the Euler equations, and by accounting for variable cross-sectional area and friction, the model is frequently called “quasi-1D” flow model.

With the 1D conservation laws, the propagation and reflection of acoustic waves in the engine duct can be tracked, allowing manifold tuning to be evaluated during simulation. De Haller first used the method of characteristics to graphically track traveling acoustic waves in an exhaust system [17]. Jenny later extended the model to include friction, heat transfer, entropy gradient, and changes in pipe area [18]. Due to the labor and time required with graphical methods, Benson *et al.* devised a numerical method for simulating flow through engine ducts using Riemann variables [19]. Benson then presented two programs for complete cycle synthesis which involved engine intake, exhaust, turbocharger, and heat release [20]. As an alternative, Blair developed a method based on acoustic wave theory [21]. Since the introduction of the method, numerous references outline the boundary condition, engine valve, and flow junction models [22]–[24]. Additionally, the model has been thoroughly validated [22]–[25].

The techniques developed by Benson and Blair track acoustic wave traveling in the 1D engine duct, but the 1D conservation laws can also be solved with various numerical schemes. Among the schemes, the single-step and two-step Lax-Wendroff method has frequently been used to simulate 1D engine flow [26]–[28]. When compared to Benson’s method of characteristics technique, the two-step Lax-Wendroff method provides a faster solution and more accurately satisfies the continuity equation [29]. However, the Lax-Wendroff produces large non-physical pressure and velocity oscillations when encountering large pressure gradients, which can frequently occur when opening the exhaust valve [1], [30]. To reduce the oscillations, artificial viscosity has frequently been added to the model [27], [28].

Numerous other methods have been explored to better represent 1D engine flow. For example, Bozza and Gimelli used a Total Variation Diminishing (TVD) method to create a comprehensive two-stroke engine model [31]. When compared to the Lax-Wendroff schemes, TVD methods produce more accurate results without generating large non-physical oscillations but require longer computation times [30]. Therefore, to capture discontinuities and reduce computation time, researchers in recent years have used Conservation Element-Solution Element (CE-SE) method for engine modelling [30], [32], [33].

## **2.4 Spark-Ignition Combustion Model**

Using a filling-and-emptying or 1D flow model, air and fuel entering and exhaust gases exiting the cylinder can be determined. Unlike the engine ducts, the cylinder volume cannot be readily assumed as 1D but is better described by a 3D flow field. The intake stroke creates large turbulent motions due to piston movement and flow through the intake valve, and the turbulent flow field becomes an important part of flame propagation and convective heat transfer. From a computational standpoint, modeling 3D in-cylinder motion for multiple cycles poses significant

problems. Therefore, when not requiring extreme detail, the cylinder is represented as a uniform control volume. Without flow velocity, the cylinder model is often referred to as “0D.”

Turbulence and swirl in the chamber during flame propagation allows the flame to propagate at a higher rate, thus making piston speed, intake geometry, and chamber design an important factor. With the 0D combustion chamber, turbulence cannot be accurately predicted, requiring a correlation to model burn rate. Frequently, 0D cylinder models employ a Wiebe mass burn profile. The Wiebe function, first introduced by Ivan Wiebe, relates engine crank angle  $\theta_c$  and ignition timing  $\theta_0$  to the fraction of burned fuel  $y_{burn}$  contained in the cylinder [34]:

$$y_{burn} = 1 - \exp\left(-a\left(\frac{\theta_c - \theta_0}{\Delta\theta}\right)^{m+1}\right), \quad (2.1)$$

where  $a$  and  $m$  are fitting parameters and  $\Delta\theta$  is the combustion duration. Parameter  $m$  defines the shape of the mass fraction burned profile, while  $a$  models combustion efficiency. Because cylinder turbulence varies with engine speed and load, parameters  $\Delta\theta$ ,  $a$ , and  $m$  vary with engine conditions and must be found experimentally. Typically, cylinder pressures are measured at various engine speeds and loads, and the burn fraction profile is derived from a first law analysis (conservation of energy) [22], [35]–[41].

More fundamental combustion representations have been proposed as well. Blizard and Keck formulated a turbulent burning model that correlates the laminar flame speed and cylinder turbulence to the turbulent flame speed [42]. After the introduction to the concept, Keck and coworkers further improved the turbulence burning law [43]–[45]. Later Tabaczynski *et al.* adapted the burning law and assumed combustion at the Kolmogorov scale to be instantaneous [46], [47]. Tabaczynski’s model, which is frequently referenced in literature, estimates that the entrainment speed as the sum of the laminar flame speed and turbulent intensity. Once entrained,

combustion is assumed to progress at in a laminar fashion at the Taylor microscale. Therefore, a time constant called the “characteristic eddy burn up time,” which is related to Taylor microscale, represents the delay between the mass entrained and the mass burned. The Taylor microscale and turbulent intensity can be estimated by valve flow, cylinder states, and piston motion. For turbulence estimation, Keck proposed a correlation based on piston speed, unburned gas density, and volumetric efficiency [43]. Established multidimensional turbulence models,  $k$ - $\epsilon$  for example, have been adapted for combustion modeling as well [48], [49].

With the burn rate determined by the Wiebe function or turbulence entrainment laws, the effects of combustion can be provided to the cylinder model. The simplest cylinder model lumps the burned and unburned gases into a single volume known as the “single-zone” model, and as combustion progresses, the burn rate predicts heat release in the 0D volume. By not separating the flame and unburned mixture, temperature dependent pollutant formation mechanisms cannot be adequately represented. However, the single-zone can predict hydrocarbon and carbon monoxide emissions due to incomplete combustion [50]. Although the single-zone model greatly simplifies the actual combustion phenomena, Cheung and Heywood concluded the method to be remarkably robust [36]. Given the accuracy and simplicity, many researchers have employed the single-zone combustion model for engine simulation, for example, [35], [37], [39], [40].

For spark-ignition engines, the combustion flame has been experimentally observed to propagate almost spherically having a thin reaction sheet [43], [51], [44]. Therefore, to better represent engine combustion, two-zone models are frequently employed, for example, [31], [41], [52], [53]. The unburned and burned gases are represented by two distinct zones. During combustion, the burned gas volume expands from the point of ignition at the rate specified by the burning law, thus predicting local flame temperatures. The localized flame temperature, when

compared to single zone model, better estimates temperature dependent pollutant formation such as nitric oxide [54]. Separating the temperature zones also better predicts heat transfer and cylinder pressure because the expanding burned gas contact area and development rate are determined by chamber geometry [55]. To more accurately predict unburned hydrocarbons emissions, Jensen and Schramm expanded the two-zone model to three zones by modeling the crevice volume [56].

## **2.5 Multidimensional Models**

With 1D assumptions, flow in engine pipes can be accurately modeled, but when considering manifold junctions and flow through valves, turbulence and 3D flow effects become more of a factor. Simulating multidimensional flow can provide more details than 1D flow models and reduce experimental tuning. On the other hand, increasing the number of dimensions significantly increases computation time and requires more detailed geometrical information. To assess the feasibility of employing multidimensional CFD in manifold design, Chapman introduced a 2D unsteady inviscid flow model in 1979 [57]. The modeling concept allowed designers to examine new manifold designs without producing a physical prototype [58]. Leschziner and Dimitriadis extended the 2D approach to 3D steady flow [59]. The model proved to accurately predict detailed flow during steady operating conditions. Zhao and Winterbone later produced a comprehensive method for simulating unsteady 3D flow in production manifolds [60].

Because flow through engine ducts can be accurately represented as 1D and complex junctions and control volumes are best modeled as 3D, the two modeling hierarchies have been combined to achieve high accuracy with less computation time than a full 3D model. Using commercial CFD packages, automotive researchers have connected 1D and 3D finite volumes to

improve accuracy, for example [61]–[63]. Predicting acoustic properties can also benefit from the coupled approach. Using a Riemann solver, Montenegro *et al.* simulated the acoustic behavior of a silencer by representing the exhaust pipe as 1D and the silencer chamber as 3D [64]. Montenegro *et al.* later developed a method that allows the seamless connection of 1D and 3D finite volumes [65], [66].

In addition to engine flow, multidimensional cylinder simulation has become a topic of interest. With a multidimensional representation of in-cylinder fluid dynamics, detailed local turbulence can be modeled [50]. Haworth, for example, used multidimensional modeling to examine in-cylinder turbulence and could observe cycle-to-cycle flow variation [67]. Brusiani *et al.* used 3D CFD to examine intake geometry's effect on tumble motion generated during intake and compression strokes [68]. Increasing turbulence and swirl of the unburned mixture prior to combustion increases burn rate, thus affecting engine performance [68], [69].

Unlike the 0D cylinder representation, multidimensional models can predict combustion burn rate based on the flow field and chemical kinetic equations but require a much longer computation time. The accurate prediction of local temperatures, when compared to the two-zone combustion model, provide a more accurate prediction of pollutant formation. Therefore, many researchers have utilized CFD and chemical kinetic simulation to predict pollutant formation. Baritaud *et al.* predicted NO and CO production using a 2D cylinder model and a relatively simple multistep mechanisms [70]. With more computation power, detailed multistep reaction mechanisms could be simulated [71]–[73]. Wallesten *et al.*, for example, used a mechanism consisting of 100 species and 475 intermediate reactions to predict flame speed [71]. The simulation matched closely to the observed flow field and measured hydrocarbon emissions.

## **CHAPTER 3**

### **ENGINE 1D FLOW MODEL**

#### **3.1 Physics-Based Approach**

An internal combustion engine can be divided into several distinct components (e.g. pipes, valves, cylinders, rotating assembly) and represented by physics-based equations. With a set of standard component models defined by physical parameters, a full engine model can be formed by assembling individual components. In this manner, an infinite number of engine configurations can be simulated with a finite set of component models. To predict intake and exhaust flow characteristics, a quasi-1D gas dynamics model is employed. Derived from conservation laws, flow component submodels include boundary, valve, junction, and pipe. The flow components predict flow into and out of the combustion chamber, accounting for intake tuning, exhaust tuning, and valve lift characteristics contribution to performance. Each cylinder component model predicts pressure increase caused by combustion based on air and fuel available at the start of combustion. Finally, the crankshaft dynamics component model uses the piston force to predict instantaneous torque and crankshaft rotational acceleration.

#### **3.2 Thermodynamic Properties**

Accurate estimation of gas thermodynamic properties is crucial for physics-based engine models. Combustion converts an air-fuel mixture into exhaust gas products, and because fractions of each gas species affect bulk behavior, intake and exhaust have distinctly different flow characteristics. Additionally, in-cylinder compressibility behavior changes as combustion

reactants are converted to products. Thermodynamic properties used to model engine components are discussed.

### 3.2.1 Equation of State

The state of a gas can be described by three state variables that relate physical conditions. An equation of state provides a relationship between the state variables. By measuring or calculating two state variables, the final variable can be determined by the equation of state. Pressure  $p$ , temperature  $T$ , and density  $\rho$  are used throughout the model derivation as the state variables. Note that density could be replaced by volume  $V$  ( $V = m/\rho$ ) or specific volume  $v$  ( $v = 1/\rho$ ), and temperature can be replaced by enthalpy  $h$  or internal energy  $e$ .

For ideal gas behavior, pressure is proportional to the product of temperature and density. The ideal gas law states

$$p = \rho RT , \quad (3.1)$$

where  $R$  is the specific ideal gas constant. The ideal gas law neglects intermolecular forces, which means that gas molecules do not interact. In reality, molecules interact. However, at low densities, the intermolecular interaction has a negligible effect. If the density becomes exceedingly high, gas molecules interact more frequently and should be represented by a real gas model such as the Van der Waals equation. Implementing a real gas model is much more computationally expensive and complicated than an ideal gas model, and because of the low pressures and relatively high temperatures observed in an engine, the accuracy gain is small. Therefore, the ideal gas model is utilized for each engine component model. The ideal gas assumption provides an accurate prediction of state variables for the low densities observed in intake and exhaust systems. However, if cylinder pressures become exceedingly high, the ideal

gas law may not be an accurate assumption. Such high pressures will only be found in compression-ignition engines with a high compression ratio [74].

### 3.2.2 Internal Energy and Enthalpy

Molecular kinetic energy and intermolecular potential energy contained within a thermodynamic system can be referred to as internal energy  $e$ . Internal energy varies with temperature and specific volume, and in general, the differential change in internal energy can be expressed by

$$de = \left( \frac{\partial e}{\partial T} \right)_v dT + \left( \frac{\partial e}{\partial v} \right)_T dv. \quad (3.2)$$

The partial derivative with respect to temperature can be expressed as a constant-volume specific heat  $C_v$ , formally defined as

$$C_v = C_v(T) = \left( \frac{\partial e}{\partial T} \right)_v. \quad (3.3)$$

For an ideal gas, intermolecular potential energy is neglected. Therefore, the partial derivative in Eq. (3.2) with respect to  $v$  becomes zero. By integrating Eq. (3.2) from a reference temperature  $T_{ref}$ , the internal energy becomes

$$e = e_{ref} + \int_{T_{ref}}^T C_v(T) dT. \quad (3.4)$$

where  $e_{ref}$  is the internal energy at  $T_{ref}$ . When evaluating a system's thermodynamic potential, work potential ( $pv$ ) must be added to the internal energy, which is referred to as enthalpy  $h$ :

$$h = e + pv = e + \frac{p}{\rho}. \quad (3.5)$$

Enthalpy varies with pressure and temperature, and in general, the change in enthalpy can be described by

$$dh = \left( \frac{\partial h}{\partial T} \right)_p dT + \left( \frac{\partial h}{\partial p} \right)_T dp. \quad (3.6)$$

The partial derivative with respect to temperature is referred to as the constant-pressure specific heat  $C_p$ , formally defined as

$$C_p = C_p(T) = \left( \frac{\partial h}{\partial T} \right)_p. \quad (3.7)$$

Integrating Eq. (3.6) and assuming an ideal gas, enthalpy can be defined as

$$h = h_{ref} + \int_{T_{ref}}^T C_p(T) dT. \quad (3.8)$$

Because the engine model deals with chemical reactions, the reference internal energy and enthalpy should be standardized. For each species, a standard enthalpy  $h_f^\circ$  can be defined based on energy associated with bonds. A chemical compound's enthalpy of formation  $h_f^\circ$  is the enthalpy increase associated with the reaction of forming one unit mass of the compound from its elements at a reference state. Several reference states have been published in literature, but temperature  $T_{ref}$  and pressure  $p_{ref}$  are typically taken at atmospheric conditions:  $T_{ref} = 298.15$  K and  $p_{ref} = 1$  atm (101325 Pa) [1]. Typically, enthalpies of formation are set as zero for species in their naturally occurring state at the reference conditions [75]. Diatomic oxygen, for example, occurs naturally at the reference temperature and pressure. Therefore, by definition, the reference enthalpy for diatomic oxygen is defined as

$$h_{ref,O_2} = h_{ref,O_2}^\circ = 0. \quad (3.9)$$

Enthalpies of formation for other species can be standardized in a similar manner, and with the reference enthalpy, Eq. (3.8) can be used to calculate the total enthalpy. Reference internal energy can then be determined using the relationship in Eq. (3.5).

### 3.2.3 Gas Mixtures

To determine bulk thermodynamic properties, each gas species is represented by a mass fraction  $y$ . Assuming an ideal mixture containing  $M$  species, the mass fraction of the  $j^{\text{th}}$  species is defined as

$$y_j = \frac{m_j}{m_{total}}, \quad (3.10)$$

where  $m_j$  is the mass of the  $j^{\text{th}}$  species and  $m_{total}$  is the total mass of the bulk mixture. The total mass, defined by

$$m_{total} = \sum_{j=1}^M m_j, \quad (3.11)$$

constrains the sum of all mass fractions to equal unity. To track each species in the engine ducts and cylinders, an array  $\mathbf{y}$  contains  $M-1$  mass fractions while the final mass fraction is determined by the relationship in Eq. (3.11). The mass fraction array is defined as

$$\mathbf{y} = [y_1 \quad y_2 \quad \dots \quad y_{M-1}], \quad (3.12)$$

where the final mass fraction is given by

$$y_M = 1 - \sum_{j=1}^{M-1} y_j. \quad (3.13)$$

Thermodynamic property accuracy and computational efficiency are influenced by the number of gas species represented.

To predict cylinder and flow behavior, the model requires several standard thermodynamic properties. The mixture specific ideal gas constant  $R$  can be determined by weighting each species constant  $R_j$  with the mass fractions, such that

$$R = R(\mathbf{y}) = \sum_{j=1}^M y_j R_j . \quad (3.14)$$

Specific heats for the  $j^{\text{th}}$  species are a function of temperature. Therefore, the specific heats of the mixture become a function of mass fractions  $\mathbf{y}$  and temperature  $T$ . Similar to the ideal gas constant, mixture specific heats can be determined by

$$C_v = C_v(\mathbf{y}, T) = \sum_{j=1}^M y_j C_{v,j} . \quad (3.15)$$

and

$$C_p = C_p(\mathbf{y}, T) = \sum_{j=1}^M y_j C_{p,j} . \quad (3.16)$$

Likewise, the specific heat ratio  $\gamma$  can be determined by

$$\gamma = \gamma(\mathbf{y}, T) = \frac{C_p}{C_v} . \quad (3.17)$$

Assuming ideal gas properties, internal energy and enthalpy vary with temperature and mass fractions. Thus, mixture internal energy and enthalpy are defined as

$$e = e(\mathbf{y}, T) = \sum_{j=1}^M y_j e_j . \quad (3.18)$$

$$h = h(\mathbf{y}, T) = \sum_{j=1}^M y_j h_j . \quad (3.19)$$

### 3.2.4 Thermodynamic tables

To evaluate mixture properties, specific heats and enthalpy for each species must be accessed from a database. Various thermochemical databases have been published, covering a large range of temperatures and compounds. Among the various resources, JANAF thermochemical tables have been frequently cited [1]. For ideal gasses, enthalpy and heat capacity  $C_p$  are tabulated at discrete temperatures, which can be inconvenient for computer simulations. Therefore, tables are typically fit to a standard high degree polynomial introduced by NASA [76]. The original NASA polynomial uses 7 coefficients to estimate heat capacity, enthalpy, and entropy. To maintain accuracy over a large temperature range, multiple polynomials are fit at different temperature intervals; therefore, a single species can have multiple sets of coefficients. Later a 9 coefficient polynomial was introduced to improve accuracy [77], [78]. The form of the NASA polynomial allows properties to be easily calculated on a per mole or per mass basis, with temperature expressed in Kelvin. The 9 coefficient NASA polynomials for enthalpy and heat capacity are defined as

$$\frac{C_p}{R} = a_1 T^{-2} + a_2 T^{-1} + a_3 + a_4 T + a_5 T^2 + a_6 T^3 + a_7 T^4. \quad (3.20)$$

and

$$\frac{h}{RT} = -a_1 T^{-2} - a_2 T^{-1} \ln(T) + a_3 + \frac{a_4 T}{2} + \frac{a_5 T^2}{3} + \frac{a_6 T^3}{4} + \frac{a_7 T^4}{5} + a_8. \quad (3.21)$$

where  $a_1$  to  $a_7$  defines the specific heat curve and  $a_8$  defines the reference enthalpy. Note that the final coefficient  $a_9$ , which is not shown, defines the reference entropy. Compiled by Burcat and others, a database of coefficients derived from various resources, including the JANAF thermochemical tables, are used for property referencing [79]. Properties for gasoline are fit to the 9 coefficient NASA polynomial using data published by Heywood [1].

### 3.3 Quasi-One-Dimensional Unsteady Flow

Gas flow associated with engine intake and exhaust systems are unsteady: internal energy, density, pressure, and velocity vary with time. Although flow within each duct is best described in three dimensions, the nature of internal flow restricts gases to flow primarily in the axial direction of the duct. Therefore, to reduce model complexity, flow states are defined along a single dimension. Geometrically, each engine duct or pipe section has an inlet and outlet, and cross-sectional flow area varies along the duct. Because flow states vary along a single dimension, the model is 1D. By including a variable cross-sectional area, the model can be considered quasi-1D. Although the quasi-1D approach cannot capture complex flow characteristics, the model can accurately represent the pulsating nature of gas exchange phenomena.

#### 3.3.1 Conservation Laws

To predict flow behavior through an engine duct, rate of changes in flow states need to be determined by conservation laws [1]. Referring to the control volume in Figure 3.1, flow velocity  $U$ , density  $\rho$ , specific internal energy  $e$ , pressure  $p$ , species mass fractions  $y$ , and area  $A$  change over the differential length  $dx$ . Area is a fixed function of  $x$ , and all flow states are a function of time and  $x$ . Wall shear  $\tau_w$  accounts for friction losses and  $\dot{q}_w$  is the wall heat flux.

According to the conservation of mass, the rate of change of the total mass contained in the control volume can be determined by the net mass flow rate crossing the control volume boundary:

$$\frac{\partial}{\partial t}(\rho A dx) = \rho A U - \left[ \rho A U + \frac{\partial}{\partial x}(\rho A U) dx \right]. \quad (3.22)$$

From Eq. (3.22), the quasi-1D differential form of the continuity equation can be derived as

$$\frac{\partial}{\partial t}(\rho A) + \frac{\partial}{\partial x}(\rho A U) = 0. \quad (3.23)$$

Similarly, the continuity equation can be applied to each species contained in the duct, resulting in

$$\frac{\partial}{\partial t}(y_j \rho A) + \frac{\partial}{\partial x}(y_j \rho A U) = 0. \quad (3.24)$$

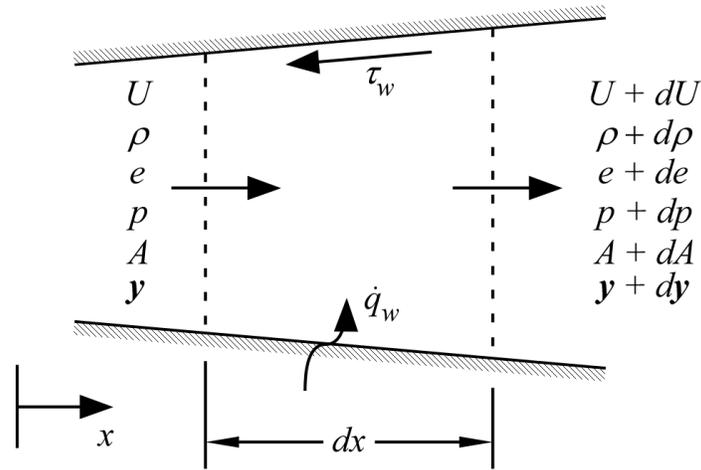


Figure 3.1: Control volume for compressible, unsteady flow

Conservation of momentum, based on Newton's second law of motion, states that the sum of the forces acting on a body equals the body's rate of change of momentum. Therefore, the rate of change of the control volume momentum equals the forces acting on the volume minus the net flow of momentum out of the control volume. For quasi-1D flow, pressure acting on each side of the control volume, pressure acting on the tapered duct, and wall shear account for the total force. The total forces acting on the body can be calculated by

$$pA - \left( p + \frac{\partial p}{\partial x} dx \right) \left( A + \frac{dA}{dx} dx \right) + p \frac{dA}{dx} dx - \tau_w \pi D dx = -A \frac{\partial p}{\partial x} dx - \tau_w \pi D dx, \quad (3.25)$$

where  $D = (4A/\pi)^{1/2}$  is the characteristic diameter. The net flow of momentum crossing the boundary is defined as

$$\left(\rho + \frac{\partial\rho}{\partial x} dx\right)\left(U + \frac{\partial U}{\partial x} dx\right)\left(A + \frac{dA}{dx} dx\right) - \rho U^2 A = \frac{\partial}{\partial x}(\rho U^2 A) dx. \quad (3.26)$$

Finally, the rate of change of the control volume momentum,  $\rho U A dx$ , can be determined by the forces acting on the volume and net momentum crossing the boundary. Using the forces determined in Eq. (3.25) and the momentum flow derived in Eq. (3.26), conservation of momentum leads to

$$\frac{\partial}{\partial t}(\rho U A) + \frac{\partial}{\partial x}(\rho U^2 A) = -A \frac{\partial p}{\partial x} - \tau_w \pi D. \quad (3.27)$$

The rate of change of the control volume's energy can be determined by the first law of thermodynamics. The total work, heat transfer, and net energy crossing volume boundary governs the total energy. Neglecting the effects of gravity, control volume energy  $E$  is the sum of internal energy and flow kinetic energy, such that

$$E = (\rho A dx) \left( e + \frac{U^2}{2} \right). \quad (3.28)$$

Flow energy can cross the boundary in the form of internal energy, kinetic energy, and the work required to transport mass across the boundary. Neglecting shear work, conservation of energy leads to

$$\frac{\partial}{\partial t} \left[ (\rho A dx) \left( e + \frac{U^2}{2} \right) \right] = \dot{q}_w \pi D dx - \frac{\partial}{\partial x} \left[ (\rho U A) \left( e + \frac{p}{\rho} + \frac{U^2}{2} \right) \right] dx. \quad (3.29)$$

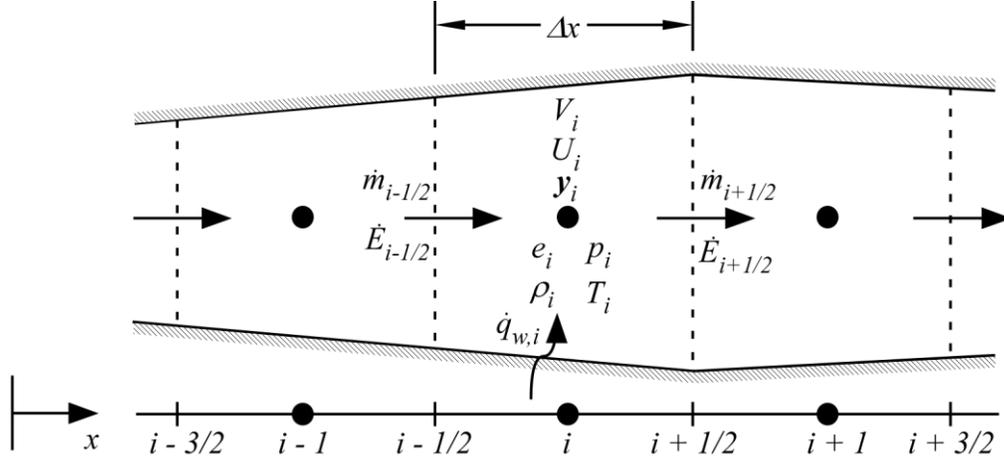
By substituting specific enthalpy  $h$  for the sum of internal energy and flow work, Eq. (3.29)

becomes

$$\frac{\partial}{\partial t} \left[ (\rho A) \left( e + \frac{U^2}{2} \right) \right] + \frac{\partial}{\partial x} \left[ (\rho UA) \left( h + \frac{U^2}{2} \right) \right] = \dot{q}_w \pi D. \quad (3.30)$$

### 3.3.2 Spatial Discretization

The nonlinear partial differential equations derived from the conservation of mass, momentum, and energy cannot be solved analytically. The equations can be converted to ordinary differential equations by replacing the infinitesimal length with a finite length  $\Delta x$  (finite difference) and integrating with a proper ODE solver (e.g. Runge-Kutta and Euler method). To discretize the flow duct, a staggered grid approach is utilized. A staggered grid approach divides the pipe or duct into sections with an equal length  $\Delta x$  as shown in Figure 3.2. At each cell center ( $i = 1, 2, \dots, n$ ), conservation of mass and energy laws determine the rates of change in density  $\rho_i$  and specific internal energy  $e_i$ , which can be used to determine cell pressure  $p_i$  and temperature  $T_i$ . At each cell boundary ( $i = 1/2, 3/2, \dots, n+1/2$ ), conservation of momentum determines the mass flow rate  $\dot{m}$  crossing each cell boundary, and energy flow rate  $\dot{E}$  can be derived using upstream cell information. The staggered grid approach was chosen over a collocated method such as the Lax-Wendroff method to improve stability and simplify Simulink block communication at the boundaries. The Lax-Wendroff method has been known to be numerically unstable during valve opening and closing events due to the abrupt changes in pressure and temperatures [80], [81].



**Figure 3.2: Staggered grid discretization**

The conservation of mass equation shown in Eq. (3.23) can be converted from the differential form by substituting  $\Delta x$  for  $\partial x$  (finite difference form) and using  $\dot{m} = \rho UA$ . The rate change in cell density becomes

$$\frac{d\rho_i}{dt} = \frac{1}{A_i \Delta x} [\dot{m}_{i-1/2} - \dot{m}_{i+1/2}]. \quad (3.31)$$

Likewise, the rate of change of each cell species density can be derived from Eq. (3.24) as

$$\frac{d}{dt}(\rho_i \mathbf{y}_i) = \frac{1}{A_i \Delta x} [\dot{m}_{i-1/2} \mathbf{y}_{i-1/2} - \dot{m}_{i+1/2} \mathbf{y}_{i+1/2}]. \quad (3.32)$$

Because species mass fractions are not explicitly defined at the cell boundaries, the values must be taken from the upwind conditions according to

$$\mathbf{y}_{i+1/2} = \begin{cases} \mathbf{y}_i & \text{if } \dot{m}_{i+1/2} \geq 0 \\ \mathbf{y}_{i+1} & \text{if } \dot{m}_{i+1/2} < 0, \end{cases} \quad (3.33)$$

which means that flow direction dictates how gas species contained in a cell are transported to neighboring cells.

Following the same procedure as the conservation of mass, the rate of change of energy at each cell center can be derived from Eq. (3.30). In the finite difference form, conservation of energy becomes

$$\frac{d}{dt} \left[ \rho_i \left( e_i + \frac{U_i^2}{2} \right) \right] = \frac{1}{A_i \Delta x} \left[ \dot{E}_{i-1/2} - \dot{E}_{i+1/2} + \dot{q}_{w,i} A_{surf,i} \right] \quad (3.34)$$

where  $A_{surf,i}$  is the  $i^{\text{th}}$  cell's wall surface area. Conservation of momentum directly determines the mass flow rate at each cell boundary. However, energy flow rate at the boundary, defined as

$$\dot{E}_{i+1/2} = \dot{m}_{i+1/2} \left( h_{i+1/2} + \frac{U_{i+1/2}^2}{2} \right), \quad (3.35)$$

requires boundary enthalpy and velocity. Similar to mass fractions, boundary enthalpy is upwinded from the neighboring cell such that

$$h_{i+1/2} = \begin{cases} e_i + \frac{p_i}{\rho_i} & \text{if } \dot{m}_{i+1/2} \geq 0 \\ e_{i+1} + \frac{p_{i+1}}{\rho_{i+1}} & \text{if } \dot{m}_{i+1/2} < 0 \end{cases} \quad (3.36)$$

Density is also upwinded in order to determine velocity at the boundary; therefore, boundary velocity becomes

$$U_{i+1/2} = \begin{cases} \frac{\dot{m}_{i+1/2}}{A_{i+1/2} \rho_i} & \dot{m}_{i+1/2} \geq 0 \\ \frac{\dot{m}_{i+1/2}}{A_{i+1/2} \rho_{i+1}} & \dot{m}_{i+1/2} < 0 \end{cases} \quad (3.37)$$

Conservation of momentum equation directly governs the boundary mass flow rate based on adjacent cell pressures, momentums, and minor losses. For generality, all pressure losses are represented by a single pressure loss coefficient  $C_{loss}$ . The total loss coefficient defined as

$$C_{loss} = C_f + C_{bend} + C_{other} = \frac{P_{loss}}{\frac{1}{2}\rho U^2}. \quad (3.38)$$

includes friction shear losses  $C_f$ , pipe bend losses  $C_{bend}$ , and any other minor losses  $C_{other}$ .

Frequently, pipe friction shear is estimated by a Darcy friction factor  $f$ , which by definition is

$$f = \frac{P_{loss}}{\Delta x} \frac{D}{\frac{1}{2}\rho U^2} = \frac{\tau_w}{\frac{1}{2}\rho U^2}. \quad (3.39)$$

Therefore, the resulting friction loss coefficient  $C_f$  becomes

$$C_f = \frac{f\Delta x}{D}. \quad (3.40)$$

Because mass flow rates are only defined at the cell boundaries, cell momentum cannot be directly calculated. Boundary mass flow rates can be averaged to determine the cell flow rate, but better numerical stability can be achieved by equating the cell flow rate to the upstream value. Using the known boundary mass flow rates, the following relationships can be derived:

$$\dot{m}_i = \begin{cases} \dot{m}_{i-1/2} & \text{if } \frac{\dot{m}_{i+1/2} + \dot{m}_{i-1/2}}{2} \geq 0 \\ \dot{m}_{i+1/2} & \text{if } \frac{\dot{m}_{i+1/2} + \dot{m}_{i-1/2}}{2} < 0 \end{cases} \quad (3.41)$$

and

$$U_i = \frac{\dot{m}_i}{A_i \rho_i}. \quad (3.42)$$

The pressure drop across a cell due to friction and minor losses can now be determined by rearranging Eq. (3.38) using cell information. The pressure loss across the  $i^{\text{th}}$  cell  $p_{loss,i}$  becomes

$$p_{loss,i} = C_{loss,i} \frac{1}{2} \rho_i U_i |U_i| \quad (3.43)$$

where the term  $U_i|U_i|$  accounts for the flow direction and  $C_{loss,i}$  describes pressure losses between two cells. The momentum equation Eq. (3.27) can now be converted into a finite difference form by replacing differential terms with the finite length  $\Delta x$  and replacing the shear loss term with the pressure loss relationship Eq. (3.43) for neighboring cells. For a staggered grid, the  $(i+1/2)^{\text{th}}$  boundary mass flow rate is defined as

$$\frac{d\dot{m}_{i+1/2}}{dt} = \frac{1}{\Delta x} \left[ \dot{m}_i U_i - \dot{m}_{i+1} U_{i+1} + A_{i+1/2} (p_i - p_{i+1}) - \frac{1}{2} (A_i p_{loss,i} + A_{i+1} p_{loss,i+1}) \right]. \quad (3.44)$$

### 3.3.3 Numerical Integration

The conservation laws produce four ODE equations, Eq. (3.31), Eq. (3.32), Eq. Eq. (3.34), and Eq. (3.44), that can be solved numerically with an ODE solver. Starting from a specified initial condition, flow variables are updated every time step  $\Delta t$ . To ensure numerical stability with explicit integration, the step size must be selected according to flow properties and discretization length  $\Delta x$ . According to the Courant-Friedrichs-Lewy (CFL) condition, stability is related to the propagation velocity, time step size, and discretization length. Assuming a first-order accurate explicit ODE solver, the CFL condition states that the system will be stable if the following condition is met:

$$C = (|U| + a) \frac{\Delta t}{\Delta x} < 1, \quad (3.45)$$

where  $C$  is the CFL number,  $U$  is the flow velocity, and  $a$  is the speed of sound. For a higher order ODE solver, the solution can be stable with  $C > 1$ , but the relationship between stability and  $U$ ,  $a$ ,  $\Delta t$ , and  $\Delta x$  remains. The speed of sound relates to the stiffness of the gas and thus relates to the mass fractions and temperature. For an ideal gas, acoustic velocity is defined as

$$a = \sqrt{\gamma(\mathbf{y}, T) \frac{P}{\rho}} = \sqrt{\gamma(\mathbf{y}, T) RT} . \quad (3.46)$$

Compared to the engine intake, the acoustic velocity will be much higher in the exhaust ports due to the high exhaust temperatures. As a result, the  $\Delta x$  needs to be adjusted for the exhaust system, or the step time  $\Delta t$  will be significantly limited by the exhaust.

The staggered grid spatial discretization is stable and can be integrated explicitly. However, step changes in boundary pressure and temperatures caused by valve opening and closing events can create spurious oscillations. The oscillations can be dampened with an artificial diffusion, but as a result, the numerical damping will affect accuracy and require more calculations. To avoid numerical damping and improve stability, the momentum equation is integrated separately from the continuity and energy equations. Therefore, mass flow rates are updated using cell variables (density, mass fractions, pressure, and temperature) from the previous time step; then, cell variables are updated using the current mass flow rates. The semi-implicit time integration greatly reduces spurious oscillations without introducing numerical damping.

### 3.3.4 Friction Factor

The Darcy friction factor correlates bulk pipe flow to shear forces at the wall surface according to Eq. (3.39). Since the wall shear relates to viscous forces, the Reynolds number  $Re$  can be used for determining the friction factor. The Reynolds number is defined as

$$Re = \frac{\rho U D}{\mu} , \quad (3.47)$$

where  $\mu$  is the gas dynamic viscosity. For laminar flow ( $Re < 2300$ ), the Darcy friction factor can be determined by

$$f = \frac{64}{Re}, \quad (3.48)$$

For turbulent flow ( $Re > 4000$ ), the friction factor depends on pipe surface roughness. Several correlations exist for determining the friction factor for turbulent flow in circular pipes and are often derived from the Colebrook-White equation [82]. The correlations represent pipe roughness with a roughness height parameter  $\varepsilon$  and the hydraulic diameter  $D$ . Due to the implicit nature of the Colebrook-White equations, the method is not computationally efficient. Therefore, for the current engine model, the explicit Haaland correlation is used [83]:

$$f = \left[ -1.8 \log_{10} \left( \frac{6.9}{Re} + \left( \frac{\varepsilon/D}{3.7} \right)^{1.11} \right) \right]^{-2}. \quad (3.49)$$

Note that any other turbulent equation can be used to replace the Haaland equation to improve accuracy or computation efficiency.

The friction factors for laminar and turbulent flow defined in Eq. (3.48) and Eq. (3.49) assume a circular pipe, but rectangular ducts are frequently encountered as well. The hydraulic diameter of a rectangular duct with height  $l_h$  and width  $l_w$  is defined as

$$D_H = \frac{2l_h l_w}{l_h + l_w}. \quad (3.50)$$

Using  $D_H$ , the friction factor  $f$  can be calculated from Eq. (3.48) and Eq. (3.49) and corrected based on the duct aspect ratio  $\alpha^* = l_h/l_w$ , assuming  $l_w > l_h$ . For laminar flow, Shah defined the correction factor  $K_{corr}$  as [84]:

$$K_{corr} = \frac{1.5}{(1 + \alpha^*)^2 \left[ 1 - \frac{192\alpha^*}{\pi^5} \tanh\left(\frac{\pi}{2\alpha^*}\right) \right]}. \quad (3.51)$$

Although defined for laminar conditions, the correction factor  $K_{corr}$  is also used for turbulent flow.

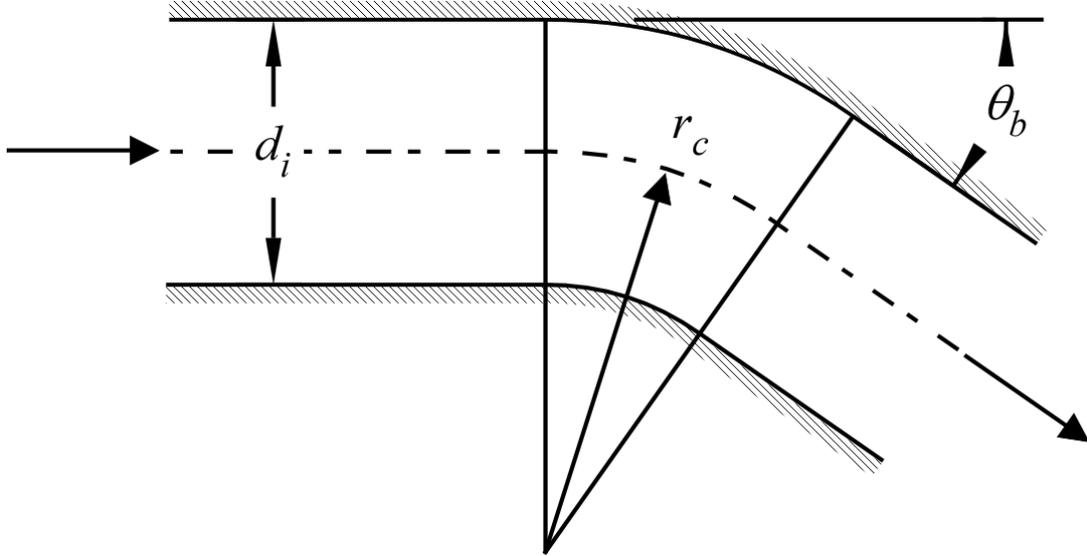
At the narrow transitional flow region ( $2300 < Re < 4000$ ), the friction factor is not defined by Eq. (3.48) or Eq. (3.49). Therefore, for the transition region, the friction factor is interpolated between the laminar  $f$  at  $Re = 2300$  according to Eq. (3.48) and the turbulent  $f$  at  $Re = 4000$  according to Eq. (3.49). Blair stated that engine flow most often resides in the turbulent region [22]. Therefore, the transitional friction factor will rarely be needed and little accuracy will be sacrificed by interpolating between the laminar and turbulent friction factors.

### 3.3.5 Pipe Bends

Bends in the engine ducts result in a pressure drop that cannot be directly modeled in 1D, but can be represented with a loss coefficient  $C_{bend}$ . In general, referring to Figure 3.3, a pipe bend can be described by a bend angle  $\theta_b$  and a centerline bend radius  $r_c$ . The loss of pressure depends on the pipe geometry and flow conditions. Miller describes the loss coefficient with the equation [85]:

$$C_{bend} = K_b^* C_{Re} C_{dev} C_{rough}, \quad (3.52)$$

where  $K_b^*$  is the bend loss coefficient at  $Re = 10^6$ ,  $C_{Re}$  is the correction factor for the actual Reynolds number, and  $C_{dev}$  is the correction factor for the outlet flow development, and  $C_{rough}$  accounts for the pipe roughness. The outlet pipe is assumed to be long enough to allow flow to fully develop; therefore,  $C_{dev} = 1$ . The remaining factors are discussed below.



**Figure 3.3: Circular pipe bend parameters**

The base loss coefficient  $K_b^*$  is taken at a fixed Reynolds number for a smooth pipe, thus only varies with the bend angle  $\theta_b$  and radius ratio  $r_c/d_i$ . For  $Re = 10^6$ , variation of the coefficient  $K_b^*$  is shown in Figure 3.4. Values shown in Figure 3.4 are tabulated in a 2D lookup table and calculated prior to simulation in the current engine model. To consider roughness, the correction factor  $C_{rough}$  is defined as

$$C_{rough} = \frac{f_{rough}}{f_{smooth}}, \quad (3.53)$$

where  $f_{rough}$  is the friction factor for a rough pipe ( $\varepsilon > 0$ ) and  $f_{smooth}$  is the smooth friction factor ( $\varepsilon = 0$ ). Both friction factors are calculated by Eq. (3.49) for turbulent flow, and because the laminar friction factor does not depend on roughness parameter  $\varepsilon$ ,  $C_{rough} = 1$  for laminar flow.

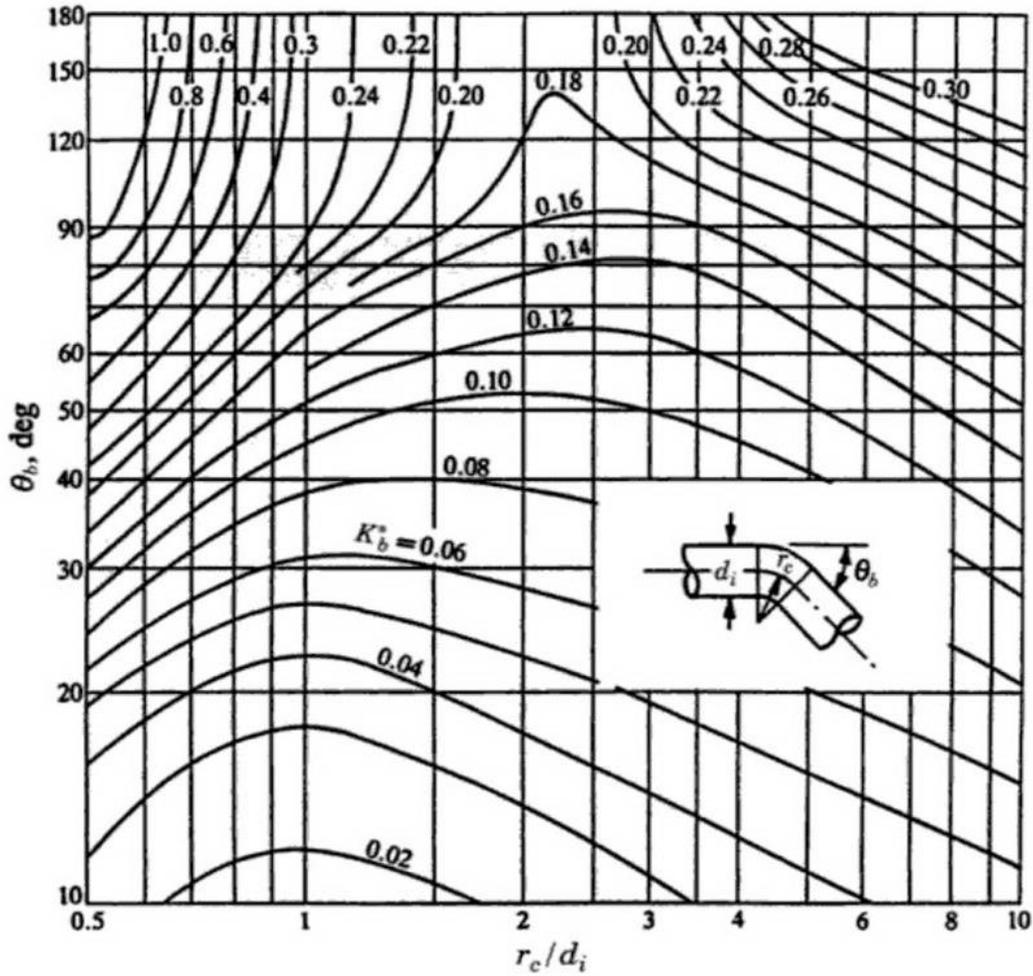


Figure 3.4: Round pipe bend loss coefficient  $K_b^*$  at  $Re = 10^6$  (From reference [85])

The Reynolds correction factor  $C_{Re}$  defined by Miller is plotted in Figure 3.5 [85]. By fitting the data to a power law,  $C_{Re}$  can be quickly calculated during the simulation. The fitting function is determined as

$$C_{Re} = \begin{cases} 22.9824Re^{-0.2547} & r_c/d_i \leq 1 \\ 13.2494Re^{-0.1949} & r_c/d_i = 1.5 \\ 10.2294Re^{-0.1668} & r_c/d_i \geq 2 \end{cases} \quad (3.54)$$

Now note that Figure 3.5 does not show the range  $Re < 10^4$  and  $C_{Re}$  remains constant at high  $Re$  with  $r_c/d_i < 2$ . To simulate cases including  $Re < 10^4$  using Figure 3.5,  $C_{Re}$  is calculated as follows:

1. If  $Re \leq 10^4$ , set  $C_{Re} = 2.2$ .
2. If  $Re > 10^4$ , calculate  $C_{Re}$  using Eq. (3.54). Interpolate value for  $1 < r_c/d_i < 2$ .
3. If  $C_{Re} < 1$ ,  $r_c/d_i < 2$ , and  $Re > 10^4$ , set  $C_{Re} = 1$ .

Although the method can be used in most geometries, Miller found that the loss coefficient depends strongly on  $Re$  when  $r_c/d_i < 1$  [85]. Therefore,  $C_{Re}$  is further constrained as follows:

4. For  $0.7 < r_c/d_i < 1$  or  $K_b^* < 0.4$ , use described method with  $r_c/d_i = 1$ .
5. Otherwise, calculate  $C_{Re}$  from the equation

$$C_{Re} = \frac{K_b^*}{K_b^* - 0.2C_{Re}' + 0.2}, \quad (3.55)$$

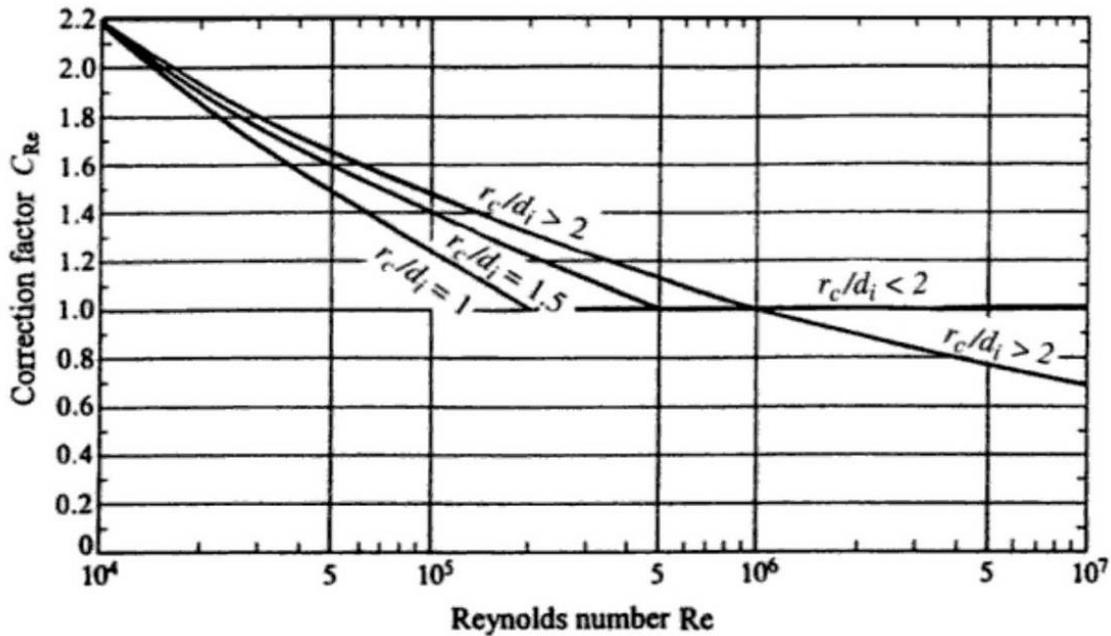


Figure 3.5: Bend loss coefficient Reynolds correction factor  $C_{Re}$  (From [85])

### 3.3.6 Heat Transfer

Conduction, convection, and radiative heat transfer contribute to the overall heat transfer. For the engine simulation, radiative bodies are assumed not to be present in the ducts, and conduction between cells can be neglected. However, convection heat transfer plays an important role in accurately simulating engine flow characteristics. Heat transferred to the air entering the cylinder decreases the air density and therefore the amount of oxygen available for combustion. On the exhaust side, a significant amount of energy is transferred from the exhaust gasses to the exhaust valves, runners, and manifold. The significant energy transfer lowers the flow temperature and therefore the acoustic wave velocity.

Assuming that the heat transfer from the wall to the gas is positive, the forced-convection heat transfer relates to the wall and gas temperatures according to the relationship

$$\dot{q}_w = h_c(T_w - T) \quad (3.56)$$

where  $h_c$  is the convection heat transfer coefficient. The heat transfer coefficient can be determined from a correlation fit to the dimensionless Reynolds ( $Re$ ), Nusselt ( $Nu$ ) and Prandtl ( $Pr$ ) numbers.  $Nu$  and  $Pr$  are defined as

$$Nu = \frac{h_c D}{k} \quad (3.57)$$

and

$$Pr = \frac{c_p \mu}{k} \quad (3.58)$$

where  $c_p$  and  $k$  are the gas specific heat and thermal conductivity evaluated at the mean gas temperature, respectively. For laminar flow ( $Re < 2300$ ), the Nussult number is constant for circular pipes [86]:

$$Nu = 3.66 \quad (3.59)$$

However, for rectangular ducts,  $Nu$  varies with aspect ratio  $\alpha^* = a/b$ , which can be approximated by the correlation [86]:

$$Nu = 7.541 \left( 1 - 2.610\alpha^* - 4.970\alpha^{*2} - 5.119\alpha^{*3} + 2.702\alpha^{*4} - 0.548\alpha^{*5} \right), \quad (3.60)$$

For turbulent internal flow ( $Re > 4000$ ), several correlations with varying accuracy and complexity exist. To maintain computational efficiency, the Colburn analogy is used [87], [88]:

$$Nu = \frac{f}{8} Re Pr^{1/3} \quad (3.61)$$

Similar to the friction factor, the heat transfer coefficient can be interpolated between the laminar and turbulent solutions for the transitional region ( $2300 < Re < 4000$ ).

### 3.4 Pressure Wave Motion

The mass and energy flow rate between neighboring cells can be determined with the momentum equation, Eq. (3.27), and cell relationships. However, boundary conditions and flow restrictions require another modeling approach. Small amplitude pressure expansions or contractions, known as acoustic pressure waves, travel in the 1D engine duct. Because the acoustic waves can travel in either direction (left to right and right to left), the combination of the two waves dictates flow characteristics. The conservation laws derived previously capture the superposition effect of pressure wave propagation. For an abrupt change in flow area or at a flow boundary, however, mass flow must be determined from the incoming wave amplitude. The incoming boundary pressure wave amplitude can be extracted from cell states. Based on the incoming wave, boundary conditions, and geometry, the reflected wave can be derived from

conservation laws. The incoming and reflected acoustic waves then dictate the boundary mass flow rate.

Riemann invariants have frequently been used to model flow in engine ducts [16]. To derive the Riemann invariants for 1D compressible flow, Euler equations (continuity, energy, and momentum) are converted into a non-conservative form in terms of primitive variables  $\rho$ ,  $U$ , and  $p$ . Assuming no source terms and an ideal gas, the Euler equations can be presented in the form

$$\frac{d}{dt} \begin{bmatrix} \rho \\ U \\ p \end{bmatrix} + \begin{bmatrix} U & \rho & 0 \\ 0 & U & 1/\rho \\ 0 & \gamma p & U \end{bmatrix} \frac{d}{dx} \begin{bmatrix} \rho \\ U \\ p \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \quad (3.62)$$

Using the eigenvectors of the coefficient matrix, Eq. (3.62) can be transformed into the form

$$\frac{d}{dt} \begin{bmatrix} RI_1 \\ RI_2 \\ RI_3 \end{bmatrix} + \begin{bmatrix} U - a & 0 & 0 \\ 0 & U & 0 \\ 0 & 0 & U + a \end{bmatrix} \frac{d}{dx} \begin{bmatrix} RI_1 \\ RI_2 \\ RI_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \quad (3.63)$$

where  $RI_1$ ,  $RI_2$ , and  $RI_3$  are the Riemann invariants. By defining entropy as  $s = p \rho^{-\gamma}$ , the Riemann invariants are

$$\begin{bmatrix} RI_1 \\ RI_2 \\ RI_3 \end{bmatrix} = \begin{bmatrix} U - \frac{2a}{\gamma - 1} \\ s \\ U + \frac{2a}{\gamma - 1} \end{bmatrix}. \quad (3.64)$$

As an alternative to using Riemann invariants, Blair developed a method based on acoustic wave propagation [21]. First predicted by Earnshaw, the amplitude of an acoustic pressure wave relates a fluid's particle velocity  $U$  [89]. Starting at a reference velocity  $U_0$ , pressure  $p_0$ , and acoustic velocity  $a_0$ , Earnshaw showed that

$$U - U_0 = \frac{2}{\gamma - 1} a_0 \left[ \left( \frac{p}{p_0} \right)^{\frac{\gamma-1}{2\gamma}} - 1 \right]. \quad (3.65)$$

The acoustic velocity is governed by a fluid's stiffness and density, and for an ideal gas, the reference acoustic velocity can be defined as

$$a_0 = \sqrt{\gamma R T_0}, \quad (3.66)$$

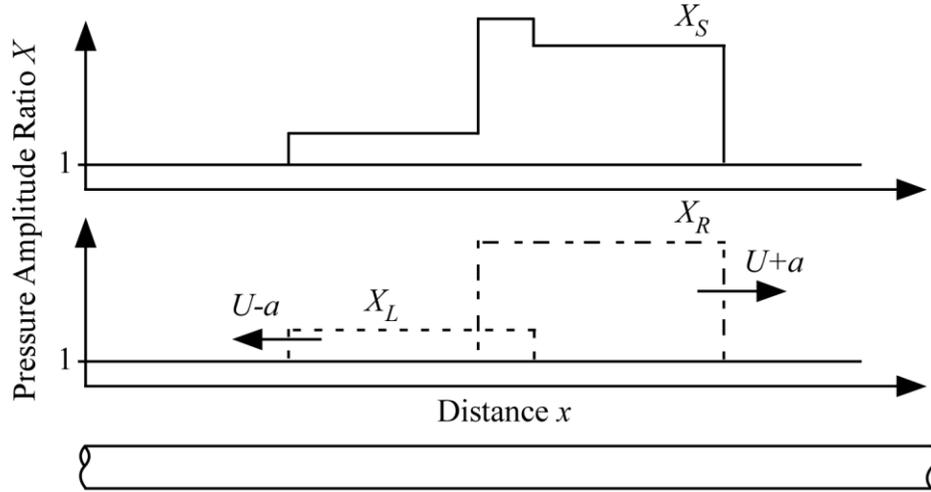
where  $R$  is the ideal gas constant and  $T_0$  is the reference temperature. To represent the acoustic wave Eq. (3.65), Blair defined a pressure amplitude ratio  $X$  as

$$X = \left( \frac{p}{p_0} \right)^{\frac{\gamma-1}{2\gamma}}. \quad (3.67)$$

By assuming  $U_0 = 0$  and substituting Eq. (3.67) into Eq. (3.65), Earnshaw's theory becomes

$$U = \frac{2}{\gamma - 1} a_0 [X - 1]. \quad (3.68)$$

Note that the reference conditions  $T_0$ ,  $p_0$ , and  $a_0$  are the same as the stagnation conditions if  $U_0 = 0$ . To be consistent with Blair's examination, these parameters will be referred to as the reference conditions. The pressure wave propagates at the acoustic velocity relative to the gas particle velocity. Therefore, in reference to a fixed coordinate, the propagation velocity is the sum of the acoustic and particle velocities.



**Figure 3.6: Acoustic waves traveling in pipe**

Shown in Figure 3.6, two pressure amplitude ratios are present in the 1D pipe model: a leftward  $X_L$  and rightward  $X_R$  traveling pressure amplitude ratio. The waves propagate in opposite directions according to the acoustic and particle velocities. By superimposing the two waves, a superposition pressure amplitude ratio  $X_S$  relates to the flow states. According to Eq. (3.67), the superposition pressure  $p_S$  is defined as

$$p_S = p_0 X_S^{\frac{2\gamma}{\gamma-1}} \quad (3.69)$$

The superposition acoustic velocity  $a_S$  and temperature  $T_S$  can be derived as

$$T_S = T_0 X_S^2 \quad (3.70)$$

and

$$a_S = a_0 X_S, \quad (3.71)$$

assuming that the state changes from the reference conditions  $p_0$  and  $T_0$  the superposition conditions  $p_S$  and  $T_S$  to be isentropic. Using Earnshaw's theory in the form of Eq. (3.68) and the relationship in Eq. (3.71), it can be shown that

$$X_S = X_R + X_L - 1 \quad (3.72)$$

$$U_s = \frac{2}{\gamma-1} a_0 [X_R - X_L]. \quad (3.73)$$

For the model, the reference pressure  $p_0$  will be assumed constant and equal to the ambient absolute pressure. The reference temperature  $T_0$  can fluctuate based on non-isentropic flow behavior.

Variables  $X_L$ ,  $T_0$ , and  $X_R$  are analogous to the Riemann invariants defined in Eq. (3.64): pressure amplitude ratios  $X_L$  and  $X_R$  travel in opposite directions at speeds  $U - a$  and  $U + a$ ; Riemann invariants  $RI_1$  and  $RI_3$  are constant along the characteristics  $dx/dt = U - a$  and  $dx/dt = U + a$ ; temperature  $T_0$ , which remains constant for isentropic conditions, travels with the particle velocity; and the Riemann invariant  $RI_2$  (entropy) is constant along the characteristic  $dx/dt = U$ . Using the acoustic and particle velocity relationships in Eq. (3.71) and Eq. (3.73), the pressure amplitude ratios  $X_R$  and  $X_L$  can be defined in terms of the Riemann invariants, such that

$$X_R = -\frac{1-\gamma}{4a_0} RI_3 - \frac{1}{2} \quad (3.74)$$

and

$$X_L = \frac{1-\gamma}{4a_0} RI_1 - \frac{1}{2}. \quad (3.75)$$

Either method could be implemented for the 1D boundaries, but since the introduction of the Blair's method, numerous references have outlined boundary condition, engine valve, and flow junction models [22]–[24]. Additionally, the model has been thoroughly validated [22]–[25]. Therefore, Blair's method is implemented at the boundaries.

### 3.5 Flow Restrictions and Adjoined Pipes

The conservation of momentum equation for 1D flow Eq. (3.44) does not consider cell boundary flow restrictions or area discontinuities. A flow restriction can be used to model an

orifice, throttle valve, or any other 1D restriction not described by a loss coefficient. Flow area discontinuities can be formed by adjoining two pipes that do not have the same cross section. Because of the abrupt change in area at a cell boundary due to restriction or adjoining pipe, pressure waves entering either side of the boundary get reflected. The amplitudes of the reflected waves are dictated by the change in area and conservation laws, and in turn, govern the cell boundary mass flow rate.

### 3.5.1 Model Setup and Conservation Laws

With the staggered grid approach, two pipes collinearly joined form a common cell boundary that may include a pipe area discontinuity and/or a flow restriction as depicted in Figure 3.7. Left and right cell information is determined by the 1D flow model described earlier. Therefore, mass and energy flow rates,  $\dot{m}_1$ ,  $\dot{m}_2$ ,  $\dot{E}_1$ , and  $\dot{E}_2$ , must be determined based on the connection geometry and adjoining cell information. By definition, the cell boundary is not a volume, and the rate change in density and energy between stations 1 and 2 become zero. According to the conservation of mass, the mass flow rates across the boundary can be equated:

$$\dot{m}_1 = \dot{m}_2 = \dot{m}_t \quad (3.76)$$

where  $\dot{m}_t$  is the mass flow rate through the restriction throat. The flow must contract to pass through the throat area  $A_t$ , and for real gas flow, a discharge coefficient  $C_D$  is typically introduced to model the contraction and velocity losses. With the discharge coefficient, the effective throat area  $A_{eff}$  can be defined as

$$A_{eff} = C_D A_t \quad (3.77)$$

From Eq. (3.76) and Eq. (3.77), the following can be concluded:

$$\dot{m}_1 - \dot{m}_2 = \rho_1 U_1 A_1 - \rho_2 U_2 A_2 = 0 \quad (3.78)$$

and

$$\dot{m}_1 - \dot{m}_t = \rho_1 U_1 A_1 - \rho_t U_t C_D A_t = 0 \quad (3.79)$$

Similarly, the conservation of energy gives

$$\left( h_1 + \frac{U_1^2}{2} \right) - \left( h_2 + \frac{U_2^2}{2} \right) = 0 \quad (3.80)$$

and

$$\left( h_1 + \frac{U_1^2}{2} \right) - \left( h_t + \frac{U_t^2}{2} \right) = 0 \quad (3.81)$$

Parameters at Station 1, Station 2, and throat are not explicitly available, and must be calculated from cell information using acoustic wave theory.

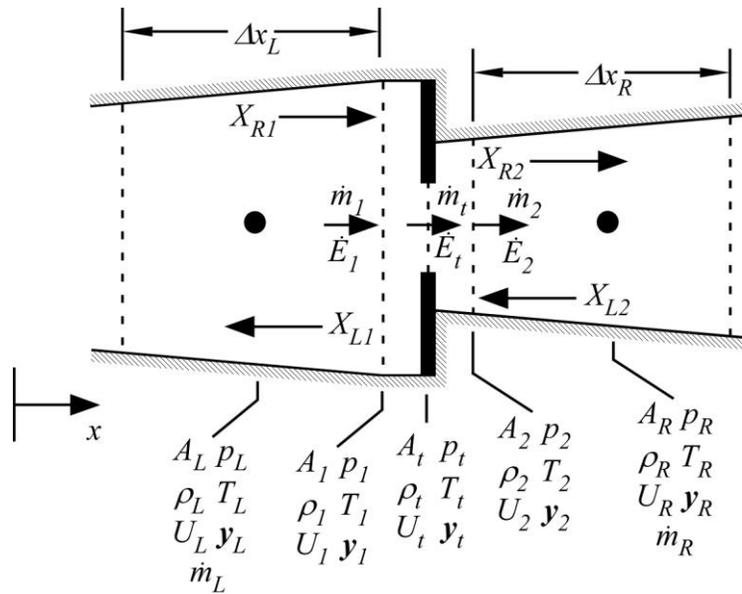


Figure 3.7: Schematic of adjoining pipes with restrictive orifice

### 3.5.2 Boundary Parameter Relationships

Referring to Figure 3.7, thermodynamic state variables and velocities at Station 1, throat, and Station 2 do not hold a direct relationship to the left and right cell states but relate to incoming acoustic waves. States defined at the cell centers can provide the incoming pressure

waves  $X_{RI}$  and  $X_{L2}$  and cell reference temperatures  $T_{0L}$  and  $T_{0R}$ . From Eq. (3.69), the superposed pressure amplitude ratios for the left  $X_{SL}$  and right  $X_{SR}$  cells are defined as

$$X_{SL} = \left( \frac{P_L}{P_0} \right)^{\frac{\gamma_L - 1}{2\gamma_L}} \quad (3.82a)$$

and

$$X_{SR} = \left( \frac{P_R}{P_0} \right)^{\frac{\gamma_R - 1}{2\gamma_R}}, \quad (3.82b)$$

where specific heat ratios  $\gamma_L$  and  $\gamma_R$  are calculated by each respective cell's temperature and mass fractions. From Eq. (3.70), the reference temperature for the left  $T_{0L}$  and right  $T_{0R}$  cells are defined as

$$T_{0L} = \frac{T_L}{X_{SL}^2} \quad (3.83a)$$

and

$$T_{0R} = \frac{T_R}{X_{SR}^2}. \quad (3.83b)$$

Now, the superposed pressure amplitude ratios defined in Eq. (3.82) can to be split into opposite traveling acoustic waves. Referring to Figure 3.7, the incoming wave  $X_{RI}$  can be determined by extrapolating the rightward traveling wave from the left cell center to Station 1 using the velocity relationship defined in Eq. (3.73); hence,

$$X_{RI} = \frac{1}{2} \left( \frac{U_L(\gamma_L - 1)}{2\sqrt{\gamma_L T_{0L} R_L}} + X_{SL} + 1 \right). \quad (3.84)$$

Similarly, the leftward traveling wave  $X_{L2}$  can be determined by extrapolating the right cell wave to Station 2, giving

$$X_{L2} = \frac{1}{2} \left( -\frac{U_R(\gamma_R - 1)}{2\sqrt{\gamma_R T_{0R} R_R}} + X_{SR} + 1 \right). \quad (3.85)$$

With the incoming pressure waves known, the reflected waves  $X_{L1}$  and  $X_{R2}$  and reference temperatures are determined by conservation laws and flow characteristics. First, each state and velocity must be expressed in a convenient form, and to express boundary states, thermodynamic properties must be evaluated at station temperatures and mass fractions. The temperatures at the boundary stations can be expressed as

$$T_1 = T_{01}(X_{R1} + X_{L1} - 1)^2, \quad (3.86a)$$

$$T_2 = T_{02}(X_{R2} + X_{L2} - 1)^2 \quad (3.86b)$$

and 
$$T_t = T_{0t}(X_t)^2, \quad (3.86c)$$

where  $X_t$  is the superposed pressure amplitude ratio at the throat. The mass fractions at each station can be determined based on flow direction: for flow from left to right, evaluate properties with left cell mass fractions, and for flow from right to left, use right cell mass fractions. However, upwinding mass fractions creates a discontinuity when switching flow directions. Because unknowns must be solved iteratively, the discontinuity creates convergence issues and jumps in the solution. Therefore, properties are evaluated at a mean mass fraction  $y_a$  defined as

$$y_a = \frac{y_L + y_R}{2}. \quad (3.87)$$

Similar issues arise when evaluating the boundary specific heat ratio  $\gamma_a$ . Therefore,  $\gamma_a$  is evaluated at  $y_a$  and the mean temperature  $T_a$  defined as

$$T_a = \frac{T_L + T_R}{2}. \quad (3.88)$$

How properties are evaluated will become more apparent when discussing the overall solution method.

Using the temperature and property information, defining the remaining boundary states becomes straightforward. According to Eq. (3.69), the pressure relationships can be derived:

$$p_1 = p_0 (X_{R1} + X_{L1} - 1)^{\frac{2\gamma_a}{\gamma_a - 1}}, \quad (3.89a)$$

$$p_2 = p_0 (X_{R2} + X_{L2} - 1)^{\frac{2\gamma_a}{\gamma_a - 1}}, \quad (3.89b)$$

and

$$p_t = p_0 X_t^{\frac{2\gamma_a}{\gamma_a - 1}}. \quad (3.89c)$$

Density at each station can be derived from Eq. (3.86), Eq. (3.89), and the ideal gas law, giving

$$\rho_1 = \frac{p_0}{R_a T_{01}} (X_{R1} + X_{L1} - 1)^{\frac{2}{\gamma_a - 1}}, \quad (3.90a)$$

$$\rho_2 = \frac{p_0}{R_a T_{02}} (X_{R2} + X_{L2} - 1)^{\frac{2}{\gamma_a - 1}}, \quad (3.90b)$$

and

$$\rho_t = \frac{p_0}{R_a T_{0t}} (X_t)^{\frac{2}{\gamma_a - 1}}, \quad (3.90c)$$

where  $R_a$  is the ideal gas constant evaluated with  $y_a$ . Velocities at stations 1 and 2 vary based on the incoming and reflected pressure amplitude ratios. According to Eq. (3.73), velocities at stations 1 and 2 can be calculated as

$$U_1 = \frac{2}{\gamma_a - 1} \sqrt{\gamma_a T_{01} R_a} [X_{R1} - X_{L1}] \quad (3.91a)$$

and

$$U_2 = \frac{2}{\gamma_a - 1} \sqrt{\gamma_a T_{02} R_a} [X_{R2} - X_{L2}] \quad (3.91b)$$

The throat variables are expressed in terms of the superposed pressure amplitude ratio  $X_t$  because neither the rightward nor leftward traveling waves are known. Therefore,  $U_t$  must be solved iteratively and does not require a relationship similar to Eq. (3.91).

### 3.5.3 Solution Overview

The boundary state variables defined in terms of reference temperatures and pressure amplitude ratios, when substituted into conservation laws, form a set of constraint equations. The equations relate adjoining pipe cell reference temperatures and incoming waves to the boundary reference temperatures and reflected waves. Examining the conservation laws and the relationships presented in the previous section,  $X_{L1}$ ,  $X_{R2}$ ,  $X_t$ ,  $T_{01}$ ,  $T_{0t}$ ,  $T_{02}$ , and  $U_t$  are unknown. Therefore, solving for the unknown variables require seven constraints. Conservation of mass and energy provide four constraints, while the remaining constraints are derived from entropy and momentum relationships. For a 1D flow model, Benson suggested modeling flow through a sudden change in area as an isentropic process [16]. Based on experience, Blair claims the assumption to be accurate for only certain situations [22]. Therefore, to ensure accuracy for all configurations, the more complete non-isentropic model proposed by Blair is used.

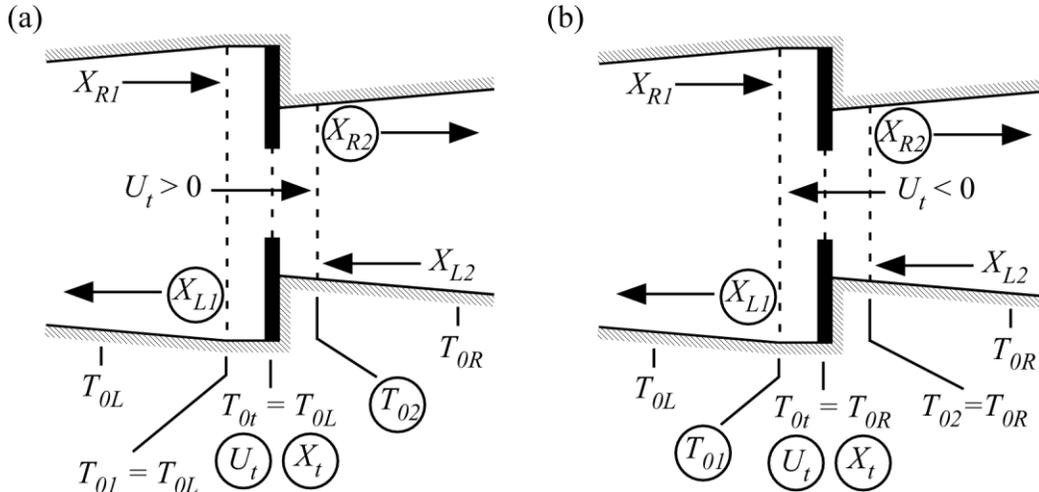
Flow must contract in order to pass through the junction throat. Gas contraction does not create flow separation or significant turbulence, and therefore, flow contraction is assumed isentropic. Likewise, flow from the left cell to Station 1 is assumed isentropic for forward flow ( $U_t > 0$ ), and flow from the right cell to Station 2 is assumed isentropic for reverse flow ( $U_t < 0$ ). By definition, the reference temperature remains constant for an isentropic process, thus providing two constraints for the reference temperatures:

$$\begin{array}{ll} \text{if } U_t \geq 0 & T_{0t} = T_{01} = T_{0L} \\ \text{else} & T_{0t} = T_{02} = T_{0R} \end{array} \quad (3.92)$$

Flow exiting the throat expands to the downstream cross section area, giving rise to particle flow separation and turbulent vortices. The flow separation implies a non-isentropic process, and another relationship must be used for calculating the downstream reference temperature. Using conservation of momentum, flow information at the throat and Station 2 can be related for forward flow, and throat and Station 1 can be related for reverse flow. The downstream momentum equation is given by

$$\begin{aligned} \text{if } U_t \geq 0 & \quad A_2(p_t - p_2) + \dot{m}_1(U_t - U_2) = 0 \\ \text{else} & \quad A_1(p_t - p_1) + \dot{m}_1(U_t - U_1) = 0. \end{aligned} \quad (3.93)$$

The relationships in Eq. (3.92) provide direct solutions to two of the unknown variables, reducing the number of unknown variables to five. Referring to Figure 3.8, the circled unknown variables must be obtained from the conservation mass equations, Eq. (3.78) and Eq. (3.79); conservation of energy equations, Eq. (3.80) and Eq. (3.81); and the momentum equation, Eq. (3.93). After substituting velocities, densities, pressures, and temperatures expressed in terms of acoustic waves, the five nonlinear constraint equations cannot be solved analytically but must be solved iteratively. Based on experience, Blair found the Newton-Raphson method to be stable, accurate, and fast for solving the equations [22]. At the start of simulation, unknown variables are approximated based on cell initial conditions, and for subsequent time steps, the initial iterative guesses are taken from the previous time step.



**Figure 3.8: Forward (a) and reverse (b) parameter constraints for adjoined pipe boundary (unknowns circled)**

The particle velocity at the throat  $U_t$  cannot exceed the local acoustic velocity. However, the flow analysis discussed previously does not restrict  $U_t$ , and depending on conditions,  $U_t$  can be found to reach or exceed the throat acoustic velocity. Therefore, a new relationship must be introduced for the velocity limit known as choked or critical flow. For choked flow,  $U_t$  can be equated to the local acoustic velocity  $a_t$ :

$$U_t = a_t \quad (3.94)$$

Velocity at other stations is assumed subsonic. The choked flow constraint must replace one of previously defined equations when solving for the five unknowns. The isentropic contraction assumption is still valid; mass and energy must be conserved between stations 1 and 2; and the momentum equation in Eq. (3.93) must be retained to account for pressure recovery. Therefore, the intermediate energy equation, Eq. (3.81), is chosen to be replaced. For choked flow, the unknown variables shown in Figure 3.8 are solved from Eqs. (3.78), (3.79), (3.80), (3.93), and (3.94).

According to the model equations, four possible situations can potentially be encountered: subsonic forward, subsonic reverse, choked forward, and choked reverse flows. Each situation requires five equations to solve for five unknowns. Before finding the unknown variables, the equations must be expressed in terms of the incoming pressure amplitude ratios, cell reference temperatures, thermodynamic properties, flow areas, throat discharge coefficient, and unknown variables. Substituting density and velocity in terms of acoustic variables, Eq. (3.90) and Eq. (3.91), into the conservation of mass equations, Eq. (3.78) and Eq. (3.79), produces the following:

$$\begin{aligned} & \frac{A_1}{T_{01}} \sqrt{\gamma_a T_{01} R_a} (X_{R1} - X_{L1}) (X_{R1} + X_{L1} - 1)^{\frac{2}{\gamma_a - 1}} - \\ & \frac{A_2}{T_{02}} \sqrt{\gamma_a T_{02} R_a} (X_{R2} - X_{L2}) (X_{R2} + X_{L2} - 1)^{\frac{2}{\gamma_a - 1}} = 0 \end{aligned} \quad (3.95)$$

and 
$$\frac{2A_1}{T_{01}(\gamma_a - 1)} \sqrt{\gamma_a T_{01} R_a} (X_{R1} - X_{L1}) (X_{R1} + X_{L1} - 1)^{\frac{2}{\gamma_a - 1}} - \frac{C_D A_t}{T_{0t}} U_t (X_t)^{\frac{2}{\gamma_a - 1}} = 0. \quad (3.96)$$

For conservation of energy, enthalpy must be calculated with the adjoined pipe mass fractions  $y_a$  and the local temperature. The conservation of energy equations in Eqs. (3.80) and (3.81) can be expressed as

$$\begin{aligned} & h_1(\mathbf{y}_a, T_1 = T_{01} (X_{R1} + X_{L1} - 1)^2) - h_2(\mathbf{y}_a, T_2 = T_{02} (X_{R2} + X_{L2} - 1)^2) + \\ & \left[ \frac{2\gamma_a R_a}{(\gamma_a - 1)^2} (T_{01} (X_{R1} - X_{L1})^2 - T_{02} (X_{R2} - X_{L2})^2) \right] = 0 \end{aligned} \quad (3.97)$$

and 
$$\begin{aligned} & h_1(\mathbf{y}_a, T_1 = T_{01} (X_{R1} + X_{L1} - 1)^2) - h_t(\mathbf{y}_a, T_t = T_{0t} (X_t)^2) + \\ & \left[ \frac{2\gamma_a T_{01} R_a}{(\gamma_a - 1)^2} (X_{R1} - X_{L1})^2 - \frac{U_t^2}{2} \right] = 0 \end{aligned} \quad (3.98)$$

using the acoustic relationships for temperature and velocity. The conservation of momentum equation defined in Eq. (3.93) can be split into two constraints depending on flow direction. For forward flow, Eq. (3.93) gives

$$A_2 \left( X_t \frac{2\gamma_a}{\gamma_a - 1} - (X_{R2} + X_{L2} - 1) \frac{2\gamma_a}{\gamma_a - 1} \right) + \frac{2A_1 \sqrt{\gamma_a T_{01} R_a}}{R_a T_{01} (\gamma_a - 1)} (X_{R1} - X_{L1}) (X_{R1} + X_{L1} - 1) \frac{2}{\gamma_a - 1} \times \left[ U_t - \frac{2}{\gamma_a - 1} \sqrt{\gamma_a T_{02} R_a} (X_{R2} - X_{L2}) \right] = 0 \quad (3.99)$$

and for reverse flow, Eq. (3.93) gives

$$A_1 \left( X_t \frac{2\gamma_a}{\gamma_a - 1} - (X_{R1} + X_{L1} - 1) \frac{2\gamma_a}{\gamma_a - 1} \right) + \frac{2A_1 \sqrt{\gamma_a T_{01} R_a}}{R_a T_{01} (\gamma_a - 1)} (X_{R1} - X_{L1}) (X_{R1} + X_{L1} - 1) \frac{2}{\gamma_a - 1} \times \left[ U_t - \frac{2}{\gamma_a - 1} \sqrt{\gamma_a T_{01} R_a} (X_{R1} - X_{L1}) \right] = 0 \quad (3.100)$$

For choked flow, the throat acoustic velocity can be expressed in terms of throat reference temperature  $T_{0t}$  and pressure amplitude ratio  $X_t$ , providing the relationship

$$a_t = X_t \sqrt{\gamma_a T_{0t} R_a} \quad (3.101)$$

Finally, the limit for choked flow, Eq. (3.94), gives

$$U_t - X_t \sqrt{\gamma_a T_{0t} R_a} = 0 \quad (3.102)$$

With the isentropic relationships given in Eq. (3.92), the unknown variables can be solved iteratively with Eq. (3.95) through Eq. (3.102) according to the flow condition (subsonic forward, subsonic reverse, choked forward, and choked reverse). For each flow situation, the velocity range, unknown variables, directly applied constraints, and iteration equation numbers are summarized in Table 3.1. Note that the throat velocity  $U_t$  dictates the solution method but is also an unknown parameter. The equations for a given flow condition are continuously

differentiable and can be solved using a Newton-type solver. However, when considering the solution as a whole, equations are not continuously differentiable at  $U_t = 0$ ,  $U_t = a_t$ , or  $U_t = -a_t$ . As a result, each flow condition is evaluated separately. The previously determined  $U_t$  dictates the solution method, and after iterating, the next solution method is determined by the updated  $U_t$ . Therefore, the overall solver can alternate between subsonic forward, subsonic reverse, choked forward, and choked reverse equations without encountering a derivative discontinuity.

**Table 3.1: Solution summary for adjoined pipes**

	Subsonic Forward	Subsonic Reverse	Choked Forward	Choked Reverse
Velocity Range	$0 \leq U_t < a_t$	$-a_t < U_t < 0$	$U_t = a_t$	$U_t = -a_t$
Unknowns	$X_{L1}, X_{R2}, U_t, X_t,$ and $T_{02}$	$X_{L1}, X_{R2}, U_t, X_t,$ and $T_{01}$	$X_{L1}, X_{R2}, U_t, X_t,$ and $T_{02}$	$X_{L1}, X_{R2}, U_t, X_t,$ and $T_{01}$
Direct Constraints	$T_{01} = T_{0L}$ $T_{0t} = T_{0L}$	$T_{02} = T_{0R}$ $T_{0t} = T_{0R}$	$T_{01} = T_{0L}$ $T_{0t} = T_{0L}$	$T_{02} = T_{0R}$ $T_{0t} = T_{0R}$
Constraint Equations	3.95, 3.96, 3.97, 3.98, and 3.99	3.95, 3.96, 3.97, 3.98, and 3.100	3.95, 3.96, 3.97, 3.99, and 3.102	3.95, 3.96, 3.97, 3.100, and 3.102

### 3.5.4 Boundary Mass and Energy Flow Rates

After calculating the unknown variables listed in Table 3.1, boundary mass and energy flow rates,  $\dot{m}_1$  and  $\dot{E}_1$ , are evaluated from the acoustic wave relationships. With  $\dot{m}_1$  and  $\dot{E}_1$  as the left and right cell boundary flow rates, the rate of change of momentum between the left and right cells is neglected, breaking the form of the staggered grid approach. To account for momentum changes, a new boundary mass flow rate  $\dot{m}_a$  is introduced for the left and right cell boundary mass flow rate. Referring to Figure 3.9, conservation of momentum can be applied between the right and left cell centers to determine the rate of change of mass flow through the boundary  $\dot{m}_a$ . Because of the discontinuity at the boundary, momentum conservation must be

applied between consecutive stations. Derived similarly to Eq. (3.44), conservation of momentum from the left cell center to Station 1 and Station 2 to the right cell center are given by

$$\frac{d\dot{m}_a}{dt} = \frac{2}{\Delta x_L} \left[ \dot{m}_L U_L - \dot{m}_1 U_1 + A_1 (p_L - p_1) - \frac{1}{2} (A_L p_{loss,L}) \right]. \quad (3.103)$$

and

$$\frac{d\dot{m}_a}{dt} = \frac{2}{\Delta x_R} \left[ \dot{m}_2 U_2 - \dot{m}_R U_R + A_2 (p_2 - p_R) - \frac{1}{2} (A_R p_{loss,R}) \right]. \quad (3.104)$$

Combining Eq. (3.103) and Eq. (3.104), the rate of change of mass flow rate through the boundary is determined by

$$\frac{d\dot{m}_a}{dt} = \frac{2}{\Delta x_L + \Delta x_R} \times \left[ \dot{m}_L U_L - \dot{m}_R U_R + \dot{m}_1 (U_2 - U_1) + A_1 (p_L - p_1) + A_2 (p_2 - p_R) - \frac{1}{2} (A_L p_{loss,L} + A_R p_{loss,R}) \right], \quad (3.105)$$

Note that it is not implied that  $\dot{m}_a = \dot{m}_1$ , but  $\dot{m}_1$ ,  $p_1$ ,  $p_2$ ,  $U_1$ , and  $U_2$  calculated from the acoustic wave relationships provide a way to determine the changes in momentum due to the area discontinuity. The boundary energy flow rate  $\dot{E}_a$  can be determined based on  $\dot{m}_a$  and using the fact that energy is conserved between stations 1 and 2. For either flow direction, the boundary energy flow rate becomes

$$\dot{E}_a = \dot{m}_a \left( h_1 + \frac{U_1^2}{2} \right), \quad (3.106)$$

where enthalpy  $h_1$  is calculated with  $y_a$  and  $T_1$ .

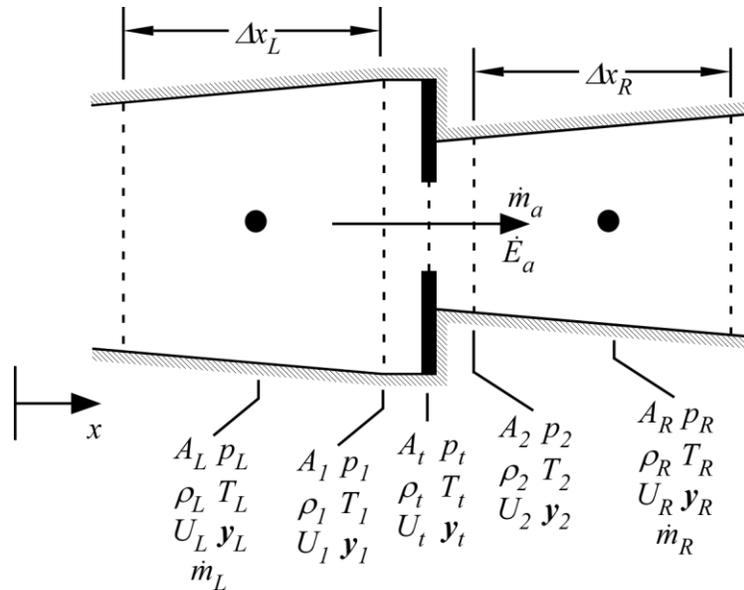


Figure 3.9: Mass and energy flow rate across adjoined pipe boundary

### 3.6 Boundary Conditions

With the 1D staggered grid, the rate of change of mass flow rate cannot be determined by the momentum equation, Eq. (3.44), at a pipe boundary—the interface between a 1D cell and a 0D volume. Therefore, mass and energy flow must be established based on external conditions and pipe boundary geometry. In some cases, the mass and energy flow rates can be explicitly defined. However, engine pipes most often connect to engine cylinders or ambient conditions, where mass and energy flow are not explicitly available. A 0D boundary (e.g. ambient boundary and engine cylinder) does not have flow velocity and is typically defined by a pressure, temperature, and flow area. The flow area can be fixed to represent the interface between a pipe and ambient conditions or vary to represent a poppet valve.

#### 3.6.1 Model Setup and Conservation Laws

Determining mass and energy flow rates at the interface between a 1D duct cell and ambient conditions or a control volume (e.g. engine cylinder, crankcase, and tank) requires

special considerations. The velocity of ambient conditions or a control volume can be best described in three dimensions. However, ambient velocity is typically assumed zero because a control volume is considered sufficiently large, which means that flow into the volume has very little influence on the volume particle velocity. As a result, ambient conditions and large volumes are modeled as a 0D. A 0D volume does not have a velocity field and can be defined by mass fractions and two thermodynamic state variables.

Referring to Figure 3.10, Station 1 represents a 0D volume, and cell parameters, denoted with subscript "C," are governed by the 1D flow model discussed previously. Flow from the 0D volume to the 1D cell is assumed positive, but depending on pipe flow convention, signs of flow rates can be switched without loss of generality. Using the cell and boundary information, flow rate  $\dot{m}_2$  and energy flow rate  $\dot{E}_2$  are determined from thermodynamic constraints. Mass must be conserved between the throat and Station 2, and by introducing a discharge coefficient  $C_D$  to represent the effective throat area, mass conservation gives

$$\dot{m}_1 - \dot{m}_2 = \rho_1 U_1 C_D A_t - \rho_2 U_2 A_2 = 0 \quad (3.107)$$

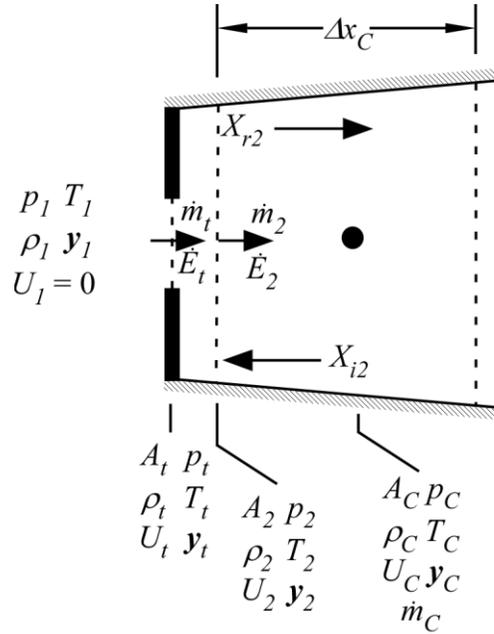
Energy must also be conserved between the throat and Station 2; thus,

$$\left( h_t + \frac{U_t^2}{2} \right) - \left( h_2 + \frac{U_2^2}{2} \right) = 0 \quad (3.108)$$

For flow into the pipe, conservation of energy states that the control volume enthalpy  $h_1$  is equivalent to the total energy per unit mass at Station 2, which can be described by

$$h_1 - \left( h_2 + \frac{U_2^2}{2} \right) = 0 \quad (3.109)$$

Throat and Station 2 parameters are not explicitly available, and must be calculated based on incoming and reflected acoustic waves. With the conservation relationships expressed in terms of pressure amplitude ratios and reference temperatures, the boundary mass and energy flow rates can be evaluated.



**Figure 3.10: Schematic of 1D pipe boundary condition**

### 3.6.2 Boundary Parameter Relationships

The thermodynamic state variables at Station 2 do not hold a direct relationship with the 0D volume or 1D cell. Instead, Station 2 parameters relate to the incident acoustic wave  $X_{i2}$  derived from cell information and the reflected wave  $X_{r2}$  shown in Figure 3.10. The incident pressure amplitude ratio  $X_{i2}$  is derived from the cell information. According to Eq. (3.69), the superposed pressure amplitude ratio  $X_{sC}$  is defined as

$$X_{sC} = \left( \frac{p_C}{p_0} \right)^{\frac{\gamma_C - 1}{2\gamma_C}}, \quad (3.110)$$

where  $\gamma_C$  is the specific heat ratio calculated with the cell's temperature and mass fractions. The incident wave relates to the cell reference temperature  $T_{0C}$  defined as

$$T_{0C} = \frac{T_C}{X_{SC}^2}. \quad (3.111)$$

The superposed pressure amplitude  $X_{SC}$  can be split into two oppositely traveling acoustic waves based on cell velocity. Referring to Figure 3.10, the incident wave  $X_{i2}$  is determined by extrapolating the leftward traveling wave from the cell center to Station 2 using the velocity relationship defined in Eq. (3.73); hence,

$$X_{i2} = \frac{1}{2} \left( -\frac{U_C(\gamma_C - 1)}{2\sqrt{\gamma_C T_{0C} R_C}} + X_{SC} + 1 \right). \quad (3.112)$$

Depending on flow direction, OD volume pressure amplitude ratio  $X_I$  and reference temperature  $T_{0I}$  are required for applying constraints. From the acoustic wave relationships, the following can be concluded:

$$X_I = \left( \frac{p_1}{p_0} \right)^{\frac{\gamma_1 - 1}{2\gamma_1}} \quad (3.113)$$

and 
$$T_{0I} = \frac{T_1}{X_I^2}, \quad (3.114)$$

where  $\gamma_I$  is the specific heat ratio calculated with the OD volume's temperature and mass fractions.

With the incoming wave and reference temperatures known, the reflected pressure amplitude ratio  $X_{r2}$  can be determined by conservation laws and flow characteristics. First, each state and velocity must be expressed in a convenient form, and to express boundary states,

thermodynamic properties must be evaluated at station temperatures and mass fractions. The temperatures at Station 2 and the throat are expressed as

$$T_2 = T_{02}(X_{r2} + X_{i2} - 1)^2 \quad (3.115a)$$

and

$$T_t = T_{0t}(X_t)^2, \quad (3.115b)$$

where  $X_t$  is the superposed pressure amplitude ratio at the throat. The mass fractions at each station can be determined based on flow direction, i.e., evaluate properties with OD volume mass fractions  $y_I$  for inflow and use cell mass fractions  $y_C$  for outflow. However, upwinding mass fractions introduces discontinuity when switching flow directions, and because unknowns must be solved iteratively, the discontinuity creates convergence issues. Therefore, properties are evaluated at a mean mass fraction  $y_b$  defined as

$$y_b = \frac{y_I + y_C}{2}. \quad (3.116)$$

Similar issues arise when evaluating the boundary specific heat ratio  $\gamma_b$ . Therefore,  $\gamma_b$  is evaluated at  $y_b$ , and the mean temperature  $T_b$  is defined as

$$T_b = \frac{T_I + T_C}{2}. \quad (3.117)$$

Before solving for unknown parameters, boundary states must be expressed in terms of pressure amplitude ratios and reference temperatures. According to Eq. (3.69), throat and Station 2 pressures are defined as

$$p_2 = p_0(X_{i2} + X_{r2} - 1)^{\frac{2\gamma_b}{\gamma_b - 1}} \quad (3.118a)$$

and

$$p_t = p_0 X_t^{\frac{2\gamma_b}{\gamma_b - 1}}. \quad (3.118b)$$

Density at each station can be derived from Eq. (3.115), Eq. (3.118), and the ideal gas law as

$$\rho_2 = \frac{P_0}{R_b T_{02}} (X_{r2} + X_{i2} - 1)^{\frac{2}{\gamma_b - 1}} \quad (3.119a)$$

and

$$\rho_t = \frac{P_0}{R_b T_{0t}} (X_t)^{\frac{2}{\gamma_b - 1}}, \quad (3.119b)$$

where  $R_b$  is the ideal gas constant evaluated with  $y_b$ . The velocity at Station 2  $U_2$  is determined by the incident and reflected pressure amplitude ratios. Assuming inflow to be positive,

$$U_2 = \frac{2}{\gamma_b - 1} \sqrt{\gamma_b T_{02} R_b} [X_{r2} - X_{i2}]. \quad (3.120)$$

The throat variables are expressed in terms of the superposed pressure amplitude ratio  $X_t$  because neither the rightward nor leftward traveling waves are known. Therefore,  $U_t$  must be solved iteratively.

### 3.6.3 Pipe Inflow Constraints

Pipe boundary inflow and outflow require distinctly different solution approaches, and therefore are discussed in separate sections. Pipe inflow, flow from OD volume into a pipe, is assumed positive, i.e.,  $U_t > 0$ . Similar to the adjoined pipe solution, pressure amplitude ratios and reference temperatures at the boundary are unknown and must be solved for using isentropic relationships and conservation laws. Examining the conservation laws and relationships presented in the previous section reveals that  $X_{r2}$ ,  $X_t$ ,  $T_{0t}$ ,  $T_{02}$ , and  $U_t$  are unknown. As a result, the boundary solution requires five constraints. Some constraints result in a direct solution to specific variables, while the remaining constraint equations must be solved iteratively. For

inflow, conservation of mass and energy equations provide three constraints. Isentropic assumptions and conservation of momentum provide the remaining relationships.

During inflow, the gas must contract to pass through the boundary throat area  $A_t$ . The contraction does not create turbulence and can be considered an isentropic process. According to the definition of the reference temperature, the OD volume and throat reference temperature can be equated; thus, for inflow,

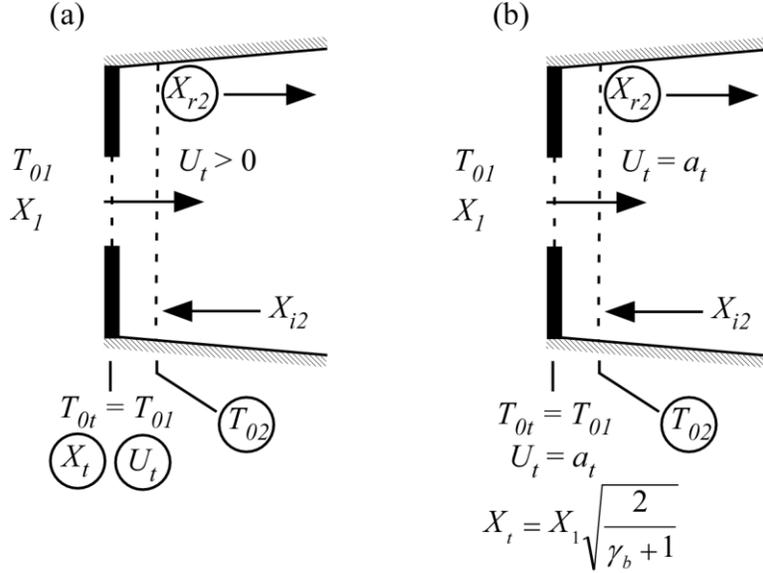
$$T_{0t} = T_{01} \quad (3.121)$$

Exiting the throat, the gas expands to the area  $A_2$ , giving rise to particle flow separation and turbulent vortices. The flow separation implies a non-isentropic process, and another relationship must be used to determine the downstream reference temperature  $T_{02}$ . Using conservation of momentum, flow information at the throat and Station 2 can be related. The downstream momentum equation can be expressed as

$$A_2(p_t - p_2) + \dot{m}_2(U_t - U_2) = 0 \quad (3.122)$$

For subsonic inflow, the isentropic contraction assumption expressed in Eq. (3.121) gives a direct solution to  $T_{0t}$ . Referring to Figure 3.11(a), the remaining variables,  $X_{r2}$ ,  $X_t$ ,  $T_{02}$ , and  $U_t$ , are determined by simultaneously solving the conservation of mass, Eq. (3.107); energy, Eq. (3.108) and Eq. (3.109); and momentum, Eq. (3.122), equations. Before solving, equations must be expressed in terms of acoustic variables and boundary information. Substituting acoustic wave variables into Eq. (3.107), conservation of mass provides the constraint

$$\frac{C_D A_t}{T_{0t}} U_t (X_t)^{\frac{2}{\gamma_b - 1}} - \frac{2A_2}{T_{02}(\gamma_b - 1)} \sqrt{\gamma_b T_{02} R_b} (X_{r2} - X_{i2})(X_{r2} + X_{i2} - 1)^{\frac{2}{\gamma_b - 1}} = 0 \quad (3.123)$$



**Figure 3.11: Boundary constraints and unknown parameters for (a) subsonic and (b) choked inflow (unknowns circled)**

The conservation of energy equations contain terms for enthalpy, and from preliminary testing, calculating enthalpy at the local temperature has convergence issues. To provide stability, enthalpy changes are assumed to have constant slope, implying a constant specific heat  $C_p$ . For an ideal gas, the change in enthalpy between the throat and Station 2 can be expressed as

$$h_t - h_2 = C_p(T_t - T_2) = \frac{\gamma_b R_b}{\gamma_b - 1} (T_t - T_2) = \frac{a_t^2 - a_2^2}{\gamma_b - 1} \quad (3.124)$$

Now, the conservation of energy given in Eq. (3.108) becomes

$$\frac{\gamma_b R_b}{\gamma_b - 1} [T_{0t}(X_t)^2 - T_{02}(X_{r2} + X_{i2} - 1)^2] + \left[ \frac{U_t^2}{2} - \frac{2\gamma_b T_{02} R_b}{(\gamma_b - 1)^2} (X_{r2} - X_{i2})^2 \right] = 0 \quad (3.125)$$

Similarly, the conservation of energy from the 0D volume to throat in Eq. (3.109) becomes

$$\frac{\gamma_b R_b}{\gamma_b - 1} [T_{01}(X_1)^2 - T_{02}(X_{r2} + X_{i2} - 1)^2] - \frac{2\gamma_b T_{02} R_b}{(\gamma_b - 1)^2} (X_{r2} - X_{i2})^2 = 0 \quad (3.126)$$

The final constraint defined in Eq. (3.122), conservation of momentum, can be expressed as

$$A_2 \left( X_t^{\frac{2\gamma_b}{\gamma_b-1}} - (X_{r2} + X_{i2} - 1)^{\frac{2\gamma_b}{\gamma_b-1}} \right) + \frac{2A_2 \sqrt{\gamma_b T_{02} R_b}}{R_b T_{02} (\gamma_b - 1)} (X_{r2} - X_{i2}) (X_{r2} + X_{i2} - 1)^{\frac{2}{\gamma_b-1}} \left[ U_t - \frac{2}{\gamma_b - 1} \sqrt{\gamma_b T_{02} R_b} (X_{r2} - X_{i2}) \right] = 0 \quad (3.127)$$

Using the constraint equations, the throat velocity may be found to reach or exceed the local acoustic velocity depending on boundary conditions and throat flow area. However, the particle velocity at the throat  $U_t$  cannot exceed the local acoustic velocity. Therefore, new relationships must be derived for choked flow. The ratio between the throat and OD volume pressures for choked flow, known as the critical pressure ratio, can be derived from the conservation of energy equations. The critical pressure ratio is derived as

$$\frac{p_t}{p_1} = \left( \frac{2}{\gamma_b + 1} \right)^{\frac{\gamma_b}{\gamma_b-1}} \quad (3.128)$$

By substituting pressure relationships into Eq. (3.128), the throat pressure amplitude ratio  $X_t$  can be directly determined by

$$X_t = X_1 \sqrt{\frac{2}{\gamma_b + 1}} \quad (3.129)$$

Now, the particle velocity at the throat can be calculated directly by

$$U_t = a_t = X_t \sqrt{\gamma_b T_{0t} R_b} \quad (3.130)$$

Choked inflow allows  $X_t$  and  $U_t$  to be calculated directly using Eq. (3.129) and Eq. (3.130).

Visualized in Figure 3.11(b), the remaining unknown parameters,  $X_{r2}$  and  $T_{02}$ , are evaluated by

solving the conservation of mass, Eq. (3.107), and energy, Eq. (3.109), equations assuming velocity at Station 2 is subsonic.

### 3.6.4 Pipe Outflow Constraints

Pipe outflow, defined by  $U_t < 0$ , requires the same number of constraints as the pipe inflow. The unknown variables,  $X_{r2}$ ,  $X_t$ ,  $T_{0t}$ ,  $T_{02}$ , and  $U_t$ , are determined from conservation laws and isentropic relationships. As before, flow contraction is assumed isentropic. Thus, for pipe outflow, the throat and Station 2 reference temperatures can be defined as

$$T_{0t} = T_{02} = T_{0c} \quad (3.131)$$

As flow exits the throat, the gas expands into the open space of the volume, creating significant turbulence. The dissipation of energy due to turbulence has traditionally been assumed to not produce pressure recovery. With no pressure recovery from the throat to the volume implying  $p_t = p_t$ , the throat pressure amplitude ratio  $X_t$  is defined as

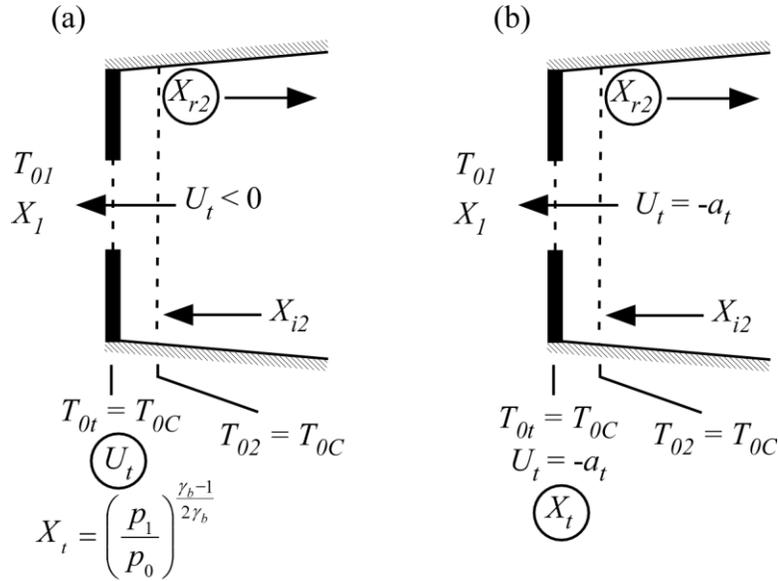
$$X_t = \left( \frac{p_t}{p_0} \right)^{\frac{\gamma_b - 1}{2\gamma_b}} \quad (3.132)$$

The assumptions in Eq. (3.131) and Eq. (3.132) provide direct solutions to  $T_{0t}$ ,  $T_{02}$ , and  $X_t$ . Shown in Figure 3.12, the remaining unknown variables,  $U_t$  and  $X_{r2}$ , can be determined from the conservation of mass and energy equations. Like pipe inflow, mass and energy must be conserved from Station 2 to the boundary throat. Substituting acoustic wave variables into Eq. (3.107), conservation of mass provides the constraint

$$C_D A_t U_t (X_t)^{\frac{2}{\gamma_b - 1}} - \frac{2A_2}{\gamma_b - 1} \sqrt{\gamma_b T_{02} R_b} (X_{r2} - X_{i2}) (X_{r2} + X_{i2} - 1)^{\frac{2}{\gamma_b - 1}} = 0 \quad (3.133)$$

Using the principles discussed for pipe inflow, the conservation of energy given in Eq. (3.108) can be expressed as

$$\frac{\gamma_b R_b}{\gamma_b - 1} [T_{0t} (X_t)^2 - T_{02} (X_{r2} + X_{i2} - 1)^2] + \left[ \frac{U_t^2}{2} - \frac{2\gamma_b T_{02} R_b}{(\gamma_b - 1)^2} (X_{r2} - X_{i2})^2 \right] = 0 \quad (3.134)$$



**Figure 3.12: Boundary constraints and unknown parameters for (a) subsonic and (b) choked outflow (unknowns circled)**

When solving for  $U_t$  and  $X_{r2}$  with the constraint equations, the throat velocity  $U_t$  may be found to reach or exceed the local acoustic velocity. New relationships must be introduced for choked flow. The isentropic relationships defined in Eq. (3.131) are still valid for choked outflow but velocity is restricted by

$$U_t = -a_t = -X_t \sqrt{\gamma_b T_{0t} R_b} \quad (3.135)$$

With the constraint, the pressure recovery assumption defined in Eq. (3.132) can be dropped for choked flow and Eq. (3.135) is substituted into the conservation of mass and energy equations. Substituting Eq. (3.135) into Eq. (3.133), conservation of mass for choked outflow becomes

$$-C_D A_t \sqrt{\gamma_b T_{0t} R_b} (X_t)^{\frac{\gamma_b+1}{\gamma_b-1}} - \frac{2A_2}{\gamma_b-1} \sqrt{\gamma_b T_{02} R_b} (X_{r2} - X_{i2})(X_{r2} + X_{i2} - 1)^{\frac{2}{\gamma_b-1}} = 0. \quad (3.136)$$

Substituting Eq. (3.135) into Eq. (3.134), conservation of energy for choked outflow then becomes

$$\frac{\gamma_b R_b}{\gamma_b - 1} [T_{0t} (X_t)^2 - T_{02} (X_{r2} + X_{i2} - 1)^2] + \left[ \frac{\gamma_b T_{0t} R_b X_t^2}{2} - \frac{2\gamma_b T_{02} R_b}{(\gamma_b - 1)^2} (X_{r2} - X_{i2})^2 \right] = 0. \quad (3.137)$$

By defining  $U_t$  directly and dropping the pressure recovery constraint,  $X_t$  and  $X_{r2}$  become unknown variables which can be solved with Eq. (3.136) and Eq. (3.137) assuming velocity at Station 2 is subsonic.

### 3.6.5 Solution Overview

When determining the mass and energy flow rates at the interface between a 0D volume and a 1D cell, four distinct situations can be encountered: subsonic inflow, choked inflow, subsonic outflow, and choked outflow. Each situation requires different solution approaches, and because boundary state variables are not available, constraints must be formed using acoustic wave information. After expressing boundary variables in terms of reference temperatures and pressure amplitude ratios, five unknown variables must be determined:  $X_{r2}$ ,  $X_t$ ,  $T_{0t}$ ,  $T_{02}$ , and  $U_t$ . Some constraints provide direct solutions to some of the unknowns, while the remaining variables must be solved iteratively. After solving for the unknown variables, the mass and energy flow rates at the boundary can be determined. Velocity range, unknown variables, directly applied constraints, and iteration equation numbers are summarized in Table 3.2 for each situation.

As mentioned previously, the constraint equations listed in Table 3.2 cannot be reduced and must be solved iteratively. Based on experience, Blair found the Newton-Raphson method to be stable, accurate, and fast for solving boundary constraints [22]. To implement a Newton-type solver, equations must be continuously differentiable. For a given flow condition, equations meet the requirement. When considering the solution as a whole, however, equations are not continuously differentiable at  $U_t = 0$ ,  $U_t = a_t$ , or  $U_t = -a_t$ . As a result, each flow condition is evaluated separately. The previously determined  $U_t$  dictates the solution method, and after each iteration, the next solution method is determined by the updated  $U_t$ . Therefore, the overall solver can alternate between subsonic inflow, choked inflow, subsonic outflow, and choked outflow equations without encountering a derivative discontinuity.

**Table 3.2: Solution summary for boundary conditions**

	Subsonic Inflow	Choked Inflow	Subsonic Outflow	Choked Outflow
Velocity Range	$0 \leq U_t < a_t$	$U_t = a_t$	$-a_t < U_t < 0$	$U_t = -a_t$
Unknowns	$X_{r2}, X_t, T_{02},$ and $U_t$	$X_{r2}$ and $T_{02}$	$X_{r2}$ and $U_t$	$X_{r2}$ and $X_t$
Direct Constraints	$T_{0t} = T_{0i}$	$T_{0t} = T_{0i}$ $X_t = X_1 \sqrt{\frac{2}{\gamma_b + 1}}$ $U_t = X_t \sqrt{\gamma_b T_{0t} R_b}$	$T_{02} = T_{0c}$ $T_{0t} = T_{0c}$ $X_t = \left(\frac{p_1}{p_0}\right)^{\frac{\gamma_b - 1}{2\gamma_b}}$	$T_{02} = T_{0c}$ $T_{0t} = T_{0c}$ $U_t = X_t \sqrt{\gamma_b T_{0t} R_b}$
Constraint Equations	3.123, 3.125, 3.126, and 3.127	3.123 and 3.126	3.133 and 3.134	3.136 and 3.137

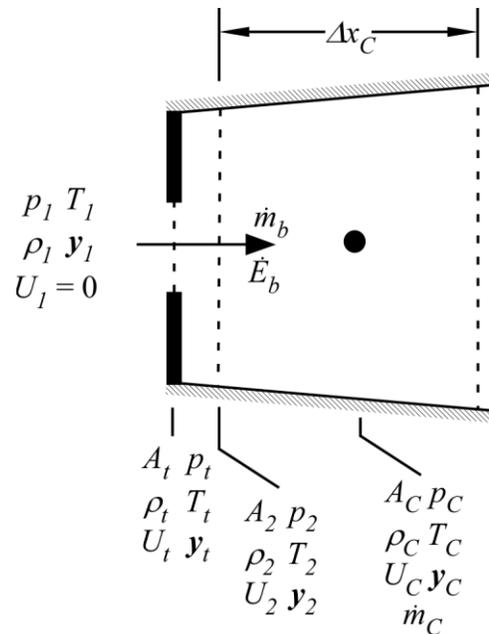
### 3.6.6 Boundary Mass and Energy Flow Rates

After calculating unknown variables listed in Table 3.2, boundary mass and energy flow rates,  $\dot{m}_2$  and  $\dot{E}_2$ , can be calculated with  $T_{02}$ ,  $X_{i2}$ , and  $X_{r2}$ . Note that an abrupt change in volume pressure  $p_1$  or temperature  $T_1$  inevitably translates into an abrupt change in boundary mass and energy flow rates, and as a result, causes stability issues. To prevent numerical instability, the

rate of change of boundary mass flow rate  $\dot{m}_b$ , shown in Figure 3.13 must be regulated. The rate of change of  $\dot{m}_b$  can be determined by applying the conservation of momentum from the cell center to Station 2. However, the resulting formulation would include half the cell length, and according to the CFL condition defined in Eq. (3.45), the stable time step would be halved as a result. Additionally, the formulation does not follow the central difference scheme utilized by the staggered grid approach. Considering the issues with applying the momentum equation, the rate of change of  $\dot{m}_b$  is regulated by a time constant  $\tau_b$ . Using  $\dot{m}_2$  as the target mass flow rate, the rate change in  $\dot{m}_b$  is defined as

$$\frac{d\dot{m}_b}{dt} = \frac{1}{\tau_b} (\dot{m}_2 - \dot{m}_b) \quad (3.138)$$

The time constant  $\tau_b$  can be selected based on the simulation time step or the CFL condition.



**Figure 3.13: Mass and energy flow rate across boundary**

### 3.7 Flow Junction

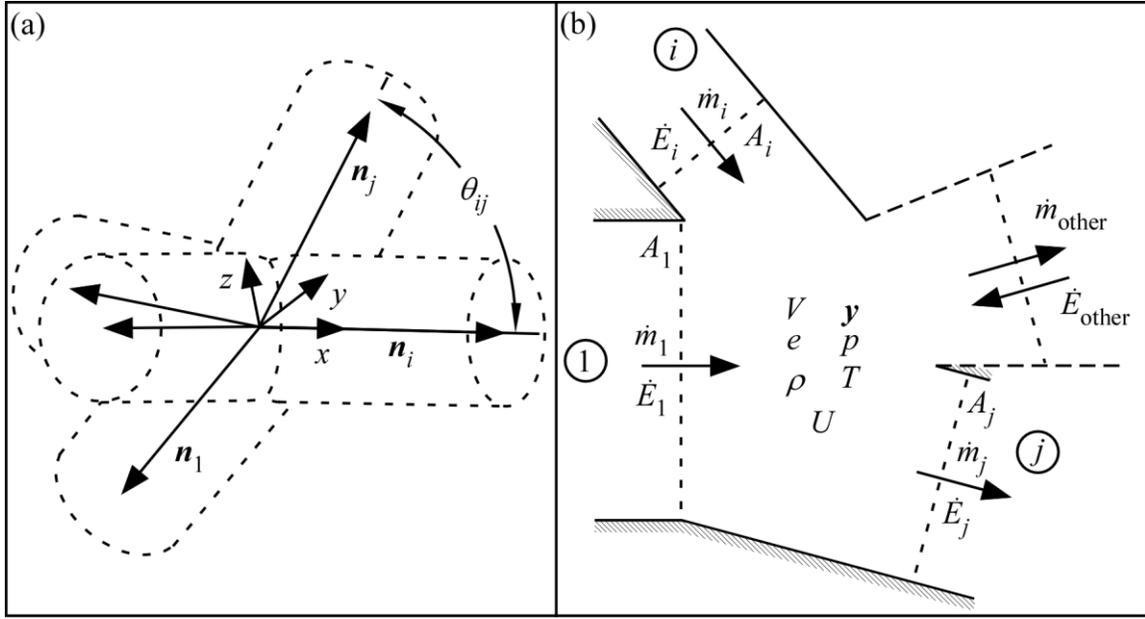
Connecting more than two pipes requires a flow junction to calculate the mass and energy flow rates for each branch. The junction, similar to 1D control volumes, conserves mass and energy, while momentum is calculated at the boundaries by the adjoined pipe model outlined previously. Unlike the 1D pipe, the junction model accounts for pressure losses at each branch according to the connection angle and mass flow rate.

#### 3.7.1 General flow junction

In general, multiple 1D pipe branches can be connected at a single point in any orientation. The direction of the  $i^{\text{th}}$  inflow branch is represented by the unit vector  $\mathbf{n}_i$ , and the  $j^{\text{th}}$  outflow branch direction is characterized by  $\mathbf{n}_j$  as shown in Figure 3.14(a), where the angle between two branches can be calculated by

$$\theta_{ij} = \cos^{-1}(\mathbf{n}_i \cdot \mathbf{n}_j) \quad (3.139)$$

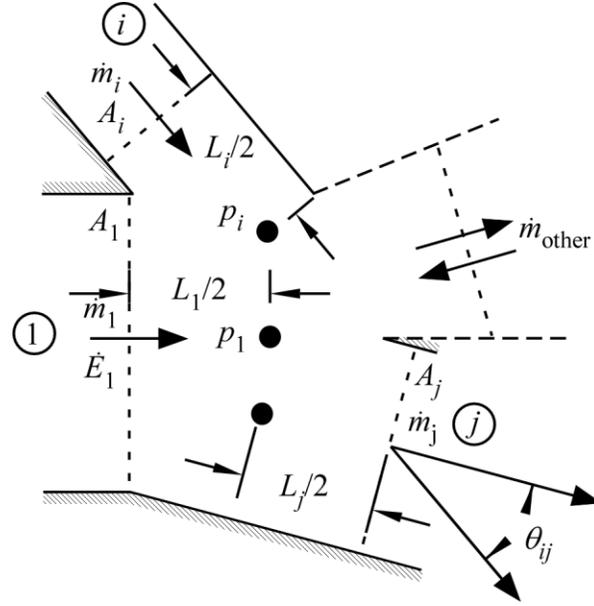
The inflow or outflow condition can switch during simulation depending on the adjoined pipe, but the unit vector remains constant. The junction has a volume  $V$ , and states variables shown in Figure 3.14(b) are determined by conservation laws and flow from each branch. To calculate rate of change of momentum at each branch, the junction provides pressure to the adjoined pipe model, and in return, the connected pipes provide mass and energy flow rates.



**Figure 3.14: Junction (a) orientation vectors and (b) general flow representation**

Adjoined pipe models require characteristic lengths to calculate the rate of change of momentum at each branch. The characteristic length for each branch is defined as the distance from the branch boundary to an opposing solid surface or another boundary. Therefore, referring to Figure 3.15, each branch has a virtual cell center at  $L/2$  from the branch pipe boundary. Pressure at each so-called cell center depends on flow direction: no change in pressure during inflow and each outflow branch has an associated loss in pressure. To simulate branch pressure, a pressure loss coefficient  $C_L$  can be introduced as a function of flow characteristics and branch angle. In general, the  $i^{\text{th}}$  inflow pressure and  $j^{\text{th}}$  outflow pressure are defined as

$$p_i = p \quad \text{and} \quad p_j = p - C_{L,j} \frac{\dot{m}_j}{A_j} . \quad (3.140)$$



**Figure 3.15: General junction characteristic lengths and pressure calculation parameters**

Pressure loss coefficients can be determined experimentally, but to remain predictive,  $C_{L,i}$  is calculated by a correlation published by Bassett *et al.* [90]:

$$C_{L,j} = 1 - \frac{A_j \dot{m}_{dat}}{A_{dat} \dot{m}_j} \cos\left(\frac{3}{4}(\pi - \theta_{dat,j})\right). \quad (3.141)$$

Bassett *et al.* suggested taking the datum, denoted with the subscript “dat,” as the branch with the largest inflow rate. Although the simplification provides accurate results in most flow conditions, the assumption can produce numerical instability when two flow rates enter the junction near the same flow rate: the loss coefficient defined in Eq. 3.141 can widely vary with different  $\theta_{dat,j}$  angles. Therefore, the loss coefficients are averaged over the  $N$  inflow branches, such that

$$C_{L,j} = \frac{\sum_{i=1}^N \left[ m_i - \frac{A_j \dot{m}_i^2}{A_i \dot{m}_j} \cos\left(\frac{3}{4}(\pi - \theta_{ij})\right) \right]}{\sum_{i=1}^N m_i}, \quad (3.142)$$

where the angle  $\theta_{ij}$  is defined in Eq. 3.139.

### 3.7.2 T-junction and Y-junction

Frequently encountered in engine modeling, T and Y-junctions are special forms of the junction model presented previously. The T-junction shown in Figure 3.16(a) has a main pipe with a branch entering the junction at an angle  $\beta$ . Like the 1D flow model, the characteristic lengths of the straight section ( $L_1$  and  $L_2$ ) are defined by the discretization length, while  $L_3$  depends on the pipe diameter  $D_{main}$  and angle  $\beta$ . By definition, the characteristic length is the distance from an adjoined pipe to another surface or the distance to another pipe boundary; thus,  $L_3$  can be defined as

$$L_3 = \begin{cases} \frac{D_{main}}{\sin(\beta)} & \text{if } \left| D_{main} \tan\left(\beta + \frac{\pi}{2}\right) \right| < \frac{L_1}{2} \\ \frac{L_1}{2|\cos(\beta)|} & \text{else} \end{cases} \quad (3.143)$$

Referring to Figure 3.16(b), the Y-junction splits from a main section into two branches at a half angle of  $\alpha$ , which can be used to characterize an intake or exhaust port with multiple valves.

Assuming a reasonably small  $\alpha$ , the characteristic lengths for each branch can be equated.

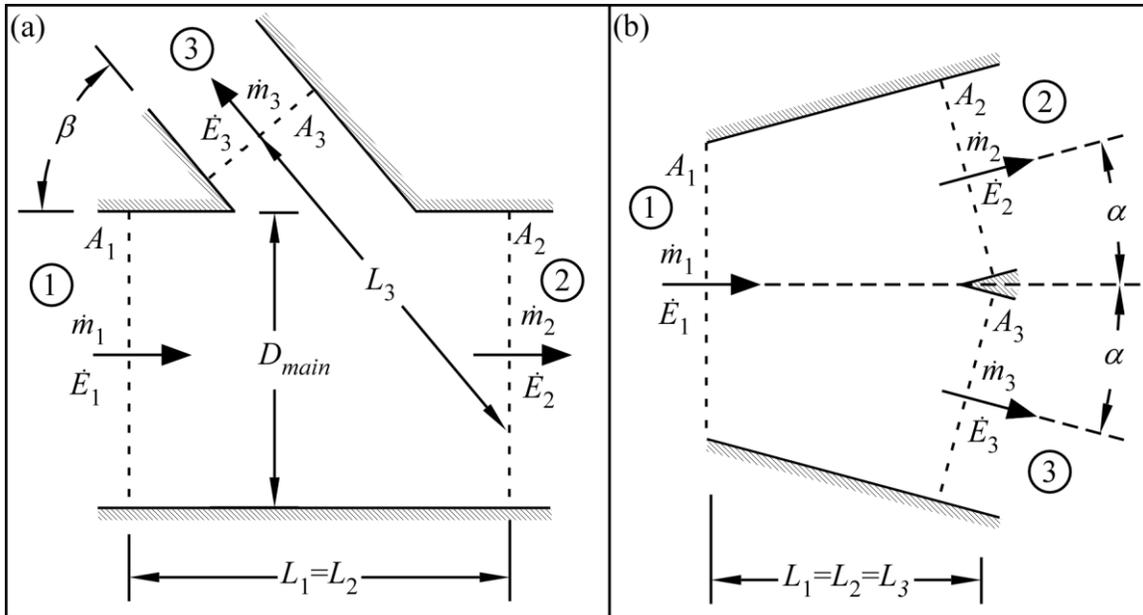


Figure 3.16: (a) T-junction and (b) Y-junction model parameters

## CHAPTER 4

### ENGINE CYLINDER AND DYNAMICS MODELS

An internal combustion engine converts chemical potential energy to mechanical energy by combusting fuel in a contained cylinder. The exothermic reaction raises the temperature of the cylinder gasses, thus raising the cylinder pressure and forcing the piston downward. A slider-crank mechanism then converts the linear piston force into a rotational torque. Fuel mass, air-fuel ratio, combustion rate, ignition timing, heat transfer, and exhaust outflow all contribute to the output torque. From a modeling standpoint, the 1D flow model predicts mass transfer into and out of the cylinder during intake and exhaust, while the cylinder model predicts piston force. During each engine cycle, conservation laws, equation of state, and a heat transfer model predict the cylinder pressure. The exothermic reaction is modeled based on the burn rate, which can be a simple fit or predicted based on cylinder turbulence.

#### 4.1 Crank Dynamics

A reciprocating piston engine utilizes a slider-crank mechanism to convert linear piston forces into rotational torque. Referring to the engine slider-crank mechanism shown in Figure 4.1, piston position  $x_p$  has a nonlinear relationship with the crank angle  $\theta_c$  that depends on the stroke  $L_s$  and rod length  $L_r$ . The distance from the crank axis to the wristpin  $s$  can be determined by the law of sines and cosines since the crank rotational radius  $a$  is half the cylinder stroke. Assuming the piston origin to be at piston Top Dead Center (TDC) and  $x_p = L_s$  at Bottom Dead Center (BDC), piston position can be calculated as

$$x_p = L_r + a - s = L_r + \frac{L_s}{2} - \left( \frac{L_s}{2} \cos(\theta_c) + \sqrt{L_r^2 - \frac{L_s^2}{4} \sin^2(\theta_c)} \right). \quad (4.1)$$

Like position, the instantaneous piston speed  $S_p$  has a direct relationship with the crank velocity  $\omega_c$ . Differentiating Eq. (4.1) with respect to time, yields

$$S_p = \frac{dx_p}{dt} = \frac{L_s}{2} \omega_c \sin(\theta_c) \left[ 1 + \frac{\cos(\theta_c)}{\sqrt{\left(\frac{2L_r}{L_s}\right)^2 - \sin^2(\theta_c)}} \right], \quad (4.2)$$

where the crank velocity  $\omega_c$  is defined as

$$\omega_c = \frac{d\theta_c}{dt}. \quad (4.3)$$

Frequently, the mean piston speed  $\bar{S}_p$  is an important parameter for discussing engine characteristics and predicting the cylinder heat transfer coefficient. Averaging Eq. (4.2) over a crank revolution, results in

$$\bar{S}_p = \frac{L_s \omega_c}{\pi}. \quad (4.4)$$

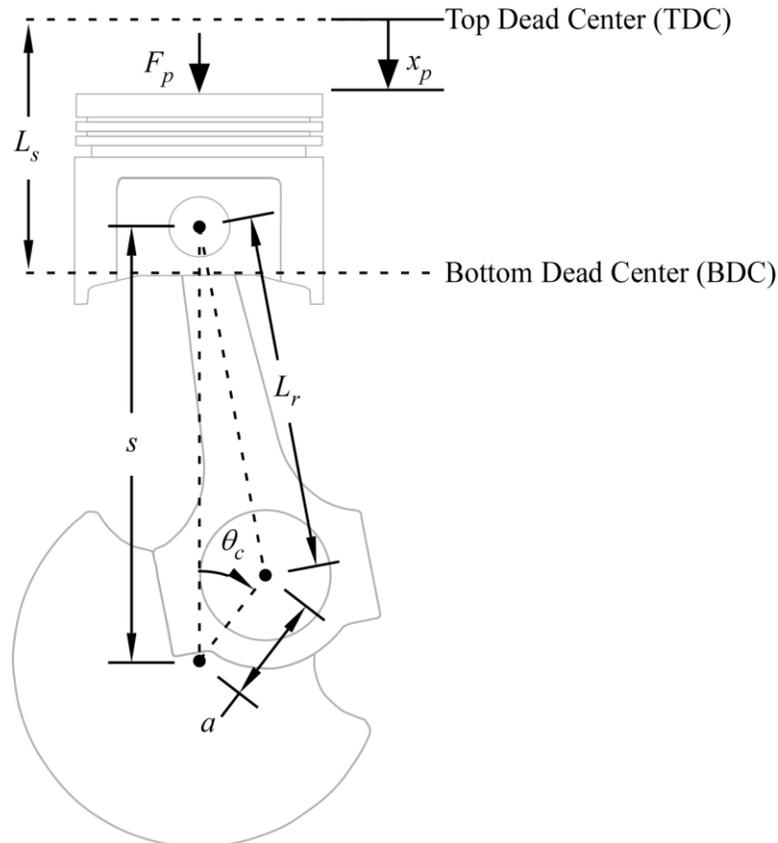
Referring to Figure 4.1, cylinder pressure produces a resultant force  $F_p$  that axially loads the connecting rod. The connecting rod then applies vertical and horizontal forces to the crankshaft, creating rotational torque. Neglecting connecting rod friction and inertial effects, the  $i^{\text{th}}$  cylinder torque  $T_{c,i}$  is given by

$$T_{c,i} = F_{p,i} \frac{\frac{L_s}{2} \sin\left(\theta_c + \arcsin\left(\frac{L_s}{2L_r} \sin(\theta_c)\right)\right)}{\cos\left(\arcsin\left(\frac{L_s}{2L_r} \sin(\theta_c)\right)\right)} \quad (4.5)$$

If the sum of the cylinder torques exceeds the load torque  $T_{load}$ , the crankshaft will accelerate, and if  $T_{load}$  exceeds the sum of cylinder torques, the crankshaft will decelerate. Applying Newton's second law for rotational motion, crank acceleration  $\alpha_c$  is governed by

$$\alpha_c = \frac{d\omega_c}{dt} = \frac{1}{J_c} \left( \sum_{i=1}^M T_{c,i} - T_{load} \right), \quad (4.6)$$

where  $J_c$  is the moment of inertia of the rotating assembly. Because piston and rod mass are neglected, all inertial effects are lumped into  $J_c$ .



**Figure 4.1: Engine slider-crank geometry**

## 4.2 Cylinder Conservation Laws

The cylinder is modeled as an open thermodynamic system, where the pressure inside the cylinder is assumed uniform, neglecting 3D flow field effects. If the combustion chamber is treated as a homogeneous single volume, the cylinder model is referred to as a single-zone “zero-dimensional” (0D) model. The volume temperature and pressure are derived from conservation of mass and energy. If the combustion chamber is split into burned and unburned zones (two-zone model), the model is sometimes referred to as “quasi-dimensional.” The flame spherically propagates from the spark location until quenched by the cylinder walls, thus requiring consideration of chamber geometry. Both zones have the same pressure but different temperatures. For modeling the gas exchange process and compression, a single-zone model is typically used. During combustion, a single-zone or two-zone approach can be used. Conservation of mass and energy for the two models are discussed.

### 4.2.1 Single-Zone and Gas Exchange Period

Cylinder temperature  $T_{cyl}$  and pressure  $p_{cyl}$  vary according to the conservation of mass and energy. Referring to Figure 4.2, mass flows across the boundary through  $N$  intake and exhaust ports. Assuming that the flow into the cylinder is positive, the conservation of mass states that

$$\frac{dm_{cyl}}{dt} = \sum_{k=1}^N \dot{m}_k, \quad (4.7)$$

where  $k$  indexes intake and exhaust valves. During intake and exhaust, the rate of change of cylinder mass fractions  $y_{cyl}$  depends on the flow direction. The  $k^{\text{th}}$  boundary mass fractions  $y_{bound,k}$  are defined as

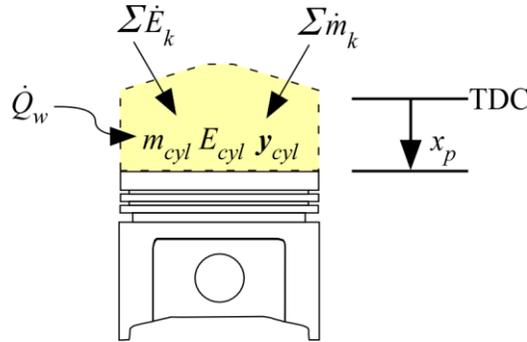
$$\mathbf{y}_{bound,k} = \begin{cases} \mathbf{y}_{port,k} & \dot{m}_k \geq 0 \\ \mathbf{y}_{cyl} & \dot{m}_k < 0, \end{cases} \quad (4.8)$$

where  $\mathbf{y}_{port,k}$  is the mass fractions in the  $k^{\text{th}}$  port. Applying conservation of mass for each gas species during intake and exhaust gas exchange results in the following:

$$\frac{d}{dt}(\mathbf{y}_{cyl} m_{cyl}) = \sum_{k=1}^N \mathbf{y}_{bound,k} \dot{m}_k. \quad (4.9)$$

For a single-zone combustion model, the rate of change of mass fractions in the cylinder  $\dot{\mathbf{y}}_{comb}$  is determined by a burned gas profile. By combining the combustion rate contribution and the conservation of mass relationship in Eq. (4.9), the cylinder gas species is governed by

$$\frac{d}{dt}(\mathbf{y}_{cyl}) = \frac{1}{m_{cyl}} \left( \sum_{k=1}^N \mathbf{y}_{bound,k} \dot{m}_k - \sum_{k=1}^N \mathbf{y}_{cyl} \dot{m}_k \right) + \dot{\mathbf{y}}_{comb}. \quad (4.10)$$



**Figure 4.2: Diagram of engine cylinder control volume**

Thermodynamic states are derived from the gas mass in the cylinder  $m_{cyl}$ , volume  $V_{cyl}$ , and internal energy  $E_{cyl}$ . Referring to Figure 4.2, the instantaneous cylinder volume changes with piston displacement  $x_p$  according to

$$V_{cyl} = V_c + x_p A_p, \quad (4.11)$$

where  $A_p$  is the piston area and  $V_c$  is the clearance volume. During expansion, the cylinder volume increases, producing work. During compression, work is applied to the control volume.

The work-transfer rate out of the cylinder volume is the product of piston force and velocity:

$$\dot{W}_{cyl} = \dot{x}_p A_p p_{cyl} \quad (4.12)$$

According to the first law of thermodynamics, work, flow, and heat transfer crossing the control volume boundaries shown in Figure 4.2 govern the rate of change of internal energy  $E_{cyl}$ .

Applying the first law of thermodynamics leads to

$$\frac{dE_{cyl}}{dt} = \sum_{k=1}^N \dot{E}_k - \dot{W}_{cyl} + \dot{Q}_w, \quad (4.13)$$

where the  $k^{\text{th}}$  energy flow rate  $\dot{E}_k$  is determined by the 1D flow model and  $E_{cyl}$  can be defined in terms of specific internal energy or enthalpy as

$$E_{cyl} = m_{cyl} e_{cyl} = m_{cyl} h_{cyl} - p_{cyl} V_{cyl} \quad (4.14)$$

Note that Eq. (4.13) does not include combustion heat addition because enthalpies and energies are expressed relative to the same datum.

Cylinder pressure  $p_{cyl}$  and temperature  $T_{cyl}$  must be derived from the volume  $V_{cyl}$ , mass  $m_{cyl}$ , and internal energy  $E_{cyl}$ . For an ideal gas, internal energy  $E_{cyl}$  is a function of temperature only, which means that  $T_{cyl}$  can be calculated from  $E_{cyl}$  and the internal energy lookup table. Using the cylinder temperature, the ideal gas law leads to

$$p_{cyl} = \frac{m_{cyl} R_{cyl} T_{cyl}}{V_{cyl}} \quad (4.15)$$

With a real gas law, internal energy  $E_{cyl}$  becomes an implicit function of pressure and temperature, and cylinder pressure and temperature must be calculated iteratively.

#### 4.2.2 Two-Zone Combustion Model

The conservation laws presented in the previous section define cylinder states during intake, exhaust, and single-zone combustion. For a more accurate representation of combustion, the burned and unburned gasses can be separated into two distinct volumes—two-zone model. Shown in Figure 4.3, the two-zone model assumes a uniform cylinder pressure  $p_{cyl}$  with separate zone temperatures ( $T_u$  and  $T_b$ ) and mass fractions ( $y_u$  and  $y_b$ ). Typically, the burned zone is approximated as a partial sphere with the origin at the spark location. Therefore, as combustion progresses, the unburned gases pass through the spherical flame front into the burned zone, consuming the unburned volume.

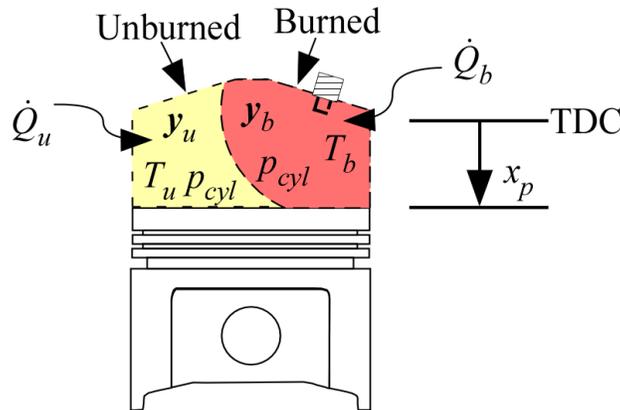


Figure 4.3: Diagram of two-zone combustion model

Physical constraints and conservation laws must be used to determine the thermodynamic states shown in Figure 4.3. Assuming no flow through the valves during combustion, conservation of mass provides the relationship

$$\frac{dm_b}{dt} = -\frac{dm_u}{dt} = \dot{m}_{comb}, \quad (4.16)$$

where  $\dot{m}_{comb}$  is the mass burn rate,  $m_b$  is the burned mass, and  $m_u$  is the unburned mass. By neglecting heat transfer between the two zones, conservation of energy applied separately to each control volume yields the following equations:

$$\frac{d(u_u m_u)}{dt} = m_u c_{v,u} \frac{dT_u}{dt} + u_u \frac{dm_u}{dt} = -P_{cyl} \frac{dV_u}{dt} - h_u \dot{m}_{comb} - \dot{Q}_u, \quad (4.17)$$

$$\frac{d(u_b m_b)}{dt} = m_b c_{v,b} \frac{dT_b}{dt} + u_b \frac{dm_b}{dt} = -P_{cyl} \frac{dV_b}{dt} + h_u \dot{m}_{comb} - \dot{Q}_b, \quad (4.18)$$

where  $c_v$  is defined as  $c_v = du/dT$ , subscript “u” denotes unburned, and subscript “b” denotes burned. In order to solve for the rate of change of zone temperatures, the volume derivatives must be derived from the ideal gas law. Differentiating the ideal gas equation (defined in Eq. 4.15) for single zone) for both control volumes leads to

$$P_{cyl} \frac{dV_u}{dt} + V_u \frac{dp_{cyl}}{dt} = R_u T_u \frac{dm_u}{dt} + m_u R_u \frac{dT_u}{dt}, \quad (4.19)$$

$$P_{cyl} \frac{dV_b}{dt} + V_u \frac{dp_{cyl}}{dt} = R_b T_b \frac{dm_b}{dt} + m_b R_b \frac{dT_b}{dt}. \quad (4.20)$$

Combining Eq. (4.17) and Eq. (4.19) and using the specific heat relationship  $c_{v,u} + R_u = c_{p,u}$ , the rate of change of the unburned gas temperature  $T_u$  becomes

$$\frac{dT_u}{dt} = \frac{1}{m_u c_{p,u}} \left( V_u \frac{dp_{cyl}}{dt} - \dot{Q}_u \right). \quad (4.21)$$

Similarly, combining Eq. (4.18) and Eq. (4.20) gives

$$\frac{dT_b}{dt} = \frac{1}{m_b c_{p,b}} \left( V_b \frac{dp_{cyl}}{dt} + \dot{m}_{comb} (h_u - h_b) - \dot{Q}_b \right). \quad (4.22)$$

The rate of change of cylinder pressure  $dp_{cyl}/dt$  in Eq. (4.21) and Eq. (4.22) can be derived from the model constraints. The total cylinder volume defined in Eq. (4.11) provides the constraints

$$V_{cyl} = V_u + V_b, \quad (4.23)$$

$$\frac{dV_{cyl}}{dt} = \frac{dV_u}{dt} + \frac{dV_b}{dt}. \quad (4.24)$$

Substituting  $dV_u/dt$  and  $dV_b/dt$  from the ideal gas law equations (Eq. (4.19) and Eq. (4.20)) into Eq. (4.24), the following can be concluded:

$$\frac{dV_{cyl}}{dt} = \left( \frac{V_b}{m_b} - \frac{V_u}{m_u} \right) \dot{m}_{comb} + \frac{V_u}{T_u} \frac{dT_u}{dt} + \frac{V_b}{T_b} \frac{dT_b}{dt} - \frac{V_{cyl}}{p_{cyl}} \frac{dp_{cyl}}{dt}. \quad (4.25)$$

Finally,  $dp_{cyl}/dt$  can be derived by substituting Eq. (4.21) and Eq. (4.22) into Eq. (4.25) and rearranging:

$$\frac{dp_{cyl}}{dt} = \frac{p_{cyl} \frac{dV_{cyl}}{dt} + \dot{m}_{comb} (R_u T_u - R_b T_b) + \frac{R_u}{c_{p,u}} \dot{Q}_u + \frac{R_b}{c_{p,b}} \dot{Q}_b + \frac{R_b \dot{m}_{comb}}{c_{p,b}} (h_b - h_u)}{\frac{R_u}{c_{p,u}} V_u + \frac{R_b}{c_{p,b}} V_b - V_{cyl}} \quad (4.26)$$

In summary, the rates of change of  $m_u$ ,  $T_u$ , and  $T_b$  are calculated by Eq. (4.16), Eq. (4.21), and Eq. (4.22), respectively. Therefore,  $m_u$  and  $T_b$  must be initialized at the start of combustion. The initial burned temperature  $T_b$  is assumed to be at the adiabatic flame temperature, and to initialize  $m_u$ , the burned mass  $m_b$  is derived from the initial spark volume and subtracted from the total cylinder mass. During combustion, gas properties are calculated with the appropriate zone mass fractions and temperature. Cylinder pressure and zone volumes are derived from the ideal gas equations of state:

$$P_{cyl} = \frac{m_u R_u T_u + m_b R_b T_b}{V_{cyl}}, \quad (4.27)$$

$$V_b = \frac{m_u R_u T_u}{P_{cyl}}, \quad (4.28)$$

$$V_u = V_{cyl} - V_b. \quad (4.29)$$

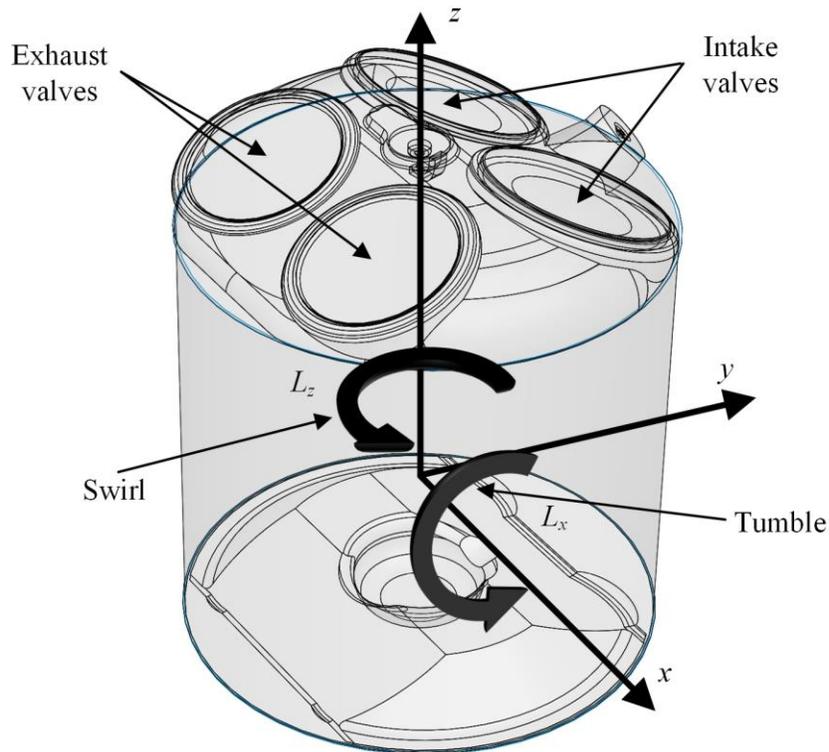
### 4.3 Cylinder Charge Motion and Turbulence

Turbulence significantly impacts combustion and cylinder heat transfer. To predict turbulence, the cascade concept is frequently employed—energy from the mean charge motion produces turbulent eddies on the scale of the flow geometry which break into progressively smaller eddies until the turbulence dissipates at the smallest scale due to viscous shear. During the intake phase, flow through the intake ports produces mean charge motion, frequently described as swirl and tumble. Intake flow kinetic energy not converted to swirl or tumble motion produces turbulence. During compression, the rapid increase in density and decay of mean charge motion result in turbulent kinetic energy production. At the same time, kinetic energy of turbulence at the smallest turbulent length scales dissipates into internal energy. During expansion, little or none of the mean flow remains, and turbulent kinetic energy continues to dissipate. The turbulent kinetic energy and eddy length scales contribute to the turbulent flame speed.

#### 4.3.1 Mean Flow Model

Intake port geometry, valve lift, engine speed, and air flow rate all contribute to the mean charge motion. Cylinder flow can also be actively controlled by variable valve lift, valve timing, swirl flap, or tumble flap. Therefore, several factors must be considered in order to estimate

cylinder flow across a wide range of operating conditions. Cylinder flow would be best described in three dimensions using the Navier-Stokes equations, but the approach is too computationally expensive. As an alternative, cylinder charge motion is often characterized by angular momentum. The rate of change in angular momentum due to intake flow and turbulence production can be calibrated to match experimental data or CFD simulations [91]. Referring to Figure 4.4, angular momentum about the  $z$ -axis  $L_z$ , referred to as “swirl,” is generated by asymmetric flow through two intake ports or by port geometry. Angular momentum about the  $x$ -axis  $L_x$ , referred to as “tumble,” is generated by port geometry and the intake port offset from the center of the cylinder. Angular momentum about the  $y$ -axis, minor tumble, is ignored. All tumble motion is assumed to be about the  $x$ -axis.



**Figure 4.4: Simplified cylinder flow model characterized by swirl angular momentum  $L_z$  and tumble angular momentum  $L_x$**

The angular momentum vector describing the cylinder charge motion  $\mathbf{L}$  is defined as

$$\mathbf{L} = \begin{bmatrix} L_x \\ L_y \\ L_z \end{bmatrix} = \begin{bmatrix} J_x \omega_x \\ J_y \omega_y \\ J_z \omega_z \end{bmatrix}, \quad (4.30)$$

where  $J_x$ ,  $J_y$ , and  $J_z$  are moment of inertias about the three axes, and  $\omega_x$ ,  $\omega_y$ , and  $\omega_z$  are angular velocities about the corresponding axes. Shown in Figure 4.4,  $L_x$  represents tumble and,  $L_z$  represents swirl. The moment of inertias are approximated for a cylinder with diameter  $B$  and height  $x_{cyl}$ , defined as

$$x_{cyl} = \frac{4V_{cyl}}{\pi B^2}. \quad (4.31)$$

Using a cylinder approximation and assuming tumble rotation to be centered at  $x_{cyl}/2$  above the piston, the  $x$  and  $y$  moments of inertia are defined as

$$J_x = J_y = \frac{m_{cyl}}{4} \left[ \frac{B^2}{4} + \frac{x_{cyl}^2}{3} \right]. \quad (4.32)$$

For rotation about the  $z$ -axis, swirl moment of inertia  $J_z$  is defined as

$$J_z = \frac{m_{cyl} B^2}{8}. \quad (4.33)$$

Note that  $J_x$  and  $J_y$  decrease during compression while  $J_z$  remains constant. Therefore, without a resistive torque, tumble velocities  $\omega_x$  and  $\omega_y$  increase while  $\omega_z$  remains constant during compression.

The rate of change of angular momentum is determined from the intake flow, exhaust flow, and interaction with combustion chamber walls. The following assumptions are made to model swirl  $L_z$  and tumble  $L_x$  angular momentum [91]:

- The minor tumble angular momentum  $L_y$  is neglected.  $L_x$  accounts for all tumble motion.
- Tumble and swirl are treated independently.
- Exhaust backflow has a negligible influence on the charge motion and turbulence. Therefore, intake flow is the only driver of angular momentum.
- Mass of gas exiting the chamber volume carries angular momentum
- Combustion does not affect mean charge motion.

Using the assumptions, the rate of change of tumble angular momentum  $L_x$  can be calculated as

$$\frac{dL_x}{dt} = \tau_{x,in} + \tau_{x,shr} - L_x \frac{\dot{m}_{ex}}{m_{cyl}}, \quad (4.34)$$

where  $\tau_{x,in}$  is the tumble torque generated from the incoming flow,  $\tau_{x,shr}$  is the tumble resistive torque resulting from shear forces, and  $\dot{m}_{ex}$  is the total mass flow rate exiting the chamber. In a similar fashion, the rate of change of swirl angular momentum  $L_z$  can be calculated as

$$\frac{dL_z}{dt} = \tau_{z,in} + \tau_{z,shr} - L_z \frac{\dot{m}_{ex}}{m_{cyl}}, \quad (4.35)$$

where  $\tau_{z,in}$  is the swirl torque generated from the incoming flow, and  $\tau_{z,shr}$  is the swirl resistive torque resulting from shear forces. In Eqs. (4.34) and (4.35), the first terms represent charge motion generated by flow through the intake valves. The second terms represent fluid shearing forces and wall friction that produce turbulent eddies. The final terms account for the loss of momentum caused by gases exiting the chamber through intake or exhaust valves. Because little to no angular momentum remains at the start of the exhaust phase, the current cycle has little impact on swirl and tumble in subsequent cycles. Therefore,  $L_x$  and  $L_z$  quickly reach steady state and can be initialized to be  $L_x = L_z = 0$ .

The ability to produce swirl and tumble is frequently quantified by dimensionless swirl and tumble numbers ( $N_S$  and  $N_T$ ) measured on a steady state flowbench. The ratio between tangential and ideal velocities is one definition of the swirl and tumble numbers published in literature [92], [93]. This definition results in the following relationships between cylinder flow and torque [92], [93]:

$$N_S = \frac{8\tau_{z,in}}{\dot{m}_{in}Bu_{in}}, \quad (4.36)$$

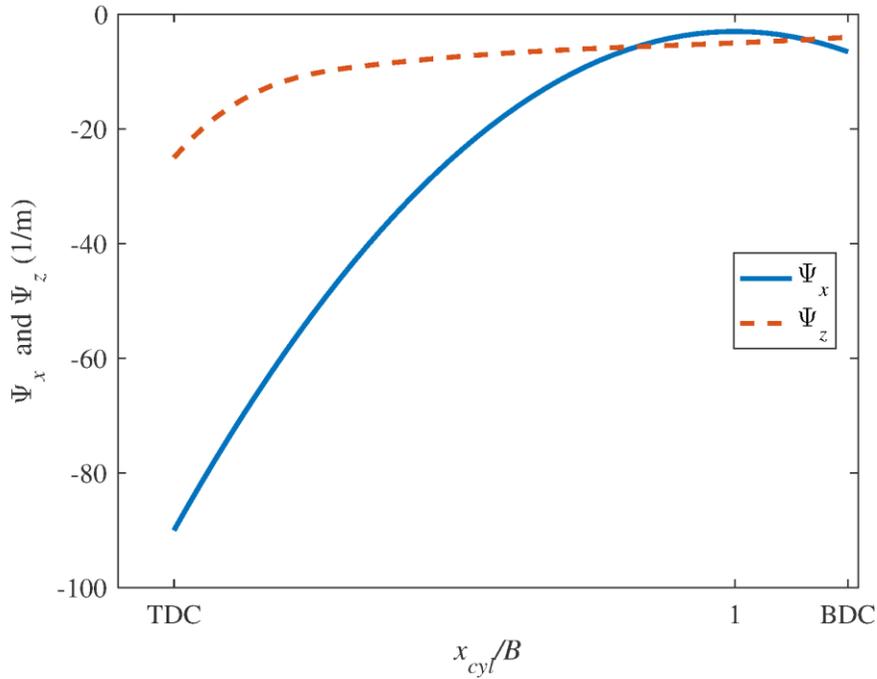
$$N_T = \frac{8\tau_{x,in}}{\dot{m}_{in}Bu_{in}}. \quad (4.37)$$

where  $\dot{m}_{in}$  is the total intake mass flow rate, and  $u_{in}$  is the intake flow velocity.  $N_S$  and  $N_T$  can be correlated to several factors, involving valve lift, tumble flap angle, swirl flap angle, and pressure drop. Because differences in flow rates between two intake valves affect swirl and tumble,  $N_S$  and  $N_T$  cannot be treated independently for each valve. Yun and Lee, for example, measured swirl and tumble numbers for several combinations of left and right intake valve lifts, producing a two-dimensional lookup table [92]. A similar approach could be used for a swirl or tumble flap. Without charge motion control,  $N_T$  and  $N_S$  can be modeled as functions of valve lift only, but the effect of pressure drop can also be included for more accuracy.

Fluid shear stresses and wall friction degrades swirl and tumble kinetic energies [94], [95]. The resulting shear torques  $\tau_{z,shr}$  and  $\tau_{x,shr}$  can be determined by integrating the wall friction over the chamber surface and fluid stress over chamber volume but would require knowledge of the spatial flow field. Grasreiner *et al.* modeled swirl and tumble flow field as a Taylor-Green vortex and correlated angular momentum decay using 3D CFD simulations [91]. The authors defined the rate of change of angular momentum due to shear as

$$\frac{dL_{dir,shr}}{dt} = \tau_{dir,shr} = L_{dir} \Psi_{dir} \sqrt{k}, \quad (4.38)$$

where  $\Psi_{dir}$  is the charge motion time decaying function for a given direction, and  $k$  is the turbulent kinetic energy. Because the flow structure depends on the chamber geometry,  $\Psi_x$  and  $\Psi_z$  vary with piston displacement and can be adapted to various piston and cylinder head designs. Shown in Figure 4.5, tumble decay increases as the piston approaches TDC due to the substantial deformation of the tumble vortex. On the other hand, the swirl vortex shape can be preserved, and the increase in  $\Psi_z$  as the piston approaches TDC can be attributed to increasing frictional losses [91].



**Figure 4.5: Charge motion decay functions  $\Psi_x$  and  $\Psi_z$  derived from 3D Taylor-Green vortex simulations (Adapted from [91])**

Intake flow kinetic energy is conserved in the form of mean charge motion, turbulent kinetic energy, and internal energy. Rotational kinetic energy in each direction  $KE_{dir,rot}$  resulting from Eqs. (4.34) and (4.35) is defined as

$$KE_{dir,rot} = \frac{1}{2} J_{dir} \omega_{dir}^2 = \frac{1}{2} \frac{L_{dir}^2}{J_{dir}}, \quad (4.39)$$

where  $dir$  can be substituted for each axis. The transfer of swirl and tumble rotational energy to turbulent kinetic energy is derived from the shear torques  $\tau_{z,shr}$  and  $\tau_{x,shr}$ . Assuming quasi-steady volume geometry (constant  $J_{dir}$ ), the rate of change of kinetic energy due to shearing forces can be determined by differentiating Eq. (4.39) and replacing the rate of change of angular momentum with the shear torque:

$$\frac{d(KE_{dir,rot})}{dt} = \frac{L_{dir}}{J_{dir}} \left( \frac{dL_{dir}}{dt} \right)_{shr} = \frac{L_{dir}}{J_{dir}} \tau_{dir,shr}. \quad (4.40)$$

Similarly, the rate of change of rotational kinetic energy due to intake flow is determined by

$$\frac{d(KE_{dir,rot})}{dt} = \frac{L_{dir}}{J_{dir}} \left( \frac{dL_{dir}}{dt} \right)_{in} = \frac{L_{dir}}{J_{dir}} \tau_{dir,in}. \quad (4.41)$$

### 4.3.2 Turbulence Model

Turbulent flow is irregular and chaotic, consisting of turbulent eddies in a broad range of length scales. The largest length scales are on the order of the flow geometry, and according to Kolmogorov's theory, energy from the larger eddies is transferred to progressively smaller eddies. The cascade process continues until energy at the smallest length scale dissipates due to viscous forces [96]. When observed velocity at a single point in space, turbulence results in fluctuations around the mean velocity. Assuming isentropic and homogeneous turbulence,

turbulent kinetic energy  $k$  is defined from the root-mean-square of the velocity fluctuations  $u'$ , such that

$$k = \frac{3}{2} u'^2 \quad (4.42)$$

Simplified versions of the  $k$ - $\varepsilon$  turbulence model have frequently been used in quasi-dimensional combustion models to determine the rate of change of turbulent kinetic energy  $k$  and dissipation rate  $\varepsilon$ . Turbulent kinetic energy and dissipation rate are averaged across the combustion chamber volume. The impact of combustion on turbulence is captured by a modifying factor in the burn rate model discussed later, assuming combustion does not affect turbulence in the unburned zone. Borgnakke *et al.* described the turbulent kinetic energy balance to have production  $P_k$ , diffusion  $F_{k,bound}$ , and dissipation  $\varepsilon$  terms [97]:

$$\frac{dk}{dt} = P_k + \sum \left[ \frac{\dot{m}_{bound}}{m_{cyl}} (F_{k,bound} - k) \right] - \varepsilon \quad (4.43)$$

The production term  $P_k$  represents the transfer of mean flow energy to turbulent kinetic energy  $P_{k,shr}$  and effects due to rapid changes in density  $P_{k,dens}$ . The diffusion term  $F_{k,bound}$  is treated as a boundary flux, where fluxes include intake flow  $F_{k,in}$  and fuel injection  $F_{k,inj}$ . The boundary mass flow rate  $\dot{m}_{bound}$  only includes flow entering the cylinder because  $k$  is defined on a per mass basis. The dissipation rate term  $\varepsilon$  represents the rate at which turbulent kinetic energy is destroyed. Like mean charge motion,  $k$  has little dependency on previous cycles and can be initialized as  $k = 0$ .

Like turbulent kinetic energy, the dissipation rate is affected by incoming flow, rapid changes in density, and mean flow. One equation  $k$ - $\varepsilon$  models estimate  $\varepsilon$  based on the integral length scale—the largest length scale. Two equation  $k$ - $\varepsilon$  models include an equation for the rate

of change of  $\varepsilon$  derived from the Navier-Stokes equations [98]. Taking a similar form as Eq. (4.43), the balance equation for  $\varepsilon$  can be represented [97]:

$$\frac{d\varepsilon}{dt} = P_\varepsilon + \sum \left[ \frac{\dot{m}_{bound}}{m_{cyl}} (F_{\varepsilon,bound} - \varepsilon) \right] - D_\varepsilon, \quad (4.44)$$

where  $P_\varepsilon$  is production,  $F_{\varepsilon,bound}$  is the boundary fluxes, and  $D_\varepsilon$  is the rate of destruction. Dissipation rate initial conditions depend initial  $k$ . By setting  $k = 0$  initially,  $\varepsilon = 0$  at start of simulation.

Along with turbulent kinetic energy, eddy length scales are important for predicting the burn rate. In the two-equation  $k$ - $\varepsilon$  model, the largest turbulent length scale, the integral length scale  $l_I$ , is defined as

$$l_I = C_\mu \frac{k^{\frac{3}{2}}}{\varepsilon}, \quad (4.45)$$

where  $C_\mu$  is an empirical constant typically suggested to be 0.09. At the smallest length scale, Kolmogorov length scale  $l_K$ , viscous stresses cause turbulent kinetic energy to dissipate. The Kolmogorov length scale is defined as

$$l_K = \left( \frac{\nu^3}{\varepsilon} \right)^{\frac{1}{4}}, \quad (4.46)$$

where  $\nu$  is the kinematic viscosity of the fluid. The Taylor microscale  $l_T$ , smaller than  $l_I$  and larger than  $l_K$ , is used to model the eddy burn up time [46], [47]. The Taylor microscale is defined as

$$l_T = \sqrt{\frac{15\nu}{\varepsilon}} u'. \quad (4.47)$$

Kinetic energy crossing the volume boundary contribute to turbulent kinetic energy and mean charge motion. By subtracting the rate of change of the mean kinetic energy due to intake flow (Eq. 4.41) from the total intake kinetic energy, the intake turbulent kinetic energy contribution  $F_{k,in}$  can be calculated as

$$F_{k,in} = \frac{1}{2} u_{in}^2 - \frac{1}{\dot{m}_{in}} \sum_{dir} \frac{L_{dir}}{J_{dir}} \tau_{dir,in} . \quad (4.48)$$

Fuel injection is assumed not to contribute to the mean charge motion. Therefore, all kinetic energy is represented as turbulent kinetic energy. The direct injection contribution  $F_{k,inj}$  is defined as

$$F_{k,inj} = \frac{1}{2} u_{inj}^2 , \quad (4.49)$$

where  $u_{inj}$  is the velocity through each injector hole. The effect of exhaust backflow on turbulence is neglected.

Intake and direct injection contributions to the dissipation rate are derived assuming the integral length scale of the incoming flow to be proportional to the flow dimension. Using intake valve lift  $L_v$  as the flow dimension and rearranging Eq. (4.45), the intake dissipation rate contribution  $F_{\varepsilon,in}$  is defined as

$$F_{\varepsilon,in} = C_{\mu} \frac{F_{k,in}^{\frac{3}{2}}}{C_{\varepsilon,valve} L_v} , \quad (4.50)$$

where  $C_{\varepsilon,valve}$  is a tuning constant. Similarly, the injection dissipation rate term  $F_{\varepsilon,inj}$  can be written as

$$F_{\varepsilon,inj} = C_{\mu} \frac{F_{k,inj}^{\frac{3}{2}}}{l_{l,inj}} , \quad (4.51)$$

where  $l_{l,inj}$  is the integral length scale of the incoming flow. Note that  $l_{l,inj}$  represents all injector holes and may not be easily determined from hole diameters but can be tuned.

Mean flow kinetic energy  $KE_{rot,dir}$  is converted to turbulent kinetic energy as a result of wall friction and fluid shear. From Eq. (4.40), turbulent kinetic energy production resulting from shear  $P_{k,shr}$  can be written as

$$P_{k,shr} = -\frac{1}{m_{cyl}} \sum_{dir} \frac{L_{dir}}{J_{dir}} \tau_{dir,shr} . \quad (4.52)$$

A standard  $k$ - $\varepsilon$  model relates turbulence production term  $P_{k,shr}$  to the rate change of the dissipation rate  $P_{\varepsilon,shr}$  by

$$P_{\varepsilon,shr} = C_{\varepsilon 1} \frac{\varepsilon}{k} P_{k,shr} , \quad (4.53)$$

where  $C_{\varepsilon 1}$  is an empirical constant typically suggested to equal 1.44. Similarly, the  $\varepsilon$  destruction rate  $D_{\varepsilon}$  is related to the  $\varepsilon$  by

$$D_{\varepsilon} = C_{\varepsilon 2} \frac{\varepsilon^2}{k} , \quad (4.54)$$

where  $C_{\varepsilon 2}$  is an empirical constant typically suggested to equal 1.92.

In addition to turbulence produced from the mean charge motion, Borgnakke *et al.* derived a turbulence production term to account for the effect of compression and expansion [97]. Using rapid distortion theory, turbulence produced by rapid changes in density  $P_{k,dens}$  is modeled as

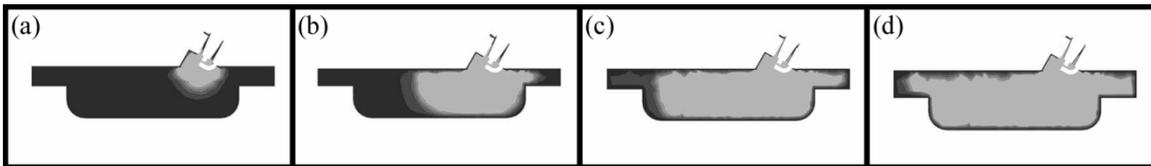
$$P_{k,dens} = \frac{2k}{3\rho_{cyl}} \frac{d\rho_{cyl}}{dt} . \quad (4.55)$$

Assuming turbulent angular momentum to be conserved, the product of  $u'$  and  $l_I$  remains constant during rapid distortion. Therefore, the following can be concluded [97]:

$$P_{\varepsilon,dens} = \frac{4\varepsilon}{3\rho_{cyl}} \frac{d\rho_{cyl}}{dt} \quad (4.56)$$

#### 4.4 Spark-Ignition Combustion Burn Rate

For a typical SI engine, air and fuel are uniformly mixed in the combustion chamber prior to combustion. After the intake valve closes, the piston begins to compress the mixture, and near the end of the compression stroke, the spark plug initiates combustion. The electric discharge between the spark plug electrodes creates a high-temperature plasma kernel [1]. The kernel then develops into a propagating flame front as shown in Figure 4.6(a). At the thin flame sheet, an exothermic chemical reaction occurs, and the unburned mixture is converted to combustion products or “burned gas.” Enclosed by the piston, cylinder wall, and cylinder head, a turbulent flame develops and spreads within the combustion chamber as shown in Figure 4.6(b) and (c). When the flame front reaches the chamber walls as shown in Figure 4.6(d), the flame can no longer propagate.



**Figure 4.6: Images of (a) early flame development, (b and c) flame propagation, and (d) flame termination (Adapted from E. Chan et al., 2010 [99])**

Several factors influence the rate in which the flame propagates. The reaction mechanism for combustion of a hydrocarbon evolves numerous intermediate reactions that dictate the flame speed. Modeling the reaction from a chemical kinetic perspective would be computationally

expensive and would still not necessarily provide an accurate prediction of cylinder pressure. Flame propagation speed also depends on flow within the combustion chamber. Increasing turbulence intensity during combustion increases the flame propagation speed, thus making piston speed, intake geometry, and chamber design important factors. Simplified models tuned with experimental data or CFD simulations must be used to represent complex combustion phenomenon when employing a 0D or quasi-dimensional model. Fitting the burn rate with a Wiebe function, which will be described in the next section, is one way to represent combustion without considering combustion chemistry and turbulence. Because fitting does not include physics-based representations of combustion, accuracy will suffer when operating outside the tuned range. A turbulence entrainment model, however, uses laminar flame speed and a turbulence model to predict the burn rate. Although an entrainment model requires tuning, the more predictive approach can better represent combustion outside the operating points used to tune the model.

#### 4.4.1 Wiebe Burn Rate

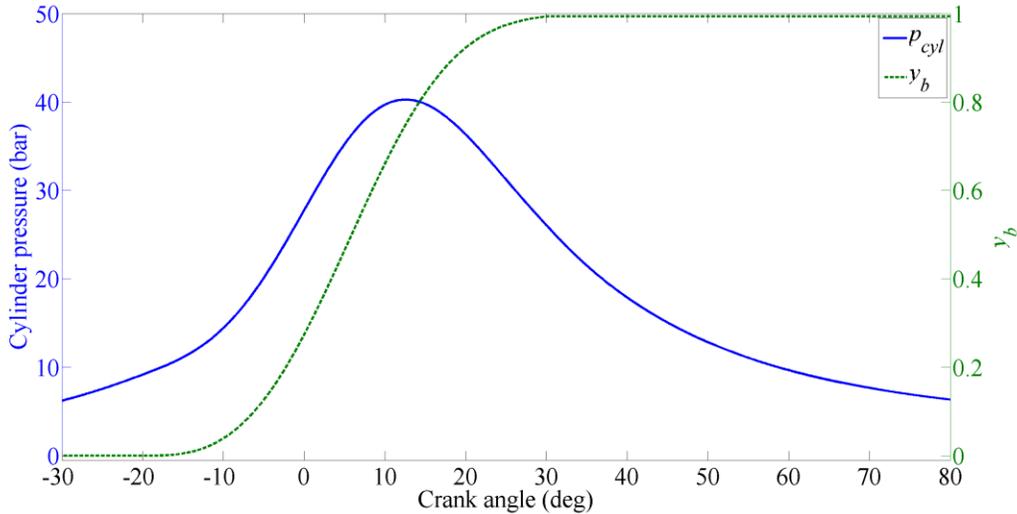
Rate of combustion has been observed to gradually increase at the start of combustion, rapidly increase during flame propagation, peak halfway through the combustion process, and rapidly decline during flame termination. Therefore, the mass fraction of burned gas  $y_{burn}$  creates an “s-shape” when plotted against crank angle (Figure 4.7). Based on experimental observations, combustion rate is frequently fit by a Wiebe function in the crank angle domain. The general Wiebe function is defined as

$$y_{burn} = 1 - \exp\left(-a\left(\frac{\theta_c - \theta_0}{\Delta\theta}\right)^{m+1}\right), \quad (4.57)$$

where  $\theta_c$  is the crank angle,  $\theta_0$  is the crank angle at the start of combustion,  $\Delta\theta$  is the combustion duration, and  $a$  and  $m$  are fitting parameters. Parameter  $m$  defines the shape of the mass fraction burned profile, while  $a$  models combustion efficiency. From Eq. (4.57), combustion efficiency  $\eta_{comb}$  can be defined as

$$\eta_{comb} = 1 - e^{-a} \quad (4.58)$$

Parameter  $a$  becomes 6.908 for an assumed efficiency of 99.9%. To tune the Wiebe function parameters, cylinder pressures are measured at different engine operating conditions, and the burn mass fraction profile can be derived by energy law analysis [1]. As a result  $a$ ,  $m$ , and  $\Delta\theta$  vary with engine speed and load.



**Figure 4.7: Cylinder pressure and Wiebe mass fraction burned profile**

#### 4.4.2 Turbulent Entrainment Model

The Wiebe mass burn profile provides an accurate estimation of cylinder pressure when simulating combustion at or near the tuning operating conditions. Because the actual burning velocity depends on turbulence and unburned gas composition, however, any changes in cylinder flow or mixture at the start of combustion will produce inaccuracy. For example, altering valve

timing affects the flow velocity into the cylinder, thus affecting turbulence and flame speed. Even spark timing has an effect on burn rate. Turbulence and flame structure vary with piston position due to changes in pressure, temperature, and cylinder volume, and by moving the start of combustion, the burn rate changes throughout the combustion process. To optimize engine parameters, a more predictive combustion model must be employed to ensure better accuracy over a wide range of operating conditions.

The turbulent entrainment model includes cylinder turbulence, geometry, and composition when predicting the burn rate [46], [47]. Depicted in Figure 4.8, the entrainment model represents combustion in two stages. In the first stage, the unburned mixture is entrained within the turbulent flame front but not consumed. In the second stage, the entrained mass is burned. The rate at which the unburned gasses are entrained by the flame is defined as

$$\dot{m}_e = A_f \rho_u S_f, \quad (4.59)$$

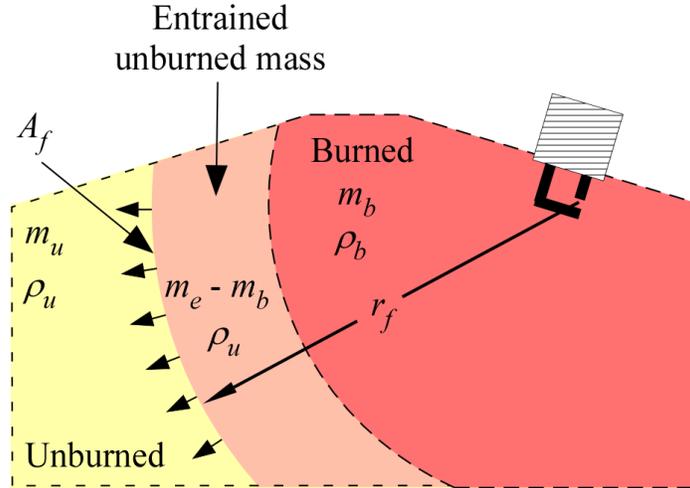
where  $m_e$  is the entrained mass,  $S_f$  is the turbulent flame speed, and  $A_f$  is the surface area between unburned and entrained zones. Once entrained, the ignition sites are stretched by turbulent eddies at the Taylor microscale  $l_T$  until the unburned gasses become fully engulfed by the flame. The mass burn rate  $\dot{m}_b$  is governed by the eddy burn up time  $\tau_b$  and unburned entrained mass:

$$\dot{m}_b = \frac{m_e - m_b}{\tau_b}. \quad (4.60)$$

Tabazynski used the Taylor microscale  $l_T$  to characterize the spacing between dissipation regions [46], [47]. By adding a tuning constant  $C_b$  to Tabazynski's model,  $\tau_b$  is defined as

$$\tau_b = C_b l_T / S_L, \quad (4.61)$$

where  $S_L$  is the laminar flame speed.



**Figure 4.8: Turbulent entrainment model zones with spherical flame front expanding from the spark location**

Tabazynski represented the flame speed  $S_f$  as the sum of the laminar flame speed  $S_L$  and the root-mean-square of velocity fluctuations  $u'$  [46], [47]:

$$S_f = S_L + u' \quad (4.62)$$

Several additions have been made by researchers to better match experimental observations.

Brehob and Newman modeled the flame speed with the following equation [100]:

$$S_f = S_L + C_f \sqrt{\frac{\rho_u}{\rho_b}} u' \left( 1 - e^{-\frac{r_f}{C_{dev} l_t}} \right), \quad (4.63)$$

where  $r_f$  is the flame radius and  $C_f$  and  $C_{dev}$  are tuning constants. During early combustion, turbulence has less impact on the flame speed. The exponential term in Eq. (4.63) models the transition from laminar to turbulent flame speed. Throughout combustion, the propagating flame impacts turbulence. Different approaches have been used to correct  $u'$  for rapid changes in density. The square root of the density ratio in Eq. (4.63) accounts for the effect combustion has on turbulence.

Area  $A_f$  can be determined from a 3D CAD model or generic representation of the chamber with the flame front approximated as sphere centered at the sparkplug. A lookup table for  $A_f$  can be generated by intersecting a sphere with the combustion chamber geometry (cylinder wall, piston, and cylinder head) at flame radius  $r_f$  and piston position  $x_p$  breakpoints. The volume inside the sphere  $V_e$  can be determined from the geometry as a function of flame radius  $r_f$  and piston position  $x_p$  as well. By integrating flame speed ( $\dot{r}_f = u' + S_L$ ),  $r_f$  can be estimated and then used to determine  $V_e$  and  $A_f$  from lookup tables. However, the model becomes over constrained when deriving  $r_f$  from the flame speed because  $V_e$ , a function of  $r_f$ , is defined by the two-zone model constraints. Thus, assuming the unburned entrained density to be  $\rho_u$ , the entrained volume  $V_e$  is defined as

$$V_e = V_b + \frac{m_e - m_b}{\rho_u}, \quad (4.64)$$

where the burned volume  $V_b$  is defined in Eq. (4.28). Now,  $V_e$  can be used as breakpoints for the  $A_f$  lookup table:

$$A_f = f(x_p, V_e). \quad (4.65)$$

#### 4.5 Laminar Flame Speed

Under laminar flow conditions, the reaction layer between the unburned and burned zones (flame front) during premixed combustion propagates in a controlled manner toward the unburned mixture. Based on the observation, premixed combustion is often characterized by the laminar flame speed  $S_L$ . Detailed reaction mechanisms can be used to predict  $S_L$ , as described in [101]. Chemical reaction simulations are computationally expensive and may not accurately

represent the burn rate. Therefore,  $S_L$  is typically measured experimentally and fit to a power law.

**Table 4.1: Laminar flame speed coefficients for methanol, propane, isooctane, and gasoline**

Fuel	$B_m$ (cm/s)	$B_\phi$ (cm/s)	$\phi_m$
Methanol [102]	36.92	-140.51	1.11
Propane [102]	34.22	-138.65	1.08
Isooctane [102]	26.32	-84.72	1.13
Gasoline [103]	30.5	-54.9	1.21

Several factors influence  $S_L$ , including pressure, unburned gas temperature  $T_u$ , equivalence ratio  $\phi$ , and diluent gas mass fraction  $y_{dil}$  (i.e. EGR or burned gas residuals). In order to determine  $S_L$  for various air-fuel mixtures, Metghalchi and Keck measured the burn rate over a large range of operating conditions and fit  $S_L$  to the equation [102]:

$$S_L = S_{L,ref} \left( \frac{T_u}{T_{ref}} \right)^\alpha \left( \frac{P}{P_{ref}} \right)^\beta (1 - 2.1y_{dil}), \quad (4.66)$$

where  $T_{ref} = 300$  K and  $P_{ref} = 101325$  Pa and  $\alpha$ ,  $\beta$ , and  $S_{L,ref}$  are fuel specific parameters that vary with  $\phi$ . The fit parameters are defined as:

$$\alpha = 2.18 - 0.8(\phi - 1), \quad (4.67)$$

$$\beta = -0.16 - 0.22(\phi - 1), \quad (4.68)$$

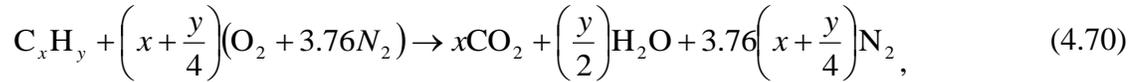
and

$$S_{L,ref} = B_m + B_\phi (\phi - \phi_m)^2, \quad (4.69)$$

where  $B_m$ ,  $B_\phi$ , and  $\phi_m$  are fuel specific fitting parameters summarized in Table 4.1 for various fuels.

#### 4.6 Combustion Chemistry

To model the combustion reaction, burned gas species must be determined from the unburned mixture, stoichiometric constraints, and chemical equilibrium. Stoichiometric combustion of an arbitrary hydrocarbon fuel  $C_xH_y$  in air can be expressed as



where  $(O_2 + 3.76N_2)$  is a simplified model of air. From Eq. (4.70), the stoichiometric air-fuel mass ratio  $AFR_s$  can be defined:

$$AFR_s = 4.76\left(x + \frac{y}{4}\right)\frac{MW_{air}}{MW_{fuel}} \quad (4.71)$$

The actual air-fuel ratio  $AFR_{actual}$  is greater than  $AFR_s$  during fuel lean combustion and less than  $AFR_s$  during rich combustion. Because  $AFR_s$  is fuel dependent, combustion conditions (lean, rich, and stoichiometric) are often indicated by the equivalence ratio  $\phi$ , defined as

$$\phi = \frac{AFR_s}{AFR_{actual}}. \quad (4.72)$$

From Eq. (4.70), the quantities of major burned gas species  $CO_2$ ,  $H_2O$ , and  $N_2$  can be determined for stoichiometric combustion or extended to fuel lean conditions ( $\phi < 1$ ) by including excess  $O_2$ . However, rich combustion ( $\phi > 1$ ) produces a significant amount of  $CO$  and  $H_2$ , requiring an additional relationship. Because the chemical reaction occurs at a much faster rate than the change in cylinder pressure and temperature, chemical equilibrium is typically applied to 0D combustion models for predicting minor burned gas species. A full equilibrium

model can include several minor species (e.g. OH, NO, CO, and H<sub>2</sub>), but to reduce computation time with little cost to accuracy, a single equilibrium reaction is considered: the water-gas shift reaction. The water-gas shift equilibrium reaction is given as



The balance of species on each side of the equilibrium reaction equation, Eq. (4.73), can be determined from the second law of thermodynamics. By assuming cylinder internal energy  $U$ , mass  $m$ , and volume  $V$  to be quasi-constant (which implies no heat transfer), the second law provides the constraint

$$(dS)_{U,V,m} = 0, \quad (4.74)$$

where  $dS$  is the net change in entropy at the given instant [75]. Gibbs free energy  $G$  is a more convenient property to utilize for chemical equilibrium calculations. Gibbs free energy is defined as

$$G = H - TS, \quad (4.75)$$

and assuming quasi-steady conditions, the second law of thermodynamics states

$$dG_{mix} = 0, \quad (4.76)$$

where  $dG_{mix}$  is the gas mixture net change in Gibbs free energy.

The  $i^{\text{th}}$  species molar specific Gibbs free energy  $\tilde{g}_i$  can be determined from the species partial pressure  $P_i$ , temperature  $T$ , and tabulated molar specific Gibbs free energy  $\tilde{g}_i^\circ$  taken at a standard pressure  $P^\circ$ . For an ideal gas,  $\tilde{g}_i$  is defined as

$$\tilde{g}_i = \tilde{g}_i^\circ + R_{un} T \ln(P_i / P^\circ), \quad (4.77)$$

where  $R_{un}$  is the universal gas constant. Applying the second law constraint (Eq. (4.76)) to the water-gas shift reaction (Eq. (4.73)) results in

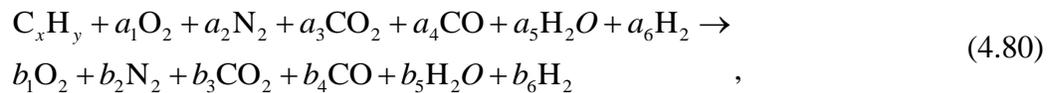
$$-\left(\tilde{g}_{CO}^{\circ} + R_{un}T \ln(P_{CO}/P^{\circ})\right) - \left(\tilde{g}_{H_2O}^{\circ} + R_{un}T \ln(P_{H_2O}/P^{\circ})\right) + \left(\tilde{g}_{CO_2}^{\circ} + R_{un}T \ln(P_{CO_2}/P^{\circ})\right) + \left(\tilde{g}_{H_2}^{\circ} + R_{un}T \ln(P_{H_2}/P^{\circ})\right) = 0 \quad (4.78)$$

Eq. (4.78) can be rearranged to relate species concentrations to an evaluable parameter: current temperature is known, or at least can be determined iteratively based on the mixture composition, and the  $\tilde{g}_i^{\circ}$  terms are tabulated for each species as a function of temperature. The partial pressure terms, which relate the species concentrations, is defined as the equilibrium constant  $K_p$ . For the water-gas shift reaction,  $K_p$  is defined as

$$K_p = \exp\left(-\frac{\tilde{g}_{CO_2}^{\circ} + \tilde{g}_{H_2}^{\circ} - \tilde{g}_{CO}^{\circ} - \tilde{g}_{H_2O}^{\circ}}{R_u T_b}\right) = \frac{(P_{CO_2}/P^{\circ})^1 (P_{H_2}/P^{\circ})^1}{(P_{CO}/P^{\circ})^1 (P_{H_2O}/P^{\circ})^1} \quad (4.79)$$

For  $K_p > 1$ , the reaction will produce higher concentrations of  $CO_2$  and  $H_2$  at equilibrium, and for  $K_p < 1$  the reaction will produce higher concentrations of  $CO$  and  $H_2O$  at equilibrium.

By adding water-gas equilibrium (Eq. (4.73)) to the combustion reaction and allowing burned gas residuals to be present in the reactants, the combustion of hydrocarbon fuel  $C_xH_y$  can now be represented in general by



where  $a$  reactant coefficients are determined from the unburned zone mass fractions and  $b$  coefficients are derived from the element balances and equilibrium constraints. Balancing elements N, C, O, and H provide four equations:

$$b_2 = a_2 \quad (4.81a)$$

$$b_3 = x + a_3 + a_4 - b_4, \quad (4.81b)$$

$$b_5 = 2a_1 + 2a_3 - b_4, \quad (4.81c)$$

and

$$b_6 = \frac{y}{2} + a_5 + a_6 - b_5. \quad (4.81d)$$

The final constraint is provided by equilibrium. From the definition of the equilibrium constant  $K_p$  in Eq. (4.79), the water-gas reaction provides the constraint

$$K_p = \frac{P_{CO_2} P_{H_2}}{P_{CO} P_{H_2O}} = \frac{b_3 b_6}{b_4 b_5}, \quad (4.82)$$

By substituting Eq. (4.81) into Eq. (4.82),  $b_4$  can be determined using the quadratic equation:

$$b_4 = \frac{-c_2 + \sqrt{c_2^2 - 4c_1 c_3}}{2c_1}, \quad (4.83a)$$

$$c_1 = K_p - 1, \quad (4.83b)$$

$$c_2 = 2(1 - K_p)(b_1 - a_1) + (2 - K_p)a_4 + a_3 + a_6 + (3 - 2K_p)x + y/2 + K_p a_5, \quad (4.83c)$$

and

$$c_3 = -(a_4 + a_3 + x)(2b_1 + a_4 + a_6 - 2a_1 + 2x + y/2). \quad (4.83d)$$

Certain considerations need to be taken when solving for  $b_4$ . For lean or stoichiometric combustion without CO and H<sub>2</sub> in the reactants,  $b_4$  will be less than 0 according to Eq. (4.83). Therefore, coefficient  $b_4$  is subject to the constraint  $b_4 \geq 0$ . Also note that the denominator in Eq. (4.83) is zero when  $K_p = 1$ . To avoid dividing by values near 0,  $b_4$  is calculated at  $K_p = 0.99$  and  $K_p = 1.01$  and interpolated based on the actual  $K_p$  between 0.99 and 1.01.

In summary, the reactant coefficients are determined from the unburned mass fractions and species molecular weights. Using the reactant coefficients, the CO product coefficient  $b_4$  is calculated by Eq. (4.83), subject to  $b_4 \geq 0$ . The remaining product coefficients are determined

from the element balances (Eq. (4.81)). Finally, product mass fractions are calculated from the coefficients.

#### 4.7 Heat Transfer

During combustion and expansion, heat loss due to convection heat transfer lowers cylinder pressure and therefore usable power. cylinder wall, cylinder head, and piston, are modeled at separate temperatures  $T_{wall}$ ,  $T_{head}$ , and  $T_{piston}$ . For a single-zone, heat transfer  $\dot{Q}_w$  from the chamber walls to the control volume is defined as

$$\dot{Q}_w = h_c [A_{wall}(T_{wall} - T_{cyl}) + A_{head}(T_{head} - T_{cyl}) + A_{piston}(T_{piston} - T_{cyl})], \quad (4.84)$$

where  $h_c$  is the heat transfer coefficient and  $A_{wall}$ ,  $A_{head}$ , and  $A_{piston}$  are the cylinder wall, head, and piston surface areas exposed to the control volume, respectively. Areas  $A_{head}$  and  $A_{piston}$  remain constant while  $A_{wall}$  varies with  $x_p$ :

$$A_{wall} = x_p \pi D_{bore}, \quad (4.85)$$

where  $D_{bore}$  is the cylinder bore diameter. For a two-zone combustion model, heat transfer is defined for both zones: unburned  $\dot{Q}_u$  and burned  $\dot{Q}_b$  heat transfer. Similar to Eq. (4.84), convection heat transfer for the zones are defined as

$$\dot{Q}_u = h_{c,u} [A_{wall,u}(T_{wall} - T_u) + A_{head,u}(T_{head} - T_u) + A_{piston,u}(T_{piston} - T_u)], \quad (4.86)$$

and

$$\dot{Q}_b = h_{c,b} [A_{wall,b}(T_{wall} - T_b) + A_{head,b}(T_{head} - T_b) + A_{piston,b}(T_{piston} - T_b)]. \quad (4.87)$$

where the surface areas are defined for each zone. Assuming a spherical flame front, each surface area is a function of flame radius and piston displacement.

Flow velocity influences the convection heat transfer coefficient  $h_c$  but is not explicitly available from the cylinder model. To predict cylinder convection heat transfer, Woschni developed a convection correlation based on a spatially averaged cylinder velocity [104]. Woschni suggested that the average gas velocity  $w$  in the cylinder should be proportional to the mean piston speed  $\bar{S}_p$ , and to account for changes in velocity during combustion and expansion, Woschni included the motoring pressure  $p_m$  in the correlation. Assuming a polytropic expansion of an ideal gas, the motoring pressure can be estimated from

$$p_m = p_r \left( \frac{V_r}{V_{cyl}} \right)^\gamma, \quad (4.88)$$

where  $p_r$  and  $V_r$  are the reference pressure and volume taken at the start of combustion. Derived from experimental testing, Woschni published the following correlation for predicting the mean cylinder gas velocity:

$$w = C_1 \bar{S}_p + C_2 \frac{V_{cyl} T_r}{p_r V_r} (p_{cyl} - p_m). \quad (4.89)$$

The correlation coefficients  $C_1$  and  $C_2$  given in Table 4.2 are adjusted based on specific engine conditions. Valves are open during gas exchange and closed during compression and expansion. As a result, the period can be determined by the flow rate through the valves and the piston speed.

With the mean velocity  $w$ , Woschni correlated the cylinder Nusselt number to the Reynolds number using the power-law relationship

$$\text{Nu} = C_h \text{Re}^{a_h}, \quad (4.90)$$

where  $C_h$  is a tunable constant and  $a_h = 0.8$ . Woschni used  $C_h = 0.035$ , but the value could be tuned to fit experimental data. With  $D_{bore}$  as the characteristic length, the heat transfer coefficient can be derived from Eq. (4.90) as

$$h_c = C_h k D_{bore}^{-0.2} w^{0.8} \mu^{-0.8} \rho^{0.8} \quad (4.91)$$

where  $\rho$  is the density,  $k$  is the thermal conductivity, and  $\mu$  is the dynamic viscosity of the gas. For the single-zone model,  $\rho$  is the cylinder density  $\rho_{cyl}$  and fluid properties are evaluated at the cylinder temperature  $T_{cyl}$ . For the two-zone model, density and temperatures are taken from the respective zones. Therefore,  $h_{c,u} \neq h_{c,b}$  due to the differences in density and fluid properties.

**Table 4.2: Coefficients for Woschni heat transfer correlation [104][103]**

Period	Criteria	$C_1$	$C_2$
Gas exchange	$\sum_{k=1}^N \dot{m}_k \neq 0$	6.18	0
Compression	$\dot{x}_p < 0$	2.28	0
Combustion and expansion	$\dot{x}_p \geq 0$	2.28	3.24e-3

## CHAPTER 5

### SIMULINK ENGINE MODEL

Among the simulation and control design software packages available, MATLAB/Simulink is widely used in the automotive industry and academia. Simulink contains powerful tools for modeling physical systems and designing controllers. However, Simulink-based high-fidelity engine modeling packages currently do not exist, and a third-party software package specialized in 1D high-fidelity engine models must be connected to Simulink in order to leverage MATLAB/Simulink's tools. Introducing a third-party software requires additional time and limits flexibility, thus motivating a Simulink-based toolbox. The proposed modeling architecture allows engine models to be built rapidly in the Simulink environment, enables the use of MATLAB's powerful tools, and allows custom models to be employed.

#### 5.1 Simulink Architecture Overview and User Interface

The goal of the Simulink-based architecture is to allow users to build engine models from a library of components that connect in a physically representative manner and communicate with existing Simulink blocks. Therefore, a user can intuitively connect flow components (junctions, boundaries, valves, throttle, and 1D sections), cylinders, and crank dynamics according to the engine design and then use the model to test controllers, optimize parameters, and predict engine performance. As a secondary goal, the framework allows new models to be easily implemented without over constraining the developer. In order to reduce initialization time and maintain flexibility, simulation is executed from a single S-function. Block connection, S-

function connection, and user interface of the Simulink-based engine modeling architecture are discussed in this chapter.

### 5.1.1 Block Connection

Engine component models are represented by masked Simulink blocks. In order to intuitively configure an engine model, individual component blocks must be connected in a physically representative manner. Therefore, Simulink blocks that represent engine components must communicate variables without requiring the user to specify each variable independently. Additionally, blocks must be able to connect to standard Simulink ports for interfacing with existing Simulink blocks. For example, referring to Figure 5.1, the 1D flow section model must be able to pass cell information to the valve model while receiving flow information from the valve. Similarly, the cylinder-to-valve and cylinder-to-dynamics connections require an exchange of information. The dynamics model then provides a crank angle and speed to Simulink and receives a load torque. All other component blocks communicate in a similar fashion: two-way communication for a physical connection and standard Simulink input/output ports for communicating with existing Simulink blocks.

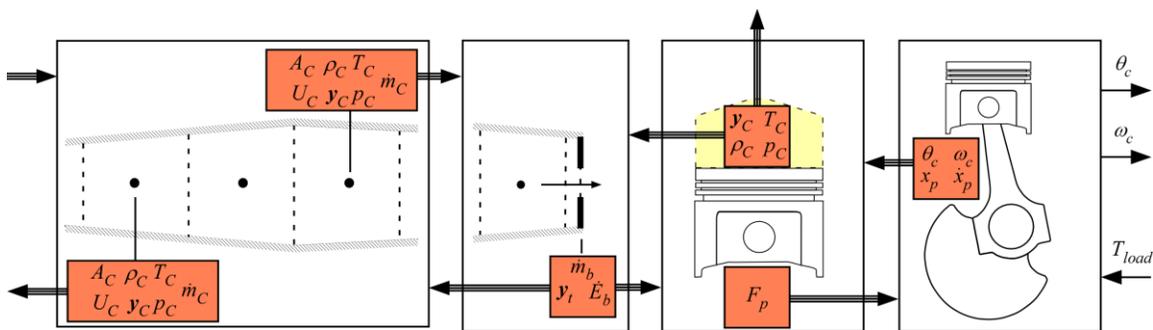
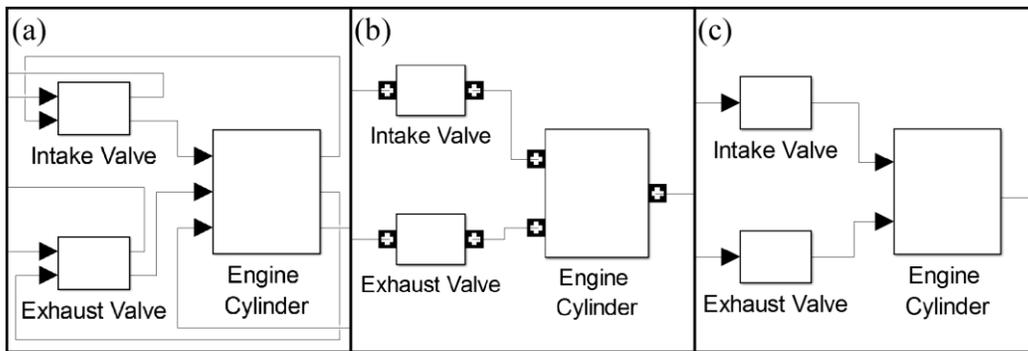


Figure 5.1: Communication of system variables through block connections

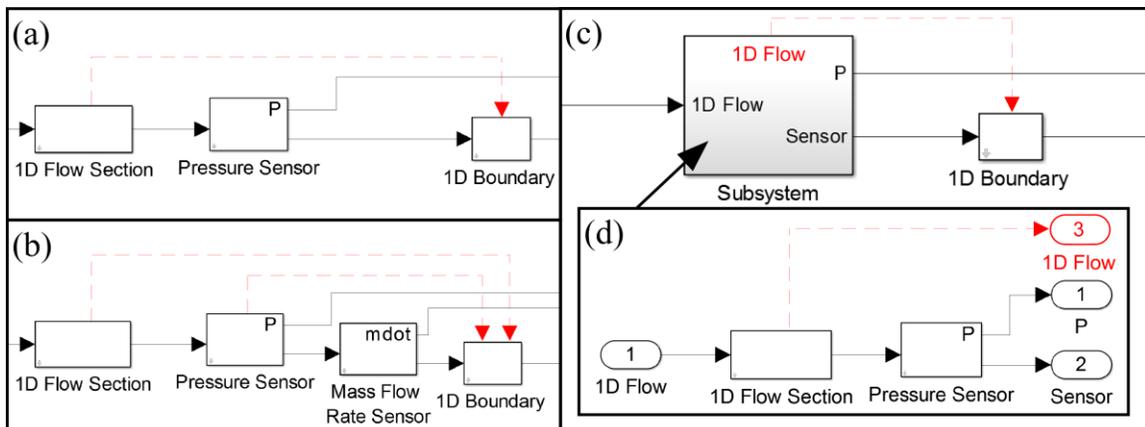
Creating physical connections with standard Simulink input/output ports would be cumbersome and lack the appearance of a physical connection as shown in Figure 5.2(a). To work round the issue, a two-way connector can be introduced (Figure 5.2(b)). A two-way connector, as the name suggests, acts as an input and output. Therefore, inter-block communication is achieved with a single connection, but the connection loses component directionality and requires an elaborate signal routing method [105]. To improve flexibility and usability, physical connections are represented by a single Simulink input/output connection as shown in Figure 5.2(c). Because standard Simulink ports only allow parameters to be passed in a single direction, the exchange of information between connected components is handled by the S-function, requiring block connectivity to be detected at the start of simulation and passed to the S-function. Discussed in the next section, input and output signals are routed to the S-function through “Goto” and “From” tags.



**Figure 5.2: Possible component block connection methodologies: (a) standard input/output, (b) two-way connector, and (c) single input/output connection**

To measure component parameters (e.g. pressure, temperature, mass fractions, and mass flow rate), virtual sensors can be added to the model. The ideal virtual sensors provide desired information to the Simulink environment without affecting the connected component. Flow sensors connected in series break the direct connection between 1D flow blocks, and therefore,

parameters from the adjacent blocks must be passed through the sensor. The special configuration creates what will be referred to as a “through port.” Through ports are essentially a virtual connection between blocks on either side of the sensor. Several example sensor connections are shown in Figure 5.3 where the dashed-line represents the virtual connection created by the through port. Shown in Figure 5.3(a), the “Pressure Sensor” block connects to the “1D Flow Section” and “1D Boundary” blocks, and the through port produces a virtual connection between the “1D Flow Section” and “1D Boundary.” Sensors connected in series allow communication between each component within the through port series (Figure 5.3(b)). As shown in Figure 5.3 (c) and (d), the scheme also applies to subsystems.



**Figure 5.3: Example sensor configurations create virtual connections (dashed lines): (a) pressure sensor, (b) pressure and mass flow rate sensor connected in series, (c) flow subsystem with pressure sensor, and (d) internal view of flow subsystem**

### 5.1.2 S-function Interface

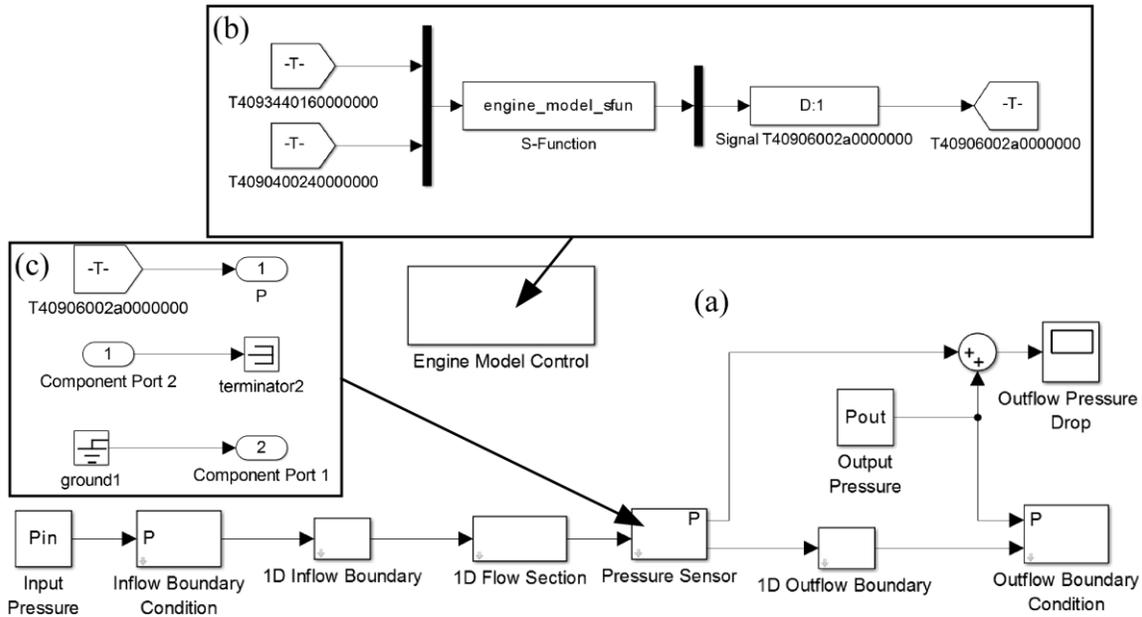
The nature of the engine component models does not permit equations to be efficiently formed from built-in Simulink blocks. Furthermore, MATLAB is an interpreted language, and any interpreted code executed during simulation would dramatically impact simulation time. Therefore, simulation is executed from a single compiled S-function block. An S-function interacts with Simulink engine and can be written in a several computer languages, such as C,

C++, and FORTRAN [106]. During initialization and simulation, the Simulink engine calls situation-specific S-function routines, and special syntax allows the S-function to interact with ODE solvers. Instead of having an S-function for each component block, implementing the entire model in a single S-function adds flexibility to the modeling structure.

At the start of simulation, all engine component blocks and connections in the root block diagram are located, recorded, and supplied to the global S-function. For example, the block diagram in Figure 5.4(a) contains: “Inflow Boundary Condition,” “1D Inflow Boundary,” “1D Flow Section,” “Outflow Boundary Condition,” “1D Outflow Boundary,” and “Pressure Sensor.” At the start of simulation, component blocks are located using a mask parameter tag, and connections are recorded. The S-function (Figure 5.4(b)), located in the “Engine Model Control” block, then assembles the model based on the Simulink block connections and then executes component specific code during simulation. Essentially, component blocks act as a graphical interface between the user and the model S-function: the user enters parameters into a dialog box and makes physically representative connections to other component blocks without interacting with the S-function directly.

Component blocks with external inputs (e.g. boundary pressure) and component blocks with outputs (e.g. pressure sensor) communicate with the S-function using global “Goto” and “From” tags provided in Simulink’s standard library. Setup, placement, and naming of the tags are automated, and as a result, the user never interacts with the S-function directly. For example, the “Pressure Sensor” block shown in Figure 5.4(a) has two component ports—ports that connect to other engine components—and a pressure output port “P” that connects to Simulink blocks. Because component ports represent a physical connection and do not carry data, inputs are terminated and outputs are connected to ground as shown in Figure 5.4(c). However, model

inputs and outputs communicate with S-function. As shown in Figure 5.4(c), the “Pressure Sensor” subsystem has a “From” tag that connects to the output port “P,” and the matching “Goto” tag is connected to the S-function block (Figure 5.4(b)). In order to route every component input and output, a "Demux" block divides S-function outputs, and a "Mux" block creates a single S-function input vector as shown in Figure 5.4(b). Using this connection method, S-function input and output dimensions vary based on the number of blocks requiring inputs or outputs and individual input/output dimensions.

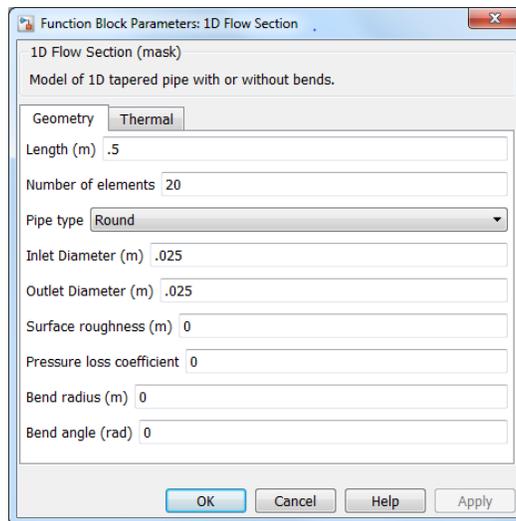


**Figure 5.4:** (a) Example of a 1D flow model with pressure boundary conditions and sensor, (b) view of “Engine Model Control” subsystem, and (c) view of “Pressure Sensor” subsystem

### 5.1.3 Block Interface

Like built-in Simulink blocks, the user enters parameters and sets model options in the mask dialog box which can be customized in Simulink’s mask editor. The “1D Flow Section” dialog box, for example, is shown in Figure 5.5. Parameters entered into edit boxes can be numerical, workspace variables, or evaluable Matlab commands that result in scalar or array

values. Parameters and options can also be programmatically set from Matlab before running the simulation, which is useful for parameter optimization. Once simulation starts, parameter inputs are disabled to prevent the user from modifying values during simulation. Changing physical parameters after initialization can be very difficult to implement and adds little value to the framework. For example, changing the number of elements in a flow section requires memory to be reallocated and connections to be reestablished. If a parameter needs to be dynamically controlled, an input port can be used. The framework allows ports to be added during setup based on the user requirements. Cam phasing can be set as an external input or a dialog parameter, for example.



**Figure 5.5: Mask dialog box for “1D Flow Section” block**

In order to check parameters and set up ports, every engine library block has an associated class. Masked Simulink blocks call an initialization callback function “MaskInitialization” after applying parameter changes. During mask initialization, the “Initialize” method in the engine model class is called to setup component ports (connection between engine blocks), setup input/output ports (connection to Simulink blocks), check parameter values, and save parameters that will be passed to the S-Function. Underlying

communication tags and ports (e.g. Figure 5.4(c)) are added or deleted automatically based on the user’s requirements. Tasks specific to start of simulation are called in the engine component class “Start” method. Final parameter and connection checks can be executed in “Start.” Default parameters can be applied as well. For example, the user can specify “auto” for a boundary area, and the connected pipe area is taken as the default value. If a boundary is placed between flow sections with different areas, the smaller area is chosen by default. The purposes of “Initialize” and “Start” methods are summarized in Table 5.1.

**Table 5.1: Engine model component class methods**

<b>Method</b>	<b>Purpose</b>
Initialize	<ul style="list-style-type: none"> <li>• Check parameter limits</li> <li>• Check parameter array size</li> <li>• Set up component ports</li> <li>• Set up Simulink input/output ports</li> <li>• Save parameter value to component object</li> </ul>
Start	<ul style="list-style-type: none"> <li>• Set default values (e.g. set boundary area based on connected 1D sections if user specifies “auto”)</li> <li>• Final parameter check if needed</li> <li>• Final block connection check if needed</li> </ul>

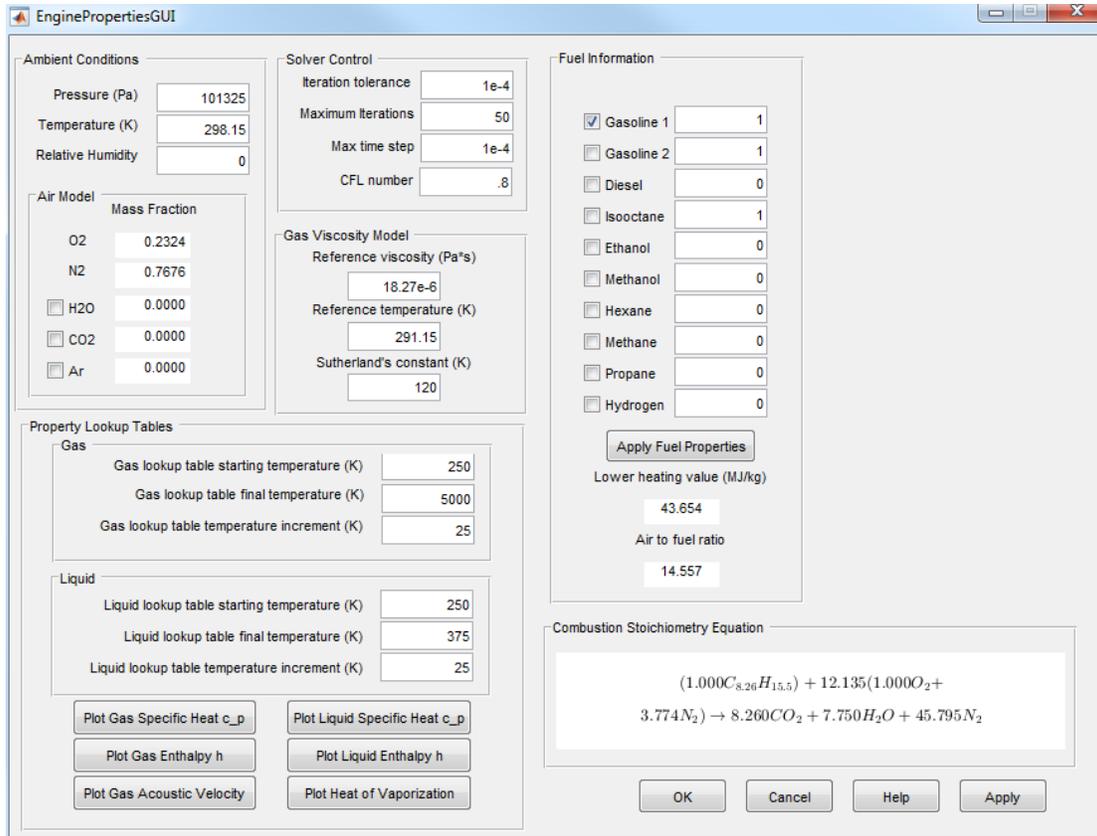
The engine component class also defines block connectivity. Class properties summarized in Table 5.2 are used to check connections after a port connection event and at the start of simulation. For an input or output connection event, the newly connected block properties, “name” and “types”, are compared to “conn\_names” and “conn\_types” list. If the component “name” or one of the “types” of the connected component does not match, an error is issued. The required connections defined by “required\_names” and “required\_types” are checked at the start of simulation. If any component has an unconnected port or any required connection criteria is not met, an error is issued. Block specific checks not covered by the name and type requirements can be placed in the “Start” method.

**Table 5.2: Engine model component class properties**

<b>Property</b>	<b>Comments</b>
name	Unique name of component
types	Categories or types of component model (e.g. “Volume0D,” “Flow1DBoundary,” “Flow1DElement,” and “EngineCylinder”)
conn_types	Types of components that can be connected
conn_names	Names of components that can be connected but not covered by “conn_type”
required_types	Types of blocks required to be connected at the start of simulation (e.g. cylinder must be connected to dynamics block)
required_names	Names of blocks required to be connected at the start of simulation but not covered by "required_types”

#### **5.1.4 Model Control Interface**

In addition to physical parameters entered into each component dialog box, model-wide, or global, parameters must be set as well. Global parameters and settings are controlled from the engine Graphical User Interface (GUI) as shown in Figure 5.6. In the GUI, the user can specify temperature ranges and increments for gas and liquid fuel properties. Thermodynamic properties can then be plotted for verification. Fuels or fuel mixtures can be selected, and in turn, the GUI provides the lower heating value, air/fuel ratio, and stoichiometry combustion equation based on the fuel selection. Nonlinear solver tolerance and maximum number of iterations, maximum time step, and CFL number are also set in the “Solver Control” group box. Finally, the user can specify ambient condition parameters: pressure, temperature, humidity, and gas species in air. Parameters set from the GUI are passed to the S-function and can be accessed by each component model.



**Figure 5.6: Global model control graphical user interface**

## 5.2 S-Function

S-functions which can be written in Matlab, C, C++ or FORTRAN languages interact with the Simulink engine through an Application Programming Interface (API) [106]. Using the S-function API, external or third-party software can communicate with Simulink, and complex models that cannot be easily or adequately represented with built-in Simulink blocks can be realized. During initialization and simulation, S-function methods are called by the Simulink engine based on the execution order, events, and S-function settings. An S-function can have continuous states integrated by the Simulink ODE solver and/or discrete states that are updated internally each time step. Execution of an S-function method varies between continuous, discrete, and hybrid systems. The S-function can also dictate the maximum allowed time step when a variable time step solver is selected.

### 5.2.1 Model Representation in S-Function

The engine model S-function, which is written in C++, produces an output vector  $\mathbf{y}$  based on the input vector  $\mathbf{u}$  and states  $\mathbf{x}$ . In general, the engine model can be described by

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \quad (5.1a)$$

and

$$\mathbf{y} = g(\mathbf{x}, \mathbf{u}), \quad (5.1b)$$

where  $f$  is the system function and  $g$  is the output function. Because the system is continuous, state derivatives can be integrated by Simulink's ODE solver (continuous S-function). However, to have control over integration and engine model time step, integration is performed internally, making the S-function discrete. The engine model S-function can then be described by

$$\mathbf{x}_{k+1} = h(\mathbf{x}_k, \mathbf{u}_k) \quad (5.2a)$$

and

$$\mathbf{y}_k = g(\mathbf{x}_k, \mathbf{u}_k). \quad (5.2b)$$

By making the S-function discrete, the order at which component models are solved can be controlled and solvers specific to compressible flow can be utilized.

### 5.2.2 Engine Component C++ class

To handle S-function setup, Simulink communication, and ODE integration, each engine component block has an associated C++ class that inherits methods and properties from the "EngineModelComponent" class. The "EngineModelComponent" superclass contains virtual methods specific to each component and non-virtual methods and properties for communicating with Simulink. During "mdlStart," class instances are created for each component in the Simulink model and block names are assigned to each instance. Tracking the block names helps with debugging; errors issued by a subclass can display the block name in Simulink during setup

or simulation. For example, if the cylinder pressure falls below zero, an error can be issued by the class instance and the full block name will be displayed. Once created, virtual methods of the superclass “EngineModelComponent” summarized in Table 5.3 are called for each component in the Simulink block diagram, allowing component-specific code to be executed in a predefined order. Note that the C++ class has two output methods (“MdotOutputs” and “Outputs”) and two derivative methods (“MdotDerivatives” and “Derivatives”). The output methods define Eq. (5.2b) and the derivative methods define Eq. (5.2a). One output-derivative pair is used to solve 1D boundary ODE’s and the other pair for all other ODE’s. The structure provides flexibility for model development while handling communication with Simulink outside the component-specific code. By understanding the purpose of each class method and execution order, complex component models can be represented in Simulink.

### **5.3 Simulation**

The engine model framework interfaces between Simulink and the compiled S-function. The user connects components blocks and enters parameters into the dialog box based on the engine design. Then, physical parameters, design options, and block connection information are passed to S-function. Therefore, initial setup is handled in Matlab while simulation is executed in the S-function. Simulation setup, S-function execution order, and internal framework are presented in the following sections.

**Table 5.3: “EngineModelComponent” C++ class methods**

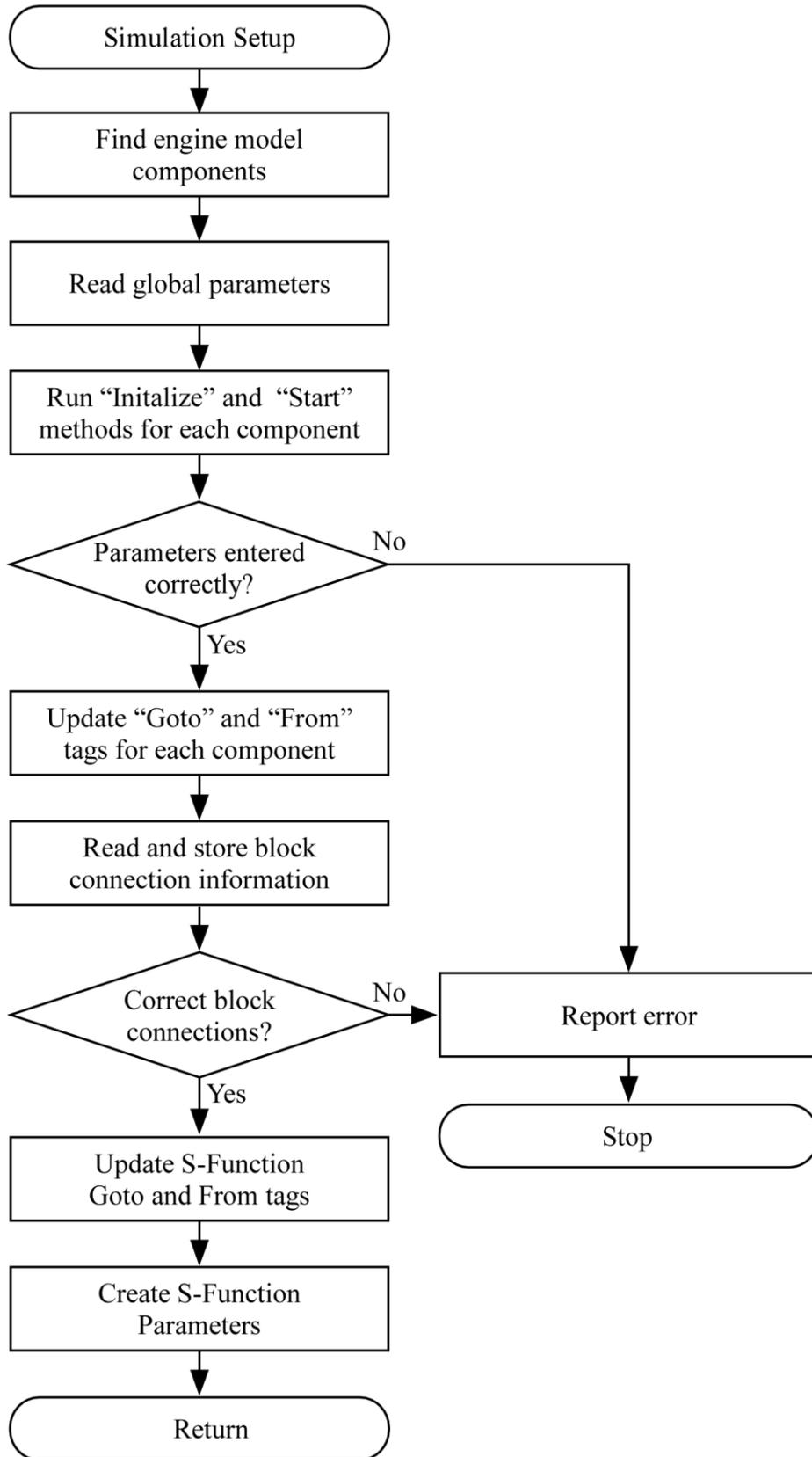
	<b>Method</b>	<b>Comments</b>
Setup	SetParameters	Set object parameter values entered into Simulink masked dialog box
	SetThroughPorts	Designate ports where all connection variables are passed through to other component (typically used for sensor blocks)
	SetThroughVariables	Designate individual variables to be passed through a component (e.g. relay crank angle through cylinder model to valve model)
	InitializeSizes	Set array lengths based on input parameters and port connections
	AddInput	Create pointer to Simulink input variable
	AddOutput	Create pointer to Simulink output variable
	AddPortVariables	Set variables required from and available to connected components
	InitializePointers	Set pointers (states, state derivatives, and any other pointer variable)
	Start	Run any required setup
	InitializeConditions	Set initial state values
Simulation	MaxTimeStep	Calculate maximum time step allowed for component (only used by flow components)
	NewTimeStep	Ran at the start of each major time step (typically used to reset state values)
	MdotOutputs	Update variables derived from the mass flow rate (used by flow components only)
	MdotDerivatives	Calculate rate change in mass flow rate (used by flow components only)
	Outputs	Use states (not mass flow rate) to update parameters
	Derivatives	Calculate derivatives that are not mass flow rate
Terminate	StopSim	Save information and clear memory if necessary

### 5.3.1 Simulation Setup

Referring to the simulation setup flowchart in Figure 5.7, several steps need to be taken in order to pass Simulink data to the S-function. The first step is to find all the engine model

components in the Simulink diagram. Each block can be located by a common read-only parameter that can be found by the “find\_system” Matlab function. After locating the blocks, global parameters (thermodynamic property tables, fuel information, and ambient conditions) are generated from values entered into the model control GUI. Then, “Initialize” and “Start” methods are ran for each block for final parameter and connection checks. If the values entered into the dialog box cannot be evaluated or are incorrect during the “Initialize” call, an error is reported and simulation stops. The “Start” method allows special implementations and final checks. Global parameters are passed to “Start,” allowing boundary or initial conditions to be equated to ambient values. As mentioned previously, block connections can also be checked during “Start.” Connection checks not specific to one component are evaluated later.

If none of the component blocks issue an error during “Initialize” or “Start” method calls, “Goto” and “From” tags are updated for each component block. Because the tags are global, the tag name must be unique and have a very low probability to be repeated elsewhere in the model by the user. Therefore, words or common variable names must be avoided. To ensure uniqueness, the tag handle is used. Simulink assigns a unique handle to every block that can be read as a numerical value. Tag names are set as “T” followed by the block handle converted to hexadecimal (e.g. “T4072a00600000000”). Tag names, corresponding port numbers, and component names are recorded for communicating with the S-function.



**Figure 5.7: Simulation setup flowchart**

After component tag assignment, general block connections are checked: all component ports must be connected, and required connections defined by properties “required\_types” and “required\_names” discussed earlier must be satisfied. If checks pass, S-function inputs and outputs are updated. Input “Mux” and output “Demux” blocks are resized according to the number of inputs and outputs contained in the model. Input “From” tags are attached to the “Mux” block and named accordingly. To allow vector outputs, “Demux” outputs are routed through “Signal Specification” blocks which dictate the output vector length to the “Goto” tags. “Goto” tags are then named. Finally, global parameters, component parameters, connection information, and input/output information are stored and passed to the S-function.

### 5.3.2 S-Function Method Execution

Several steps must be taken to initialize and simulate Eq (5.2) in the S-function. Referring to the flowchart in Figure 5.8, parameters and connection information determined in Matlab are passed to the S-function at the start of simulation. Once in the S-function, two required API functions are called: “mdlInitializeSizes” and “mdlInitializeSampleTimes.” The length of the input and output vectors are set in “mdlInitializeSizes,” and the type of system (discrete with variable time step) is set in “mdlInitializeSampleTimes.” In “mdlStart,” a C++ class instance, which will be discussed in detail later, is created for each component model. Parameters and connection information are then assigned to each instance, and the state vector  $\mathbf{x}$  is initialized. In the simulation loop, the time step is first set in “mdlGetTimeOfNextVarHit” based on the CFL condition. Then, outputs  $\mathbf{y}_k$  are determined from the current state vector  $\mathbf{x}_k$  and inputs  $\mathbf{u}_k$  in “mdlOutputs.” The state vector is updated in “mdlUpdate” using the specified time step. Simulation continues until the user stops simulation or the final time is reached. Finally, model parameters and class instances are deleted in “mdlTerminate.” Subroutines are discussed later.

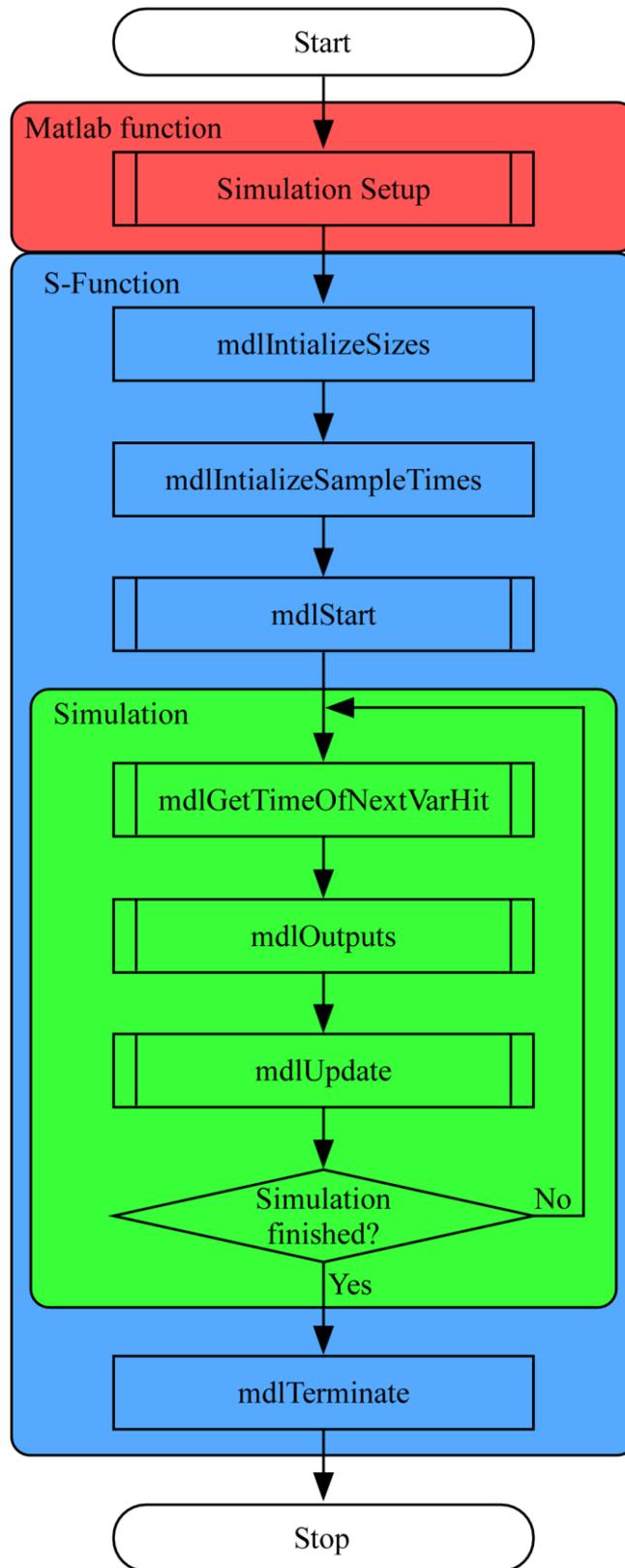


Figure 5.8: S-function setup and simulation flowchart

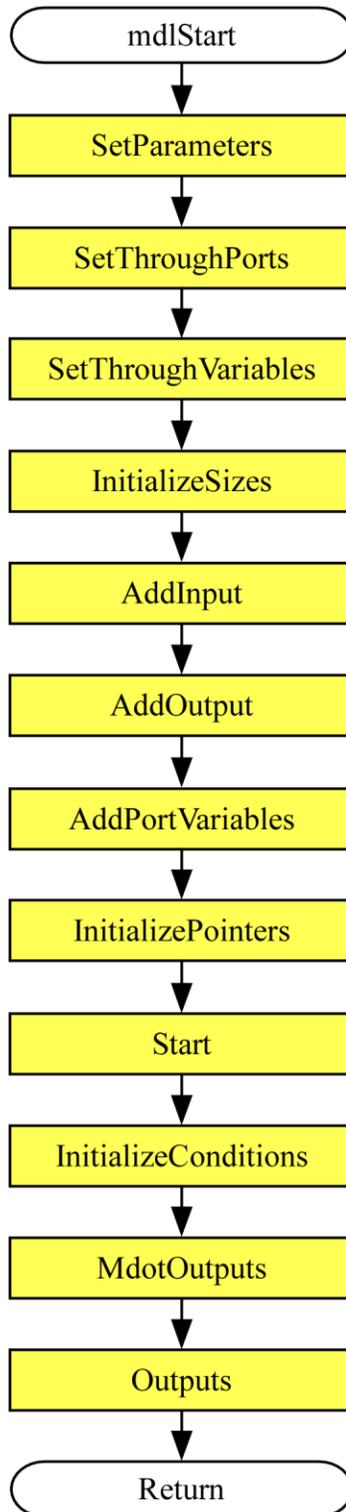
### 5.3.3 C++ Method Execution

All S-function setup takes place in the “mdlStart” method. Once instances are created for all engine component blocks, referring to Figure 5.9, parameters are passed to objects through the “SetParameters” method. Setting parameters first allows ports options configured in Simulink to be reflected in the model and block communication. The next task is to set “through ports” and “through variables.” Through ports and variables allow information to pass through a block, creating a virtual connection between components. For example, a mass flow rate sensor can be placed between boundary and 1D flow blocks, and by setting sensor ports as through ports, all the information output by the boundary block can be received by the 1D flow block and vice versa. A “through variable” allows a subset of variables to pass from one port to another. An engine valve model, for example, requires the crank angle to determine the lift. Therefore, the engine cylinder block assigns crank angle as a through variable between the dynamics and valve models. After assigning through ports and variables, array sizes are set in the “InitializeSizes” method. Array lengths can be defined by the user. For example, the number of cells in a 1D flow section or the number of connections to a port can be defined.

The next step in the flowchart in Figure 5.9, the next step is to setup port connections with “AddInput,” “AddOutput,” and “AddPortVariables” methods. For each input port, the input name determined during simulation setup is passed to the “AddInput” method of the appropriate object. During the “AddInput” call, a pointer to an element of the input array is set, and logic related to the input variable is executed. Similarly, output ports are set up by the “AddOutput” method. Once Simulink communication is established, communication between connected components is defined in the “AddPortVariables” method. The component connection acts as a two-way port; blocks provide pertinent variables to and require specific variables from the

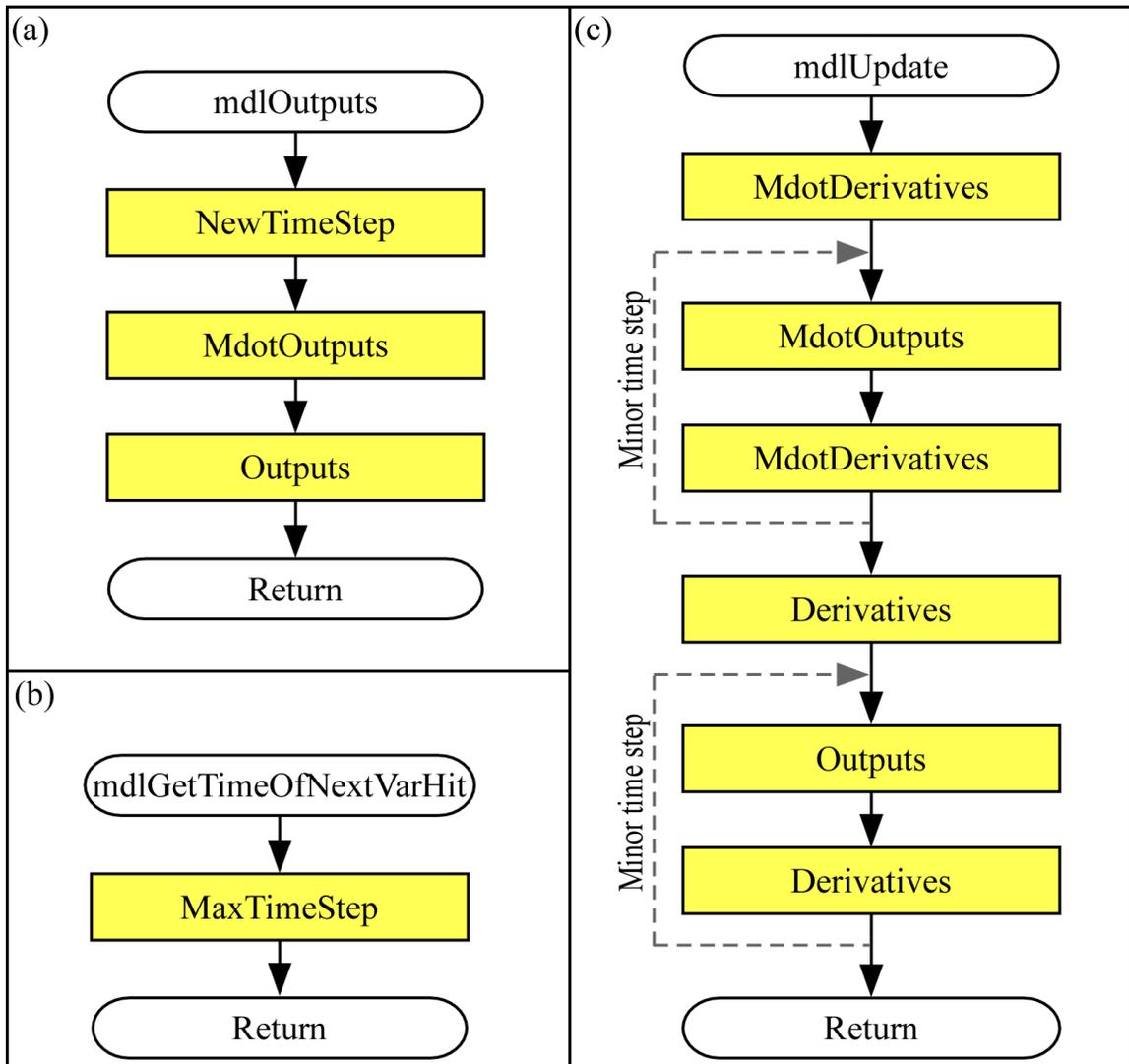
connected component. Variables available and required at the connection, which can depend on the type of connected blocks, are defined in the “AddPortVariables” method. For example, a valve connected to an engine cylinder block requires pressure, temperature, and mass fractions, and in turn, provides mass and energy flow rates to the cylinder block. A connection between a cylinder and crank dynamics model exchange difference parameters. Therefore, the type of connected component is passed to “AddPortVariables,” in order to define suitable variable availability and connection requirements.

Full model states and state derivatives are stored in arrays, and components with continuous states point to elements of the arrays. Prior to manipulating any of the state variables, state and state derivative pointers are set in the “InitializePointers” method. “InitializePointer” can also be used to set other variable or function pointers. Having a designated method for setting pointers reduces the probability of trying to access a null pointer: all memory is allocated in “InitializeSizes” and pointer variables should only be utilized after “InitializePointer” method. In Figure 5.9, the remaining methods called during “mdlStart” initialize parameter values. The “Start” method is used to calculate constants derived from port connections or user-defined parameters as well as initialize iteratively-solved variables. After parameter initialization, states are initialized in the “InitializeConditions” method. To end setup, output methods, “Outputs” and “MdotOutputs,” are called. Although the S-function calls the methods during simulation, values determined during output methods can be used to calculate the first time step.



**Figure 5.9: EngineModelComponent virtual method execution order for mdlStart**

After setup, the S-function enters the simulation loop (Figure 5.8), and at the beginning of each loop, “mdlGetTimeOfNextVarHit” determines the current time step by calling the “MaxTimeStep” method for each component, which returns the maximum allowable step size (Figure 5.10(b)). The minimum value among those returned by all components is set as the current time step. The method allows the CFL condition to be satisfied for the entire model. If the model does not include 1D flow components, the maximum allowable time step set at the start of the simulation is used.



**Figure 5.10: EngineModelComponent virtual method execution order for (a) mdlOutputs, (b) mdlGetTimeofNextVarHit, and (c) mdlUpdate**

Within the simulation loop, the next task is to calculate outputs and set up integration in “mdlOutputs.” Referring to Figure 5.10(a), the “NewTimeStep” method signals the start of simulation and executes tasks that can only be executed once per major time step. For example, crank angle determines several events in the cylinder model, and by comparing the current angle to the angle at the previous time step, an event can be triggered. Such triggers cannot be properly handled in output methods during integration and therefore must be handled each major time step. After “NewTimeStep,” output methods “MdotOutputs” and “Outputs” are called in order to calculate the output array and set up integration.

States are updated during “mdlUpdate” by “EngineModelComponent” output and derivative methods. First, the mass flow rates at each cell boundary  $\dot{\mathbf{m}}$  are updated using current states. Then, remaining states associated with 1D flow cells, cylinders, and crank dynamic states  $\mathbf{x}$  are updated. For each component, mass flow rate equations are represented in continuous form as

$$\dot{\mathbf{m}} = f_m(\mathbf{x}, \dot{\mathbf{m}}, \mathbf{u}) \quad (5.3a)$$

and

$$\mathbf{z} = g_m(\mathbf{x}, \dot{\mathbf{m}}, \mathbf{u}), \quad (5.3b)$$

where  $\mathbf{u}$  are inputs to the Simulink model and  $\mathbf{z}$  are outputs related to the momentum equation.

All other state equations are defined by

$$\dot{\mathbf{x}} = f(\mathbf{x}, \dot{\mathbf{m}}, \mathbf{u}) \quad (5.4a)$$

and

$$\mathbf{y} = g(\mathbf{x}, \dot{\mathbf{m}}, \mathbf{u}). \quad (5.4b)$$

Functions  $f_m$ ,  $g_m$ ,  $f$ , and  $g$  in Eq. (5.3) and Eq. (5.4) are represent by “EngineModelComponent” methods “MdotDerivatives,” “MdotOutputs” and “Derivatives,” and “Outputs,” respectively.

The methods can be used to simulate Eq. (5.3) and Eq. (5.4) using any number of ODE solvers.

As shown in Figure 5.10(c), derivative methods are called before entering the minor time step loop. At the beginning of each minor time step loop, output methods are called to update S-function outputs and variables communicated to connected components. Minor time steps, as the name suggests, are subdivisions of a major time step determined by an ODE solver. For example, the first order accurate Euler method integrates directly from current time to the new time without a minor time step. On the other hand, the fourth-order accurate Runge-Kutta method has three minor time steps.

#### **5.4 Steady State Save and Restart**

To generate performance maps and control lookup tables, engines are tested at steady state operating points. Engine simulation can reduce required testing by replicating test and calibration procedures with a virtual representation. The engine model must first be tuned to match experimental data at small number of operating points. Once tuned, the engine control parameters and physical properties can be optimized at steady state points. If the engine model is not properly initialized, the time required to reach a steady state takes much of the computation time due to initial manifold emptying and filling. Simulation time can be reduced by initializing the flow states near the steady state values. Because the engine is evaluated at a wide range of operating conditions, manually determining and setting appropriate initial conditions is time consuming. In order to overcome this difficulty, states can be saved at the end of a steady state simulation and used to restart the model in the next round of simulation. When the model restarts, states and outputs match the final result of the previous simulation. Changing the engine control or tuning parameters from simulation to simulation will change the final result, but restarting from a previous simulation reduces time to reach a steady state.

All variables in addition to states (e.g. initial values for Newton-Raphson iteration) needed to restart the model are registered in the “InitializeConditions” method. When enabled, all states and registered variables are recorded into an array and assigned to the Matlab base workspace at the end of simulation. The array is then passed back to the S-function when restarting from the previous simulation. Saved states and variables are set after running “InitializeConditions” method for each component. Note that Simulink states outside the engine model S-function must be saved and used to restart the model.

When tuning or evaluating the combustion model at steady-state operating conditions, flow through valves vary little between simulations. Therefore, by saving flow results from a full model simulation, the intake and exhaust models can be eliminated from subsequent simulations by replaying flow results. All intake valve, exhaust valve, and fuel injector information communicated to the combustion model are saved as a function of crank angle and reproduced in subsequent simulations. To prevent drift in cylinder pressure, mass fractions, mass, and energy prior to ignition, cylinder mass fractions and energy are saved and reproduced during the intake and exhaust phases. States related to cylinder turbulence are not overwritten throughout the cycle. In some cases, combustion can affect valve flow characteristics. A full model can be simulated occasionally to compare results and update valve flow results.

Cylinder, valve, and fuel injector models have an option to save crank angle-based data. Variables registered in the “InitializeConditions” method are logged based on the crank angle during simulation. At each time step, the model overwrites saved information from the previous cycle. When the model reaches a steady state, the final cycle is recorded and assigned to the Matlab base workspace. In a separate model without an intake or exhaust model, the recorded data is passed to the S-function. Recorded data is interpolated as a function of crank angle to

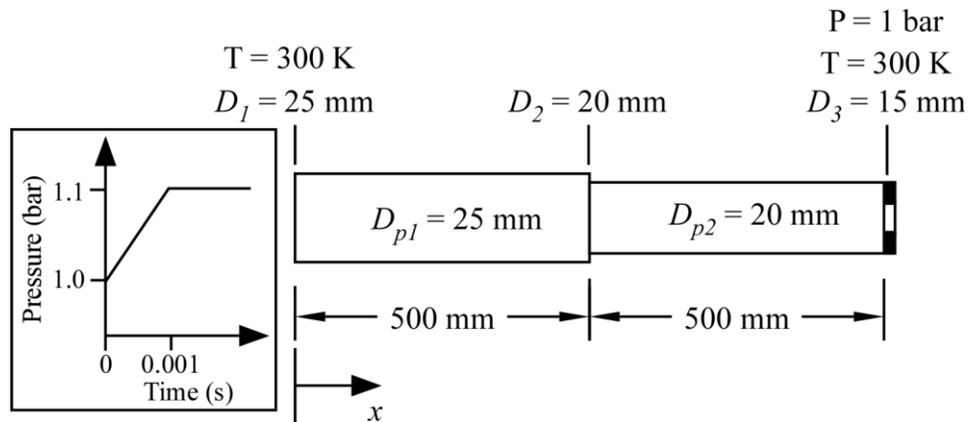
mimic the full engine simulation. Cylinder energy and mass fractions are only updated by the recorded data during the intake and exhaust phases.

## 5.5 Example Model

Comparing the proposed Simulink-based architecture and models to commercial software provides understanding of accuracy and usability of the new approach. For an initial validation, 1D flow and boundary models are compared to GT-Power. Researchers and automotive manufactures frequently employ GT-Power for engine simulation because of the proven accuracy of the software. Much like the proposed model, GT-Power allows users to connect 1D flow components in a physically representative manner, and based on user inputs, unsteady flow can be predicted by conservation laws. The block diagram and unsteady flow for the two methods are compared in the following sections.

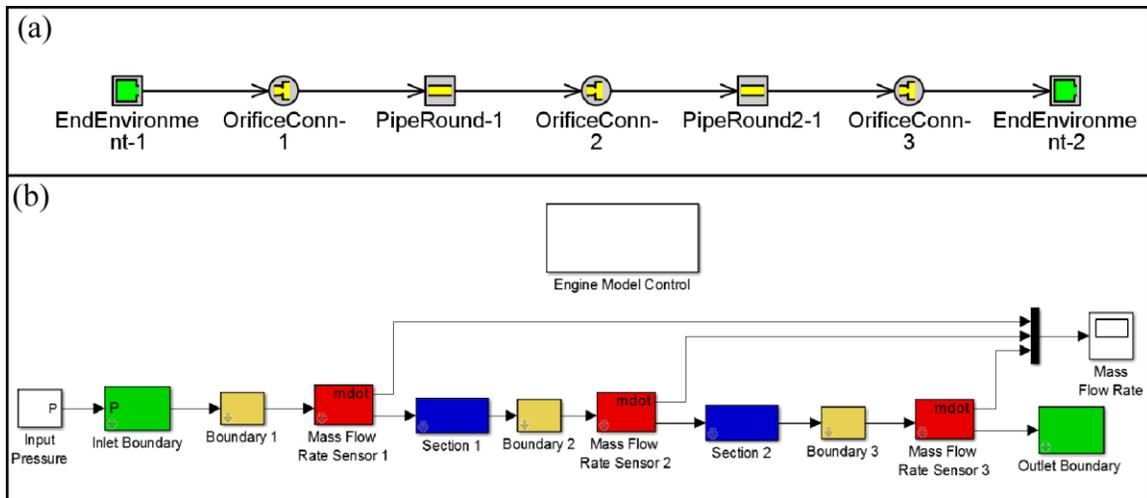
### 5.5.1 Simulation Setup

In general, a 1D pipe system consists of boundary conditions, flow sections, and a possible abrupt change in flow area. To compare the proposed Simulink model to GT-Power, the pipe system shown in Figure 5.11 is simulated. Temperature at both boundaries are held at 300 K, and the inlet pressure starts at 1 bar and increases to a steady 1.1 bar after 0.001 s, while the outlet pressure remains at 1 bar. Due to the increasing inlet pressure, flow enters the 25 mm pipe, and at the pipe exit, the gas must restrict to pass through the 20 mm pipe, creating a pressure drop at the pipe interface. The outlet boundary has a 15 mm orifice to represent a boundary restriction loss. Both the GT-Power and Simulink models assume adiabatic flow, surface roughness of  $\varepsilon = 0.046$  mm, and thermodynamic properties of dry air. Initially, the flow in the pipes is at rest with pressure at 1 bar and temperature at 300 K.



**Figure 5.11: Simulated pipe parameters and boundary conditions**

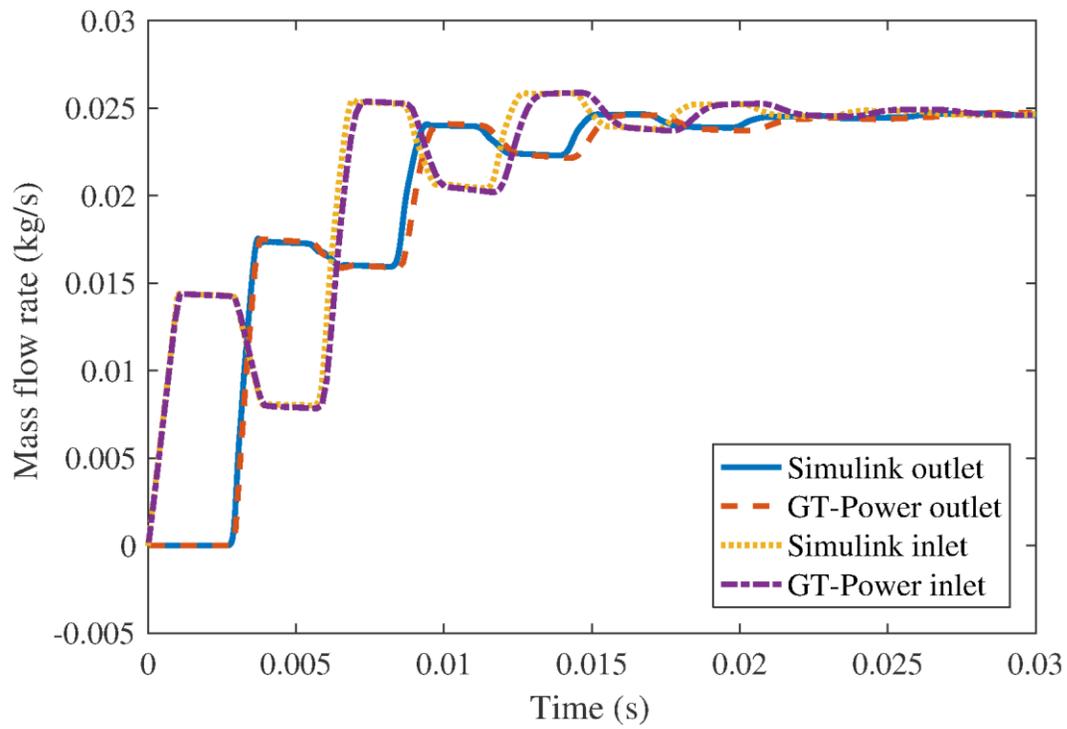
As shown in Figure 5.12(a), the GT-Power represents each type of component with blocks: “EndEnvironment,” “OrificeConn,” and “PipeRound.” The blocks handle data logging and inlet boundary pressure internally. The developed model shown in Figure 5.12(b) represents the pipe in a similar manner within the Simulink environment. However, mass flow rates are measured by optional mass flow rate sensors, and inlet pressure is provided by an external Simulink block. The mass flow rates are logged by the Simulink “Scope”, and inlet pressure is provided by a source block. In general, outputs from the engine model blocks can be connected to any Simulink block, and inputs can be provided by any traditional block. To accept the inputs and provide outputs, the S-function contained in the “Engine Model Control” block remotely communicates to the “Inlet Boundary” and mass flow rate sensors.



**Figure 5.12: (a) GT-Power and (b) Simulink block diagrams of the 1D flow model**

### 5.5.2 Results and Discussion

The simulation results obtained by the Simulink model closely match those provided by GT-Power as shown in Figure 5.13. At the start of simulation, the increasing pressure at the inlet boundary causes a gradual rise in inlet mass flow rate, and after reaching a steady boundary pressure, mass flow becomes steady until a pressure wave reflects back to the boundary. As expected, the outlet flow rate does not increase until the initial acoustic wave reaches the 15 mm orifice at the exit. The step changes in mass flow rates during unsteady flow are a result of the initial pressure wave propagating and reflecting at the pipe interface, inlet boundary, and outlet restriction. Note that the Simulink and GT-Power produce nearly identical results at the start of simulation, but as time progresses, the wave front produced by Simulink model tends to lag behind the GT-Power model due to a difference in wave propagation velocity. The difference in wave velocity can be attributed to minor differences in thermodynamic properties or model assumptions. Variation in the steady flow rates are likely a result of differences in friction factor models.



**Figure 5.13: Simulated inlet and outlet mass flow rates from GT-Power and Simulink**

## CHAPTER 6

### MODEL TUNING AND VALIDATION

The proposed 1D flow and quasi-dimensional combustion model can predict torque and flow characteristics across an engine's operating region. To provide an accurate representation, the model must first be tuned to match experimental data or CFD simulations at a subset of operating points. The flow model can be created from physical dimensions and measured discharge coefficients. The combustion model, however, contains several parameters that must be tuned to match cylinder turbulence and pressure. A method for calibrating the combustion model at a wide range of operating points is presented as well as model validation.

#### 6.1 Mazda Skyactiv-G® Engine

To evaluate potential future technologies, the United States Environmental Protection Agency (EPA) has benchmarked and modified a 2.0 L Mazda Skyactiv-G® engine [107]–[109]. Experimental data collected by the EPA was used to tune and validate the engine model. Table 6.1 provides high level specifications of the 2.0 L Skyactiv-G engine [107]. The Skyactiv-G engine is a naturally aspirated Gasoline Direct Injection (GDI) engine with variable cam phasing. The tested version has a 13:1 compression ratio and no EGR. Mazda attributes the ability to achieve the high compression ratio and avoid knock to a few key features: 4-2-1 exhaust manifold and combustion improvement [110]. The 4-2-1 exhaust manifold has long runners with a collector for cylinders 1 and 3 and a separate collector for cylinders 2 and 4. The two collectors for the cylinder pairs are then combined into a single pipe. The design provides better exhaust

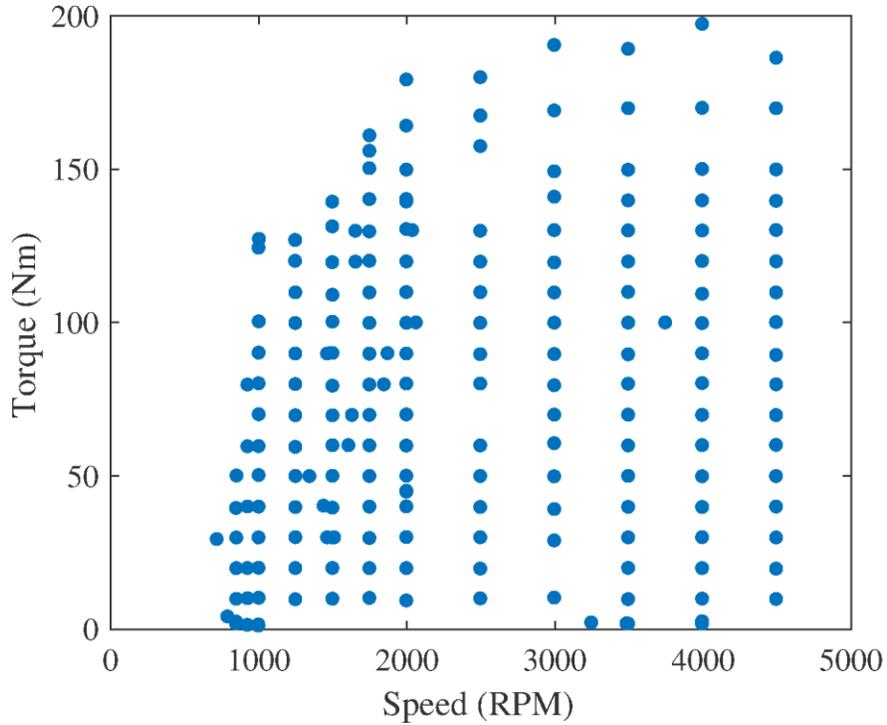
tuning characteristics than the traditional single collector exhaust manifold with short runners. The better tuning reduces the fraction of residual gas at a wide range of operating conditions, thus reducing combustion temperatures and the probability of knock. Knock resistance is also improved by shorting the combustion duration. Rapid burn rates are achieved by centering the spark plug and increasing turbulence.

**Table 6.1: Specifications for 2.0L Mazda Skyactiv-G engine [107]**

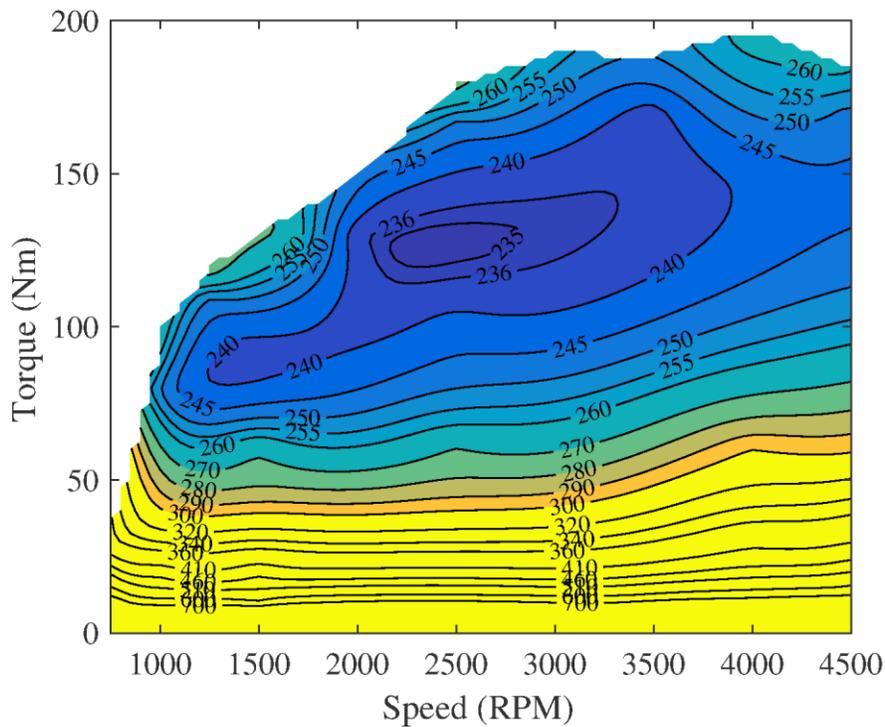
Displaced volume	1998 cc
Bore	91.2 mm
Stroke	83.5
Rated torque	203 Nm at 4000 RPM
Rated power	115 kW at 6000 RPM
Compression ratio	13:1

The EPA provided steady state dynamometer test data with speeds ranging from idle to 4500 RPM and torques ranging from no load to Wide Open Throttle (WOT). Speed and torque test points are plotted in Figure 6.1. At each operating point, engine control parameters, data averaged over several cycles, and derived performance indicators were recorded. Control parameters include: spark advance, intake cam phase, exhaust cam phase, throttle angle, injection timing, injection pulse width, and measured air-fuel ratio. Average data includes information such as peak cylinder pressure for each cylinder, air mass flow rate, and manifold pressures. Derived performance indicators include information such as gross and net indicated mean effective pressures ( $IMEP_g$  and  $IMEP_n$ ), burn fraction angles, volumetric efficiency, and Brake Specific Fuel Consumption (BSFC). BSFC is a common indicator of the overall engine

efficiency (Figure 6.2). Crank angle-based cylinder pressure, exhaust manifold pressure, and intake manifold pressure were provided for 19 of the 229 valid operation points. A portion of the recorded data did not include air mass flow rate and could not be interpolated from other operating points. The engine model was not validated at these operating points.



**Figure 6.1: Steady state speed and torque points tested by EPA**

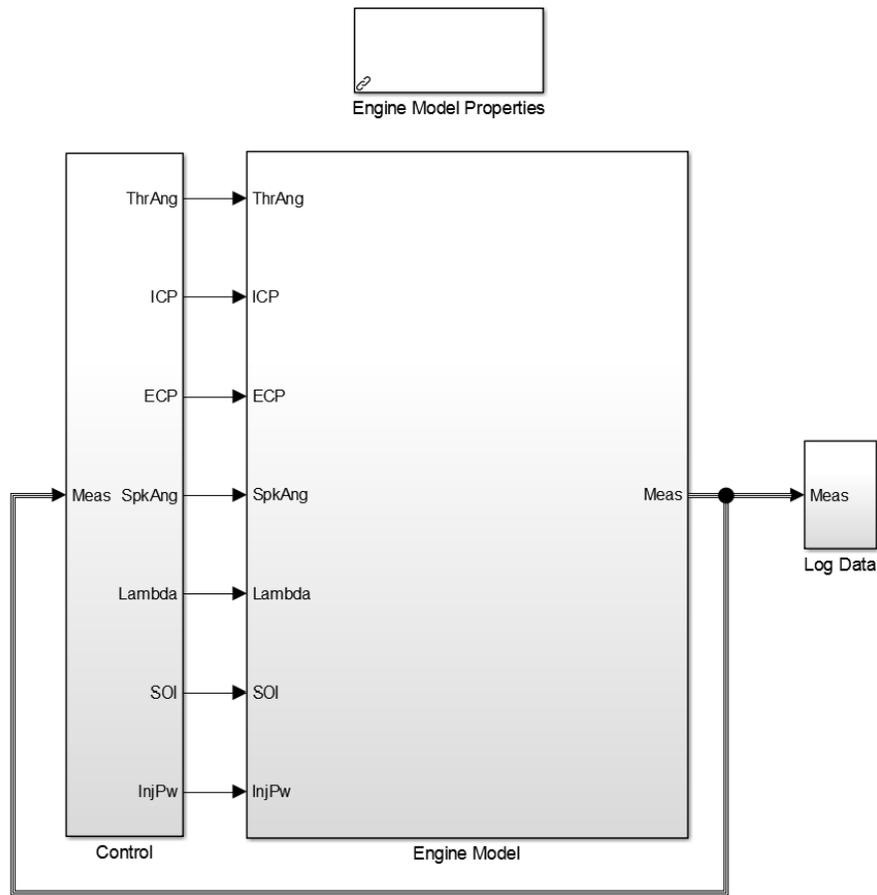


**Figure 6.2: Measured BSFC (g/kWh) for 2.0L Mazda Skyactiv-G engine**

## 6.2 Simulink Model

### 6.2.1 Overview

The Skyactiv engine Simulink model shown in Figure 6.3 consists of a controller and plant model. The Simulink model is configured to simulate a single speed and torque operating point but could be reconfigured to simulate transient cycles. The controller determines control inputs based on the selected operating point and stops simulation once steady state criteria have been met. Throttle control and steady state detection require plant model feedback from the plant model. The plant model consists of the proposed 1D flow components and quasi-dimensional combustion model. Plant model environment settings and fuel type are set in the “Engine Model Properties” block which contains the plant model S-function. Data measured during simulation is logged for post processing.



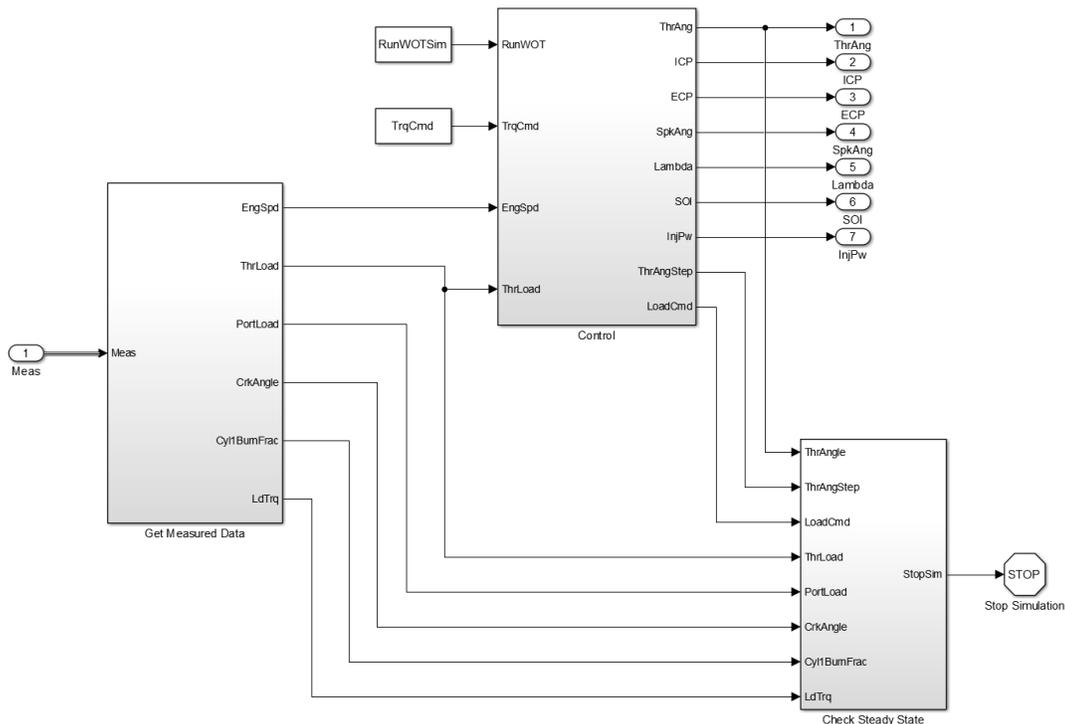
**Figure 6.3: Simulink model of Skyactiv engine**

## 6.2.2 Control

The controller subsystem shown in Figure 6.4 sets throttle angle, intake cam phase, exhaust cam phase, spark angle, injection timing, injection pulse width, and air-fuel equivalence ratio for a given engine speed and torque command and stops simulation once a steady state has been reached. Cam phasing, spark angle, injection timing, injection pulse width, and air-fuel equivalence ratio are determined from 2D lookup tables as a function of set speed and torque command. Therefore, the engine can be simulated anywhere within the tested envelope, not only at the specific test points. Note that the control parameters depend on the air flow rate but can be correlated to torque by meeting the specified air flow rate. Data for each control parameter is fit

to variations of Gaussian Process Model (GPM) and Radial Basis Function (RBF) models using Model-Based Calibration Toolbox™. Fit models that best represented the data are used to create the control tables. Tables and fit information are provided in Appendix A.

Depending on the operating condition, the Skyactiv engine has one or two injection events. Total fuel mass required to meet the specified air-fuel ratio is determined at the start of each cycle based on the trapped air mass of the previous cycle. Assuming the mass flow rate to be equivalent for each injection event, mass flow rate is determined by dividing the total required fuel mass by the total injection pulse width.



**Figure 6.4: Controller subsystem in Simulink engine model**

A feedback controller sets throttle angle in order to match the desired air mass flow rate, which is represented as normalized air charge  $L_{eng}$ . In the Simulink model,  $L_{eng}$  is defined as

$$L_{eng} = \frac{2\dot{m}_{air}}{V_d N \rho_{std}}, \quad (6.1)$$

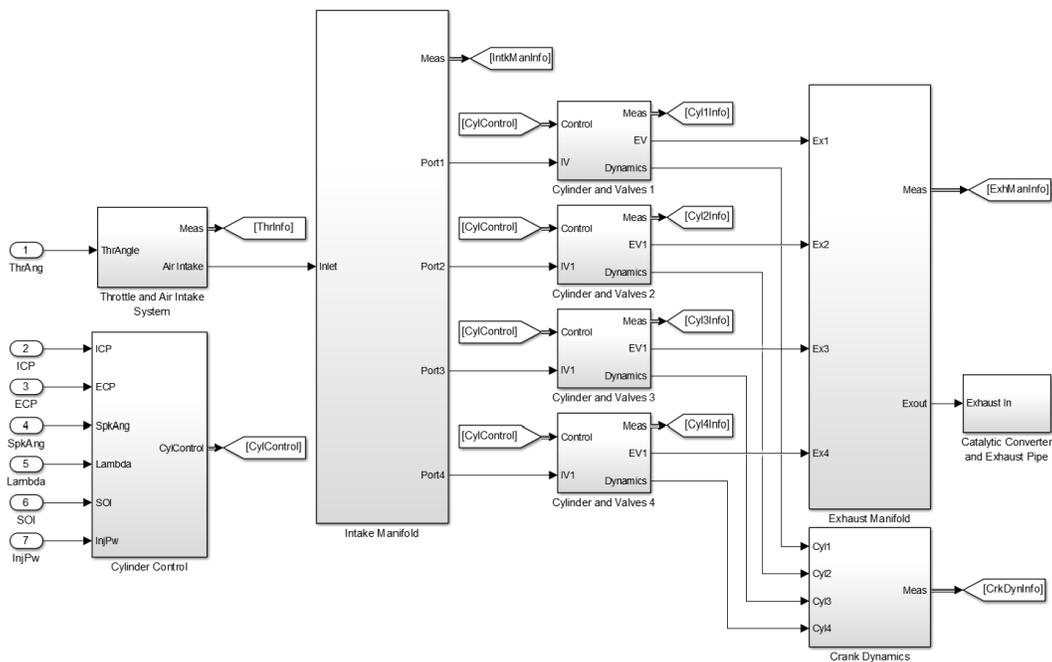
$V_d$  is the total displaced volume,  $N$  is the engine speed,  $\dot{m}_{air}$  is the air mass flow rate, and  $\rho_{std}$  is the density of air at a standard temperature (298.15 K) and pressure (101325 Pa). Eq. (6.1) is equivalent to the definition of volumetric efficiency when operating the engine at a standard temperature and pressure. Because the relationship between throttle angle and air flow is highly nonlinear and the plant model could be modified from simulation to simulation, implementing a traditional nonlinear controller would be very challenging and require additional simulations to create the controller. Therefore, a logic based controller has been implemented using Stateflow®. The controller initially sets the throttle angle at the measured value and increases or decreases the throttle depending the sign of the load error. After waiting a specified period of time, the throttle angle is increased or decreased again. If the error crosses zero, the throttle step size is halved and switches directions. The process continues until the minimum step size is reached.

Simulation stops when the cycle averaged torque, normalized air charge at the throttle, and normalized air charge at the ports remain relatively constant for three cycles. Air flow at the throttle and ports are treated separately to ensure the flow through the intake manifold has reached a steady state, which is especially important at low load. When all criteria have been met, simulation stops after completing five additional cycles. This control logic is implemented in a Stateflow® chart.

### 6.2.3 Plant Model

Two plant models were created for validation and parameter optimization: full engine with 1D model for intake and exhaust (Figure 6.5) and model with intake and exhaust flow imported from full model simulation (Figure 6.6). All geometric parameters, discharge

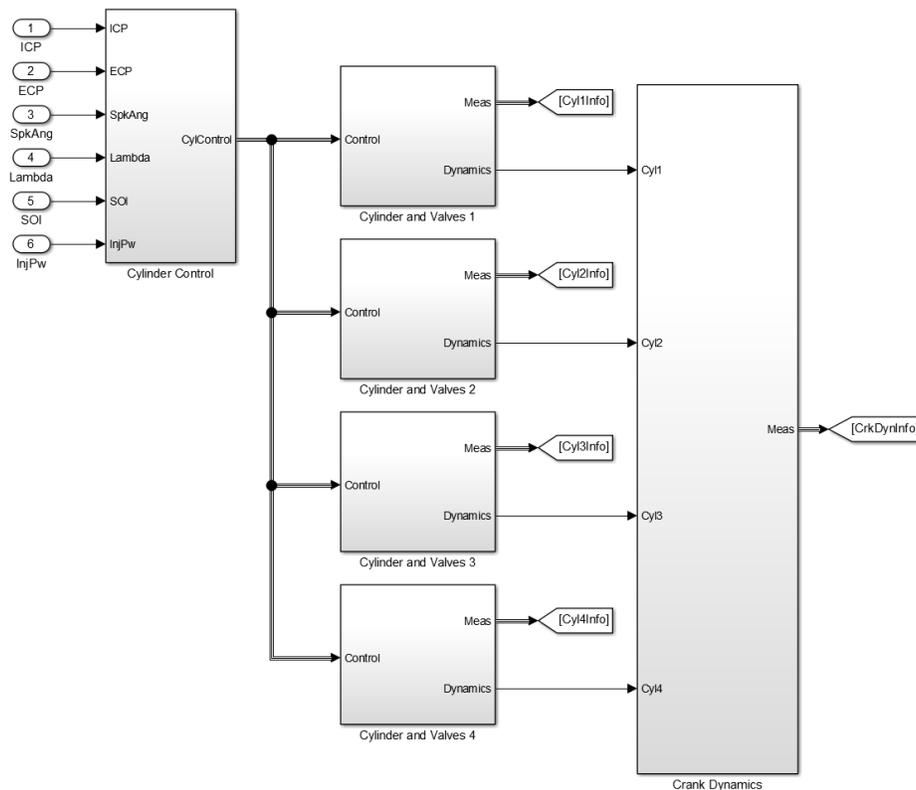
coefficients, and valve lifts used to build the flow model were taken from a GT-Power model provided by the EPA. Valve flow information is provided in Appendix B. The EPA did not provide intake swirl and tumble numbers. Based on the Skyactiv engine port geometry, however, swirl was neglected. Tumble number as a function of valve lift and combustion tuning coefficients were optimized to match cylinder pressure. Flame surface areas were calculated using a CAD model of the combustion chamber.



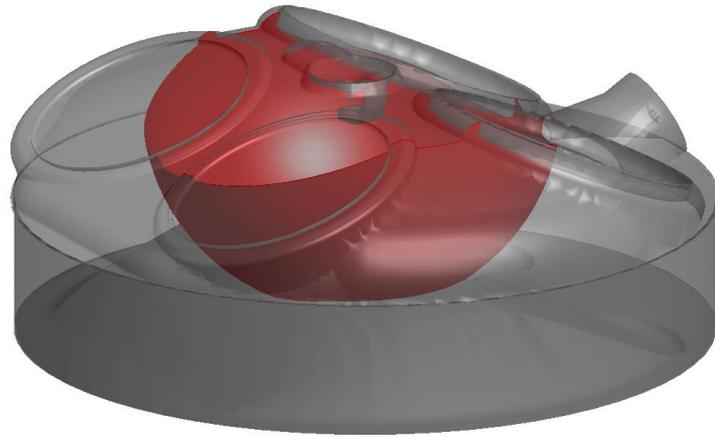
**Figure 6.5: Full Simulink model for Skyactiv engine with throttle, intake manifold, cylinders, crank dynamics, and exhaust system**

Combustion and heat transfer models require flame surface area and burned volume measurements. By modeling the flame as a sphere originating at the spark plug, flame geometry varies with piston position  $x_p$  and flame radius  $r_f$ . Lookup tables were generated using cylinder CAD drawings provided by the EPA. The internal cylinder head and piston crown surfaces were converted to the STL format, a triangular mesh representation of a 3D surface, and imported to Matlab. In Matlab, the gptoolbox was used to calculate volumes, intersect meshes, and surface

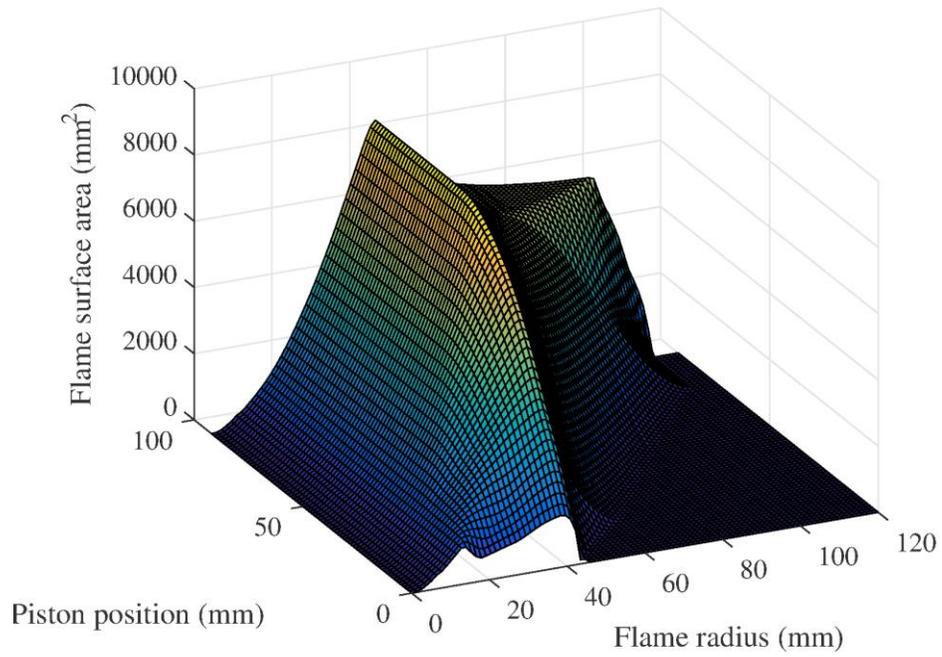
areas [111]. The combustion chamber volume was generated in Matlab by moving the piston surface relative to the cylinder head and intersecting a cylinder with the two meshes. Cylinder wall below the piston was removed, creating an enclosed volume at the given piston position. Total piston, head, and cylinder wall surface areas and total volume could then be calculated. At each flame radius, a sphere mesh with origin at the spark plug was generated and intersected with the chamber volume. Sections of the sphere outside the combustion chamber were removed, leaving burned and unburned volumes as shown in Figure 6.7. Surface area and volume information recorded at each piston position and flame radius were used to generate area and volume tables. The calculated flame surface area and burned volume ratio are shown in Figure 6.8 and Figure 6.9, respectively.



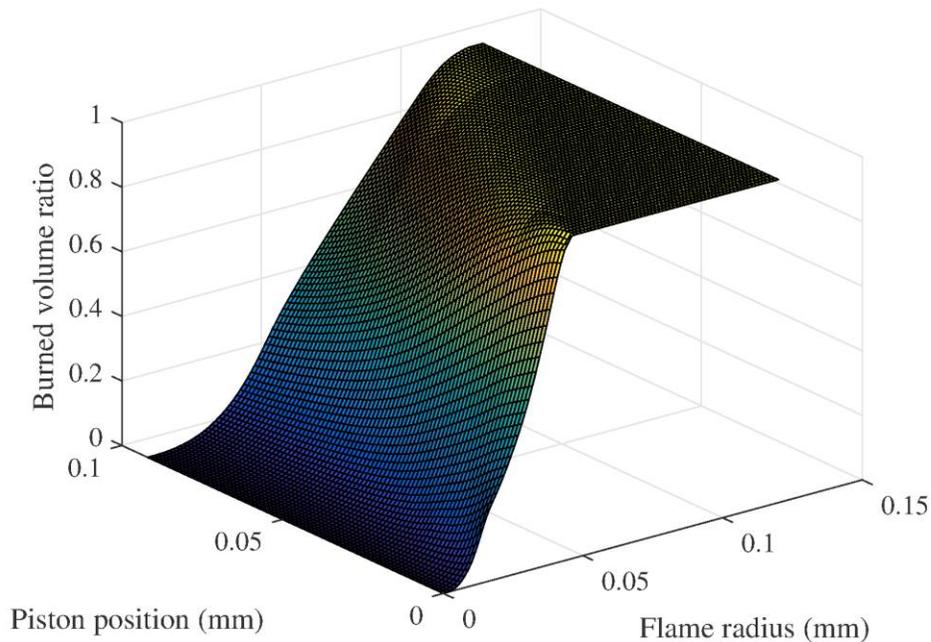
**Figure 6.6: Simplified Simulink model for Skyactiv engine with cylinder and crank dynamics only, requiring port flow to be imported from full engine simulation**



**Figure 6.7: Skyactiv engine combustion chamber and spherical flame displayed in Matlab**



**Figure 6.8: Spherical flame surface area for Skyactiv engine combustion chamber**



**Figure 6.9: Burned volume ratio  $V_b/V_{cyl}$  for Skyactiv engine combustion chamber**

### 6.3 Combustion Model Tuning

Combustion burn rate depends on the type of fuel, burned gas residuals, cylinder geometry, and turbulence. Fuel and burned gas residuals contributions are represented by the laminar flame speed correlation. The effect of cylinder geometry is captured by the flame surface area and volume tables. A turbulence model requires tuning using experimental test data and/or CFD simulation results. During the engine design phase, detailed CFD or experimental data from a similar engine would be needed to determine turbulence parameters. Engine simulations can significantly reduce engine testing for engine calibration: physically test an engine at a relatively small number of operating points, tune a combustion model to match experimental data, and run desktop-based engine calibrations. The process reduces engine dynamometer testing and calibration time. As the complexity of internal combustion engines increase, the use of desktop calibration becomes more beneficial.

The combustion model tuning procedure matches measured cylinder pressure at a wide range of operating points. Spreading test points across the engine's operating range is necessary to capture the engine's behavior. The number of operating points required to tune the model depends primarily on the number of unknown parameters. Without swirl and tumble information, several parameters must be determined. Manual tuning can be very challenging even with a small number of parameters, and optimization is computationally expensive when considering a large set of tuning parameters. The presented tuning procedure estimates tumble and determines combustion tuning parameters. Computation time for the optimization process can be reduced by eliminating the flow model when calculating the cost function.

### 6.3.1 Optimization Parameters

As explained previously, intake swirl was neglected, leaving tumble as the only form of mean cylinder charge motion. Without tumble flow bench measurements, tumble number was included in the optimization, assuming that tumble number  $N_T$  varies only with valve lift. In Figure 6.10, three constants,  $N_{T,1}$ ,  $N_{T,2}$ , and  $N_{T,3}$  were used to generate a cubic spline for  $N_T$  as a function of valve lift, where valve diameter  $D_v$  is used to normalize the valve lift. Setting  $N_T = 0$  at Intake Valve Open (IVO), constants could be equally spaced along valve lift, with  $N_{T,3}$  defining  $N_T$  at the maximum lift. Tumble numbers were assumed to be between 0 and 1, and increase with valve lift, thus

$$1 \geq N_{T,3} \geq N_{T,2} \geq N_{T,1} \geq 0 \quad (6.2)$$

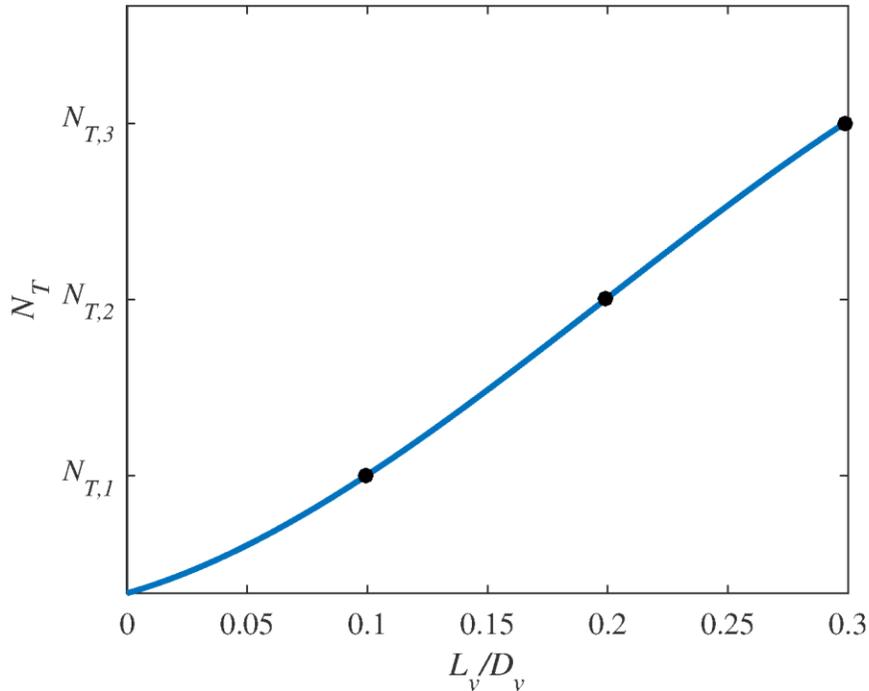
Because certain combination of  $N_{T,1}$ ,  $N_{T,2}$ , and  $N_{T,3}$  can result in the cubic spline going outside the specified range, the following constraint was applied once the spline was generated:

$$0 \leq N_T \leq 1, \quad (6.3)$$

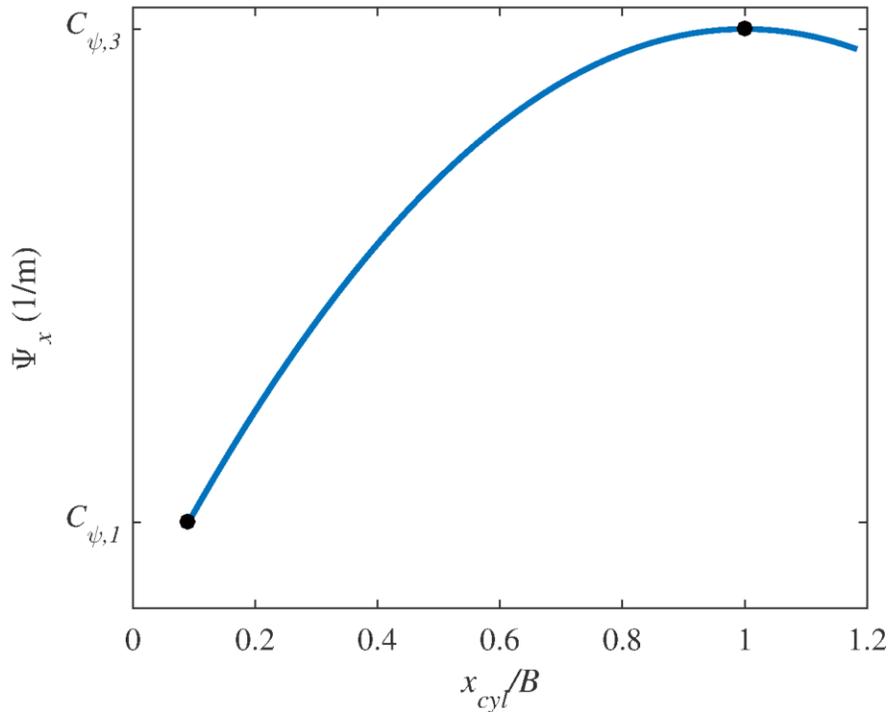
The tumble decay function  $\Psi_x$  depends on the combustion chamber geometry and varies with piston position. Grasreiner *et al.* represented tumble as a Taylor-Green vortex and determined  $\Psi_x$  by conducting CFD simulations at different cylinder positions [91]. As expected,  $\Psi_x$  decreased (increase shear torque) as the piston approached TDC and was maximized at  $x_{cyl}/B = 1$  (minimal tumble decay). Based on the relationship,  $\Psi_x$  was fit to the equation

$$\Psi_x = \frac{C_{\psi,1} - C_{\psi,3}}{\left| \frac{x_{cyl,TDC}}{B} - 1 \right|^{C_{\psi,2}}} \left| \frac{x_{cyl}}{B} - 1 \right|^{C_{\psi,2}} + C_{\psi,3}, \quad (6.4)$$

where  $C_{\psi,1}$ ,  $C_{\psi,2}$ , and  $C_{\psi,3}$  are fitting constants, and  $x_{cyl,TDC}$  is the cylinder height at TDC. As shown in Figure 6.11, Eq. (6.4) fits a power function between  $C_{\psi,1}$ , minimum  $\Psi_x$ , and  $C_{\psi,3}$ , maximum  $\Psi_x$ .  $C_{\psi,1}$ ,  $C_{\psi,2}$ , and  $C_{\psi,3}$  were included in the optimization.



**Figure 6.10: Tumble number parameterization**



**Figure 6.11: Tumble decay function  $\Psi_x$  parameters**

The turbulence model has several tunable parameters. Standard  $k$ - $\varepsilon$  model parameters  $C_{\varepsilon 1}$ ,  $C_{\varepsilon 2}$ , and  $C_{\mu}$  were kept constant. Valve lift turbulence constant  $C_{\varepsilon, valve}$  was included in the optimization assuming that the integral length scale of the incoming flow is proportional to and does not exceed valve lift. The injection integral length scale  $l_{I, inj}$  was optimized in order to capture the effect of injection on overall combustion. Because premixed combustion requires early fuel injection and fuel is a fraction of the total flow,  $l_{I, inj}$  has less effect than  $C_{\varepsilon, valve}$  on turbulence. More investigation is needed to adequately represent late injection (i.e diesel engine).

The burn rate model includes turbulent flame speed and eddy burn up time constant  $\tau_b$ . Early combustion is tuned by  $C_{dev}$  and initial flame kernel radius  $r_{f0}$ .  $C_{dev}$  modifies the time required to transition from laminar to turbulent combustion based on the integral length scale. The flame kernel created by the spark is on the order of 1 mm. Herweg and Maly observed a

kernel radius of approximately 2 mm at about 200  $\mu$ s after spark onset [112]. Coefficients  $C_f$  and  $C_b$  modify the burn rate throughout combustion.  $C_f$  has more effect during fully developed turbulent combustion. The eddy burn time coefficient  $C_b$  has an effect throughout combustion, but impact increases late in combustion.

Optimized parameters are summarized in Table 6.3 with minimum and maximum values. Tumble number constants  $N_{T,1}$ ,  $N_{T,2}$ , and  $N_{T,3}$  were subject to the constraint given in Eq. (6.2). The initial values and ranges selected for the tumble decay coefficients  $C_{\psi,1}$ ,  $C_{\psi,2}$ , and  $C_{\psi,3}$  were selected based on previously published results [91]. Constants  $C_f$ ,  $C_b$ , and  $C_{dev}$  should be near 1. Turbulence constants are summarized in Table 6.2.

**Table 6.2: Constant turbulence parameters**

$C_{\varepsilon 1}$	$C_{\varepsilon 2}$	$C_{\mu}$
1.44	1.92	0.09

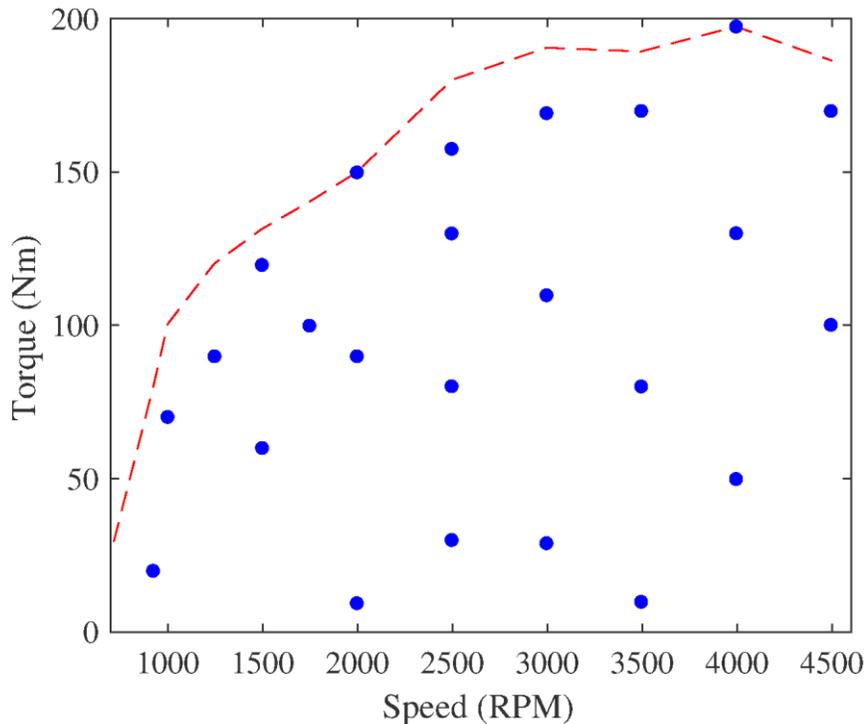
**Table 6.3: Combustion model parameters and constraints**

	Initial	Minimum	Maximum
$N_{T,1}$	0.1	0	1
$N_{T,2}$	0.3	0	1
$N_{T,3}$	0.5	0	1
$C_{\psi,1}$	-100	-160	-60
$C_{\psi,2}$	2	2	4
$C_{\psi,3}$	-5	-30	0
$C_f$	1	0.5	1.5
$C_b$	1	0.5	1.5
$C_{dev}$	1	0.5	1.5
$r_{f0}$	1 mm	0.5 mm	4 mm
$C_{\epsilon, valve}$	1	0.3	1
$l_{I, inj}$	1 mm	10 mm	5 mm

### 6.3.2 Fitness Function

The goal of parameter tuning is to determine fitting coefficients that provide the best match between simulations and measurements. In the case of the combustion model, parameters

in Table 6.3 must be tuned to match measured cylinder pressure across the engine’s operating region. The optimization fitness function defines the error between the simulated and measured cylinder pressures. To match results, the 24 representative operating points shown in Figure 6.12 were simulated for evaluation of a fitness function. Each of the operating points coincides with a dynamometer test and was selected to cover the tested region. Given the number of tuning constants, reducing the number of operating points can result in overfitting. Increasing the number of points increases time required to evaluate the fitness function.



**Figure 6.12: Fitness function evaluation points**

Cylinder pressure was represented by the net Indicated Mean Effective Pressure ( $IMEP_n$ ) and peak cylinder pressure for each cylinder. Each objective function was defined as the weighted Mean Squared Error (MSE) of the respective values.  $IMEP_n$  relates to the torque production, and by matching measured air flow and air-fuel ratio, minimizing  $IMEP_n$  error also

minimizes thermal efficiency error. Matching peak cylinder pressure is important for predicting knock, which is necessary for calibrating spark advance. Minimizing MSE reduces the absolute error at the optimized operating conditions but can result in high relative error at low load. Constructing a fitness function based on the relative error places emphasis on low load conditions and will likely not provide acceptable results due to the relative error in the measured air flow. Relative error in measured air flow can be significant near idle condition.

The IMEP<sub>n</sub> and peak cylinder pressure fitness functions were defined as

$$f_{IMEP_n} = w_1 MSE_{IMEP_n} \quad (6.5)$$

and

$$f_{peak} = w_2 \frac{(MSE_{peak1} + MSE_{peak2} + MSE_{peak3} + MSE_{peak4})}{4}, \quad (6.6)$$

where  $w_1$  and  $w_2$  are the IMEP<sub>n</sub> and peak cylinder pressure weights, respectively. Fitness functions  $f_{IMEP_n}$  and  $f_{peak}$  can only be optimized using multi-objective optimization techniques but can be summed for a single objective optimization:

$$f_{total} = f_{IMEP_n} + f_{peak} \quad (6.7)$$

Cycle-to-cycle variations were used to determine  $w_1$  and  $w_2$ . At each operating point, cylinder pressure was measured for multiple cycles. Average IMEP<sub>n</sub> and peak cylinder pressures with respective standard deviations were calculated from the measurements. Maximum IMEP<sub>n</sub> standard deviation  $\sigma_{IMEP_n}$  and maximum peak pressure standard deviation  $\sigma_{peak}$  were used to define weights  $w_1$  and  $w_2$  as

$$w_1 = \frac{1}{\max(\sigma_{IMEP_n})^2} \quad (6.8)$$

and

$$w_2 = \frac{1}{\max(\sigma_{peak})^2}. \quad (6.9)$$

The weights balance the sensitivity of  $IMEP_n$  and peak cylinder pressure.

### 6.3.3 Optimization Procedure

The optimization problem includes several conflicting parameters that can result in local minima. For example, increasing  $C_{e, valve}$  increases the burn rate which can be counter acted by decreasing  $C_f$ . Furthermore, one set of parameters may minimize error at one operating point but increase the error at another. Having numerous local minima in the search space eliminates the use of gradient-based optimization techniques, even with multi-start. Therefore, non-gradient based global optimization technique must be employed. Pattern search and the genetic algorithm were considered.

Pattern search techniques poll points around the current position, and if one of the polled points produces a better fitness value, the new point is selected and the polling step size increases [113]. If polling does not result in a better fitness value, polling continues with a reduced step size. The polling step size can frequently oscillate due to numerous local minima and has a slow rate of convergence when optimizing a large number of parameters [113]. Most polling methods evaluate the fitness function two times the optimization dimension. Polling can be stopped if a new minimum has been located or continue for a complete poll. During each iteration, polling is independent. Therefore, each function evaluation can be calculated in parallel.

The genetic algorithm-based optimization technique mimics natural evolution by generating a population of possible solutions (parents) and promoting combinations of the strongest candidates to the next generation (children) [114]. For each generation, a fraction of the candidate's genes (optimization variables) are mutated in order to better explore the design space. Attributes producing the best fitness are passed to the next generation, until the maximum number of generations have been reached or other stopping criteria have been met. The fitness of

all the members of the current generation can be evaluated in parallel, allowing parallel computing to be leveraged for large optimization problems.

The genetic algorithm can be adapted for multi-objective optimization. Combining two objectives into a single fitness function can miss potential solutions and require appropriate weights, which is difficult with conflicting fitness functions. Multi-objective optimization provides a series of non-dominated solutions. A solution  $\mathbf{x}^{(1)}$  is dominated over solution  $\mathbf{x}^{(2)}$  if both of the following rules are met [115]:

1.  $\mathbf{x}^{(1)}$  results in a solution no worse than  $\mathbf{x}^{(2)}$  for all objectives;
2.  $\mathbf{x}^{(1)}$  results in a better solution than  $\mathbf{x}^{(2)}$  in at least one objective.

The set of non-dominated solutions, called Pareto-optimal solutions, produces a Pareto frontier with tradeoffs between objective functions. Once optimization has completed, the conflicting behavior of the fitness functions can be observed, and a single solution can be chosen without having to trade off objectives prior to optimization.

Based on preliminary optimizations, the genetic algorithm was chosen over pattern search. Pattern search optimization converged slowly and frequently resulted in a worse solution. The slow convergence resulted in more fitness function evaluations than required with the genetic algorithm due to the fitness function behavior and large search space. Both single and multi-objective genetic algorithm optimizations were employed to tune the combustion model.

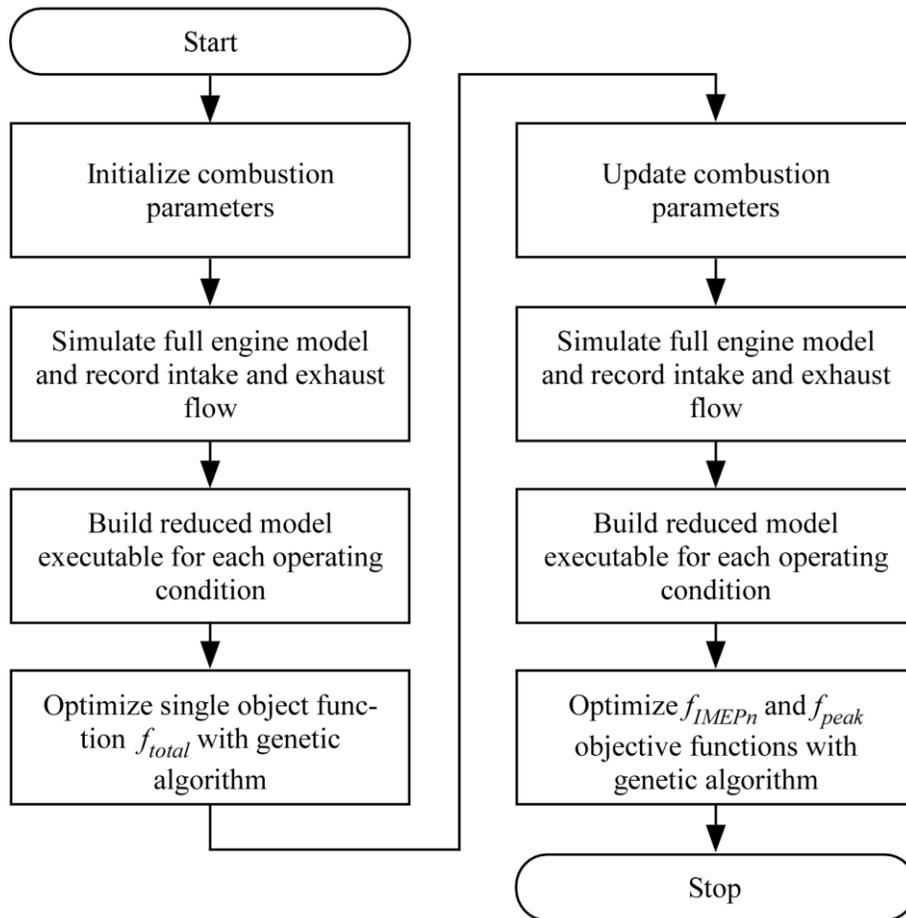
As with any global optimization technique, the genetic algorithm requires a large number of fitness function evaluations in order to converge on the optimal solution. To greatly reduce optimization time, the full engine (1D flow and quasi-dimensional combustion) was simulated prior to optimization and used to restart a cylinder-only model during optimization. Alternatively, Three Pressure Analysis (TPA), which requires cylinder, intake port, and exhaust

port pressures, could be used to reduce simulation time by simulating a reduced the flow model (e.g. [108]). However, the EPA did not provide port pressures.

As shown in the flowchart in Figure 6.13, the full engine model was simulated with initial parameters listed in Table 6.3 at the fitness function evaluation points shown in Figure 6.12. Intake flow, exhaust flow, fuel injection, and states during gas exchange were saved as a function of crank angle and used to run the optimization model. The optimization model only included four cylinders and crank dynamics. Optimization time could be further reduced by simulating the model in Simulink's "Rapid Accelerator" mode. "Rapid Accelerator" mode builds a standalone executable that contains the solver and model [116]. Once compiled, the standalone executable can be simulated without running model initialization. Parameters must be "tunable" in order to change the value between simulations without recompiling. The crank angle based data saved from the full engine simulation was not tunable in the reduced model, requiring a standalone executable for each operating point. All optimization constants in Table 6.3 were tunable. For each fitness function evaluation, the executables returned cylinder pressure profiles resulting from the current test values. IMEP<sub>n</sub> and peak pressures were then determined from the last simulated cycle.

The restart method assumes modifying the combustion parameters does not impact intake and exhaust flow. However, if optimization results in significantly different pressure profiles than the initial full engine simulation, the assumption may no longer be valid. Dissimilar pressures at Exhaust Valve Open (EVO) can result in residual burned gas fraction variation and affect the transient intake mass flow rate. To minimized differences between the full and reduced models, optimization was completed in two steps as shown in Figure 6.13. To bring the combustion parameters closer to the final solution, the single-objective fitness function  $f_{total}$ ,

defined in Eq. (6.7), was optimized in the first step. At the start of the second step, the full engine was simulated with the current best parameters, and the results were used to update the reduced model standalone executables. Objective functions  $f_{IMEP_n}$  and  $f_{peak}$  (Eqs. (6.5) and (6.6)) were optimized in the second step using the multi-objective genetic algorithm. Multi-objective optimization was employed to ensure that weights  $w_1$  and  $w_2$  produce a reasonable single-objective result and to observe IMEP<sub>n</sub> and peak pressure dependency.



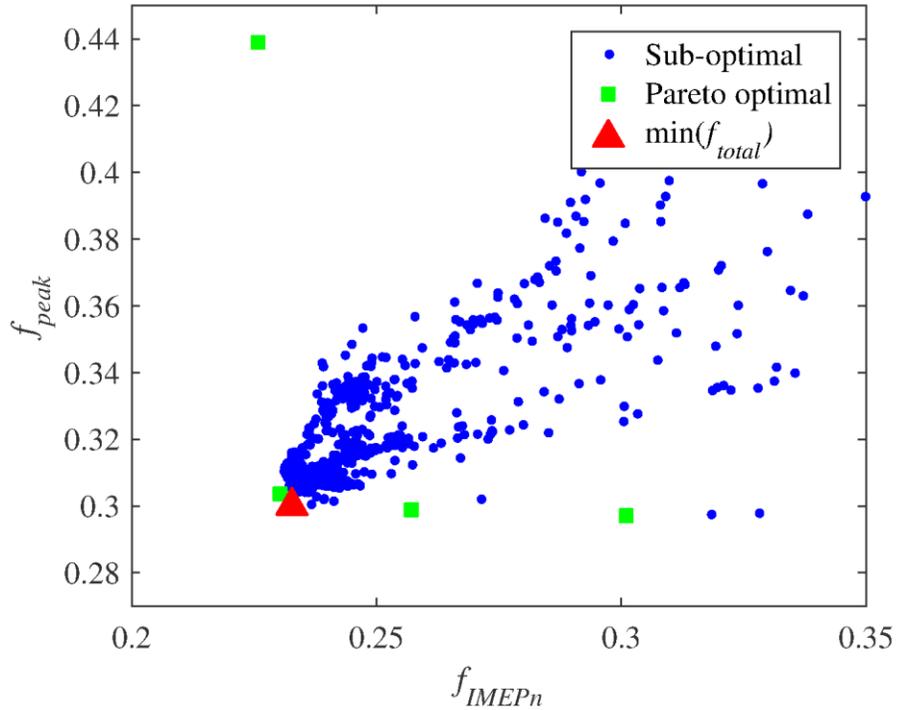
**Figure 6.13 Flowchart for parameter optimization**

The Matlab Global Optimization Toolbox™ was used for optimization. Using minimum and maximum values listed in Table 6.3, all tuning constants were normalized and bounded. Population size of 150 was used for both the single and multi-objective optimizations. The

single-objective optimization (first step) was terminated after 10 generations, and the multi-objective optimization (second step) was terminated after 20 generations. Increasing the number of generations did not significantly improve the results. For each generation, fitness functions were evaluated in parallel on a six-core processor. Multiple processors could be leveraged to decrease the optimization time.

### 6.3.4 Optimization Results

The full optimization procedure completed in approximately 11 hours on a six-core desktop computer. Fitness values for the multi-objective optimization are plotted in Figure 6.14 for each fitness function. As expected, the  $IMEP_n$  error related closely to the peak cylinder pressure. Because of the relationship, the  $f_{total}$  fitness function provided reasonable result and could be used for a single-objective optimization. Other Pareto optimal results sacrificed either  $f_{IMEP_n}$  or  $f_{peak}$  without significant benefit over minimum  $f_{total}$ . Referring to Figure 6.14, the optimization resulted in sparse fitness values along the Pareto front. Density could be improved by including more generations or increasing the population size but would not provide a significantly better solution. Based on the trend in Figure 6.14, the multi-objective optimization approaches a narrow set of Pareto optimal solutions.



**Figure 6.14 Pareto optimal and sub-optimal results from multi-objective optimization (not all fitness values are shown in order to view Pareto optimal results)**

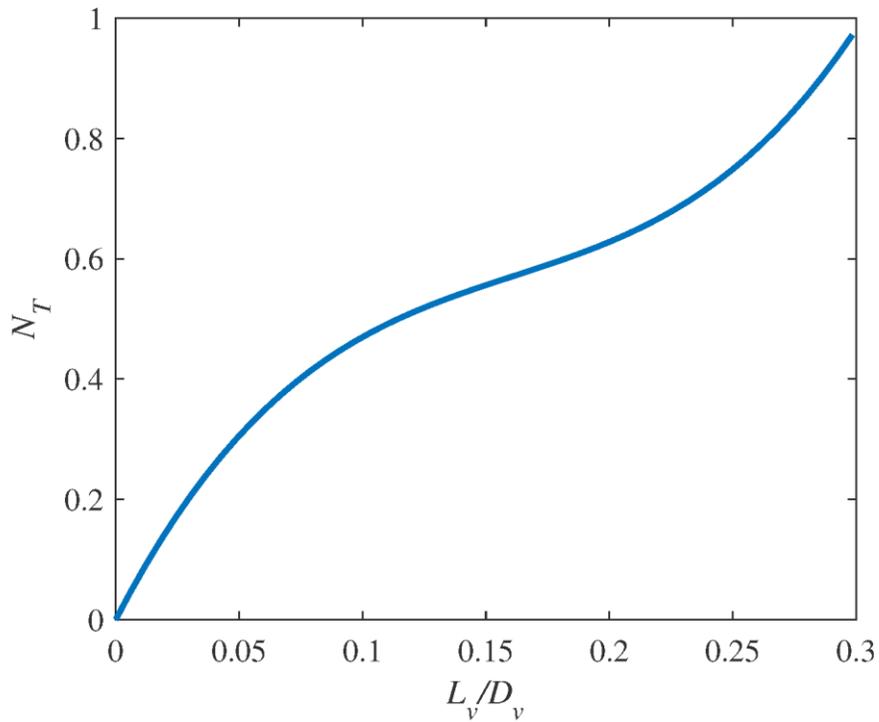
Fitness values are summarized in Table 6.5 for initial parameters, minimum  $f_{total}$ , minimum  $f_{IMEP_n}$ , and minimum  $f_{peak}$ . The single-objective fitness function  $f_{total}$  yielded fitness values near the minimum values found for  $f_{IMEP_n}$  and  $f_{peak}$ , and all improved over the initial parameters. When considering IMEP<sub>n</sub> or peak pressure errors only, the sum of the fitness values increased. Summarized in Table 6.4, the three optimal solutions yielded similar optimal combustion parameters, with some notable differences in tumble number and decay coefficients. The  $f_{total}$  optimal solution required higher tumble numbers which was counter-acted by lower tumble decay coefficients and lower eddy burn up factor  $C_b$ . For minimum  $f_{total}$ , the tumble number coefficients resulted in the spline plotted in Figure 6.15, and from Eq. 6.4, tumble decay coefficients resulted in the tumble decay function plotted in Figure 6.16.

**Table 6.4: Initial and optimized combustion model parameters for minimum IMEP<sub>n</sub> error, minimum peak cylinder pressure error, and minimum combined fitness function**

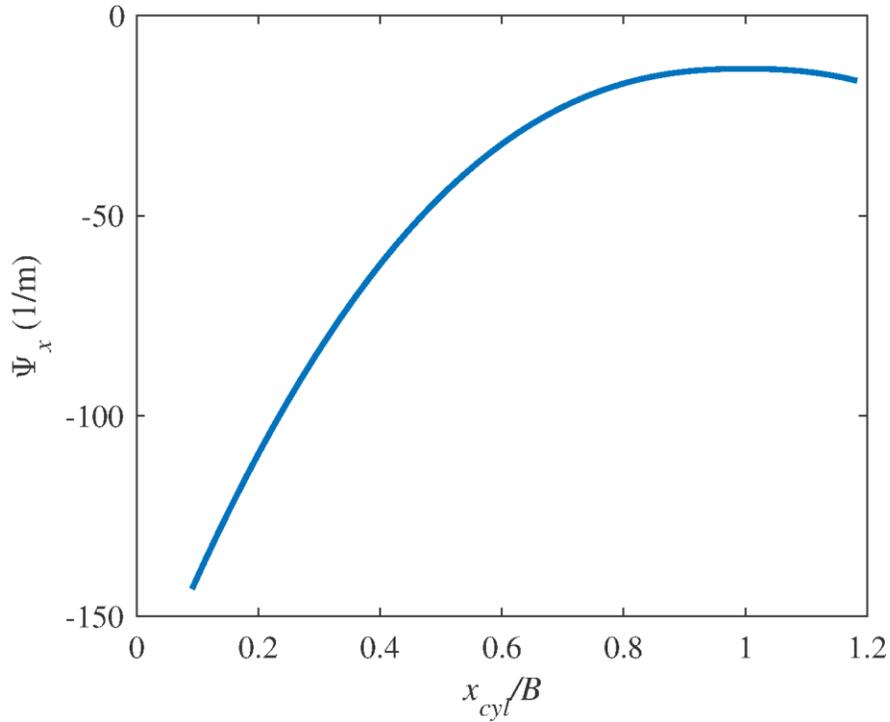
Parameters	Initial	min( $f_{total}$ )	min( $f_{IMEP_n}$ )	min( $f_{peak}$ )
$N_{T,1}$	0.100	0.469	0.112	0.112
$N_{T,2}$	0.300	0.626	0.330	0.356
$N_{T,3}$	0.500	0.972	0.849	0.869
$C_{\psi,1}$	-100	-143	-96.2	-103
$C_{\psi,2}$	2	2.36	2.69	2.68
$C_{\psi,3}$	-10	-13.3	-8.64	-9.15
$C_f$	1	0.794	0.800	0.800
$C_b$	1	1.34	1.44	1.43
$C_{dev}$	1	0.766	0.524	0.774
$r_{f0}$	1 mm	1.83 mm	1.83 mm	1.83 mm
$C_{\varepsilon, valve}$	1	0.386	0.385	0.385
$l_{i, inj}$	5 mm	8.11 mm	8.12 mm	8.12 mm

**Table 6.5: Fitness values for initial parameters, minimum IMEP<sub>n</sub> error, minimum peak cylinder pressure error, and minimum combined fitness function**

Fitness values	Initial	min( $f_{total}$ )	min( $f_{IMEP_n}$ )	min( $f_{peak}$ )
$f_{total}$	1.30	0.533	0.664	0.598
$f_{IMEP_n}$	0.758	0.232	0.226	0.301
$f_{peak}$	0.541	0.300	0.439	0.296

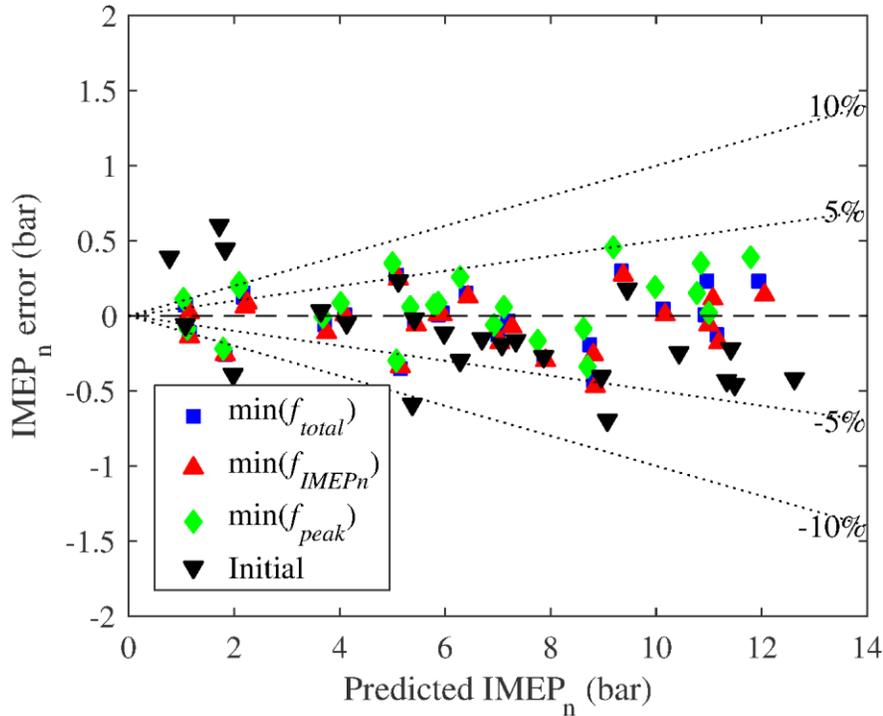


**Figure 6.15: Resulting tumble number cubic spline for minimum  $f_{total}$**



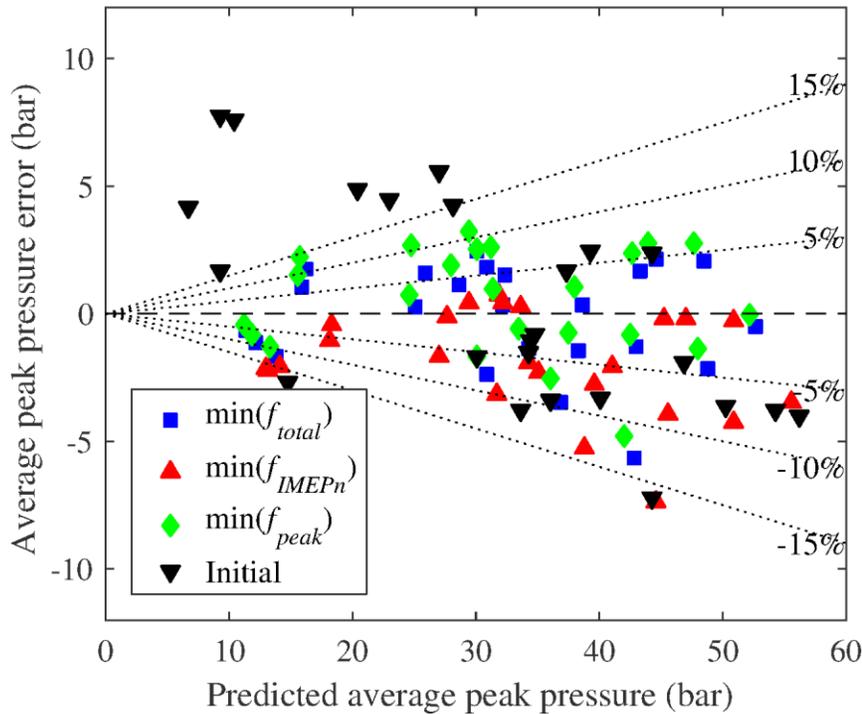
**Figure 6.16: Resulting tumble decay function for minimum  $f_{total}$**

At low load, simulation resulted in higher relative IMEP<sub>n</sub> error but exhibited very little variation between the three optimal solutions as shown in Figure 6.17. The initial parameters, however, resulted significantly higher error at low load. The higher relative error for the optimal parameters can be attributed to higher air mass flow relative error, as can be expected for low measured air mass flow rate. When IMEP<sub>n</sub> was above 4 bar, the simulated IMEP<sub>n</sub> remained within approximately 7% of the measured value with the optimized parameters and within 11% with initial parameters. The differences between the three optimal solutions are more noticeable when IMEP<sub>n</sub> is above 8 bar. Comparing the variation between three solutions at high and low IMEP<sub>n</sub>'s show that optimizing relative error would potentially provide very little improvement at low load and a with a significant increase in absolute error at high loads.



**Figure 6.17: IMEP<sub>n</sub> error at optimization operating points for minimum  $f_{IMEP_n}$ ,  $f_{peak}$ , and  $f_{total}$  fitness functions and initial parameters**

Peak cylinder pressure errors were determined by fitting measured peak pressures as a function of speed and torque and looking up the measured values based on the simulated speed and torque. As shown in Figure 6.18, peak cylinder pressure remained primarily within 15% of measured values for optimized solutions, which is higher than the optimized IMEP<sub>n</sub> relative error. The initial parameters resulted in noticeably higher error at middle and low loads. At high loads, the minimum  $f_{IMEP_n}$  parameters produced similar results as the initial parameters. The relative error with the optimized combustion parameters is acceptable considering the measured cycle-to-cycle variations. For example, at 3000 rpm and 170 Nm, the average measured peak cylinder pressure is 43 bar with a standard deviation of 5 bar for cylinder 1. Comparing errors for each solution in Figure 6.18, sensitivity to combustion parameters can also be observed in the simulated errors.



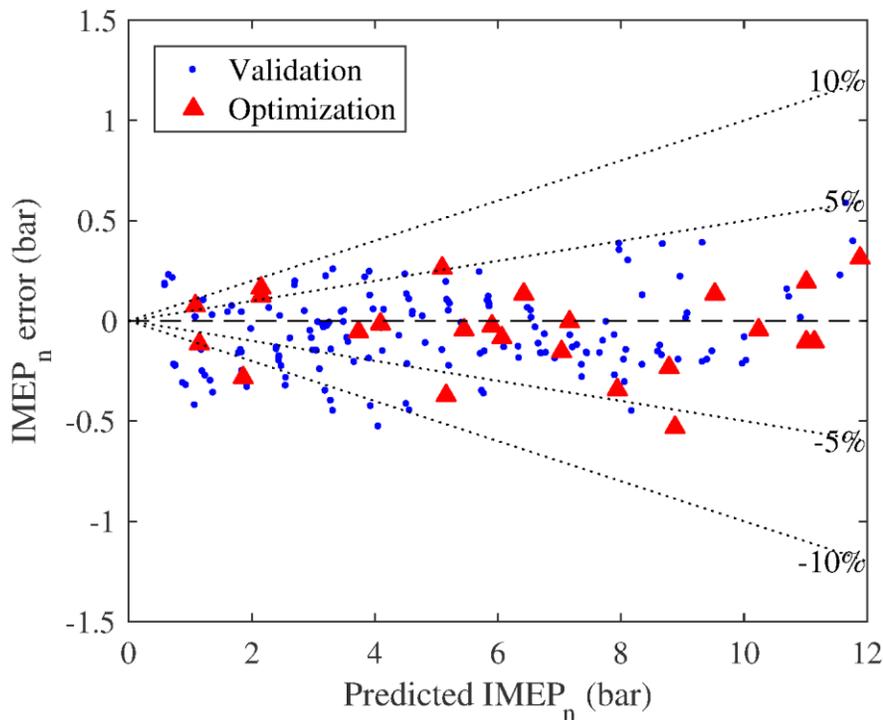
**Figure 6.18: Peak cylinder pressure error averaged across cylinders at optimization operating points for minimum  $f_{IMEP_n}$ ,  $f_{peak}$ , and  $f_{total}$  fitness functions and initial parameters**

Based on the  $IMEP_n$  and peak cylinder pressure errors in Figure 6.17 and Figure 6.18, minimum  $f_{total}$  solution was selected and used for validation. The alternative solutions, minimum  $f_{IMEP_n}$  and  $f_{peak}$ , provided reasonable results when comparing errors, but minimum  $f_{total}$  provided the best tradeoff. Based on the multi-objective optimization,  $f_{total}$  should be sufficient for single-objective optimization in future work. Although the three compared solutions provided similar results,  $f_{IMEP_n}$  or  $f_{peak}$  would not be appropriate for single-objective optimization. However, optimization could behave differently for other engines.

## 6.4 Validation

Using the optimized combustion parameters, the full engine model was simulated at the experimentally-tested operating points. At each operating point, the throttle controller matched

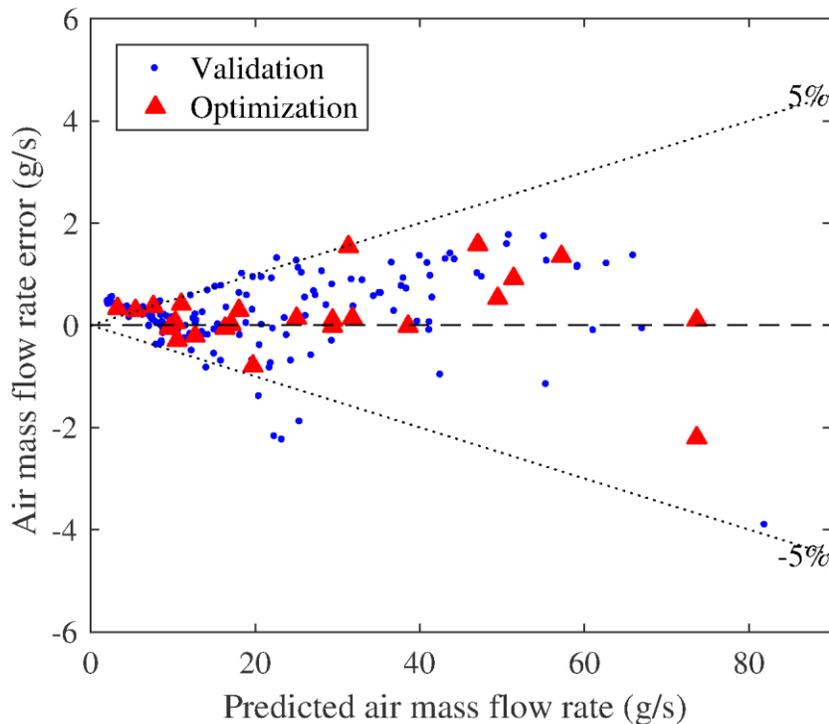
the measured normalized air charge and simulation stopped once steady state criterion had been reached. The resulting steady-state IMEP<sub>n</sub> errors are plotted on Figure 6.19. Simulated IMEP<sub>n</sub> exhibited similar behaviors at the validation and optimization operating points (speed and torque points included in fitness function). At low torque, simulation resulted in higher relative error. For IMEP<sub>n</sub>'s above 6 bar, relative error remained below 6%, with validation points not exceeding errors at the optimization points. If the optimization operating points predominantly produced lower errors than the validation points, the number of optimization points would need to be increased.



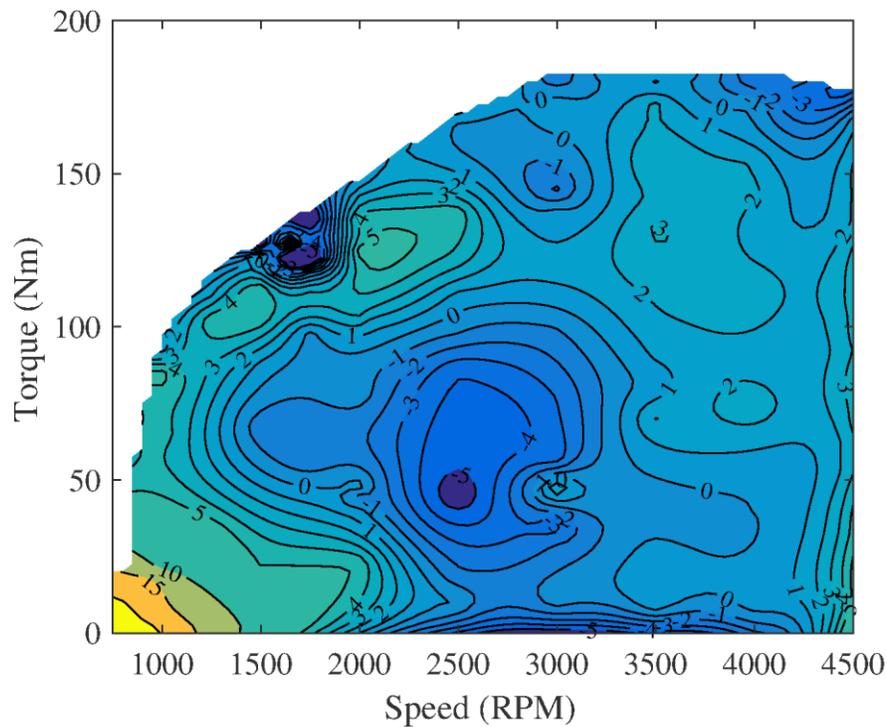
**Figure 6.19: Air flow controlled IMEP<sub>n</sub> error at validation and optimization operating points**

Air mass flow rate also gives insight into simulation errors. By fitting measured air mass flowrate as a function of engine speed and torque, air mass flow rate errors in Figure 6.20 were determined at the simulated engine torques. Air mass flow rate remained predominantly within

5% of the measured data. Referring to Figure 6.21, the largest relative errors were found near the idle condition. The highest relative error ( $\approx 20\%$ ) translates in to an absolute error of approximately 0.5 g/s, which is small considering the highest measured flow rate to be near 80 g/s. The grouping of errors near the lowest flow rates explains the higher IMEP<sub>n</sub> error at low load. Overall, the high relative error at low flow rates can likely be attributed to measurement error. At higher flow rates, the IMEP<sub>n</sub> errors relate more to the burn rate model's ability to predict cylinder pressure across wide range of operating conditions.

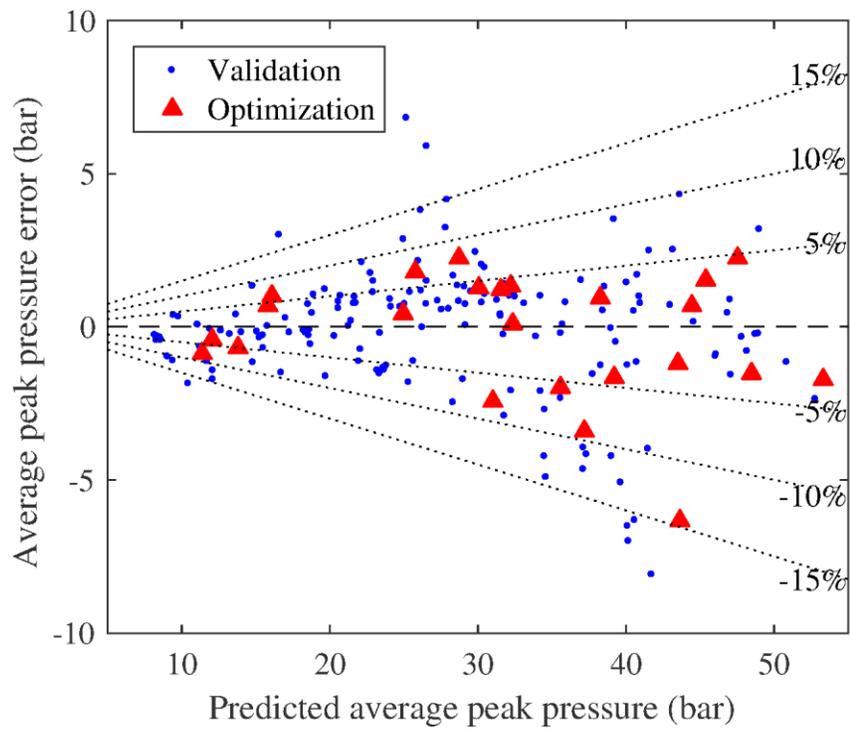


**Figure 6.20: Air mass flow rate error (derived from measured air mass flow fit) at validation and optimization operating points**

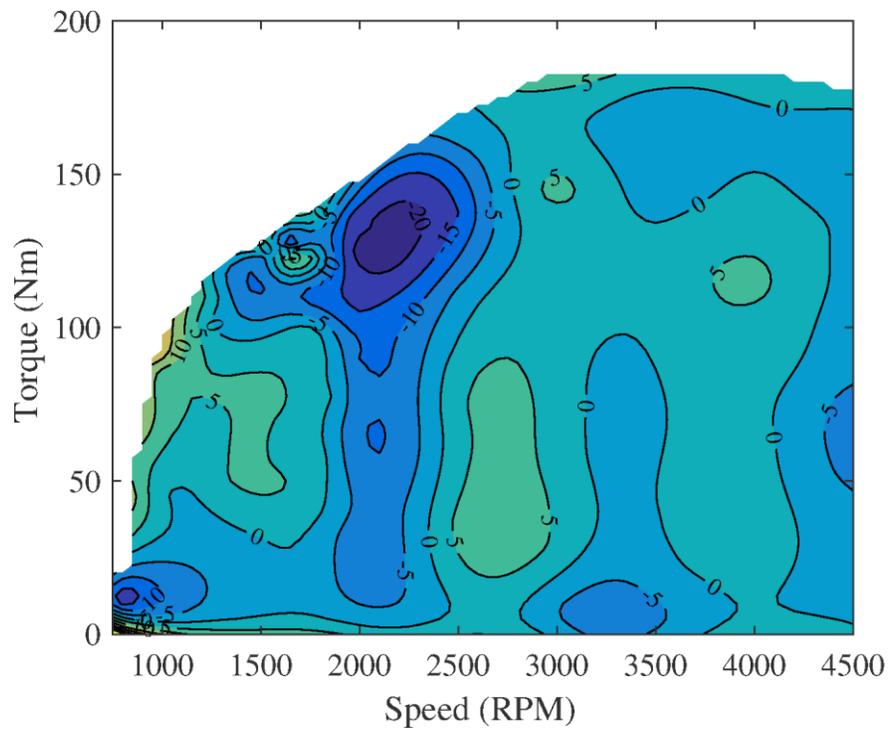


**Figure 6.21: Percent air mass flow rate error at simulated engine speed and torque**

The peak pressure errors shown in Figure 6.22 remained within 15% of the measured values for the majority of the tested operating points and had a maximum error of 27%. With an exception to a small number of outliers, simulation produced similar errors at the optimization and validation points. Referring to Figure 6.23, the largest relative errors were found around 2000 rpm and 130 Nm, which is reflected in the air flow (Figure 6.21) and IMEP<sub>n</sub> error. As noted in the optimization results, the relative error is acceptable when considering the measured peak cylinder pressure cycle-to-cycle variation. Also note that the errors for each cylinder were averaged. Of the four cylinders, Cylinder 1 produced a maximum percent error of 30%. The differences in peak pressure depend on several factors. Including all cylinders in the combustion parameter tuning ensures that burn rate and flow prediction reflects every cylinder's behavior.

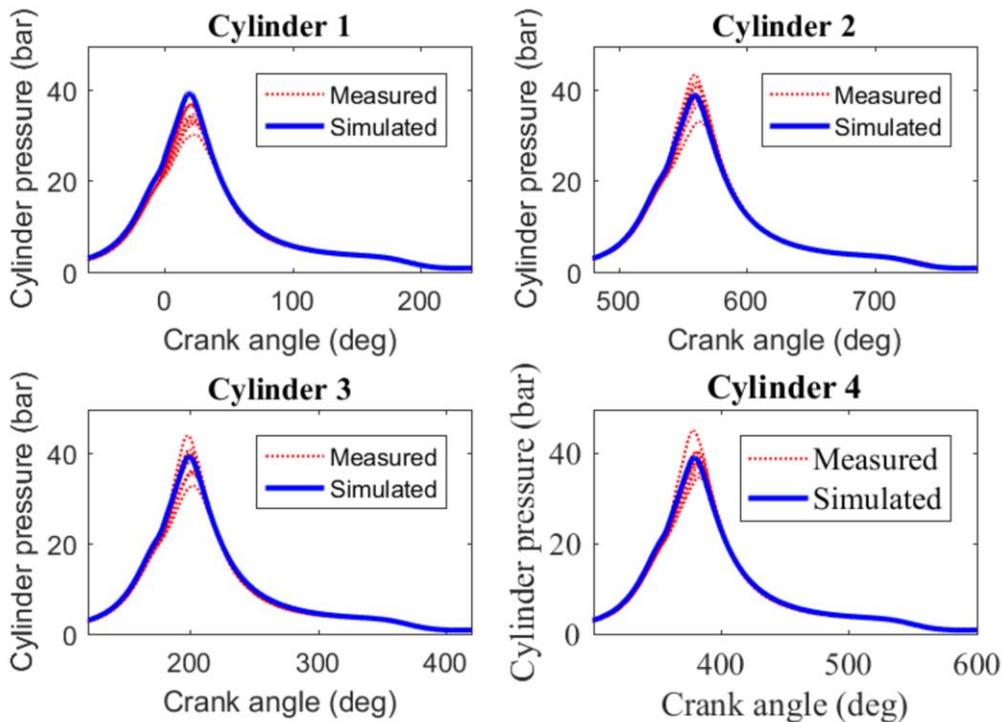


**Figure 6.22: Peak cylinder pressure error averaged across cylinders at validation and optimization operating points**



**Figure 6.23: Percent peak cylinder pressure error averaged across cylinders at simulated engine speed and torque**

In Figure 6.24, simulated cylinder pressures are compared to 10 measured cycles at the 2000 rpm and 130 Nm operating point, near the maximum relative peak cylinder pressure error. Simulation produced a higher peak pressure than all 10 measured cycles for cylinder 1. For the remaining cylinders, however, the simulated peak pressure was closer to the mean value. A noticeable difference can be observed between the measured pressure profiles for each cylinder, while simulation resulted in nearly uniform profiles. Variation in trapped air mass and turbulence can explain both the cycle-to-cycle and cylinder-to-cylinder differences. The flow model would require additional tuning to better represent the variation between cylinders. A statistics-based method would be needed to model cycle-to-cycle variations.



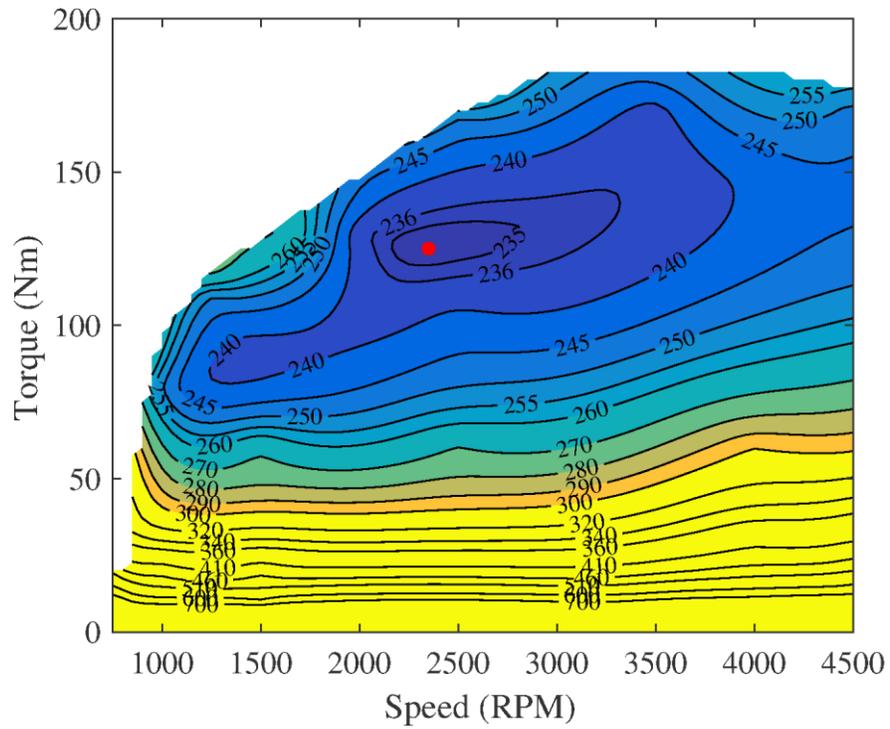
**Figure 6.24: Measured (10 cycles) and simulated cylinder pressures at 2000 rpm and 130 Nm**

Engine efficiency is frequently expressed in terms of Brake Specific Fuel Consumption (BSFC), which is the ratio of fuel mass flow rate to mechanical power. Based on a surface fit of

the simulated BSFC, simulation resulted in a minimum BSFC of 226 g/(kW·h) at 2200 rpm and 128 Nm, compared to a measured minimum BSFC of 234 g/(kW·h) at 2350 rpm and 125 Nm. The simulated BSFC (Figure 6.26) followed similar trends as the measured data (Figure 6.25). For torques below 50 Nm, friction and pumping losses begin to dominate the available power, increasing BSFC. At high torque, ignition timing must be retarded below the Maximum Brake Torque (MBT) spark to avoid knock, which can be observed in the measured peak pressure. The Skyactiv engine also runs fuel rich at high loads and speeds (Figure A.4) to reduce probability of knock and control combustion temperature. Retarding spark and decreasing the relative air-fuel ratio causes an increase in BSFC which can be observed in both the measured and simulated data. Comparing Figure 6.25 and Figure 6.26, the simulated BSFC error began to increase at the highest speed and torque points. Based on the combustion model results,  $IMEP_n$  is sensitive to changes in the combustion model at high speeds and loads. If Wide Open Throttle (WOT) performance is the primary focus, weights could be included in the fitness function to favor WOT conditions.

From a calibration standpoint, differences between simulated and measured cylinder pressure can be viewed as spark advance error. For example, simulation resulted in a  $IMEP_n$  error of 0.5 bar at the 4500 rpm and 185 Nm operating point. By advancing spark angle by 3 degrees, the measured  $IMEP_n$  could be met while operating at the measured air flow rate. As shown in Figure 6.27, the engine was not operating near MBT spark timing in order to avoid knock, making  $IMEP_n$  more sensitive to spark advance. In general,  $IMEP_n$  is less sensitive to changes in spark angle near MBT, and as spark advance decreases, small changes in ignition angle become more significant. Based on the results in Figure 6.27, shifting the  $IMEP_n$  and peak cylinder pressure curves by 3 degrees of spark advance would provide a close match to the

measured values. The effect if increasing spark advance by 3 degrees on cylinder pressure is shown in Figure 6.28.



**Figure 6.25: Measured BSFC (g/(kW·h))**

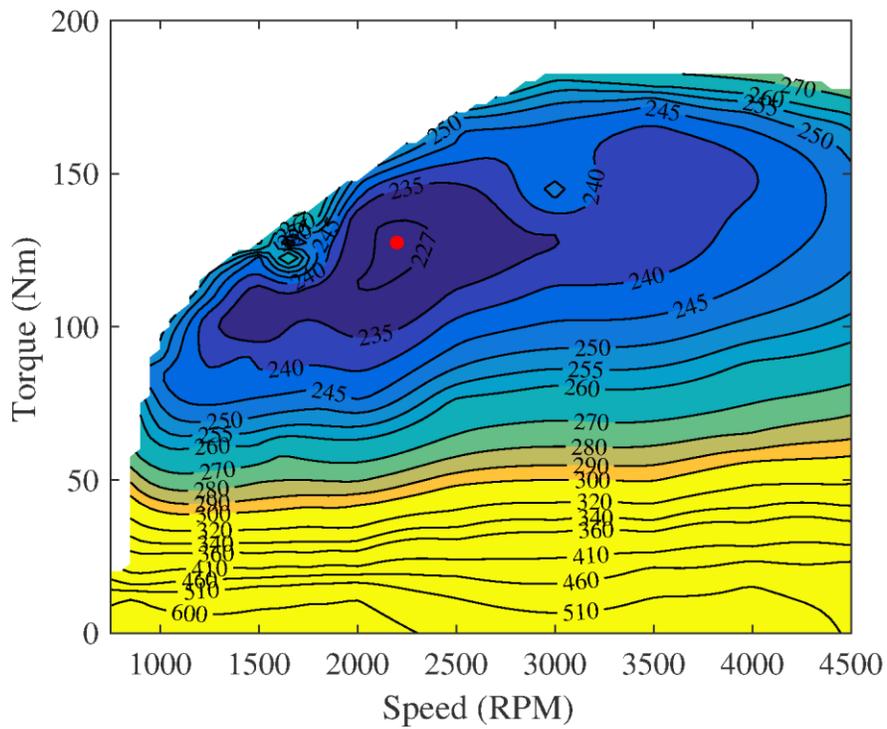


Figure 6.26: Simulated BSFC (g/(kW·h))

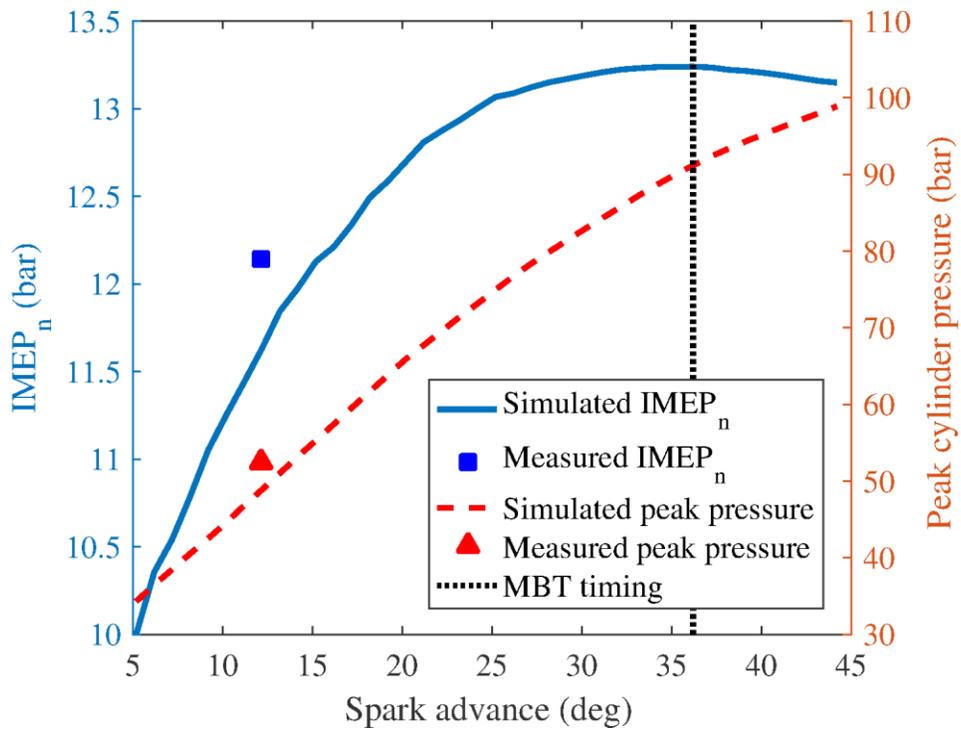
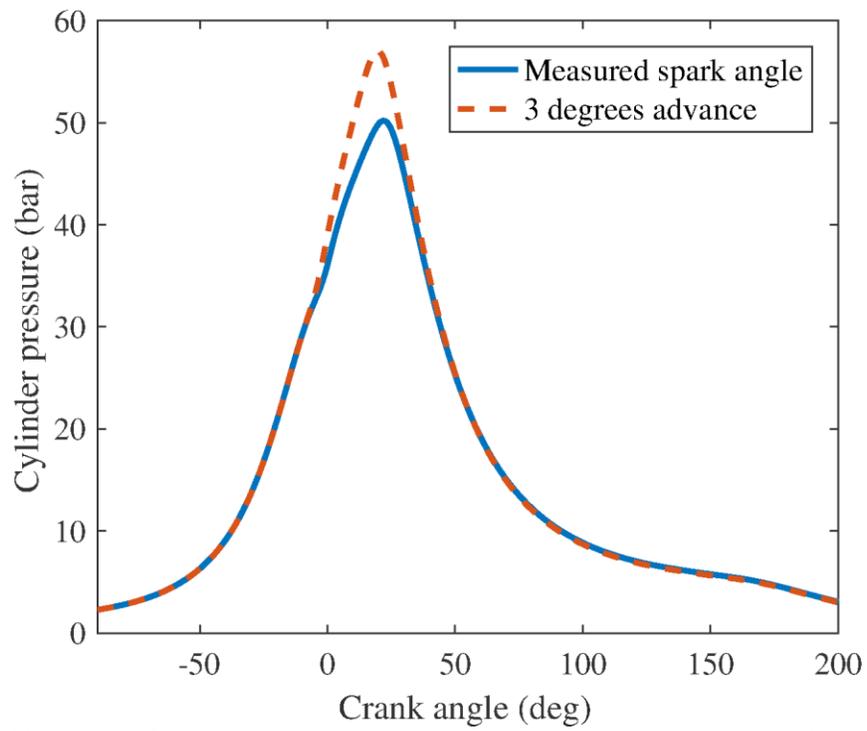


Figure 6.27: IMEP<sub>n</sub> and peak pressure averaged across cylinders as a function of spark advance at 4500 rpm and 0.92 normalized air charge (185 Nm torque at measured spark advance)



**Figure 6.28: Simulated Cylinder 1 pressure using measured spark angle and 3 additional degrees advance at 4500 rpm and 0.92 normalized air charge (185 Nm torque at measured spark advance)**

## **CHAPTER 7**

### **CONCLUSIONS**

#### **7.1 Summary**

To meet stringent government regulations and customer demands, internal combustion engines have become increasingly more complex. Emission and fuel consumption reduction strategies require hardware optimization, control optimization, and calibration. Appropriate hardware combinations can only be selected after calibration, through simulation or experiment. Adding controllable features are typically required to improve efficiency, and as a result, calibration becomes significantly more time consuming. Model-based design reduces the testing burden by replacing experiments with simulations. Depending on the design task, the appropriate level of fidelity must be selected. While multidimensional CFD simulation aid detailed engine geometry optimization during the early stages of development, simplified 1D flow and quasi-dimensional combustion models greatly reduce simulation time and predict engine performance. Control-oriented models, mean-value models for example, run in real-time and can be deployed on ECUs.

Simplified 1D flow model coupled with a quasi-dimensional combustion model can be utilized for desktop computer-based calibration and hardware design. The 1D flow can simulate manifold tuning and pumping losses. When tuned correctly, semi-predictive combustion models have shown to accurately predict cylinder pressure based on the air and fuel flow. The simplified phenomenological model can be used to optimize configurations (e.g. intake runner lengths,

exhaust runner length, and turbocharger size) and control inputs (e.g. cam phasing, injection timing, and spark advance). To be accurate, however, the model must be tuned to match experimental data at a relatively small number of operating points.

In the research presented, 1D flow and quasi-dimensional combustion models were developed in Simulink. Although Matlab/Simulink is frequently used in the automotive industry for model-based design, a high-fidelity engine model is currently not available in Simulink. The presented architecture allows users to connect engine components in the Simulink environment and interact with built-in blocks or in-house models. Block connection and parameter information are collected at the start of simulation and communicated to a C++ S-Function, effectively using the Simulink as a user interface. Using the modeling structure, new combustion and flow models can be implemented by creating C++ and Matlab classes for a new component or adding options to the existing models.

The presented semi-predictive spark ignition combustion correlates the burn rate to combustion chamber geometry, laminar flame speed, and turbulence. Laminar flame speed depends on fuel and not a specific engine, and combustion chamber geometry considerations can be derived from CAD drawings. Turbulence depends on intake mean charge motion, engine speed, fuel injection, and compression ratio. By lumping in-cylinder flow into a single volume, turbulence cannot be predicted based on dissimilar engines or a CAD model but must be tuned. The proposed model includes swirl and tumble flow numbers that can be experimentally measured, derived from CFD simulations, or tuned to match cylinder pressure. The model also includes burn rate and turbulence kinetic energy dissipation tuning constants that do not depend on engine operating conditions. The constants can be tuned to match an engine's cylinder

pressure at a relatively small number of operating points and then used to predict performance throughout the engine's operating region.

Experimental engine data provided by the EPA for a 2.0 L Mazda Skyactiv-G engine was used to tune the combustion model. Without intake tumble data, tumble numbers required tuning, resulting in a total of 12 unknown parameters. Using the multi-objective genetic algorithm, burn rate and turbulence parameters were determined by minimizing  $IMEP_n$  and peak cylinder pressure errors at 24 steady state operating points. Computation time was reduced by eliminating the intake and exhaust manifold models during optimization. At the start of optimization, the full Simulink engine model recorded port flow, injector flow, and cylinder states for the final cycle. During optimization, the reduced model replicated intake and exhaust flow simulation with the recorded data, allowing combustion parameters to be tested without reproducing the steady state flow simulation.

Once tuned, the full model was validated at each of the tested operating points. Operating points and validation points exhibited similar behaviors. The larger  $IMEP_n$  and air flow errors near the idle condition can likely be attributed to error in the measured air mass flow rate. For  $IMEP_n$ 's above 6 bars, the majority of the simulations predicted  $IMEP_n$  within 5% of the measured value and a maximum error of 6%. Peak cylinder pressure errors were averaged across each cylinder. The model predicted average peak cylinder pressure within 15% of the measured values for a majority of the test points and a maximum error of 27%. Considering the measured cycle-to-cycle variations in peak pressure, the relative error of the simulation results is within an acceptable range. Based on a surface fit of the simulated BSFC, simulation resulted in a minimum BSFC of 226 g/(kW·h) at 2200 rpm and 128 Nm, compared to a measured minimum BSFC of 234 g/(kW·h) at 2350 rpm and 125 Nm.

## 7.2 Future Works

The presented model accounted for tumble and swirl mean charge motion but was only validated with tumble flow. Swirl and tumble are treated independently. In the future, the assumption needs to be evaluated at a wide range of operating conditions using swirl control strategies (i.e. swirl flap and variable valve lift). Introducing a new control input requires additional tuning. Swirl and tumble numbers must be experimentally measured or optimized as a function of the control input and valve lift. Predictive methods not requiring a table of flow numbers can be explored as well.

Knock and emissions limit spark ignition engine performance at high loads. Therefore, accurate virtual calibration requires knock and emissions prediction and can be added to the presented Simulink model. Much like the burn rate, knock and emissions cannot be accurately predicted without tuning. The knock and emissions model can be tuned in a similar manner as the presented burn rate model. The procedure may need to be modified to account for the link between gas constituents at the start of each cycle and the optimization parameters. The interaction depends primarily on the residual gas fraction. One possible solution would be to update the reduced model more frequently, requiring more full engine simulations.

## REFERENCES

- [1] J. B. Heywood, *Internal combustion engine fundamentals*. McGraw-Hill, 1988.
- [2] E. Hendricks, "Engine Modelling for Control Applications: A Critical Survey," *Meccanica*, vol. 32, no. 5, pp. 387–396, Oct. 1997.
- [3] J. D. Powell, "A Review of IC Engine Models for Control System Design," in *Proc. of the 10th World Congress on Automatic Control*, Munich, Germany, 1987, vol. 3.
- [4] B. K. Powell, "A dynamic model for automotive engine control analysis," in *1979 18th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, 1979, vol. 2, pp. 120–126.
- [5] R. G. DeLosh, K. J. Brewer, L. H. Buch, T. F. W. Ferguson, and W. E. Tobler, "Dynamic Computer Simulation of a Vehicle with Electronic Engine Control," SAE International, Warrendale, PA, SAE Technical Paper 810447, Feb. 1981.
- [6] D. J. Dohner, "A Mathematical Engine Model for Development of Dynamic Engine Control," SAE International, Warrendale, PA, SAE Technical Paper 800054, Feb. 1980.
- [7] D. J. Dobner, "Dynamic Engine Models for Control Development, Part 1: Nonlinear and Linear Model Formulation," *International Journal of Vehicle Design*, vol. SP4, 1982.
- [8] D. J. Dobner and R. D. Fruechte, "An Engine Model for Dynamic Engine Control Development," in *American Control Conference, 1983*, 1983, pp. 73–78.
- [9] D. Cho and J. K. Hedrick, "Automotive Powertrain Modeling for Control," *J. Dyn. Sys., Meas., Control*, vol. 111, no. 4, pp. 568–576, Dec. 1989.
- [10] J. J. Moskwa, "Automotive engine modeling for real time control," Thesis, Massachusetts Institute of Technology, 1988.
- [11] J. J. Moskwa and J. K. Hedrick, "Automotive Engine Modeling for Real Time Control Application," in *American Control Conference, 1987*, 1987, pp. 341–346.
- [12] J. J. Moskwa and J. K. Hedrick, "Modeling and Validation of Automotive Engines for Control Algorithm Development," *J. Dyn. Sys., Meas., Control*, vol. 114, no. 2, pp. 278–285, Jun. 1992.
- [13] E. Hendricks and S. C. Sorenson, "Mean Value Modelling of Spark Ignition Engines," SAE International, Warrendale, PA, 900616, Feb. 1990.

- [14] M. Müller, E. Hendricks, and S. C. Sorenson, "Mean Value Modelling of Turbocharged Spark Ignition Engines," SAE International, Warrendale, PA, 980784, Feb. 1998.
- [15] K. Steigerwald, B. Zelenka, G. Hohenberg, and W. M. Silvis, "Development of an Intake Flow Based Model Calculating Real Time Exhaust Flow by Accounting for Filling and Emptying of the Engine Manifolds," SAE International, Warrendale, PA, SAE Technical Paper 2007-1-324, Apr. 2007.
- [16] R. S. Benson, *The Thermodynamics and Gas Dynamics of Internal Combustion Engines, I.* Oxford University Press, Incorporated, 1982.
- [17] P. De Haller, "The Application of a Graphical Method to Dynamic Problems in Gases," *Sulzer Technical Review*, vol. 27, no. 1, pp. 6-24, 1945.
- [18] E. Jenny, "Unidimensional transient flow with consideration of friction, heat transfer and change of section," *The Brown Boveri Review*, vol. 37, no. 11, pp. 447-461, Nov. 1950.
- [19] R. S. Benson, R. D. Garg, and D. Woollatt, "A numerical solution of unsteady flow problems," *International Journal of Mechanical Sciences*, vol. 6, no. 1, pp. 117-144, Feb. 1964.
- [20] R. S. Benson, "A Comprehensive Digital Computer Program to Simulate a Compression Ignition Engine Including Intake and Exhaust Systems," SAE International, Warrendale, PA, SAE Technical Paper 710173, Feb. 1971.
- [21] G. P. Blair, "An Alternative Method for the Prediction of Unsteady Gas Flow Through the Internal Combustion Engine," SAE International, Warrendale, PA, SAE Technical Paper 911850, Sep. 1991.
- [22] G. P. Blair, *Design and Simulation of Four-stroke Engines.* SAE, 1999.
- [23] G. P. Blair, *Design and Simulation of Two-stroke Engines.* SAE, 1996.
- [24] G. P. Blair and S. J. Magee, "Non-Isentropic Analysis of Varying Area Flow in Engine Ducting," SAE International, Warrendale, PA, 932399, Sep. 1993.
- [25] G. P. Blair, "Correlation of an Alternative Method for the Prediction of Engine Performance Characteristics with Measured Data," SAE International, Warrendale, PA, 930501, Mar. 1993.
- [26] J. D. Ledger, "A Finite-Difference Approach for Solving the Gas Dynamics in an Engine Exhaust," *Journal of Mechanical Engineering Science*, vol. 17, no. 3, pp. 125-132, Jun. 1975.
- [27] M. Takizawa, T. Uno, T. Oue, and T. Yura, "A Study of Gas Exchange Process Simulation of an Automotive Multi-Cylinder Internal Combustion Engine," SAE International, Warrendale, PA, SAE Technical Paper 820410, Feb. 1982.

- [28] T. Bulaty and H. Niessner, "Calculation of 1-D Unsteady Flows in Pipe Systems of I.C. Engines," *Journal of Fluids Engineering*, vol. 107, no. 3, p. 407, 1985.
- [29] M. Poloni, D. E. Winterbone, and J. R. Nichols, "Comparison of unsteady flow calculations in a pipe by the method of characteristics and the two-step differential Lax-Wendroff method," *International Journal of Mechanical Sciences*, vol. 29, no. 5, pp. 367–378, 1987.
- [30] J. M. Arnau, S. Jerez, L. Jódar, and J. V. Romero, "A semi-implicit conservation element–solution element method for chemical species transport simulation to tapered ducts of internal combustion engine," *Mathematics and Computers in Simulation*, vol. 73, no. 1–4, pp. 28–37, Nov. 2006.
- [31] F. Bozza and A. Gimelli, "A Comprehensive 1D Model for the Simulation of a Small-Size Two-Stroke SI Engine," SAE International, Warrendale, PA, SAE Technical Paper 2004-1-999, Mar. 2004.
- [32] S. Jerez, J. V. Romero, M. D. Roselló, and J. M. Arnau, "A nonuniform mesh semi-implicit CE–SE method modelling unsteady flow in tapered ducts," *Mathematics and Computers in Simulation*, vol. 76, no. 1–3, pp. 94–98, Oct. 2007.
- [33] S. Jerez, J. V. Romero, M. D. Roselló, and F. J. Arnau, "A semi-implicit space-time CE-SE method to improve mass conservation through tapered ducts in internal combustion engines," *Mathematical and Computer Modelling*, vol. 40, no. 9–10, pp. 941–951, Nov. 2004.
- [34] I. Wiebe, "Semi-empirical expression for combustion rate in engines," in *Conference on Piston Engines*, USSR, 1956, pp. 185–191.
- [35] J. A. Gatowski, E. N. Balles, K. M. Chun, F. E. Nelson, J. A. Ekchian, and J. B. Heywood, "Heat Release Analysis of Engine Pressure Data," SAE International, Warrendale, PA, SAE Technical Paper 841359, Oct. 1984.
- [36] H. M. Cheung and J. B. Heywood, "Evaluation of a One-Zone Burn-Rate Analysis Procedure Using Production SI Engine Pressure Data," SAE International, Warrendale, PA, SAE Technical Paper 932749, Oct. 1993.
- [37] Y. Yeliana, C. Cooney, J. Worm, D. J. Michalek, and J. D. Naber, "Estimation of double-Wiebe function parameters using least square method for burn durations of ethanol-gasoline blends in spark ignition engine over variable compression ratios and EGR levels," *Applied thermal engineering*, vol. 31, no. 14–15, pp. 2213–2220, 2011.
- [38] 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 1999-1-218, Mar. 1999.

- [39] M. F. J. Brunt, H. Rai, and A. L. Emtage, “The Calculation of Heat Release Energy from Engine Cylinder Pressure Data,” SAE International, Warrendale, PA, SAE Technical Paper 981052, Feb. 1998.
- [40] K. M. Chun and J. B. Heywood, “Estimating Heat-Release and Mass-of-Mixture Burned from Spark-Ignition Engine Pressure Data,” *Combustion Science and Technology - COMBUST SCI TECHNOL*, vol. 54, pp. 133–143, 1987.
- [41] C. W. Hong, “A Combustion Correlation for Spark-Ignition Engine Simulation Under Steady and Transient Conditions,” SAE International, Warrendale, PA, 901602, Sep. 1990.
- [42] N. C. Blizard and J. C. Keck, “Experimental and Theoretical Investigation of Turbulent Burning Model for Internal Combustion Engines,” SAE International, Warrendale, PA, SAE Technical Paper 740191, Feb. 1974.
- [43] J. C. Keck, “Turbulent flame structure and speed in spark-ignition engines,” *Symposium (International) on Combustion*, vol. 19, no. 1, pp. 1451–1466, 1982.
- [44] G. P. Beretta, M. Rashidi, and J. C. Keck, “Turbulent flame propagation and combustion in spark ignition engines,” *Combustion and Flame*, vol. 52, pp. 217–245, 1983.
- [45] J. C. Keck, J. B. Heywood, and G. Noske, “Early Flame Development and Burning Rates in Spark Ignition Engines and Their Cyclic Variability,” SAE International, Warrendale, PA, SAE Technical Paper 870164, Feb. 1987.
- [46] R. J. Tabaczynski, C. R. Ferguson, and K. Radhakrishnan, “A Turbulent Entrainment Model for Spark-Ignition Engine Combustion,” SAE International, Warrendale, PA, SAE Technical Paper 770647, Feb. 1977.
- [47] R. J. Tabaczynski, F. H. Trinker, and B. A. S. Shannon, “Further refinement and validation of a turbulent flame propagation model for spark-ignition engines,” *Combustion and Flame*, vol. 39, no. 2, pp. 111–121, Oct. 1980.
- [48] M. Sjeric, D. Kozarac, and M. Bogensperger, “Implementation of a Single Zone  $k$ - $\epsilon$  Turbulence Model in a Multi Zone Combustion Model,” SAE International, Warrendale, PA, SAE Technical Paper 2012-1–130, Apr. 2012.
- [49] A. Agarwal, Z. S. Filipi, D. N. Assanis, and D. M. Baker, “Assessment of Single- and Two-Zone Turbulence Formulations for Quasi-Dimensional Modeling of Spark-Ignition Engine Combustion,” *Combustion Science and Technology*, vol. 136, no. 1–6, pp. 13–39, Jul. 1998.
- [50] A. Medina, P. L. Curto-Risso, A. C. Hernández, L. Guzmán-Vargas, F. Angulo-Brown, and A. K. Sen, *Quasi-Dimensional Simulation of Spark Ignition Engines*, 1st ed. Springer-Verlag, 2014.

- [51] P. O. Witze, "The Effect of Spark Location on Combustion in a Variable-Swirl Engine," SAE International, Warrendale, PA, SAE Technical Paper 820044, Feb. 1982.
- [52] Y. Yeliana, C. Cooney, J. Worm, and J. D. Naber, "The Calculation of Mass Fraction Burn of Ethanol-Gasoline Blended Fuels Using Single and Two-Zone Models," SAE International, Warrendale, PA, SAE Technical Paper 2008-1-320, Apr. 2008.
- [53] I. Arsie, C. Pianese, and G. Rizzo, "Models for the Prediction of Performance and Emissions in a Spark Ignition Engine - A Sequentially Structured Approach," SAE International, Warrendale, PA, SAE Technical Paper 980779, Feb. 1998.
- [54] R. R. Raine, C. R. Stone, and J. Gould, "Modeling of nitric oxide formation in spark ignition engines with a multizone burned gas," *Combustion and Flame*, vol. 102, no. 3, pp. 241-255, Aug. 1995.
- [55] S. G. Poulos and J. B. Heywood, "The Effect of Chamber Geometry on Spark-Ignition Engine Combustion," SAE International, Warrendale, PA, SAE Technical Paper 830334, Feb. 1983.
- [56] T. K. Jensen and J. Schramm, "A Three-Zone Heat Release Model for Combustion Analysis in a Natural Gas SI Engine. -Effects of Crevices and Cyclic Variations on UHC Emissions," SAE International, Warrendale, PA, SAE Technical Paper 2000-01-2802, Oct. 2000.
- [57] M. Chapman, "Two Dimensional Numerical Simulation of Inlet Manifold Flow in a Four Cylinder Internal Combustion Engine," SAE International, Warrendale, PA, SAE Technical Paper 790244, Feb. 1979.
- [58] P. Li, "A two-dimensional intake manifold flow simulation," *Mathematical Modelling*, vol. 8, pp. 437-442, 1987.
- [59] M. Leschziner and K. Dimitriadis, "Numerical simulation of three-dimensional turbulent flow in exhaust-manifold junctions," presented at the International Conference Computers in Engine Technology, Proceedings of the Institution of Mechanical Engineers, 1987, pp. 183-190.
- [60] Y. Zhao and D. E. Winterbone, "A Study of Multi-Dimensional Gas Flow in Engine Manifolds," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 208, no. 2, pp. 139-145, Apr. 1994.
- [61] R. Sinclair, T. Strauss, and P. Schindler, "Code Coupling, a New Approach to Enhance CFD Analysis of Engines," SAE International, Warrendale, PA, SAE Technical Paper 2000-1-660, Mar. 2000.
- [62] M. Claywell, D. Horkheimer, and G. Stockburger, "Investigation of Intake Concepts for a Formula SAE Four-Cylinder Engine Using 1D/3D (Ricardo WAVE-VECTIS) Coupled Modeling Techniques," SAE International, Warrendale, PA, SAE Technical Paper 2006-01-3652, Dec. 2006.

- [63] Z. Liu, S. F. Benjamin, C. A. Roberts, H. Zhao, and A. Arias-Garcia, "A Coupled 1D/3D Simulation for the Flow Behaviour inside a Close-Coupled Catalytic Converter," SAE International, Warrendale, PA, SAE Technical Paper 2003-01-1875, May 2003.
- [64] G. Montenegro and A. Onorati, "Modeling of Silencers for I.C. Engine Intake and Exhaust Systems by Means of an Integrated 1D-multiD Approach," SAE International, Warrendale, PA, SAE Technical Paper 2008-1-677, Apr. 2008.
- [65] G. Montenegro, A. D. Torre, and A. Onorati, "A general 3D cell method for the acoustic modelling of perforates with sound absorbing material for I.C. engine exhaust systems," pp. 3319–3326, 2010.
- [66] G. Montenegro, A. Onorati, T. Cerri, and A. D. Torre, "A Quasi-3D Model for the Simulation of the Unsteady Flows in I.C. Engine Pipe Systems," SAE International, Warrendale, PA, SAE Technical Paper 2012-1-675, Apr. 2012.
- [67] D. C. Haworth, "Large-Eddy Simulation of In-Cylinder Flows," *Oil & Gas Science and Technology*, vol. 54, no. 2, pp. 175–185, Mar. 1999.
- [68] F. Brusiani, S. Falfari, and G. Cazzoli, "Tumble Motion Generation in Small Gasoline Engines: A New Methodological Approach for the Analysis of the Influence of the Intake Duct Geometrical Parameters," *Energy Procedia*, vol. 45, pp. 997–1006, 2014.
- [69] P. G. Hill and D. Zhang, "The effects of swirl and tumble on combustion in spark-ignition engines," *Progress in Energy and Combustion Science*, vol. 20, no. 5, pp. 373–429, 1994.
- [70] T. A. Baritaud, J. M. Duglos, and A. Fusco, "Modeling turbulent combustion and pollutant formation in stratified charge SI engines," *Symposium (International) on Combustion*, vol. 26, no. 2, pp. 2627–2635, 1996.
- [71] J. Wallesten, A. Lipatnikov, and J. Chomiak, "Modeling of stratified combustion in a direct-ignition, spark-ignition engine accounting for complex chemistry," *Proceedings of the Combustion Institute*, vol. 29, no. 1, pp. 703–709, 2002.
- [72] Z. Tan and R. D. Reitz, "An ignition and combustion model based on the level-set method for spark ignition engine multidimensional modeling," *Combustion and Flame*, vol. 145, no. 1–2, pp. 1–15, Apr. 2006.
- [73] L. Liang and R. D. Reitz, "Spark Ignition Engine Combustion Modeling Using a Level Set Method with Detailed Chemistry," SAE International, Warrendale, PA, SAE Technical Paper 2006-1-243, Apr. 2006.
- [74] G. Stiesch, *Modeling Engine Spray and Combustion Processes*, Softcover reprint of the original 1st edition. Berlin; New York: Springer, 2013.
- [75] S. Turns, *An Introduction to Combustion: Concepts and Applications*, 3 edition. New York: McGraw-Hill Science/Engineering/Math, 2011.

- [76] F. J. Zeleznik and S. Gordon, *Simultaneous least-squares approximation of a function and its first integrals with application to thermodynamic data*. Washington, DC: National Aeronautics and Space Administration, 1961.
- [77] B. J. M. Sanford Gordon, "Computer program for calculation of complex chemical equilibrium compositions and applications. Part 1: Analysis," 1994.
- [78] S. G. Bonnie J. McBride, "Computer Program for Calculation of Complex Chemical Equilibrium Compositions and Applications," 1996.
- [79] A. Burcat, B. Ruscic, Chemistry, and T.-I. I. of Tech, "Third Millenium Ideal Gas and Condensed Phase Thermochemical Database for Combustion (with Update from Active Thermochemical Tables).," Argonne National Laboratory (ANL), ANL-05/20, Jul. 2005.
- [80] J. van Leersum, "A numerical model of a high performance two-stroke engine," *Applied Numerical Mathematics*, vol. 27, no. 1, pp. 83–108, May 1998.
- [81] J. R. Serrano, F. J. Arnau, P. Piqueras, A. Onorati, and G. Montenegro, "1D gas dynamic modelling of mass conservation in engine duct systems with thermal contact discontinuities," *Mathematical and Computer Modelling*, vol. 49, no. 5–6, pp. 1078–1088, Mar. 2009.
- [82] C. F. Colebrook and C. M. White, "Experiments with Fluid Friction in Roughened Pipes," *Proc. R. Soc. Lond. A*, vol. 161, no. 906, pp. 367–381, Aug. 1937.
- [83] S. E. Haaland, "Simple and explicit formulas for the friction factor in turbulent pipe flow," *J. Fluids Eng.*, vol. 105:1, 1983.
- [84] R. K. Shah and A. L. London, "Chapter VII - Rectangular Ducts," in *Laminar Flow Forced Convection in Ducts*, R. K. S. L. London, Ed. Academic Press, 1978, pp. 196–222.
- [85] D. S. Miller, *Internal Flow Systems*, 2nd edition. Cranfield, Bedford: British Hydromechanics Research Association, 1990.
- [86] R. K. Shah and D. P. Sekulic, *Fundamentals of Heat Exchanger Design*, 1 edition. Hoboken, NJ: Wiley, 2002.
- [87] T. L. Bergman, A. S. Lavine, F. P. Incropera, and D. P. DeWitt, *Fundamentals of Heat and Mass Transfer*, 7 edition. Hoboken, NJ: Wiley, 2011.
- [88] A. P. Colburn, "A method of correlating forced convection heat-transfer data and a comparison with fluid friction," *International Journal of Heat and Mass Transfer*, vol. 7, no. 12, pp. 1359–1384, Dec. 1964.
- [89] S. Earnshaw, "On the Mathematical Theory of Sound," *Philosophical Transactions of the Royal Society of London*, vol. 150, pp. 133–148, Jan. 1860.

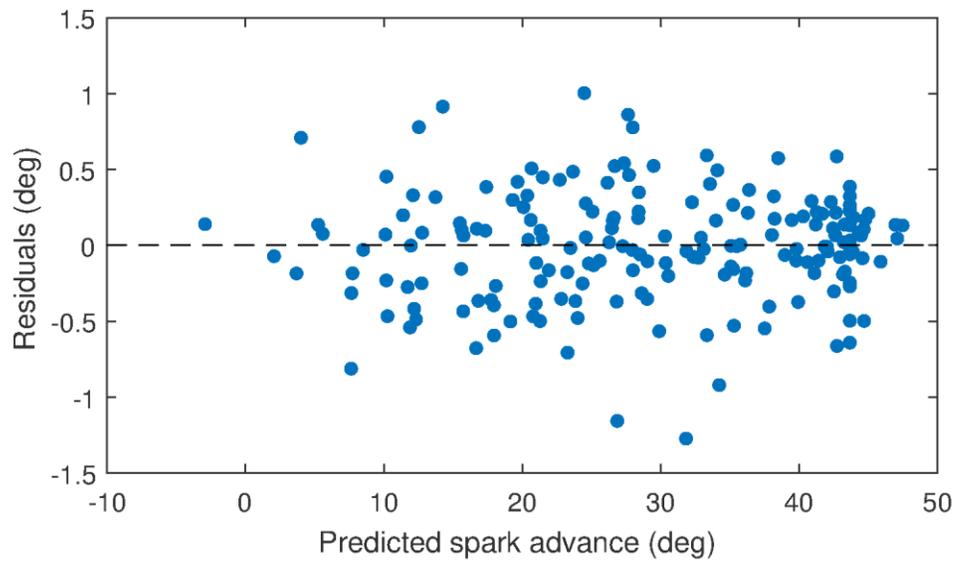
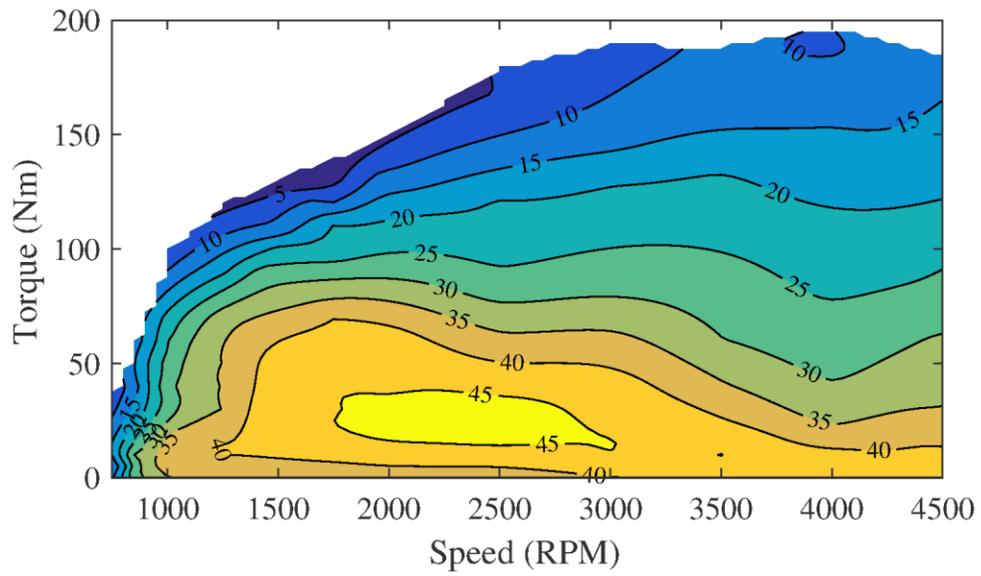
- [90] M. D. Bassett, R. J. Pearson, N. P. Fleming, and D. E. Winterbone, "A Multi-Pipe Junction Model for One-Dimensional Gas-Dynamic Simulations," SAE International, Warrendale, PA, SAE Technical Paper 2003-1-370, Mar. 2003.
- [91] S. Grasreiner, J. Neumann, C. Luttermann, M. Wensing, and C. Hasse, "A quasi-dimensional model of turbulence and global charge motion for spark ignition engines with fully variable valvetrains," *International Journal of Engine Research*, vol. 15, no. 7, pp. 805–816, Oct. 2014.
- [92] J.-E. Yun and J.-J. Lee, "A Study on Combine Effects between Swirl and Tumble Flow of Intake Port System in Cylinder Head," presented at the Seoul 2000 FISITA World Automotive Congress, 2000.
- [93] H. Xu, "Some Critical Technical Issues on the Steady Flow Testing of Cylinder Heads," Warrendale, PA, SAE Technical Paper 2001-01-1308, Mar. 2001.
- [94] S. F. Benjamin, "A Phenomenological Model for 'Barrel' Swirl in Reciprocating Engines," *Proceedings of the IMechE*, vol. 206, no. 1, pp. 63–71, Jan. 1992.
- [95] M. Achuth and P. S. Mehta, "Predictions of tumble and turbulence in four-valve pentroof spark ignition engines," *International Journal of Engine Research*, vol. 2, no. 3, pp. 209–227, Jun. 2001.
- [96] A. N. Kolmogorov, "Dissipation of Energy in the Locally Isotropic Turbulence," *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 434, no. 1890, pp. 15–17, Jul. 1991.
- [97] C. Borgnakke, V. S. Arpaci, and R. J. Tabaczynski, "A Model for the Instantaneous Heat Transfer and Turbulence in a Spark Ignition Engine," SAE Technical Paper, Warrendale, PA, SAE Technical Paper 800287, Feb. 1980.
- [98] N. C. Markatos, "The mathematical modelling of turbulent flows," *Applied Mathematical Modelling*, vol. 10, no. 3, pp. 190–220, Jun. 1986.
- [99] E. C. Chan, G. de Simone, V. Mulone, and M. H. Davy, "Numerical and Experimental Characterization of a Natural Gas Engine With Partially Stratified Charge Spark Ignition," *J. Eng. Gas Turbines Power*, vol. 133, no. 2, pp. 022801–022801, Oct. 2010.
- [100] D. D. Brehob and C. E. Newman, "Monte Carlo Simulation of Cycle by Cycle Variability," presented at the International Fuels and Lubricants Meeting and Exposition, 1992.
- [101] S. Ratnak, J. Kusaka, and Y. Daisho, "3D Simulationson Premixed Laminar Flame Propagation of iso-Octane/Air Mixture at Elevated Pressure and Temperature," SAE Technical Paper, Warrendale, PA, SAE Technical Paper 2015-01-0015, Mar. 2015.

- [102] M. Metghalchi and J. C. Keck, "Burning velocities of mixtures of air with methanol, isooctane, and indolene at high pressure and temperature," *Combustion and Flame*, vol. 48, pp. 191–210, 1982.
- [103] D. B. Rhodes and J. C. Keck, "Laminar Burning Speed Measurements of Indolene-Air-Diluent Mixtures at High Pressures and Temperatures," SAE International, Warrendale, PA, SAE Technical Paper 850047, Feb. 1985.
- [104] G. Woschni, "A Universally Applicable Equation for the Instantaneous Heat Transfer Coefficient in the Internal Combustion Engine," SAE International, Warrendale, PA, 670931, Feb. 1967.
- [105] B. Thompson and H.-S. Yoon, "Development of a High-Fidelity 1D Physics-Based Engine Simulation model in MATLAB/Simulink," SAE International, Warrendale, PA, SAE Technical Paper 2014-01-1102, Apr. 2014.
- [106] The MathWorks, *Simulink--Developing S-Functions*. 2014.
- [107] B. Ellies, C. Schenk, and P. Dekraker, "Benchmarking and Hardware-in-the-Loop Operation of a 2014 MAZDA SkyActiv 2.0L 13:1 Compression Ratio Engine," Warrendale, PA, SAE Technical Paper 2016-01-1007, Apr. 2016.
- [108] S. Lee, C. Schenk, and J. McDonald, "Air Flow Optimization and Calibration in High-Compression-Ratio Naturally Aspirated SI Engines with Cooled-EGR," Warrendale, PA, SAE Technical Paper 2016-1-565, Apr. 2016.
- [109] C. Schenk and P. Dekraker, "Potential Fuel Economy Improvements from the Implementation of cEGR and CDA on an Atkinson Cycle Engine," Warrendale, PA, SAE Technical Paper 2017-01-1016.
- [110] "MAZDA: SKYACTIV-G | SKYACTIV TECHNOLOGY." [Online]. Available: <http://www.mazda.com/en/innovation/technology/skyactiv/skyactiv-g/>. [Accessed: 06-Feb-2017].
- [111] "gptoolbox - File Exchange - MATLAB Central." [Online]. Available: <http://www.mathworks.com/matlabcentral/fileexchange/49692-gptoolbox>. [Accessed: 06-Feb-2017].
- [112] R. Herweg and R. Maly, "A Fundamental Model for Flame Kernel Formation in S. I. Engines," SAE International, Warrendale, PA, SAE Technical Paper 922243, Oct. 1992.
- [113] T. Kolda, R. Lewis, and V. Torczon, "Optimization by Direct Search: New Perspectives on Some Classical and Modern Methods," *SIAM Rev.*, vol. 45, no. 3, pp. 385–482, Jan. 2003.
- [114] A. M. S. Zalzala and P. J. Fleming, *Genetic Algorithms in Engineering Systems*. IET, 1997.

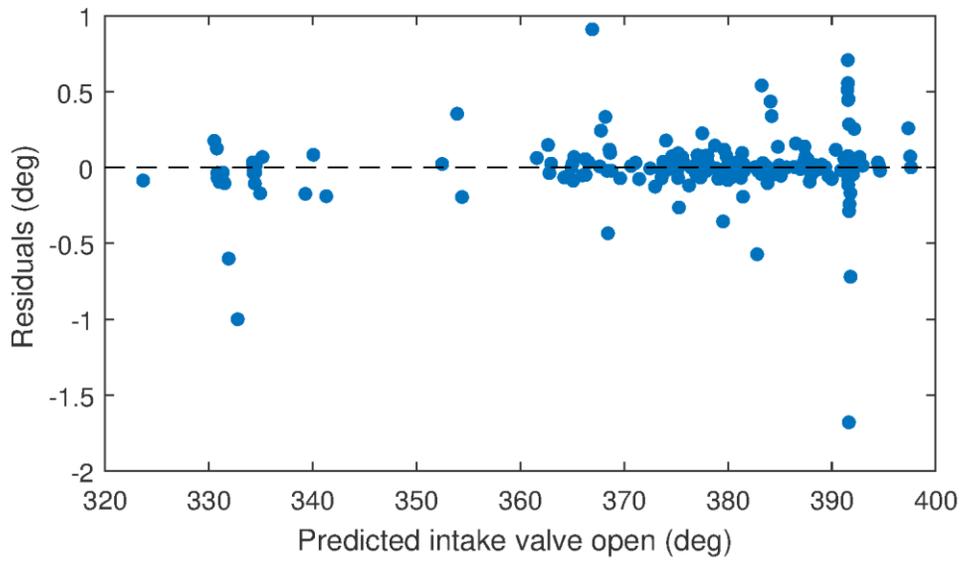
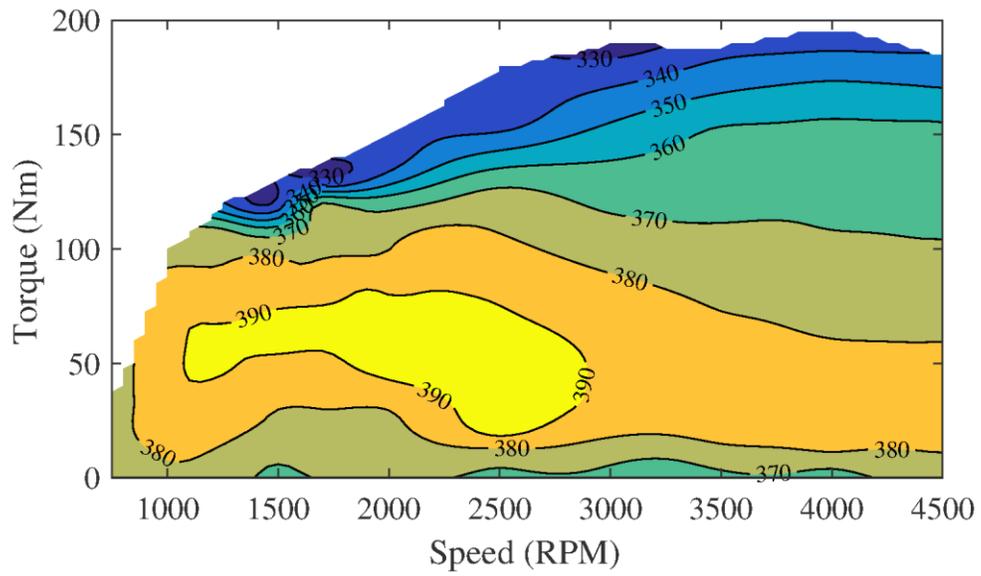
- [115] K. Deb, “Multi-objective Optimisation Using Evolutionary Algorithms: An Introduction,” in *Multi-objective Evolutionary Optimisation for Product Design and Manufacturing*, L. Wang, A. H. C. Ng, and K. Deb, Eds. Springer London, 2011, pp. 3–34.
- [116] The MathWorks, *Simulink User’s Guide 2016b*. 2016.

## **APPENDIX A: CONTROL TABLES**

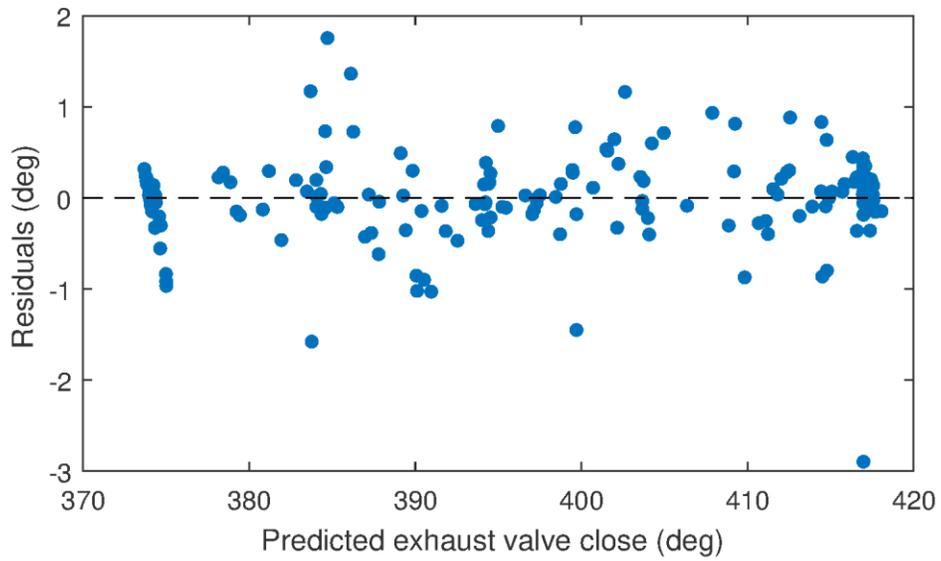
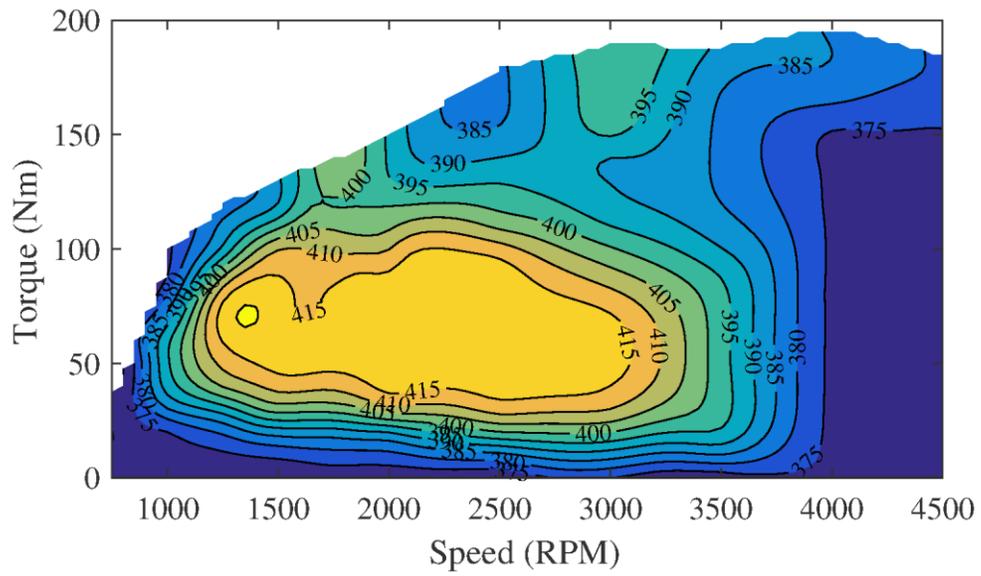
Data for each control parameter was fit to Gaussian Process Model (GPM) and Radial Basis Function (RBF) models using Model-Based Calibration Toolbox™ (MBC). MBC offers a large variety of kernel and basis functions for GPM and various kernel functions and adjustable number of centers for RBF models. Depending on the data, GPM or RBF provided a better fit. Several alternatives were created for comparison. GPMs were selected based lowest Predicted Residual Error Sum of Squares (PRESS). RBF models were selected based on the Akaike Information Criterion (AICc). Spark advance, Intake Valve Open (IVO) angle, Exhaust Valve Close (EVC) angle, relative air-fuel ratio, injection timing, and injection pulse width tables are shown below.



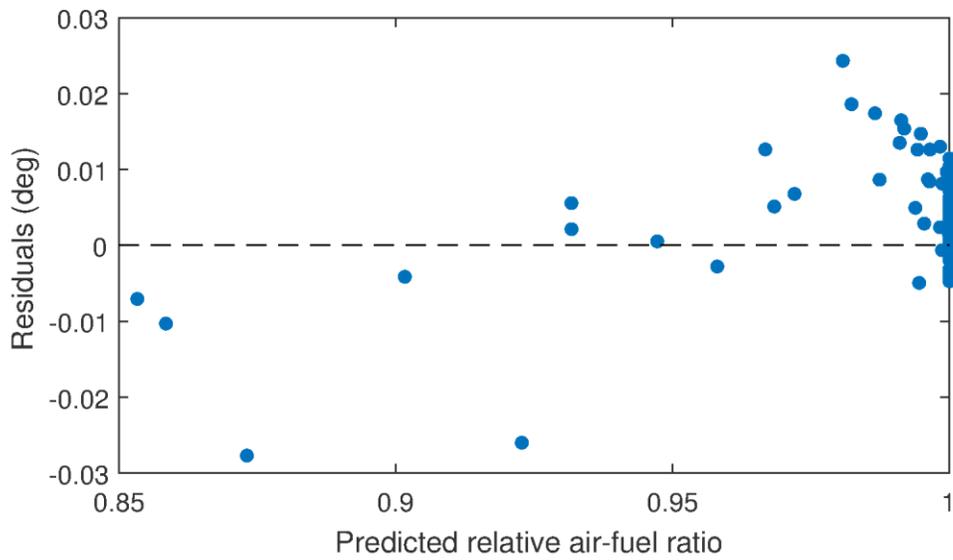
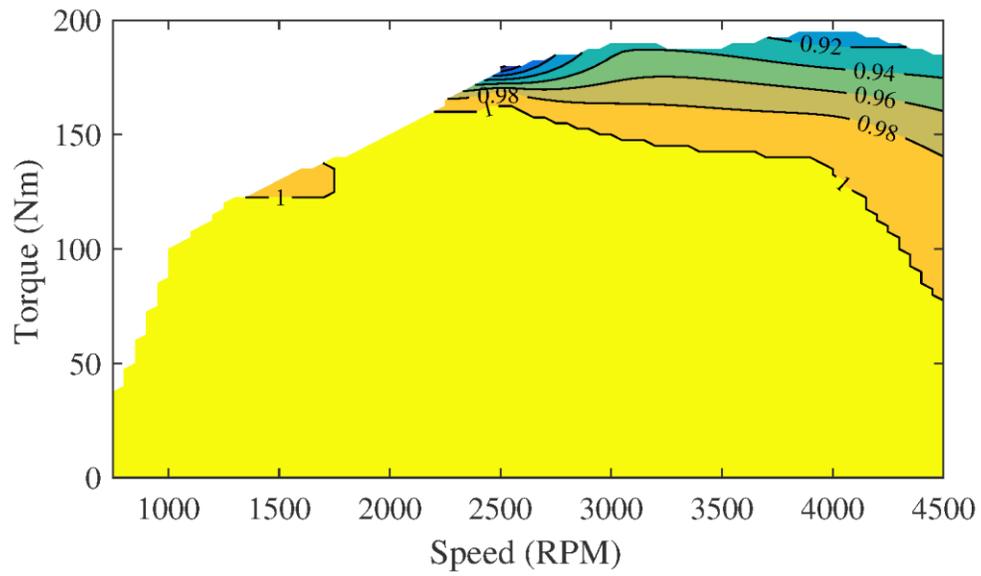
**Figure A.1: Spark advance table (deg) fit with RBF model and residuals**



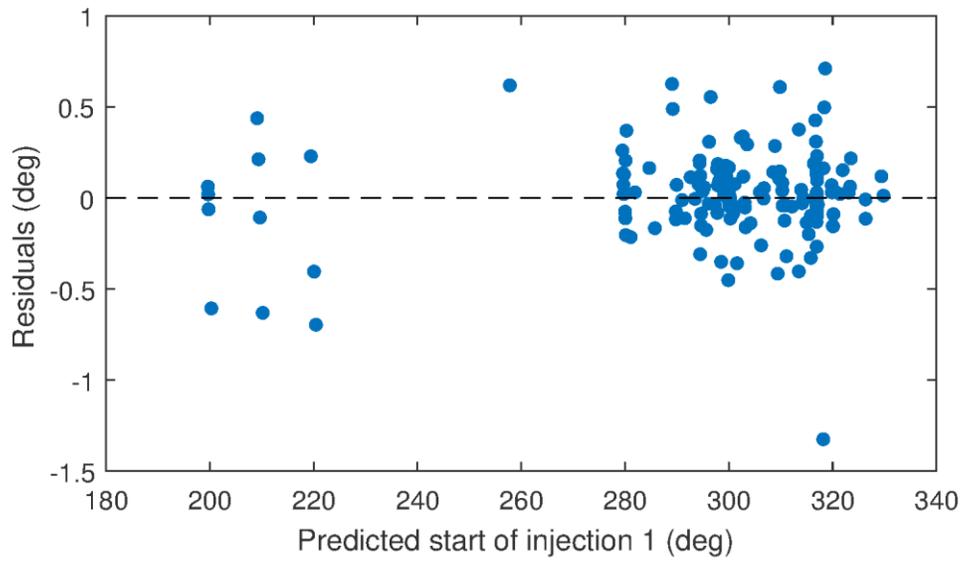
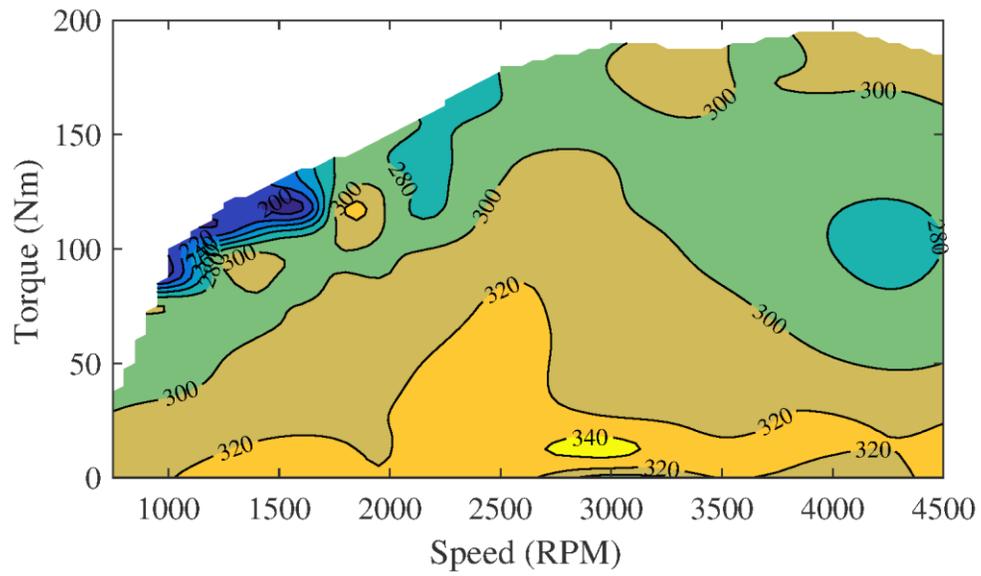
**Figure A.2: Intake valve open crank angle table fit with GPM and residuals**



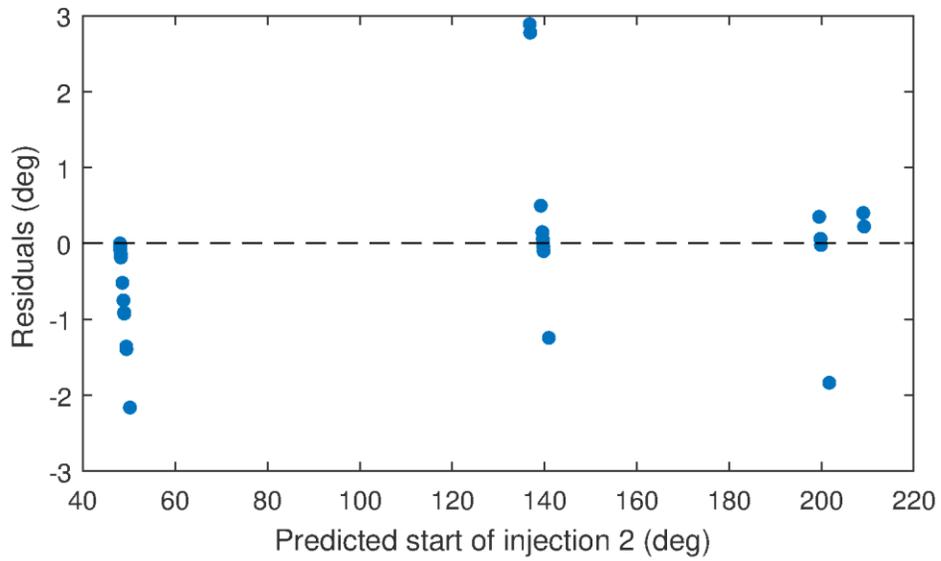
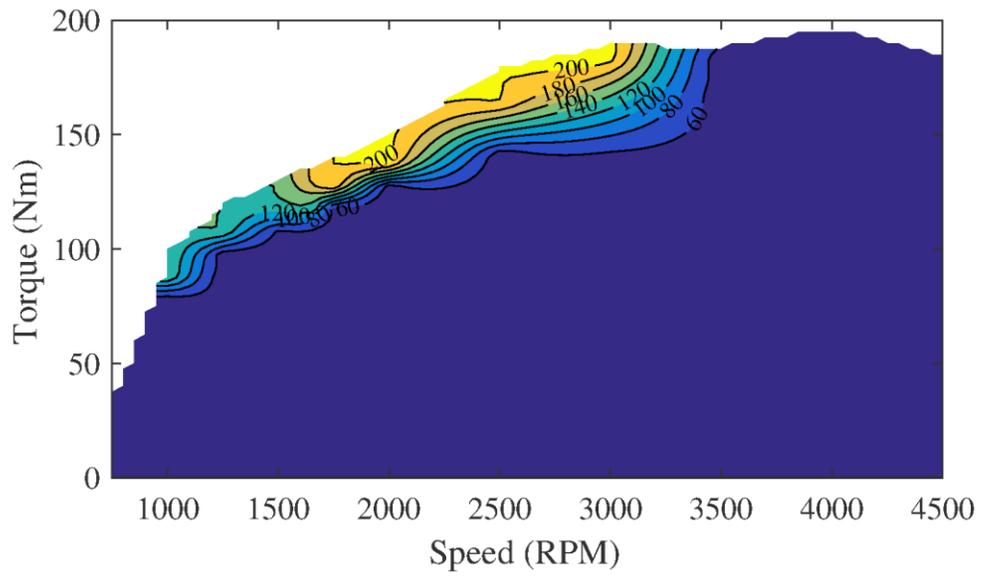
**Figure A.3: Exhaust valve close crank angle table (deg) fit with GPM and residuals**



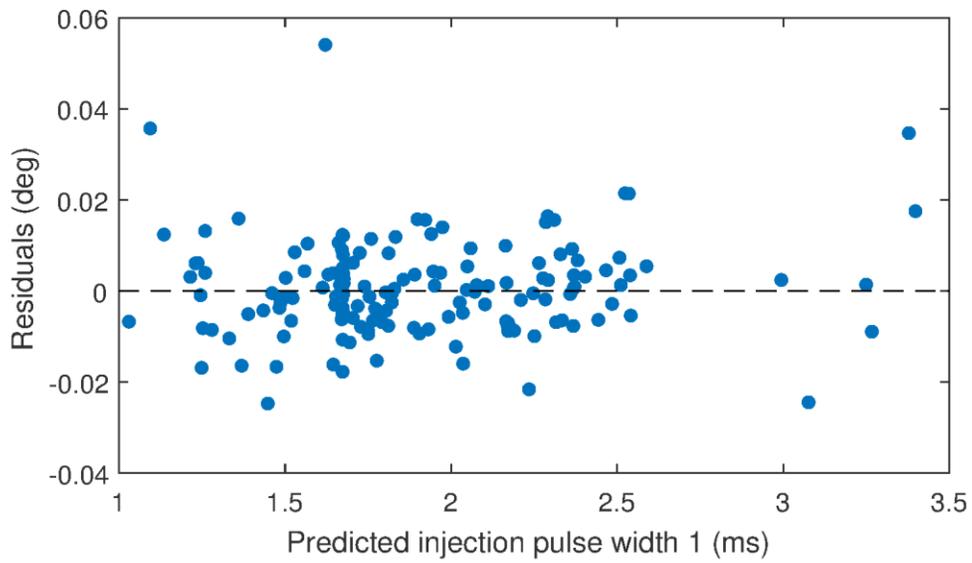
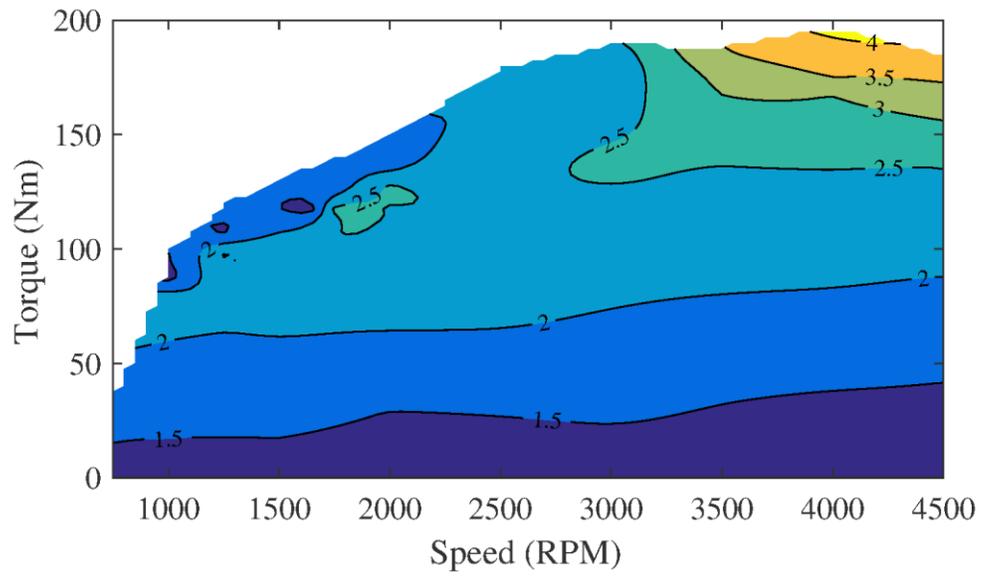
**Figure A.4: Relative air-fuel ratio table fit with RBF and residuals**



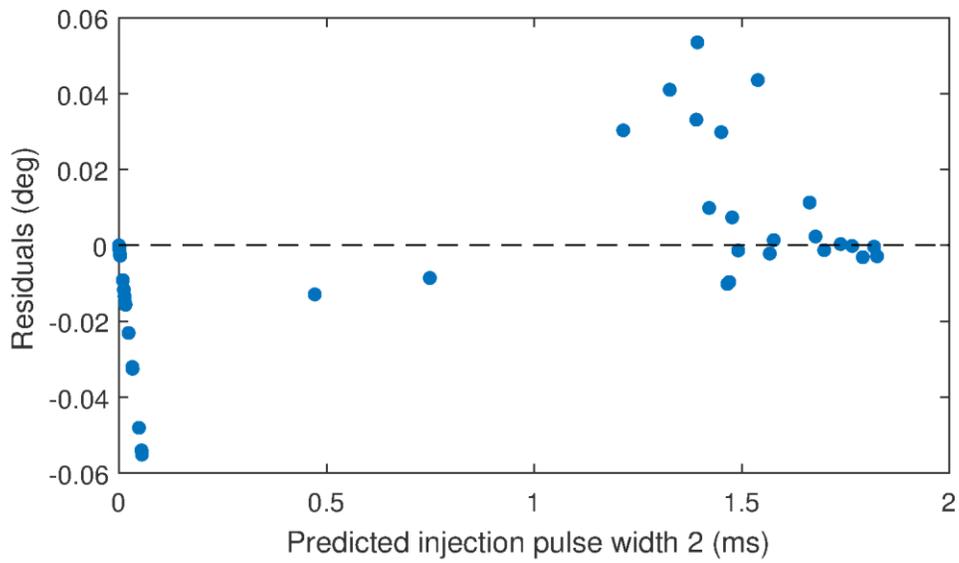
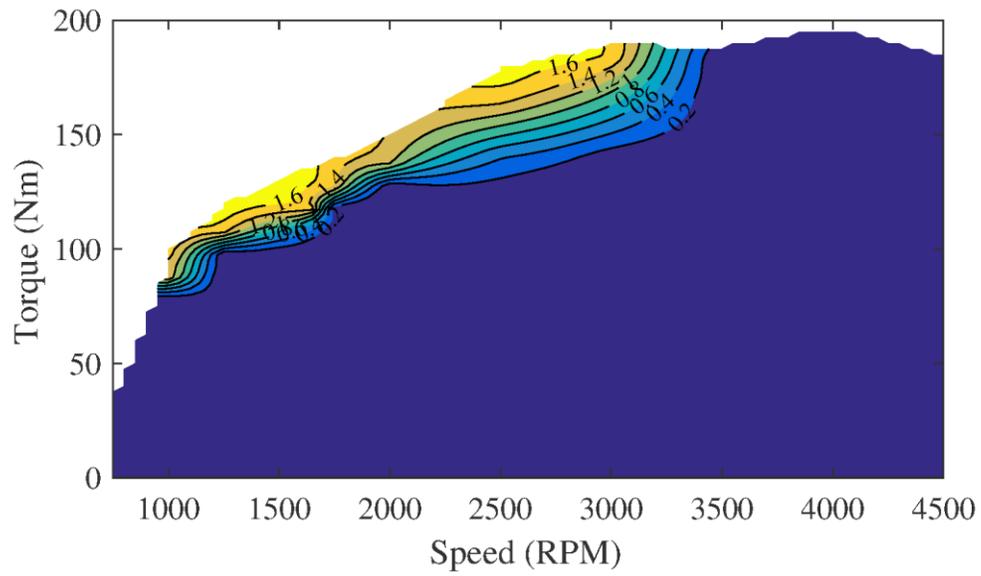
**Figure A.5: First start injection angle table (deg) fit with RBF and residuals**



**Figure A.6: Second start injection angle table (deg) fit with RBF and residuals**



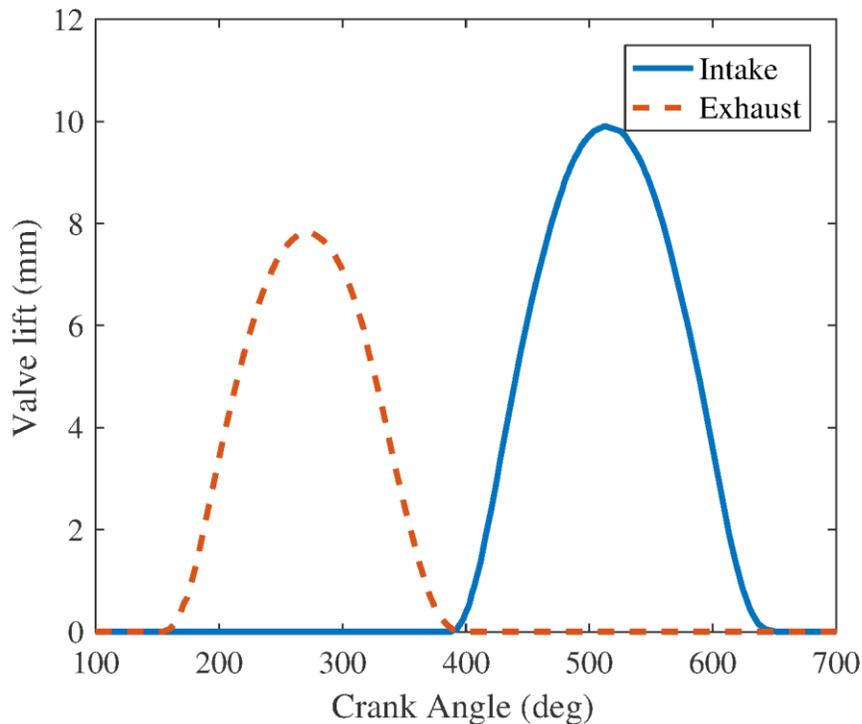
**Figure A.7: First injection pulse width table (ms) fit with RBF and residuals**



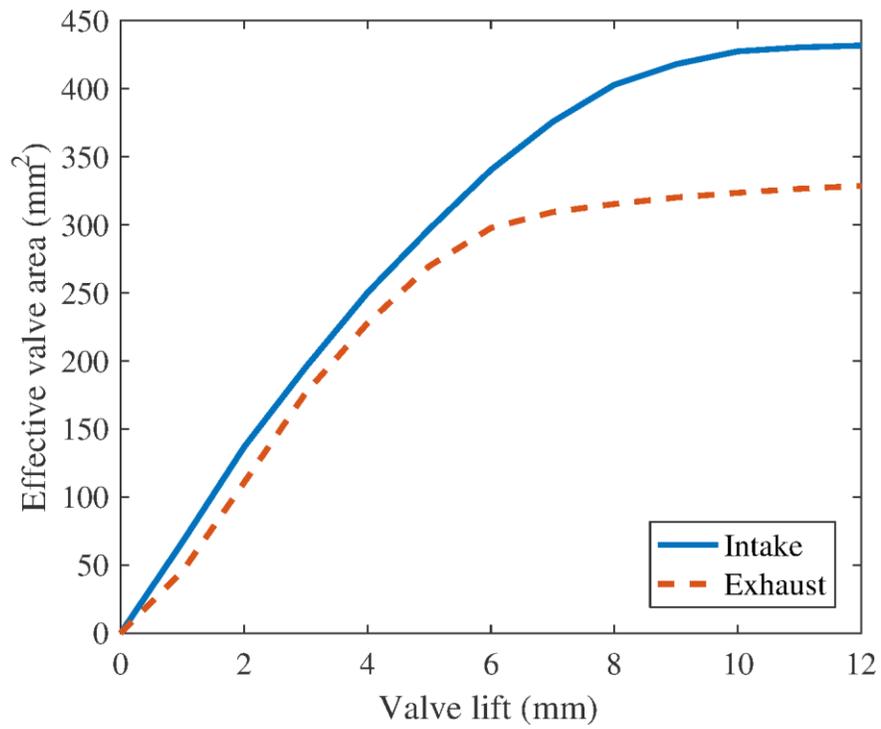
**Figure A.8: Second injection pulse width table (ms) fit with RBF and residuals**

## APPENDIX B: SKYACTIV ENGINE MODEL

The 2.0 L Mazda Skyactiv engine has two intake and exhaust valves. Valve lift and effective flow area provided by the EPA were used in the Simulink model. Plotted in Figure B.1, intake valves have a maximum lift of 10 mm and exhaust valves have a maximum lift of 8 mm. The effective valve flow areas are plotted in Figure B.2.



**Figure B.1: Intake and exhaust valve lift for 2.0 L Mazda Skyactiv engine**



**Figure B.2: Intake and exhaust valve effective flow area used in Simulink engine model**