Machine Learning Techniques for High Performance Engine Calibration

By
Adam Kenton Van Horn
B.S. Computer Science, December 2008

Submitted to the graduate degree program in Computer Science and the Graduate Faculty of the University of Kansas in partial fulfillment of the requirements for the degree of Master of Science.

_____
Chairperson: Dr. Arvin Agah

_____
Dr. Jerzy Grzymala-Busse

_____
Dr. James Miller

_____
Dr. Christopher Depcik

_____
Date Defended

The Thesis Committee for Adam Kenton Van Horn certifies
that this is the approved version of the following thesis:

Machine Learning Techniques for High Performance Engine Calibration

_____
Chairperson: Dr. Arvin Agah

_____
Date Approved

Abstract

Ever since the advent of electronic fuel injection, auto manufacturers have been able to increase fuel efficiency and power production, and to meet stricter emission standards. Most of these systems use engine sensors (Speed, Throttle Position, etc.) in concert with look-up tables to determine the correct amount of fuel to inject. While these systems work well, it is time and labor intensive to fine tune the parameters for these look-up tables. In general, automobile manufacturers are able to absorb the cost of this calibration since the variation between engines in a new model line is often small enough as to be inconsequential for a specific calibration.

However, a growing number of drivers are interested in modifying their vehicles with the intent of improving performance. While some aftermarket performance upgrades can be accounted for by the original manufacturer equipped (OEM) electronic control unit (ECU), other more significant changes, such as adding a turbocharger or installing larger fuel injectors, require more drastic accommodations. These modifications often require an entirely new ECU calibration or an aftermarket ECU to properly control the upgraded engine. The problem is now that the driver becomes responsible for the calibration of the ECU for this "new" engine. However, most drivers are unable to devote the resources required to achieve a calibration of the same quality as the original manufacturers. At best, this results in reduced fuel economy and performance, and at worst, unsafe and possibly destructive operation of the engine.

The purpose of this thesis is to design and develop—using machine learning techniques—an approximate predictive model from current engine data logs, which can be used to rapidly and incrementally improve the calibration of the engine. While there has been research into novel control methods for engine air-fuel ratio control, these methods are inaccessible to the majority of end users, either due to cost or the required expertise with engine

calibration. This study shows that there is a great deal of promise in applying machine learning techniques to engine calibration and that the process of engine calibration can be expedited by the application of these techniques.

Acknowledgements

I would like to thank my advisor, Dr. Arvin Agah, for allowing me to pursue a topic of great personal interest and his continued support throughout the process.

I would also like to thank Dr. Christopher Depcik for elucidating the finer points of engine design, operation, and control, giving me a deeper understanding of the mechanisms at work.

Additionally, I want to give a great many thanks to the online RX-7 community, specifically Chuck Westbrook, Kyle Krutilek, Barry Bordes, Mark White, Brian Davies, and Raymond Herchenroder. Without your previous work and experience with RX-7s and the A'PEXi Power FC hardware and software, none of this would have been possible.

Finally, I would like to thank my loving and understanding fiancée, Cindy, for her continued patience and support. She constantly kept me motivated and focused. My parents, Jackie and Keith, and my sister Rachel, also deserve special mention for their continued support throughout my educational career.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Motivation

Modern automobiles and their engines are truly astounding works of engineering. Even with well over a hundred years of development, there are improvements being made continually. Modern vehicles are designed with the average consumer in mind, which results in a number of compromises. The styling of the vehicle cannot be too radical, or else it will not have broad appeal. The performance of the engine cannot be maximized since it may be required to operate in vastly different environments and situations. The handling of the vehicle must take into account the imperfections found on public roadways so as not to transmit too much noise, vibration, or harshness (NVH) to the drivers and passengers.

There is however a small segment of the population that is not satisfied with the results of these many compromises. Therefore, an entire industry has developed to address these needs. In 2012 alone, the overall aftermarket sales totaled $307.7 billion [1]. Moreover, as the vehicles become more and more complex and dependent on modern technology, the knowledge required to modify these vehicles increases.

Within the past 30 years, nearly all of the mechanical control systems have been replaced with electronic control systems [2]. For example, carburetors have been replaced with Electronic Fuel Injection (EFI), and distributors have been replaced by electronic ignition. No longer are customers able to tinker on their vehicles with only simple hand tools.

When enthusiasts or competitors desire more radical performance, oftentimes the original Electronic Control Unit (ECU) is unable to adequately account for the new additions. This requires an aftermarket ECU, which in turn needs to be calibrated in order to control the new system.

Engine calibration is a long and expensive iterative process [3,4]. Since internal combustion engines are extremely non-linear systems, modifying a single variable has far reaching consequences. Whenever a new calibration is developed, it needs to be tested under many possible operating conditions in order to ensure peak performance.

This thesis will explore various approaches to expedite the calibration of electronic fuel injection systems using real world data, gathered from the engine being calibrated, to develop a model of its modes of operation, and to determine what values to change and how to change them, to achieve the desired results.

1.2 Thesis Organization

This thesis consists of five chapters. Chapter 2 provides the required background information on internal combustion engine operation, design, and control systems. It also covers the machine learning techniques used to explore the problem, criteria used to evaluate the performance of the machine learning techniques, and related work and current solutions in the industry. Chapter 3 delves into the research methodology used while exploring this issue and the specific tools, software, and hardware used. Chapter 4 presents the experimental results of this project. Chapter 5 discusses the impact of these results, the contributions, the limitations, and future extensions to the proposed methodology.

Chapter 2 Background and Related Work

2.1 Internal Combustion Engine Operation

An internal combustion engine can be thought of as a living, breathing organism. It requires air, fuel, and a way to metabolize that fuel. Virtually all modern automobiles use Electronic Control Units (ECUs) to control the operation of the engine, transmission, antilock brakes and other vehicle systems. Today, nearly every aspect of the engine, especially ignition timing and fuel injection, is controlled electronically.

2.1.1 General Engine Design and Operation

An Internal Combustion Engine (ICE) converts chemical energy to mechanical energy by combusting a traditionally hydrocarbon fuel with ambient oxygen. The most common operating cycle for passenger vehicles today is the spark ignition four-stroke. This consists of four distinct (but possibly overlapping) stages or strokes. Each thermodynamic cycle begins with the intake stroke, where the fuel-air mixture is drawn into the cylinder. The compression stroke compresses this mixture to extract the maximum amount of energy from the fuel. Next, the mixture is ignited, which leads to a drastic increase in temperature and pressure in the cylinder. The expansion or power stroke transforms this pressure into mechanical work as the piston expands. Finally, the burned mixture is expelled through the exhaust valve as the piston returns to the top of the cylinder, allowing the next cycle to take place (Figure 2.1) [5].

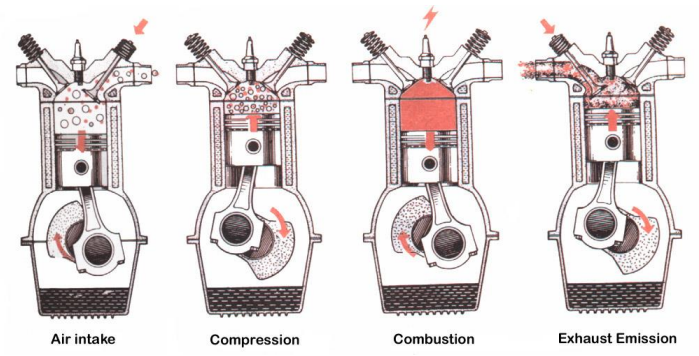


Figure 2.1: Four-stroke thermodynamic cycle [23]

2.1.2 Rotary Engine Design and Operation

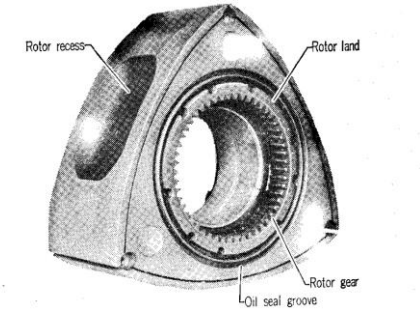


Figure 2.2: Rotary Engine Rotor [6]

The Wankel Rotary Engine was patented by Felix Wankel in 1929 and began commercial development by Wankel while working at NSU Motorenwerke AG in the 1950s [6]. It differs from conventional reciprocating engines in that there are no reciprocating parts (pistons), no valve-train, and an unusually high power to weight ratio. During the 1960s, various automotive manufacturers licensed the technology from NSU, specifically Mazda's precursor corporation Toyo Kogyo for their 1967 Cosmo sports car. While the four-stroke thermodynamic cycle is still in use, each stroke occurs in a fixed location in a rotary engine (Figure 2.2).

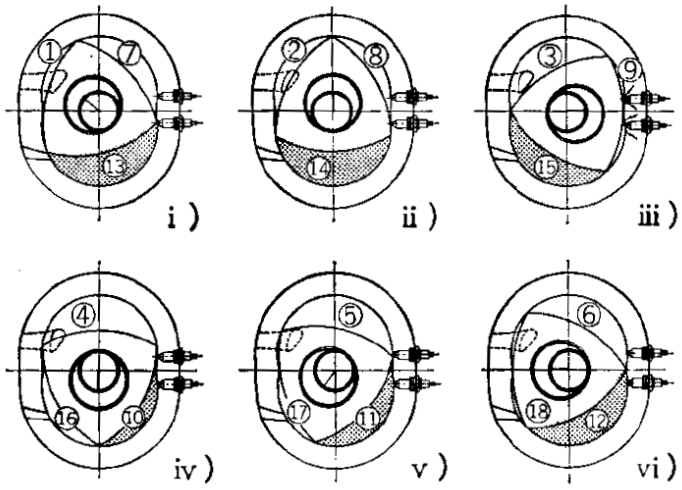


Figure 2.3: Rotary Engine Operation [6].

Figure 2.3 demonstrates the operation of a rotary engine. The points marked 1-6 represent the intake stroke, 7-9, the compression stroke and ignition of the mixture, and 10, 11, and 12 show the expansion stroke. Finally, points 13-18 demonstrate the exhaust stroke.

A number of differences can be seen between reciprocating engines and rotary engines. The most obvious feature is that the rotor never changes direction the way a piston does in a reciprocating engine. This leads to a "smooth" engine since there are no vibrations caused by reciprocating motion. Furthermore, there is no valve-train used to control engine breathing, but rather fixed ports located in the housing for intake and exhaust. This leads to better engine breathing, especially at higher engine speeds. Also, during each revolution of the crankshaft, one complete thermodynamic cycle begins and ends. The rotor, therefore, rotates at only ⅓ the speed of the crankshaft allowing for high engine speeds. A few disadvantages of the rotary engine include difficulty in sealing at the edges of the rotor face, and poor fuel economy and emissions, due to incomplete burning of the fuel air mixture caused by a short expansion stroke.

2.1.3 Turbocharging

A turbocharger is a mechanical device used to recover waste energy from the exhaust gases of an internal combustion engine [5,7]. It consists of two main parts, the turbine and the compressor. As figure 2.4 illustrates, hot exhaust gases impinge on the blades of the turbine causing it to rotate. This, in turn, causes the compressor to begin to rotate, causing an increase in both the pressure and mass flow rate of the intake air. By increasing the density of the intake charge and its mass flow rate, the engine is able to ingest (and combust) more air than normally possible. Thus, turbocharging is often used to increase the volumetric efficiency (by forcing a larger volume of air into the engine), power (by adding additional fuel to compensate for the increase airflow), and thermal efficiency of an engine (by recovering waste heat from the exhaust gas stream). The engine industry is currently moving towards downsized, turbocharged engines in an effort to maintain power levels and increase fuel efficiency.

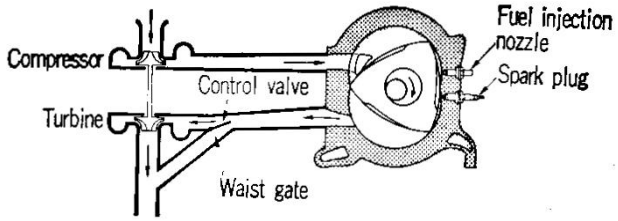


Figure 2.4 Turbocharged Rotary Engine [6].

2.2 Engine Control Systems

There are a number of ways in which fuel can be injected into an internal combustion engine (ICE). The oldest method of fuel injection was single point injection, also known as throttle body injection. Continued developments in injector design, fuel pump technology, and control systems, have since given rise to *continuous injection*, where fuel is injected continuously but the flow rate is adjusted, *multi-point injection*, where there are multiple injectors placed closer to the intake ports, *port fuel injection*, where fuel is injected just before

the combustion chambers intake port, and *direct injection*, where fuel is injected directly into the combustion chamber [5]. A common feature of all of these systems is that they operate the injectors based on pulse width, or the number of milliseconds an injector is open and injecting fuel per injection event. In all but direct injection, there is only one fueling event per thermodynamic cycle; direct injection may have multiple injection events per cycle. The amount of fuel injected and ingested by the engine is proportional to the injector "on-time" and therefore needs to be adjusted as engine conditions change. Figure 2.5 shows the inner workings of a fuel injector.

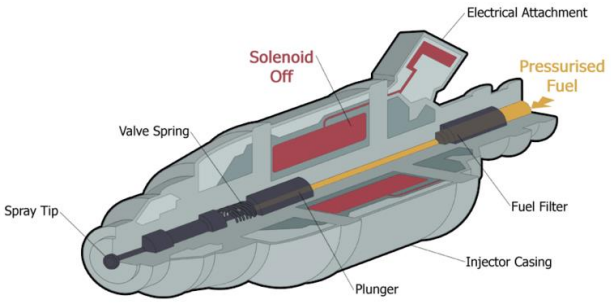


Figure 2.5: Automotive Fuel Injector [24].

2.2.1 Electronic Fuel Injection

When determining the amount of fuel to be injected, the most important quantity that must be measured or calculated is the amount of air, specifically the mass of oxygen, entering the engine. Historically, carburetors were used for this, and being mechanical systems, they would add an amount of fuel proportional to the airflow rate past the *venturi* or narrowing in the carburetor [5]. As air passes through the venture, its flow rate would increase and a resulting decrease in pressure, due to Bernoulli's Principle, would draw fuel from *jets*, or orifices in the carburetor. The larger the pressure drop, the more fuel would mix with the airstream. While this is a rather simple system, it does not add precisely measured amounts of fuel, and with

increasing demands on fuel economy and emissions, most manufacturers turned to EFI (Electronic Fuel Injection) to have more control over the fueling process.

N-Alpha

The oldest method of electronic fuel injection *N-Alpha*, uses only the current engine speed (RPM, revolutions per minute) as N and opening angle (Alpha) of the throttle (as reported by the Throttle Position Sensor or TPS) to determine how much fuel should be injected [8]. While this system can work well in some circumstances (e.g., wide-open throttle) it suffers in almost every other driving regime since it is not directly measuring or calculating airflow. It is often used in conjunction with one or possibly both of the following control systems.

MAP

Manifold Absolute Pressure (MAP) or *speed-density* systems are an improvement over N-Alpha but also do not directly measure airflow [5,7,9,10]. Rather they infer the mass flow rate using the relationship described by the Ideal Gas Law:

$$\mathrm{PV} = \mathrm{nRT}$$

where n is the number of moles of the gas R is the ideal gas constant, This can be rewritten in terms of the mass of the gas as:

$$P = \rho R_{specific} T$$

where ρ is the density of the gas and $R_{specific}$ is the ratio of R to the molar mass of the gas.

In this system, there are sensors that measure intake air temperature (T) and manifold absolute pressure (P). By knowing the Volumetric Efficiency (VE), or the percentage of occupied volume versus geometric volume of the engine at various engine speeds, the mass of air can be calculated. Once the mass of air has been determined, it is trivial to add the correct amount of fuel to reach the target air-fuel ratio for the operating conditions. The drawback to

this system is that the VE table needs to be properly calibrated for the various RPM and pressure ranges in which the engine is expected to operate. It is this drawback that this thesis seeks to address.

MAF

Mass Airflow (MAF) is a third system of load control. These systems directly measure the amount of air entering the engine. This is usually accomplished with a "hot wire sensor", where a filament is held in the intake stream and heated to a constant temperature [7]. The amount of electrical current required to maintain this temperature is directly proportional to the mass of airflow entering the engine and cooling the wire. Thus, the mass airflow of the engine can be measured directly as the voltage drop caused in the sensor. This type of sensor is also invariant to the density of the incoming air. Since denser air will provide more cooling of the wire, there is no need for auxiliary sensors.

2.2.2 Air-Fuel Ratio

Air-Fuel Ratio (AFR) is a measure of the ratio of the mass of air to mass of fuel in a mixture. If a mixture has equal amounts fuel and oxygen, it is said to be a stoichiometric mixture [5,7]. This means that there is just enough oxygen present to completely combust the fuel in the mixture. The isooctane combustion reaction below denotes a stoichiometric reaction.

$$2\,C_8H_{18} + 25\,O_2 \rightarrow 16\,CO_2 + 18\,H_2O$$

While pump gasoline is a composition of various hydrocarbon fuels, it has an approximate stoichiometric AFR of 14.7:1; hence, for every gram of fuel, 14.7 grams of oxygen are required to ensure complete combustion. A mixture can be described as *lean* or *rich* depending on whether there is an excess of oxygen or fuel, respectively. *Lambda (*λ), or the air-fuel equivalence ratio, is a fuel normalized measure of AFR. It is defined as:

$$\lambda = \mathrm{AFR} / \mathrm{AFR}_{\mathrm{Stoich}}$$

Therefore, for any fuel, $\lambda = 1$ denotes a stoichiometric mixture, $\lambda > 1$ denotes a lean mixture, and $\lambda < 1$ a rich mixture.

There is no "ideal" AFR. It greatly depends on the goals of the engine calibrator and the design of the engine. For example, due to gas dissociation effects, maximum power is produced slightly rich of stoichiometry with an AFR between 12.5 and 13.3 (λ = 0.85- 0.9) [5]. Likewise maximum economy occurs in the lean region of operation, but not all engines can properly ignite a lean mixture and less power is produced (Figure 2.6).

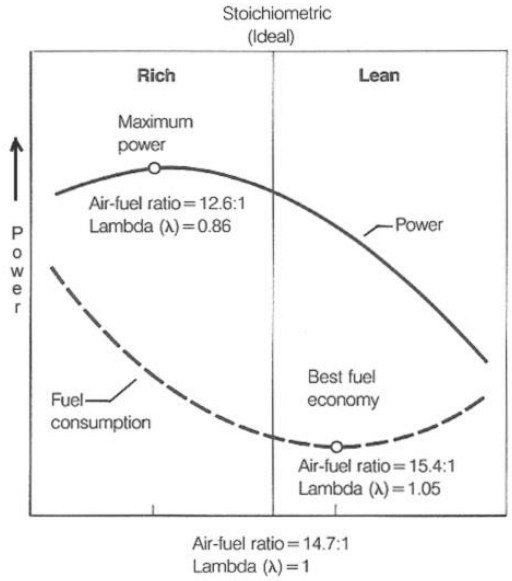


Figure 2.6: Power and Economy vs. λ [25].
Image courtesy of the Wikipedia Foundation, Inc.

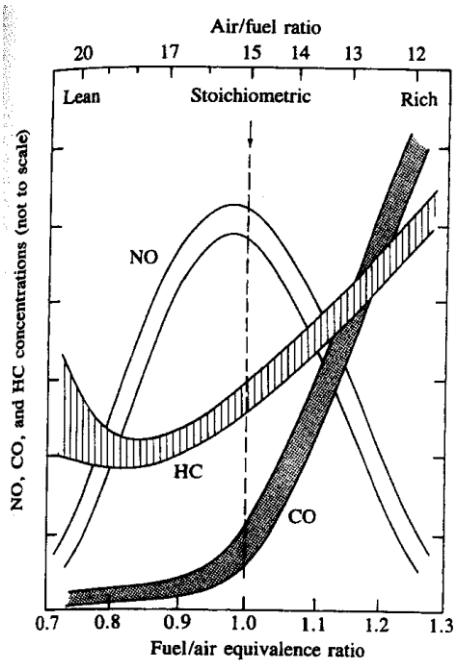


Figure 2.7: HC-CO-NOx and AFR tradeoff [5].

Power and economy are not the only factors that must be considered. As shown in Figure 2.7, as the mixture is enriched from stoichiometric, there is an increase in hydrocarbon and carbon monoxide emissions due to incomplete combustion. Furthermore, as the mixture is leaned from stoichiometric, nitrogen oxide emissions increase, caused by excess air, which is ~70% nitrogen, and high combustion temperatures. Emission control is largely the reason why modern spark ignition engines attempt to run at stoichiometry, since *three-way catalytic converters* work most efficiently at a stoichiometric AFR [5].

Additionally, turbocharged spark ignition engines experience elevated chamber temperatures and pressures, which contribute towards their propensity for pre-ignition and "*knock*", the uncontrolled auto-ignition of the fuel air mixture [2,5-10]. Since knock can quickly destroy an engine if uncorrected, often times calibrators will enrichen the AFR, using the excess fuel as a cooling agent to lower combustion temperatures and mitigate knock.

Exhaust Gas Oxygen Sensors

Universal Exhaust Gas Oxygen Sensors (UEGO) are sensors placed in the exhaust stream that measure the amount of oxygen present in the exhaust gases. This information is indicative of the completeness of the previous combustion event. UEGOs are also useful in determining whether an engine is operating lean, rich, or stoich [2,7].

Figure 2.8: Automotive Fuel Injector [26].

2.2.3 Industry Standards

Currently there is a well-established procedure for engine calibration used in industry but most end users do not have access to such advanced technology and equipment.

Design of Experiments (DOE)

Design of Experiments is a methodology with the goal of obtaining maximum information from the minimum number of experiments [3]. Since ICEs are staggeringly multivariate and non-linear, it can be impossible to fully explore the possible space of engine calibrations. For instance, there is not a single ignition timing that will work well for all engine speeds, or all load levels. It depends on the condition of the spark plugs, the properties of the fuel, the shape of the combustion chamber, the likelihood of knock, cylinder to cylinder variations, etc. Because of this complexity, it is necessary for calibrators to determine what parameters or combination of parameters they will attempt to optimize, and also determine the acceptable level of optimization. Many turn to DOE in an effort to reduce testing to the minimum required so that they can reduce cost and time to market.

In the aforementioned example, the calibrator may know from previous experience or data gathered from a similar engine, the possible range of valid spark advance values. Then, instead of performing a sweep of all values in the range, they may pick key values at uniform intervals and only test these. Once a "good" value has been determined from the above subset, further exploration may happen within the narrowed range, or if the results are sufficient, a new experiment can be conducted. Similarly, the timing requirements of an engine at 2000 RPM are most likely similar to the timing requirements at 2050 RPM. So, the calibration space to be explored by the calibrator can be divided up appropriately, depending on the calibration goals and intended use of the engine.

2.2.4 Engine Modeling Technologies

ETAS ASCMO

ETAS, a subsidiary of the Bosch Group, has developed a software suite, namely, ETAS ASCMO, aimed at providing manufacturers with the ability to accurately model the behavior of complex systems such as internal combustion engines [11]. It claims to utilize "newly developed statistical learning procedures" [11] that lead to a high model accuracy. However, the cost to license this software puts it outside the grasp of most aftermarket customers.

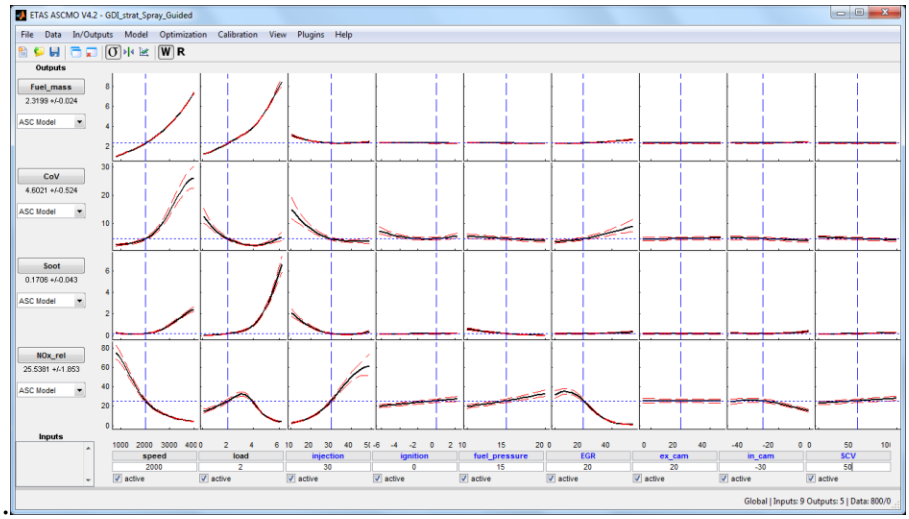


Figure 2.9: ETAS ASCMO [11].

MATLAB Model Based Calibration Toolbox

MATLAB is a popular "high-level language and interactive environment for numerical computation, visualization, and programming" [13]. It also has a great number of add-on "Toolboxes" enabling specialized functionality for a number of domains.

The Model-Based Calibration Toolbox™ is squarely aimed at engine calibration and simulation. It uses "statistical modeling and numerical optimization" to design test plans and create simulations for ICEs [12,13]. It also features some integration with ETAS software. The disadvantage of this solution, however, is that it again requires paid for licensing, and a deeper understanding of programming to take advantage of its more complex features. Furthermore, its model fitting capabilities are limited to "polynomials, splines, radial basis functions, growth models, user-defined MATLAB files, and Simulink models" [13], requiring the end user to define more complicated models if required.

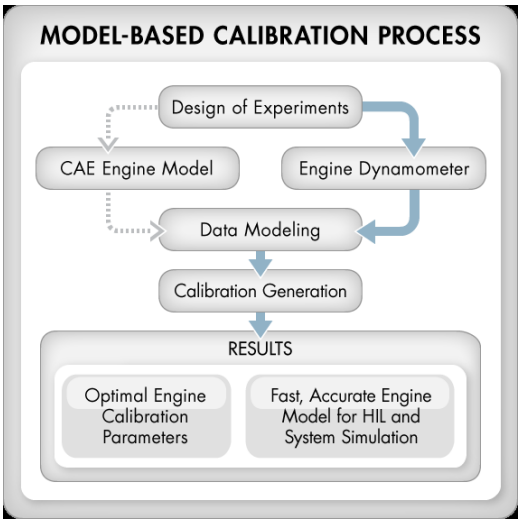


Figure 2.10: MBC Workflow [13].

2.3 Machine Learning Techniques

2.3.1 Machine Learning

Machine Learning has been defined by Arthur Samuel as a "Field of study that gives computers the ability to learn without being explicitly programmed" [14]. All machine learning algorithms attempt to determine rules or define relationships in the data that are presented to them for training. By generalizing over these training samples, the idea is that they will later be able to correctly classify or make some prediction when presented with a novel sample.

2.3.2 Weka

Weka [15] is a suite of tools for machine learning. Written in Java, it contains tools to aid in preprocessing of data, data analysis, data visualization, and predictive modeling, specifically, classification, regression, and association rules. It is often used to rapidly prototype and explore machine learning applications for new fields and datasets [15].

2.3.3 Decision Trees

Decision Trees are a method of classification or regression which operate by determining the most important attributes in the data set and partitioning the data with respect to key values for that attribute [16]. A common method of determining "importance" of attributes is by calculating the *information gain* of the attribute. The information gain for attribute A is defined as:

$$IG(A) = H(S) - \sum_{t \in T} p(t)\, H(t) \quad [16]$$

where *H* denotes *information entropy* of set *S* defined as:

$$H(S) = -\sum_{x \in X} p(x) log_2 p(x) \quad [16]$$

By using information gain, the algorithm attempts to discover which attributes most closely describe the desired decision value and also which attribute values correspond to that decision.

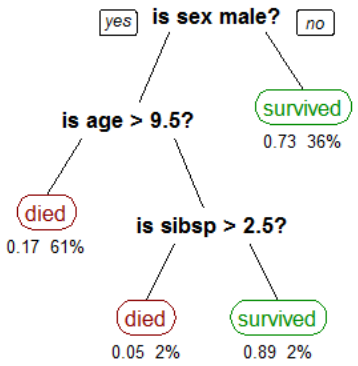


Figure 2.11: Titanic Survival Rate Decision Tree [27].

Decision trees have a number of advantages over other learning algorithms. They are easy to understand and justify. Being a "white-box" model, end users are able to examine the various steps that the algorithm takes when making a decision. This gives users more information about the underlying dataset and builds confidence in the system. Decision trees also perform well with large datasets of both numerical and symbolic data. However, unless care

is taken to employ good training and/or pruning techniques, it can be easy for decision trees to become overly complex and overfit the training data. Also to ensure a globally optimal tree can be very computationally intensive, so most algorithms use local heuristics and are not guaranteed to generate globally optimal results.

2.3.4 Artificial Neural Networks

Artificial Neural Networks (ANNs) attempt to mimic the workings of a biological brain. All ANNs have the same basic component, namely, neurons, which are interconnected units of computation that receive signals, perform some processing, and propagate the results to other neurons in the network [16]. Most ANNs have a shared hierarchy. They have an input layer of neurons that receive the input samples, a variable number of hidden layers, with a varying number of neurons per layer, and finally, an output layer containing at least one neuron.

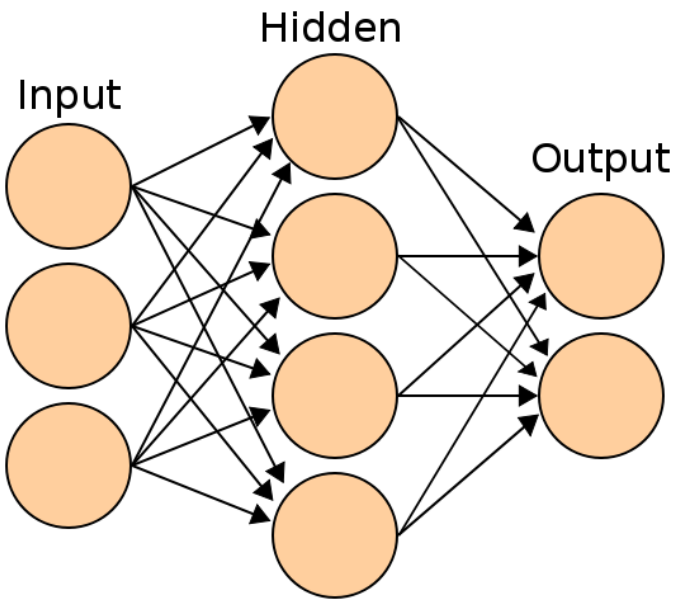


Figure 2.12 : Neural Network Structure [28]

All hidden layer and output layer neurons have *weights* associated with all inputs received by that neuron; this can be thought of as the strength of the connection between those neurons. Additionally, each neuron has an *activation function* that determines the output value

of that neuron based on the inputs it receives. Commonly, this takes the form of a *sigmoid* function, such as:

$$y = \frac{1}{1 + e^{-x}}$$

The advantage of using a non-linear activation function for a neuron is that the output is always bounded, in the above example, between 0 and 1 [16]. This prevents a single neuron from dominating the output of the entire network and allows the network more flexibility when being trained on more complex data sets.

Training a neural network involves two main steps: (1) showing the network a training sample, and based on the networks output, (2) updating the weights for each neuron based on the amount of error between the networks output and the "true" output for that sample. A common method of updating weights is called *backpropagation*, where the amount of error between the target output and the actual output is used to update the neuron weights one layer at a time from the output layer back to the input layer [16].

There are a few disadvantages to using neural networks. Once trained, a neural network becomes a "black-box" classifier. That is, they may perform well at classification or regression, but contribute little to the fundamental understanding of the data. It is unknown which features are highly relevant and which are irrelevant. Without experimentation, it is often difficult or impossible to determine how many neurons and how many hidden layers are needed to find an optimal solution for the current problem [16]. Finally, ANNs are prone to overfitting if great care is not taken during validation; overfitting is synonymous with memorization of the input data and leads to poor testing performance.

2.4 Analysis Techniques

Machine Learning techniques' success can be measured in a number of ways: (1) how long it takes to train the algorithm, (2) the accuracy rate of the algorithm, and (3) whether or not the algorithm can perform well on testing data. This thesis will use the following methods for evaluating model performance.

2.4.1 Ten-Fold Testing

Ten-Fold Testing is a specific instance of *k-fold cross-validation,* a method for training machine learning algorithms where the dataset is first divided into ten distinct "folds" [16]. After the dataset has been partitioned, the algorithm is trained with the first nine partitions and the tenth partition is left for model validation. The process is then repeated, where a different fold is held back for validation until the process is complete. This way, each sample participates during validation (nine times) when they were not used during training. This allows for a more complete measure of the model's accuracy for all data points and aims to avoid overfitting the training data, which usually results in poor testing performance.

2.4.2 Regression Model Performance Evaluation

The following techniques are statistical measures of how well a regression model fits the original dataset.

Pearson Product-Moment Correlation Coefficient

The *Pearson product-moment correlation* coefficient is a statistical measure of a regression model's "fit" to the data set [16]. The value ranges from -1 to 1 with the following meanings:

1 : perfect positive correlation, 0 : no correlation, -1 : perfect negative correlation. It is defined as:

$$\rho_{x,y} = \frac{cov(X,Y)}{\sigma_x \sigma_y}$$

Where *cov(X,Y)* is the covariance between populations *X* and *Y* and *σ* represents that population's standard deviation. For most regression models, the task is to model the existing data as closely as possible; thus, a correlation coefficient close to 1 is desired. However, another measure of model accuracy is needed since different datasets may have the same correlation coefficient but have a vastly different "shape" as demonstrated in Figure 2.13.

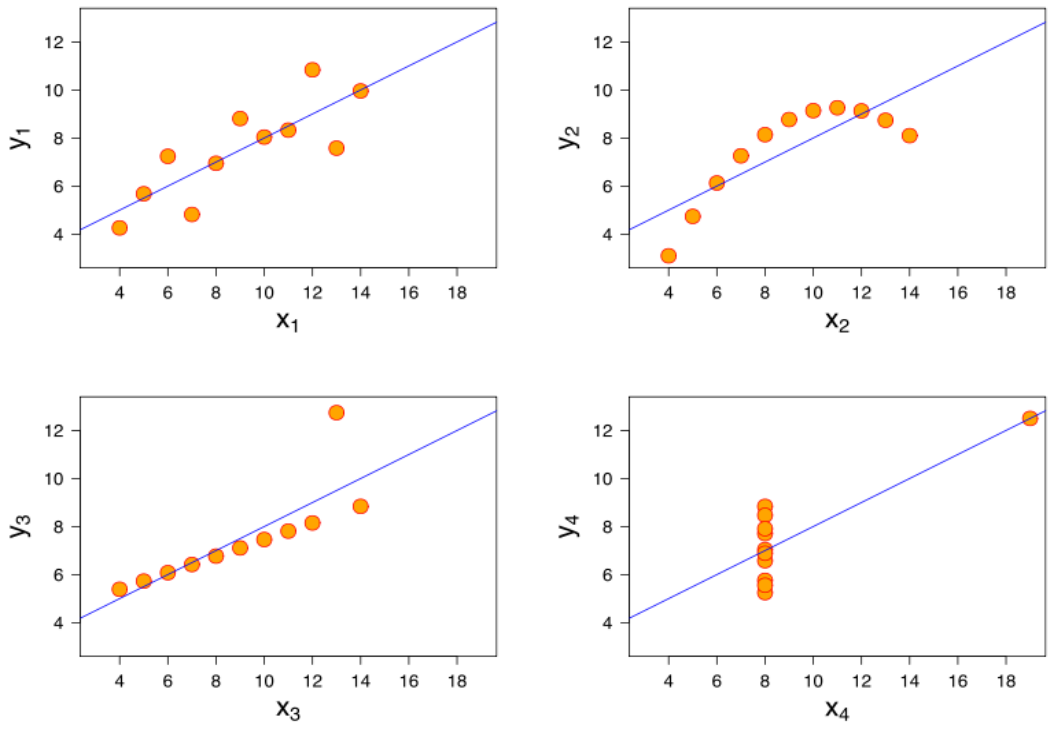


Figure 2.13: Anscombe's Quartet [29]

Mean Absolute Error

Mean absolute error, MAE, is a calculated value that is used to measure the amount of error between the model's prediction and the ground truth. Defined as: [16]

$$MAE = 1/n \sum_{i=1}^{n} |f_i - y_i|$$

Where *n* is the number of samples, f_i is the model's predicted value, and y_i is the actual value. Mean absolute error is a useful measure for determining how far the model's prediction deviates from the true value. However, it does not tell whether the error was an overestimation or underestimation of the true value, only the aggregate magnitude.

Root Mean Squared Error

Root mean squared error, or RMSE

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(f_i - y_i)^2}{n}}$$

Where *n* is the number of samples, f_i is the model's predicted value, and y_i is the actual value [16].

RMSE is more sensitive to large differences between the predicted value and the actual value. It is also more heavily influenced by "outliers" and often useful when the application cannot tolerate "large" prediction errors.

Relative Absolute Error and Root Relative Squared Error

Relative absolute error (RAE) is computed by dividing the MAE generated by the model by the MAE obtained with using the population mean as the prediction value:

$$RAE = 1/n \sum_{i=1}^{n} \frac{|f_i - y_i|}{|y_\mu - y_i|}$$

Root relative squared error (RRSE) is similar to RAE in that it compares the model's prediction performance with using a fixed prediction of the data's mean.

$$RRSE = 1/n \sqrt{\sum_{i=1}^{n} \frac{(f_i - y_i)^2}{(y_\mu - y_i)^2}}$$

Both are an indication of whether the model's prediction is more accurate than if the mean of the population had been used for prediction [16].

2.5 Related Work

There has been some previous exploration of calibration and control of ICE with various machine learning techniques in the past. Wang et al., (2006), and Zhai and Yu, () have explored the use of neural networks for steady state AFR control in engine simulations [4,18]. Vong Wong and Li, (2006), and Wong, et al. (2012), have explored the use of neural networks along with LS-SVM for prediction of engine power and torque and AFR control respectively using real world data [19,20]. Wong, (2012), experimented with compensation map calibration but did not attempt base map calibration [3]. However, all of the previous experiments have been performed on naturally aspirated four cylinder engines. These works show that the concept in itself is not novel, and has shown promise when used on different engine configurations, but has not yet been applied to turbocharged engines or Wankel rotary engines. Furthermore, the previous work was accomplished in a laboratory environment, which sophisticated and precise testing equipment, whereas this thesis will explore what can be accomplished with only the simple addition of a readily available exhaust gas oxygen sensor.

Chapter 3 Research Methodology

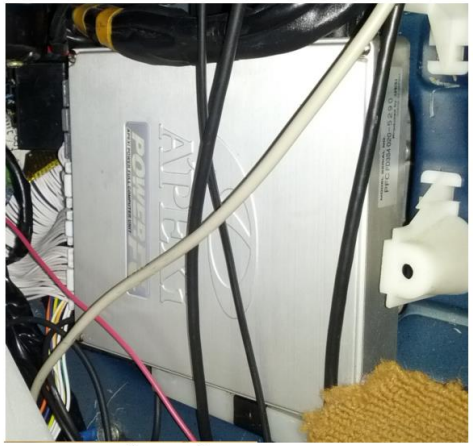
3.1 Data Collection Process

The data for this work was collected using a 1993 Mazda RX-7 equipped with a 13B-REW two-rotor Wankel rotary engine featuring sequential twin-turbochargers (Figure 3.1). The original ECU has been replaced by a programmable A'PEXi Power FC unit (Figure 3.1) [30]. Additionally an FC-Datalogit data acquisition system allows the end user to interact with the Power FC ECU directly using a laptop and Windows software package FC-Edit (Figure 3.1) [21]. Finally, an aftermarket Lambda sensor [31] was installed and integrated with the FC-Datalogit to be able to determine the engines air-fuel ratio by measuring the oxygen content of the exhaust gasses. Numerous other enhancements have been made to the intake, exhaust, cooling, and fueling systems which require the engine's calibration to be updated.

This particular engine management system has a myriad of tables and options to configure in order to properly calibrate an engine, including: Water Temperature Correction, Accelerator Enrichment, Throttle Position vs. Injection, Injection vs. Intake Temperature, etc. However, the most important and time-consuming table to calibrate is the "Base Map." The base map is a 20 x 20 table (Figure 3.2) where the horizontal axis is engine speed and the vertical axis is Pressure in Manifold (PIM).

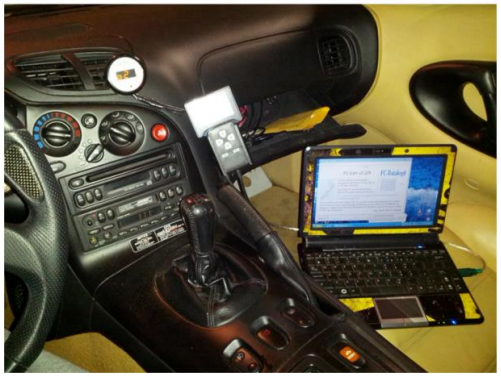
Test Vehicle
1993 Mazda RX-7

Test Engine
13B-REW



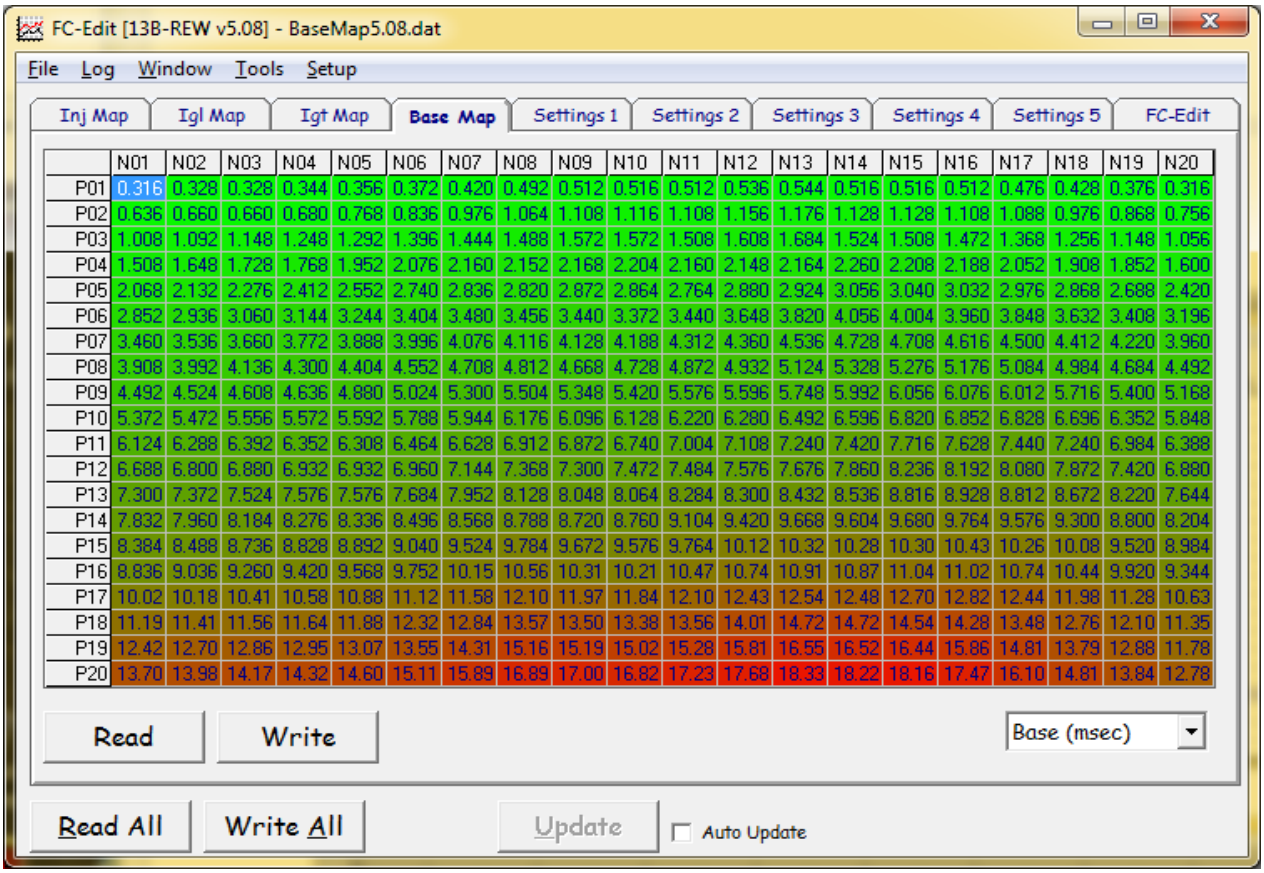
A'PEXi Power FC

FC-Datalogit



PC used for Data Collection

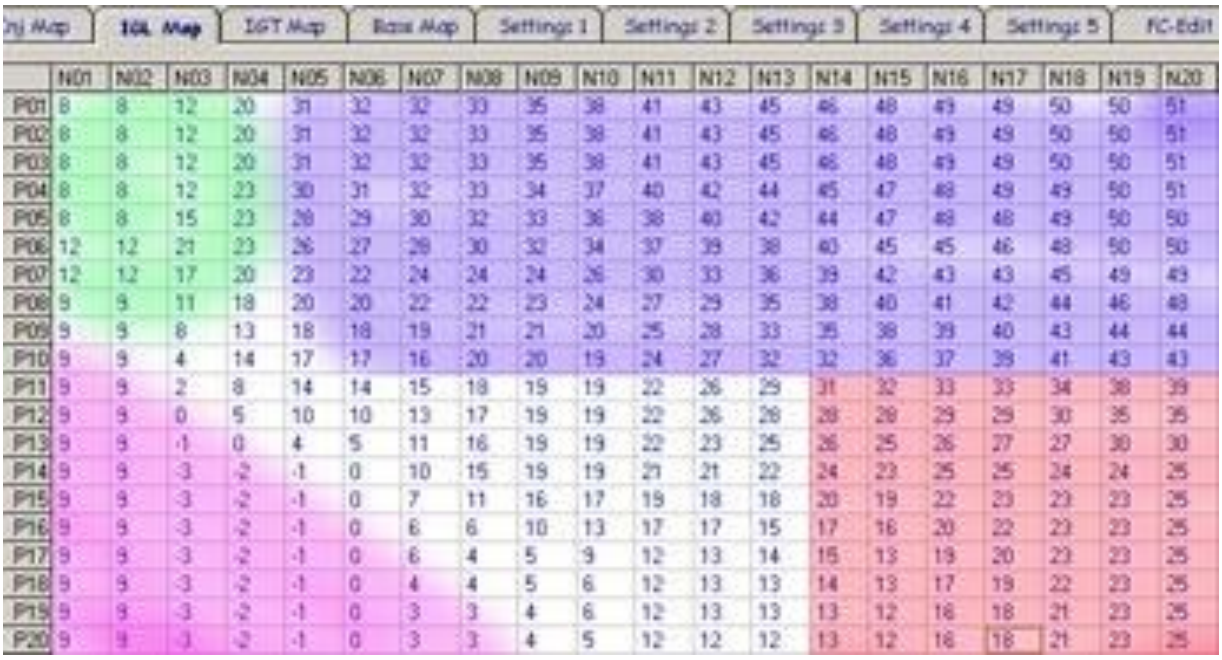
Figure 3.1: Experimental Setup



	N01	N02	N03	N04	N05	N06	N07	N08	N09	N10	N11	N12	N13	N14	N15	N16	N17	N18	N19	N20
P01	0.316	0.328	0.328	0.344	0.356	0.372	0.420	0.492	0.512	0.516	0.512	0.536	0.544	0.516	0.516	0.512	0.476	0.428	0.376	0.316
P02	0.636	0.660	0.660	0.680	0.768	0.836	0.976	1.064	1.108	1.116	1.108	1.156	1.176	1.128	1.128	1.108	1.088	0.976	0.868	0.756
P03	1.008	1.092	1.148	1.248	1.292	1.396	1.444	1.488	1.572	1.572	1.508	1.608	1.684	1.524	1.508	1.472	1.368	1.256	1.148	1.056
P04	1.508	1.648	1.728	1.768	1.952	2.076	2.160	2.152	2.168	2.204	2.160	2.148	2.164	2.260	2.208	2.188	2.052	1.908	1.852	1.600
P05	2.068	2.132	2.276	2.412	2.552	2.740	2.836	2.820	2.872	2.864	2.764	2.880	2.924	3.056	3.040	3.032	2.976	2.868	2.688	2.420
P06	2.852	2.936	3.060	3.144	3.244	3.404	3.480	3.456	3.440	3.372	3.440	3.648	3.820	4.056	4.004	3.960	3.848	3.632	3.408	3.196
P07	3.460	3.536	3.660	3.772	3.888	3.996	4.076	4.116	4.128	4.188	4.312	4.360	4.536	4.728	4.708	4.616	4.500	4.412	4.220	3.960
P08	3.908	3.992	4.136	4.300	4.404	4.552	4.708	4.812	4.668	4.728	4.872	4.932	5.124	5.328	5.276	5.176	5.084	4.984	4.684	4.492
P09	4.492	4.524	4.608	4.636	4.880	5.024	5.300	5.504	5.348	5.420	5.576	5.596	5.748	5.992	6.056	6.076	6.012	5.716	5.400	5.168
P10	5.372	5.472	5.556	5.572	5.592	5.788	5.944	6.176	6.096	6.128	6.220	6.280	6.492	6.596	6.820	6.852	6.828	6.696	6.352	5.848
P11	6.124	6.288	6.392	6.352	6.308	6.464	6.628	6.912	6.872	6.740	7.004	7.108	7.240	7.420	7.716	7.628	7.440	7.240	6.984	6.388
P12	6.688	6.800	6.880	6.932	6.932	6.960	7.144	7.368	7.300	7.472	7.484	7.576	7.676	7.860	8.236	8.192	8.080	7.872	7.420	6.880
P13	7.300	7.372	7.524	7.576	7.576	7.684	7.952	8.128	8.048	8.064	8.284	8.300	8.432	8.536	8.816	8.928	8.812	8.672	8.220	7.644
P14	7.832	7.960	8.184	8.276	8.336	8.496	8.568	8.788	8.720	8.760	9.104	9.420	9.668	9.604	9.680	9.764	9.576	9.300	8.800	8.204
P15	8.384	8.488	8.736	8.828	8.892	9.040	9.524	9.784	9.672	9.576	9.764	10.12	10.32	10.28	10.30	10.43	10.26	10.08	9.520	8.984
P16	8.836	9.036	9.260	9.420	9.568	9.752	10.15	10.56	10.31	10.21	10.47	10.74	10.91	10.87	11.04	11.02	10.74	10.44	9.920	9.344
P17	10.02	10.18	10.41	10.58	10.88	11.12	11.58	12.10	11.97	11.84	12.10	12.43	12.54	12.48	12.70	12.82	12.44	11.98	11.28	10.63
P18	11.19	11.41	11.56	11.64	11.88	12.32	12.84	13.57	13.50	13.38	13.56	14.01	14.72	14.72	14.54	14.28	13.48	12.76	12.10	11.35
P19	12.42	12.70	12.86	12.95	13.07	13.55	14.31	15.16	15.19	15.02	15.28	15.81	16.55	16.52	16.44	15.86	14.81	13.79	12.88	11.78
P20	13.70	13.98	14.17	14.32	14.60	15.11	15.89	16.89	17.00	16.82	17.23	17.68	18.33	18.22	18.16	17.47	16.10	14.81	13.84	12.78

Figure 3.2: Base Map.

The ranges and increments of this table can be modified by the user but the default settings were used in this experiment. The default ranges for this table for engine speed start at 400 RPM and increase in increments of 400 until 8000 RPM (N01-N20) and from 1000 to 24000 PIM, in increments of 1000. The Power FC documentation states that a PIM value of 10,000 is equal to 1 kg/cm^2, so it can be inferred that any row below P10 is a "vacuum" region, where the pressure is less than atmospheric (i.e., "throttled") [9,10,21]. Likewise, any row above and including P10 is where the pressure in the manifold is greater than or equal to atmospheric pressure (i.e., "unthrottled" or "boosted"). The values in this table represent the milliseconds that a fuel injector needs to open. For example, at P10-N05, atmospheric pressure and ~2000 RPM, each cycle requires 5.592 milliseconds of on-time. Actual fuel deliver depends on injector size and fuel line pressure.



	N01	N02	N03	N04	N05	N06	N07	N08	N09	N10	N11	N12	N13	N14	N15	N16	N17	N18	N19	N20
P01	8	8	12	20	31	32	32	33	35	38	41	43	45	46	48	49	49	50	50	51
P02	8	8	12	20	31	32	32	33	35	38	41	43	45	46	48	49	49	50	50	51
P03	8	8	12	20	31	32	32	33	35	38	41	43	45	46	48	49	49	50	50	51
P04	8	8	12	23	30	31	32	33	34	37	40	42	44	45	47	48	49	49	50	51
P05	8	8	15	23	28	29	30	32	33	36	38	40	42	44	47	48	48	49	50	50
P06	12	12	21	23	26	27	28	30	32	34	37	39	38	40	45	45	46	48	50	50
P07	12	12	17	20	23	22	24	24	24	26	30	33	36	39	42	43	43	45	49	49
P08	9	9	11	18	20	20	22	22	23	24	27	29	35	38	40	41	42	44	46	48
P09	9	9	8	13	18	18	19	21	21	20	25	28	33	35	38	39	40	43	44	44
P10	9	9	4	14	17	17	16	20	20	19	24	27	32	32	36	37	39	41	43	43
P11	9	9	2	8	14	14	15	18	19	19	22	26	29	31	32	33	33	34	38	39
P12	9	9	0	5	10	10	13	17	19	19	22	26	28	28	28	29	29	30	35	35
P13	9	9	-1	0	4	5	11	16	19	19	22	23	25	26	25	26	27	27	30	30
P14	9	9	-3	-2	-1	0	10	15	19	19	21	21	22	24	23	25	25	24	24	25
P15	9	9	-3	-2	-1	0	7	11	16	17	19	18	18	20	19	22	23	23	23	25
P16	9	9	-3	-2	-1	0	6	6	10	13	17	17	15	17	16	20	22	23	23	25
P17	9	9	-3	-2	-1	0	6	4	5	9	12	13	14	15	13	19	20	23	23	25
P18	9	9	-3	-2	-1	0	4	4	5	6	12	13	13	14	13	17	19	22	23	25
P19	9	9	-3	-2	-1	0	3	3	4	6	12	13	13	13	12	16	18	21	23	25
P20	9	9	-3	-2	-1	0	3	3	4	5	12	12	12	13	12	16	18	21	23	25

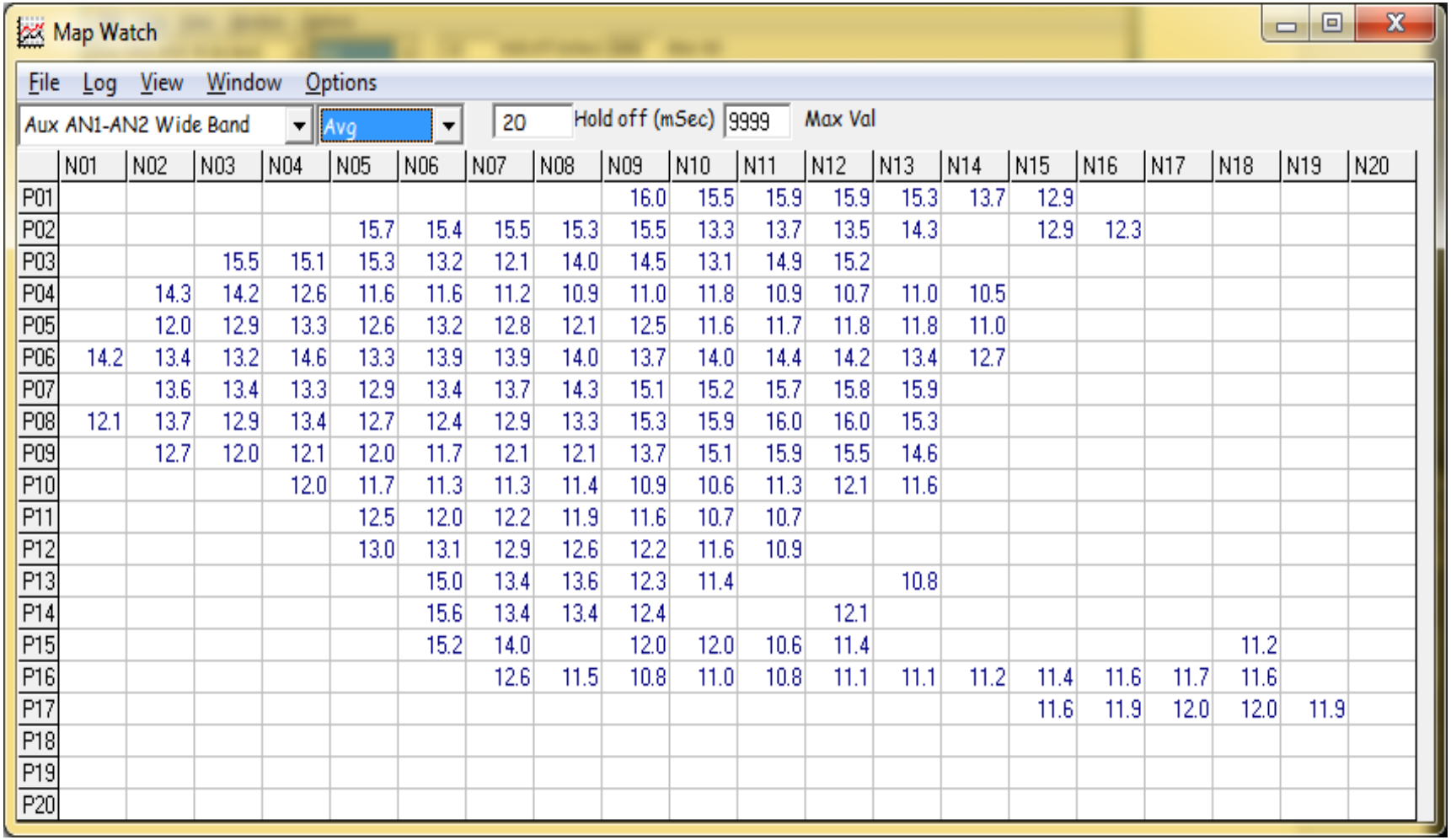
Figure 3.3: Engine Operating Regimes [9].

In Figure 3.3, shows the same table as Figure 3.2 but with the various operating regimes highlighted. The shaded section in the upper left corner is the engine idling and startup region. Here, torque requirements are low and just enough fuel is added to keep the engine running. The rectangle in the upper right represents part-throttle cruising area. This is the area that is most often used in everyday driving, low to moderate load and low to high engine speeds. The triangular area in the lower left is high load (wide open throttle) and low engine speed. Since this is a turbocharged engine, there is not enough energy in the exhaust gasses to adequately spin up the turbocharger in this region, so it is unlikely that the engine will ever operate in this area. Finally, the square in the lower right is the high-load high engine speed region. This is where the most power is produced and peak chamber pressures occur [9,10].

The values in each cell of the base map correspond to injector “on-time” in milliseconds, or how long the injector should remain open to provide adequate fueling [9,10]. Further complicating matters, RX-7s are equipped with a staged fuel injection system. The reason for this is that if each rotor were equipped with a single injector in the intake port, that injector

would be required to service three rotor faces (cylinders in a conventional engine). To meet fueling requirements at high engine speed this would require an oversized injector, but larger injectors are more difficult to control during low load and low engine speeds, and they may provide more fuel than required. With staged injection, the engine uses a single small injector, 550 cc/min, for each rotor during low load and low engine speed operation. When more fuel is required, the secondary injectors, which were originally 850 cc/min but have since been replaced with 1680 cc/min units, begin to operate in parallel with the primary injectors to meet fueling needs [9,10]. This complicates calibration, since larger injectors deliver greater quantities of fuel at lower pulse widths.

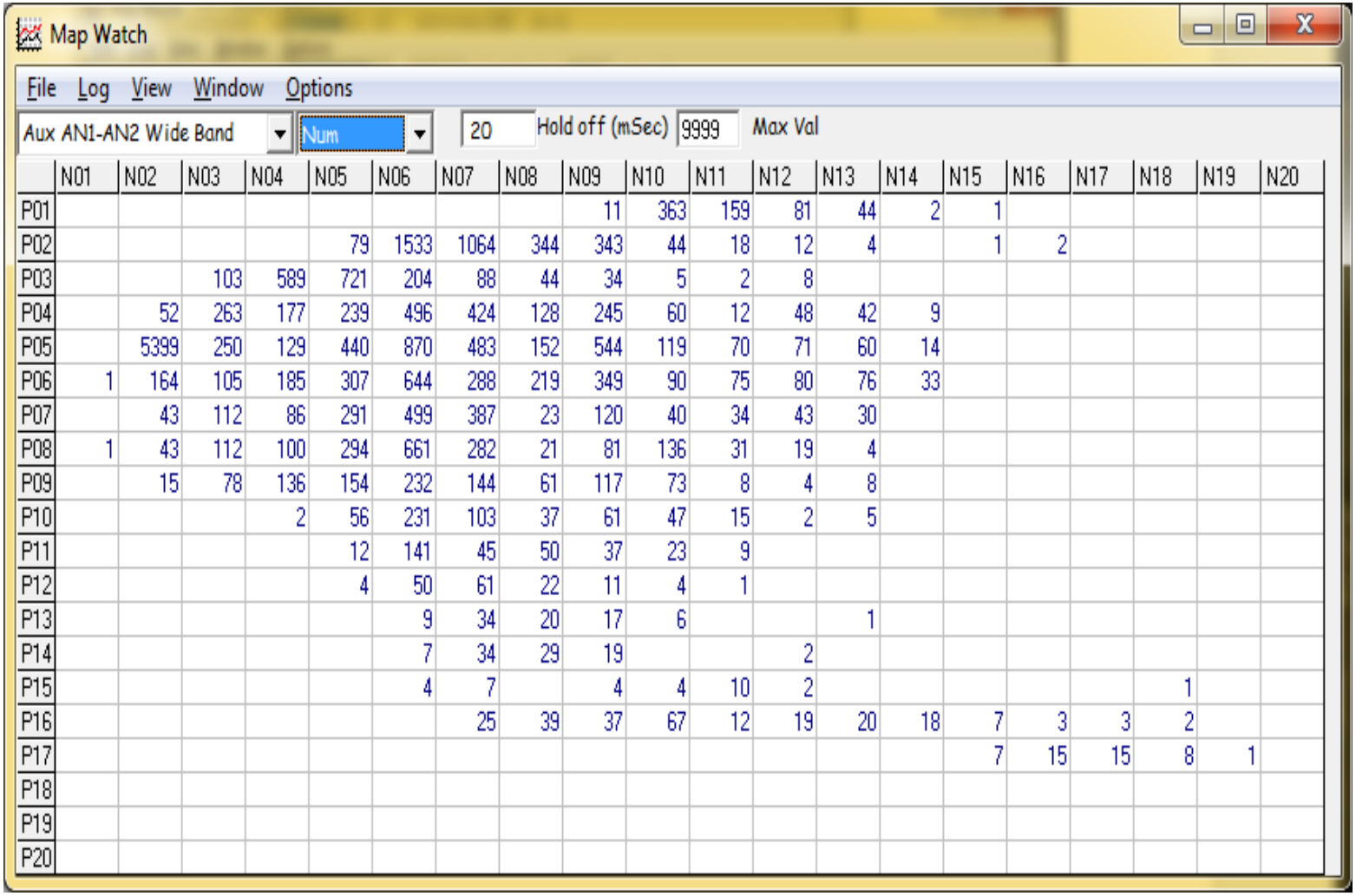
Data was gathered on 74 different channels for approximately 40 minutes of mixed city and highway driving. The goal while gathering data was to "exercise" the engine and try to have it operate in as many cells of the base map as possible to determine the overall accuracy of the supplied base map. Figure 3.4 shows the cumulative average AFR for each cell of the base map. It should be noted that adjacent cells should have similar values. There should be a general "smoothness" in the AFR transition between cells. An example where this is not the case occurs in column N06 between P12 and P13. The AFR reading jumps from 13.1 to 15.0 indicating that this area of the map is not properly calibrated. Another area that requires calibration is column N10 between P09 and P10, with a jump from 15.1 to 10.6 AFR between adjacent cells. This data shows the AFRs produced from the current engine calibration and can be used to determine how far the calibration is from ideal and to train our model.



	N01	N02	N03	N04	N05	N06	N07	N08	N09	N10	N11	N12	N13	N14	N15	N16	N17	N18	N19	N20
P01									16.0	15.5	15.9	15.9	15.3	13.7	12.9					
P02					15.7	15.4	15.5	15.3	15.5	13.3	13.7	13.5	14.3		12.9	12.3				
P03			15.5	15.1	15.3	13.2	12.1	14.0	14.5	13.1	14.9	15.2								
P04		14.3	14.2	12.6	11.6	11.6	11.2	10.9	11.0	11.8	10.9	10.7	11.0	10.5						
P05		12.0	12.9	13.3	12.6	13.2	12.8	12.1	12.5	11.6	11.7	11.8	11.8	11.0						
P06	14.2	13.4	13.2	14.6	13.3	13.9	13.9	14.0	13.7	14.0	14.4	14.2	13.4	12.7						
P07		13.6	13.4	13.3	12.9	13.4	13.7	14.3	15.1	15.2	15.7	15.8	15.9							
P08	12.1	13.7	12.9	13.4	12.7	12.4	12.9	13.3	15.3	15.9	16.0	16.0	15.3							
P09		12.7	12.0	12.1	12.0	11.7	12.1	12.1	13.7	15.1	15.9	15.5	14.6							
P10				12.0	11.7	11.3	11.3	11.4	10.9	10.6	11.3	12.1	11.6							
P11					12.5	12.0	12.2	11.9	11.6	10.7	10.7									
P12					13.0	13.1	12.9	12.6	12.2	11.6	10.9									
P13						15.0	13.4	13.6	12.3	11.4			10.8							
P14						15.6	13.4	13.4	12.4			12.1								
P15						15.2	14.0		12.0	12.0	10.6	11.4						11.2		
P16							12.6	11.5	10.8	11.0	10.8	11.1	11.1	11.2	11.4	11.6	11.7	11.6		
P17															11.6	11.9	12.0	12.0	11.9	
P18																				
P19																				
P20																				

Figure 3.4: Map Watch Average AFR.

Figure 3.5 is a histogram of how many samples are related to each cell in the base map. It illustrates the relative frequency of engine speed and load values. For example, the large number of samples occurring at N02 and P05 show where engine idling usually occurs, approximately 800-1000 RPM and 17 inHg vacuum. The first two P rows, P01 and P02, represent closed-throttle deceleration, where the accelerator is not being pressed and the engine and vehicle are slowing. this is also the area that corresponds to very high AFRs, since fuel is not injected during deceleration. The main cruising area resides between N04 - N10 (1200 - 4000 RPM), and P03-P10 (part throttle until slight positive pressure). This also explains why the majority of the samples occur in this area, as most driving takes place in this regime. The bottom grouping between P16-P17 represents maximum acceleration, such as merging onto the highway from an on-ramp, and the few number of samples in this area are a testament to how infrequently maximum power is required during everyday operation.



	N01	N02	N03	N04	N05	N06	N07	N08	N09	N10	N11	N12	N13	N14	N15	N16	N17	N18	N19	N20
P01									11	363	159	81	44	2	1					
P02					79	1533	1064	344	343	44	18	12	4		1	2				
P03			103	589	721	204	88	44	34	5	2	8								
P04		52	263	177	239	496	424	128	245	60	12	48	42	9						
P05		5399	250	129	440	870	483	152	544	119	70	71	60	14						
P06	1	164	105	185	307	644	288	219	349	90	75	80	76	33						
P07		43	112	86	291	499	387	23	120	40	34	43	30							
P08	1	43	112	100	294	661	282	21	81	136	31	19	4							
P09		15	78	136	154	232	144	61	117	73	8	4	8							
P10				2	56	231	103	37	61	47	15	2	5							
P11					12	141	45	50	37	23	9									
P12					4	50	61	22	11	4	1									
P13						9	34	20	17	6			1							
P14						7	34	29	19			2								
P15						4	7		4	4	10	2						1		
P16							25	39	37	67	12	19	20	18	7	3	3	2		
P17															7	15	15	8	1	
P18																				
P19																				
P20																				

Figure 3.5: Map Watch Hit Count.

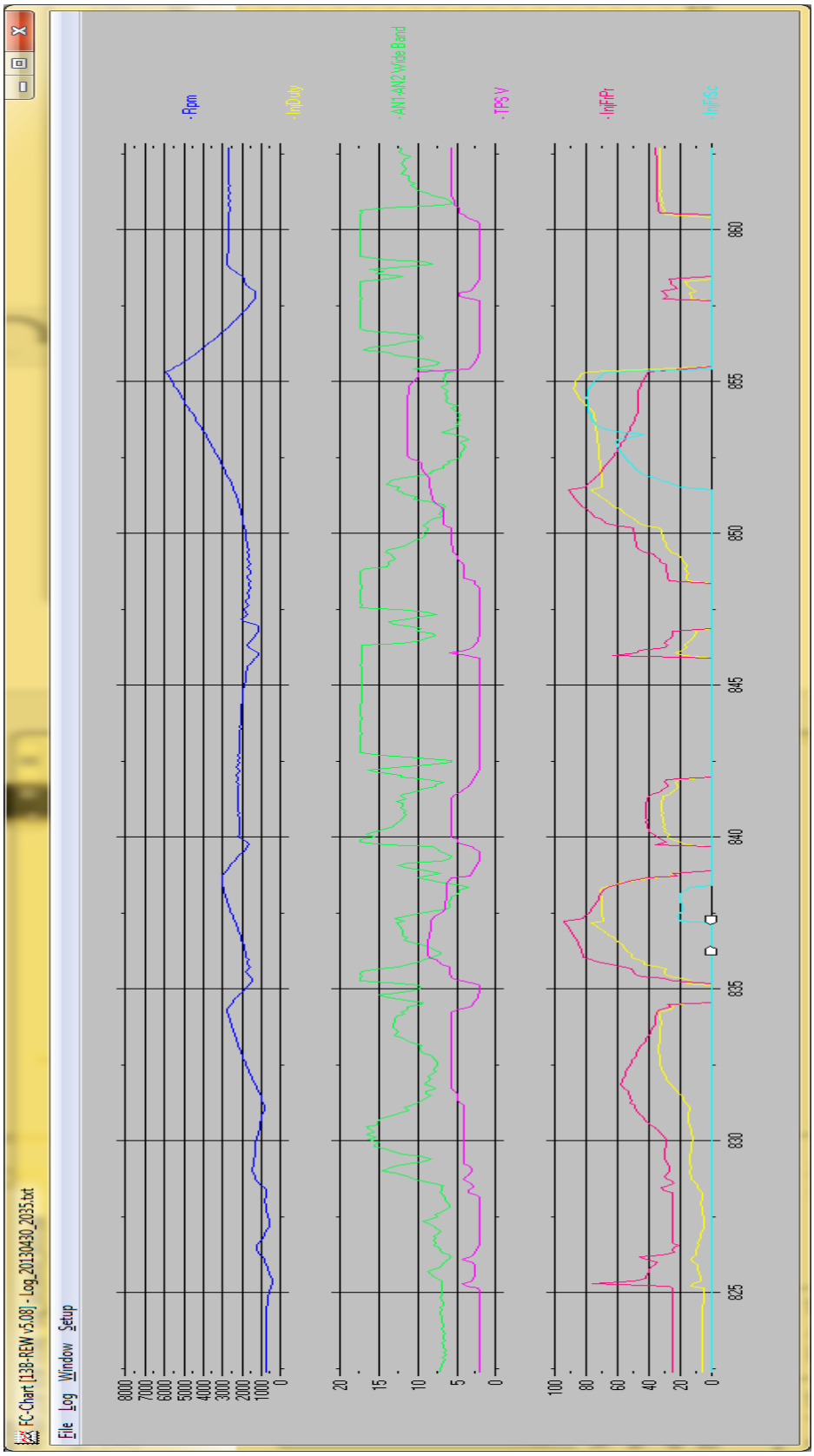


Figure 3.6: Sample Section Taken from Datalog.

3.2 Preprocessing and Feature Selection

In addition to AFR, a number of other samples were logged. A sample of the available channels is shown in Figure 3.6. This original data set required preprocessing to remove duplicate channels and non-relevant data. These include cases when the engine was turned off, during deceleration when no fuel is being injected, channels that did not vary throughout the drive, etc. Table 3.1 illustrates the final channels used to train the models.

Logged Data Channels		
TimeS timestamp	**MAPN** column reference	**PWS** power steering sensor
InjDuty injector duty (ms)	**MAPP** row reference	**NTR** neutral switch
RPM engine speed	**IGL** leading spark plug timing	**CLT** clutch switch
PIM pressure in manifold (MAP sensor)	**IGT** trailing spark plug timing	**STP** stop sensor
WtrT water temperature (C)	**Inj** ???	**HWL** egt warning light
AirT intake air temperature (C)	**Oil** oil injection	**FPR** fuel pressure regulator switch
Knock knock sensor value	**PC** pre-control	APR - ???
Speed speed sensor (km/hr)	**InjFrPr** primary injector on-time	**CCN** charge control
WideBand AFR sensor	**InjFrSc** secondary injector on-time	**TCN** turbo control
	TPSV throttle position sensor voltage	**VTA1** throttle voltage, full range
	PIM MAP sensor voltage	**VTA2** throttle voltage, narrow range

Table 3.1: Selected Data Channels

Due to the location of the UEGO sensor, AFR as recorded in the datalog, actually corresponded to events occurring approximately one second, or 10 samples, previously. Figure 3.7 shows the time-dependent nature of the exhaust gas sensor. The top plot is for the exhaust gas sensor and the middle and bottom plots represent injector duty and throttle position.

When the throttle is closed, the injector duty immediately drops to zero. During this time AFR fluctuates briefly before stabilizing at the maximum lean value. This delay required a translation of the AFR signal so that the values would accurately reflect the correct thermodynamic cycle. However, this is a rather simplistic approach, and different approaches should be explored in future work to account for engine speeds effect on exhaust gas flow rate.

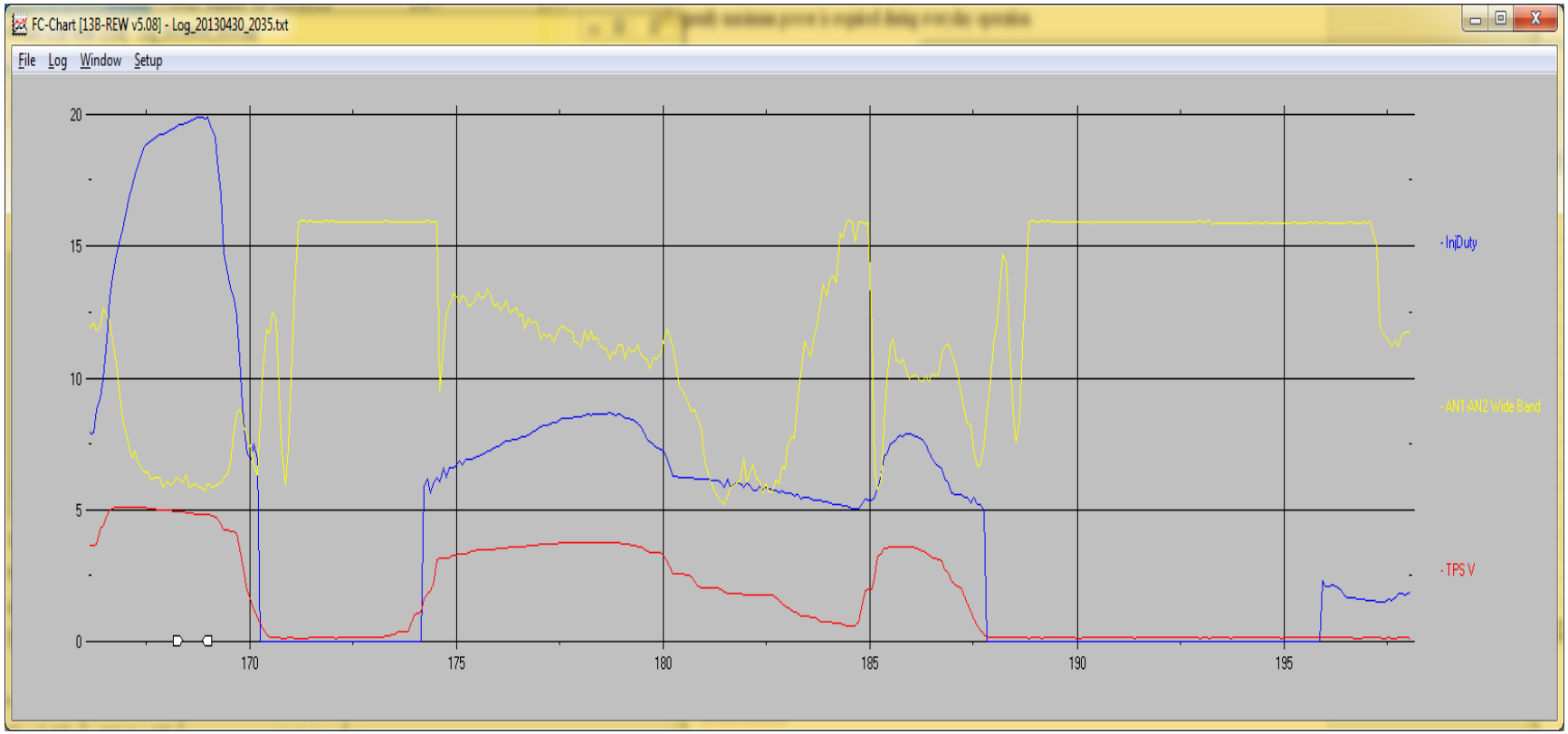


Figure 3.7: Lambda Sensor Response Time

After the redundant or invariant channels and samples had been removed, the raw data contained 37 total channels with 23,477 samples. This data was converted to ARFF file format for importation into Weka [15,22].

3.3 AI Technique Selection

A number of machine learning algorithms were evaluated, specifically a few variations of decision trees and neural networks.

3.3.1 Decision Trees

Weka's *REPTree* is an algorithm designed to build regression trees with speed in mind. It uses information gain/variance reduction for choosing attributes and prunes the resulting tree using "reduced-error pruning" similar to C4.5. It was chosen for this experiment because it is suitable for regression models, as well as classification systems [16].

3.3.2 Neural Networks

Weka's *Multilayer Perceptron* ANN is a feedforward, backpropagation neural network containing a single fully-connected hidden layer and a single output neuron by default. It also can be used for either classification or regression [16].

3.4 Fuel Map Correction and Validation

Once the model has been created and validated, new values for the base map are calculated based on the Mazda target AFR map.

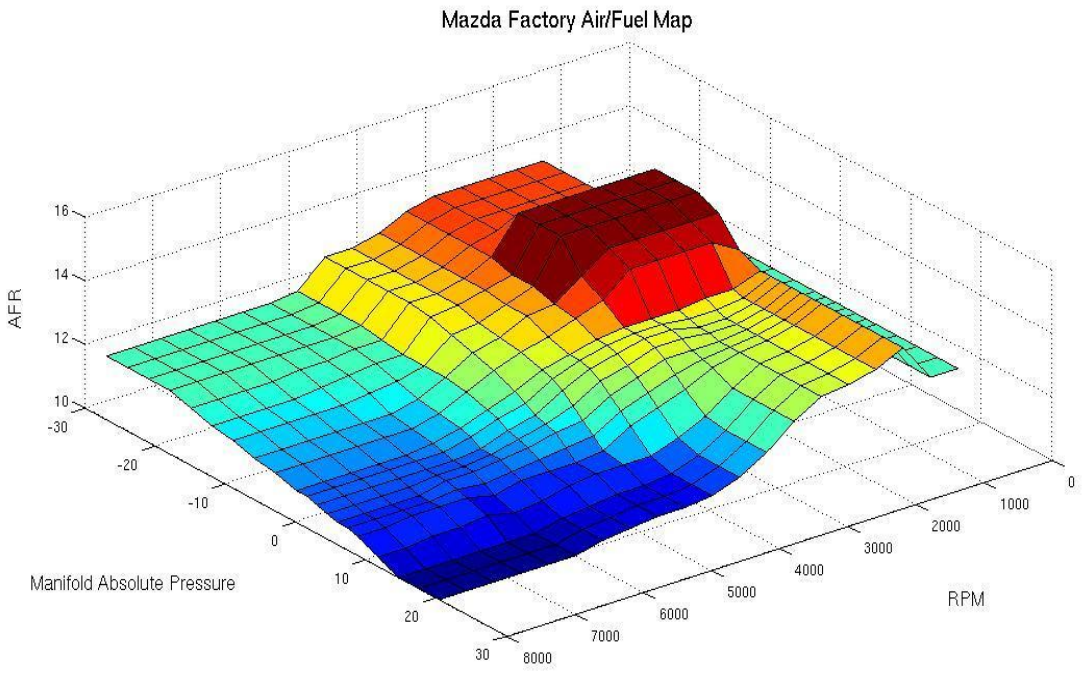


Figure 3.8: Mazda Factory RX-7 Target AFRs []

Figure 3.8 shows the target air-fuel ratio map. Since this is an OEM AFR map emissions are a primary concern. Therefore, the dark red high area, which has an AFR of 14.7, corresponds to the low load, low engine speed cruising area. As both engine speed (RPM) and load (MAP) increase, there is a gradual richening of the AFR, with a steeper increase along the load axis than the RPM axis. Since peak power occurs at approximately 13.1 AFR, the reason for richening past this point is most likely knock control and to keep combustion chamber temperatures from being excessive.

Figure 3.9 shows the percent error difference between the measured log average AFR values and the Mazda target AFRs. It has been normalized so that only cells present in the datalog are represented. Red areas indicate sections of the map that are richer than the target AFR; blue areas are leaner.

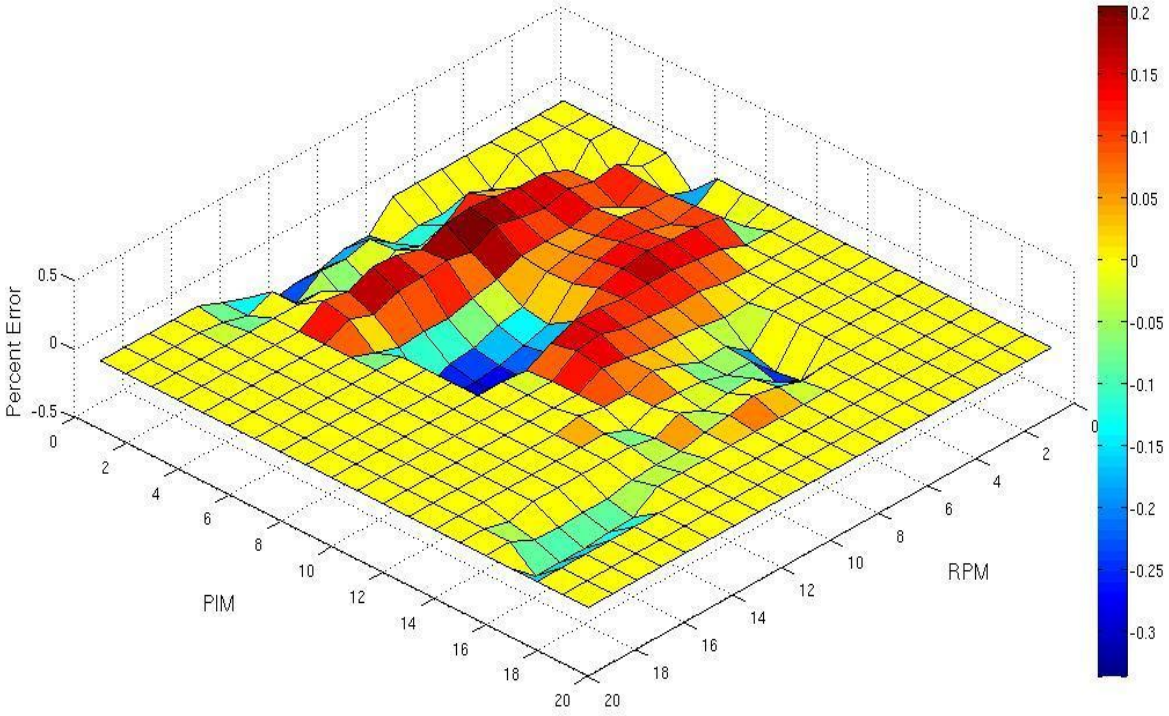


Figure 3.9: Percent Error, Measured AFR vs. Mazda Target AFR.

This gives a good indication of how "close" the starting base map is to the target map. It is readily apparent that some sections of the map are far too rich while other adjacent areas are too lean. The goal is to get this map as close to the target AFRs as possible so that the percent difference is nearly zero.

Finally, Figure 3.10 shows the three dimensional representation of the fuel injection basemap. These values are roughly equivalent to the volumetric efficiency of the engine, multiplied by the fuel equivalence ratio or 1/λ. It is this table that needs to be properly calibrated for the new engine configuration. As load and engine speed increase, VE increases and with it, fueling requirements. It is also a relatively smooth, continuous, and nearly monotonic surface. This is expected since the amount of air an engine consumes should be a function of engine speed and throttle position (load). Also around the 10th PIM row the rate of increase along the

PIM axis undergoes a noticeable increase. Since this is a turbocharged engine, this area of rapid increase marks the transition from throttled operation to boosted operation. Since the turbochargers are forcing more air into the engine than would normally be possible, a proportional amount of fuel also needs to be added as reflected on the map.

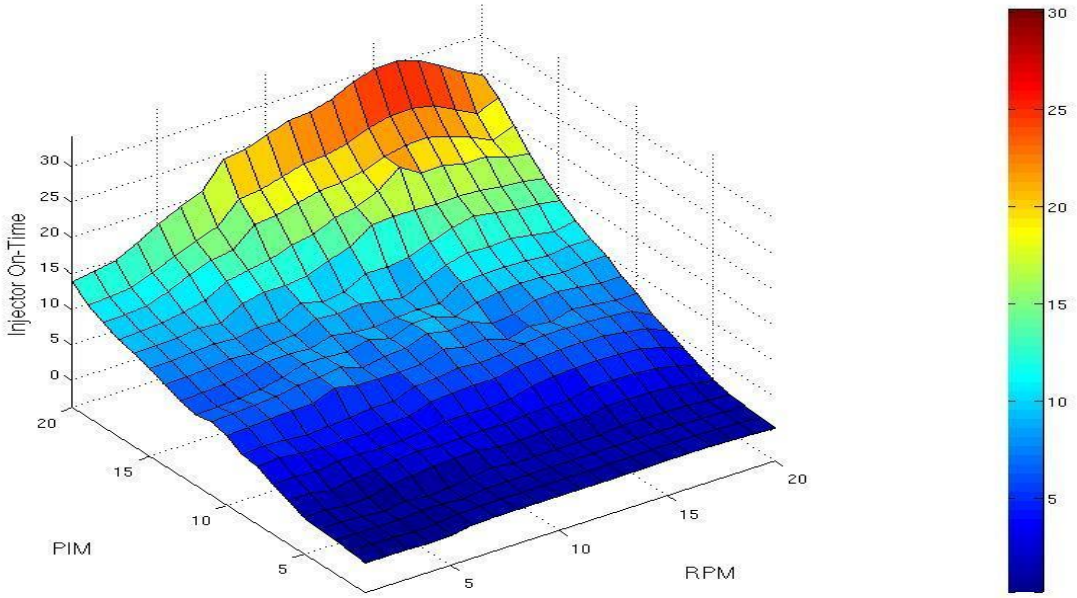


Figure 3.10: Fuel Injection Basemap

Chapter 4 Experimental Results

4.1 Predicting AFR

The first attempt to develop a predictive model using the logged engine data was to see if the AFR could be predicted. All models were validated using ten-fold testing. All channels in table 3.1, with the exception of Wideband-AFR, sensor were used as input in an attempt to predict the Wideband-AFR value.

4.1.1 Decision Trees

Using the REPTree algorithm in Weka, the results in Table 4.1 were obtained.

Correlation coefficient	0.872
Mean absolute error	0.370
Root mean squared error	0.672
Relative absolute error	33.81 %
Root relative squared error	49.15 %
Total number of instances	23477
Size	2017
Training Time	0.8 seconds

Table 4.1: REPTree (AFR).

The results of the REPTree algorithm were slightly improved with Bagging (Bootstrap Aggregating).

Correlation coefficient	0.907
Mean absolute error	0.325
Root mean squared error	0.54
Relative absolute error	31.32 %
Root relative squared error	42.20 %
Total Number of Instances	23477
Size	2419
Training time	3.92 seconds

Table 4.2 Bagged REPTree (AFR)

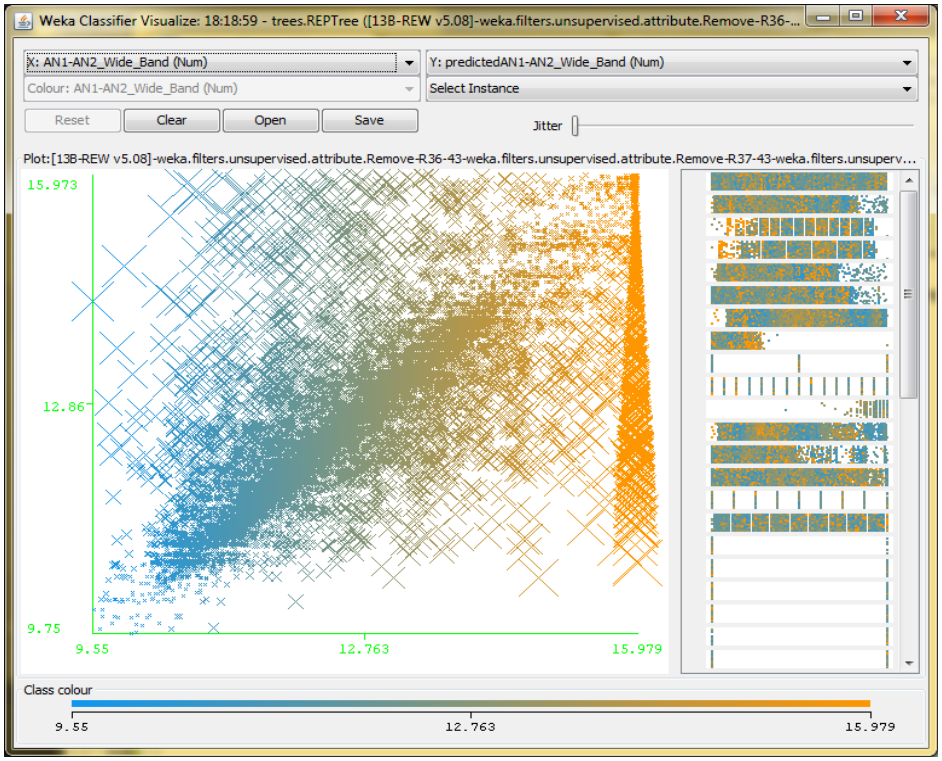


Figure 4.1: Decision Tree Classification Errors (AFR)

While the correlation coefficient was reasonably high, all error measurements were too high for usable predictions. Notably, a large number of prediction errors occurred when predicting truly lean samples, the grouping on the far right vertical axis (Figure), and as indicated by the high RMSE. While there is a nearly linear correlation, the magnitude and of a number of the errors prevent this model from being used for meaningful predictions (RRSE slightly better than chance) (Table 4.1, Table 4.2).

4.1.2 Neural Networks

The multilayer perceptron neural network gave the results shown in Table 4.3 when attempting to predict AFR.

Correlation coefficient	0.802
Mean absolute error	0.567
Root mean squared error	0.828
Relative absolute error	51.84 %
Root relative squared error	60.58 %
Total Number of Instances	23477
Training Time	186 seconds

Table 4.3 Multilayer Perceptron (AFR)

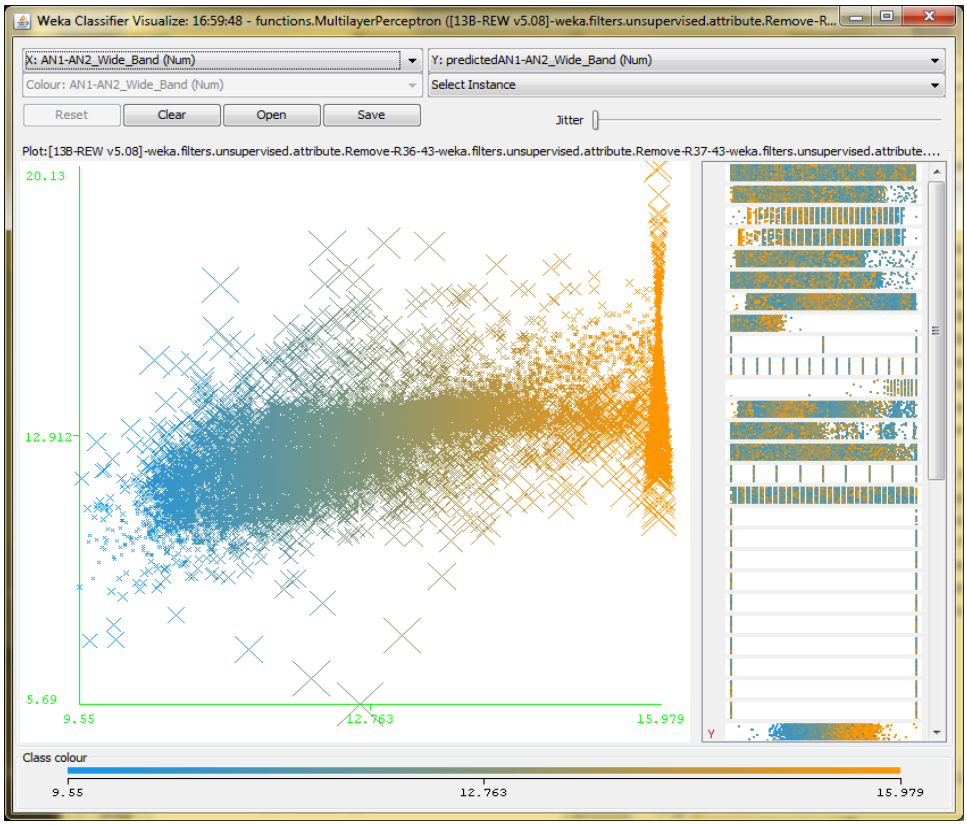


Figure 4.2: Neural Network Classification Errors (AFR)

Figure 4.2 shows the results of the neural network. The overall predictive value of the neural network is far worse, in every regard, when compared with decision trees (Table 4.3). Some of the failures were similar, especially considering the extremely lean datapoints, but the overall model is less accurate and instead tends to predict a value around ~12.9 AFR for most cases. Clearly, another approach must be taken to find a useable model for engine calibration.

4.2 Predicting Injector Duty Cycle

The next experiment attempted to model injector duty cycle based on the other observed data points (Table 3.1). The motivation for this experiment is that once the model is generated, all that is required to determine the new base map is to test the model with fabricated data based on the target AFR values taken from the Mazda target AFR map.

4.2.1 Decision Trees

Again Weka's REPTree was used to model the relationship between injector duty cycle and the other monitored channels with the results in Table 4.4.

Correlation coefficient	0.9981
Mean absolute error	0.379
Root mean squared error	0.693
Relative absolute error	4.21 %
Root relative squared error	6.20 %
Total Number of Instances	23477
Size	1193
Training Time	0.31 seconds

Table 4.4 REPTree (Inj. Duty Cycle)

These results are already significantly better than those obtained by trying to predict AFR. The error rate has dropped precipitously, in the case of RRSE by a factor of 10 (Table 4.4) and our model looks to have a much "tighter" linear relationship with the dataset (Figure 4.3).

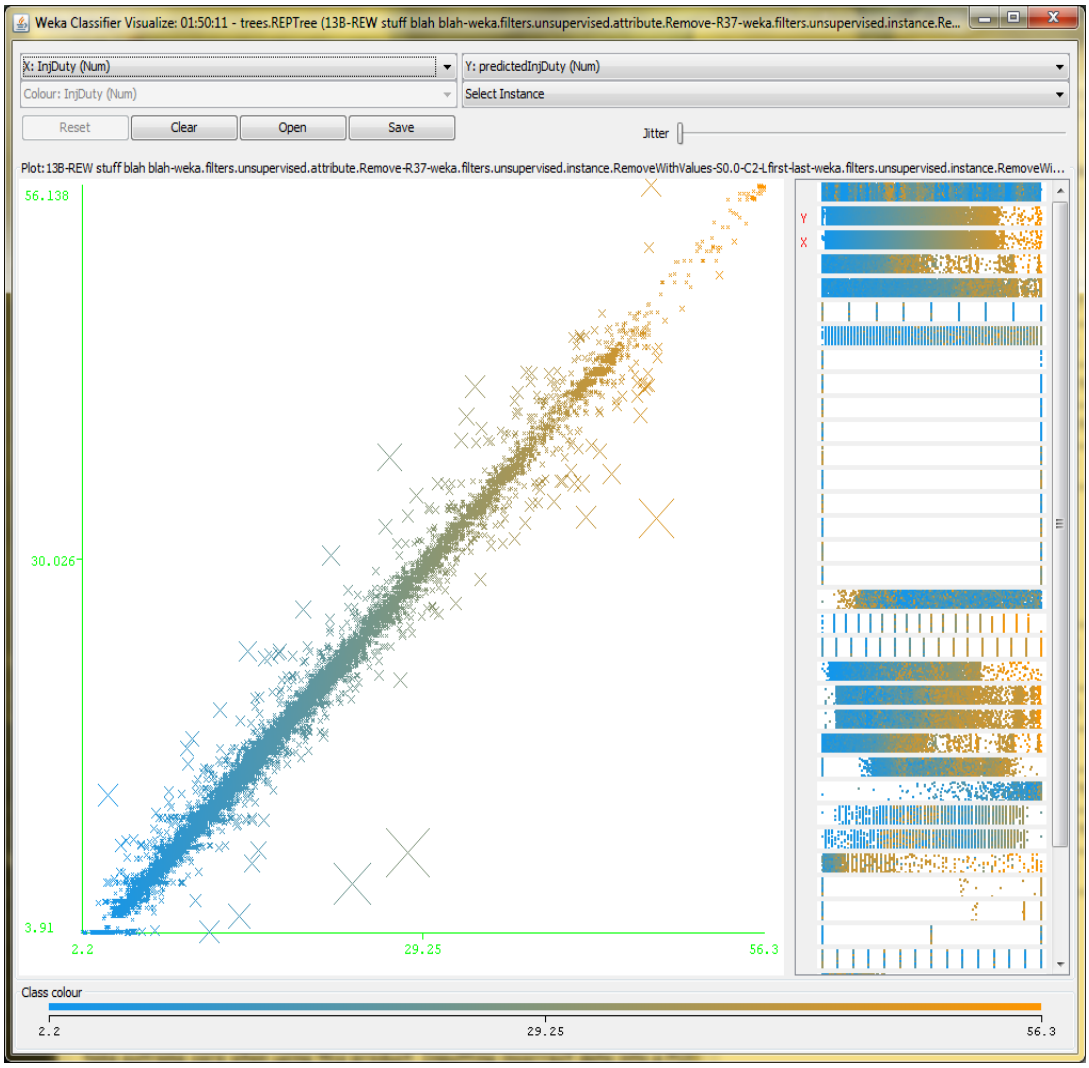


Figure 4.3 Decision Tree Prediction Errors (Inj. Duty Cycle

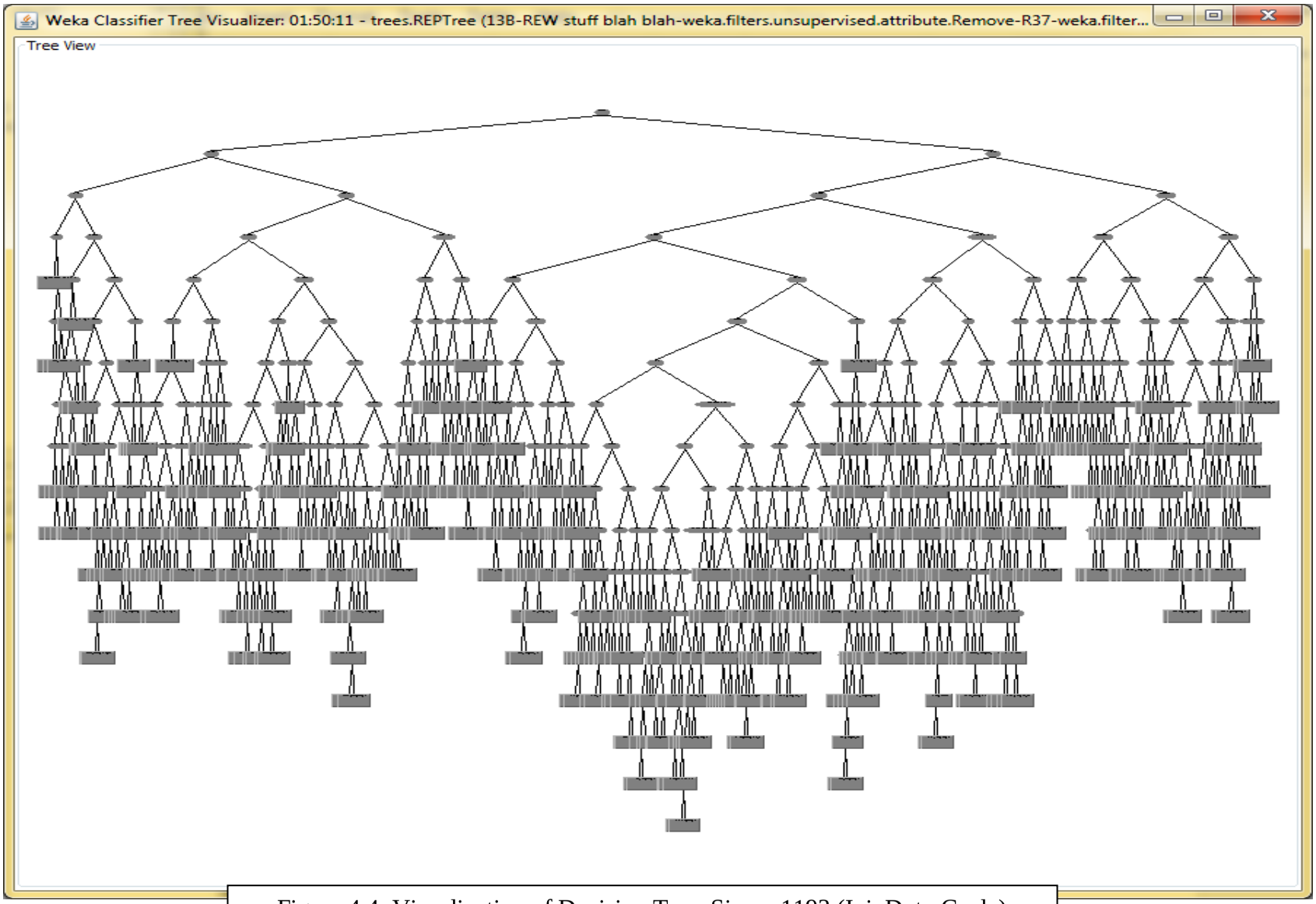


Figure 4.4: Visualization of Decision Tree, Size = 1193 (Inj. Duty Cycle)

Correlation coefficient	0.9898
Mean absolute error	0.9831
Root mean squared error	1.5993
Relative absolute error	10.92 %
Root relative squared error	14.27 %
Total Number of Instances	23477
Size	61
Training Time	0.21 seconds

Table 4.5 REPTree, Max Depth = 5 (Inj. Duty Cycle)

Table 4.5 demonstrates the performance of the REPTree when the depth of the tree is limited to only 5. While this increases the error rate it makes it much easier for the user to interpret the results (Figure 4.5).

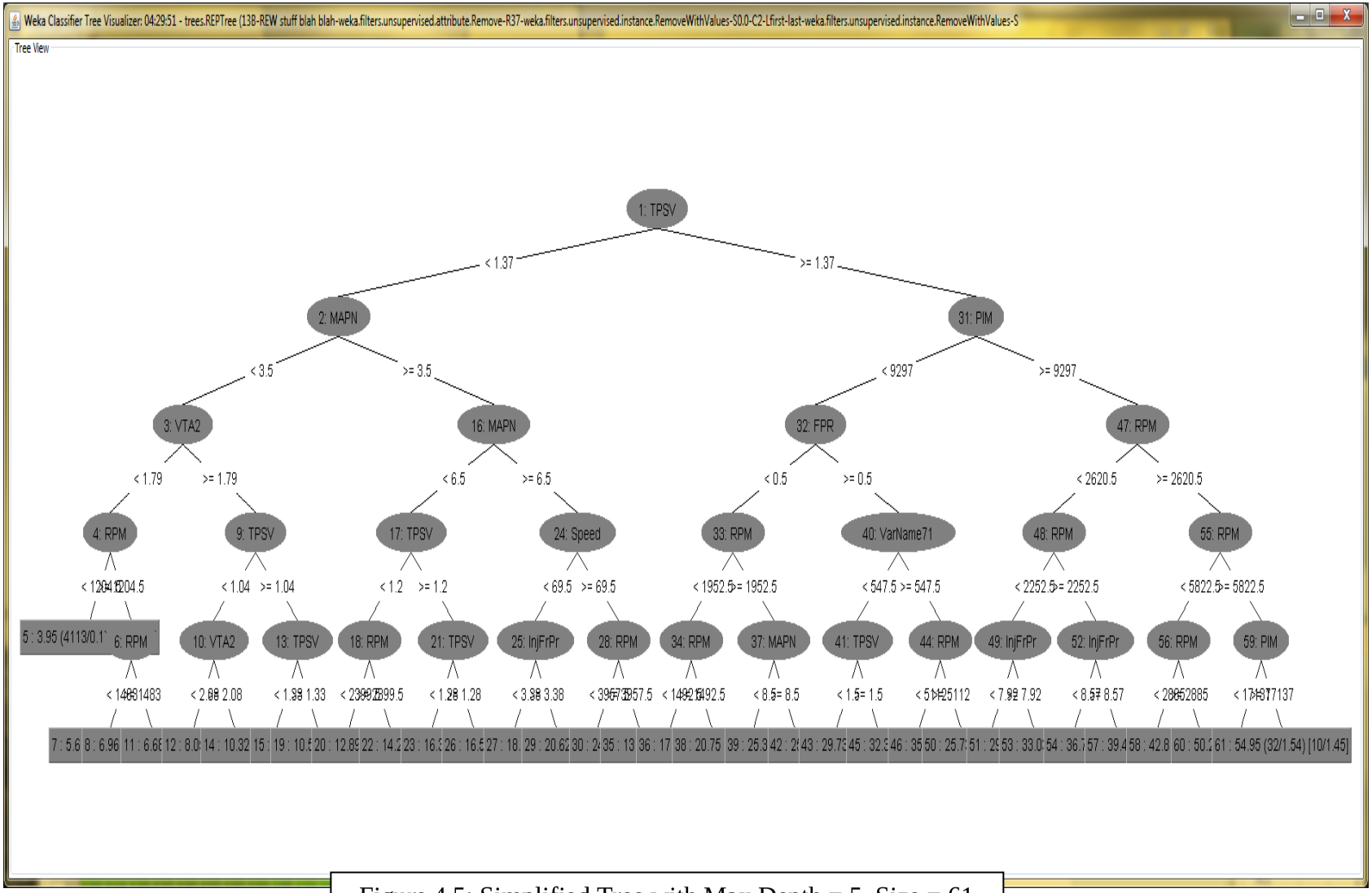


Figure 4.5: Simplified Tree with Max Depth = 5, Size = 61

Again bagging was able to improve the performance of the REPTree:

Correlation coefficient	0.9989
Mean absolute error	0.2653
Root mean squared error	0.5161
Relative absolute error	2.95 %
Root relative squared error	4.61 %
Total Number of Instances	23477
Size	1217
Training Time	2.86 seconds

Table 4.6 Bagged REPTree (Inj. Duty Cycle)

Table 4.6 shows that Bagging was able to reduce the error rate over the unbagged REPTree with regard to every error measure at the cost of an increase in model training time, 2.86 seconds versus 0.31 seconds (Figure 4.8). Comparing the visualization of errors, both bagged and unbagged REPTrees, they have very similar shapes, with larger errors occurring in more or less the same locations (Figure 43, Figure 4.6).

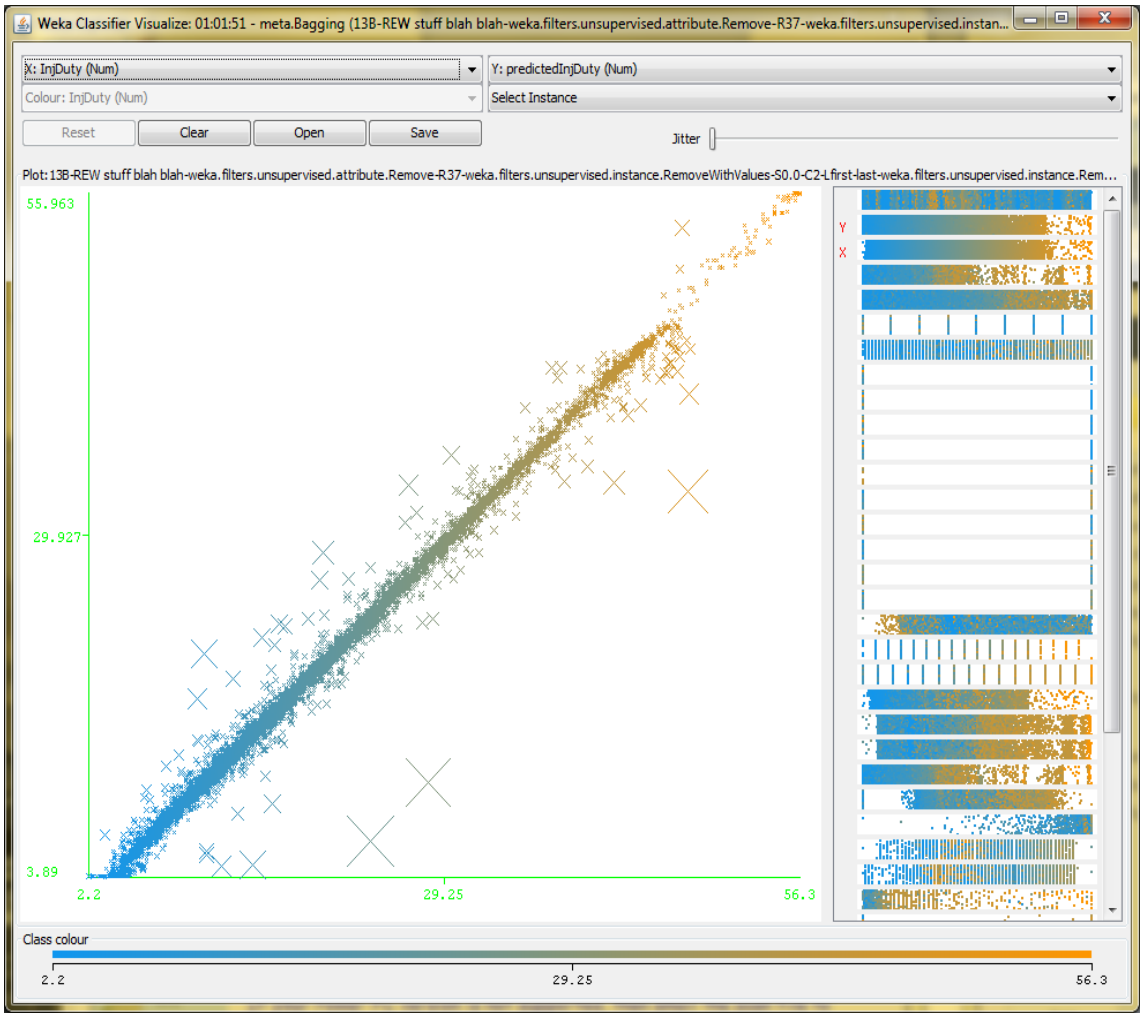


Figure 4.6: Decision Tree Prediction Errors (Inj. Duty Cycle).

4.2.2 Neural Networks

In this instance the multilayer perceptron neural network performed nearly as well as the REPTree algorithm, and shows a marked improvement over the first experiment, as shown in Table 4.7.

Correlation coefficient	0.9993
Mean absolute error	0.2712
Root mean squared error	0.4514
Relative absolute error	3.0123 %
Root relative squared error	4.0292 %
Total Number of Instances	23477
Training Time	104 seconds

Table 4.7 Neural Network (Inj. Duty Cycle)

Another interesting feature is that the magnitude of the prediction errors seems to be smaller than the errors produced by the decision tree (Figure 4.7 versus Figure 4.6). However, both suffer from possible catastrophic prediction errors towards the right of the graph below the linear cluster.

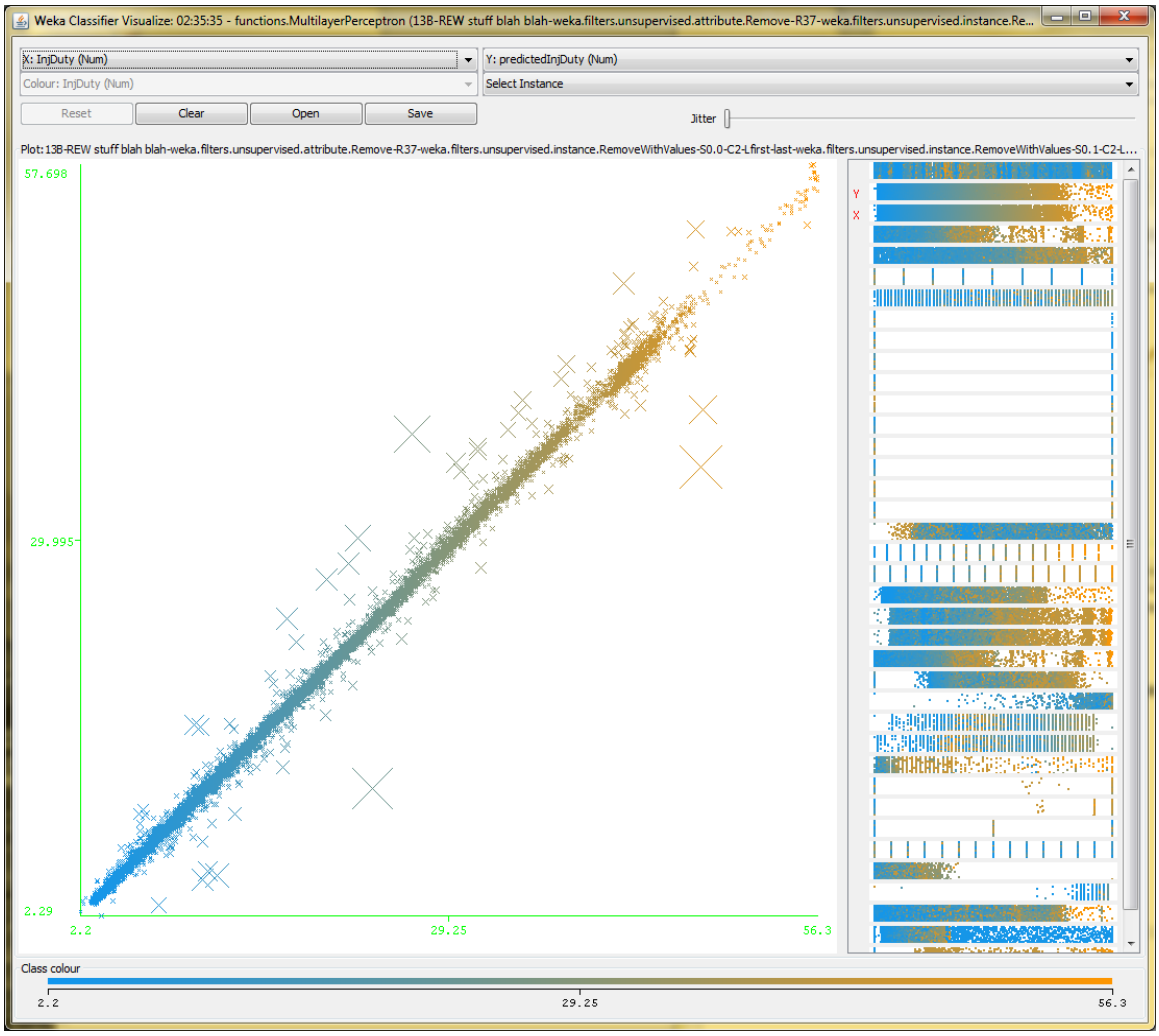


Figure 4.7: Neural Network Prediction Errors (Inj. Duty Cycle)

4.3 Discussion

The poor performance of the AFR prediction models could be explained by a combination of factors. One issue is there is a false ceiling of the range of values. Innovate Motorsports, the producers of the lambda sensor controller, recommend setting the maximum value to the lowest expected AFR that the engine will run on. However, when no fuel is being injected, the engine acts as an air pump, having an effectively infinite AFR. So, instead of an AFR of 16, this value should probably be closer to 20 or even higher. An attempt was made during preprocessing to mitigate this effect, but it may still be present. Also, the non-monotonic

surface described by the Mazda target AFR, and more specifically the greatly irregular surface shown in the percent error plot (Figure 3.9) could contribute to the difficulty in predicting AFR using only the datalog. Finally, the model is attempting to predict the AFR of the current base map, which is not necessarily calibrated with the same target AFRs as the Mazda target map, and may in fact not be associated with a proper starting AFR goal.

The success of predicting the injector duty cycle can likewise be described by inspecting the surface described by the values contained in the injector basemap (Figure 3.10). Since this surface is relatively smooth, it should be much easier to perform regression modeling using data collected from the data logs. Also, injector duty cycle is a calculated parameter that depends on the other sensors that are being logged, and is directly related to their values. Whereas AFR is certainly related to injector duty cycle load and engine speed but does not have the same monotonic relationship even for well-calibrated engines (Figure 3.8).

An additional advantage that decision trees have over neural networks is that the output is easily understood by end users. In both the full tree and reduced height tree, it can be easily seen which attributes are most important. They both have throttle position sensor (TPSV) as the first split point followed by either RPM or a representation of the load of the engine. This enables the end user or engine calibrator to verify the model's output using their own intuition rather than pouring through the resulting predicted base map looking for errors (Figure 4.8).

	Predicting AFR			Predicting Inj. Duty Cycle		
Results	Decision Tree	Decision Tree (Bagging)	Neural Network	Decision Tree	Decision Tree (Bagging)	Neural Network
Correlation Coefficient	0.8719	0.9072	0.802	0.9981	0.9989	0.9993
Mean Absolute Error	0.3694	0.3252	0.5665	0.3787	0.2653	0.2712
Root Mean Squared Error	0.6719	0.54	0.828	0.6939	0.5161	0.4514
Relative Absolute Error	33.8104 %	31.3236 %	51.8462 %	4.2065 %	2.9467 %	3.0123 %
Root Relative Squared Error	49.1539 %	42.2046 %	60.5757 %	6.1939 %	4.6071 %	4.0292 %
Total Number of Instances	23477	23477	23477	23477	23477	23477
Size	2017	2419	--	1193	1217	--
Training Time	0.8 seconds	3.92 seconds	186 seconds	0.31 seconds	2.86 seconds	104 seconds

Table 4.8: Results Summary

Chapter 5 Conclusion

5.1 Contributions

The main contributions of this thesis are the affirmation of the theory that machine learning can be used to expedite the engine calibration process. While the process does not produce perfect results, the systems predictions can be used to greatly expedite the calibration process, when compared with the usual practice of tuning one load point and engine speed at a time. Also this process enables the end users to drive their vehicle normally, without the express intent of calibration, and allows them to focus on the act of driving and roadway traffic. With some improvement in both the predictions and safeguards, it also reduces the dependence on a chassis dynamometer and a calibration specialist, either entirely, or by reducing the amount of time to arrive at the desired calibration, which can be significantly costly to the end users.

5.2 Limitations

It should be noted that this system depends entirely on the sensors that are currently equipped on the vehicle. If there is a problem with the calibration of any of the sensors or if the sensors malfunction and produce incorrect output, it could result in dangerously incorrect predictions. There is no validation of sensor output or sanity checking of the input data. It is entirely up to the user to ensure that the model's output is reasonable.

Since the model is not perfect, there will occasionally be errors that are a great deal different from the required value. In instances where the error predicts that more fuel be added than necessary, this is less of an issue. However, when it predicts a value that would result in a leaner mixture than desired, if the error is not identified, possible engine damage could result. This shortcoming could be overcome by some basic "smoothing" of the predicted fuel map and some safeguards to ensure the user is made aware of possible issues with predicted results.

5.3 Future work

While the proposed approach is currently only implemented for prediction and calibration of steady state air-fuel ratio, it should be possible to extend the system to make predictions that would improve transient response. Similarly, the addition of a few more sensors, (such as quick reaction *Exhaust Gas Temperature* probes (EGT), or *combustion chamber pressure sensors*), might enable a system similar to this one to be used to optimize the ignition timing of an engine.

Also more work could be done to simplify and automate the workflow process, from data collection to model prediction. Right now the user must first collect the log data in the vehicle and later create a model and generate predictions from the data.

Currently, this system has only been tested on a single engine. While it should be able to generalize to other ICEs, more testing is required. And with more testing and data of other ICEs, the opportunities for non-parametric or aggregate learning arise.

References

[1] "*AAIA*." Automotive Aftermarket Industry Association. Automotive Aftermarket Industry Association, Accessed October 2013. <http://www.aftermarket.org/AbouttheAftermarket>.

[2] M. Warner, *Street Turbocharging: Design, Fabrication, Installation and Tuning of High Performance Turbocharger Systems*. New York: HP, 2006.

[3] P. K. Wong, "*Compensation Map Calibration of Engine Management Systems Using Least-Squares Support Vector Committee Machine and Evolutionary Optimization*." Vehicle Engineering (VE) 1.1 (2013): Vehicle Engineering.

[4] Y. J. Zhai and D. L. Yu. 2007. *A Neural Network Model Based MPC of Engine AFR with Single-Dimensional Optimization*. In Proceedings of the 4th international symposium on Neural Networks: Advances in Neural Networks (ISNN '07), Springer-Verlag, Berlin, Heidelberg, 339-348.

[5] J. B. Heywood, *Internal Combustion Engine Fundamentals*. New York: McGraw-Hill, 1988.

[6] K. Yamamoto. *Rotary Engine*. Vol. 6. Tokyo, Japan: Society of Automotive Engineers, 1981. Automotive Engineering.

[7] Karl-Heinz Dietsche, ed. *Automotive Handbook*. 8th ed. Plochingen, Germany: Robert Bosch, 2011.

[8] B. Bowling and A. Grippo. "Tuning Your MegaSquirt." *Tuning Your MegaSquirt*. MegaSquirt, Accessed November 2013. <http://74.124.198.224/~lance/v22manual/mtune.htm>.

[9] C. Westbrook, and R. Herchenroder. "*Datalogit & P-FC*

Preparation/Tuning/Usage Notes." Online Enthusiast Community. PFC-DL-Tuning. Yahoo Groups, Accessed June 2012. https://groups.yahoo.com/neo/groups/PFC-DL-Tuning/files

[10] B. Davies. "*Single Turbo RX-7s and the Power FC*." Online posting. Mazda Rotary Club, Accessed June. 2012. <http://mazdarotaryclub.com/general/apeximanuals/RX7Tuning.pdf>.

[11] *"ETAS - ETAS ASCMO."* - Software Products & Systems. ETAS, Accessed June 2013. <http://www.etas.com/en/products/ascmo.php>.

[12] "*Model-Based Calibration Toolbox*." Engine Calibration--MATLAB. MathWorks, Accessed Nov. 2013. <http://www.mathworks.com/products/mbc/>.

[13] H. Uzun.. " *Calibration of EnginePerformance at Mercedes-AMG*. MathWorks / Mercedes-AMG GmbH, 2010. Web. 3 Nov. 2013. <http://www.mathworks.com/company/newsletters/articles/calibration-of-engine-performance-at-mercedes-amg.html>.

[14] P. Simon. *Too Big to Ignore: The Business Case for Big Data*. 1st ed. Wiley, 2013. Wiley and SAS Business Ser. p. 89

[15] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten (2009); *The WEKA Data Mining Software: An Update;* SIGKDD Explorations, Volume 11, Issue 1

[16] I. H. Witten, E. Frank, and M. A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Burlington, MA: Morgan Kaufmann, 2011.

[17] S. Lawrence, L. C. Giles, A. C. Tsoi "*What Size Neural Network Gives Optimal*

Generalization? Convergence Properties of Backpropagation" No. UMIACS-TR-96-22 and CS-TR-3617.April 1996.

[18] S.W. Wang, D.L. Yu, J.B. Gomm, G.F. Page, S.S. Douglas, "*Adaptive neural network model based predictive control for air–fuel ratio of SI engines"*, Engineering Applications of Artificial Intelligence, Volume 19, Issue 2, March 2006, Pages 189-200

[19] C. M. Vong, P. K. Wong, Y. P. Li, "*Prediction of automotive engine power and torque using least squares support vector machines and Bayesian inference"*, Engineering Applications of Artificial Intelligence, Volume 19, Issue 3, April 2006, Pages 277-287, ISSN 0952-1976

[20] H. C. Wong, P. K. Wong, and C. M. Vong, "*Model Predictive Engine Air-Ratio Control Using Online Sequential Relevance Vector Machine*," Journal of Control Science and Engineering, vol. 2012, Article ID 731825, 2012.

[21] *FC-Datalogit Mazda / Efini RX7 FD3S User Manual*. N.p.: Http://www.fc-datalogit.co.nz, 30 Sept. 2002.

[22] *MATLAB*. Computer software. Vers. 2013a. MathWorks, n.d. Web.

[23] "*Technical Articles | Forced Induction 101 | Turbosmart USA*." Turbosmart USA. Accessed November 2013.

[24] "*Fuel injection.*" Wikipedia. Wikimedia Foundation, Inc. Accessed November 2013. <http://en.wikipedia.org/wiki/Fuel_injection>.

[25] "*Air-Fuel Ratio*." Wikipedia. Wikimedia Foundation, Inc. Accessed November 2013. <http://en.wikipedia.org/wiki/Air-fuel_ratio>.

[26] "*Broadband Lambda Sensor*." NGK -. Accessed October. 2013. <http://www.ngk.de/en/products-technologies/lambda-sensors/lambda-sensor-

technologies/broadband-lambda-sensor/>.

[27] "*Decision tree learning*." Wikipedia, Wikimedia Foundation, Inc. Accessed November 2013. <http://en.wikipedia.org/wiki/Decision_tree_learning>.

[28] "*Artificial neural network*." Wikipedia, Wikimedia Foundation, Inc. Accessed November 2013. <http://en.wikipedia.org/wiki/Artificial_neural_network>.

[29] "*Anscombe's quartet*." Wikipedia, Wikimedia Foundation, Inc. Accessed November 2013. <http://en.wikipedia.org/wiki/Anscombe's_quartet>.

[30] *"Power FC, 1992-1995 Mazda RX-7 (Ver. I-III)."* Power FC, 1992-1995 Mazda RX-7 (Ver. I-III). Accessed Feb. 2011. <http://www.apexi-usa.com/store/electronics/power-fc/power-fc-1992-1995-mazda-rx-7-ver-i-iii.html>.

[31] "*LC-1 Lambda Cable with 02 Sensor : Wideband Controller Cable for Dyno, ECU, Data Acquisition, or Gauge Applications.*" LC-1 Lambda Cable with 02 Sensor : Wideband Controller Cable for Dyno, ECU, Data Acquisition, or Gauge Applications. Accessed March 2011. <http://www.innovatemotorsports.com/products/lc1.php>.