Online Adaptive Fuzzy Neural Network Automotive Engine Control

By

Keith James

A Doctorial Thesis

Submitted in partial fulfilment of the requirements for the award of

Degree of Doctor of Philosophy of Loughborough University

Acknowledgments

A PhD thesis is often a task that requires the effort and support of many people to reach completion, this PhD has been of no exception. For this reason I would like to acknowledge the people that have supported me, guided me and generally made my life as a research student as pleasant and as smooth sailing as possible.

I would like to express my gratitude to my supervisor Rui Chen, for firstly, seeing potential in me and offering the opportunity to do the PhD and secondly, for his guidance. I would also like to thank the powertrain technician staff: Graham Smith, Adrian Broster and Steve Horner for their time, knowledge and assistance over the years.

 Outside of the University, I have been fortunate to be surrounded by friends and family that could offer invaluable emotional support. I would like to give thanks to my partner, Jenny Simmonds, for her love, support and patience. My parents, Phillip, Linda and my brother Andrew for providing me with a loving refuge I could always turn to in times of general confusion or dismay. I would also like to give special thanks to my parents, for their unwavering support throughout my education and allowing me the opportunities that have got me where I am today.

I would also like to thank my Father, Phillip, for giving up hours of their bed time reading to help me bring this PhD to completion.

Finally, I would like to thank my close friend and housemate during my study, Laurent Lebrun, for his support, friendship and laughter and generally to all my friends and family for making my life as a research student as fun and memorable as possible.

Abstract

Automotive manufacturers are investing in research and development for hybridization and more modern advanced combustion strategies. These new powertrain systems can offer the higher efficiency required to meet future emission legislation, but come at the cost of significantly increased complexity. The addition of new systems to modernise an engine increases the degrees of freedom of the control problem and the number of control variables. Advanced combustion strategies also display interlinked behaviour between control variables. This type of behaviour requires a more orchestrated multi-input multi-output control approach. Model based control is a common solution, but accurate control models can be difficult to achieve and calibrate due to the nonlinear dynamics of the engines. The modelling problem becomes worse when some advanced combustion systems display nonlinear dynamics that can change with time. Any fixed model control system would suffer from increasing model/system mismatch. Direct feedback would help reduce a degree or error from model/system mismatch, but feedback methods are often limited by cost and are generally indirect and slow response.

This research addresses these problems with the development of a mobile ionisation sensor and an online adaptive control architecture for multi-input multi-output engine control. The mobile ionisation system offers a cheap, fast response, direct in-cylinder feedback for combustion control. Feedback from 30 averaged cycles can be related to combustion timing with variance as small as 0.275 crank angle degrees. The control architecture combines neural networks and fuzzy logic for the control and reduced modelling effort for complex nonlinear systems. The combined control architecture allows continuous online control adaption for calibration against model/plant mismatch and time varying dynamics. In simulation, set point tracking could be maintained for combustion timing to 4 CAD and AFR to 4, for significant dynamics shifts in plant dynamics during a transient drive cycle.

Contents

List of Figures

List of Tables

Abbreviations

ADALINE	Adaptive linear neuron
AFR	Air fuel ratio
AMPC	Adaptive model predictive control
ANNs	Artificial neural networks
ATDC	After top dead centre
BDC	Bottom dead centre
BMEP	Brake mean effective pressure
CA	Crank angles
CAD	Crank angle degrees
CAI	Control auto ignition
CI	Compression ignition
CMAC	Cerebellar model articulation controller
CO	Carbon monoxide
COA	Centroid of area
CR	Compression ratio
CTVS	Charge trapping valve system
DISI	Direct injection spark ignition
DME	Dimethyl ether
DOE	Design of experiments
ECU	Electronic control unit
EGR	Exhaust gas residual

EPA	Environment protection agency
FFTD	Feed-forward time delay
FIR	Finite impulse response
FIS	Fuzzy inference system
FLC	Fuzzy logic controllers
FNN	Fuzzy neural networks
GDI	Gasoline direct injection
GHG	Greenhouse gas
GPC	Generalized predictive control
HCCI	Homogenous charge compression ignition
HT	High tension
IC	Internal combustion
IMEP	Indicated mean effective pressure
IVC	Intake valve closing
LES	Lotus Engine Simulation
LOLIMO	Local linear model tree
LPV	Linear parameter varying
LQG	Linear quadratic gaussian
MF	Membership functions
MFB	Mass fraction burn
MIMO	Multi-input multi-output
MISO	Multi-input single output
MLP	Multi-layer perceptron

MPC	Model predictive control
MPCI	Model predictive control and identification
NMPC	Nonlinear model predictive controller
NOx	Nitrous oxides
NRTC	Non road transient cycle
NVH	Noise vibration and harshness
PE	Persistent excitation
PFI	Port fuel injection
PID	Proportional integral and derivative
RMS	Root mean square
RNNs	Recurrent neural networks
SI	Spark ignition
SISO	Single input single output
SLFNN	Self learning fuzzy neural network
TDC	Top dead centre
TRG	Trapped residual gas
TWC	Three way catalysts
UEGO	Universal exhaust gas oxygen
UHC	Unburned hydrocarbons
VCR	Variable compression ratio
VI	Virtual Instrument
VVA	Variable valve actuation
VVT	Variable valve timing

1. INTRODUCTION

Climate change is a global problem. As development of countries rapidly increase, so does the use of fossil fuels to run their transport and industry. It is clear that human activities are adding to the greenhouse effects and the pollution problems across the world.

As the demand for energy increases more fossil fuels are consumed. As fossil fuels are a non-renewable resource, it is clear they cannot last forever. This, combined with the global climate problem, drives the need to find more efficient ways to extract energy from these high-density sources, and cleaner methods to reduce the damaging effects on the environment.

Transport is considered to contribute approximately 23 percent of the annual Green House Gas (GHG) emission worldwide [146]. Road transport cars, buses and trucks contribute the majority of the responsible GHG emissions. Vehicle emissions have many impacts on health ranging across mortality, exacerbation of asthma, chronic bronchitis, respiratory tract infections, ischaemic heart disease and stroke. The pollution and health problems are a major concern for governments over the world, and it is recognised that cleaner combustion is essential for the road transport industry. This is reflected in the current vehicle emissions legislation (1).

Vehicle manufacturers need to keep up with emission legislations if they are to continue to sell vehicles within the countries that enforce them. The vehicle manufacturers attempt to conform to these emission requirements, through continuous research and development. The R & D targets cleaner combustion technologies, reduction devices for emissions and alternative power sources, such as hybrid and full electric vehicles.

1.1 THE INTERNAL COMBUSTION ENGINE COMBUSTION MODES

The Internal Combustion (IC) engine was first designed by a Dutch physicist Christian Huygens, in 1680; it was to be fuelled using gunpowder, however the design was never built [147]. It was later in 1807 that Francois Isaac de Rivaz of Switzerland invented an IC engine that ran on a mixture of oxygen and hydrogen and was combined with a vehicle. The design was not very successful but it was the birth of the automobile. A vast amount of development has occurred on the IC engine to give the highly advanced engines that are in production today. The designs that have found major success are the "spark ignition" engine and the "compression ignition" engine.

(removed for copywrite)

Figure 1.1 A standard Piston attached to a crank (2)

Figure [1.1] is a diagram of a standard piston attached to a crank in a reciprocating engine. During the rotation of the crank, the piston moves between Top Dead Centre (TDC), and Bottom Dead Centre (BDC) and back. This motion is referred to as the stroke of the engine. The engine can be configured to run in a number of different cycles, one of the most common being the four stroke cycle. In a four stroke cycle, there are two complete revolutions of the crank and four strokes of the piston; "intake", "compression", "expansion" and "exhaust".

During the intake stroke the piston moves down the cylinder creating a pressure difference between the cylinder environment and the external environment. When the intake valve is open, it allows air, Direct Injection (DI) engine, or an air/fuel mixture, Port Fuel Injection (PFI) engine, to be drawn into the cylinder. If the engine has a throttle plate, or any form of restriction in the intake system design, the engine will need to work against this to charge the

cylinder. Any negative work caused through the intake system is referred to as pumping losses. Once the cylinder has a fresh inducted charge, it will compress the mixture through the motion of the piston from BDC to TDC, the compression stroke. Generally towards the end of the compression stroke the cylinder charge is ignited. As the piston approaches TDC the charge will have been compressed from the full cylinder volume to the small clearance volume. The ratio between these two volumes is known as the compression ratio (CR). At the clearance volume the charge is at a higher pressure and temperature creating an environment to encourage complete combustion of the cylinder charge. As the charge combusts, the pressure and temperature raise rapidly causing force to be exerted onto the piston. The force causes the piston to continue its motion between TDC to BDC, the expansion stroke. During the expansion stroke, the force exerted onto the piston is converted into positive work out. The final stroke of the cycle is the exhaust stroke. As the piston moves back toward TDC, the exhaust valve is opened allowing the combustion products to be expelled from the cylinder before the process begins again.

1.1.1 SPARK IGNITION

Port injected Spark Ignition (SI) engines induct a homogeneous mixture of air and fuel. A spark plug is used to ignite this mixture before the end of the compression stroke. Once ignition has occurred the flame propagates throughout the cylinder to complete the combustion. For flame propagation to occur effectively, a stoichiometric or near stoichiometric mixture must be used. A stoichiometric mixture offers the exact proportions of air and fuel to allow complete combustion without excess air. An added benefit of running with a stoichiometric mixture is that it allows operation with a three-way catalyst (TWC) as an exhaust after-treatment method.

(removed for copywrite)

Figure 1.2 Ideal OTTO cycle

To understand the limitations of the conventional SI engine it is useful to introduce the OTTO cycle, figure [1.2]. The OTTO cycle is an ideal air standard cycle that represents the four strokes of the combustion cycle. In the ideal OTTO cycle the compression and expansions are assumed reversible isentropic processes and energy transfer occurs during constant volume. The intake and exhaust is assumed to occur during isobaric compression and expansion. The efficiency of the cycle is determined by how much input energy can be converted into work out during the expansion stroke. The limitations for the conventional SI engine come from two main sources, throttling the input and flame propagation speed. Throttling is required in order to control the load, this causes pumping losses during the intake stroke. The intake stroke therefore departs from the ideal OTTO isobaric expansion and occurs over negative pressure. The greater the throttling the larger the negative pressure and work required by the engine to draw the cylinder charge. Increasing pumping losses also decreases the trapping efficiency causing a reduction in the effective CR. Reduced CR will decrease thermodynamic efficiency. The second drawback comes from flame propagation limiting the

speed of combustion. To achieve efficiency of the OTTO cycle heat addition would need to occur at constant volume to allow maximum work out to be achieved during the expansion stroke. In premixed combustion, the speed of the flame propagation after the compression stroke is determined by the mixtures laminar and turbulent flame speed. Laminar flame speed is determined by the mixtures properties, pressure and temperature. Turbulent flame speed is determined by unstable mixing between the fuel and hot products allowing entrainment and hence increase mixing/ignition of fresh mixture. Premixed combustion flame speeds are greatly enhanced when a turbulent flame front is present and can be approximately 30-40 m/s. However, as the combustion takes place as the piston begins to travel towards BDC, the combustion efficiency is deteriorated. This is due to the change in volume and the reduced effective expansion ratio, allowing less energy to be extracted as work output. In order to follow the ideal Otto cycle more closely, higher flame speeds are needed. The use of higher CR would promote higher flame speeds as temperature and pressure would be increased at TDC. However, CR range is limited by the occurrence of knock. Knock describes the phenomenon of the mixture outside of the propagating "flame ball" auto igniting due to the higher pressures and temperatures. When auto ignition occurs flame propagation reaches supersonic speeds (order of 2000 m/sec) and instead of achieving deflagration, detonation occurs.

When both of these limitations are present, i.e. an engine design with CRs that will allow operation over the entire speed and load range and high amounts of throttling, part-load efficiency drops considerably. This implies that a PFI SI engine at idling exhibits very low flame propagation speeds and very high pumping losses.

1.1.2 COMPRESSION IGNITION

Compression Ignition (CI) differs from SI as the fuel is directly injected during the compression stroke leading to a heterogeneous combustion. The load for a CI engine is altered through the amount of fuel that is injected, and the fuel ignites because of the pressure, temperature and mixing conditions in the cylinder. Diesel engines operate under higher CRs from 12:1 and 24:1 depending on whether they are naturally aspirated or charged. The higher CRs used achieve higher power output.

When the fuel is injected during the compression stroke it is generally staged just before TDC, approximately 20 degrees. As injection occurs, the liquid fuel atomizes as it passes through the injector nozzle and evaporates due to the temperature and pressure of the gas under compression. After a delay period the fuel air/mix reach an activation energy that allows auto ignition to occur. Ignition can take place at several locations simultaneously and the flame will propagate through the cylinder increasing the pressure and temperature giving further mixing to the un-burnt charge in the first part of the expansion stroke. At full load the mass of the fuel injected into the cylinder can be expected to be around 5% of the mass of air (2).

CI engines can use very high CRs and operate un-throttled throughout the operating range. This means that very high part-load efficiencies are achievable. This is possible due to the CI engine operating principle, the inducted air is compressed to the point at which the directly injected fuel can ignite.

The CI engine can offer combustion with high efficiency potential. However, due to the principle of operation there are some inherent disadvantages in the form of emissions. Given that the fuel is ignited as it is injected inside the combustion chamber, it is consumed through diffusion flame. This means that on the outside boundaries of the fuel jet, very lean combustion occurs, and the jet core very rich combustion occurs. Although inadequate mixing can be reduced using swirl imposed on the intake air-steams, the results can't match the emission levels of SI engines. As a general comment, it can be said that Nitrogen Oxides (NOx)

and smoke are high in diesels, but Unburned Hydrocarbons (UHC) and Carbon Monoxide (CO) tend to be lower. In addition, due to the necessarily low air utilization, stoichiometric mixtures would result in very high particulate emissions (smoke limit). Therefore, specific power outputs of normally aspirated diesels are low. Thermodynamic efficiency is limited due to the slow heat release during expansion.

However, CI engines can be complemented with turbochargers to further increase their thermodynamic efficiency. Due to the knock free operation of diffusion flame combustion turbocharging can be used with CI engines without reducing the CR, which is required with charged SI engines. When turbocharging is combined with common rail injection systems, the resulting power density can be on a par with SI engines.

1.2 THE FUTURE OF THE INTERNAL COMBUSTION ENGINE

The IC engine is a very valuable energy conversion system, and is used to support energy requirements in almost every industry. The popularity of the IC engine stem from some key attributes:

- The IC engine converts high-energy density fuels into usable power, allowing for long operating ranges.

- As the combustion is internal and intermittent, the combustion chambers run cool compared to the combustion temperature. This allows cheaper materials to be used for the construction of the combustion chamber.

- The IC engines have a high power output to mass ratio.

- High durability of the engine design and materials allow long periods between service intervals.

- IC engines are also cost effective, due to well-developed manufacturing techniques and use of cheap and recyclable materials.

- The IC engine can operate on a range of fuels.

To understand the future of the IC engine it is important to understand its advantages and disadvantages and where new technologies can be used to develop and compliment the IC engine or where it may need replacing. With increasing emission legislation and diminishing fossil fuel supplies the IC engine needs to develop for the future if it is to survive.

1.2.1 ELECTRIC VEHICLES, HYBRIDISATION AND WELL-TO-WHEEL COMPARISON

With the demand for reduced emissions combined with technology advancement, alternative power sources have begun to be considered and implemented into production vehicles.

Alternative power sources have allowed the development of electric vehicles, with automotive companies now offering hybrid and full electric in their vehicle ranges. The electric vehicles currently offer advantages over the IC engine counterparts as they can be charged from the household and have zero GHG emissions emitted from the vehicle. With governments giving road Tax benefits to low emission vehicles [145], the electric or hybrid vehicle will become very popular. However, they do have downsides; the major of these is with the energy storage methods.

Although electric motors have much higher conversion efficiency then the IC engine their overall vehicle operating range will be less [146]. Batteries offer a much lower energy density (0.6 MJ/Kg) than petrol/diesel (46 MJ/Kg). Therefore, the operating range is significantly reduced. The batteries also bring other complications. Batteries require more sophisticated control systems for operation and charging. They also have a limited life cycle before replacement is necessary.

Electric vehicles may appear at first to be the answer to efficiency and emission problems, but this is not exactly true, the energy has to be generated somewhere. This energy is generated in power stations and distributed to where the vehicles are charged and there are a number of inefficiencies associated with this process. To give an accurate comparison of current IC engine efficiency and GHG emissions with hybrid and electric vehicles a comparison of energy well-to-wheel is more appropriate.

The well-to-wheel compares the efficiency and GHG emission of all the major processes between the original energy sources and the energy driving the wheels. These processes can be grouped into two main areas, upstream and the vehicle. Upstream includes any losses or GHG production associated with the generation and distribution of electricity, the refining of crude oil for fuel production and fuel distribution. The vehicle includes the efficiency of the engine and powertrain system of target vehicles.

J. Louis produced a report comparing a number of vehicles using the well-to-wheel analysis provided by a commercially available mathematical model called GREET (3).

(removed for copywrite)

Figure 1.3 well-to-wheel comparison for energy consumption for IC, hybrid and full electric reference vehicles(3).

Figure [1.3] shows the well-to-wheel energy consumption in MJ/Km for each of the reference vehicles. The results are split onto two sections the upstream energy consumption and the vehicle energy consumption. The electric vehicle clearly has a far more efficient powertrain than the other vehicles but overall is significantly let down by the upstream efficiency. Electricity in the UK is generated from a mixture of raw materials or sources. The combination of Oil, Gas, Coal, Uranium and renewable energy sources gives an overall average energy conversion efficiency of 40.7% [3]. However, this is not the only efficiency that affects electricity production. Oil and gas also require production and distribution and where gas is stored it often requires local compression or liquefaction. There are also energy losses associated with the distribution of electricity through the national grid. All these upstream systems in total make the electric vehicle in this report the least efficient.

The hybrid IC engine/electric vehicles have an approximate 35% lower fuel consumption then conventional IC engine vehicles. This difference translates directly into the reduced well-to-wheel energy.

Figure [1.41] shows the well-to-wheel comparison for GHG emissions in g/Km for each of the reference vehicles. The gasoline IC engine has the most GHG emissions. This is due to its lower combustion efficiency and use of more refined fuels. The full electric vehicle has no vehicle emissions but overall is almost on par with conventional diesel and worse than the diesel IC engine/electric hybrid.

(removed for copywrite)

Figure 1.4 well-to-wheel comparison for the GHG emissions for IC, hybrid and full electric reference

vehicles (3)

From the well-to-wheel analysis it is clear the full electric vehicle is currently limited by the upstream efficiency of electricity generation and distribution, even if the vehicles could be improved to 100% efficiency, they would still be less efficient overall to the hybrid. The IC engines have much lower upstream efficiency and emission problems to the full electric vehicle but is let down by poor conversion efficiency. The IC engine is however continuously developing and this is an area that can be improved in the future. From the result it is clear there is currently a considerable advantage for improving efficiency and reducing GHG emission in hybrid powertrain systems.

The comparison shows the IC engine will remain a competitive choice for vehicle powertrain application in the future, either through hybrid systems or conventional IC engine powertrains. It is clear that the IC engine requires continuous development to improve the energy conversion efficiency. Any improvements will benefit IC and hybrid applications. From figure [1.4] it can be seen that if small improvements in efficiency could be made to reduce GHG emissions, it would allow hybrid gasoline and conventional diesel IC engine powetrains to have overall less emission then full electric vehicles.

Although the electric vehicle has become a major competitor to the IC engine for vehicle powertrains, there is still a strong future for the IC engine as long as development continues for improving emissions, efficiency and use with renewable fuels. The findings from this comparison coincide with the technology road map proposed in an independent report on the future of the automotive industry in the UK [147].

(removed for copywrite)

Figure 1.5 Automotive industry road map for development [147].

Figure [1.5] is a graphical representation for the predicted technological development in the UK automotive industry over forty five years. Development of the IC engine innovations will be of great importance in the future development of hybrid powertrain systems.

1.3 CURRENT AND FUTURE ADVANCED COMBUSTION MODES

The above description of the SI and CI engines has given a brief introduction into the basic operation of the combustion modes, detailing the differences between the two combustion strategies, their advantages and disadvantages. It is also now clear that development is required to decrease harmful emissions to meet new future legislations, as well as improve efficiency to maintain the IC engine as a competitive power source for vehicle powertrain applications. This requirement for development has spurred the evolution of the classic SI and CI engines to a new generation of internal combustion modes. In this work, they will be referred to as advanced combustion modes and are as follows.

1.3.1 CYLINDER DEACTIVATION

Cylinder deactivation allows for large capacity engines to run on fewer cylinders when high power and torque is not required. At part load operation, the SI engine's efficiency is low. By using fewer cylinders it is possible to shift the operating range of the remaining cylinders to a higher load where greater efficiency can be achieved. This allows for a significant reduction in emissions and improvements on efficiency for large multi-cylinder engines. To achieve an engine capable of cylinder deactivation more complex variable valve timing mechanisms are needed, as well as, control systems for selecting operating mode and fuel cut off. M Fujiwara

et al, developed a V6 engine with variable cylinder management (4). The system could operate between three, four and six cylinder operation modes. During start up and acceleration, or times of high load demand, the engine would operate on six cylinders. For moderate acceleration, mid-load demand such as mild hills, the engine would operate on four cylinders. Then for low load, only three cylinders would be used. The Honda VCM system can finely tailor the working displacement of the engine to match the driving requirements. The technology allowed for a 10% increase over a US Environment Protection Agency (EPA) highway mode cycle for fuel efficiency for a 3.5l V6 engine over the previous 3.0l V6 model (4).

The main difference between conventional SI engine control and cylinder deactivation SI engine control is the requirement for additional combustion technology such as Variable Valve Actuation (VVA) systems with hydraulic control mechanisms to achieve the cylinder deactivation (4). A higher level control that seamlessly phases the engine between operating modes without affecting drivability is also required.

Therefore the additional control requirements can be summarised;

- Control requirement for high level predictive control or logic to transition engine through operating modes depending on driver demand.

1.3.2 Downsized SI for high Brake Mean Effective Pressure

Downsizing is the use of smaller capacity engine operating at higher specific engine loads and therefore better operational efficiency points. Downsizing is one of the most proven paths to CO_2 emission reduction. Through careful design using the downsize strategy it is possible to replace a 2.4l engine with a 1.2l engine that has superior torque at all speeds and on-road fuel consumption benefits of 25-30% (5). Downsizing is an attractive path for vehicle

manufacturers as the base powertrain system requires the least modification in comparison to integrating hybrid or fuel cell solutions. Downsized SI engine often use a combination of key combustion technologies including; high pressure direct injection, high boost capability through twin/variable turbo charging, integrated intercooler system, dual variable valve timing, light weight valve systems, high energy coils and carefully optimised engine/head cooling design (5).

With the multiplication of the control actuators to achieve downsizing, more advanced engine control is , often necessary to obtain efficient control. Proportional Integral and Derivative (PID) controllers are often used in conventional air actuator control in automotive applications. PID is often difficult to tune, producing overshoot and can have bad set point tracking due to systems nonlinearities. Model based control methods have been described as the best alternative to enhance engine torque control (6).

The common additional control characteristics can be summarised;

- Descriptive models need to be dynamic and nonlinear. They often require a vast amount of work to be determined, particularly to fix the parameters specific to each engine type in mapping.

- In engine control, the sample period changes with engine speed, very short periods need to be considered in the control design.

- Control saturation can occur from large disturbances and or model/system mismatch.

- Many useful internal state feedback variables are not measured either due to the difficulties in justifying the cost of setting up additional sensors, or concerns with senor reliability.

- Control systems are becoming more multi-objective, in order to satisfy contradictory requirements such as performance, comfort, lower fuel consumption and emissions.

- The control problem is becoming more complex, yet the control solution is required to be implemented on on-board ECUs. The ECU computer power is increasing but still remains limited.

1.3.4 GASOLINE DIRECT INJECTION SI

The Gasoline Direct Injection (GDI) SI engine, also known as the Direct Injection, Spark Ignition (DISI) engine, is another advanced combustion strategy. The main differences between a GDI engine and a conventional port fuel injection (PFI) SI engine are where the air and fuel are mixed and when the mixing takes place. In GDI engines there are two distinct operating ranges. The fuel can either be directly injected into the cylinder during the intake stoke to promote a homogenous mixture, or late during the compression stroke to promote a stratified charge. Through switching between stratified and homogenous charge operating regions it is possible to trade off fuel economy, power output and emissions. In homogenous mode the characteristics are very similar to that of the PFI SI engine in terms of performance and emission, with Air Fuel Ratio (AFR) operating around stoichiometric. In the stratified operation, fuel and air are guided through special intake, cylinder head and piston designs to concentrate the charge around the spark plug. Typical operating AFR in stratified charge is 35:1, significantly higher the PFI engines.

Key technologies for GDI engines often include turbo charging, integrated intercooler, high pressure fuel and injector systems, optimization of pistons, intakes and valves for air flow and fuel atomisation and VVA.

The GDI engines gains are found from more accurate control over the fuel injection amount and timing. The engine design, in general, induce "tumble" through wall or piston guided air streams, achieving the presentation of an ignitable mixture around the spark plug, despite an overall excess of air in the cylinder. The GDI engines negate the need for homogeneity throughout the entire combustion chamber, by confining combustion in a smaller volume

near the spark plug. GDI engines increase their effective CR and reduce their pumping losses by being able to operate throttle-less at low engine loads, owing to stratified combustion. Stratified 3-cylinder turbocharged GDI systems have been reported to offer up to 22% CO2 reduction compared to baseline PFI engine and up to 18% in stoichiometric GDI operation with VVA and a conventional TWC (7). The combustion operating mode depends upon the driver torque demanded and the catalyst state. When torque demands are high the engine must be operated in homogenous mode. When the torque demands are low to moderate and the catalyst is operating efficiently, stratified operation should be used to increase fuel economy.

As both GDI and downsized SI for high Brake Mean Effective Pressure (BMEP) require a range of key combustion technologies to operate, many of the control problems highlighted with downsize SI, also apply to GDI. However, the GDI SI combustion also has its own additional control problems. There are two clearly different operational regions for GDI, each combustion mode requires very different optimal coordination of the numerous control variables to achieve the desired torque and AFR control. As the two combustion modes require different control strategies, there becomes an additional control challenge to create seamless transition between these combustion modes depending on demands.

In stratified lean burn GDI operation the combustion is lean and leads to an increase in NOx emissions. These emissions can be trapped in an aftertreatment NOx trap, but this requires regeneration through operation in stoichiometric mode. Currently there is an absence of on-line emissions measurement for the control of lean burn engine aftertreatment systems. Control is often achieved in a feed forward approach based on models of gas emissions and the aftertreatment subsystems. However, these models become inaccurate and reduce the performance of the controller with changes in ambient conditions, engine component age or immeasurable influences that affect catalyst performance (8).

Another challenge for lean burn GDI is accurate AFR control. As the universal exhaust gas oxygen (UEGO) sensor is located after the lean NOx trap, a significant time delay occurs between the UEGO sensor and the effective change of the engine feed gas's AFR. A UEGO sensor needs to be located after the NOx trap for feedback during the NOx trap purge process. The purge is initiated when the mass of NOx reaches a threshold and terminated when the UEGO sensor downstream detects a lean to rich switch in exhaust gas. The time delay found from the UEGO sensor is largely dependent on the engine operational conditions defined by the engine speed and the mass flow of air. Throughout the engine operational window the time delay change can be significant, 0.3-2.7 seconds (9). This large variation in time delay is the main obstacle in achieving accurate control performance. Delayed response can cause an increase in HC and CO exhaust emissions.

A key feature of the GDI system is the interactive and coupled nature of its subsystems. These characteristics dictate that a multivariable control strategy can be employed to coordinate all the actuators in an orchestrated fashion to achieve optimal control performance (10). The interactive nature and multi-mode combustion represents a significant change from the conventional control for PFI engines in terms of algorithmic control, strategy and architecture, design methodology and calibration process.

Therefore in addition to the control problems highlighted for downsize SI engines the GDI strategies also require:

- High level predictive or logic control to allow for seamless transition between different combustion strategy modes.
- Multi-input multi-output control for optimal coordination of all control variables.
- Adaptive modelling techniques or predictive control that can take account or reduce the effects of time varying dynamics for some sub-system control.

- More sophisticated feedback methods or control architectures to improve AFR set point control.

1.3.5 CONTROLLED AUTO IGNITION/HOMOGENOUS CHARGE COMPRESSION IGNITION

Control Auto Ignition (CAI) or also commonly known as Homogenous Charge Compression Ignition (HCCI), can be used in both the four stroke and two stroke reciprocating engines and can offer some advantages from SI and CI engines while eliminating a number of the disadvantages associated with the two current combustion mechanisms.

(removed for copywrite)

Figure 1.6 HCCI combustion (144)

The HCCI process combines a number of different parts of CI and SI cycles. The combustion process uses a homogenous Air/Fuel mixture like the SI engines and the mixture is compressed using a higher CR like CI engines. This HCCI combustion leads to auto ignition occurring simultaneously at several sites where local in-homogeneities create warm patches where the temperature can exceed the threshold of the auto ignition temperature for the mixture, figure [1.6]. These increase the temperature and pressure of the mixture and leads to a full-scale ignition. HCCI is a thermal and pressure related auto-ignition process and is highly controlled and dominated by the chemical kinetics of the charge mixture. The kinetic reactions depend on the concentrations of reactants and products in the cylinder as well as the conditions of pressure and temperature. This is supported by the fact there is little or no flame propagation during the HCCI combustion (11).

The HCCI combustion can offer efficiency as high as CI diesel engines while producing low levels of NO_x and particulate matter. This is achieved through throttle-less operation that reduces pumping losses, high CRs that increase combustion efficiency and the short combustion durations that are inherent to HCCI. The short combustion duration occurs as it does not rely on flame propagation to burn the air/fuel mix. As flame propagation is limited, dilution levels can be much higher than SI or CI engines. Dilution also keeps the combustion controllable by decreasing knock tendency and lowering combustion peak temperature. Low combustion temperature maintains the lower NO_x emissions (11). The NO_x emissions can typically be expected to be 90% lower than SI and up to 98% lower than GDI engines. The particulate emissions are also reduced due to the homogeneity of the mixture being combusted eliminating rich AFR regions (12).

Another advantage of HCCI is its flexibility of fuel that can be combusted. It is possible to operate HCCI with a large range of different fuels or blends of fuels including diesel, gasoline, hydrogen, dimethyl ether (DME), bio ethanol and natural gas to name a few (13-17). The HCCI combustion offers some significant advantages over current combustion strategies for the future of the IC engine, however in achieving these advantages it has its own set of unique problems and disadvantages.

As the homogenous mixture is compressed until it ignites, there is no direct control over the combustion timing, its rate, or the combustions duration. It is also difficult to know if or when the combustion occurs or the quality of the combustion until after it has occurred. As the HCCI combustion is similar to knock in SI engines, if the rate and timing is not controlled accurately there can be substantial damage to the engine. Combustion control has to be achieved through indirect methods that can influence the temperature and pressures that are reached at TDC, as well as level of dilution of the charge to limit the rate of the combustion.

Worldwide there have been a significant number of experimental studies in recent years regarding the physical control of the HCCI combustion process. The main interest from these is the development of methods that allow for stable HCCI combustion over a wide operating range of engine conditions. The indirect control methods can in general, be split into two possible directions, modification of the air/fuel mixture properties and modifying the engine operation and design parameters, table [1.1].

Table 1.1 HCCI indirect control methods (18).

Modification of Air/Fuel mixture properties	Modification of engine operation and control parameters
Intake Temperature	Compression Ratio
Air/Fuel Ratio	Coolent Temperature
Exhaust Gas Recirculation	Variable Valve Timing
Additives	Intake pressure
Fuel Modification and Injection strategies	Water injection
Variable Valve Timing	

Modifying the air/fuel mixture properties alters the reactivity of the intake charge by changing its thermo-chemical properties, thus influencing the HCCI combustion. Modifying the engine operation and control parameters allows control and stabilisation of the HCCI combustion through manipulating the time-temperature history of the engine charge for various engine operating conditions (18).

The HCCI combustion is highly affected by the temperature history of the engine. In multi-cylinder engines, it becomes even more important that cylinders are timed correctly but also have balanced rates of combustion. If one cylinder is not balanced the difference in heat transfer and exhaust temperatures can have significant effects on future cycle combustion quality and timing, and can also cause instability in other cylinders.

The HCCI operating range is also limited to mid-load range. At high engine loads the hydrocarbon emissions are similar to SI combustion. At low engine loads hydrocarbon emission become higher because temperatures are so low that fuel near the walls cannot burn completely. At very low loads, temperatures are too low to complete the $C+O_2$ to CO_2 reactions. The NO_x emissions as described earlier are very low except at high loads. To obtain the higher load, fuel rate has to be increased to produce more power and the charge becomes less diluted. It is also very difficult to start an engine in the HCCI combustion mode. When an engine is first started, the charge receives no heating from the intake manifolds or ports. There are also no trapped residual gasses to mix with the fresh charge as no previous cycles have been completed, making it very difficult to create the temperatures, pressures and Exhaust Gas Residual (EGR) dilution needed for the HCCI combustion (11).

Due to the operating range limitations and difficulty with cold start capabilities there is a real requirement for combustion mode transition control, allowing the engine to operate under SI or CI combustion during start up, idle and high load conditions.

 Finding the optimal control for HCCI combustion is difficult; there is a need for fast response whilst also requiring a stable system that will be robust against fluctuations and external conditions. From table [1.1] no single control method can offer all this, so a compromise and combination of methods are often used (19-23). From a control prospective, the HCCI control parameters can be generally grouped into two further sections, methods that exhibit a potential for dynamic fast response and methods that will serve better for stabilising and extending the range, table [1.2].

Table 1.2 HCCI indirect control parameters for fast response or stabilising and range extension.

Fast Response	Slower Response for Stabilising and Range Extension
Variable Compression Ratio, Variable Charging	Intake Heating
Variable Valve Timing/Trapping Valves	External EGR
Fuel Blending combined with Injection Strategies	Static Intake Charging
	Coolent Temperature Control

1.3.5.1 Fast Response

The fast response systems are generally formed by methods that have a direct controlling effect over the in-cylinder conditions. Variable Compression Ratio (VCR) alters both the in-cylinder temperature and pressure achieved at TDC offering a compounded effect on the start of ignition for HCCI (24, 25). When IC engines are developed specifically for VCR, fast response can be achieved (23). Variable Valve Timing (VVT) alters the quantity of hot EGR, offering thermal effects as well as dilution for quick cycle-to-cycle control (19, 26-28). Direct fuel injection offers fast response system for the direct control over equivalence ratio and fuel stratification (29, 30). When combined with fuel blending, combustion duration control can also be achieved to some extent(31).

1.3.5.2 Slower response for stabilising and range extension

The slower response systems are generally external systems that have associated inertia and therefore have slower time response to effect the in cylinder conditions. Externally heating the inlet air for example allows control over the temperature of the charge before compression and is used regularly for HCCI control under steady state conditions (19, 20, 24, 30, 32-36). Intake heating can stabilize combustion, reducing effects of fuel quality/composition and coolant temperature on engine control (34-36). However, the intake heating method is an external thermal system that has thermal inertia. The temperature of

the charge and intake system cannot change instantly and therefore the dynamic response is inherently slow (30, 34). Altering the coolant temperature also has inherently slow response due to the thermal inertia of the system (33). However, coolant temperature control can create natural thermal stratification, smoothing the heat release rate, extending combustion duration and lowering the tendency of knock at high load (33). Cool external EGR can also lower combustion temperature and reduce the oxygen concentration. This affects the rate of combustion and permits the use of lower air/fuel ratios, allowing an increase in the load limit (37, 38). Static intake charging can offer advantages in extending and stabilising the HCCI combustion at the HCCI operating range extremes. When intake pressure is increased the load limit is extended and higher combustion efficiency can be reached (21, 25). Intake charging can be used to increase the dilution of the mixture reducing the heat release rate. The combustion becomes less sensitive and more stable at the higher load/speed conditions (35).

1.3.5.3 HCCI control requirement

Many successful implementations of HCCI control require a combination of fast dynamic and slow response control variables. These enable the system the response to catch dynamic transients while having slow but stable control variables to increase the range and stability of the combustion. Having fast cycle-to-cycle control can also allow for the potential to transition control between HCCI combustion and SI combustion (39).

The HCCI combustion therefore requires Multi-Input Multi-Output (MIMO) control architecture to control the combustion. HCCI control dynamics are further complicated through the interaction of control mechanisms. For example if CR changes it will alter the effect of many other control variables on the combustion outcome. There are non-linear dynamic relationships between the control variables as well as the combustion.

HCCI combustion engines would not only require global control for load demand, but also cylinder specific adjustment of the auto ignition conditions. During transient operation, temperature history changes beginning in one cylinder can affect the timing of other cylinders (40). Other influences, from intake design, manufacturing tolerances, coolant flow and distribution can have an effect on cylinder-to-cylinder variation. Without cylinder trim control the imbalance between cylinders will result in unbalanced torque, increase in Noise Vibration and Harshness (NVH), excessive stress on mechanical parts and increase the potential for unstable HCCI combustion. The conventional open-loop controller based on tabulated feed forward and look-up table for linear control gains are not sufficient. The sensitivity of the HCCI combustion is greater than the tolerance of response offered from the temperature and pressure sensors used in conventional engines (40).

HCCI combustion requires an accurate feedback that can offer information on cylinder-to-cylinder combustion quality. As the combustion is controlled through indirect methods, it is difficult to guarantee the timing or the quality of the combustion. Without a direct mean of ignition and the added complexity of the control, it is important to have control feedback to maintain stability.

1.3.5.4 Uncontrollable influences and changing dynamics with time in HCCI engines

There are certain factors that cannot be influenced directly through current control strategies, either because they are defined by the engine geometry, caused by environmental conditions or physical changes to the system. These types of influences can affect the systems dynamics over time. If the systems dynamics can change with time then any fixed control methods designed around an operating region will become mismatched to the actual system dynamics. Feedback control can offer robustness against a degree of mismatch, but if the

dynamics or operating region changes significantly, control will become unstable. There are a several areas where HCCI combustion maybe susceptible to such uncontrollable influences.

Thermal constraints such as deposits can build up with in the cylinder. These deposits can be anything from oil to carbon deposits. The deposits can affect heat transfer and store heat energy. This can lead to advance in ignition timing and overall combustion phasing and location (8,41).

The distribution of the coolant temperature can affects heat transfer through cylinder walls. If the coolant flow is not uniform within the cooling jacket, the difference in heat transfer will affect combustion timing and duration (42).

Variation in cylinder-to-cylinder charge characteristics, such as temperature, pressure, humidity, density and composition of the air can have effects on volumetric efficiency and combustion timing (43, 44). These characteristics can be different from cylinder-to-cylinder depending on the environmental conditions and the intake system design.

Manufacturing tolerances can also affect any of the control systems. For example CR difference from manufacturing tolerances can affect combustion phasing up to 10-20 Crank Angles (CA). Manufacturing tolerances will affect all parts of the engine from valve train to injection systems (45).

The composition of the fuel can also have dramatic effect on timing and duration. Each different grade of fuel or ethanol blend available at fuelling stations will have different auto ignition behaviour.

While considering all these dynamic effects, there is also aging of the engine and its components. With age there will be wear, causing margins of error to increase. This can increase the negative effects on combustion of all the dynamic influences mentioned.

1.3.5.5 Overall HCCI Control Requirement and Behaviour

Overall for the HCCI combustion it is possible to outline a control requirement for its dynamic behaviour. To control the combustion the following requirements would be advantageous:

- A multi-input multi-output control architecture that allows for the complicated interacting dynamics of the system to be modelled.

- A feedback control architecture to allow information regarding the timing and quality of the combustion.

- A feedback signal that allows for direct in-cylinder feedback of combustion, or an accurate prediction of in-cylinder conditions that can provide combustion information quick enough for control to be stable.

- A level of adaptability to counteract uncontrollable influences. As a number of these influences cannot be directly measured it is important for the control to be able to adapt online to calibrate the control to the shifting dynamics of the system.

1.4 RESEARCH PROPOSAL

The future of powertrain systems is in electrification and hybridisation combined with more modern advanced IC engine strategies such as; cylinder deactivation, downsizing SI for high BMEP, GDI and HCCI. Hybridisation and advanced combustion strategies combine efficient powertrain systems with higher efficient modes of combustion. However, these systems come at a cost of a significant increase in complexity. The added complexity can add a considerable number of problems. We can begin to see some common control problems forming that are common between the advanced combustion modes:

- All advanced combustion strategies require increase in number of actuators. This creates a greater requirement for multi-input multi-output control. Descriptive models need to be dynamic and nonlinear. Using mapping requires a vast amount of work to determine optimal settings for specific engine applications. A real advantage would be found from a simple intuitive method to simplify and reduce the modelling requirement.

- Many useful internal state variables are not measured due to the difficulties in justifying the cost of additional sensors. Current feedback sensors are of indirect states throughout the engine system such as MAP or UEGO. Some of these sensors can have long time delay causing complications with feedback control. A real advantage would be a direct in cylinder feedback giving information on combustion quality and in-cylinder conditions.

- There are fast becoming a number of hybrid combustion mechanisms that allow the engines to operate in a number of different modes. This brings a requirement for a high level predictive control for intelligent seamless mode transition. Predictive control can be computationally intensive; the solution would need to be able to operate on ECU control units.

- With the added complexity of the advanced combustion systems there are a number of uncontrollable influences upon the systems. This can be in the form or component aging, environmental effects or even manufacturing tolerances and in cylinder deposits when considering HCCI combustion. Therefore, there is also a requirement for more adaptability in the control models to counteract these uncontrollable influences. As a number of these influences cannot be directly measured it is important for the control to be able to adapt its dynamic model with the shifting dynamics of the system. Feedback control will add a level of robustness, but if the dynamics change significantly with time, the feedback control will saturate.

2. ADVANCED COMBUSTION CONTROL

A typical control strategy for a PFI SI engine would be based on a map based open loop control for the majority of the engine operation range, a PID control for the idle control air loop, a proportional feedback control for the AFR loop, several feed forward controls using accessory load information, and other ad hoc compensation schemes for temperature, barometric pressure and other environmental conditions. From chapter one, we can see that there are a number of control problems associated with the advanced combustion strategies where this typical control structure becomes inadequate. This chapter will review direct feedback methods and the current control algorithms for GDI, downsize SI and HCCI combustion. The review will conclude to a proposed control structure for online adaptive MIMO automotive engine control.

2.1 DIRECT FEEDBACK METHODS

Chapter one has introduced the associated problems of indirect feedback for the control of advanced combustion modes such as GDI and HCCI. There are two possible avenues to address these problems; increase control complexity to cope with current limitations of feedback, or address the route of the problem, the feedback method.

A clear advantage for advanced combustion control would be the development of direct in-cylinder feedback that could allow information on AFR, ignition timing, combustion duration and rate of combustion for each cylinder. A number of these can be estimated from a pressure trace (83). But the combustion pressure sensors have high costs and low long term reliability and require redesign of the combustion chamber. An alternative to using the pressure transducer is to reconstruct the pressure trace from alternative feedback sources. Pressure trace reconstruction can be achieved through several methods including; measured crank kinematics (148) and inverse crank-dynamic models and measured structural vibrations (149). Both methods often use physical inverse models or neural networks to relate highly non-linear relationships back to pressure trace information. Crank based methods can offer prediction of pressure trace over a range or engine operating conditions to an accuracy of 5-3 per cent with discrepancies, while structure vibration methods can vary between 20-4 per cent error (148). However, pressure trace reconstruction is an indirect

method that by its nature can only offer a prediction of cylinder pressure. The prediction accuracy is limited by the speed and fidelity of the model and filtering methods used.

An alternative direct feedback could be to use ion current. Ion current sensing for combustion research has a long history that dates back to 1934 (94). The fundamental studies on ion current sensing for combustion diagnostics show that by using the electrical characteristics of the combustion reactions it is possible to form feedback on combustion performance. Flames have an electrical characteristic of ionization plasma, this plasma consists of both positive ions and electrons. Positive ions CHO+ and H_3O+ are considered to be the positive ions of interest that are generated by the chemical ionisation reaction, and can explained through the following principle mechanisms, equations [4.1-4.3] (95).

$$CH + O \leftrightarrow CHO^+ + e^-$$
(4.1)

$$CHO^+ + H_2O \leftrightarrow H_3O^+ + CO$$
(4.2)

(4.3)
$$H_3O^+ + e^- \leftrightarrow H_2O + H$$
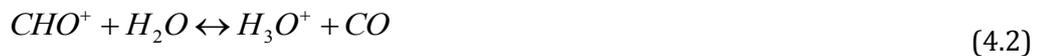
The positive ions such as CHO+ and H_3O+ are related to electrically excited CH that is decomposed from fuel. Although there are several proposed ionisation mechanisms (96-98) the process of ionisation under high temperature and pressure is not completely understood, though for SI engines it can be concluded there are two distinct stages. The first stage is the chemical phase, as fuel reacts with oxygen during combustion. The second phase, the thermal phase where compression occurs due to burning of fuel further away from the spark plug (99).

A sensor is required to detect the level of ionisation within the cylinder. To detect ionisation the sensor needs to have a positive and a negative electrode. The spark plug can be used as a sensor due to its design and the fact is has a dedicated mounting position in the cylinder. Other dedicated sensors can be designed and used. Having a dedicated sensor avoids the difficult problem of using a firing spark plug to also do a second task of detecting ionisation within the cylinder.

By applying a voltage across two electrodes, ions in the region of the electrodes pass between the earth and the positive electrode, this flow of ions creates a current. As the level of free ions within the cylinder changes with combustion, the rate of current will change. By creating

a circuit to measure this change in current it is possible to measure the ionisation within the cylinder.

The ion current signal can be processed empirically and has led to ion current probes being used extensively in SI engine research for diagnosis of combustion variability(100-103), estimation of peak pressure position and pressure curve(104-110), torque estimation(111) and estimation of Air Fuel ratio(92, 105, 112-115) . Ionisation has also been considered for combustion control strategies(110, 116-122) and is often used for diagnosis of abnormal combustion such as knock(123, 124) and misfire detection systems(92, 125, 126) These have been applied to systems in high end vehicle models such as the Mayback (127).

Ionisation has also been investigated in advanced combustion strategies such as HCCI. In HCCI combustion, the ionisation signal acquired often contains only one peak. The temperatures during the engine cycle are low in comparison to SI engine; therefore the signal is thought to come mainly from chemical phase ionisation during the reactions of combustion (86, 90).

During HCCI combustion, the ion current signal measurements are sufficient to provide information about combustion over a wide range of speeds and loads conditions (84,90). Misfire detection, pre-ignition detection and estimation of Trapped Residual Gas (TRG) percentage (84) are possible through mathematical approaches. A remarkable correlation between ion current and heat release rate has been noted by several researchers allowing Mass Fraction Burnt at 50% (MFB 50) to be calculated from ion current measurements with a high level of accuracy (87,128). There is also good correlation between the timing of maximum ion current and the timing of the maximum pressure during each cycle (86, 129). Peak pressure position can also be determined with Root Mean Square (RMS) error less than 2 Crank Angle Degrees (CAD) (84). This could allow for monitoring of variations in combustion phasing during HCCI combustion. Ion current sensing can therefore offer a potential cost effective feedback signal for both SI and HCCI engine control (84). However, there are some concerns with ion current as a feedback. It has been noted that ion current measurement can be noisy and are only representative of the area near the sensor, compared to cylinder pressure feedback that is global (90). Inhomogeneous gas composition can also affect ion concentration and thus ion current amplitude (130). Noisy signals can however be filtered for information that is required and some control structures can also be designed to be robust against nosy error prone signals (81). A local reading of ionisation in some applications could potentially be more beneficial, such as lean burn GDI engines for AFR feedback.

(removed for copywrite)

Figure 2.1 Ford's Lean burn GDI (91)

During GDI lean burn the overall global AFR is approximately 35-50 AFR, but for the combustion to initiate the fuel needs to be locally richer around the spark plug to allow ignition, depicted in figure [3.2]. The richer Air/fuel mix surrounding the spark plug will constitute the majority of the fuel charge. A local in-cylinder feedback of AFR near the spark plug could be very advantageous for more accurate control over GDI lean burn AFR. The conventional UEGO feedback offers an averaged AFR feedback of multiple combustion cycles and from multiple cylinders. Although this can be sufficient for homogenous charge GDI operation, it offers limited feedback control over the richness of AFR local A/F zone in stratified charge GDI operation. A local AFR feedback signal from the spark plug could potentially offer a means for faster, more accurate control over cycle-to-cycle and cylinder-to-cylinder variations.

2.2 CURRENT CONTROL METHODS

2.2.1 OPEN-LOOP CONTROL

(removed for copywrite)

Figure 2.2 Typical ignition timing map

Open-loop control based on calibrated tabulated maps figure [2.1], can still be used to some extent for GDI and downsize SI. However there are some limitations and problems associated with them. The open loop approach is not robust against parameter variation. For example if soot deposits build up around the EGR valve the flow rate will be affected. Due to these uncertainties, the desired mass flow rate may not be achieved with the open loop approach and the emissions and performance of the engine at a given engine speed and load point may shift away from the desired nominal performance (8).

Conventional engine management systems are dominated by feed forward control based lookup tables containing inverse approximations of different engine maps. The reason for this is the limited use of more direct sensor technology such as torque feedback or in-cylinder pressure due to expense. These open loop approaches are sensitive to engine-to-engine variation, aging and wear. Since feedback is not possible without a sensor, the lookup tables need to be very accurate.

2.2.2 PROPORTIONAL INTEGRAL DERIVATIVE CONTROL

To reduce the effort in calibration and improve the accuracy of control systems more direct feedback would be needed on the state of combustion, cylinder pressure or torque output. This would allow the control to move away from open loop control to closed loop.

The PID control structure is well established and proven in many fields of control. It is natural to look towards this classical control structure for a potential solution to advance combustion mode control.

(removed for copywrite)

Figure 2.3 Typical PID control structure

Figure [2.2] shows a typical PID control structure. As the control name suggests there are three components. The proportional gain is K_p, and multiplies the error between the desired state and the measured state. Its effect on reducing the control signal error will therefore be proportional to the magnitude of the error term. The integral gain is K_i, and multiplies the integration of the error term. The integral term has an accumulative effect on reducing the control signal error as the area under the error curve becomes larger; this is particularly useful at reducing a constant offset error. The derivative term is K_d, and multiplies the time derivative of the error term. The derivative term damps or boosts the control signal depending on the magnitude of the rate of change of error.

Gafvert et al, demonstrated that with the availability of torque for feedback a control structure based on extensive PID feedback with simple feed forward paths and extremum-based model-free online optimisation could be implemented for the control of GDI engines. The control design was a structure based on several sub-controllers. The sub-controllers could be tuned individually, with only rough process-knowledge, and with only a few parameters to calibrate. This design of engine control system requires less prior knowledge of engine parameters as long as feedback is available (46).

The PID control structure has also been applied to HCCI combustion in combination with several hardware control solutions (47-50). Simulation results show that using a PI control had promise for accurate control (51). However, when the structure was implemented on a real engine the cycle-to-cycle variation limited the speed that the controller could be implemented. PI controllers could not be made fast enough without magnifying the cycle-to-cycle variation from the combustion into instability. This implies that a more advanced structure should be investigated for HCCI combustion (52).

When PID structures are used for complex nonlinear systems control often gain scheduling is incorporated into the control design. Gain scheduling is often used by control engineers when the controlled plant is highly nonlinear and linear control techniques can no longer be applied. The designer can select an appropriate number of operating points to cover the entire operating range, then a controller is designed at each operating point using a linearised model of the plant at each operating point. Between points, the parameters are interpolated to cover the whole operating window. The overall result is a global feedback system for the entire operating window obtained through gain scheduling. The main drawback is that the resulting non-linear controller and the guarantees of the linear synthesis method do not hold anymore. The designer will need to perform extensive computer simulation to confirm stability over the non-linear control window.

There are alternative gain scheduling methods. The Linear Parameter Varying (LPV) method provides a systematic way to design controllers that are scheduled on the operating point of the system. An LPV system is a linear system whose describing matrices depend on time varying parameters. LPV techniques do not require the heuristic interpolation of the locally designed controllers and allows the designing of a family of linear controllers with the theoretical guarantees of stability and performance for the operating window.

Kwiatkowski et al, applied the LPV technique to charge control for an SI engine, where the LPV model was defined through a parameter dependent state-space model. The LPV controller demonstrated speed and response with little overshoot, simulations with the nonlinear plant indicate the controller was appropriately scheduled (53).

PID structures used for HCCI control also implement gain scheduling techniques to allow the gain values to be a function of other control-effecting variables (47, 48). However, control of HCCI combustion timing was not affected by just a single variable, but many variables. These often include octane number, engine speed, inlet temperature, fuel quantity and intake pressure to name just a few. It is important to isolate the different dependences to understand their effects on the combustion phasing but it is also very difficult to do this. It is impossible to change these parameters individually (47). The following methods are used to attempt to capture the relationships and adjust the control more accurately:

Sensitivity Function:

Sensitivity function is based on the assumption that the change in sensitivity can be represented as a product of different functions, where each individual function is for only one variable. The gain parameters are divided by the estimated sensitivity before entering the control algorithm. The control was limited and all tests suffered stability problems. The control performance was dependent on the accuracy of the estimation of sensitivity. The optimization of the controller is limited by the quality of the sensitivity map (47).

Mapping the control variables:

Several researches have implemented mapping of non-linear relationships to alter PID controller outputs.(49). This method allows optimization of parameters for each value of the control variable. This however is a time consuming process, trial and error iteration is often conducted untill a tuned controller is produced. Nonetheless the feed-forward mapping method produces better control response than the sensitivity function method, but suffered problems with control saturation (48).

M. Hillion et al (54), achieve full control of HCCI combustion through a combination of maps, dynamic feed forward terms and PID control. This control was for diesel HCCI combustion through air and fuel path manipulation. This method used a physical model prediction based on the knock integral model. From this model the control maps were generated for the control architecture. The knock integral algorithm uses in cylinder thermodynamic quantities to calculate the start of ignition in relation to the start of injection. This method allowed for prediction of combustion and the potential of more accurate maps to be calculated. However in the development of the control it was found that there was a significant difference between the control signal responses for the air path and fuel path, and that the air path had a much greater inertia then the fuel path control. This created an imbalance between the control signals, causing instability, with no direct feedback the incorrect air path caused the engine to stall. By adding a correction calculation, the controller could alter the response of the fuel path control in accordance to the error in the air path control. The added correction calculation improves the overshoot error and steady state error for the constant speed varying torque demand tests. This Control structure has been embedded into a vehicle that allowed for HCCI to be tested over European drive cycles (55).

2.2.3 MODEL BASED CONTROL METHODS

To obtain control over advanced combustion strategies using PID methods requires the use of multiple controllers to be synchronized (46, 55). However, I understand that a key feature of the GDI and HCCI system is the interactive and coupled nature of its subsystems(10, 47). It is these characteristics that dictate that a multivariable control strategy is more ideally used to coordinate all the actuators in an orchestrated fashion, this is why when using multiple PID controllers they need to be synchronized with correction functions (54).

An alternative control architecture that can be applied to MIMO systems is model-based control. This type of control structure requires an identification model of the control problem to be able to make its control decision and minimise an error function. The two main methods of producing this identification model are:

1. Construct a model from experimental input-output data (56, 57). The control accuracy will depend on the model accuracy. If the experimental data used to create a model was only taken over a specific region of operation, then the controller would only be accurate at these points. (56, 57). Therefore the experimental data needs to be wide enough to capture the entire behaviour of the engine.

2. Produce the identification model from physical models of the system (58-61). It is common practice to break down the cycle-to-cycle dynamics and chemical kinetics into separate processes to be modeled separately (58, 60, 61). The processes together would be very complicated to understand and difficult to model accurately. From the completed sub models there are several ways to gain the desired control model:

   - The resulting physical model can be linearised about an operating condition and used to synthesize a controller. Although this only gives correct control when the control signal remains within the particular linear map(59).

   - The physical model can be reduced to form a lower order model that is more appropriate for control use. To reduce a model assumptions have to be made to decrease the mathematical compexity within the model (62).

   - The physical model can be used itself to generate the HCCI engine input and output data. This data can replace the experimental data used in the first method described for generating a control model.

Blom et al. used a physical based model to create a state space model used with a Linear Quadratic Gaussian (LQG) controller for HCCI combustion control (60). The state space model and LQG controller allowed for a MIMO control with two control variables. The two control variables offer one with fast response and one with slow response. By combining the two

control actions a mid-ranging control was achieved. The control has the capability to use one slow signal and one fast signal in a way to cover both fast transients and load disturbances. This is only possible due to the well-developed state space model combined with model-based control. The model-based controller however did struggle with control saturation.

J. Kang et al from General Motors, show successful HCCI control architecture based on simple models on bulk energy release. The undisclosed model based control architecture is chosen to have fast and slow control loops that regulate external EGR and air-fuel ratio values, based on measured/estimated external EGR and AFR. The slower control loop ensures the combustion timing is stable at steady state, the faster loop, is used to keep stability during transients. This structure also uses a second controller for combustion balancing purposes. Individual pressure sensors and high-pressure fuel injection allow the fuel strategy to balance cylinder-to-cylinder variation. This engine controller has been implemented with demonstration vehicles and allows for HCCI operation as well as HCCI to SI transitions (63).

2.2.4 MODEL PREDICTIVE CONTROL

Model Predictive Control (MPC) is an advanced control methodology that has significant advantages over model based control methods.

The main reasons for MPC success in control are:

- It can operate with multivariable control problems.
- It can account for actuator limitations.
- It allows for operation closer to constraints which can lead to more beneficial operation.

The MPC control is suited to controlling MIMO plants, and can be used to control a great variety of processes, including those with non-minimum phase, long time delays, or open loop unstable processes. There are many variations of the standard MPC controller, but they all usually share some common properties (64).

- Explicit use of a model to predict the future process output.
- Calculation of a control sequence by optimizing, usually on-line, a performance index.
- A receding horizon strategy, meaning that at each instant the horizon is moved towards the future and only the current signal is applied at each step.

(removed for copywrite)

Figure 2.4 Model predictive control horizon

MPC is based on an iterative, finite horizon optimization of a plant model, figure [2.3]. At each sample interval of an MPC controller the current plant state is sampled, then a cost minimizing control strategy is computed. This online calculation is used to explore trajectories from the current plant state to achieve a desired reference trajectory. Only the first step of the control strategy is implemented, then the plant state is sampled again to begin the calculations from the new plant state, therefore providing a new control and new predicted state path. By altering the number of time steps in the predicted horizon the complexity of the optimization can be altered.

Most real processes are only approximately linear within limited operating points. Despite this, linear MPC approaches are used in the majority of applications. With the feedback mechanism of the MPC, it is possible to compensate for prediction errors due to mismatch between the model and the plant process. When only linear models are used, the superposition principles of linear algebra allows the effects of changes in multiple independent variables to be added together to predict the response of the dependent variables. This simplifies the control problem to a series of direct matrix algebra calculations that can be fast and robust.

MPC control can explicitly handle MIMO systems with constraints (60). The constraints for advanced combustion modes will be the limits of the control actuators and operating ranges, for example AFR. The problems associated with MPC control come from the complexity of the models used within the control structure. If complex physical non-linear models of engine combined with their constraints on the operating ranges of the actuators are used the computational requirement are unfeasible. However, using appropriate control model techniques such as state-space modelling can allow for the dynamics to be modelled while reducing the computation expense. Dutka et al, uses state space models with an MPC architecture for simultaneous MIMO control over engine torque and AFR for multivariable SI engine control. The multivariable solution balances the accuracy in the regulation of AFR and tracking the torque demand profile. Simplifications in the modelling were introduced for the relationships between torque and air intake charge to reduce computational effort. The numerical simulations indicated the controller was capable of simultaneously influencing lambda and torque tracking with superior performance in response time and accuracy over conventional control (65).

(removed for copywrite)

Figure 2.5 Block diagram of SISO MPC control for inlet valve timing control for HCCI combustion (66)

Standh et el, achieved control over diesel HCCI using a combination of PD controllers and a MPC controller, figure [2.4] (66). In the structure a state-space model was used for modelling the engine dynamics for Intake Valve Closing (IVC) at or after the BDC. The model was used with the SISO MPC control. Simplifications where also made by estimating the nonlinear relationship between valve timing and CA50 as a polynomial of fourth order and that the nonlinearity is static and independent on operating conditions. These simplifications decrease the computational effort of the controller. Control designs for non-linear systems are more complex then linear methods. The non-linear inverse relationships for the control variables where linearised to allow the system to be approximately linear. When the models used for a MPC controller are linear, the control problem can be solved by finding the minimum in a convex optimization problem. For the intake valve control model a quadratic programming solver was used to find the minimum.

Several tests were conducted with the MPC controller. The results show that the MPC controller can follow desired Indicated Mean Effective Pressure (IMEP) changes with and without constraints on IVC. The MPC controller was also tested against simultaneous changes in intake temperature, engine speed and fuel ratio. The results showed that the model could maintain a set point load while limiting disturbance, showing a level of robustness against disturbances.

2.2.5 NONLINEAR MODEL PREDICTIVE CONTROL

MPC often uses simplified linear models to ease the computation expense of online calculation of future horizons. However, when a linear model becomes insufficient in capturing the plant processes nonlinearities, the process may require control with a nonlinear Model Predictive Controller (NMPC). Like linear MPC, NMPC also requires the iterative solution to an optimal control model for a finite prediction of horizon. However, the NMPC model may no longer have a convex solution. This can create challenges for NMPC stability and numerical solutions (67). With any MPC scheme the model used for carrying out the on-line predictions of future plant behaviour is the key component. In many continuously operating plants, an MPC is implemented as part of an online optimization scheme, where the

operating points keep shifting. The linear prediction models used in most MPC schemes, are typically developed only once in the neighbourhood of a nominal operation point, at the beginning of the controller implementation. Such linear-model-based MPC schemes may prove ineffective if large transitions have to be made in the operation conditions. Also, if plant dynamics can change with time, a large mismatch would develop between the model and the system. Though MPC formulations are equipped to deal with a certain degree of plant-model mismatch, the presence of large plant-model mismatch can significantly deteriorate the controller performance.

The problems of maintaining MPC performance during changing operating conditions and/or plant characteristics has been tackled through:

- Incorporating robustness at the controller design stage.
- Employing nonlinear models for prediction.
- Updating parameters of the linear predictive model online.

The last alternative is attractive from the viewpoint of implementation. However, it has received relatively less attention in the MPC literature compared to the first two alternatives.

Over last two decades, a number of Adaptive Model Predictive Control (AMPC) schemes have been proposed in literature, table [2.1] (68).

Table 2.1 Comparison of adaptive MPC schemes(68)

Adaptive MPC scheme	Advantages	Disadvantages
Generalized predictive control (GPC)	Control for continuous as well as semi batch processes. Models are integrated with white-noise disturbance needed for online estimation.	High-pass filtering of input–output data can lead to large bias errors in identifying the model. Can cause poor model predictions and poor controller performance.
Simultaneous MPC and identification (MPCI) approach	Solves on-line optimization problem and includes constraints that ensure persistent excitation (PE). PE is necessary for on-line parameter estimation.	This method is computationally demanding.
Finite impulse response (FIR) identification model With MPC	FIR models generate excellent long-range predictions with respect to manipulated inputs.	For multivariable systems, estimating FIR model coefficients on-line can become inaccurate. FIR models cannot capture the dynamics of unmeasured disturbances affecting the plant.
Formulate MPCI based on the deterministic auto-regressive moving	Suited for capturing the effects of stationary as well as non-stationary disturbances.	Requires a large number of model parameters to be estimated on-line, causing increase

average (DARX) model		computational demand.
The certainty equivalence adaptive-control approach	Assumes the identified linear model gives exact representation of the process dynamics at the current operating point. The estimated model must be controllable or stabilizable. Laguerre filter-based models are often used.	Laguerre or any other orthonormal basis filter-based models have fixed poles. Adaptive control based on fixed-pole models may not provide uniformly satisfactory performance over the entire operating range.

The key issues while developing an adaptive MPC scheme is choosing a model structure that can generate accurate long-range predictions and is capable of capturing the non-stationary unmeasured disturbances such as compression ratio change or aging, while keeping computational demand low.

One study has shown a proposed two-tier modelling scheme that successfully captures time varying system dynamics with respect to manipulated inputs as well as immeasurable disturbances over a wide operating range (68). The work presented shows a comparison between the performance of a conventional nonlinear MPC and the developed adaptive MPC scheme. The conventional nonlinear MPC lead to large overshoots in responses due to the time varying dynamics. However, the proposed adaptive MPC was able to adapt to the time varying behaviour of the system and achieve the desired transitions without overshooting. However, the industrial plant simulations and examples used in this work all operated with long time constraints, with sample rate ranging from 6mins to 3 seconds.

2.2.6 ARTIFICIAL NEURAL NETWORKS

(removed for copywrite)

Figure 2.6 Typical feed-forward neural net

Artificial neural networks (ANNs) have been the focus of a great deal of attention during the last two decades, due to their capabilities of modelling nonlinear systems by learning from data. The most common ANNs used are often feed-forward structures that are trained through back propagation for system identification (69), figure [2.5].

Two layer ANNs, with sigmoid transfer functions in the hidden layer and linear transfer functions in the output layer can approximate virtually any nonlinear function of interest to any degree of accuracy, provided sufficient hidden units and training data are available (70). ANNs are often used in indirect control systems. The outputs of the ANNs are processed via algorithms to determine the control actions. Direct control structures can also be implemented with ANNs. The direct outputs of the ANNs determine the control actions. Often the nonlinear Recurrent Neural Networks (RNNs) are used. RNNs simulate the steady state dynamics of the plant and also the dynamic behaviour of the control system. RNNs have been implemented to good effect for AFR control in SI engines (71). During transient tests that simulated fast throttle open and closing, the AFR could be maintained within 1% of stoichiometric.

ANNs can be used at a variety of different levels in control structures. Collins et al, demonstrate the use of ANNs for several models in a control architecture proposed for downsize SI combustion control (6). The control proposed uses coordinated, control modules that are easy to synthesize, to implement and to tune. The control comprised of internal model control for throttle, MPC for waste gate and model-based control for VVA. ANNs were used to replace physical models that were too complex to be embedded for online control calculation. MPC often uses linear models to decrease the computational requirement of the control. However, most real systems are nonlinear. Using a nonlinear model for a MPC control would increase the complexity as the solution would be based on solving a non-convex optimization problem online. This would lead to a large computational burden. Very few applications of NMPC deal with real time implementation and generally are applied to systems that have long time constants. By using ANNs for the nonlinear modelling of the plant an instantaneous linearization and constraints holding method can be applied that simplifies the problem of NMPC and decreases the computational burden. ANNs were also used in the modelling of the dynamic relationships between intake and exhaust valve timing with burned gas mass and scavenged air mass, for the neural based control of VVA.

ANN models have the capability to learn non-linear relationships of HCCI combustion (72). Their ability to learn over a short training period, generalize the knowledge, and give accurate predictions of engine behaviour is very attractive for this application. ANNs offer an alternative method to capture the non-linear dynamics in a non-linear model, with accuracy, and the potential for online adaptation.

(removed for copywrite)

Figure 2.7 Model Reference Control configuration (72)

ANNs can be implemented as controllers, The ADAptive LInear NEuron (ADALINE) ANN and standard back propagation ANNs, are used in model reference control (MRC) structure figure [2.6]. The ANN can train on the error between the plant output and the reference to adjust its parameters online. The back propagation ANN out-performed the ADALINE network for following transient changes. The ADALINE networks maybe better suited to modelling non-transient system behaviour (72).

(removed for copywrite)

Figure 2.8 Adaptive control architecture (40)

J. Franz et al, developed control architecture for HCCI combustion that allows for online adaption of the control rules through an ANN model, figure [2.7] (40). Lookup tables are restricted to two-dimensional applications as memory and computational requirements grow with the number of inputs. There are a lot of variables that can affect the HCCI combustion. For such high dimensional problems, ANNs lend themselves well. The ANN structure LOcal LInear MOdel Tree (LOLIMOT) was combined with a robust PID feedback control architecture for online adaptation. By using the LOLIMOT it was possible to adapt, online, for global and cylinder specific auto ignition time varying dynamics. Decoupling is unfortunately required to release or lock the control adaptation by the LOLIMOT to specific times. If the adaptation took place out of coordination with the PID controller, the controller is affected and the whole system swings into turbulence. Nonetheless this shows an application where ANNs are used for modelling the non-linear behaviour of HCCI, while also allowing online adaptation for a time varying dynamic system.

2.2.7 ALTERNATIVE NEURAL NETWORK CONTROL ARCHITECTURE

(removed for copywrite)

Figure 2.9 Real-time dynamic controller for robot arm control (74,75)

The Cerebellar Model Articulation Controller (CMAC) is an ANN that offers a unique control scheme developed from models of human memory and neuromuscular control. The CMAC is capable of learning nonlinear functions extremely quickly due to the nature of its training algorithm for weight update. The update can be much faster than that of a feed forward ANN using back propagation training. Due to the training speed, it lends itself well to on-line learning. The CMAC has been used in a real-time dynamic controller for robot arm control (74, 75), figure [2.8]. This CMAC control system proposed by Miller (75) is a type of MRC structure. The CMAC network is used as a feed-forward controller, which learns the inverse dynamics of the plant. In this scheme, the constant gain controller helps the CMAC to learn the inverse of the plant. However, these two controllers are independent, and conflict between their control response can cause instability (76).

(removed for copywrite)

Figure 2.10 Modified Real-time dynamic controller for robot arm control (77).

An alternative structure has been proposed by Yuan (77), figure [2.9]. The difference between this structure and control system proposed by Miller (75) is the input to the CMAC. In the new scheme the input is the tracking error of the system rather than the desired output and the system output. Instead of having the system inverse solely learned by CMAC, the constant gain controller and the CMAC are integrated to approximate the system inverse together. The CMAC will therefore be used to learn the difference between the system inverse and the constant gain controller (77). This method allows for a much more stable control system. Although these methods are used for robot arm dynamics, the CMAC has been shown to be capable of modelling and controlling the fuel injection system for SI engines (78, 79).

2.2.8 FUZZY LOGIC CONTROL

Fuzzy logic control (FLC) can be used as an alternative to PID control, adaptive control, observer-based control, and supervisory control. Fuzzy logic is a mathematical system that

analyses analogue input values in terms of logical variables that take on continuous values between 0 and 1. This is where fuzzy logic differs from classical or digital logic. Digital logic operates on discrete values of either 1 or 0, true or false. Fuzzy logic has the advantage that the solution to the problem can be cast in terms that human operators can understand. This allows for operator experience to be used in the design of the controller.

FLC is best suited for the control of nonlinear dynamic systems. Fuzzy logic rules and Membership Functions (MF) can be used to approximate any continuous function to any degree of precision.

(removed for copywrite)

Figure 2.11 Fuzzy Logic membership functions

Figure [2.10] shows a simple non-linear relationship between temperature and fan speed. The input variables in a fuzzy control system are mapped by sets of MFs known as fuzzy sets. There are four MFs mapping the two variables. If more precision was required more memberships could be added.
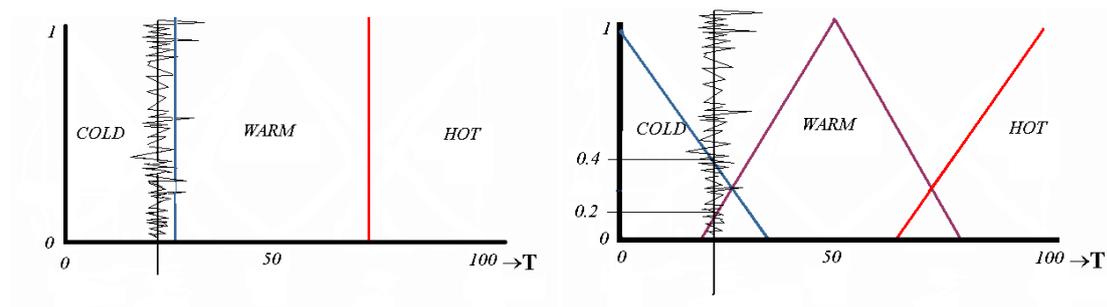
Figure 2.12 comparison of Fuzzy logic to logic for separating input space

The degree of activation of the MFs will dictate the fuzzified level output of that input space, this process is called fuzzification. The benefit of fuzzy logic over conventional logic as mentioned previously, is that it can be classified as more than one input membership at once, i.e., it does not have to be 0 or 1, but can be any value in-between. This allows for a level of robustness against nose/error prone signals. For example, figure [3.3] shows an input space for temperature. If the input temperature signal is on the border of being cold and warm in normal logic, noise or error on a signal could cause it to be determined as warm. This is because in normal logic its membership with the variables cold, warm and hot can only be 0, no membership, or 1, full membership. If the same signal was input on a fuzzy system with triangular MFs it would have a degree of membership with both cold and warm. In this

example approximate membership of 0.2 and 0.4 respectively. Therefore, there is a transition between states through MF design allowing for robustness against noise or error on signals. The shape and location of the MFs allow the dynamics of a system and its transition between states and variables to be modelled to any degree of precision. This robustness against noise and error on signals would be very beneficial for use with cheap feedback methods such as ionisation and current conventional engine feedback sensor measurements.

FLC advantages over traditional control lies with systems where mathematical models of the process do not exist or are maybe too difficult to model. A control system built on empirical rules can be more effective.

Advantages have been found where FLC has been applied to engine control applications. For example, when FLC was applied to idle speed control, error was reduced significantly compared to conventional control, and in some cases eliminated (80). Combining FLC with the feedback technique of ion sensing offered excellent spark timing performance with robustness in the presence of disturbances (81). An advantage of FLC is the ability to interpret unclear noise prone input signals. Input signals are classified into MFs allowing fuzzy logic to determine values from noisy signals. Using FLC with ionisation sensing could potentially improve its feedback robustness.

FLC can be inbedded into ANN architecture. FLC requires an expert on the system to determine the control linguistic rules and fine-tune them. If the expert knowledge is not available this process can be difficult to get correct.

ANN can model non-linear systems, but they are often used like black boxes. Through combining the two methods it allows for implementation of linguistic fuzzy rules, and allows for training of the MFs to an optimal output. These new types of structures are known as Fuzzy Neural Networks (FNN).

(removed for copywrite)

Figure 2.13 SLFNN for water injection control (82)

A Self Learning Fuzzy Neural Network (SLFNN) has been applied as an optimal online water injection controller, figure [2.11] (82). By using the engine's feedback for knock intensity, the

controller uses an on-line training process to update the weighting factors associated with each fuzzy rule. The control required no initialization or training of weight factors. It can self-generate the desired control map after a short period of engine running.

2.2.9 *CURRENT ADVANCED COMBUSTION CONTROL CONCLUSION*

Each of the control methodologies and architectures above has contributed towards solutions of the problems associated with MIMO control or adaption to control for time varying systems. However, each control methodology has its own advantages, disadvantages and aspects that could be further improved to solve the control problems associated with advanced combustion control. Table [2.2] summaries these points:

Table 2.2 Comparison of control architecture for advanced combustion control

Control architecture	Advantages	Disadvantages
Proportional Integral Derivative	Well established and trusted. Combined with physical based models/maps or gain scheduling techniques allows control over system nonlinearities.	If control relies on fixed feed-forward models, the performance will decrease with time varying dynamics. (40) PID response is described insufficient for HCCI, without magnifying cycle-to-cycle variation (52). The PID structure is not ideally suited for MIMO systems and can suffer with control saturation.
Model based	Explicitly handle MIMO control problems. Allowing for simultaneous synchronization of fast and slow response hardware.	Problems with control saturation. If control signals are based upon fixed models, the control will become less optimal with the changing dynamics over time.
Model Predictive Control	Model predictive type control can explicitly handle control saturation (60) MPC control can control MIMO systems with constraints, often allowing control closer to the boundaries of operations where advantages can be found.	Due to complexity of the engine dynamics, linear model approximations are needed to ease computational requirements.
Adaptive MPC	AMPC has proven to be able to adapt to time varying behaviour of industrial systems and achieve desired transitions without overshooting in the presence of time varying disturbances.	Difficulty achieving model structures that can generate accurate long-range predictions and is capable of capturing the dynamics during unmeasured disturbances, as well as keeping computational demand low.

Control Architecture	Advantages	Disadvantages
Neural Network	Can be used for system identification and control for MIMO systems. ANNs can also be used for non-linear modelling, or control adaption for other types of control structures. ANNs allow for continuous online learning for optimal control solution.	ANNs are black box systems, the process in-between input and output can be unclear. Training accuracy can depend on many variables; choice of ANN, ANN size, architecture, training data and training algorithms. Online adaption can be unstable.
Fuzzy Logic Control	Alternative MIMO control to PID, adaptive, observer-based and supervisory control. Can model dynamics through linguistic rules and MFs based on expert knowledge. Can be implemented within ANN architecture to allow training to optimize relationships. Robust to noise prone inputs, allowing of cheap sensors, low resolution analogue to digital converters. Proven capability for application in control with adaptation online, allowing for robustness against time varying dynamics.	Requires expert knowledge to generate linguistic rules and MFs, it is difficult to estimate such functions. Difficult to combine outputs of several fuzzy rules and defuzzifing the output, this often requires trial and error. As the rules and MF are fuzzy (continuous values between 0 and 1) the rule formation may never truly give zero error as the result will always be neither 1 or 0 but some value describing some set difference from this.

2.3 WHAT IS MISSING?

PID architectures struggle with synchronised control over MIMO systems. It is possible to achieve synchronised MIMO control through combining several controllers with additional correlation functions (54). But this is not an elegant solution when there are other potential structures that explicitly handle MIMO systems. PID also suffered with control saturation. There is a requirement for more accurate dynamic modelling of MIMO system and for the potential of adaption to system dynamics that can change with time.

Model reference control like the LQG controller, showed an improvement over PID control with the ability for synchronized MIMO control. However, it still suffers from control saturation and does not allow for online adaption.

MPC can bridge the gap and explicitly handle control saturation through prediction based on a model of the system (60). MPC control can also control MIMO systems with constraints, often allowing control closer to the boundaries of operations where advantages can be found. However, the models used often need to be simplified to linear approximations to keep computational expense low. Sometimes these linear models are not sufficient for accurate control of nonlinear systems. Feedback can often add robustness against a mismatch in model and real plant dynamics but if the plant's dynamics can change with time this mismatch will increase. If the models used are simplified to lower computational expense and are fixed, then the predictions can only be as good as the model's accuracy. If the dynamics of the system can change with time, the models will become inaccurate.

Adaptive MPC control can be used for systems that have dynamics that change with time. However, there is real difficulty in achieving model structures that can generate accurate future horizon predictions and have the capability of modelling the non-stationary dynamics, while keeping computational demand low. ANNs have been applied with MPC control to great effect (6). ANNs have exceptional learning ability. If the ANNs used for creating nonlinear models for online linearization could also be trained online for changing plant dynamics, this could be an interesting direction for the problems associated with adaptive MPC control.

The ANN architectures investigated can allow for online adaption and control for MIMO systems. Once the ANN type structures are trained, they are very fast at control as no predictive calculations need to take place. They have been implemented into SI control systems with good performance. J. Franz et al, implementation showed excellent promise for ANN adaptive control for HCCI combustion (40). The main obstacle highlighted by this work is combining ANN adaption with a MIMO control while maintaining stability during ANN update. Ideally a more robust control architecture that allows for continuous online adaption is required, similar to that seen in the work of (75) and (77).

Fuzzy logic offers an attractive alternative to conventional MIMO control as it does not require mathematical modelling of the system to generate a nonlinear control solution. The dynamics of the system are modelled internally through a number of linguistic rules and MFs. Fuzzy logic is complimented with ANNs for control as they share similar architectures. When combined they can produce self-learning nonlinear control systems (82). Combining ANN's learning with fuzzy logic's MIMO nonlinear control and robustness against noise prone

signals could allow for a very attractive control for advanced combustion strategies. This control could be ideal where online adaption is needed of time varying dynamics and feedback can only be obtained through cheap sensors.

The missing elements for advanced combustion control are finding a combination of a MIMO feedback control that allows continuous adaption without causing instability or excessive computational requirements. Some advanced combustion modes, like HCCI may require cycle-to-cycle control, if control architecture is based on complex nonlinear models this would be unfeasible.

The control structure is required to remain stable in the presence of measurable disturbances as well as un-measurable disturbances in the form of dynamics that can shift with time. It would also be advantageous if noisy, error prone feedback can be interpreted. This would allow for less expensive feedback sensors to be used. It would also be beneficial to find a solution that can reduce the modelling and calibration burden of the increasingly complex automotive powertrain systems.

2.4 PROPOSED CONTROL STRUCTURE

The following proposed control structure for advanced combustion control can be broken down into three main components, the direct feedback, MIMO control structure and method for adaption.

2.4.1 DIRECT FEEDBACK

From chapter one we know that direct feedback is desirable for advanced combustion modes to reduce the reliance on calibration and mapping of nonlinear dynamics. In some cases direct feedback is essential. However, feedback is limited by cost and reliability to indirect sensors.

From the review in this chapter we can see direct feedback of combustion state and in cylinder conditions would allow more optimal control. If indirect feedback is used to predict past combustion performance, the control will be based on model prediction. The prediction accuracy is limited by the speed and fidelity of the model and filtering methods used. Direct feedback from a cylinder pressure transducer can offer the feedback information required,

but comes at high costs and low long term reliability and require redesign of the combustion chamber.

Ionisation offers a direct in-cylinder feedback technique that can use a readily available spark plug as the sensor. Therefore, the Ionisation feedback system can offer direct in-cylinder feedback but at low cost. Ionisation feedback on combustion characteristics has a strong research background with SI and HCCI applications, often detailing relationships with knock, AFR, ignition timing, combustion rate and duration. If the information available from ionisation can be extracted and combined with an appropriate control structure it could potentially solve a number of problems associated with the control of advanced combustion strategies.

2.4.2 MULTI-INPUT MULTI-OUTPUT CONTROL FOR NONLINEAR ENGINE CONTROL

From chapter one we can see downsize SI, GDI and HCCI engines rely on an increase in key combustion technologies to operate. The multiplication of control parameters and their interactive and coupled nature dictate that a multivariable control strategy be adopted.

From the review of control in this chapter, the flowing conclusions on multivariable control for advanced combustion systems can be made. PID control struggles to operate MIMO control and often requires several sub controllers to be linked to enable a MIMO solution (46). Model based control has been described as the best option to enhance engine control (6). However, control models are often a trade-off between model accuracy and computational expense. Many control models are fixed linear approximations of more complex nonlinear models (62, 65). Most systems are often nonlinear, using a nonlinear model is more complex and the computational burden would limit the application for control. If nonlinear modelling techniques could allow for accuracy with reduced computational demand this would offer advantages for complex nonlinear control problems where linearization becomes less effective (68).

Fuzzy logic offers an alternative method of MIMO control. The modelling of the nonlinear behaviour is through linguistic rules and MF. The control dynamics can be tuned to nonlinear requirements through MF choice and design of rules. By using FLC instead of other model based control could reduce the modelling effort for MIMO systems (80). FLC systems share very similar architecture to ANNs. ANNs can model non-linear systems, but they are often used like black boxes, where the process in-between inputs and outputs are unknown. By combining NN with FLC it allows the intuitive MFs and rule base method for creating a

control system, but also allows those MFs to be tuned via training data. This allows for a structure where the process between the inputs and outputs is more clear and controlled, while allowing a degree of automated calibration through ANN training (82). From chapter one, we know modelling and calibration efforts are more demanding for advanced combustion modes. FLC can offer an alternative method for modelling through linguistic rules and MF decreasing nonlinear modelling complexity. But it also offers a control structure that architecture can be combined with ANNs for automated calibration. Reducing the amount of effort required in determining and calibrating the control parameters specific to each engine.

2.4.3 ADAPTION OF CONTROL FOR TIME VARYING DYNAMICS

From the review in chapter one, we can also see there are certain factors that cannot be influenced directly through control strategies. This is either because they are defined by the engine geometry, caused by environmental conditions or physical changes to the system. These types of influences can affect the systems dynamics with time. If the systems dynamics can change with time then any fixed control methods designed around an operating window will become mismatched to the actual system dynamics. As described in chapter one, all advanced combustion strategies will suffer from time varying dynamics to some extent.

There are two main possibilities for online adaptive control for time varying dynamics; the AMPC methods and ANN methods. However, due to the computational demand of MPC and adapting models online, very few applications of adaptive MPC deal with real time implementation or are applied to systems that have long time constants (68).

ANN methods on the other hand lend themselves well to adaptive control applications. The ANNs are able to learn nonlinear dynamic relationships for systems from input and output data. The ANNs are particularly useful for modelling systems where the dynamics between the inputs and the outputs are uncertain and difficult to model mathematically. The ANNs parallel architecture allows a generalization of the NN trained knowledge so that estimations can be calculated on the dynamics relationships very fast. In control applications, this means the relationships of the system can be modelled with a nonlinear method, adapted online and due to the simple mathematical structure of ANNs, the model will have low computational expense.

2.4.4 OVERALL ARCHITECTURE TO COMBINE TECHNOLOGY

Each of the technologies chosen for feedback, MIMO modelling and control and adaption have successful applications and reliable background for their area of application. Some of these technologies have already had some combined applications, ionisation feedback with FLC (81) and FLC with ANN adaption for MISO automotive control (82). Where these key technologies have been combined advantages have been achieved over conventional methods. Through the authors knowledge there is no work combining all three methods, fuzzy logic, ionisation as feedback and ANNs for online adaption. A combination of the three could offer an alternative MIMO control structure which simplifies modelling and calibration, offers online adaption to time varying dynamics and incorporating experimental direct in-cylinder feedback methods too potentially solve several feedback control problems associated with advanced combustion methods.

However, there are problems associated with combining these technologies. Ionisation is experimental feedback and is often related back to pressure trace for understanding. As ionisation holds information on combustion characteristics, the signal could be used directly for feedback. The ionisation signal is noisy, error prone and only locally representative of cylinder conditions; this could cause problems if it is to be relied upon for control. The key to using ionisation is finding the exact parameters from the signals that give information on the combustion parameters required. ANNs have the capability to learn dynamic systems and adapt online. But their performance relies on adequate information in their training data, and the methods used for training and adapting. This can often be very difficult to balance. If there is not enough information in the training data it will not represent the full system. If there is too much information in the training data, it can overload the ANN and it will not be able to train to the required relationships accurately. Even if the training data is of correct size and holds enough information, the training algorithm used can greatly affect the performance of the learning. If the same set of data is presented to an ANN too many times, it can over train to the specific relationships in the training set and loose its ability to generalise knowledge. If the training data is presented to few times, the ANN may under train and not generalize any relationship behaviour. The process of training ANNs and adapting ANNs also requires different approaches. To train an ANN you may want to model an entire system, where with adapting an ANN, you only want to change its understanding of a relationship for a small part of the system. Changing a small part of a nonlinear relationship of a system without effecting the generalisation of the overall relationships can be difficult. Applying ANN adaption online to control can also be problematic. J Franz et al, found if online adaption takes place when the coordination of the cumulative outputs of controller and ANN are not

constant, the adaption interacts badly with the controller and the system swings into turbulence (40). It was also important to make sure the ANN only adapted to particular influences, i.e. global immeasurable disturbances and not transients.

It is therefore important to consider the interaction between the FLC and the ANN adaption. To eliminate interaction between control modules the ANNs can be combined into a direct controller with fuzzy logic for adapting control (82). However, with more complex MIMO systems, adapting a sole controller may raise some safety concern. If the ANN adapts incorrectly and it is the sole controller, there will be no way to hold and/or reset without deactivating the engine. Using a parallel type control has the risk of undesirable interaction but would offer added safety by having a fixed feedback controller to fall back on.

A potential solution is to look at the work by Yuan (77) and Miller (74, 75), for combining ANNs with fixed feedback controllers for the adaptive control over robot arms. Miller also found that if the ANNs used for adaption are independent of the feedback controller a conflict between them can cause instability (76). It was understood that because the ANN solely learned the inverse of the plant it caused the ANN to act as a separate controller causing the conflict and instability. Yuan altered the architecture, so instead of having the system inverse solely learned by the ANN, the constant gain controller and the ANN were more integrated and used to approximate the system inverse together. The ANN was used to learn the difference between the system inverse and the constant gain controller (77). Through changing the structure to allow the ANN and the fixed feedback control to model the inverse of the plant together it allows the two controllers to work more harmoniously.

The MIMO nonlinear dynamics of engine control is more complex than the robot arm problem. The dynamics of combustion control can be nonlinear depending on a number of different variables (47), and there will be the requirement for multiple outputs. However, the theory behind the architecture is proven and would only need subtle modifications to incorporate the modelling and adaption required for automotive application.
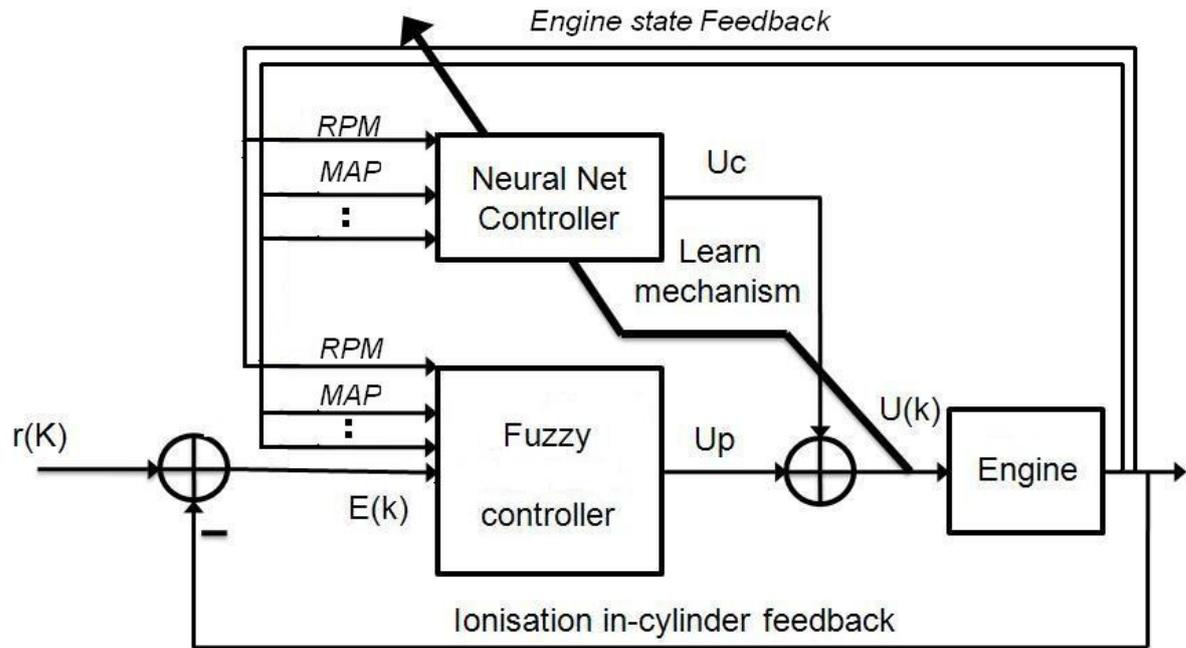
Figure 2.14 Proposed modified control architecture for MIMO fuzzy neural network online adaptive automotive engine control

Figure [2.14] shows the proposed modified control architecture that incorporates fuzzy logic feedback control with ionisation and ANN adaption. To modify the structure to engine control a number of different feedback parameters are required. There is the direct feedback for in-cylinder conditions for the feedback controller, but also a number of engine state feedback parameters. At different engine states the engine control parameters will need different levels of actuation. The relationship between different combustion states and the controls are nonlinear. For example spark timing at engine speed 1000RPM and throttle position 50% may be 12 degrees advance. But changes in engine temperature, intake pressure and fuel amount may require this to be more advanced or retarded to maintain constant combustion timing. As combustion is affected nonlinearly by a number of parameters, it is useful to feedback available information. In the original control structure used for robot arm control a PID structure was used for the fixed feedback controller. As the new control structure is being modified for MIMO application and the PID was seen less appropriate for multi-variable control, the fixed feedback control is being replaced with FLC. In the proposed control structure the fuzzy controller is required to be fixed, but this does not stop the fuzzy controller being combined with an ANN for offline optimisation and calibration beforehand if required.

The ANN within the structure will need to be modified to accommodate an increase in number of inputs and outputs. However the methods of implementation will remain mostly unchanged. The ANN will train on the difference between the engine inverse dynamics and

the fixed control model based in the FLC. If there is an error between the FLC model output and the inverse model's predicted ideal control input for a given state, then the ANN will learn the difference and apply it as a feed-forward term. This will allow for a self-calibration to adapt to model/system mismatch and allow for immeasurable dynamic changes with time. Without the adaption, dynamic changes and model/system mismatch would cause the feedback control to become saturated.

Overall the modified architecture can potentially offer a number of advantages for automotive engine applications:

- In cylinder feedback measurement from ionisation is a cheap alternative to pressure trace for information on combustion.

- FLC can allow robustness for interpretation of noise prone feedback signals. Therefore, combining FLC with ionisation could make the feedback more robust.

- FLC allows nonlinear control for nonlinear MIMO systems through linguistic rules and membership functions. The intuitive process to create a FLC system can offer an alternative method for modelling systems where mathematical solutions are unclear or difficult to achieve.

- FLC can potentially bridge the gap between expert knowledge and automating the control of advanced combustion.

- Fuzzy logic MFs can be tuned via implementation in an ANN architecture. If optimisation of rules and MFs is needed this reduces the need for trial and error development.

- A modified version of the structure proposed by Yuan et al, could allow ANNs to be used to continually learn the difference between the FLC and the inverse dynamics of a system. This allows for changes in dynamics to be counteracted via an adaptive feed forward NN control term.

- The proposed control structure is potentially more robust then using just an adaptive ANN or FNN as direct control. If the ANN adapts incorrectly there is still the fixed FLC to maintain a level of stable control.

- The fixed FLC also helps the ANN learn the correct input/output dynamics through maintaining an acceptable range of control. This minimizing the possibility for the ANN to update incorrectly.

- The FLC and ANNs work together to minimise the same error, this allows for a more stable control system that will be robust against the two controllers causing instability between each other.

3. FUZZY LOGIC AND ARTIFICIAL NEURAL NETWORK THEORY

3.1 FUZZY LOGIC CONTROL

(removed for copywrite)

Figure 3.1 Typical Fuzzy logic control system

A fuzzy system is fundamentally a static nonlinear mapping between inputs and outputs. A simple FLC system can be seen in figure [3.1]. The FLC is based upon four main elements.

1. A Rule Base. The if-then rules contain fuzzy logic quantification from an expert's linguistic description on how to control a process.
2. An inference mechanism. This can also be known as inference engine or fuzzy inference. The fuzzy inference mechanism emulates the expert's decision making in interpreting and applying knowledge about how to control the process.
3. A fuzzification interface. This interprets the controller's inputs into a format that the inference mechanism can easily apply to the rule base.
4. A defuzzification interface. This converts the conclusions of the inference mechanism into actual inputs that can be used for control of the process.

The inputs and outputs are described as "crisp", this description indicated that they are real numbers not a fuzzy (linguistic) description of a state. The fuzzification block converts the "crisp" inputs into "fuzzy" sets. The inference mechanism uses the rule base to produce fuzzy conclusions. The de-fuzification block converts the fuzzy conclusions back into real "crisp" values that can be used for control.

In this section, each of the elements of fuzzy logic will be explained to provide an understanding of the mathematical theory behind the architecture of the fuzzy logic controller.

3.1.1 FUZZY LOGIC INPUTS AND OUTPUTS

The fuzzy logic controller is a control architecture that can be designed around the automation of how a human expert would successfully control a system. To be able to control

a system an expert needs to understand what information will be needed to use as inputs in the decision making process. Often error terms are used as this compares the feedback to a desired set value. This often makes intuitive sense, but other information can often be used depending on the system that requires control.

The human expert would also need to be able to identify appropriate control variables for the system. These control variables would need to be able to affect the systems state to allow change to the feedback information. The more complex a system the more difficult it can become to identify appropriate inputs and outputs (inputs to the system) that can allow robust control. Essentially the key is to choose appropriate feedback and control variables that allow the controller enough information on the state of the system to make decisions and have appropriate control outputs to be able to steer the system in the desired direction.

The inputs, outputs and desired set points for a system are required before any development of the control system can begin. By selecting appropriate inputs, outputs and set points, it adds constraints to the control structure. If accurate information on the system states is not provided to the controller, it will be very difficult to create appropriate rule base or inference mechanisms. It therefore needs to be understood that the choice of inputs and outputs are a fundamentally important part of the fuzzy control design process.

If a Human expert can describe how best to control a system, then for a fuzzy logic controller we desire to take this linguistic description and embed it into the fuzzy controller. Linguistic variables describe each of the time varying fuzzy controller inputs or outputs. For example equations 3.1-3.3:

u_1 "Error" could describe $e(t) = r(t) - y(t)$ (3.1)

u_2 "Rate of error" could describe $\frac{d}{t}e(t)$ (3.2)

y_1 "Stepper motor degrees" $\emptyset(t)$ (3.3)

Where e(t) is the error between the set point(r) and the feedback (y), and (θ) is the set point degree of a step motor at time (t). The words with in quotation marks are the linguistic descriptions of the possible inputs and outputs. The number of words or style of description is purely to the discretion of the control designer and do not affect the controller performance. However, it is beneficial to use linguistic descriptions that are short but offer enough description to make sense to anyone reading. The linguistic variable notation is purely to help facilitate the construction of a fuzzy logic rules for the controller.

3.1.2 UNIVERSE OF DISCOURSE

Each linguistic variable has a domain, a range of real numbers that the linguistic variable can be equal too. A domain for a linguistic value is called a universe of discourse (135). In figure [3.1], the sets U_i and Y_i are called the universes of discourse with input linguistic variables u_i and y_i .

Universe of Discourse for linguistic variable "Error"
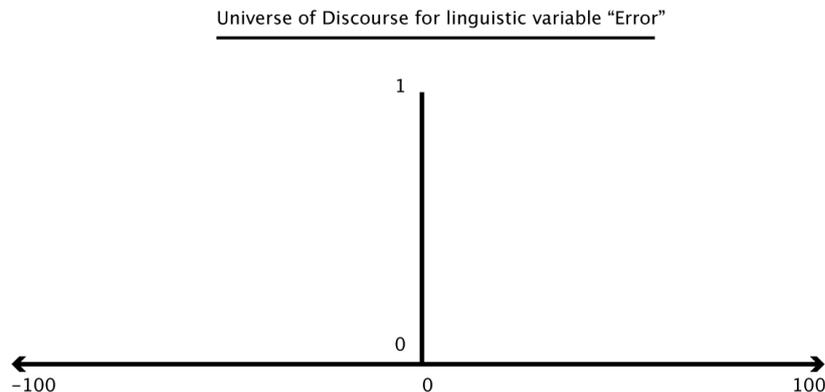
Figure 3.2 Example of a Universe of Discourse

For example, figure [3.2] could be the universe of discourse U_i for the linguistic variable "Error". The range in this case is -100 (α) , 100(β) where α and β can represent the points at the outmost membership of the universe of discourse [α, β]. The "Width" of the universe of discourse is $|\beta - \alpha|$.

3.1.3 LINGUISTIC VALUES

Linguistic variables (u_i, y_i)can change with time over a range of different values, the effective universe of discourse U_i [α, β], Y_i [α, β] respectively. The universe of discourse of a linguistic variable can be split into several linguistic values $\tilde{A}_i^j$, $\tilde{B}_i^j$ that separate the input space and describe characteristic of the variable.

$\tilde{A}_i^j$ denotes the jth linguistic value for the linguistic input variable u_i for the universe of discourse U_i.

$$\tilde{A}_i = \left\{ \tilde{A}_i^j : j = 1,2, \ldots, n_i \right\} \tag{3.4}$$

$\tilde{B}_i^j$ denotes the jth linguistic value for the linguistic output variable y_i for the universe of discourse Y_i.

$$\tilde{B}_i = \left\{ \tilde{B}_i^j : j = 1,2, \ldots, m_i \right\} \tag{3.5}$$

For example the linguistic input variable u_1 "error" could have the following linguistic values:

$\tilde{A}_1^1$ "Negative Large"

$\tilde{A}_1^2$ "Negative Small"

$\tilde{A}_1^3$ "Zero"

$\tilde{A}_1^4$ "Positive Small"

$\tilde{A}_1^5$ "Positive Large"

These linguistic values are short and are descriptively precise for a state of the Linguistic variable "error". However, the linguistic value is imprecise at describing the exact value of the linguistic variable, it is an abstract description. These accurately imprecise descriptions allow the input space to be separated into areas, each area a linguistic value.

3.1.4 MEMBERSHIP FUNCTIONS

Each linguistic value requires a function to map the crisp inputs into the various linguistic values. These functions are what are known as Membership Functions (MF). MF's describe the "certainty" that an element of U_i, denoted by u_i, with a linguistic description $\tilde{u}_i$, may be classified linguistically by $\tilde{A}_i^j$. MF's are individually specified form experience or intuition. The mathematical representation of the most common MF's can be seen in table [3.1].

Table 3.1 Mathematical representation of common fuzzy logic membership functions

Triangular MF specified by three parameters {a,b,c}. a < b < c, determine the x coordinates of the three corners of the triangle. Triangle (x; a, b, c)	$\begin{cases} 0, & x \le a. \\ \dfrac{x-a}{b-a}, & a \le x \le b. \\ \dfrac{c-x}{c-b}, & b \le x \le c. \\ 0, & c \le x. \end{cases}$ $\qquad(3.6)$	
Trapezoidal MF specified by four parameters {a,b,c,d}. a < b ≤ c < d, determine the x coordinates of the four corners of the trapezoidal. Trapezoid (x; a,b,c,d)	$\begin{cases} 0, & x \le a. \\ \dfrac{x-a}{b-a}, & a \le x \le b. \\ 1, & b \le x \le c. \\ \dfrac{d-x}{d-c}, & c \le x \le d. \\ 0, & d \le x. \end{cases}$ $\qquad(3.7)$	
Gaussian MF specified by two parameters {c,σ}. c represents the MF centre, σ determines the width. Gaussian(x; c,σ)	$e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2}$ $\qquad(3.8)$	
Generalised Bell MF specified by three parameters {a,b,c}. c represents the MF centre, a determines the width while b can be used to control the slope at the crossover point. Bell(x;a,b,c)	$\dfrac{1}{1+\left\lvert\dfrac{x-c}{a}\right\rvert^{2b}}$ $\qquad(3.9)$	
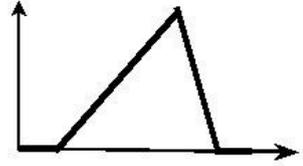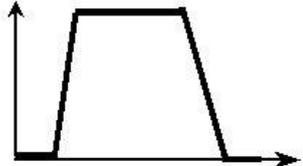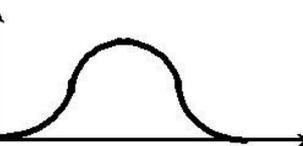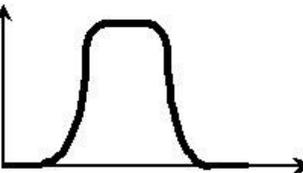
Due to the simple formula and computational efficiency of the triangular and trapezoidal MFs, they have been used extensively in real time implementations. However, as these MFs are composed of straight lines, they are not smooth at the corner points specified by the parameters. Where smoother relationships are required the continuous nonlinear function of Gaussian and bell MFs are better suited.
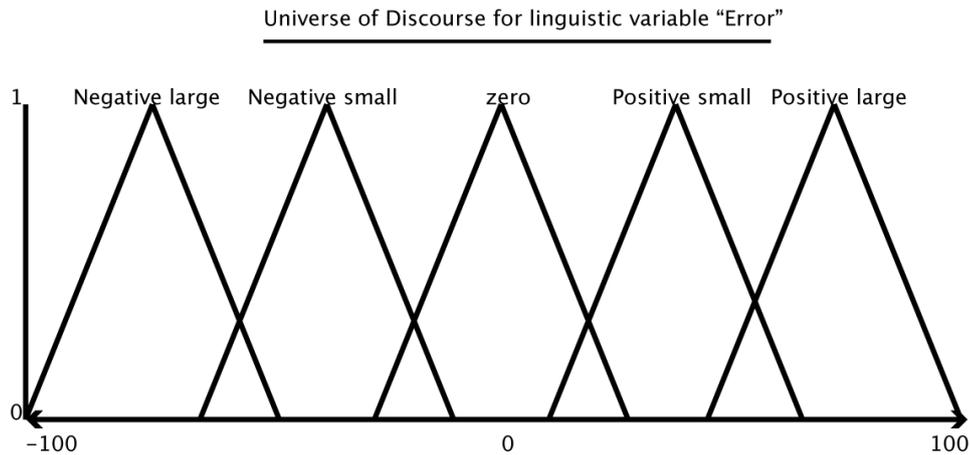
Figure 3.3 Example of a Universe of Discourse with membership functions

If we return back to the example for the universe of discourse for input "error" but include the linguistic variables with triangular MFs we get the following input mapping, figure [3.3]. This example shows that by using fuzzy logic for input interpretation the degree of membership can be over more than one linguistic value. If linguistic variable "error" activates linguistic value "zero", the value can be numerically = to zero, but it can also be some value close to zero and potentially have a degree of membership to "negative small" or " positive small". This is a key difference between classic logic and fuzzy logic. The inputs can have degree of membership 0 and 1 but also any value between them.

3.1.5 LINGUISTIC RULES

FLC is constructed around a set of linguistic rules that facilitates mapping between the inputs and the outputs. Linguistic rules are determined by an expert on the system. The linguistic rules are in the form:

IF premise **THEN** consequence

The inputs of the fuzzy system are normally associated to the premise part of the equation while the outputs of the system associated to the consequence. Two standard forms for the IF-THEN rules are MIMO and MISO.

The MISO form of the rule is as follows:

$$IF\ \tilde{u}_1\ is\ \tilde{A}_1^j\ AND\ \tilde{u}_2\ is\ \tilde{A}_2^k\ AND, \dots , AND\ \tilde{u}_n\ is\ \tilde{A}_n^l\ THEN\ \tilde{y}_m\ is\ B_m^j \qquad (3.10)$$

The MIMO form of the rule is as follows:

$IF\ \tilde{u}_1\ is\ \tilde{A}_1^j\ AND\ \tilde{u}_2\ is\ \tilde{A}_2^k\ AND, \dots\ , AND\ \tilde{u}_n\ is\ \tilde{A}_n^l\ THEN\ \tilde{y}_1\ is\ B_1^p\ AND\ \tilde{y}_2\ is\ B_2^q\ AND, \dots , AND\ \tilde{y}_m\ is\ B_m^r$

$$(3.11)$$

The MIMO form of the rule can also be represented by several MISO rules:

$$IF\ \tilde{u}_1\ is\ \tilde{A}_1^j\ AND\ \tilde{u}_2\ is\ \tilde{A}_2^k\ AND, \dots\ , AND\ \tilde{u}_n\ is\ \tilde{A}_n^l\ THEN\ \tilde{y}_1\ is\ B_1^p$$

$$IF\ \tilde{u}_1\ is\ \tilde{A}_1^j\ AND\ \tilde{u}_2\ is\ \tilde{A}_2^k\ AND, \dots\ , AND\ \tilde{u}_n\ is\ \tilde{A}_n^l\ THEN\ \tilde{y}_2\ is\ B_2^q$$

-

$$IF\ \tilde{u}_1\ is\ \tilde{A}_1^j\ AND\ \tilde{u}_2\ is\ \tilde{A}_2^k\ AND, \dots\ , AND\ \tilde{u}_n\ is\ \tilde{A}_n^l\ THEN\ \ \tilde{y}_m\ is\ B_m^r \quad (3.12)$$

This stands as the "and" logic in the MIMO rules is still represented if all the MISO rules activate. This allows MIMO systems to be represented by multiple fuzzy systems if required.

When constructing rules for a system not all the linguistic terms have to be used, only variables of significant effect need to be included into rules and is down to the system and the expert knowledge which linguistic terms are used or not used.

3.1.6 FUZZY INFERENCE SYSTEMS

Fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy logic. The mapping then provides a basis from which decisions can be made, or patterns discerned. The process of fuzzy inference involves all of the pieces that are described in the previous sections.

The inference mechanism has two basic tasks:

1. Determine the extent to which each rule is relevant to the current situation as characterised by the inputs.

2. Drawing conclusions using the current inputs and the information that is stored in the rule base.

There are two main fuzzy inference systems. The difference between these systems lies in the consequence of their fuzzy rule base and therefore the methods used for aggregation and defuzzification of the outputs. The Mamdani method (135), is the most commonly used in applications, this is due to its simple structure of 'min- max' operations. These operators are required if there is more than one premise per rule and there are more than one rule. The

choice of operation would depend on the system being controlled and how the expert would want the rules interpreted. The resulting membership value determines the level of activation of a fuzzy set in the output universe of discourse. If there are multiple rules then there also needs to be a method of combining all the active MFs together for each output. The process of combining all the active output from multiple rules is called aggregation. Some of the most commonly used aggregation operators are the maximum, the sum and the probabilistic sum. The purpose of aggregation is to combine all the active MFs in a desirable method to produce a single fuzzy set.

The Sugeno fuzzy system (135), is the second main fuzzy inference system. The Sugeno fuzzy method was derived in an effort to develop a systematic approach to generating fuzzy rules from a given input output data set A typical fuzzy rule in a Sugeno fuzzy system is in the from:

If X is A and Y is B then Z =f(x, y)   (3.13)

Where A and B are the fuzzy sets in the premise, while z=f(x,y) is a crisp function in the consequence. Usually f(x,y) is a polynomial in the input variables x and y, but it can be any function as long as it can appropriately describe the output of the model within the fuzzy region specified by the antecedent of the rule.

If the function is of first order polynomial, then the system would be described as a first order Sugeno fuzzy model. If the function is a constant, then the resulting model would be described as a zero order Sugeno fuzzy model. This can be viewed as a special case of mandani fuzzy system, in which each rule's consequence is specified by a fuzzy singleton (or a pre defuzzified consequence).

The Sugeno fuzzy system can be more compact and computational efficient then the Mamdani method. The Sugeno method also lends itself well to integration with ANN architecture, due to the potential of having pre defuzzified outputs. However the Mandani method has an intuitive structure and widespread acceptance due to its ability of automating human knowledge.

3.1.7 DEFUZZIFICATION

The output of the fuzzy system is often required to be a crisp value (a number) and not a fuzzy set. Therefore the aggregated fuzzy set needs to be transformed into a single numerical value. There are numerous methods for defuzzification (135), the most popular method is the Centroid Of Area (COA), equation (3.14) and is commonly used with the Mamdani fuzzy models.

$$Y_{COA} = \frac{\int \mu_i(y) y \, dy}{\int \mu_i(y) dy}$$ (3.14)

Where Y_{COA} is the defuzzified output, $\mu_i(y)$ is the aggregated MF and y is the output variable.

3.2 NEURAL NETWORKS

ANNs are networks of artificial neurons that are conceptual models of the biological neurons found in the human brain. Neurons are capable of learning patterns between input signals and desired output signals.
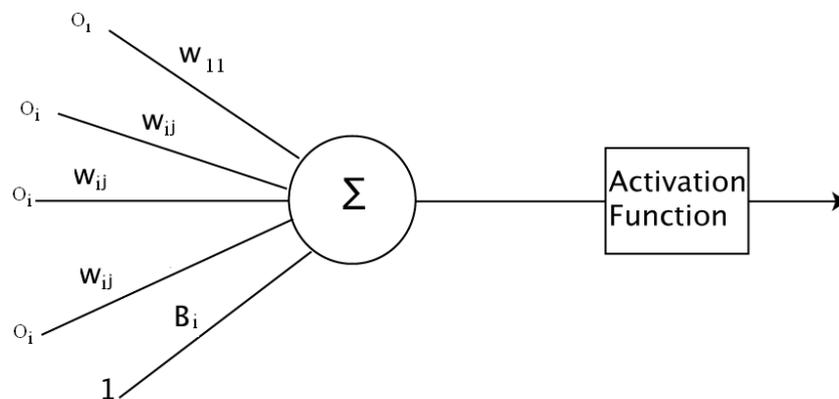
Figure 3.4 Example of an artificial neuron

Figure [3.4] shows an example of an artificial neuron. An artificial neuron is built up of several elements;

1. Activation function
2. Weights
3. Bias

The artificial neuron can receive one of more inputs; this represents the dendrites in a human neuron. The sum of all the inputs is used as part of the output of the neuron, this is like the synapse In a human neuron. Each input to the sum is normally weighted, and the sum of all inputs is passed to an activation function. The output of the activation function will depend on the level of activation.

3.2.1 ACTIVATION FUNCTIONS

Activation functions can be linear or nonlinear functions, there are many different types; the most popular transfer functions are listed in table [3.2].

Table 3.2 Mathematical representation of common neural network activation functions

Activation Function	Equation	Diagram
Linear	$Purelin\,(n) = \; n$ (3.15)	 $a = purelin(n)$ Linear Transfer Function
Saturated linear	$Satlin(n) = 0, if\; n \le 0$ $n, if\; 0\; \le n \le 1$ $1, if\; 1 <= n$ (3.16)	 $a = satlin(n)$ Satlin Transfer Function
Sigmoid	$Logsig(n) = \dfrac{1}{\left(1 + exp(-n)\right)}$ (3.17)	 $a = logsig(n)$ Log-Sigmoid Transfer Function
tan sigmoid	$Tansig(n)$ $= \dfrac{2}{\left(1 + exp(-2 \cdot n)\right)^{-1}}$ (3.18)	 $a = tansig(n)$ Tan-Sigmoid Transfer Function
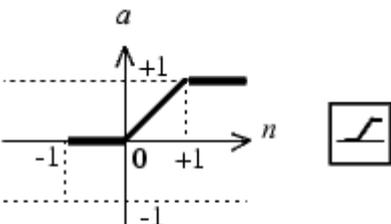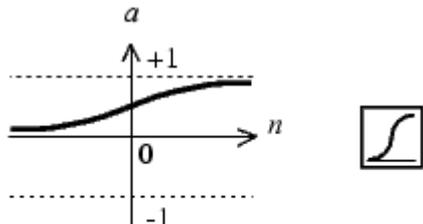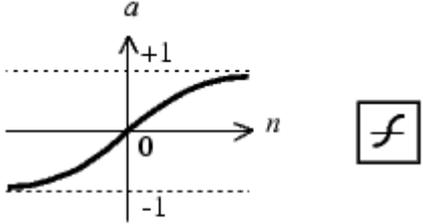
3.2.2 MULTILAYER FEED-FORWARD NEURAL NETWORK
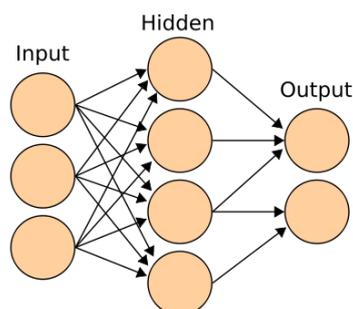
Figure 3.5 Example of a multi-later neural network

The multi-layer feed-forward ANN or multi-layer perceptron (MLP), is one of the simplest multi-layer ANN, figure [3.5]. MLP networks are general-purpose, flexible, nonlinear models consisting of a number of units organised into multiple layers. The complexity of the MLP network can be changed by varying the number of layers and the number of units in each layer. Given enough hidden units and enough data, it has been shown that MLPs can approximate virtually any function to any degree of accuracy (70). In other words, MLPs are universal approximates. MLPs are valuable tools in problems when there is little or no knowledge about the form of the relationship between a system's inputs and their corresponding outputs.
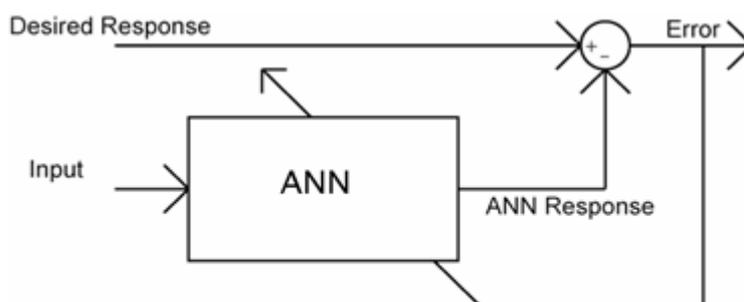
Figure 3.6 Example of neural network training

The MLP ANN is trained by supervised learning using the iterative back-propagation algorithm, figure [3.6]. In the learning phase a set of input patterns, called the training set, are presented at the input layer as feature vectors, together with their corresponding desired output pattern which usually represents the classification for the input pattern. Beginning

with small random weights, for each input pattern the ANN is required to adjust the weights attached to the connections so that the difference between the ANN's output and the desired output for that input pattern is decreased.

A typical ANN might have between ten and hundred weighs whose values must be found to produce an optimal solution. If ANNs were linear models like linear regression, it would be a relatively simple task to find the optimal set of weights. But the output of an ANN as a function of the inputs is often highly nonlinear; this makes the optimization process complex.

3.2.3 BACK PROPAGATION
Back propagation is a method for performing supervised learning of ANNs. Back propagation adapts ANNs so that their actual output comes close to a target output from a training set of data. The main goal through training is to adapt the ANNs parameters to obtain a generalized behaviour of patterns so that it can perform well for situation outside that of the training data. The learning problem is to find the optimal combination of weights and bias terms so that the ANN approximates a given function as closely as possible. However we often do not know the function explicitly but only implicitly through input/output data.

3.2.3.1 Learning using Back Propagation
To reduce the error function of a ANN, all the weights in the ANN are adjusted one at a time. If we consider a single weight, w_{11}, this weight can be considered as an input channel into a sub network made of all paths starting with weights w_{ij} and ending in the single output unit of the sub network, figure [3.4]. The information fed into this sub network in the feed-forward step would be $o_1 w_{11}$ where o_1 is the stored output of a previous unit "1". The back propagation step computes the gradient of E with respect to this input,

$$\frac{\partial E}{\partial o_i w_{ij}}$$
(3.19)

Since in the back propagation step o_i is treated as constant, this leads to:

$$\frac{\partial E}{\partial w_{ij}} = o_i \frac{\partial E}{\partial o_i w_{ij}}$$
(3.20)

All the sub network defined by each weight of the network can be handled simultaneously, but at each node "i" the following needs to be stored:

1. The output o_i of the node in the feed forward step
2. The cumulative result of the backward computation on the back propagation step up to this node, known as the back propagation error

The back propagation error at the j-th node can be expressed by δ_j, the partial derivative of E with respect to w_{ij} can therefore then be expressed by equation (3.21).

$$\frac{\partial E}{\partial w_{ij}} = o_i \delta_i \qquad (3.21)$$

Once all the partial derivatives have been computed, the gradient decent can be performed by adjusting each weight w_{ij} increment with equation (3.22).

$$\Delta w_{ij} = -\gamma o_i \delta_i \qquad (3.22)$$

Where $-\gamma$ is the learning rate.

This incremental correction step transforms the back propagation method into a learning method for ANN.

It is important to make the corrections to the weights only after the back propagated error has been computed for all units in the network. If the weights are updated before all partial derivatives are calculated the corrections become mixed with the back propagation of the error and the corrections will not follow the negative gradient direction.

If there are more than one set of training data. The process of extending the network to calculate the error function is undertaken for each input-output pattern separately. The weight corrections are then calculated from each pattern error. Therefore the weight update becomes equation [3.23].

$$\Delta w_{ij}^{(1)} = \Delta 1 w_{ij}^{(1)} + \Delta 2 w_{ij}^{(1)} + \Delta 3 w_{ij}^{(1)} + \cdots + \Delta p w_{ij}^{(1)}, for\ input\ patterns = 1, \dots, p, \qquad (3.23)$$

This is known as batch or off-line updates when the weights are updated in this format. When weights are updated sequentially after each pattern is presented to the ANN, this is known as on-line training.

In on-line training the corrections do not exactly follow the negative gradient direction, but as long as the training patterns are chosen at random the search direction should oscillate around the exact gradient direction and on average forms a decent on the error term.

Using the on-line training methods can be advantageous as by adding some noise to the gradient direction, i .e the oscillation, it can help avoid the algorithm being stuck in shallow local minima. When the training set contains thousands of training patterns, it can be very computationally expensive to calculate the gradient direction as each epoch (one round of presentation of all data patterns) will have many feed forward passes and on-line training becomes more efficient.

For more detailed description of the application and procedure of the back propagation algorithm on multi-layer ANNs, please see reference (136).

The back propagation learning method forms the basis of most other learning mechanisms. The problem with the standard back propagation learning method is that it can be susceptible to finding local minima and not the global minimum of the entire error function. The following learning methods are some of the more popular modifications of the standard back propagation learning method to try and avoid the problems of local minima.

3.2.3.2 Gradient Decent with Adaptive Learning Rule Back Propagation

Global Minimum
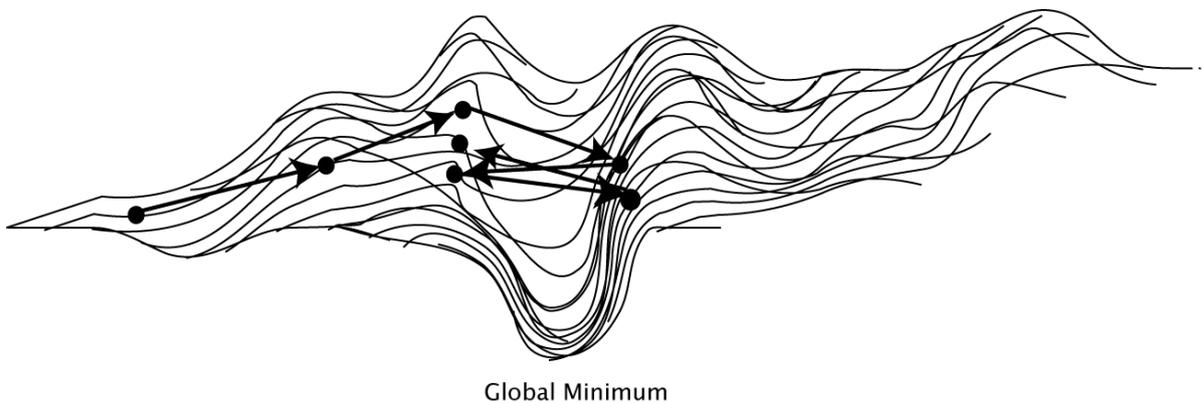
Figure 3.7 Example of back propagation with large learning rate oscillating around minimum

With the standard back propagation algorithm described above, the learning rate γ, is held constant throughout training. The performance of the method is very sensitive to the appropriate setting of the learning rate. If the learning rate is set too high, the algorithm can oscillate around the minimum of the error function and become unstable, figure [3.7].
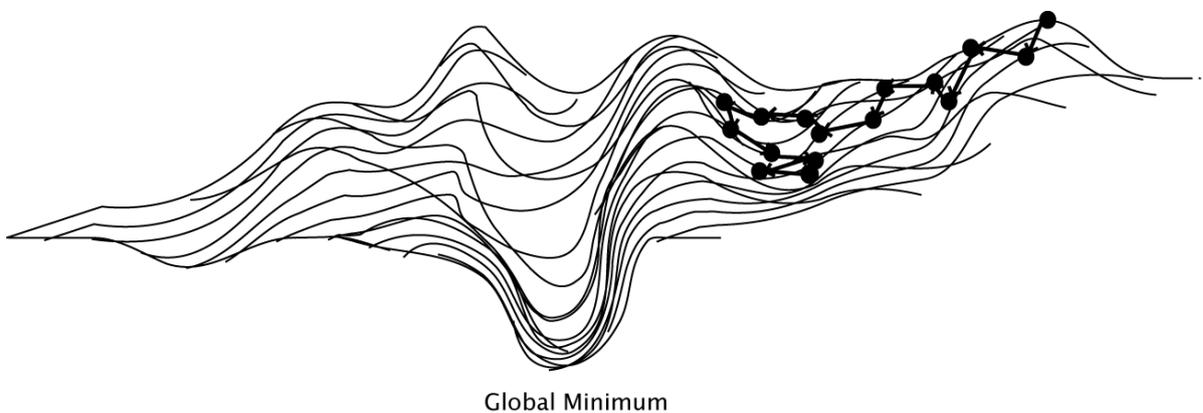
Global Minimum

Figure 3.8 Example of back propagation with small learning rate becoming stuck in local minimum

However, if the learning rate is too small, the algorithm can take a long time to converge to the minimum of the error function, figure [3.8]. It is difficult to determine an optimal setting for the learning rate before training, and, in fact, the optimal learning rate changes during the training process, as the algorithm moves across the performance surface. When the learning rate is fixed it needs to be set at a value that can offer a balance between speed of convergence and ability to find the global minimum.

Global Minimum
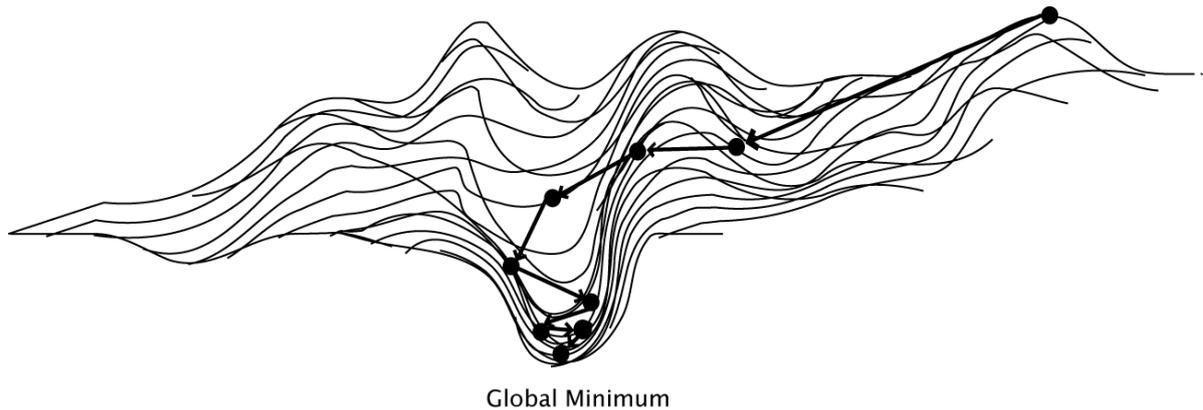
Figure 3.9 Example of back propagation with variable learning rate

The performance can be significantly improved if the learning rate can change during the training process. By using an adaptive learning rate, it offers the potential to keep the learning step size as large as possible while keeping learning stable, figure [3.9]. The learning rate is made responsive to the complexity of the local error surface.

An adaptive learning rate requires some changes in the training procedure. First, the initial ANN output and error are calculated. At each epoch new weights and biases are calculated using the current learning rate. New outputs and errors are then calculated. If the new error exceeds the old error by more than a predefined ratio, (typically 1.04), the new weights and biases are discarded. In addition, the learning rate is decreased (typically by multiplying by 0.7). Otherwise, the new weights, etc., are kept. If the new error is less than the old error, the learning rate is increased (typically by multiplying by 1.05).

This procedure increases the learning rate, but only to the extent that the ANN can learn without large error increases. Thus, a near-optimal learning rate is obtained for the local terrain, although a larger learning rate could still result in stable learning.

3.2.3.3 Gradient Decent with Momentum Back Propagation
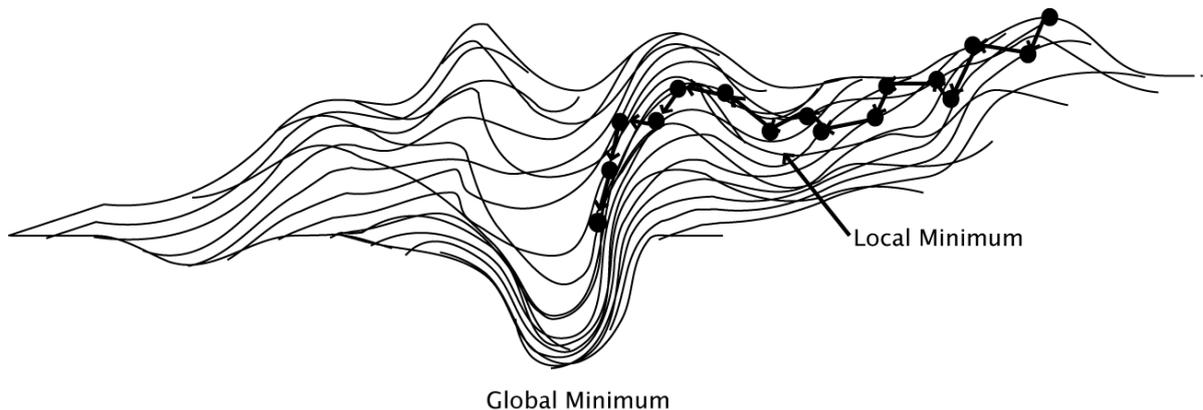
Figure 3.10 Example of back propagation with momentum

Gradient descent with momentum, allows a ANN to respond not only to the local gradient, but also to recent trends in the error surface. Acting like a low pass filter, momentum allows the ANN to ignore small features in the error surface, figure [3.10]. Without momentum an ANN can get stuck in a shallow local minimum. With momentum an ANN can slide through such a minimum. Gradient descent with momentum depends on two training parameters. The Learning Rate (LR), similar to the simple gradient descent and the Momentum Constant (MC), that defines the amount of momentum. MC is set between 0, no momentum and 1, large momentum. A momentum constant of 1 results in a ANN that is completely insensitive to the local gradient and, therefore, does not learn properly.

3.2.3.4 Quasi-Newton Back Propagation

Quasi-Newton methods are based on Newton's method to find the stationary point of a function, where the gradient is equal to zero. Newton's method assumes that the function can be locally approximated as a quadratic in the region around the optimum, and use the first and second derivatives (gradient and curvature) to find the stationary point.

Newton's method often converges faster than conjugate gradient methods due to the added curvature information allowing better determination between local and global minimum. Unfortunately, the curvature is calculated through the Hessian matrix and is complex and expensive to compute for feed forward ANNs.

The Quasi-Newton method does not require calculation of the Hessian matrix. Instead the Hessian matrix is approximated from the function and gradient values from previous steps. The method will initiate with gradient decent, but as the function is explored a better

approximation of the curvature can be estimated, allowing the method to determine future steps on gradient and curvature.

3.2.3.5 Levenberg-Marquardt Back Propagation

The Levenberg-Marquardt algorithm was also developed to allow for fast convergence without having to compute the complex Hessian matrix. The Levenberg-Marquardt approximates the Hessian matrix with a Jacobian matrix. The Jacobian matrix can be used to approximate the Hessian matrix when the performance function is in the form of a sum of squares. This is typical for feed forward ANNs. The elements of the Jacobian matrix can be calculated through a standard back propagation method that is much less computationally intense as computing the Hessian matrix.

Levenberg-Marquardt method is a blend of the gradient decent and Newton method to combine their advantages while reducing disadvantages. The curvature calculated through the approximate Hessian matrix is used to scale the gradient decent. Large steps are taken in directions of low curvature and small steps taken in directions with high curvature. The resulting training reduces convergence problems associated with error valley or nonlinearity of start location. It also facilitates variable step size for reducing problems with skipping out of global minimum or getting stuck in a local minimum (143).

3.2.3.6 Dynamic Back Propagation
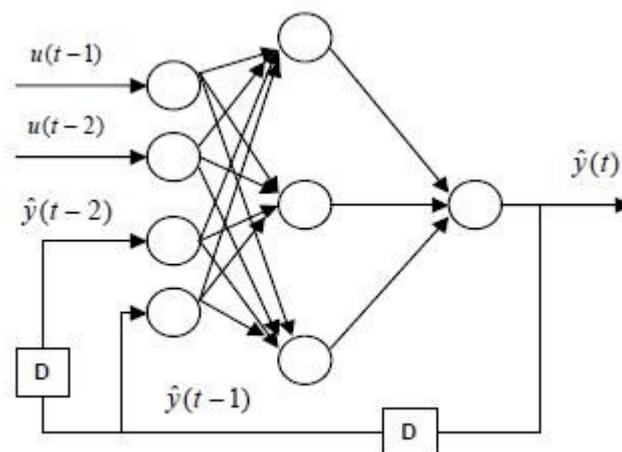
Figure 3.11 Example of Dynamic Feedback Neural Network

So far discussed is the theory behind back propagation for linear feed-forward structured ANNs. Dynamic ANNs differ from the linear as they contain both feed-forward and feedback connections between the neural layers, figure [3.11].

The back propagation method used for dynamic ANNs is often referred to as back propagation through time. This is referred to this way as the feed forward and feedback loops are unfolded to create a feed forward network that represents the dynamic ANN. Once it is in this state it can be adjusted using the same principles as a standard feed forward ANN. This training however is an offline technique. Training online is possible using an alternative technique called real time recurrent learning. This type of learning updates the weights immediately after seeing each data point. The online learning is often much faster, it can be used when there is no fixed training set, where new data keeps arriving. On-line training is better at teaching problems that have non stationary environments, where what would be considered the best model changes over time.

However for dynamic ANN, the training is much more computationally expensive. For online training this can limit the applicable training algorithms to basic gradient descent.

3.2.4 DYNAMIC NEURAL NETWORKS

The feed-forward ANN's (FFNN) have all forward passing links between neurons. This offers a simple design this is fast to train and can accurately model nonlinear relationships between many variables. However, these relationships need to be in the form of states. The feed forward ANN's have limited capability of learning how a system moves from one state to another, i.e. the dynamics between states. Often this information is not required and the FFNN should be the first choice of a relationship modelling application.

By introducing time delayed inputs as well as current inputs into a feed-forward ANN we create a new network called the Feed-Forward Time Delay (FFTD) ANN. This FFTD structure is the first step in introducing dynamics into the static ANN architecture; all the dynamics are limited to the input layer of the ANN. All links are still forward passing allowing the standard back propagation style training to be used. However, the ANN output can be based not only on current state but on the previous time step state. This type of network dynamics allows the ANN to have information regarding magnitude, direction and rate of change between input states. This type of ANN is well suited to time series prediction of the next output state from the relationship between previous input states. This allows this type of ANN to be used either as a form of control or a form of prediction.

 By introducing a feedback loop from a FFTD ANN outputs to a new inputs of the ANN we create a recurrent dynamic network. A commonly used recurrent ANN used is called the Nonlinear AutoRegressive network with eXogenous inputs (NARX). In a recurrent ANN the output signals are dependent on previous output signals as well as previous values of input signals and current inputs. This allows the ANN to know the same information as a FFTD

network but also the added information of magnitude, direction and rate of change of previous outputs and how these relate to previous inputs signals. Due to this added level of dynamics there are many applications for the NARX network. It can be used as a predictor, to predict the next value of series. It can also be used for nonlinear filtering, in which the target output is a noise-free version of the input signal or the modelling of nonlinear dynamic systems and control. However, as the network structure incorporates time delay inputs from output signals the training problem becomes much more complex. The ANN cannot use normal training algorithms and the ANN requires unfolding with time and back propagation through time methods. This is a much more computational expensive training procedure.

3.2.4.1 Network Application Choice

The typical application for each of the types of ANN structures can be seen in table [3.3]. For any ANN application it is often advised that a simple feed forward ANN architecture should be investigated first, if this fails to meet the requirements, then look toward the dynamic ANN architectures.

Table 3.3 Typical application choice for the different neural network structure types

Neural Networks	Application
Feed Forward ANNs	Mapping of nonlinear relationships between Multiple inputs and outputs states. i.e. if IN(1) is Y and IN(2) is Z then OUT(1) is J and OUT(2) is C This type of network can be thought of as a map of states.
Time Delay ANNs	Mapping of nonlinear relationships between Multiple inputs and outputs states. But also the relationships between the dynamics of the rate of change, magnitude and direction of input states to the output state. I.e. If IN(1) was Z and still Z but IN(2) was Q but now R then OUT(1) is F ... etc This type of network can be thought of a map of static states with knowledge of the dynamics between the input states allows future predictions of the output state depending on changes in input states.
Recurrent ANNs	Mapping of nonlinear relationships between Multiple inputs and outputs states. But also the relationships between the dynamics of the rate of change, magnitude and direction of previous and current input and output signals on the choice of the next output signal. i.e. If IN(1) was Z and still Z but IN(2) was Q but now R and OUT(1) was F and now K then OUT(1) is F This type of network can be thought of a map of static states with knowledge of the dynamics between the input states and outputs. Allowing for output prediction to alter depending on the effect of previous outputs on the inputs dynamics, vital for direct control applications.

4. IONISATION FOR COMBUSTION FEEDBACK

This research main focus is the development of control architecture for advanced combustion control. Ionisation can potentially be a key component to the overall control structure offering an inexpensive in-cylinder feedback technique. For the development of the control it would be useful to develop a mobile universal ionisation sensing system. If an ionisation system can be developed that can offer ionisation feedback from the firing spark plug or dedicated sensor it would allow for a system that could be applied to a number of different engines. However, there are a number of key concerns and design considerations with developing such a research tool.

4.1 DESIGN CONSIDERATIONS

The basic automotive SI engine ignition system is built up of several components; these can include spark plugs, High Tension (HT) leads, coil and a control unit for triggering and distribution. In the past, control of the spark timing was achieved through a mechanical system comprising of a rotor arm driven from the engine and a distributor cap. This is an old design that was susceptible to error from numerous sources including excessive wear, corrosion, degrading of electrical conductivity and water intrusion. The mechanical control system was phased out by the multi-coil systems. The multi-coil system is electronically controlled, rather than mechanical; it offers greater precision on timing control as well as maintaining spark quality over the life span of the components. The basic electronic ignition system is as follows, figure [4.1].
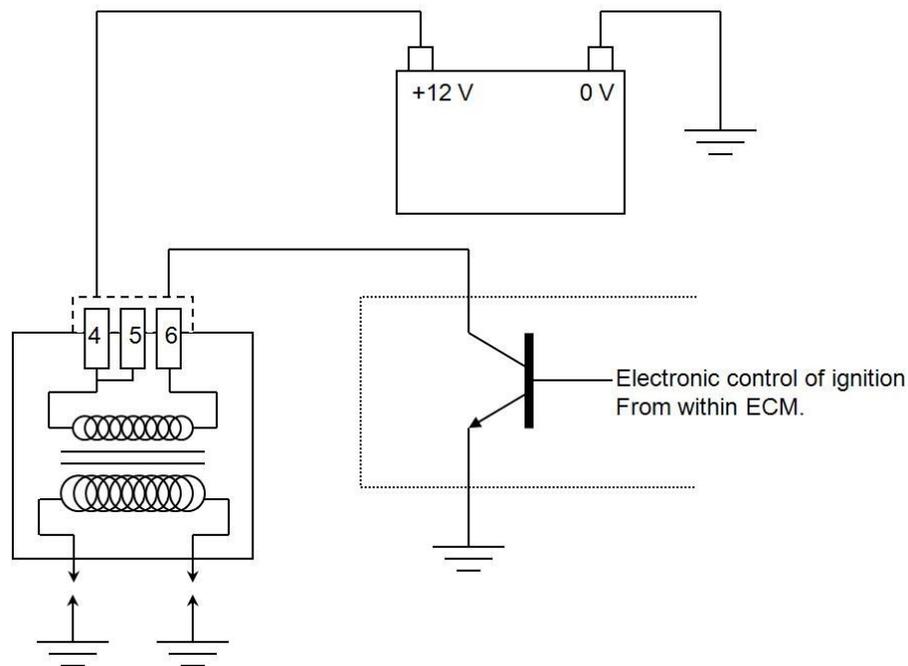
Figure 4.1 Basic electronic ignition system with double ended coil

Originally the multi core ignition system was incorporated in wasted spark implementations where each coil is used to drive two spark plugs. This setup relied on HT leads to route the high voltage to the spark plugs. The development of smaller resin filled coils allowed for a compact coil to be placed over the spark plug for each cylinder. The "coil on plug" system offers a more efficient method of controlling spark timing. With the coil being directly on the plug there are less losses through HT leads, allowing for a more reliable and stronger spark. Each coil can also be controlled individually allowing for more accurate control and no wasted spark, therefore increasing the life span of the components.

To obtain an ionisation signal from a spark plug we need the capability of supplying a voltage continuously across the gap. With the electronic ignition systems discussed this is not possible without modifications. Therefore, there needs to be a consideration in the design of the mobile ionisation sensor that would allow the system to be applied to a number of engines without requiring modification to the original ignition control system. For the ionisation system to be unobtrusive to the original control system and have a simple plug in application it would ideally require its own internal coil. By having an ignition coil integrated into the design, the system could be configured to fire the spark plug when given the control signal from the ECU. Therefore, no modification is required to the existing ignition system to gain ionisation feedback from a firing spark plug.

The ionisation circuit operates by applying a voltage across the spark plug, as ions flow across the spark plug gap this creates a voltage drop in the system. The change in voltage can

be measured and amplified into an output signal. This works well when the voltage across the sparkplug is in the magnitude of hundreds of volts. When the spark plug is fired this causes voltages in the magnitude or tens of thousands of volts. This high voltage has the potential to destroy the ion sensor circuit and is a major reason why separate probes are often preferred. Circuit design would have to protect the amplifier from the high voltage while allowing feedback during the rest of the cycle.

Spark plugs for different engines have a range of designs for, material, size and gap width depending on engine operation and conditions. An ionisation voltage that works on one spark plug will not necessarily work for another spark plug. Another design consideration would be for a variable voltage that could be applied for measuring ionisation. This would allow the portable system to have a degree of calibration when applied to different engines.
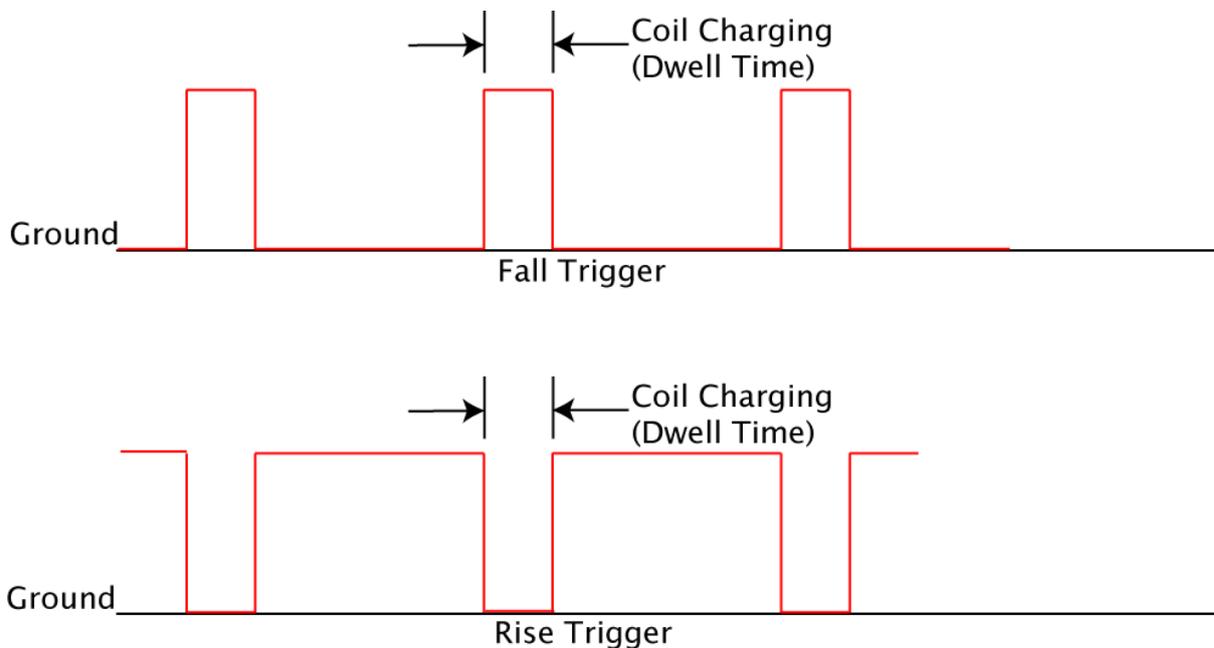
Figure 4.2 Rising and Falling ignition trigger signals

Different control systems can also use different trigger signals. Some engine manufactures use a falling edge trigger signal, some use a rising edge, figure [4.2]. If the system is only designed for one and attached to a vehicle with the opposite type of trigger it will either; not work, have incorrect timing or damage components. The design of the ionisation circuit needs to be able to operate on both types of trigger signals if the system is to be used on multiple engines.

(removed for copywrite)

Figure 4.3 Ignition coil secondary winding voltage

Ignition coils are primarily designed for their original purpose of creating a high voltage for firing the spark plug. There are some side effects when trying to sense ionisation through the secondary side of the ignition coil for a firing plug. The major concern is coil oscillation, figure [4.3]. The oscillation of the coil can completely cover any ionisation signal taken from the spark plug as the magnitude of the oscillation is significantly higher than that created from the change in voltage due to ionisation. A consideration is needed into the choice of coil used with the ionisation system. Ideally a small coil would minimize the ringing on the secondary winding.

Overall there are several design considerations that need addressing in the development of the mobile ionisation sensor. To summaries these are:

- Requirement for integrated ignition coil.
- Choice of coil to limit coil oscillation.
- Circuit protection for amplifier against high voltage from coil.
- Adjustable ionisation voltage.
- Ability to operate with different ignition trigger signals.

4.2 DEVELOPMENT OF MOBILE UNIVERSAL IONISATION SYSTEM

4.2.1 VARIABLE VOLTAGE IONISATION SIGNAL AMPLIFIER CIRCUIT

The magnitude of the voltage across the spark plug can be very high. It is therefore important in the design of the sensor circuit to protect the components and amplifier from this high voltage.
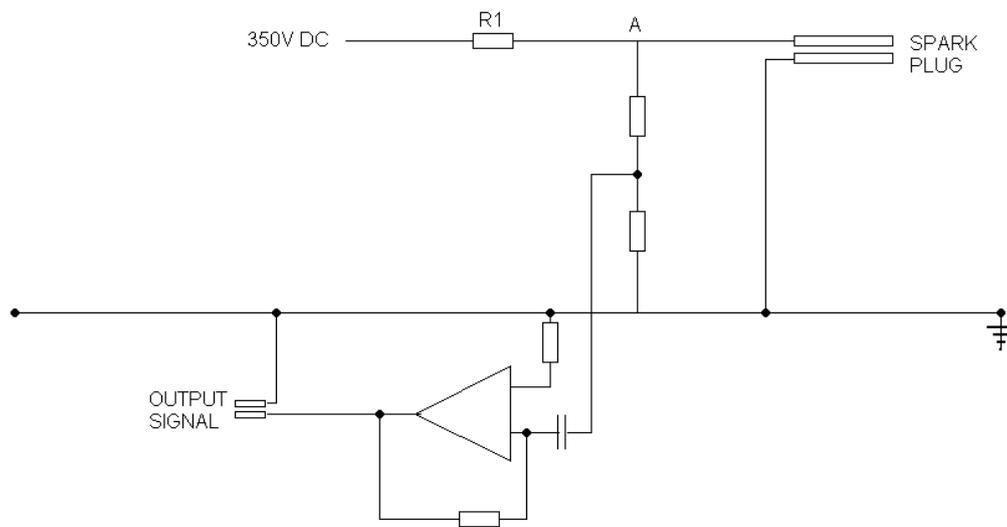
Figure 4.4 Simplified ionisation sensor circuit

The sensor circuit is shown in figure [4.4], a continuous ionisation voltage is applied across the spark plug. Depending on the concentration of ions in the cylinder the current flow between the sparkplug gap will change proportionally. This change in current will cause a voltage drop across Resistor "R1" that is proportional to the rate of change of ionisation. As this voltage can be high it is useful to divide this voltage from point "A" through resistors before it reaches the amplifier. The continuous voltage that is applied across the spark plug can be adjusted to alter the strength of ionisation. The ionisation voltage is created from a transformer. The primary voltage is adjustable through a variable resistor. This in turn allows control over the secondary voltage, between 100 and 400 volts.

4.2.2 CHOICE OF COIL

The ignition coil is a key component in the ionisation system if it is to operate with a firing spark plug. To be able to use a coil for firing and measuring ionisation, the coil needs to be of specific design .The secondary and primary windings must be isolated, figure [4.1].
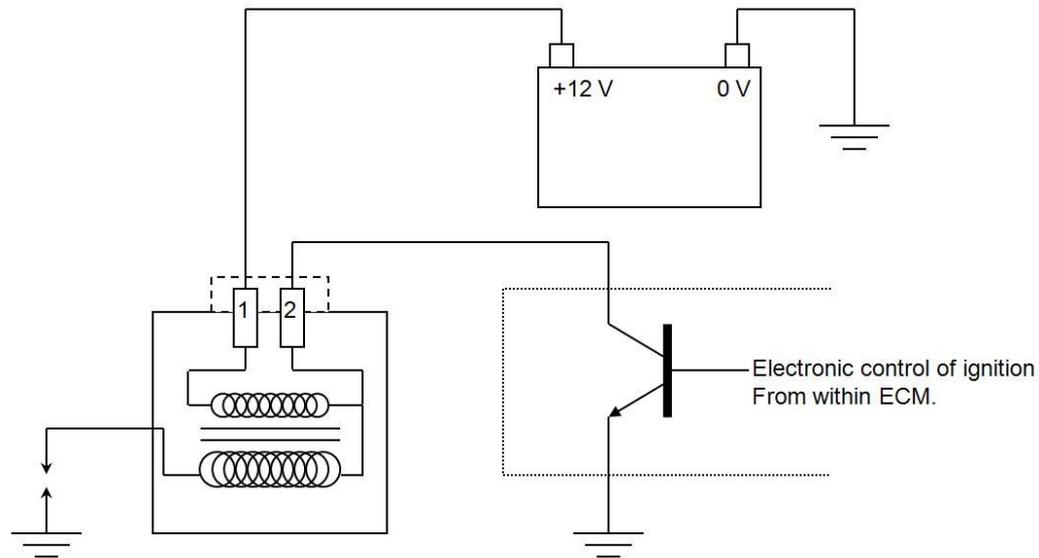
Figure 4.5 Basic electronic ignition system with single ended coil

In a number of coils the windings are connected, figure [4.5]. If this is the case the coil is unusable for ionisation as the secondary winding is linked to ground. The wasted spark type coils are of the type to have isolated primary and secondary windings. However, as they are used to fire two spark plugs simultaneously they are often large, giving excessive coil oscillation. A coil on plug type is a much smaller coil, they can also be found with isolated windings. Because the coil is much smaller and optimized for firing a plug without the losses of the HT lead it has a lower electrical capacitance reducing the coil oscillation problem. For this reason a coil on plug type coil was used in the ionisation circuit design.

4.2.3 CIRCUIT FOR COIL OPERATION AND TRIGGER SIGNALS

Circuit for rising edge and falling edge triggering systems
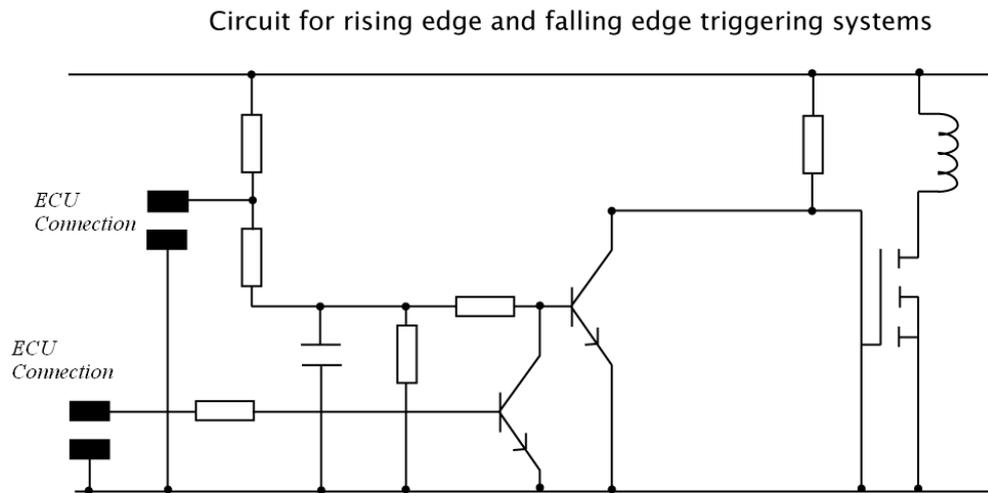
Figure 4.6 Ionisation system circuit for Coil operation with rising or falling trigger signals

Figure [4.6] shows the circuit designed to operate the coil with either rising or falling trigger signals. From figure [4.6], it is possible to see two possible ECU connections. Depending if the ECU trigger signal is rising or falling type, the ECU will need to be connected to one of these points. The coil is located in the top right corner, with the primary coil represented by an inductor symbol. The coil is connected to ground through a MOSFET. The MOSFET in this structure operates in the form of a switching unit that is controlled by the level of current supplied. Through the combination of transistors, it is possible to alter the level of current applied to the MOSFET and therefore connect and disconnect the primary coil from ground. To give an explanation of how the above circuit can operate with the two different trigger signals it is easiest to break it down and show how each signal is dealt with individually.

Circuit for firing on falling edge trigger
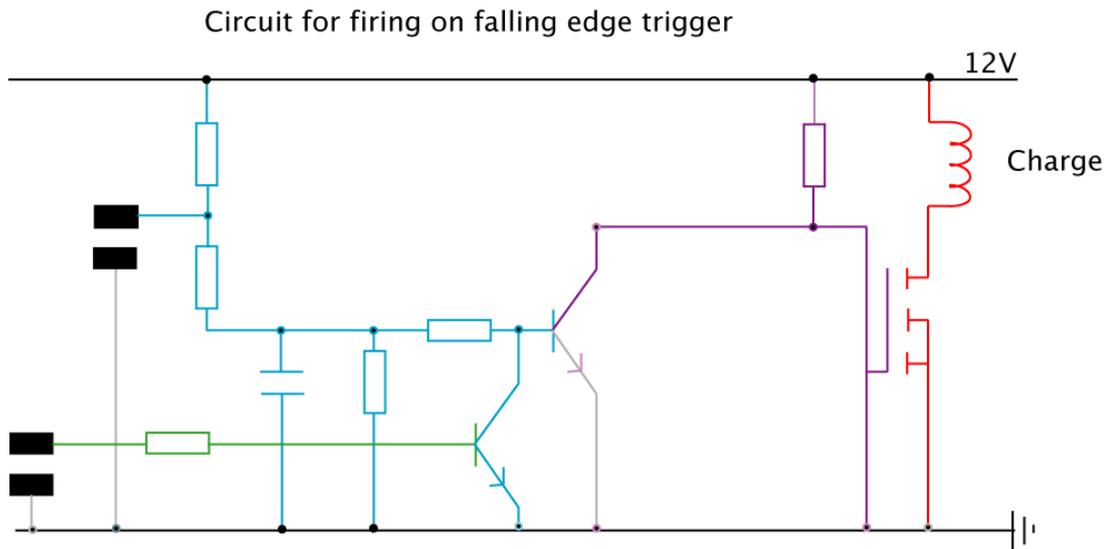
Figure 4.7 Ionisation system circuit for Coil operation with falling trigger during coil charge

For the falling edge trigger signal the ECU would require connecting to the lower connection. A switch on the main unit allows the other connection to be deactivated and 12 volts to be continually connected to that section of the circuit, depicted in light blue. For the falling edge trigger, the coil is to charge during the positive step as seen in figure [4.7]. During this time a positive signal is received from the ECU and is depicted in green in figure [4.7]. Applying this positive voltage allows a current to be applied to the connected transistor. This allows the transistor to conduct and allow current to sink to ground from the light blue line above. Sinking the current to ground reduces the current seen at the second Transistor. This prevents the transistor from sink any current from the purple line. The current in the purple line therefore activates the MOSFET allowing the red line to flow to ground. In this operation the coil can charge.

Circuit for firing on falling edge trigger
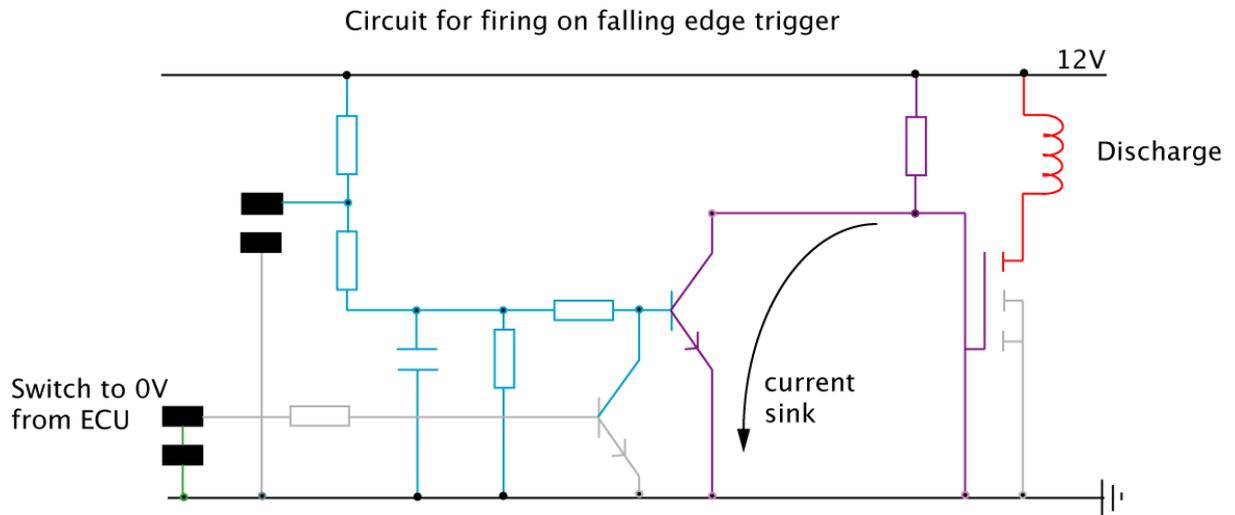
Figure 4.8 Ionisation system circuit for Coil operation with falling trigger during coil discharge

To fire the charged coil, the ECU pulls the applied voltage down to ground, creating the falling trigger edge. With the ECU signal voltage pulled to ground the current from the first transistor is removed, depicted in figure [4.8] through the green line switching to ground. With the first transistor deactivated, the full current of the light blue line is applied to the second transistor. With the extra current the second transistor activates sinking the current in the purple line to ground. This effectively reduces the current seen by the MOSFET causing it to deactivate. The deactivated MOSFET disconnects the primary coil from ground causing the coil magnetic field to collapse and discharge to ground through the spark.

Circuit for firing on rising trigger edge
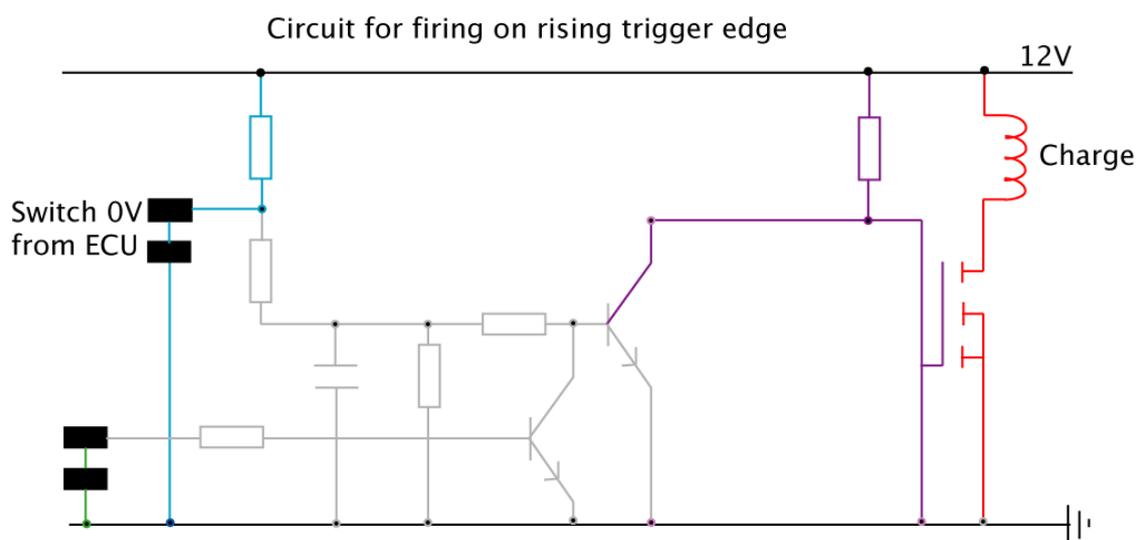
Figure 4.9 Ionisation system circuit for Coil operation with rising trigger during coil charge

For the rising edge trigger system the ECU requires connecting to the uppermost ECU connector. As the lower connector is no longer in use there is no voltage applied to it. For the rising trigger edge system the coil needs to charge during the period the signal is at ground, figure [4.9]. The ECU pulls the 12v to ground and directs the light blue line to ground causing the second transistor to be deactivated. As the second transistor is deactivated, the current in the purple line allows the MOSFET to activate. The active MOSFET allows the primary coil winding to connect with ground and charge the coil.

Figure 4.10 Ionisation system circuit for Coil operation with falling trigger during coil discharge

To fire the charged coil, the ECU allows the 12V to flow to the second transistor, figure [4.10]. This activates the transistor and causes the current in the purple line to sink to ground. This effectively reduces the current seen at the MOSFET. With the lower current the MOSFET deactivates disconnecting the primary coil to ground causing the magnetic field to collapse and fire the spark plug.

Therefore this circuit allows coil activation of either a rising or falling edge trigger signal from an ECU.

4.2.4 CIRCUIT FOR TRIMMING COIL OSCILLATION

The following part of the circuit is designed to address the resulting problem of the coil oscillation. Although a small coil is used to reduce the oscillation problem there is still going to be some interference. This interference can be clipped off the signal by using the following circuit.
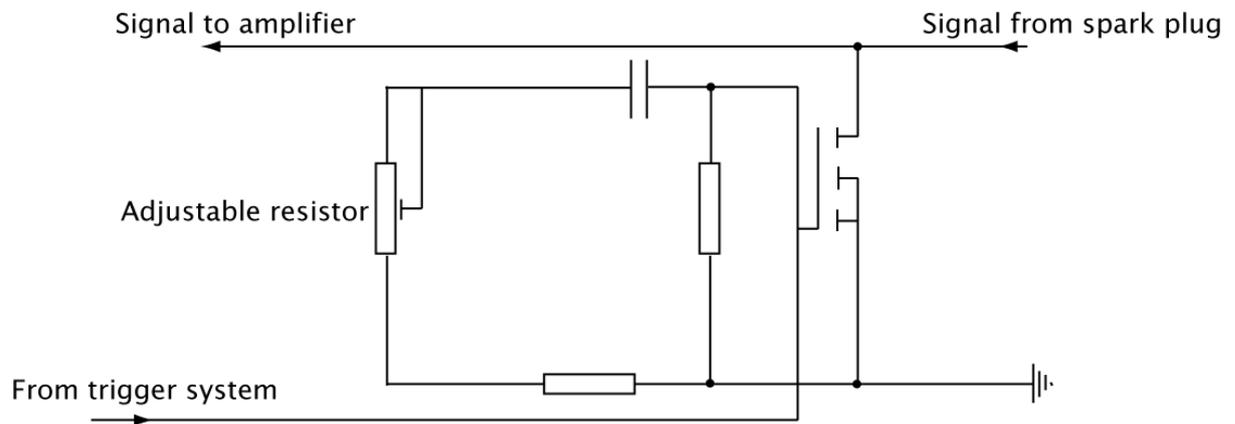
Figure 4.11 Ionisation system circuit for trimming coil oscillation

When the coil has been triggered for discharge, the current for operating the MOSFET in the previous ignition control circuit is also used to activate the MOSFET in figure [4.11]. When this MOSFET is activated it allows the voltage seen by the amplifier to sink to ground, causing the output signal to be clipped. By using the capacitor and variable resistor in the circuit, this provides decay to the clipping that can be adjusted. Therefore, depending on the required ionisation output signal the coil oscillation can be varied from completely clipped to no clipping of signal.

4.2.5 OVERALL IONISATION SYSTEM
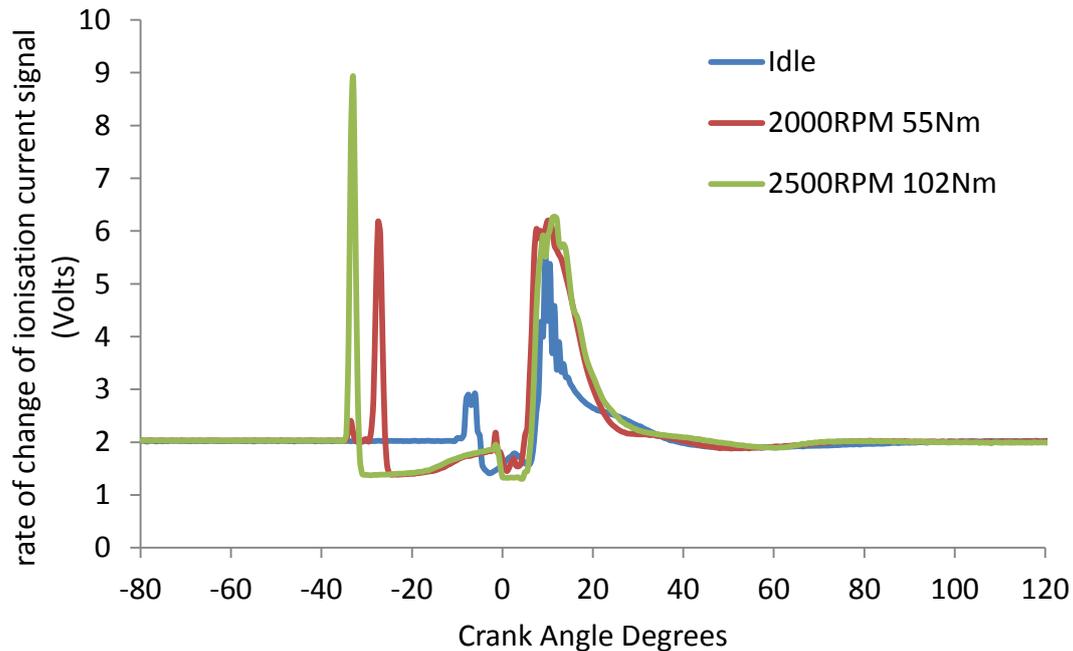
Figure 4.12 Example of rate of ionisation current signal taken from firing spark plug for three operating conditions of a Ford Zetec SI engine

Overall, this circuit offers a portable ionisation instrument that can allow ionisation sensing for multiple target engines while allowing firing of the spark plug. Figure [4.12], shows a typical example of three hundred cycle averaged rate of change of ionisation current signal as voltage taken from the firing plug of a Ford Zetec SI engine. The spark effect on the ionisation feedback can be seen at all three operating conditions as a positive voltage spike prior to rate of ionization for combustion. The voltage used for sensing is adjustable. This allows the system to be used with different sparkplugs and tuned for different operational conditions. If the sensor being used is also the firing plug it can be triggered from the standard ECU regardless of what type of trigger system is used by the manufacturer. When a firing plug is used as the sensor, the circuit also allows for a level of clipping to be applied to the signal to remove the coil oscillation from the ionisation trace.

The overall circuit diagram for the ionisation system can be seen in appendix A.

4.3 IONISATION FEEDBACK FOR HCCI COMBUSTION CHARACTERISTICS

The following work investigates the capability of the developed mobile ionisation systems signal for an efficient feedback method for information on start of combustion, combustion duration, rate of combustion, and peak pressure magnitude. To evaluate the ionisation feedback it was possible to apply the system to a two stroke HCCI engine. From the review in chapter two, we know that relationships between ionisation and cylinder pressure have been investigated for four-stroke HCCI engines, showing a strong potential for a feedback signal. The two-stroke HCCI combustion tends to have higher levels of in-homogeneity. With the known concerns of ionisation feedback quality, the in-homogeneity could potentially alter the reliability of relationships that have been previously observed for four-stroke HCCI. Often ionisation is used to construct predictions of cylinder pressure for feedback information. Comparing ionisation to cylinder pressure allows the relationships between ionisation and combustion to be understood. For feedback this offers two possible avenues;

- Does ionisation need to be related to cylinder pressure for feedback?
- Or can ionisation be used directly for feedback using its own attributes?

The adaptive feedback control architecture would rely upon a feedback signal that is accurate, reliable and fast. If ionisation can offer the reliability to be used directly it could potentially offer a mathematically simple solution for feedback. To evaluate the ionisation feedback signal for control, the mobile ionisation system was linked with the Lotus Omnivore engine and National Instruments Labview was used for data acquisition.

(removed for copywrite)

Figure 4.13 Lotus Omnivore engine (23)

The Lotus Omnivore engine is a spark-ignition-based research engine that has been designed to investigate the potential for wide-range HCCI operation using a variety of fossil and renewable liquid fuels, figure [4.13]. The Omnivore engine is able to operate under HCCI combustion due to the addition of a number of key combustion technologies. VCR is made possible through a variable position junk piston in the cylinder head known as the "puck". The puck does not move at engine speed but is controlled electronically through an electric motor and worm drive arrangement, allowing for a CR range of 10-40:1. The puck is also water cooled and carries the spark plug and two injector positions that allow for a direct injection fuelling system. By using the simple piston-ported 2-stroke engine the cylinder head is free of moving parts. Having no poppet valves in the head is a key enabler in the VCR design. The injector system used on the Omnivore is an airblast type injection system. Airblast injection permits the rapid introduction and vaporization of the fuel in the combustion chamber.

(removed for copywrite)

Figure 4.14 Lotus Omnivore charge trapping valve system(23)

The Omnivore engine also incorporates a charge trapping valve system (CTVS), figure [4.14]. The valve is caused to oscillate by a short articulated connecting link from an engine speed eccentric shaft, rotated by a drive belt from the crankshaft. The articulated link between the eccentric shaft and the trapping valve actuating arm allows for independent variability of the opening and/or closing points of the CTVS. Therefore, the CTVS system permits the control over the mass of trapped exhaust gas in the cylinder. For more detailed information on the Lotus omnivore engine see descriptive publication (23).

For the experimental work the engine was operated using Ethanol E85[1] fuel. The operation conditions for the experimental tests for ionisation are shown in table [4.1].

---

1 E85 is 85% by volume Ethanol in bulk Gasoline. Stoichiometry for E85 is 9.7:1

Table 4.1 Lotus Omnivore operation condition for experimental tests

Compression Ratio	18.5 : 1
Engine speed	2000 RPM
Trapping valve position	Position 37%, EGR is proportional to actuator position.
Intake Pressure	103.5 Kpa at upper plenum
Intake Temperature	25°C
Injection timing	0 degrees BDC

During the test the fuel injection pulse width was adjusted for each operating point. The following durations in table [4.2] where used.

Table 4.2 Fuel injection pulse width during Lotus Omnivore test

Injection Pulse width	Measured AFR	Corresponding Load
2000μ S	11.33	17Nm
1900μ S	11.75	16Nm
1660μ S	13.3	13.4Nm

The data collected at these operating points allows us to investigate the relationships between the cylinder pressure and the ionisation for combustion characteristics.

4.3.1 EXPERIMENTAL DATA ANALYSIS

During the operation of the Lotus Omnivore engine it was possible to measure the rate of ion current signal and the pressure trace under steady state conditions for HCCI combustion for the three different AFRs. The purpose of the experiment was to investigate the relationships between ion current and pressure trace for combustion characteristics for the potential of it being used as a feedback signal for control. Although the ion current circuit only produced an output of rate of ionisation, it is possible to integrate this signal to gain a set of data for ionisation. Both will be used in the investigation as they show different characteristics of the ion current and by investigating them both it will show if either holds advantages for giving information on combustion characteristics.

4.3.2 DEFINING DATA OF INTEREST

The pressure trace is currently used widely for gathering information on in-cylinder conditions and allows for information on peak pressure position, peak pressure magnitude and detection of knock. The pressure trace can be used further in predictive calculations for

useful combustion properties such as MFB and rate of heat release. The pressure trace is formed from a combination of pressure change due to volume change through compression and expansion of a gas and pressure change due to combustion. This makes it difficult to predict the start of combustion from the pressure trace. However it is possible to estimate the MFB profile from the pressure trace using a method established by Rassweiler and Withrow (2). This allows for an estimation to be made on the start of combustion from the start of MFB.

$$\Delta p = \Delta p_c + \Delta p_v \tag{4.4}$$

$$p_i V_i^n = p_j V_j^n \tag{4.5}$$

$$\Delta p_v = p_j - p_i = \left[\left(\frac{V_i}{V_j} \right)^n - 1 \right] \tag{4.6}$$

$$\frac{m_{b(i)}}{m_{b(total)}} = \frac{\sum_0^i \Delta p_c}{\sum_o^N \Delta p_c} \tag{4.7}$$

Equations [4.4-4.7] are developed by Rassweiler and Withrow for calculation of MFB (2). This method is widely used, though it contains several approximations. Heat transfer effects are included only to the extent that the polytropic exponent n differs approximately from the adiabatic isentropic exponent y which is the ratio of specific heats for a gas. The pressure rise due to combustion is proportional to the amount of fuel chemical energy released rather than the mass of mixture burned. Also the polytropic exponent n is not constant during combustion (128). The prediction of the start of combustion will therefore only be as reliable as the polytropic exponents used to predict the compression and expansion processes.
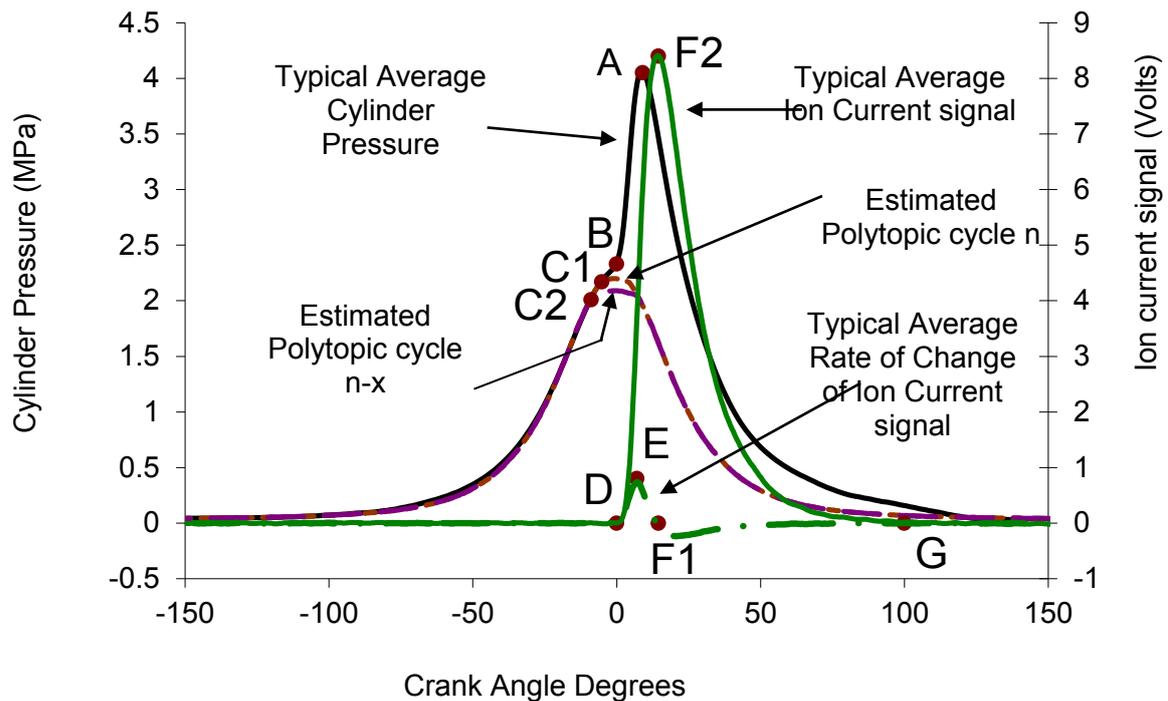
Figure 4.15 Typical Averaged Cylinder Pressure and Rate of change of Ionisation current signal and ionisation current signals

Figure [4.15] shows a comparison for a typical averaged cylinder pressure, a typical averaged rate of change of ionisation current signal and a typical averaged ionisation current signal. From figure [4.15] it is possible to highlight the following points of interest in this investigation.

Point (A) shows the peak pressure position, this offers information on peak pressure position and magnitude of peak pressure. This could be used for information on phasing of the combustion and understanding the quality of the combustion.

Point (B) is an inversion on the cylinder pressure trace and is due to the change in pressure as the piston begins its expansion stroke from TDC. This inversion will be more or less apparent depending on the location of ignition. The further ignition is before TDC the less predominant the inflection will be as the pressure due to combustion will have a more dominating affect.

Point (C1 & C2) shows the range for the approximate location of start of ignition. As the value of polytropic index *n* changes the height of the pressure change due to geometric compression changes. Therefore, the location of where the pressure curve leaves the

predicted polytrophic curve (point C1-C2) also changes. It is known to be very difficult to accurately estimate an appropriate value for *n* as it is known to vary during combustion (2).

Point (D) is the start of ionisation current signal, this can be seen to be later than start of the predicted combustion. The start of ionisation is a local measurement; it will depend on start location of combustion, timing of combustion and location of the sensor.

Point (E) is the peak of the rate of change of ionisation current signal. As the ion current circuit amplifier is differential, the trace represents the first differential of the ionisation current signal. Therefore, the peak of this signal is proportional to the peak rate of change in ionisation current.

Point (F1, F2) is the peak concentration of ionisation where rate of ionisation crosses zero (F1). At this point the positive increase in ionisation has reached its maximum, as the value drops below zero, this represents the start of the negative change in ionisation.

Point (G) is the end of the ionisation current signal. There is no longer any rate of change positive or negative and this will be the end of ionisation detection.

From visual inspection there appears to be relationships between the pressure trace and the ion current traces. These relationships can be unclear; ion current gives a local reading from the products of the combustion reactions, whereas cylinder pressure is an accumulative averaged reading from pressure rise due to combustion and volume change. The following investigation will compare the key points highlighted from the pressure trace and ionisation signals. If there is correlation with known relationships it will validate the mobile ionisation system as a tool for feedback and allow the ionisation current signal to be investigated as an efficient feedback signal.

4.3.3 DIRECT IONISATION ANALYSIS

The first investigation was to determine if the ionisation current signals had any relationships or correlation with points of interest with pressure trace. But first the acquired signal required a level of processing.
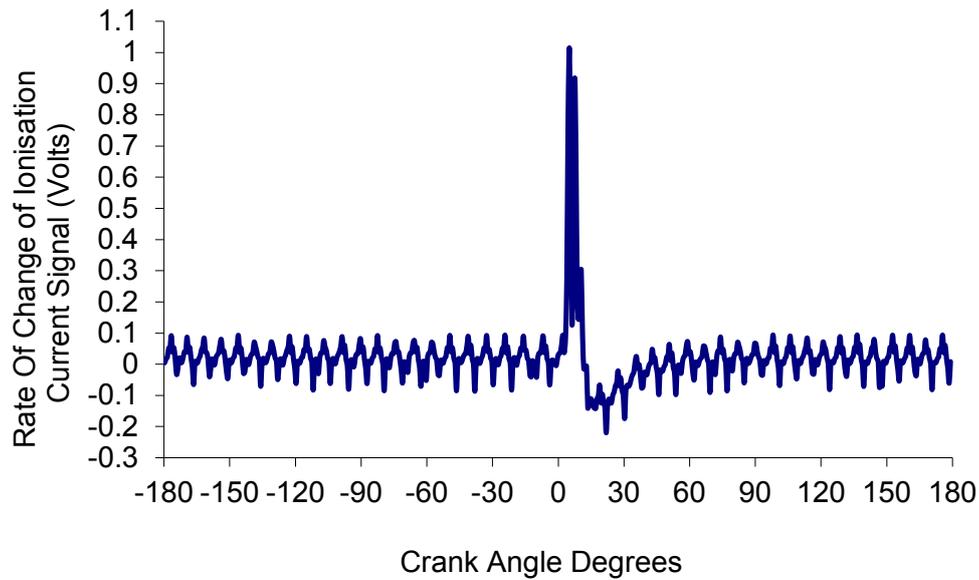
Figure 4.16 Typical Unprocessed Rate of Ionisation current Signal

Figure [4.16] shows a typical unprocessed rate of change of ionisation current signal. There is significant baseline noise and this makes it difficult to accurately pinpoint a start, a peak value and an end. The only value that could be defined would be where the signal crosses zero, previously described as point (F1). Although this can become unclear if the noise level increased. The problems found with noise can be improved in future work by having better screening on high voltage cables. This data can be filtered and averaged to improve the clarity of the signal and make finding points of interest more reliable.
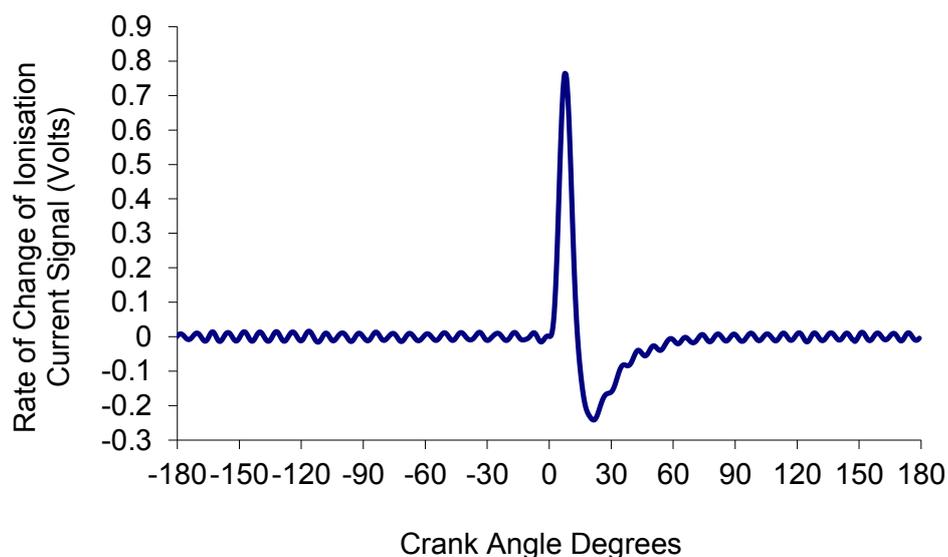
Figure 4.17 Typical Processed Rate of Ionisation current Signal

Figure [4.17] shows a typical processed rate of change of ionisation current signal. The processed signal is formed from averaging ten consecutive cycles then removing noise through the Wavelet toolbox in Matlab. The processed signal has a much more defined start location, peak location, zero crossing point and end. By averaging and using Wavelet Toolbox for filtering the signal it has made it possible for further analysis to be carried out for comparison with cylinder pressure for combustion characteristics.

4.3.4 KNOCK COMPARISON
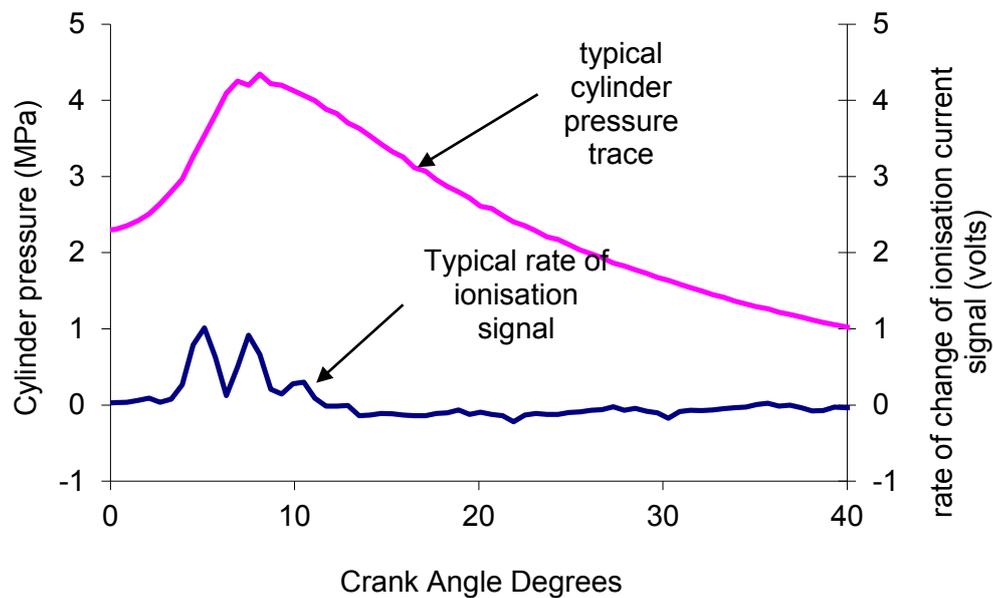
Figure 4.18 Comparison Between Typical Unprocessed Rate of Ionisation current signal and Typical Cylinder Pressure Trace

Figure [4.18] shows a comparison between typical unprocessed rate of change of ionisation current signal with the typical cylinder pressure trace from the Omnivore engine. The pressure trace shows a small level of knock indicated by the spikes beginning to show at the top of the pressure trace peak. This is an indicator that auto ignition is taking place. The rate of change of ionisation current signal also shows multiple peaks from the combustion showing rapid changes in rate of ionisation. The multiple peaks could be due to the nature of the rapid combustion of HCCI representing knock. It has been reported that ionisation is capable of detecting knock (131).

4.3.5 PHASING CHARACTERISTICS

To investigate the characteristics of the timing of ionisation with combustion timing, comparisons have been made between points along the ionisation current and rate of change of ionisation current signals with peak pressure position. Points of interest investigated are; the start of the signal, peak location of the signals and points at 15% 50% and 75% of the rising edge. Relationships between these points could represent a relationship with combustion timing. When relationships are compared for accuracy, Microsoft Excel Function "VARP" is used to describe the spread of data.

$$\frac{\sum(x-\bar{x})^2}{n}$$

(4.8)

Where *x* is the sample mean average and *n* is the sample size. VARP calculates the variance based on the entire sample n. For all the calculations the sample size was 300 cycles.
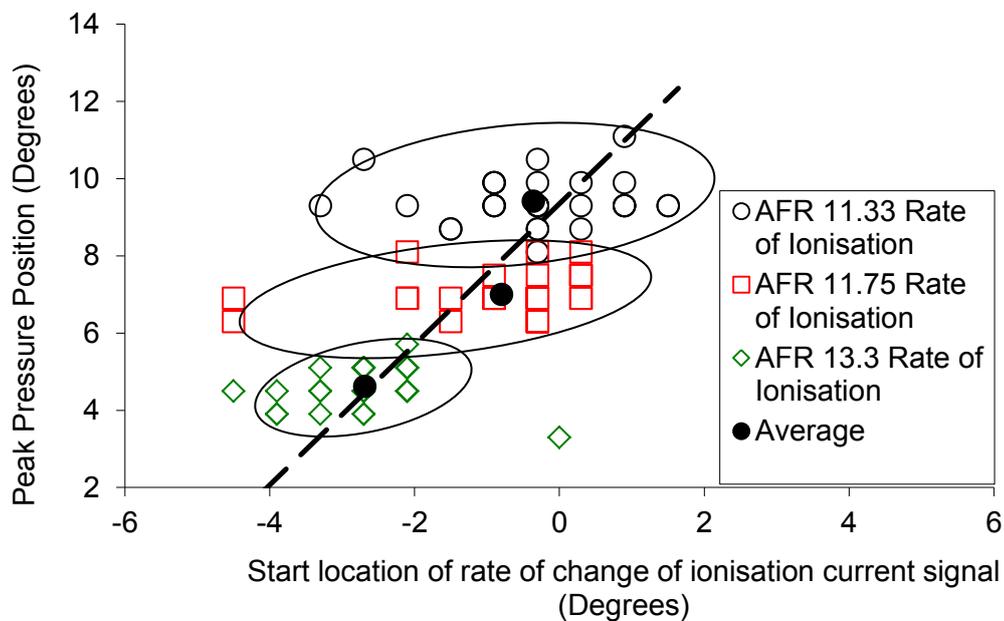
Figure 4.19 Correlation between averaged peak pressure position and averaged start location of rate of change of ionisation current signal

Figure [4.19] shows the correlation between averaged peak pressure position and averaged start location of rate of change of ionisation current signal for three different AFRs. As the AFR becomes leaner the peak pressure position advances as well as the start timing for rate of change of ionisation current signal. This indicates that as the AFR becomes leaner the start of combustion is becoming earlier and is represented in both the peak pressure position and the start location of rate of change of ionisation current signal. The ignition becomes earlier

as there is less fuel in the cylinder, leading to less energy used in heating the fuel. Therefore, during compression the temperature/pressure requirements for auto ignition to initiate occur earlier in the cycle. The variability of the peak pressure position shows that start of ignition is variable at each of the AFR operations. There is however a larger variance of 1.12 CAD for start location of ionisation current signal compared to peak pressure position variance of 0.3 CAD. When combustion begins in HCCI it will start at a location that has local inhomogeneous variations. This means that from cycle-to-cycle the start timing may be very similar but the location can be any point within the cylinder (132). The rate of change of ionisation current signal is measured from the spark plug and represents only a local region close to and around the spark plug probe. If start of ignition location varies from cycle-to-cycle then the distance from start of ignition and the sensor will vary. This variation in distance will cause a variation in delay between start of combustion and sensing of ions at the spark plug. From figure [4.19] it is therefore possible to see that start of combustion advances with leaner AFR and that this is shown through change in peak pressure position and start of rate of change of ionisation current signal. Peak pressure position also shows that combustion timing has some variability, and start of ionisation current signalshows that start location of combustion is variable, typical of HCCI.
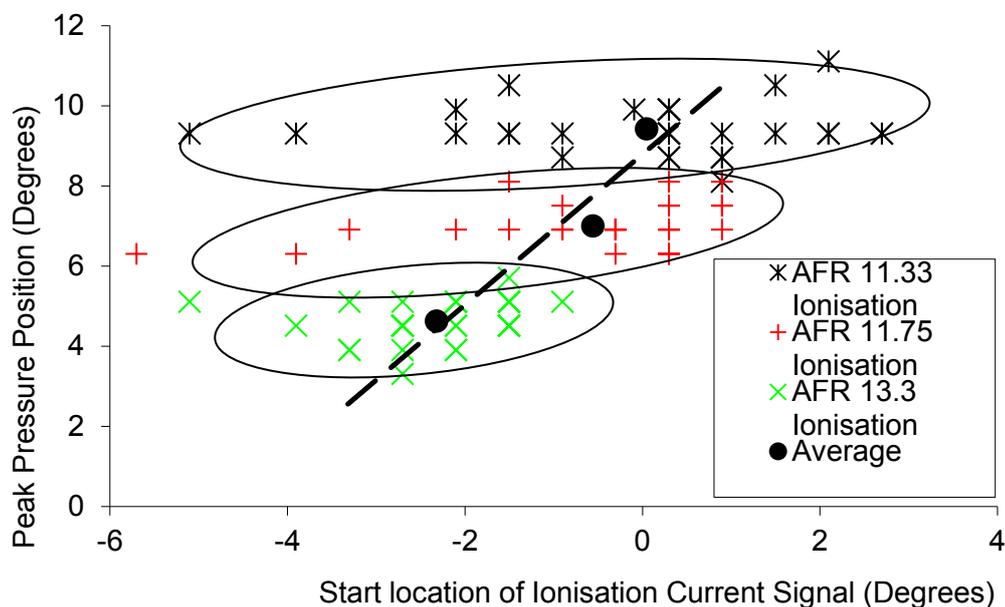
Figure 4.20 Correlation between Averaged Peak Pressure and Averaged Start Location of ionisation current signal

Figure [4.20] shows correlation between averaged peak pressure and averaged start location of ionisation current signal for three different AFRs. The timing of start location of ionisation current signal can be seen to have a larger variance of 2.04 CAD then the start of rate of

change of ionisation current signal. When considering methods for establishing the presence of a peak, most use the comparison of some signal attribute. For example, rate of change (first differential) against a user adjustable threshold. This method discriminates peaks from the background signal noise (133).The rate of change of ionisation current signal is the first differential of the ionisation current signal, the start of the rate is therefore the point that ionisation starts to increase. The interpretation of where the start location is appears to be more accurate with the rate then from determining a rise with in the noise on the ionisation current signal.
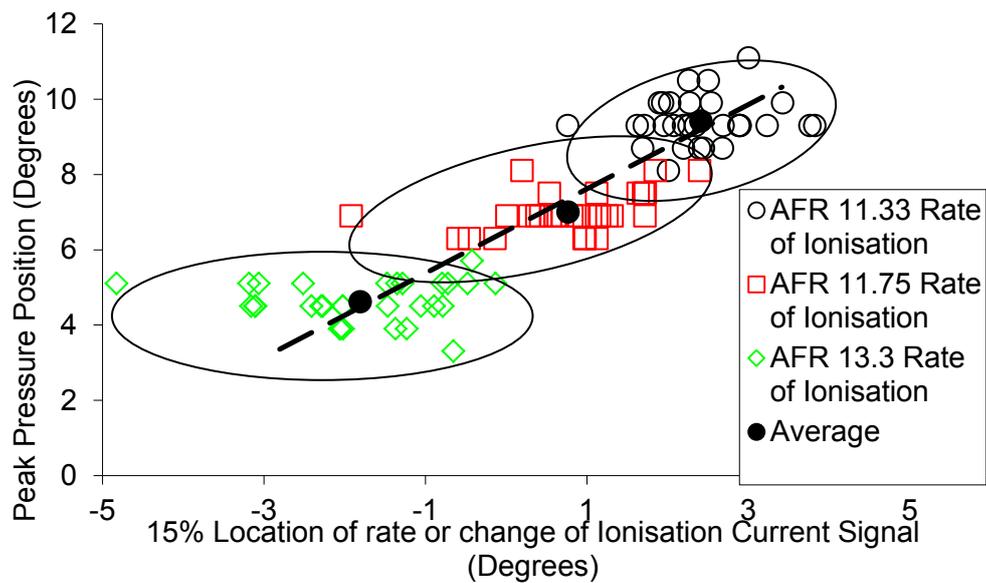
Figure 4.21 Correlation between Averaged Peak Pressure Position and Averaged 15% location of Rate of change of Ionisation current signal

Figure [4.21] shows the correlation between averaged peak pressure position and averaged 15% value location of rate of change of ionisation current signal for three different AFRs. The variance for the 15% value location of rate of change of ionisation current signal is reduced to 0.74 CAD compared to start of rate of change of ionisation current signal that had variance of 1.12 CAD. The baseline noise will decrease the accuracy of finding the start location. By taking a point at 15% value of rate of change of ionisation current signal it is close to the start of combustion but will decrease the influence of the baseline noise.
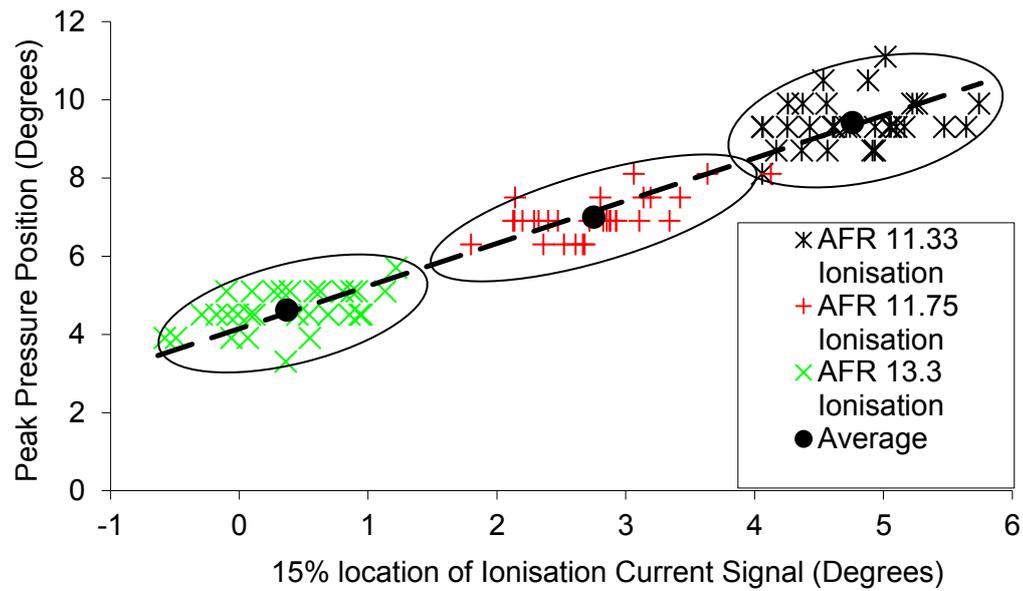
Figure 4.22 Correlation between Averaged Peak Pressure and Averaged 15% value location of ionisation current signal

Figure [4.22] shows the correlation between averaged peak pressure and averaged 15% value location of ionisation current signal for three different AFRs. The variance for 15% value location for ionisation current signal is 0.23 CAD. This result is significant reductions compared to the start location and lower than the 15% location for rate of change of ionisation current signal result. rate of change of ionisation current signal has a larger signal to noise ratio. The data from ionisation current signal 15% location has less variance due to its larger magnitude of signal compared with the noise.

Figure 4.23 Correlation Between Averaged Peak Pressure Position and Averaged 50% value location of rate of change of ionisation current signal

Figure [4.23] shows correlation between average peak pressure position and averaged 50% value location of rate of change of ionisation current signal for three different AFRs. The variance has reduced to 0.36 CAD and is lower than that at 15% value location. The rate at 50% of the signal is less variable because once the combustion has become established the rate will be determined by the dilution of the mixture through residuals. Although the residuals were set by a constant trappimg valve position there will still be some variation in quantity from cycle-to-cycle. Once combustion has initiated due to the local inhomogeneous conditions, the initial rate will be variable, but as the combustion becomes more established the rate will be determined by the bulk cylinder charge ratio of dilution and EGR.

Figure 4.24 Correlation between Averaged Peak Pressure Postion and 50% value location of ionisation current signal

Figure [2.24] shows correlation between averaged peak pressure position and 50% value location of ionisation current signal for three different AFRs. The variance is 0.2 CAD, which is smaller than that at 15% value location. Once combustion has begun, it will take a certain time to complete. In HCCI combustion this is a fast process and happens throughout the cylinder with little or no flame front. As the bulk of combustion establishes, the effect from local in-homogeneities at the start of combustion are reduced on the local ionisation reading. Variations in the signal will be mainly due to cycle-to-cycle variation in the dilution through EGR, AFR and the start of combustion. The 50% location of ionisation current signal offers a more repeatable accuracy of feedback than the 15% position and start of ionisation current signal.
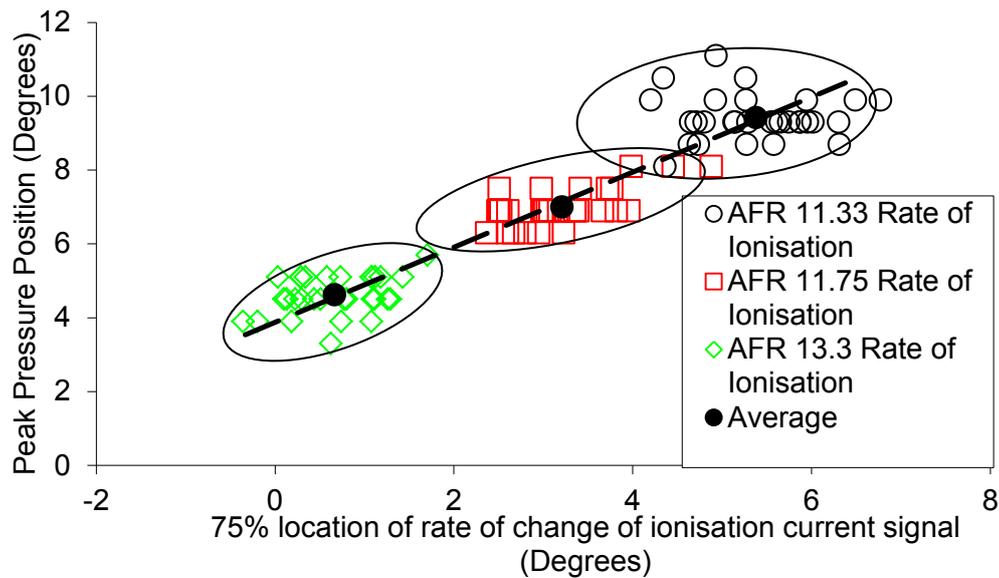
Figure 4.25 Correlation between Averaged Peak Pressure Position and 75% Value of rate of change of ionisation current signal

Figure [4.25] shows the correlation between averaged peak pressure position and 75% value of rate of ionisation current signal for three different AFRs. The variance has reduced marginally to 0.35 CAD showing a small reduction over the 50% value location. The change in variance is small if the combustion has become established by 50% then by 75% there will be little difference in variation as the nature of the combustion means that once it is established it is burning over the entire cylinder (134).
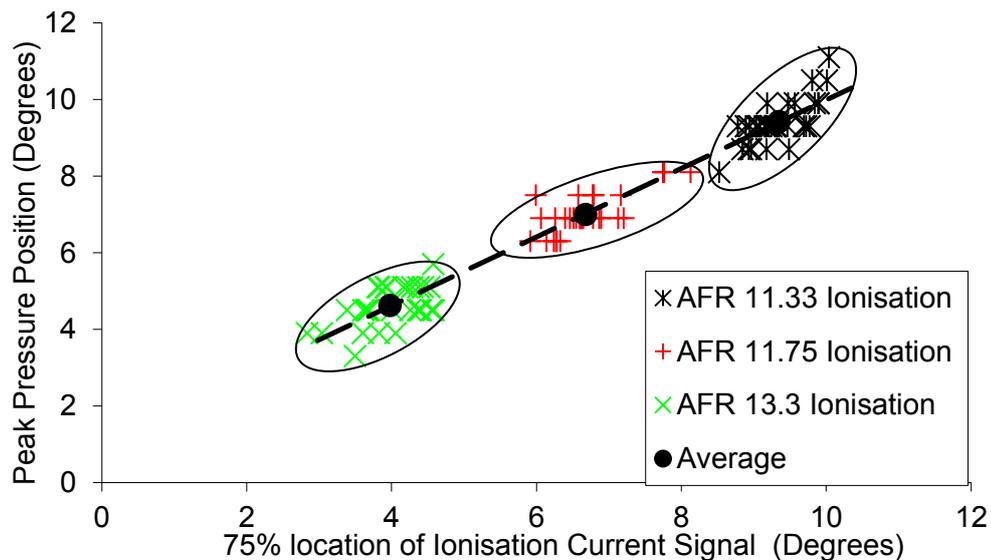
Figure 4.26 Correlation between Averaged Peak Pressure Position and Averaged 75% value Location of ionisation current signal

Figure [4.26] shows the correlation between average peak pressure position and 75% value of ionisation current signal for three different AFR. The variance has marginally increased to 0.207 CAD from that of 50% value location. The change in 75% for ionisation current signal can be considered due to small effects as the value is very close to that of 50% ionisation current signal.
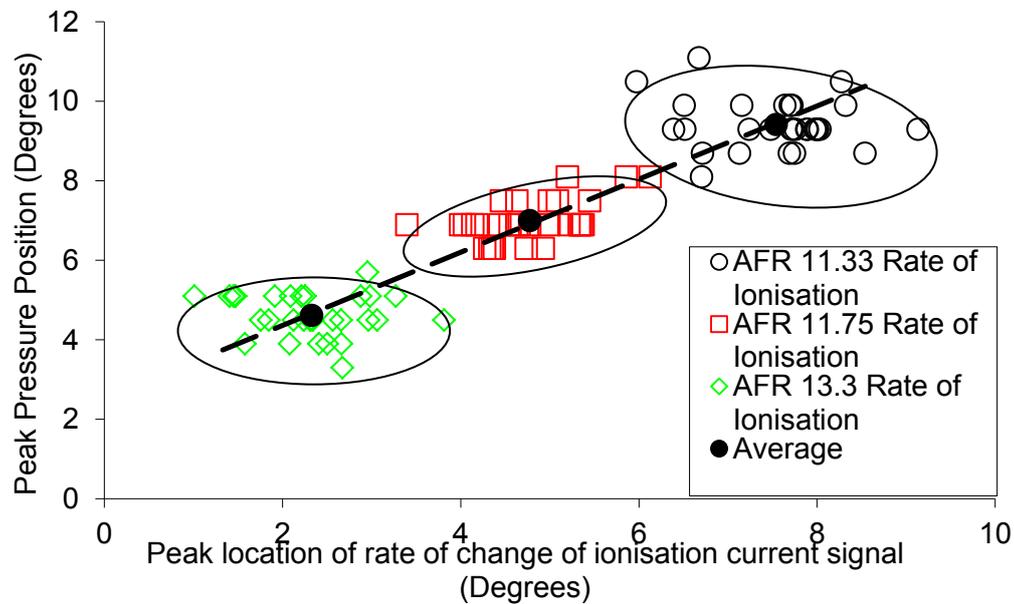
Figure 4.27 Correlation between Averaged Peak Pressure Position and Averaged Peak location of rate of ionisation current signal
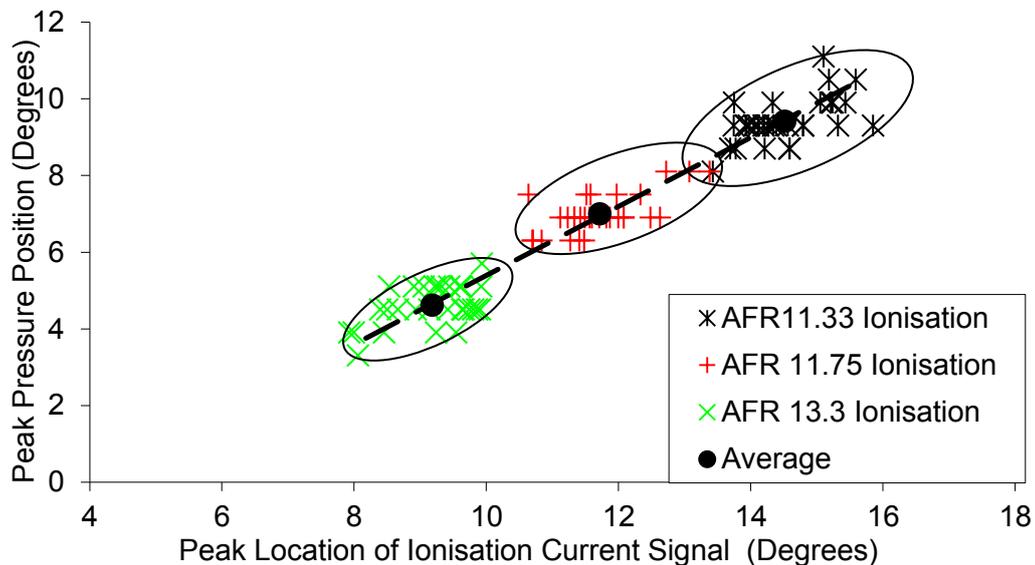
Figure 4.28 Correlation between Averaged Peak Pressure Position and Averaged Peak Location of ionisation current signal

Figure [4.27] shows the correlation between average peak pressure position and average peak location of rate of ionisation current signal. The variance has increased to 0.401 CAD.

This increase potentially could be due to the effect of averaging the multiple peaks of the unprocessed signals as described in section 4.3.3. The rate of ionisation is also not an accumulated reading like pressure. Any variations in charge mixture will affect the rate of combustion, which will further affect the magnitude and location of peak value of rate of ionization. The results suggested a variation in rate of combustion from cycle-to-cycle. As the same amount of fuel is being delivered each cycle, the increase in variance seen in figure [4.27] could be due to the effect of rate of combustion. For example, if there is a slower rate it will take longer to burn all the fuel and therefore longer to get to peak value of ionisation. Conversely, if there is a faster rate than the fuel will ignite faster and the peak value of ionisation will be sooner.
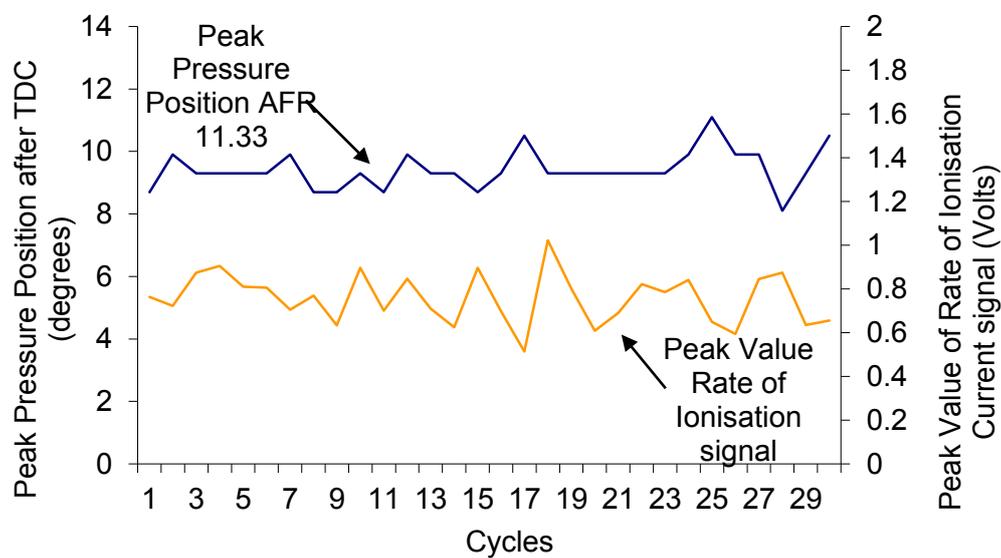
Figure 4.29 Comparison of Peak value of rate of ionisation current signal and Peak Pressure Position

Figure [4.29] shows a comparison of peak pressure position and peak value of rate of ionisation current signal for AFR 11.33. The plot shows that if the peak rate of ionisation current signal is lower then the timing of the peak pressure position tends to be later. Conversely if peak rate of ionisation current signal is higher, than the peak pressure position becomes advanced towards TDC.

Figure [4.28] shows the correlation between average peak pressure position and average peak location of ionisation current signal. The variance has increased to 0.4001 CAD. The peak locations of ionisation current signal are affected by start of combustion and rate of combustion. As the peak of ionisation current signal will be detected after the bulk of

combustion has taken place, any variation of rate will alter the location and magnitude more than values taken earlier in the cycle.

The trend for peak pressure to vary inversely with the change in peak rate of ionisation current signal potentially provides information on rate of combustion.
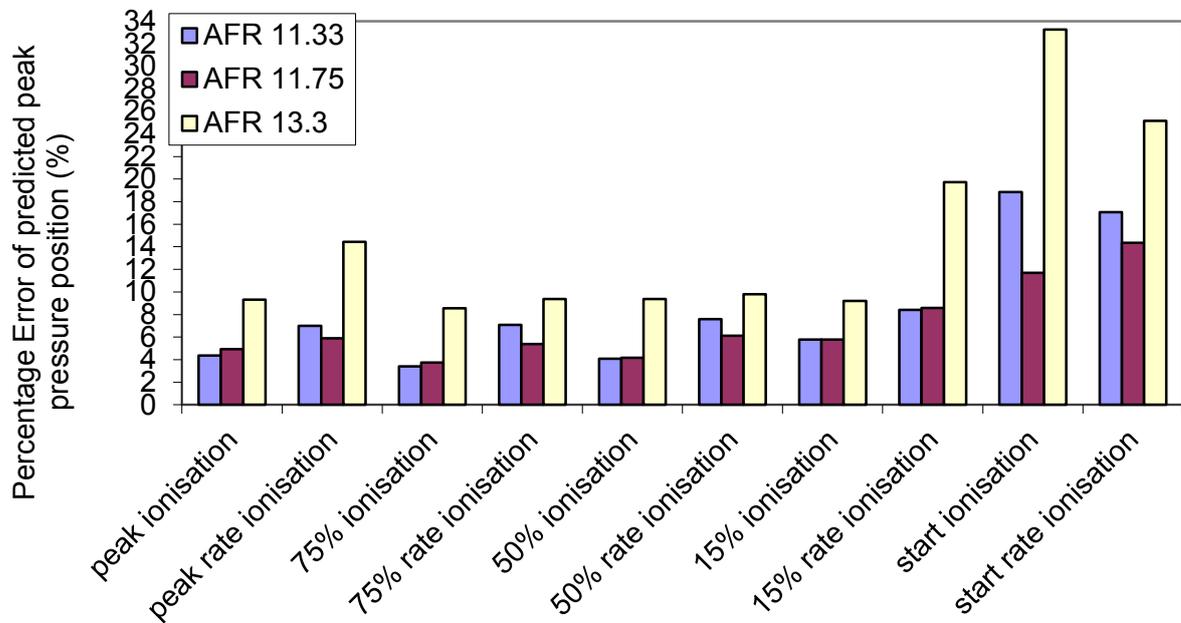
Figure 4.30 Comparison of Averaged error of Prediction of Peak Pressure Position from Points of Interest on ionisation current signal and rate of ionisation current signal

Figure [4.30] shows the average error for prediction of peak pressure position from ionisation current and rate of ionisation current signals data from the three AFRs. It shows that overall the start location has the largest error. This error reduces at locations along the signal to the 15%,50% and the 75% locations. The 75% locations tend to offer the lowest error, while peak positions show an increase in error. Points taken from the ionisation current signal tend to give less error than the equivalent point on the rate of ionisation current signal, with exception to the start position. At the start locations there are improvements on error for AFR 11.33 and 13.3 for start of rate of ionisation current signal. These results are what would be expected from the different levels of variance that have been discussed for each point of interest. This therefore illustrates that the more reliable locations for timing relationships are between the locations of 50% and 75% ionisation current signal. Around these points combustion can be established enough to reduce the effects of local in-

homogeneities at the start of combustion and provide high enough level of ions to be reliably sensed, repeatable and relate to the timing of combustion.
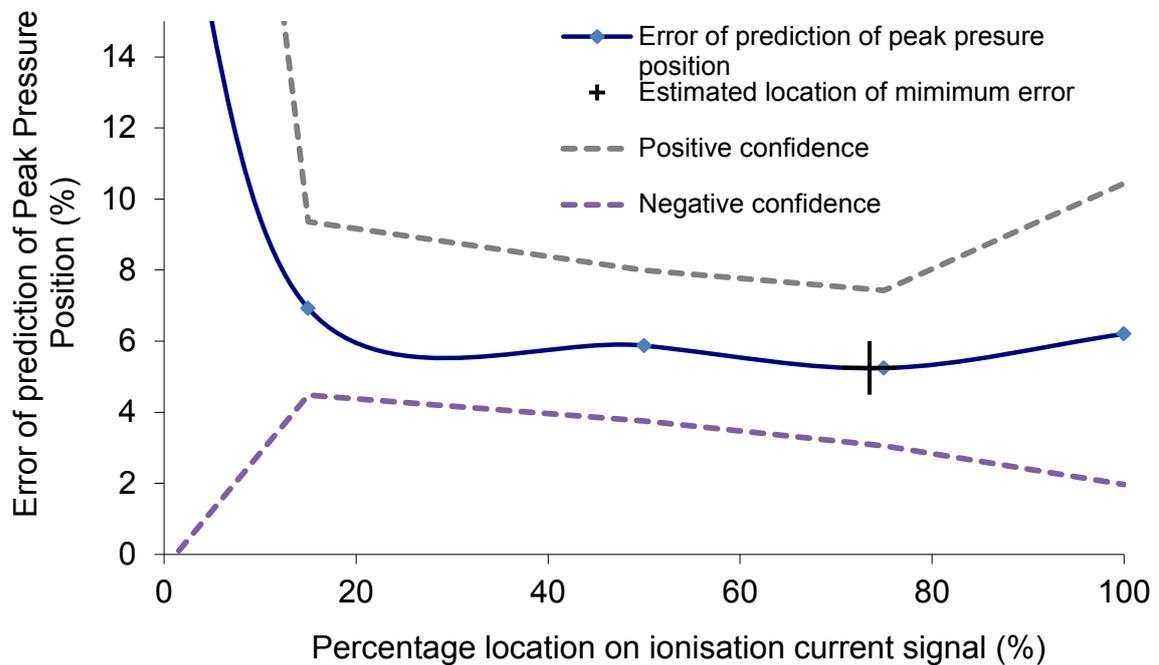
Figure 4.31 The change in Average Error of Prediction of Peak Pressure Position with change in Comparison Location of ionisation current signal

Figure [4.31] shows change in averaged prediction error of the three AFRs at different locations along the ion current signal rising edge with confidence intervals. The positive and negative confidence has been generated from the variance calculated at each location. 73.5% location is estimated to give the best results with an estimated average error of 5.23% in peak pressure position.

4.3.6 DURATION COMPARISON

For investigating combustion duration it would be difficult to take points directly from the pressure trace that could be reliably stated as combustion and not pressure change due to volume change. Therefore, for investigating duration the MFB was calculated using the equations [4.4-4.7] that were developed by Rassweiler and Withrow (2). Using the MFB allowed the theoretical separation of the pressure change due to volume change and combustion.
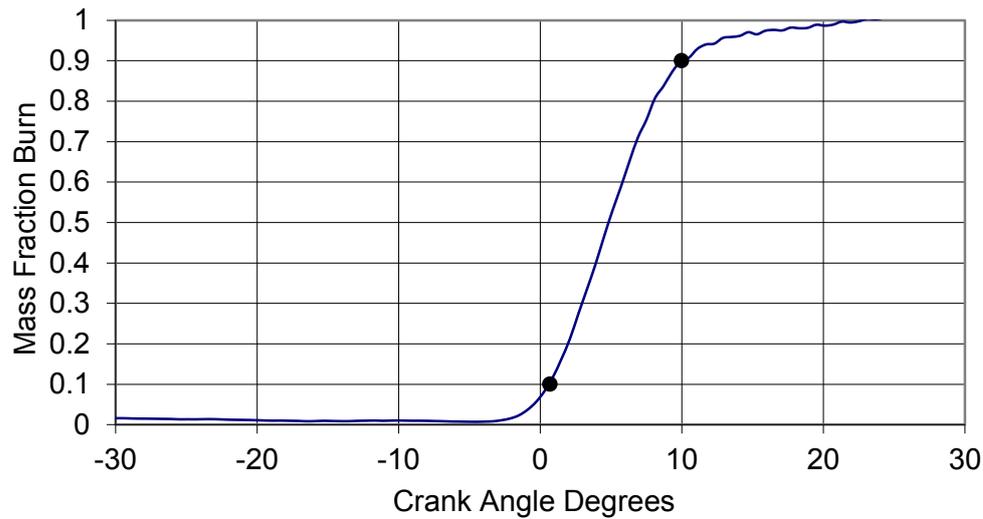
Figure 4.32 Typical calcualted mass Fraction Burn from cylinder pressure

Figure [4.32] shows a typical calculated MFB curve for the HCCI combustion. The duration that has been taken for the MFB is between 10% of the curve and 90% of the curve as these points are less influenced by any noise and should provide a repeatable measurement. For ionisation current signal the duration was taken from 15% of the rising edge of the signal and the location peak. This point is more distinguishable than attempting to find the end of the rate of ionisation current signal as discussed in section 4.3.2. The duration for rate of change of ionisation current signal is the same as the duration of ionisation current signal so only one comparison is produced.
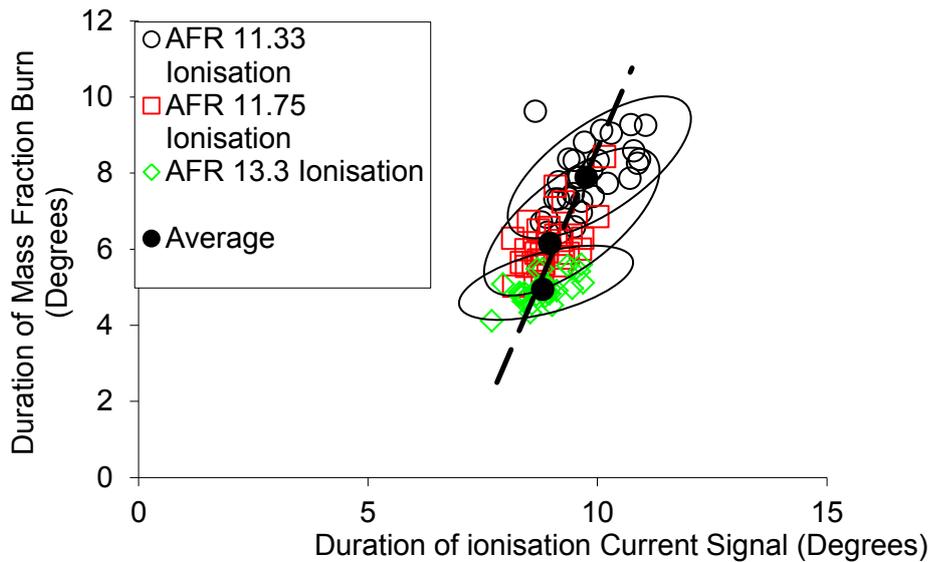
Figure 4.33 Correlation between duration MFB and duration of ionisation current signal

Figure [4.33] shows the correlation between the duration of 10-90% MFB and duration of ionisation current signal for three different AFRs. It can be seen that as the AFR becomes leaner the combustion duration decreases and the duration of ionisation current signal follows this trend. As the mixture becomes leaner there is less fuel to burn, this will lead to a shorter duration as it will take less time to burn all of the fuel unless the rate of combustion is changed.
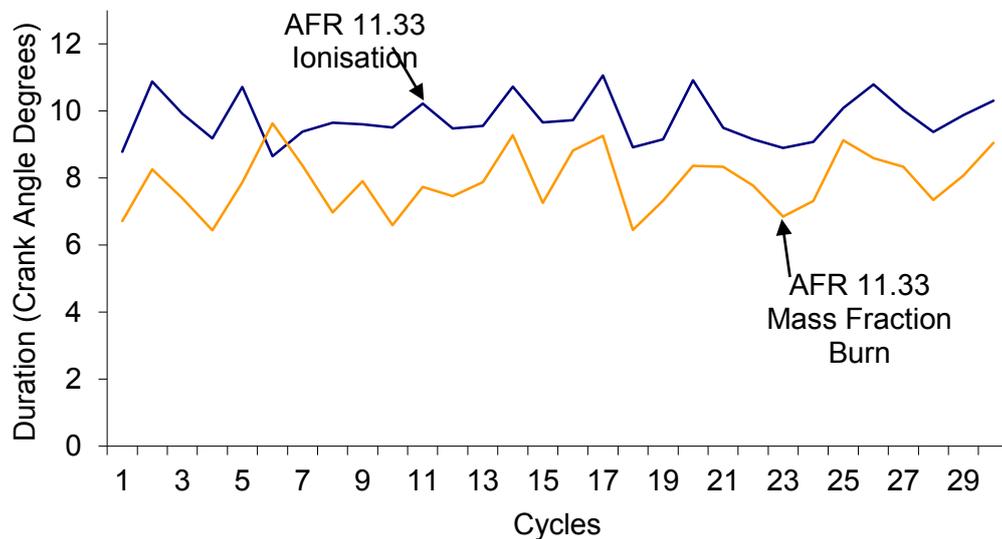
Figure 4.34 Comparison of Variation in Duration for MFB and Rate of ionisation current signal

Figure [4.34] shows the comparison between the variation in duration for 10-90% MFB and rate of ionisation current signal for 11.33 AFR. It can be noticed that as the combustion duration varies the duration of ionisation also tends to follows this variation.

4.3.7 PEAK PRESSURE MAGNITUDE RELATION

The peak pressure magnitude gives information on combustion performance; it would be of interest if there is a correlation between the peak pressure and the peak values of ionisation current signal and rate of ionisation current signal.
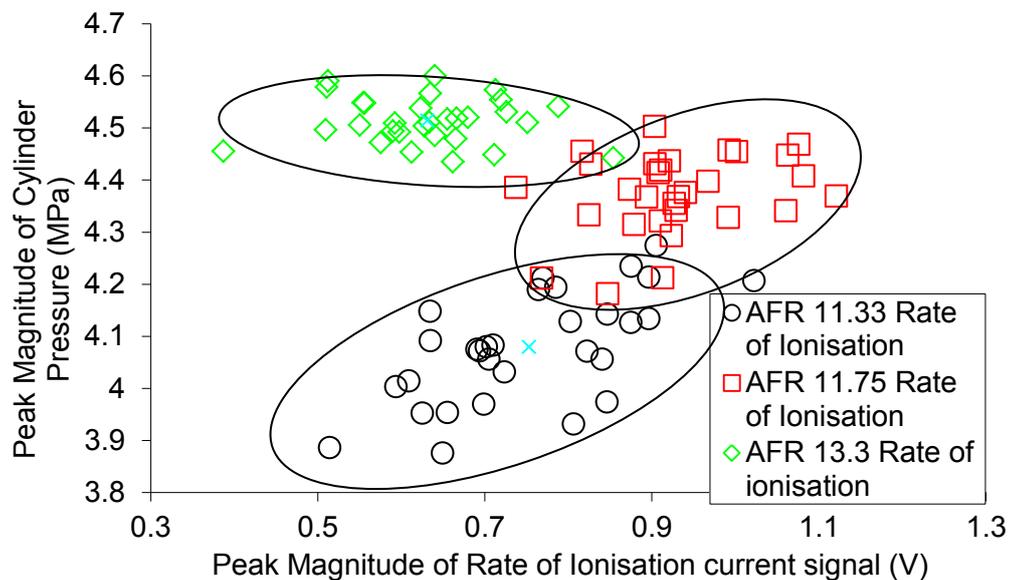
Figure 4.35 Correlation of Peak Pressure Magnitude and Peak rate of ionisation current signal Magnitude

Figure [4.35] shows the correlation between peak pressure magnitude and peak rate of ionisation current signal magnitude. As the AFR becomes leaner the cylinder pressure is seen to rise for this range of AFRs; however this is not followed by the peak magnitude of ionisation current signal. For the leanest AFR the peak magnitude of rate of ionisation current signal decreases. At each of the AFRs there appears to be correlation showing that there are relationships but without further evidence this is difficult to clarify.
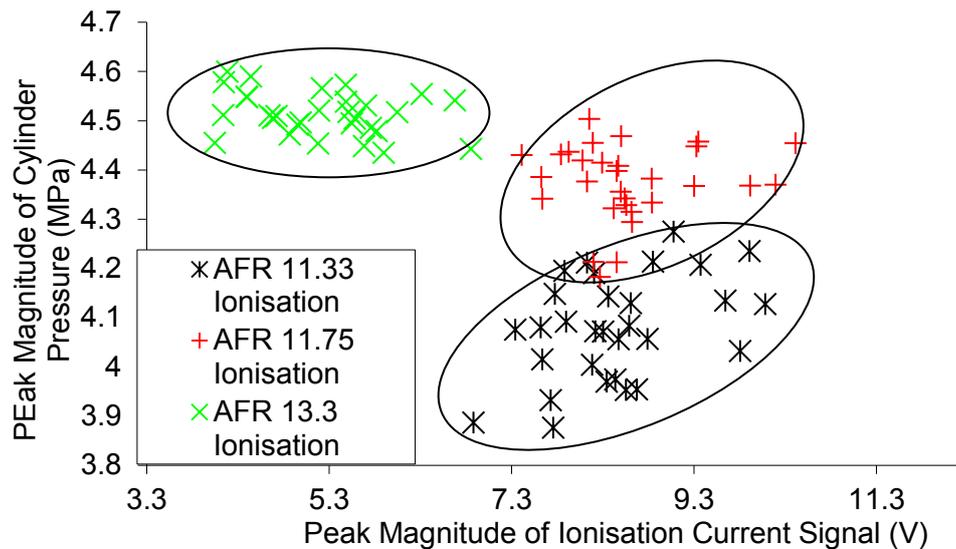
Figure 4.36 Correlation of Peak Pressure Magnitude and Peak ionisation current signal Magnitude

Figure [4.36] shows the correlation of peak pressure magnitude and peak ionisation current signal for three different AFRs. Again it can be seen that the peak of ionisation current signal does not follow the same trend as the peak pressure magnitude and at the leanest AFR 13.3 the magnitude decreases. The ionisation current signal is a measurement of the ions from combustion and the peak magnitude represents the maximum concentration of ions in the vicinity of the spark plug. The peak cylinder pressure is an accumulation of pressure due to volume change and pressure due to compression of a gas. As the AFR becomes leaner there is less fuel within the cylinder. The mixture compresses to a higher temperature as less energy is used to heat the fuel to the point of ignition, as there is less fuel. This extra heat causes an increase in pressure and will give a small increase to peak pressure for this range of AFRs. As there is less fuel in the cylinder there will be fewer ions to be sensed causing the magnitude of the signal to be less. Therefore, due to the nature of the two readings they may not show the same relationships between peak magnitudes.

4.3.8 AFR EFFECTS ON IONISATION SIGNALS
The AFR has been altered to gain the three different operating points of HCCI for this investigation. It has been seen in the investigation so far that as the AFR changes it has produced a change in ionisation current signal. Currently the changes have only been compared to changes in pressure signal, some of the relationships maybe clarified by investigating the effect on the ionisation current signal with change in AFR.
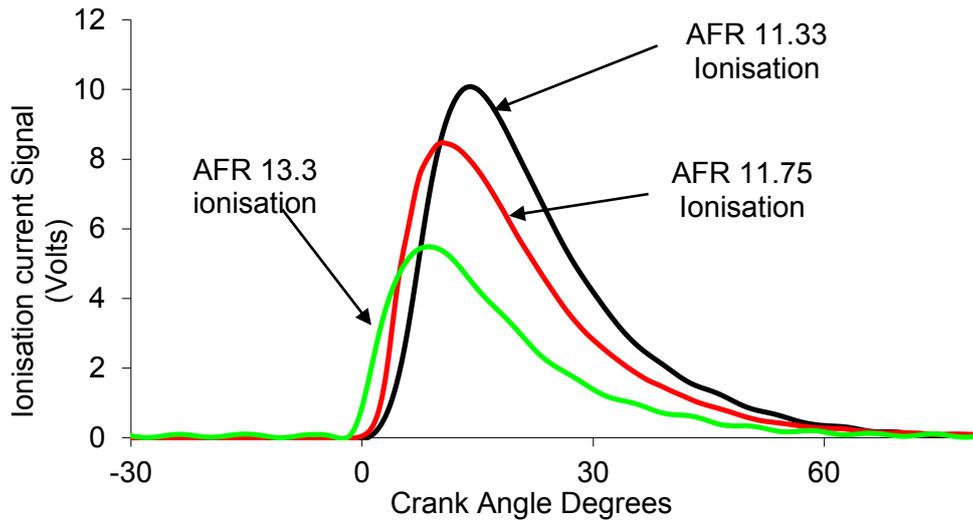
Figure 4.37 Comarison of typical ionisation current signal

Figure [4.37] shows the ionisation current signal decreases in magnitude as the AFR becomes leaner; this has been reported in other publications (88). If the concentration of fuel in the cylinder decreases then the products through combustion of this fuel will also decrease. The ionisation current signal is a reading of the ions from combustion; if less fuel is combusted there will be fewer ions to be sensed. Although figure [4.36] shows for a number of cycles the magnitude of the AFR 11.75 is very close if not higher then AFR 11.33 magnitudes. This is not entirely explained by the above theory. It is apparent from figure [4.33] and figure [4.37] the duration of ionisation current signal does decrease with AFR.
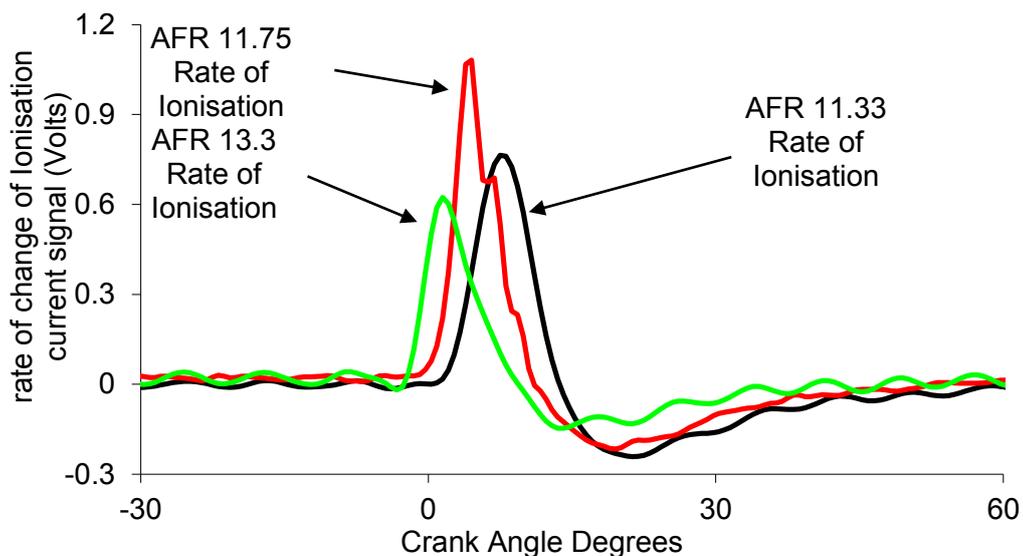
Figure 4.38 Comparison of typical rate of ionisation current signals

Figure [4.38] shows a comparison of typical rate of ionisation current signal for the three different AFR. From figure [4.35] and [4.38] the rate of ionisation current signal appears greater for AFR 11.75 then for AFR 11.33. If the small change in fueling between 11.33 and 11.75 AFR has allowed combustion timing to be earlier, the combustion can fall at a smaller volume close to TDC. At the smaller volume the density will be higher and rate of combustion will be greater. At the increased rate of combustion, more ions will be available at the smaller volume, increasing ionisation magnitude. The overall ionisation is still less due to less fuel in the cylinder overall. Therefore peak ionisation is affected by AFR but also combustion timing and density due to volume change. Due to the fuelling change between 11.33 and 11.75 being small, the affects of AFR change on combustion are small enough to allow the effects on timing and density due to volume change to be seen. Larger changes in AFR will decrease the magnitude of ionisation sufficiently to make this effect of density at volume negligible as seen between AFR 11.33 and 13.3.

4.3.9 ARTIFICIAL NEURAL NETWORK PROCESSING

From the above investigation it has been seen that the ionisation current and rate of ionisation current signals have relationships with duration of combustion, timing of combustion and to a certain extent may show information on rate of combustion and knock. However for peak pressure magnitude it was difficult to see a relationship as the AFR changes, although there did seem to be correlation with peak cylinder pressure at each of the AFRs. To this point, the investigation has covered point-to-point relationships and linear relationships between them. By using an ANN it is possible through training to find relationships between ionisation current signal, rate of change of ionisation current signal and the cylinder pressure trace. By using the ANN it may be possible to find relationships between all the points of the signals and relationships that may not be linear, that are not clear through visual inspection of the signals.

The "nftool" in Matlab has been used to produce the ANNs. The initial nftool neural network has one hidden layer of 20 neurons, and the training data is split to allow for training, validation and testing in the following percentages, 70%, 15% and 15% respectively. The standard ANN from nftools will not be optimal but will offer an ANN that can be trained to find and generalise relationships. Two ANNs were created, one for ionisation current signal and the other for rate of ionisation current signal. For each ANN the training data comprised of the ionisation current signals data for AFR tests at 11.33 and 13.3 as inputs and the corresponding pressure signals were used as the desired outputs. As the Ionisation current and pressure trace signals are averaged over 10 combustion cycles the total training data comprised of 60 averaged combustion cycles. These cycles where randomised then split into

the ratios above giving, 42 averaged combustion cycles for training , 9 for validation and 9 for testing. For each training epoch of the ANN an entire cycle of ionisation current or rate of change of ionisation current signal is presented, the corresponding output will be an entire predicted cylinder pressure trace.

At the beginning of the training there will be large error between the ANN output and the desired pressure trace. This error is back-propagated through the ANN layers and used to update weights at each neuron. Through this process of weight adaption, the relationships between the inputs and the outputs are generalized and stored within the combination of weights throughout the ANN. When the ANN has an input not within its training knowledge, it can approximate the output from the generalized relationships held within its knowledge. If the unknown input differs significantly from the inputs that the ANN has been trained upon; the generalized relationships will be less applicable and the output error will increase. As the ANNs have been trained from data taken at AFR of 11.33 and 13.3, the obtained knowledge will give an approximation of the outputs for the unknown inputs at an AFR of 11.75. Once the two ANNs where created, the appropriate ion signals for test condition AFR 11.75 where presented to the ANNs as inputs to give predictions of cylinder pressure. The predicted pressure traces are compared against the real pressure traces for this operating condition.
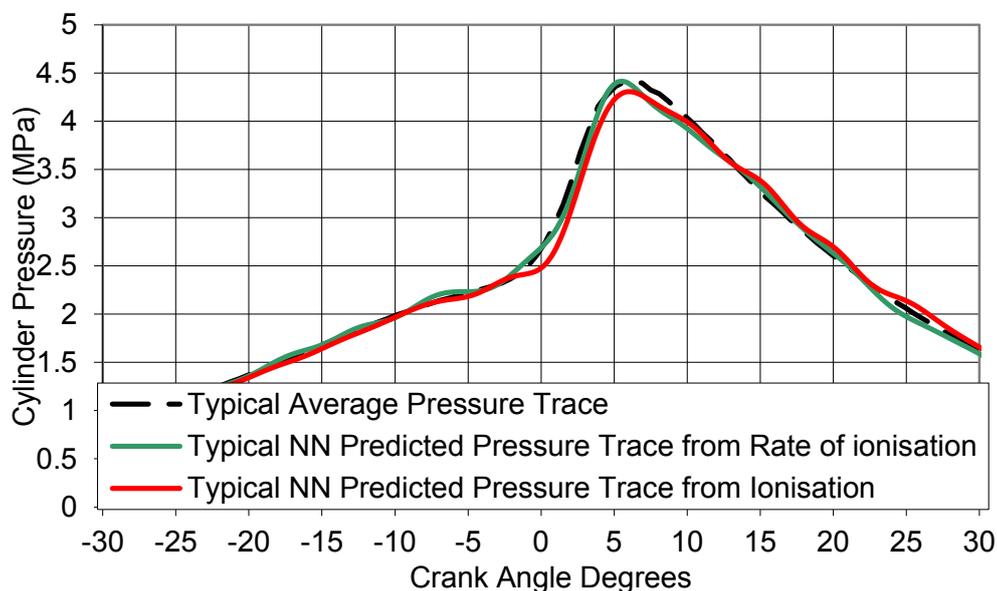
Figure 4.39 Comparison of 30 cycle Average Pressure Trace for AFR 11.75 and Neural Network Prediction of Pressure Trace form 30 cycle averaged ionisation current signals.

Figure [4.39] shows a comparison between typical pressure trace and predicted pressure trace from ANN trained from rate of change of ionisation current signal and ionisation current signals for test condition AFR 11.75. It is possible to see that even with a limited data

set it can predict the basic shape of the pressure trace, although there are errors around peak position and magnitude and around the point of inversion. From the predicted pressure trace it will be possible to estimate peak pressure position, peak pressure magnitude and combustion duration.
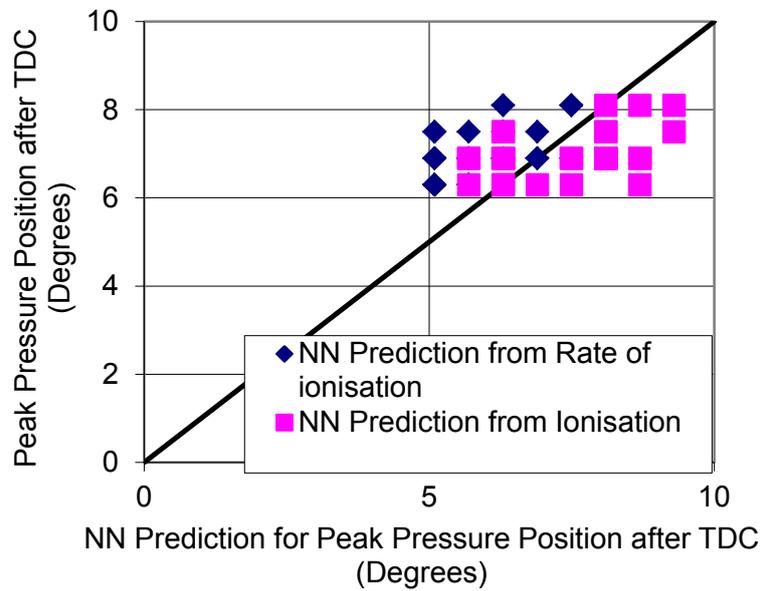
Figure 4.40 Correlation of Peak Pressure and Neural Network Predicted Peak Pressure Position for AFR 11.75

Figure [4.40] shows the correlation between peak pressure position and the ANNs predictions of peak pressure position for ANNs trained on rate of ionisation current and ionisation current signals. The mean absolute error for prediction from rate of ionisation current signal is 17.71% and 13.7% for ionisation current signal.
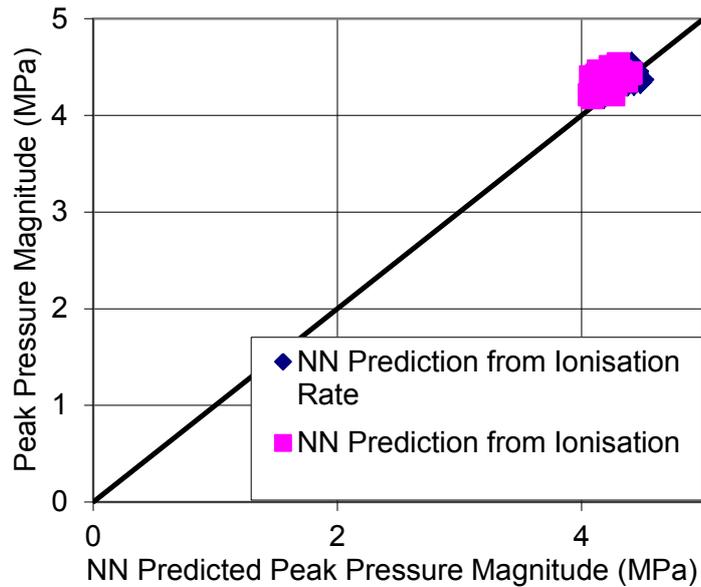
Figure 4.41 Correlation of Peak Pressure Magnitude and Neural Network Predicted Peak Pressure Magnitude for AFR 11.75

Figure [4.41] shows the correlation of peak pressure magnitude and predictions of peak pressure magnitude from ANNs trained from the rate of ionisation current and ionisation current signals. The mean absolute error for the ANN prediction trained from rate of ionisation current signal is 2.38 % and for ionisation current signal trained 3.72%. This level of accuracy is surprising considering the relationship between both peak ionisation current signal rate/peak ionisation current signals, and pressure magnitude did not show the same behaviour. For the purpose of comparison a prediction of the AFR 11.75 peak pressure magnitude was calculated from the results of AFR11.33 and AFR13.3 peak rate of ionisation current signal magnitude, the mean absolute error was 16%. Therefore trained ANNs are capable of finding relationships between ionisation current signals and pressure trace that at this time are not clear from visual inspection of the signals alone. The ANNs may well be finding relationships between several points on the input and output data to make these predictions whereas with the visual inspection we only consider a single point-to-point comparison. This result is a good demonstration of the use of ANNs for their capability of pattern recognition.
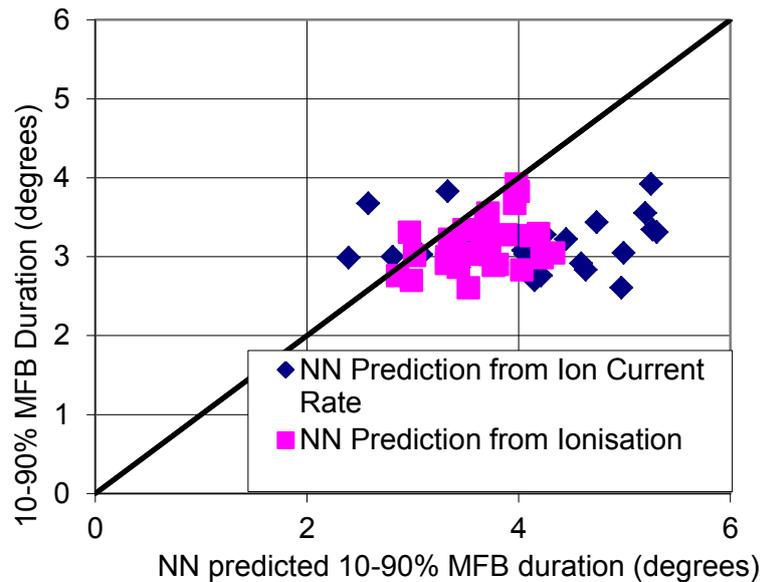
Figure 4.42 Correlation Between MFB duration and MFB duration calculated from NN prediction for AFR 11.75

Figure [4.42] shows the correlation between 10-90 MFB duration and predicted 10-90 MFB duration calculated from pressure traces predicted from ANN trained from rate of ionisation current signal and ionisation current signal. The mean absolute error for the ANN trained on rate of ionisation current signal is 34.66% and for the ANN trained ionisation current signal is 16.25%.

4.3.10 SINGLE INPUT-OUTPUT NEURAL NETWORKS

The ANNs so far have been constructed to accept the entire ionisation current or rate of ionisation current signals as inputs, to give an entire cylinder pressure trace output. A simplification to make feedback more efficient could be a reduced ANNs architecture that only accepts singular values for the point-to-point relationships. From the point-to-point relationships for combustion timing, the results of 75% ionisation current signal offered the smallest error as shown in Figure [4.30]. To train an ANN to predict peak pressure position from the timing values of the 75% ionisation current signal, input-output data pairs need to be presented to the ANN architecture. The ANN architecture is smaller than the previous. The numbers of hidden neurons have been reduced from 20 to 8 to get better representation of the data. The training data was again based data from the averaged cycles for AFR11.33 and AFR13.3 giving 60 sets of input output training pairs.
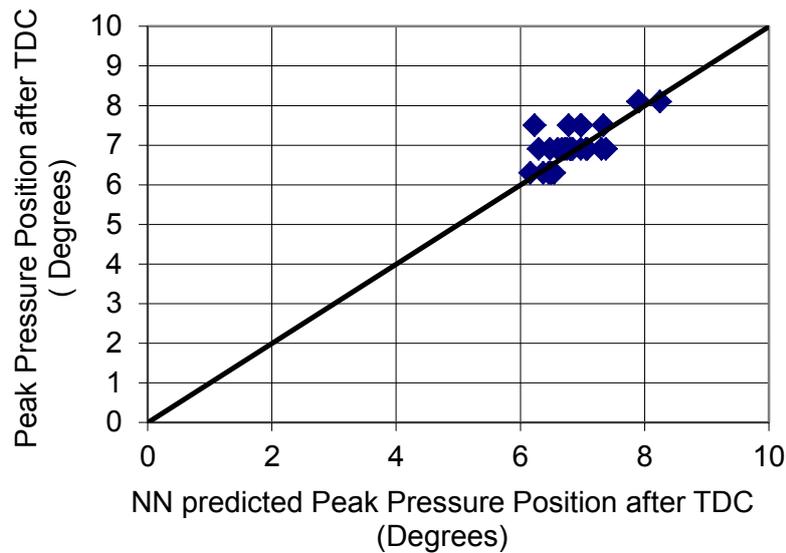
Figure 4.43 Correlation between Peak Pressure Position and Neural Network Prediction Peak Pressure Position from 75% ionisation current signal location

Figure [4.43] shows the correlation between measured peak pressure position and predicted peak pressure position for AFR 11.75 from the reduced ANN architecture trained on 75% ionisation current signal location. The average error of the prediction is 4%. This is an improvement over the prediction seen in figure [4.42] with an error of 16.25%. By reducing the input/output data to the key components of the signal that hold the relationship, the reduced architecture prediction can have better accuracy. By reducing the input/output data size the ANN is no longer trying to generalize relationships for the entire trace but for a single point. This allows the ANN to represent that point more clearly.

Overall the results from all the ANNs are promising. They have shown the capability of predicting the shape of the pressure trace, and show that there may be further relationships between ion current signals and pressure signals that are not initially obvious from inspection. The accuracy of the prediction of peak pressure magnitude is one instance where the ANNs have been able to find relationships that have been unclear. Where relationships are more clearly seen, like combustion timing, the reduction of the input/output data for the ANN has improved the performance. By limiting the data it allows just the key information about the relationships to be mapped in the ANNs weights. The drawback of ANNs is that the accuracy is directly related to the spread of data used for training. To develop an applicable feedback system, full range engine data would be needed. The ANN can generalize its knowledge to predict outputs for unknown inputs between the training data sets. However,

the further the unknown inputs are away from the training knowledge, the accuracy would decrease.

4.3.11 COMPARISON OF FEEDBACK

This study has shown several methods for gaining feedback information from ionization. The ionisation current signals can be used as a whole or in part with ANNs. This allows prediction of pressure trace or its parameters. The ionisation current signal could also be used directly. Ionisation has shown relationships with pressure trace proving it to be related to combustion. The purpose of a feedback signal is for supplying control architecture with information. Feedback control architecture's performance will be affected by the efficiency and accuracy of the feedback signal. It can be seen already that the ANNs performance can be improved by carefully selecting the appropriate input and output data.

Table 4.3 Comparison of feedback method variance compared to peak pressure position

	Real peak pressure position	NN using entire ion trace predicted peak pressure position	NN 75% value ion trace predicted peak pressure position	Direct ion trace 75% value
Variance (CAD)	0.275	1.275	0.248	0.275

Table [4.3] shows the comparison of the variance for the three feedback methods compared to peak pressure position variance. The direct ion trace 75% value offers a feedback with the closest variance match to that of real peak pressure position. The ANN prediction using the entire ionisation current signal offers the worst. However the reduced input, 75% ionisation current signal location ANN, shows substantial improvements. However, the predictions from the reduced input ANN have lower variance than the real peak pressure position. The lower variance indicates inaccuracy in the prediction. The ionisation current signal is a direct feedback of the products of combustion. It offers the closest match to combustion timing variation than any predictions made from it. Using the ionisation current signal itself can therefore offer a more efficient feedback for combustion timing than converting to cylinder pressure for information.

4.3.12 CONCLUSIONS ON IONISATION

Ionisation current and rate of ionisation current signals both show a relationship of timing to combustion. This has been demonstrated to varying accuracy at points along the signals in comparison to peak pressure position. Overall the ionisation current signal was seen to have better accuracy for information on combustion timing compared to the rate of ionisation current signal. Ion current has shown the ability to predict the location of peak pressure position to an average error at the three operating conditions of 5.24%. This shows that ion current has a relationship with combustion timing and could be used for timing feedback. Further conclusions from the comparison of the locations compared are:

• Start of ionisation current signal has a larger variability than peak pressure position. The start of ionisation current signal is affected not only by the timing of the combustion but also the location of the combustion.

• Taking points further up the rising edge of the signals offers more accurate timing results as it reduces the effects of base line noise and potentially local in-homogeneities from start of combustion. Points between 50 and 75% have offered the smallest error on timing predictions.

• Peak position of rate of ionisation current signal and ionisation current signal tended to have slightly larger variation than results from locations of 50% and 75%.

• The ionisation current signals have also shown a relationship between the duration of ionisation and the duration of the 10-90% MFB for the three different AFRs. As the 10-90% MFB increases and decreases cycle-to-cycle, the ionisation current signal follows the same trend. This shows that there is a potential relationship between ionisation current signal duration and combustion duration.

• The rate of ionisation current signal has shown strong spikes within the peak of the signal, this represents rapid changes in the rate of ionisation. This characteristic maybe linked with combustion occurring at multiple sites of the cylinder. This relationship could give the ability to detect knock, and auto ignition combustion.

• The peak value of the rate of ionisation current signal tends to show an inverse relationship with timing location of peak pressure. This shows a potential relationship for combustion duration.

• Leaner AFRs were seen to have the effect of reducing the magnitude of ionisation current signal and of decreasing duration.

• Peak magnitude of ionisation current signal is affected not only by AFR but also timing of combustion and change in density due to volume change.

• ANNs can be trained to predict pressure trace from ionisation current signals and rate of ionisation current signals. From these predicted pressure traces it is possible to estimate combustion duration, peak pressure position and peak pressure magnitude.

• The ANN prediction of peak pressure magnitude had a much smaller error or 2.38% compared to prediction from point to point relationships 16% error. This shows ANNs capability of pattern recognition and capturing relationships that may not be clear through inspection of the signals.

• ANNs performance was seen to improve if input/output data can be reduced to the key parameters of interest with a particular relationship.

• The overall pressure trace can offer visual feedback for peak pressure position, peak pressure magnitude, knock and combustion phasing. Through calculation, combustion duration and start of combustion can be estimated.

• Feedback parameters directly obtained from ionisation current signal match the variance of combustion timing with better accuracy than predictions of peak pressure position from ionisation current signal through ANNs. The ionisation current signal is therefore a potentially efficient feedback for two stroke HCCI combustion.

• The mobile ionisation system's feedback signal has been validated for a range of in-cylinder feedback information related to combustion characteristics. The mobile ionisation system can be used as an experimental research tool for the further development of the adaptive control architecture.

5. DEVELOPMENT OF THE ONLINE ADAPTIVE CONTROL ARCHITECTURE

For the development of the online adaptive control architecture the programing software package Matlab and Simulink was used. Matlab is a high level language and interactive environment that allows computationally intensive tasks to be programmed and performed faster than traditional programming languages such as C, C++ and FORTRAN. Matlab has thousands of pre-determined functions that facilitate many simple operations. By combining these pre-determined functions it is possible to use this programming language to investigate and develop mathematical tasks very quickly. If a function that is required is not pre-determined it is also possible to modify an existing or write your own function and add it to the Matlab library. By having such a range of functions already written and available it can save time in writing and debugging.

Simulink is a part of the Matlab package, it offers an environment for multi-domain simulation and model based design for dynamic and embedded systems. It provides an interactive graphical environment and a customizable set of block libraries that allow design, simulation, implementation, and testing for a variety of time varying systems. Including communications, control, signal processing, video processing and image processing. Of main interest with the work conducted is the development of advanced control structures using fuzzy logic and ANNs. Matlab and Simulink both have the capability and toolboxes that enable this type of programming to be achieved with pre-defined functions and libraries.

The fuzzy logic toolbox is a collection of function that is built on the Matlab technical environment. The functions allow the user to create and edit fuzzy inference systems within the Matlab environment with the fuzzy Graphical User Interface (GUI) or alternatively command line code. The resulting fuzzy systems can be integrated into the Simulink environment for simulation and control.

(removed for copywrite)

Figure 5.1 Matlab fuzzy logic graphical user interface (145)

The GUI is the most convenient method for designing and implementing fuzzy inference systems. The Matlab fuzzy GUI has five main windows, figure [5.1]:

- Fuzzy Inference System (FIS) editor.
- Membership function editor.
- Rule editor.
- Rule viewer.
- Surface viewer.

The FIS editor is the main part of the GUI and is used to determine the highest level parameters of the system such as the type of inference mechanism, number of inputs and outputs. From this window it is possible to open any of the further windows by selecting the appropriate box on the fuzzy system diagram or drop down menu.

There is a separate MF editor window for each input and output. The MF editor allows the user to define the number, type, and shape of MFs used. The range of the universe of discourse can also be defined using this editor. The rule editor is used to generate the rule base for the fuzzy system, it allows the freedom to use different fuzzy operators with any combination of input and output MFs. Each rule is stored in a rule table and can be viewed through either the rule viewer or the surface viewer. The rule viewer and surface viewer are purely read only tools and are used for evaluating the fuzzy inference system. By using the rule view it is possible to see how each rule is activated and how they affect the consequence and final crisp outputs of the system.
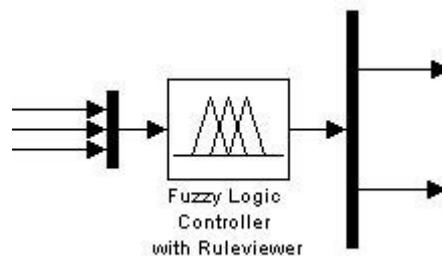
Figure 5.2 Example of a Simulink fuzzy logic controller

Once a FCS is designed it can be integrated with Simulink. The FCs can be saved as a fuzzy structure either in the workspace or to file. Once it is stored in the workspace it can be referenced to from the Simulink model, figure [5.2]. Once imbedded into the Simulink block, the fuzzy system can be evaluated through simulation with a model or system to be controlled.

The ANN toolbox is another software package for Matlab and Simulink that allows the user to design, implement, visualise and simulate ANNs. The toolbox allows its functions to be used at command line level as well as through several GUIs. The graphical tools make an environment where standard ANNs can be created for tasks such as pattern recognition and data fitting. For more advanced implementations such as system identification and control of non-linear systems, command line code offers more freedom in design. Once a ANN is created it is often saved in the workspace as a net structure.
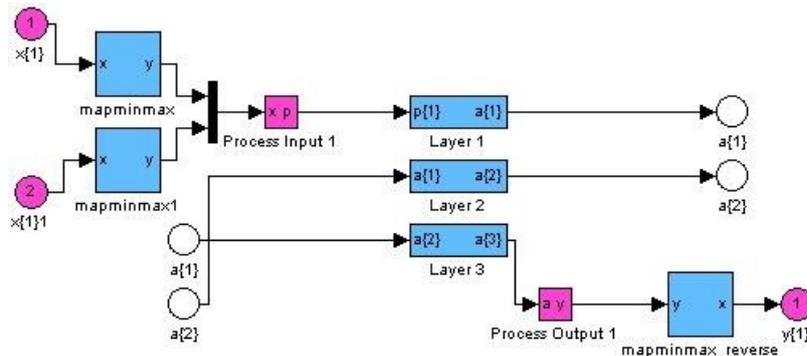
Figure 5.3 Example of a Simulink neural network

Once a net structure is finished being trained it can be converted into a Simulink implementation of the ANN, figure [5.3], or an m-file function to automatically process data arrays. Once in the Simulink environment, the ANN can be evaluated or used as controller in a simulation of a model like the fuzzy system.

The Matlab and Simulink software offers a set of powerful programming tools that can allow the development of the proposed control architecture. However there are some problems with the implementation of these toolboxes and software packages to achieve the goals. These problems are in the form of limitations with Matlab and Simulink and the ANN toolbox.

5.1 LIMITATIONS OF THE SOFTWARE PACKAGE

The ANN toolbox is very powerful development tool. It allows the user to use a wide range of popular ANN architectures while also allowing the user to define the attributes of the ANNs such as transfer functions, number of feedback loops, number of neurons, and number of hidden layers. A major concern with the ANN toolbox is the requirement for all major development and training to be conducted in the Matlab environment and not in the Simulink

environment. Once an ANN is configured and integrated into a Simulink model there is no facility with Simulink or Matlab to adapt the ANN any further. The package is limited to offline adaption only with Simulink implementations. As the aim is to develop a control structure, the Simulink environment is essential for its control development and ability to link with third party software. Therefore, a method to manipulate Matlab and Simulink is going to be required to allow the integrated Simulink ANN to adapt online if the original control concept is to be developed and evaluated.

Matlab and Simulink software is designed to operate on a single processor. For example when running a Simulink model, all the calculations are calculated in sequence. If part of a model takes longer to calculate, the program will wait till this is finished before moving on to the next time step. Therefore, the simulation time scale is not real life time scale, and all processes are sequenced and run through a single processor, not run in parallel. The desired control architecture ideally requires two processors, one processor for the control to run continuously and a second for the ANN weight update calculations. Once the new weights are calculated they could be updated at a point where the main processor is at idle. However, this is not possible with conventional running of Simulink. As the purpose of this research is the development of online adaptive control architecture this limitation does not prevent the development, it only limits its application for real time implementation. Matlab and Simulink offer a powerful software environment for developing and testing the control architecture. If in the future the control requires real time implementation there are methods for implementing Simulink based control into hardware such as dSPACE (137). Using the software and facilities currently available it is possible to first develop the control architecture.

5.2 MANIPULATING NEURAL NETWORKS IN MATLAB AND SIMULINK

To develop the desired control architecture it is possible to use the Matlab and Simulink software tools but some adaptation on how they operate is required. Using the ANN toolbox in the Matlab environment offers the fastest method to develop ANNs. The toolbox allows the user to change ANN type, structure, number of layers, number of neurons per layer, transfer functions, learning algorithms and initialisation methods. This offers a very broad development foundation. The command line code can be written into an M-file so that a program can be written to investigate best performance based on:

- Size of net.
- Number of layers.
- Type transfer function.
- Initialisation, normalisation ranges.
- Different ANN structure, feed forward, time delay, recurrent as required.

By investigating these variables it will be possible to find a ANN that has a balance or performance and speed or training. Once the desired ANN is found it can be converted into the Simulink environment. However, in the Simulink environment the ANN will have the desired model performance only for the knowledge acquired during initialization and offline training. To achieve the desired adaptive control architecture there will need to be several alterations to the standard ANN structure in the Simulink environment.

5.2.1 CREATING UPDATABLE WEIGHTS
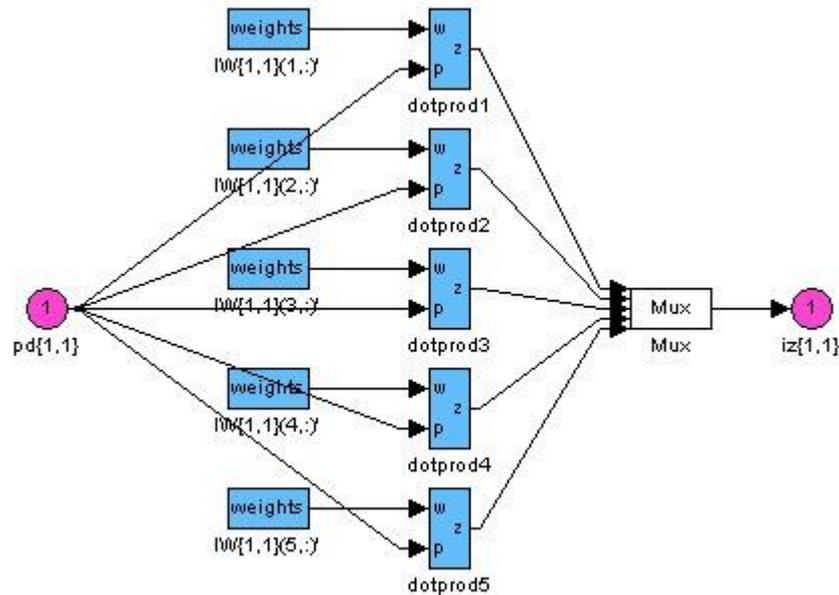
Figure 5.4 Example of a single layer of Simulink neural network with standard weights and bias values

When an ANN is converted into the Simulink environment all the weights are stored in "Constant source block", figure [5.4]. The constant block allows a matrix of values to be stored and outputs of these values at each time step. The values within the constant block cannot be altered during simulation. Therefore, the ANN is fixed and not adaptable. To the authors knowledge there are no current methods available to perform adaption of ANNs in Simulink. Therefore, for this work the following method has been developed.

To adapt the ANN these constant blocks need to be removed and replaced by another storage mechanism that allows adaptation to the values throughout the simulation. Simulink has a data store block-set that allow information to be passed between different sub models without wired links. Therefore, by using the data store block-set to manipulate how the initial weights and biases are stored in a Simulink model could offer a mechanism for ANN adaption. The data store block sets consist of the following blocks:

- Data store read.
- Data store write.
- Data store memory.

These three blocks work together to allow data to be written, stored and read without connection across the Simulink model.
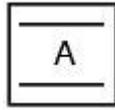
Figure 5.5 Data store memory Simulink block

The data store memory block figure [5.5], defines and initializes the data to be stored for a given user defined variable "A". The Block requires locating in the top level system or at least a sub system above all its associated read and write data store bocks to operate correctly.
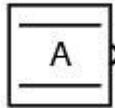
Figure 5.6 Data store read Simulink block

The data store read block figure [5.6], allows the data stored in the corresponding data store memory block to be read at every user defined sample period.
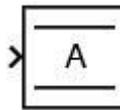
Figure 5.7 Data store write Simulink block

The data store write block figure [5.7], updates the data stored within the data store memory block at each user defined time step.

By using these three blocks it is possible to change the constant weight system to a dynamic weight system that can change throughout the Simulink simulation.

As the number of weights and bias values in an ANN can be very large, it is not practical to use a read, write and memory block for each individual weight or bias. It is useful to store the weights in matrix form then separate at the ANN using a sub matrix block.
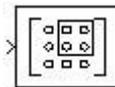
Figure 5.8 Sub matrix Simulink block

The sub matrix block figure [5.8], allows a user defined proportion of an input matrix to be passed through to the output. This can be a row, a column, single value or a sub matrix or partial rows and columns.
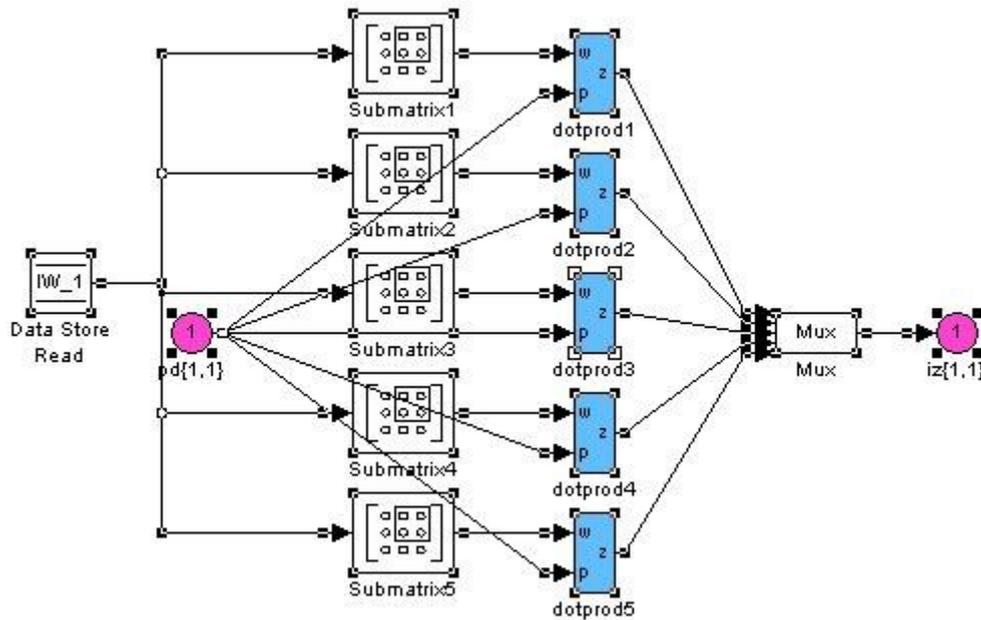
Figure 5.9 Example of Simulink neural network with modified adjustable weights and bias values

The resulting adapted ANN, figure [5.9], allows for the weights to be stored in a data store memory block at the top level. The weights can be read by the data store read block at each control cycle and passed to the ANN architecture. The weights can be updated by using the data store write block if appropriate new weighs can be calculated for the ANN.

5.2.2 TRAINING CUSTOM NEURAL NETWORKS IN SIMULINK
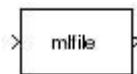
Figure 5.10 Level 2 M-File S-Fuction Simulink block

Updating the ANN weights requires a retraining of the ANN against new input/output data sets. As the ANN is fixed and being used to process data this is not possible in the Simulink environment. ANN adaption or training can only be conducted in the Matlab environment. A method is required to attempt to bring the Matlab training code into the Simulink environment. Simulink has a variety of blocks that can achieve this task depending on the complexity of the Matlab program required. As this task requires a program using several pre-determined function from the ANN toolbox the most appropriate Simulink block is the " Level-2 M-File S-Function Block" figure [5.10].

The Level-2 M-file S-function allows the use of Matlab programming language to create custom Simulink blocks with multiple input and output ports. The custom blocks are capable of handling any type of signal produced in Simulink, including matrix and frame signals of any data type. The S-function file provides the implementations of call-back methods for determining the block attributes (e.g., ports, parameters, and states) and behaviour (e.g., the block outputs as a function of time and the block inputs, states, and parameters). By creating an S-function with an appropriate set of call-back methods, it is possible to define a block type that meets the specific requirements for the application.

A Level-2 M-file S-function must include the following call-back methods:

- A setup function to initialize the basic S-function characteristics.
- An output function to calculate the S-function outputs.

The S-function can contain many other methods, depending on the requirements of the block.

For the task of training ANNs the following call-back methods are used:

- Function "m-file name".
- Function Setup.
- Function Output.
- Function Terminate.

The first function allocates a name to the S-function block; this name is used to call back the function in the Simulink S-function block. Under the setup function, the basic attributes are set:

- Number of input ports.
- Number of output ports.
- Parameters for each port including, data type and size.
- Sample time.
- Simstate compliance.

The output function is where any Matlab language code is written for the calucation of output data. For the ANN adaption, the Matlab training functions will be used. Once the calculation of the new weights is completed they would be passed to the pre-determined output ports.

The terminate function calls the end of simulation for the S-function block and cleans up all the memory stores for the file to begin a fresh simulation on the next active time step.

This M-file S-function block can therefore offer the ability to bring the Matlab programming code into Simulink for calculating weight adaption.

There are some complications with using the S-function method to complete the weight update for the Simulink ANN. As the ANN in Simulink was originally created in the M-file language it is stored as a net structure in the workspace of Matlab. Ideally the stored net structure would be called into the training function and used again to adapt the origianl weights and bias values to new training data. However, it is not possible to call net structures from the Matlab workspace into the Simulink environment. Therefore, the only alternative is to initialise a brand new net structure for each S-function cycle. The new net structure would need to be of an identical structure to the original created in the Matlab environment. To mimic the original net structure after the original offline training, the random weights/biases need to be replaced by the current weights/biases being used in the Simulink environment. This process is critical to maintain previous trained knowledge and therefore needs to occur before any further training can take place.

To replicate the original net structure, not only the weights/biases need to be replaced but also the original initialisation of the ANN also needs to be replicated. The standard initialisation of a ANN is based on the range of the input and output data. This data is normalised between -1 and 1. When initialising the new net structure its initialisation is based on the range of the new training data. If there is a mismatch between the range of the original data and the new training data, then the scaling between -1 and 1 will be different. Therefore, when the new weights are updated they will be optimised for a different scaled range to that set in the original Simulink ANN. This can have a detrimental effect on the accuracy and performance. To avoid this problem the standard initialisation has to be overridden with some extra Matlab code. If we inspect the full possible range of the input and output data we will find they have finite limits that are defined by the physical system. For example, we know that peak pressure position can only really be between TDC and BDC. If the range of all the inputs and outputs can be predetermined, these ranges can be used to set the range of normalisation for the initial ANN training as well as subsequent S-function initialization. All new data should theoretically fall within this physical range eliminating the mismatch problem. Once the new initialised net structure is identical to the original structure, training can commence with new data sets. Once the training is completed the new weights can be passed to the data store write blocks to update the Simulink ANN. Therefore, this method is facilitating online training. New training data can be stored with in buffer memory blocks; the size of the buffer can determine the size of the training sets. Single values

would mimic true online training, with training occurring every cycle. Larger data sets would mimic batch learning, but through an online approach.

5.3 SIMULINK ADAPTIVE CONTROL ARCHITECTURE
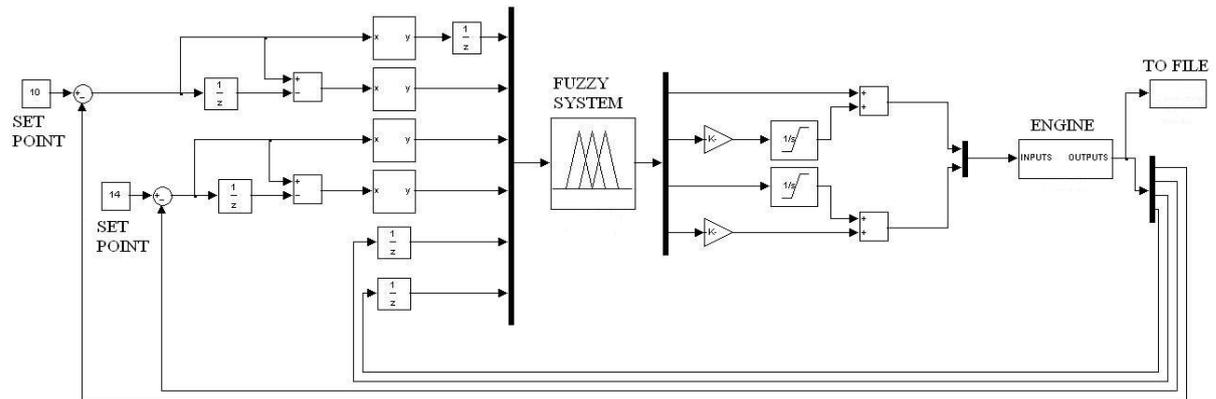
Figure 5.11 Fuzzy feedback controller

Using the fuzzy logic toolbox in Simulink allows the fast development of a fuzzy feedback controller, figure [5.11]. The main fuzzy system is created in the GUI as described at the beginning of the chapter. Once the fuzzy system is developed it can be called into a fuzzy block in the Simulink environment. The Fuzzy block loads a user defined FCS file from the Matlab workspace at the start of a simulation. The Fuzzy block will then run the fuzzy system in the Simulink simulation. The input and output signals are application specific and are required to be determined in the the development of the Fuzzy system using the GUI toolbox. For the engine control application where spark timing and AFR is controlled, the following input data has been used:

- Desired combustion timing set point.
- Desired AFR set point.
- Combustion timing error.
- AFR error.
- Rate of change of error signals.
- Engine state conditions, RPM, MAP.

 The outputs for the SI engine control over combustion timing and AFR are:

- Feed forward terms for spark timing and injection pulse width based on engine state information.

- Spark timing and injection pulse width feedback trim control for error reduction based on error and rate of error terms.

The FLC uses a fixed nonlinear model of the engine based on input and output states to determine the feed-forward control components. The feed-forward components improve the FLC control over the engine by allowing the feedback control components to remain within a smaller range for each operating condition. The feed-forward component accounts for the engine's nonlinearity, while allowing the feedback control to trim error due to disturbances and model/plant mismatch. With the FLC having a fixed model based on engine states, it allows the inverse modelling of the engine's dynamics to be shared between the FLC and the ANN. Further detail on the design and calibration of the FLC will be discussed in the following chapter.
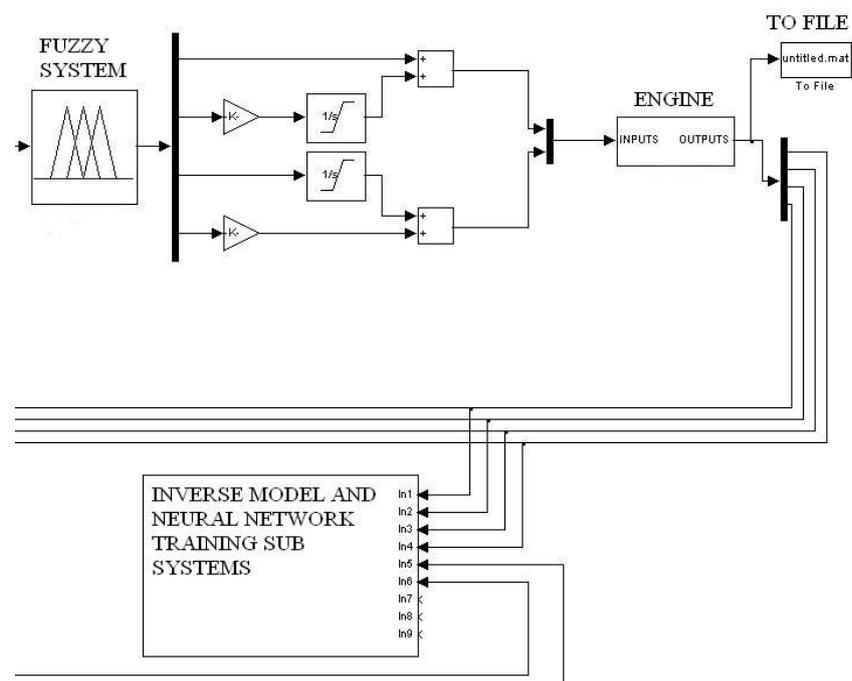
Figure 5.12 Subsystem for inverse model and neural network training sub systems

To transform the basic FLC into an adaptive control architecture several ANNs and their training and weight update system need to be added. When creating complex systems using Simulink it is often useful to split the system into subsystems. Figure [5.12], shows the main

subsystem for the inverse ANN of the engine. The Simulink programming within this subsystem includes:

- ANNs for ideal control prediction.
- ANNs for estimating prediction accuracy.
- Logic system for triggering ANNS training depending on prediction accuracy.
- Buffer system for collecting training data for ANN's adaptations.
- M-file S-function blocks for ANN training.
- Weight and bias update systems.

The adaptive control architecture relies upon the ANNs learning the difference between the inverse of the engine dynamics and the current fixed FLC model. To achieve this, there needs to be an accurate inverse model of the engine.
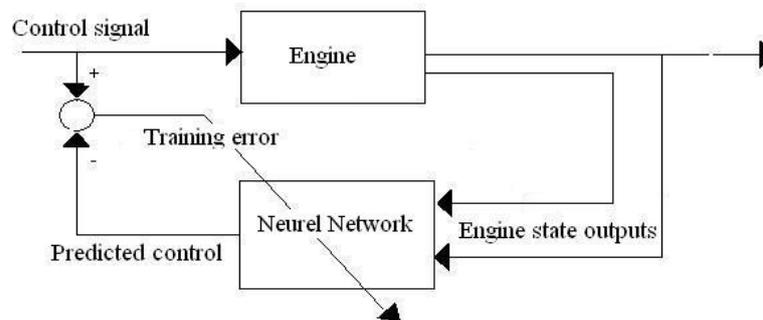
Figure 5.13 Inverse model training

The ANN in figure [5.13] can learn the inverse dynamics between the engines state and the control variable by using the engine state outputs as inputs to the ANN and the control signal as the desired output. Through training, the ANN will learn the relationships between the control signals and the engine behaviour. Once the training error is reduced over a range of operating conditions the ANN will be able to make predictions on what the ideal control signal should be for a given set of engine states.
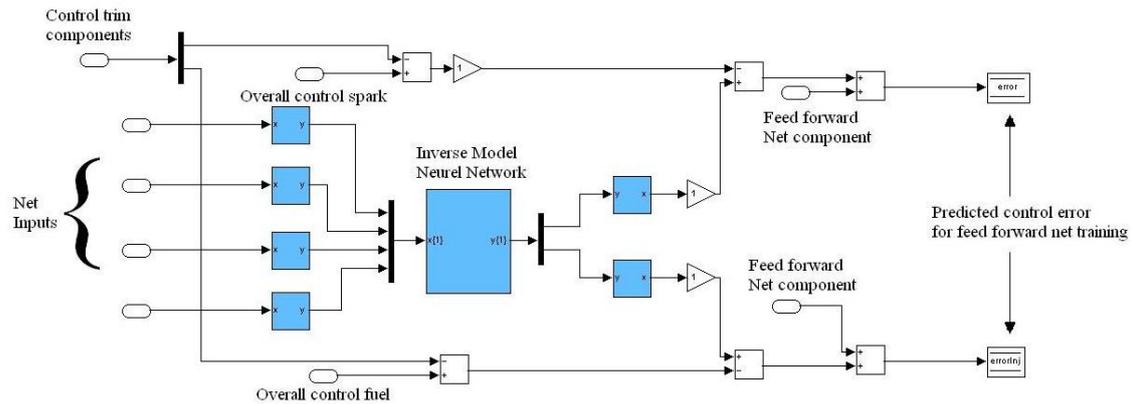
Figure 5.14 Inverse neural network for ideal control prediction

Figure [5.14] shows the layout of the inverse ANN model for control signal prediction. Overall the control signal is based on three components; the fuzzy model component, the fuzzy feedback trim component and the feed forward ANN component. The feed forward ANN is required to learn the difference between the current control signal and the ideal control signal. The diagram in figure [514], demonstrates how the inverse ANN model is used to derive this control error term for the feed forward ANN. The inputs to the inverse ANN need to give information on current engine state and the desired control feedback set points. This allows the ANN to make a prediction based on the desired control requirements and the current engine state. This predicted control is then compared to the current control signal to produce the control signal error. The comparison of the two control signals needs to be carefully considered. Ideally we would like the feedback fuzzy controller to have as little input as possible. If the fuzzy feed forward model was an ideal model of the system and there were no disturbances, then the feedback control component would be a very small part of the overall control signal. If there is a large degree of mismatch between the engine dynamics and the fuzzy feed forward model then the feedback control will need to compensate for this error. The feedback control will become less effective at control over disturbances if it is needed for compensating large model/plant mismatch. If the mismatch becomes too large the feedback control will saturate and be unable to offer any control against disturbances. Therefore, if the feed-forward ANN can learn the difference between the fuzzy model and the inverse model this would allow a complete feed-forward model between the two control systems. The combination of the fuzzy- feed forward model complemented with the feed-forward ANN will therefore relieve the reliance of the feedback controller to reduce model/plant mismatch and allow more effective control over disturbances.

$$\left[C_{Predict} - [C_{overall} - C_{trim}]\right] + C_{net} = C_{net\_train} \qquad (5.1)$$

- $C_{predict}$ = Predicted ideal control for a given state and desired set point
- $C_{overall}$ = Overall control component for the given state
- C_{trim} = Fuzzy feedback control trim component
- C_{net} = Feed-forward ANN component
- C_{net_train} = Feed-forward ANN training data

Equation [5.1], is used for calculating the error term for the feed-forward ANN. The Feed-forward ANN training data is calculated from the predicted control minus the difference between the overall actual control signal and the feedback trim component, plus the current feed-forward ANN component. Using this equation allows the feed-forward ANN to learn the difference between the combined fuzzy model and feed-forward ANN component with the predicted ideal control. If the error is zero, the current feed forward ANN component is stored. If the addition of the current feed-forward ANN component was not in the equation, when the control error is equal to zero, the error term would be equal to zero. If the current C_{net} value was not equal to zero, this would cause the next training of the feed-forward net to train incorrectly and cause the training to oscillate between including a network feed-forward component and removing it. This equation relies heavily on an accurate inverse model to form the prediction. If the prediction is inaccurate then the feed forward ANN will be training to add a feed forward component that would be incorrect. Therefore it is important to insure the inverse ANN model also trains online when required.
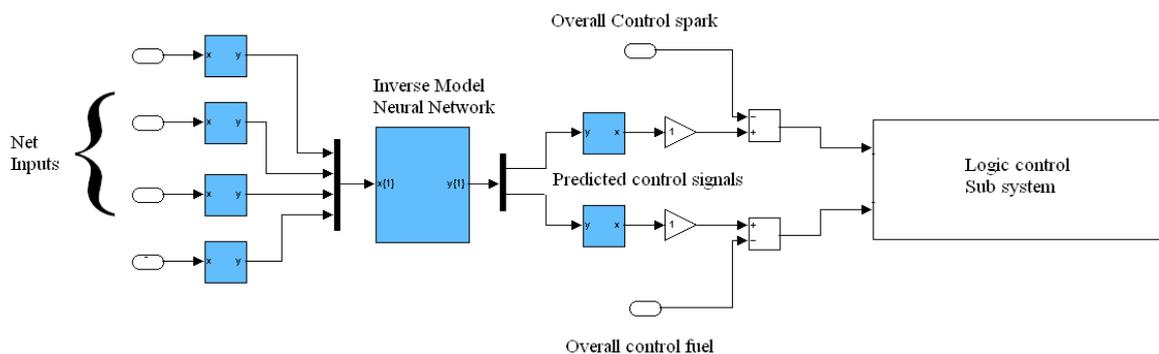
Figure 5.15 Method for controlling inverse neural network training

Figure [5.15], demonstrates how the inverse ANN can also be configured to assess the current accuracy of the ANN model. By using inputs for current state and current feedback signals instead of desired set point, the ANN should give a prediction of the current control

signal to achieve that specific state. As we know the overall control signal used, it is possible to find the difference between the predicted control and the actual control. As the ANN is not just trained on ideal conditions it is able to predict control variables for any given state within the input range. The error of the ANN is then passed into a logic control subsystem.
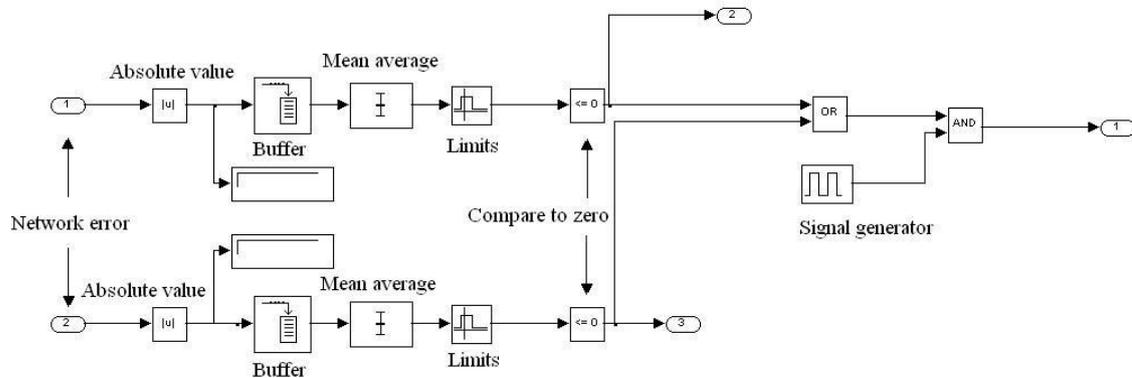
Figure 5.16 Logic control subsystem

Figure [5.16], depicts the logic control for the training of the inverse model ANNs. The predicted error signal is transformed into absolute values and buffered to a user defined amount. The average of the buffered array of error signals is taken and compared to limits for expectable error. If the error is reasonable a Boolean output determines if the ANN requires training. A signal generator block is used to create a timed Boolean output to ensure the ANN can only train at periodic time gaps allowing time for new training data to be collected. Figure [5.16] shows the logic used if more than one ANN is used for constructing the inverse model. In some applications two smaller ANNs can outperform one large ANN for training speed and accuracy. If more than one ANN is used to create the inverse model they may not all need training. The overall Boolean control will only activate the ANN training files if the signal generator is in the correct phase, and one of the ANNs is out of acceptable error limits. To reduce computation expense the separate Boolean operator for each ANN will activate different parts of the code within the training files. This method insures that ANNs train on new training data and only ANNs out of expectable error limits are trained. This simple logic control reduces the likelihood of ANNs being over trained on a specific set of data or due to training after the ANN reaches acceptable performance.
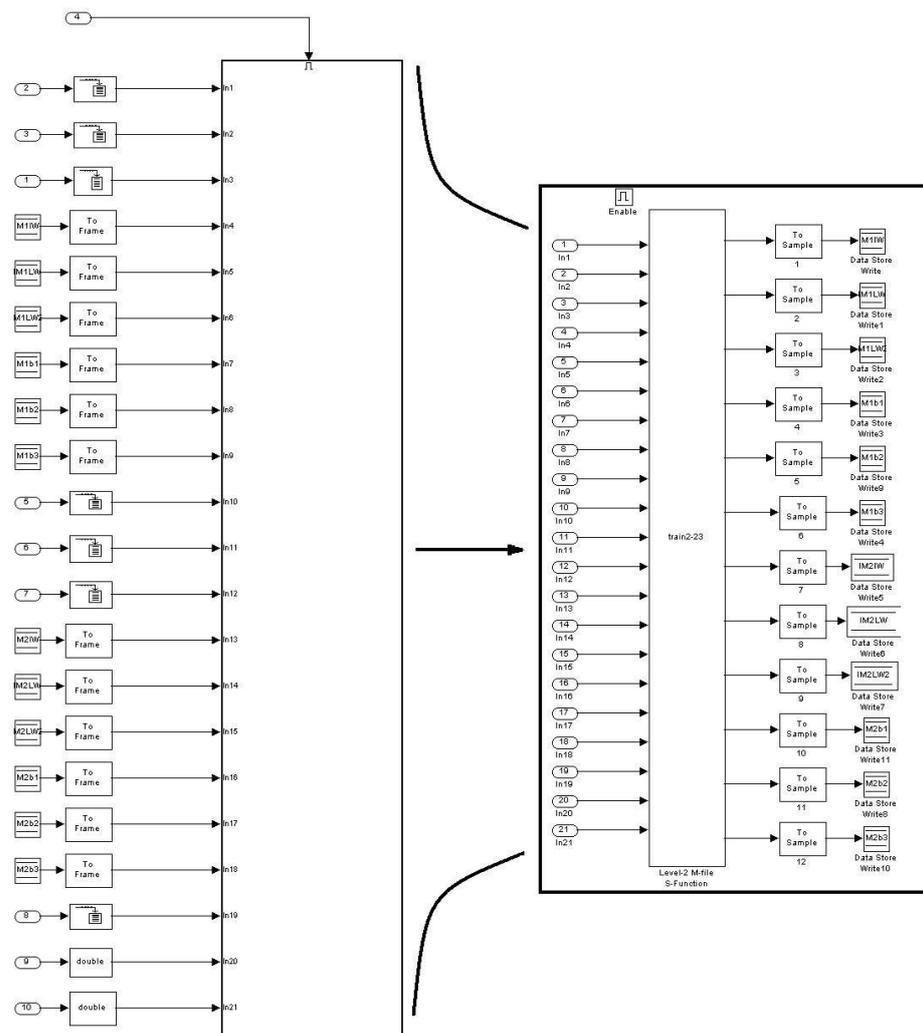
Figure 5.17 Training file subsystem

Figure [5.17], shows the subsystems for the training M-file S-function block for the inverse ANN. The M-File S-function block is placed within a triggered subsystem that is controlled by the overall Boolean operator dependent on ANN error and the signal generator phase. The inputs to the subsystem comprise of buffered training data, current ANN weights and bias values and individual ANN Boolean control signals. The amount of buffered training data is application specific. Too little data will not offer the ANN a spread of data to train with, to many data points will increase the training time required and potentially risk overtraining. A multi-layer ANN will have a matrix or weights and biases for each layer. When using the data store block set it is possible to read these matrices from the data store memory file for input to the training file. Once the subsystem is activated the M-file S-function will call a user defined program. Once the training program has completed the outputs are the new weights and bias matrices for the ANNs. The new matrices can be used to overwrite the old weights

and biases in the data store memory, allowing any data store read block to begin using the new data.

Within the M-file S-function block a number of process are required before the ANN can be trained or adapted. Once the inputs are registered and properties initialized, the input and output data is randomized.

```matlab
%%network 1
%peak pressure position
i(1,:)=(block.InputPort(1).data')
%engine speed RPM
i(2,:)=(block.InputPort(2).data');
%Mass Air Flow
i(3,:)=(block.InputPort(10).data');
%spark timing
o=(block.InputPort(3).data')


% randomize sequence
sze = size(i);
col=randperm(sze(1,2));
i=i(:,col);
o=o(:,col);
```

Figure 5.18 Matlab code for initialising input and outputs and randomising their sequence

The Matlab code in figure [5.18] demonstrates how the input and output data is combined into input and output matrices. Once in the matrix form the size of the matrix is called and used to create an array of random numbers from 1 to the maximum number of data pairs in the training data. The input and output matrices are then reorganized to the random number array. This method allows the input and output data to be randomized, but allows input and output data to remain in pairs. Having the option to randomize the sequence of the training data can help prevent the ANN from overtraining. If an ANN is trained on a sequence of numbers there is a potential for the ANN to learn the pattern of the sequence of number more than the relationship between the data pairs. By randomizing the data it removes this potential problem.

Once the data is randomized, the inputs and outputs need to be scaled as previously mentioned.

```
%input scale for engine speed RPM
ps1.name='mapminmax';
ps1.xrows=1;
ps1.xmax= 6000;
ps1.xmin= 1000;
ps1.xrange=5000;
ps1.yrows=1;
ps1.ymax=1;
ps1.ymin=-1;
ps1.yrange=2;


i(2,:)=mapminmax('apply',i(2,:),ps1);
```

Figure 5.19 Matlab code example for scaling a set of data

The Matlab code in figure [5.19] demonstrates how for the input 'engine speed' a scaling is performed between 6000 and 1000 to 1 and -1. The scaling is achieved using the 'mapminmax' function that is an pre-written function available in Matlab for this specific task. Once the input and output data is in the correct format the ANN requires initiating.

```
inputr=[-1 1; -1 1;-1 1];
outputr=[-1 1];
net=newff(inputr,outputr,[20 10],{'tansig' 'tansig' 'satlins'});
```

Figure 5.20 Matlab code example of initialising a new feed forward neural network

The Matlab code in figure [5.20] demonstrates the initiation of a new feed forward ANN with three inputs. Newff is a predefined function from the Matlab ANN toolbox. Using this function allows the user to define input and output data range, number of hidden layers and neurons and what transfer functions are used in each layer. For the new ANN 'net' the input range for all three inputs and single output has been forced to -1 to1. The number of hidden layers and neurons are specified in the array "[20 10]", this denotes two layers, one of 20 neurons and one of 10 neurons. If more or less layers or neurons are required the magnitude and number of entries in the array change accordingly. This ANN has also been specified to have the two hidden layers as tan sigmoid functions and a saturated linear output neuron. The choice of the activation functions for each layer is purely dependent on the requirements for the ANN.

For example, the saturated linear output function offers a level of safety against an incorrect ANN adaption. If an ANN trains badly the ANN output can saturate at a predetermined value within safe limits. If the output had no saturation and the ANN trained badly the output could output to infinity. This is undesirable if it is integrated into a control system.

All the weights and bias values of the ANN are in matrix form. Replacing the new initialized weights with those input from Simulink is a simple process of overwriting the current values. After the training is completed the weight and bias matrices can be called out and output back to Simulink. To see in detail the full M-file S-function files for the MIMO control, please see appendix [B].
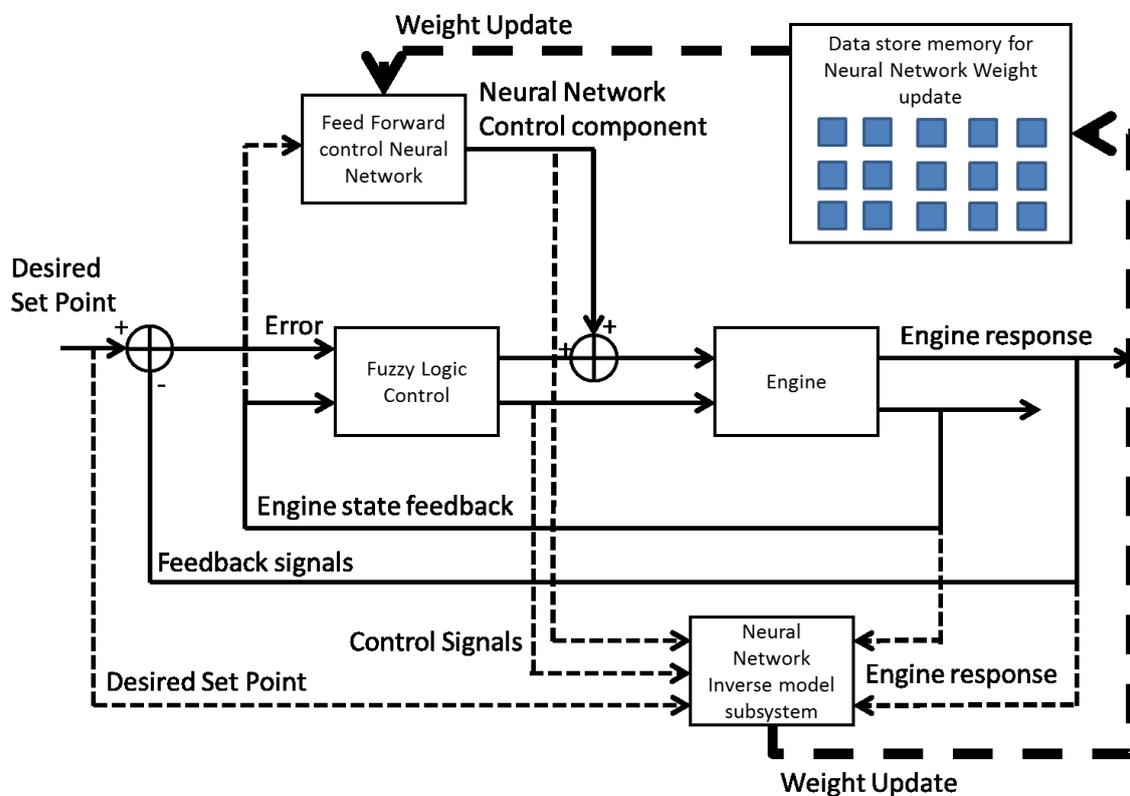
Figure 5.21 Simplified overall control structure

The final component of the adaptive control architecture is the feed-forward ANN. Figure [5.21] demonstrates how the feed forward ANN is incorporated with the FLC system. The inputs to the feed forward ANN are the same as the engine state feedback for the FLC. The desired outputs of the feed forward ANN will be the training data, C_{net_train} from equation [5.1]. By using this equation for calculating the training data and using the same inputs as the FLC, it allows the two systems to model the inverse dynamics of the engine together. By allowing

the inverse of the engine to be modeled through a combination of the two controllers it reduces the possibility of the controllers working against each other and causing instability. The online training of the feed-forward ANN is approached much in the same way as the inverse model ANNs. Diagram [5.21], also shows the data store memory blocks for feed-forward ANN. Form the training files the new weights and bias values are written two these memory stores at the top level. By locating the memory stores at the top level it allows read blocks at any sublevel, i.e. the ANN, to read data from them. To see the more detailed Simulink program files please see appendix [C].

The description of the developed adaptive control architecture so far has demonstrated the control methodology and how the control system is organised and linked together. The developed control structure is capable of learning the mismatch between the MIMO fixed feedback controller and the plants dynamics. The error in the control can then be reduced through a continuously online adapting ANN system. This control structure therefore allows a mathematically simple approach to modelling and control of MIMO engine control problems through FLC and ANNs. The combined architecture allows for robust online adaption and calibration to model/plant mismatch and for control over systems that dynamics can change with time. The following chapters will demonstrate the process of the offline training and initial calibration of the ANN structures and FLC system for control over combustion in an IC engine.

6. Control of GT-Power Engine Model

For the engine model there was three possible software solutions available, Ricardo wave, Lotus Engine Simulation (LES) and GT-Power. GT-Power was chosen due to its combination of predictive SI turbulent flame model and simple method for linking with Simulink for control testing. LES and Ricardo wave can also be linked with Simulink but the author found the debugging more problematic and they lacked the predictive combustion model for SI engines that was available with GT-Power.

GT power is designed for steady state and transient engine simulations and can be used for the analyses of engine and powertrain control. It is applicable to petrol and diesel IC engines and provides the user with many components to model new concepts.

The mathematical solutions are based on a one-dimensional fluid dynamics model. This model represents the flow and heat transfer in the piping and other flow components of an engine system. In addition to the fluid dynamics model there are many other specialized models for systems analysis. GT-power uses a powerful high level GUI that allows a simple approach to managing object libraries and building, editing, executing and post processing models. GT-power limits the amount of input data entry so that only unique geometrical elements must be defined. Models are built using the GUI from libraries of pre-defined objects or user defined reusable objects.

GT-power can be linked to several third party software packages if additional detail or functionality is required. These software packages include CHEMKIN for more detail kinetic chemistry, STAR_CD, Fluent for fluid dynamics and Matlab-Simulink for control. The ability to link GT-power and Simulink is advantageous for developing control systems. By using Matlab-Simulink for control and GT-power for engine simulation, this allows the use of each program for their best capabilities, and allows the use of well recognised commercially available engine simulation software that can model engine behaviour adequately.

The GT-Power model used in this work is based on the example given with software for a four cylinder SI engine. Modifications to the model were made to incorporate the SI turbulent flame model and data link to Simulink. The modifications allow the model to be used as a virtual engine to allow the testing of the developed adaptive control. This chapter details the main components of the GT-Power model used and their initial conditions and parameters.

6.1 GT-POWER MODEL
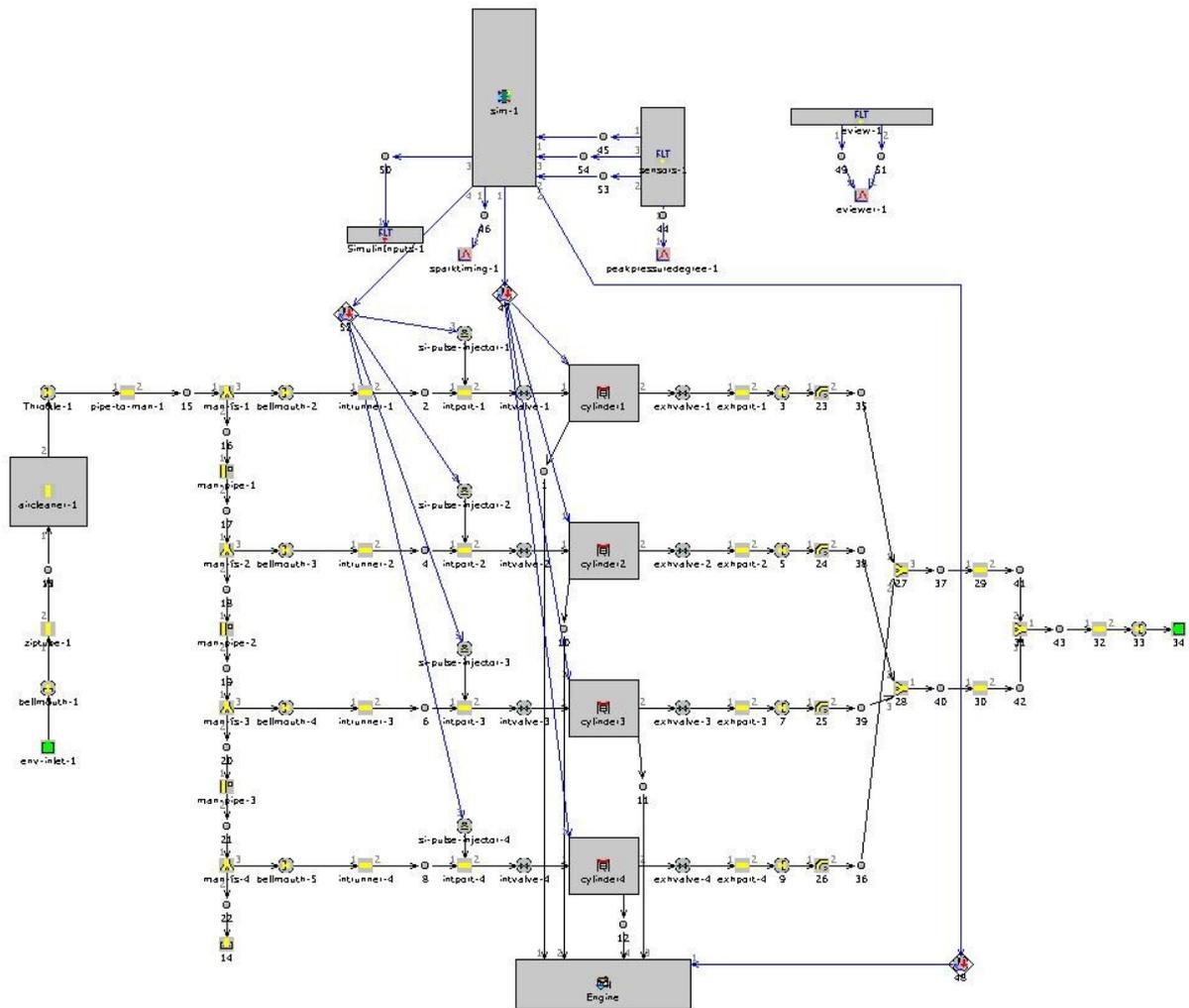
Figure 6.1 Overall GT-power model of a four cylinder port fuel injection SI engine

The GT-Power engine model is of a four cylinder SI engine. It has a full intake system and exhaust system, port fuel injection, valve dynamics, multi cylinder predictive combustion models and engine crack train dynamics. The model is created using the GT power GUI, where each of these systems are configured and modelled in individual objects. These objects are connected in a flow style system that represents a real engine, figure [6.1]. To explain how the GT-power model for this work is constructed it is useful to separate this system into several sub categories where similar objects are used for the construction.

6.1.1 INLET AND EXHAUST, ENVIRONMENT AND SYSTEM

The intake and exhaust system for the GT-Power model are very similar in construction; it is therefore useful to describe the construction of these subsystems together.

 If we first consider a real engine, at the extremes of the flow system we have the intake at the air filter and the exhaust pipe exit. At these locations the real engine is open to atmospheric conditions. In creating a GT-Power model of an engine we also need to define the environment as well as the system. The environment allows the model to have information regarding the environment outside of the engine. The environment like many of the components that need modelling for engine dynamics can be defined using a pre-determined object. The object used for describing the environment is called the "End Environment". The end environment object allows us to specify condition of the inducted gas such as the ambient pressure, temperature and composition. For the four cylinder model the following settings are used, table [6.1].

Table 6.1 Attributes for GT-power End Environment

Attribute	Object / Value
Pressure	1 bar
Temperature	300 Kelvin
Pressure Flag	Standard total
Composition	"air"

The "standard total" setting for pressure flag allows for pulsations to be considered in both directions across the intake boundary caused from intake flow dynamics. The composition of the gas is set to a standard model for "air". Air is defined as a mixture consisting of a mass fraction for nitrogen of 0.767 and a mass fraction for oxygen of 0.233. Attributes for altitude and humidity are set as default for this model. The end environment is used as both ends of the model where the system is normally open to the atmosphere.
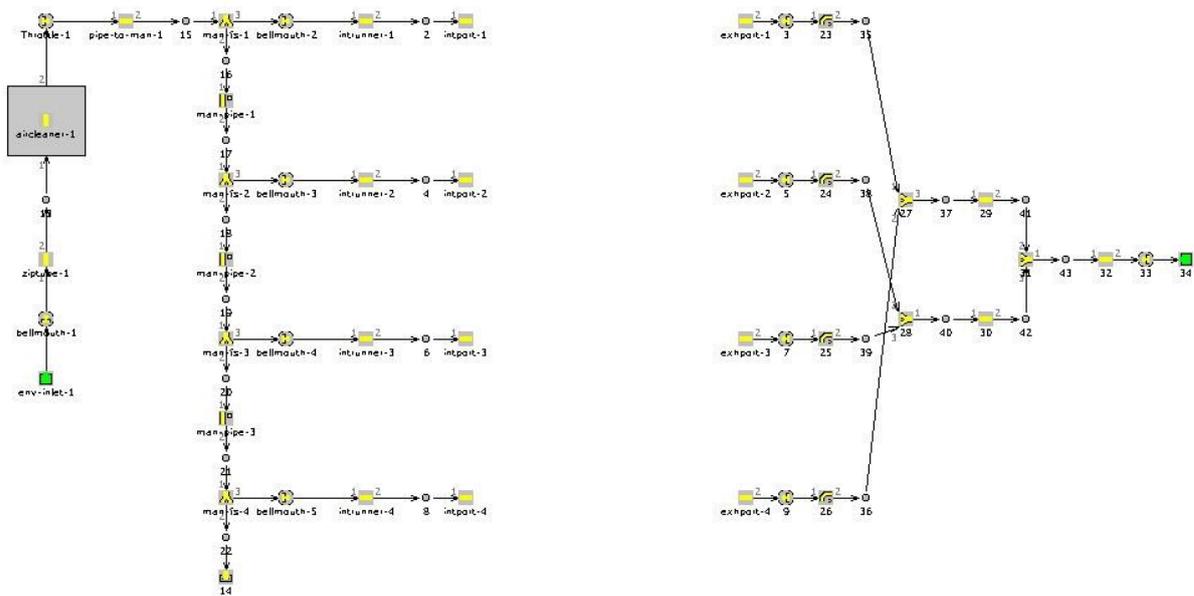
Figure 6.2 GT-power model of intake and exhaust system for a four cylinder port fuel injection SI engine

The remaining intake and exhaust systems is defined by a combination of different pre-determined objects that are linked together to create the system of pipes, figure [6.2]. These pre-determined objects include the following, "Orifice Conn", "Pipe Round", "F split General" and "Pipe Rectangle".

The orifice connector "Orifice Conn" is used to connect any two flow components together. The pipe round object can be used to define any round pipe used within the intake and exhaust system. By manipulating the attributes of the object it can also be used for modelling objects such as the air filter housing, intake runners and intake and exhaust ports on the engine. The rectangle pipe object has all the same attributes as the round pipe but with different dimensional properties. Between these two pipe objects most pipe flow objects can be represented. The "F split General" object allows the user to define a flow split volume connected to one of more flow components. It can be used to define any shape of flow split. This object is therefore very useful for creating intake and exhaust manifold systems. All of the flow components share very similar definable attributes that fall under one of the following categories;

- Dimensions and volume of object.
- Coefficients for discharge, pressure loss or friction.
- Material properties.

- Heat transfer effects and sub models.

6.1.2 CONFIGURING CYLINDER AND COMBUSTION

Fuel injectors can be defined by single objects, but the engine is a more complex system and requires a combination of systems including, "valve Cam Conn", " Eng Cylinder", Engine Crank Train" and a number of subsystems for each object.

To model the SI port fuel injectors either the "IngAFSeqConn" object or the "IngPulseConn" object can be used. The "Ing Pulse Conn" object is a sequential pulse fuel injector. This injector model allows the user to define the delivery rate of fuel and the injector pulse width. The "IngAFSeqConn" object on the other hand, allows the user to specify the AFR required, and the injector will add the appropriate amount of fuel to the system to maintain the AFR. The first type of injector is useful if the control problem is to constrain AFR, whereas the second fuel injector object is more desirable if controlling AFR is not of interest. Independent of type of injector object used the implementation is the same; the injector block is connected to the intake port for each cylinder.

The intake and exhaust valves can be defined using the "ValveCamConn" object. For this work it was convenient to copy the object from a template file. This allows for a basic valve profile to be used and their attributes to be applied to the model. Using this template profile is useful as the model is not of a specific engine, but we still require realistic valve behaviour. By defining different attribute values for each valve allows for the different intake and exhaust valve dynamics to be modelled. Table [6.2], shows the attribute values taken from the basic valve template file.

Table 6.2 Attributes for GT-power ValveCamConn

Attribute	Intake	Exhaust
Valve diameter	45.5mm	37.5mm
Valve Lash	0.1mm	0.1mm
Cam Angle	420°	290°
Cam timing Anchor reference	TDC firing	TDC firing
Cam timing lift Array reference	Theta=0°	Theta =0°

The basic attributes have defined the intake and exhaust valve diameters as well as their basic timing. From the valve template the valves are anchored to the firing TDC, the peak valve lift timing is defined by the number of degrees set by the "cam angle" attribute. Using the advanced settings allows the valve profiles to be defined.
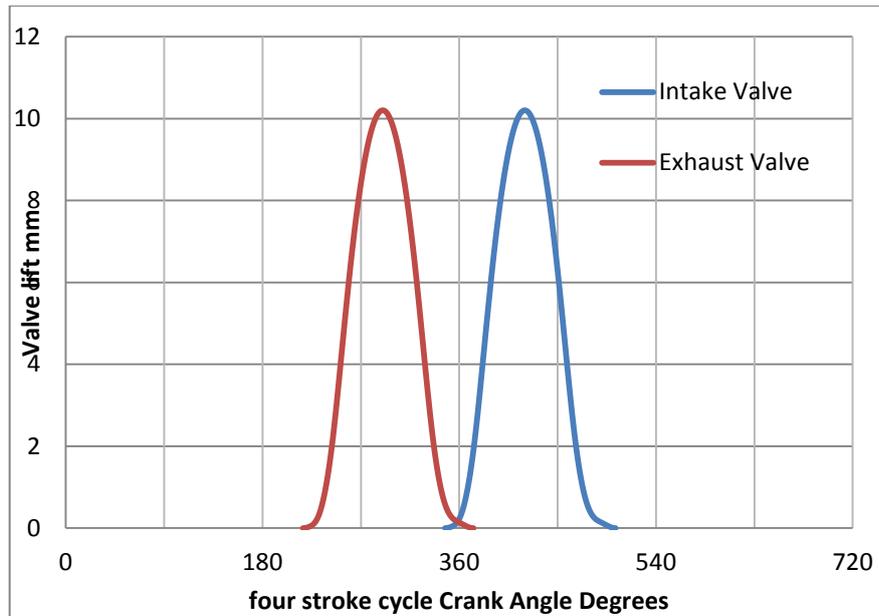
Figure 6.3 GT-power template for intake and exhuast profiles for SI engine

Both the intake and exhaust valve share a common valve profile taken from the valve template, figure [6.3]. The valve profiles from the template are not optimal, but it gives a basic profile at appropriate timing that allows the engine model to operate.
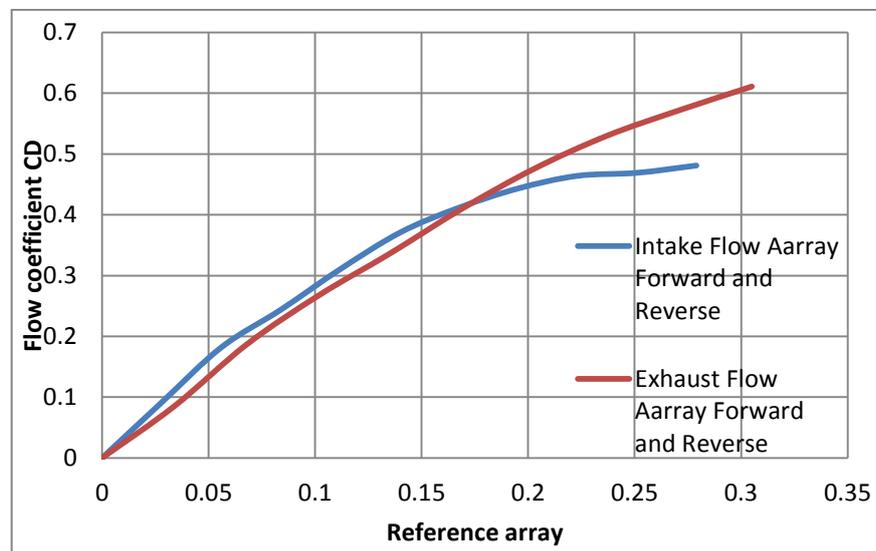
Figure 6.4 GT-power template for intake and exhaust valve flow coefficients

The "ValveCamConn" object also stores the attribute for the forward and reverse flow array, figure [6.4]. The flow array holds information of flow coefficients that contain all the flow losses present in the intake and exhaust ports.

The cylinder object is a unique GT power object as a large number of the attributes are defined by reference objects with defined sub-models, such as geometry, wall temperature, heat transfer, in-cylinder flow and combustion. The cylinder object brings these sub-models together to create the overall model of the dynamics of the cylinder. The cylinder object has the following Attributes, table [6.3].

Table 6.3 Attributes for GT-power cylinder object

Attribute	Object /Value
Initial State Object	"initial"
Wall Temperature Object	"twall"
Heat Transfer Object	"htr"
Flow Object	"flow"
Cylinder Combustion mode	Independent
Measured Cylinder Pressure Analysis object	Ign
Cylinder Pressure Analysis Mode	Off

The cylinder initial state determines the pressure of the cylinder, the temperature of the charge and the composition. Throughout this engine model the initial conditions are set to a single initial state object called "initial". The Wall temperature object sets the initial conditions for the solver for the head, piston and cylinder temperatures. For this model the object is "twall". The heat transfer object is used to determine how the heat transfer is calculated away from the engine cylinder and crank case objects. This heat transfer object has a number of attributes, table [6.4].

Table 6.4 Attributes for GT-power heat transfer object

Attribute	Object / Value
Heat Transfer Model	WoschniGT
User Model Object Name	Ign
Convection Multiplier	1
Head/Bore Area Ratio	1.15
Piston/Bore Area Ratio	1
Radiation Multiplier	Ign
Normalized-Hg profile	Ign
Convection Temperature Evaluation	Linear

There are a number of different heat transfer models pre-programmed to choose from. For this engine model the "Woschni GT model" is used. The in cylinder heat transfer will be calculated by a formula that closely emulates the classical Woschni correlation without swirl (2). The most important difference lies in the treatment of heat transfer coefficients during the period when the valves are open, where the heat transfer is increased by inflow velocities through the intake valves and also by backflow through the exhaust valves. This option is

recommended by GT-Power when measured swirl data is not available. The in-cylinder flow model calculates the swirl, tumble and turbulence within the cylinder. This information is then used by other models such as heat transfer and combustion if the appropriate models are used for these cylinder attributes.

"EngCylCombSITurb" is the cylinder combustion object used. This model is used to predict in-cylinder burn rate for spark-ignited engines. The model is well-suited for homogeneous fuel-to-air mixtures, but the Entraining Mixture Phi attribute can be used to impose non-homogeneous fuel-to-air mixtures.

There are a number of attributes that can alter the combustion prediction, these are as follows:

- Flame Geometry Object.
- Combustion Anchoring Option.
- Spark Timing or Anchor Angle.
- Initial Spark Size.
- Entraining Mixture Phi.

The SI turbulent flame model

The SI turbulent flame model in GT-power is based on work published in literature (139, 140, 141). The prediction accounts for cylinder geometry spark timing, air motion and fuel properties. The method is based on three main equations that govern the burn rate and the mass entrainment rate of the unburned charge.

$$\frac{dM_e}{dt} = \rho_u A_e (S_T + S_L) \qquad\qquad (6.1)$$

$$\frac{dM_b}{dt} = \frac{(M_e - M_b)}{\tau} \qquad\qquad (6.2)$$

$$\tau = \frac{\lambda}{S_L} \qquad\qquad (6.3)$$

$M_e = Entrained\ mass\ of\ the\ unburned\ mixture$

$t = Time$

$\rho_u = Unburned\ density$

$A_e = Entrainment\ surface\ area\ at\ the\ edge\ of\ the\ flame\ front$

$S_T = Turbulent\ flame\ speed$

$S_L = Laminar\ flame\ speed$

$\tau = Time\ constant$

$\lambda = Taylor\ microscale\ length$

The unburned air fuel mixture enters into the flame front through the flame area at a rate proportional to the sum of the turbulent and laminar flame speeds. The burn rate is proportional to the amount of unburned mixture behind the flame front,$(M_e - M_b)$, divided by a time constant, τ. The time constant is calculated by dividing the Taylor micro scale, λ, by the laminar flame speed. For this model to operate an in-cylinder flow object must also be used. This allows the turbulence intensity and length scale to be calculated and provided for the combustion model.

The engine crank train object

The engine crank train object completes the engine model; it specifies the kinematics and rigid dynamics of the common reciprocating IC engine crack train. The rigid dynamics model translates the phased pressure forces acting on each piston to torques at the crankpin. These are intern added to produce an engine total torque output. The crank train object has a wide range of attributes to describe such things as defining the type of engine, cylinder geometry, firing order, inertia, bearing loads and vibration.

A number of the advanced settings remain inactive for this engine model; the following attributes have been activated and have the following settings, table [6.5].

Table 6.5 Attributes for GT-power engine crank train object

Attribute	Object / Value
Engine Type	4 – stroke
Speed or Load specification	Speed(RPM)
Engine speed	[RPM]
Engine friction object	"friction"
Start of Cycle (CA at IVC)	-180
Cylinder Geometry Object	"geom"
Firing intervals	Cyl1: 0, Cyl3: 180, Cyl4: 180, Cyl2: 180
Reference state for Volumetric Efficiency	"initial"
TDC Angle Convention	Piston Position

By setting the speed or load specification to speed the simulation will operate at the desired RPM while the corresponding load variation will be calculated. A friction object is required to define the friction characteristics of the engine. The start of the engine cycle is also defined at a crank angle degree before TDC. The cylinder geometry object defines the cylinder geometry, while the cylinder numbers are defined by the port numbers where each cylinder attaches to the crank train object. The firing intervals are then specified for each cylinder. The

initial state object is used again to represent ambient conditions; this enables the solver to calculate a reference density for the calculation of volumetric efficiency.

The combination of objects discussed form the basic four stroke SI model with predictive turbulent combustion model seen in figure [6.1].

6.1.3 GT-POWER LINK TO MATLAB-SIMULINK

GT-power has been designed to be linked with a number of third party software programs and has pre-defined harnesses to allow information to pass between GT-power and the third party software.

The Simulink harness block allows information from sensors to be relayed to Simulink while receiving any number of control signals per sample period. To relay the control signals between the Simulink harness block and engine component objects an actuator connector needs to be connected in series. This actuator connector object allows an attribute to be selected and updated for each time step of the simulation.

When using the GT power model with Simulink for evaluation of a control system, consideration is required for the GT-power run setup. When running simulation using only GT-power it is often useful to allow automatic shut off when steady state conditions are reached. Once steady-state conditions are reached it is no longer useful to continue the simulation as there will be no change. When performing simulation of multiple case scenarios this allows time to be saved as the solver can move to the next case once steady-state condition are reached. However, when the simulation is linked to Simulink for evaluating control we may require the simulation to control to steady-state conditions. It is therefore important to change this setting to deactivate the automatic shut off. GT-power simulation can operate in two time base modes; time clock or a cycle clock. For use with control, a cycle clock is prefered. The cycle clock will allow signal feedback after each cycle. If the simulation was based on a time clock, data feedback could be passed back at a set frequency rather than linked to engine speed. The overall simulation length is also best controlled through Simulink. To connect GT-power to Simulink there also needs to be a corresponding GT-power connection block in the Simulink control environment. The Simulink GT-power block allows the desired GT-power model to be initiated as well as settings for;

- Time step
- Case no
- Define number of inputs and outputs from the Simulink environment.

6.1.4 PERFORMANCE OF GT POWER MODEL BEFORE AND AFTER DYNAMIC SHIFT

The GT-Power model can be used to generate simulation data very quickly. This is an advantage of using engine simulation software in the development stages for control systems. In the development of the control we require an architecture that can adapt to time varying dynamics. Through the use of the GT-power engine model, it is possible to vary a large number of engine parameters and characteristics to force dynamics to change with time. The CR of the engine is a parameter that can have dramatic effect on combustion efficiency. By altering the CR the pressure and temperatures reached after the compression stroke will change. Combustion taking place under lower pressure and temperature will have a significant effect on combustion duration.
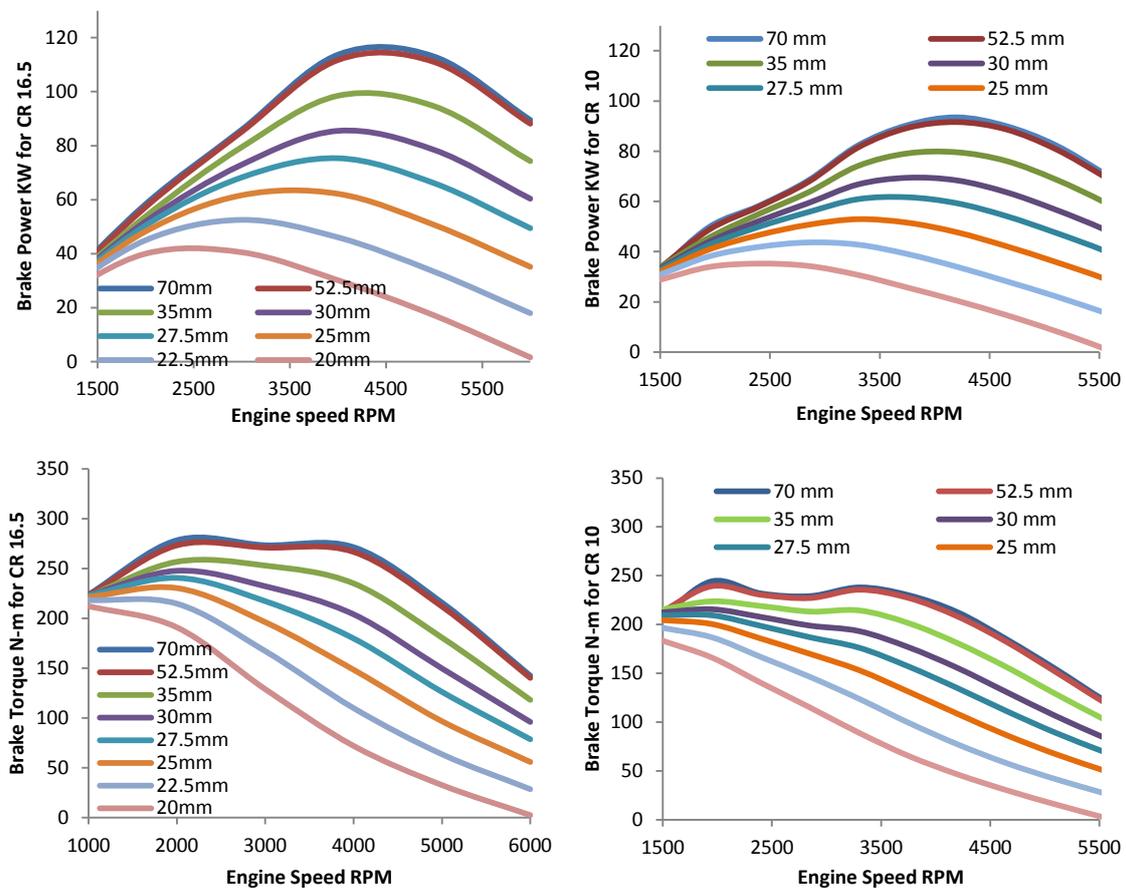
Figure 6.5 Comparison of GT-power port fuel injection SI engine performance at compression ratio 16.5 and 10, all throttle values given as an orifice diameter in mm.

Figure [6.5], shows the GT-power model performance for power and torque for various throttle orifice diameters at a geometric CR of 16.5:1 and 10:1. When the CR is reduced the combustion efficiency is decreased. When the combustion takes place at the lower temperature and pressure the fuel/air mixture is at a lower density and lower energy, this

auses flame propagation to be slower through the mixture and a decrease in the energy conversion efficiency. The effect of the slower combustion rate has a significant effect on power and torque output for the GT-power model, figure[6.5]. The change in power and torque from the change in CR indicate that the predictive combustion model used in the GT-power model is adequate for predicting realistic relationships and trends of SI combustion to changes is parameters that can affect the dynamics of combustion.

Spark timing maps
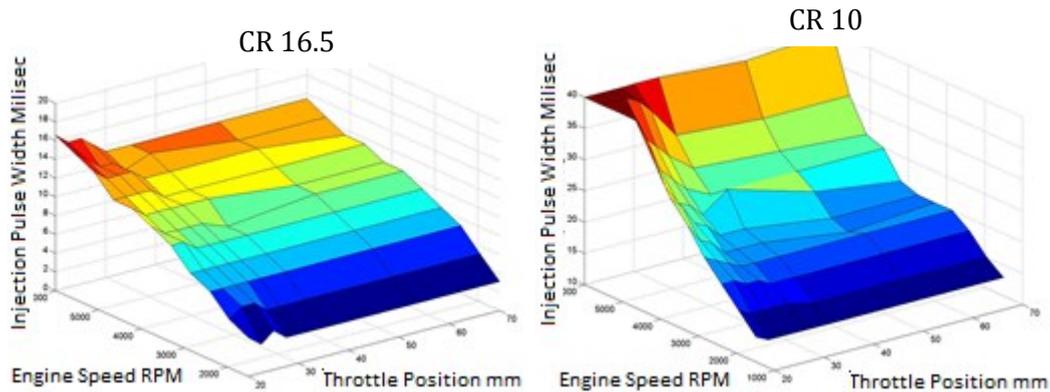
Figure 6.6 Comparison of GT-power port fuel injection SI engine SI timing control at compression ratio 16.5 and 10

Fuel injection maps
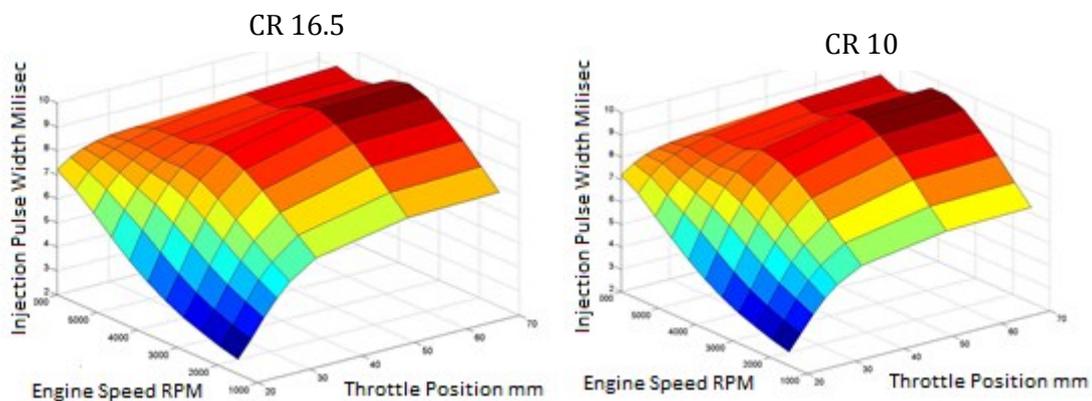
Figure 6.7 Comparison of GT-power port fuel injection SI engine fuel injection control at compression ratio 16.5 and 10

As the dynamics of the combustion change with CR it is also important to see if the control to achieve maximum combustion efficiency changes with the two CR settings. Figure [6.6] and [6.7] show the approximate spark timing and fuel injection control tables respectively for the CRs 16.5:1 and 10:1. These control maps where created through the design of experiment

(DOE) method that is available with GT-Power. By specifying the range of the control variables over the range of engine operating conditions, the DOE generated a reduced test plan to encompass the range of possible. Constraints can also be applied to the DOE where control inputs ranges would be limited by engine operating conditions. For example, low speed and throttle would require restraints on fuel quantity. Large amounts of fuel would cause over rich mixtures that would cause misfire. From the results of the DOE it was possible to interpolate the control maps above based on finding injection pulse width for stoichiometric AFR and spark timing for peak Indicated Mean Effective Pressure (IMEP).

 At the two CRs there will be some small change to volume at TDC, however this change will be very small in comparison to the overall charge of the cylinder. For this reason there are only small changes to fuel amount for the change in CR if AFR is to remain constant. However, the change in CR is known to have a significant effect on combustion duration. Therefore this will require a significant change in combustion timing to maintain peak pressure position at a location to yield maximum conversion efficiency. The change in the ignition map is clear from figure [6.6]. The spark advance has almost doubled for all speed and load locations on the spark timing map between CR 16.5 to 10. This indicates as the predicted combustion model dynamics change, the combustion model requires new control to maintain the new combustion dynamics at optimal conditions. Therefore the GT-power engine model can be used to develop the control for combustion that dynamics can change with time.

6.2 CONFIGURING THE CONTROL ARCHITECTURE

When applying the developed control architecture to a system there is a requirement for configuration depending on the control variables and the dynamics of the system. The FLC require defined inputs and outputs with appropriate MFs and a rule base to draw control decisions from. The ANNs also require a degree of configuration. Before an ANN can be transferred into the Simulink environment it needs to be initiated in Matlab. In this process the number of layers, neurons and activation functions are determined. There is also some requirement for basic offline training so when the control is first initiated it has a vague knowledge of the system to be controlled.

6.2.1 CALIBRATION OF THE FUZZY LOGIC CONTROL SYSTEM

To evaluate the proposed adaptive control architecture a MIMO controller can be created for the GT-power model of a SI engine. The control and control adaption will be evaluated for first a MISO control for combustion timing control, followed by a MIMO control for combustion timing and AFR.

The inputs required for combustion timing and AFR control are:

- Engine speed.
- Throttle position.
- Combustion timing error.
- Rate of change of combustion timing error.
- AFR set point error.
- Rate of change of AFR error.

The outputs required for combustion timing and AFR control are:

- Feed-forward spark timing component.
- Feed-forward fuel injection pulse width.
- Feedback spark timing control trim component.
- Feedback injection pulse width trim component.

Combustion timing and AFR are largely dependent on engine load and engine speed, the inputs for engine speed and throttle position allow the control system information regarding the current engine state.   From the engine state appropriate spark ignition timing and injection pulse width can be predicted if an adequate mapping of the engine dynamics is achieved through appropriate rules and MFs.   If there are large disturbances and/or a mismatch between the dynamic model and the engine dynamics the feedback control can be used to minimize the error. Feedback error and the rate of change of the error signal allows the control to trim the feed-forward control component based on the magnitude and direction of the error and how fast that error is changing.

6.2.2 DESIGN OF THE UNIVERSE OF DISCOURSE

In a conventional feed forward ECU control systems there are maps built up of many values, calibration of these maps is a time consuming task. Using FLC offers an alternative method to map the dynamics of an engine. FLC mapping is also not limited to two dimensions as the relationship between variables is determined through linguistic rules.
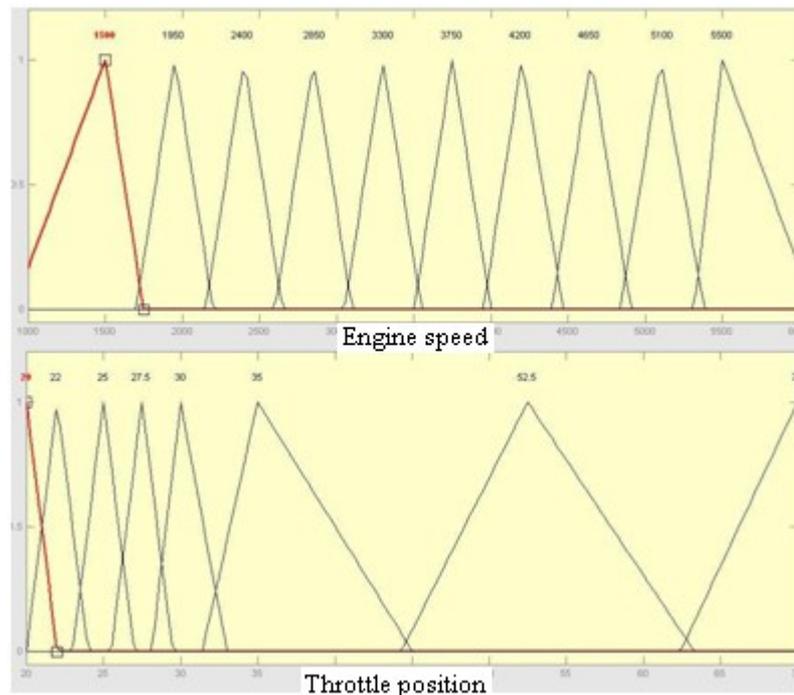
Figure 6.8 Universe of discourse for throttle and engine speed

Figure [6.8], shows the universe of discourse for inputs engine speed and throttle position. By positioning MF into the universe of discourse it allows the input space to be separated into areas. The number of areas depends on the resolution and accuracy required from the mapping of the input space. For the input throttle position, a large proportion of the nonlinear dynamics associated with airflow past the throttle plate is during the first 50 percent of opening. Between 50 percent and wide-open throttle the mass air flow change is far more linear. As a larger proportion of the fluid dynamics are during the smaller throttle opening angles it is useful to increase the proportion of MFs to increase the resolution of the map. When there are smaller non-linear changes in the dynamics less MFs are required reducing number of rules and memory requirements.
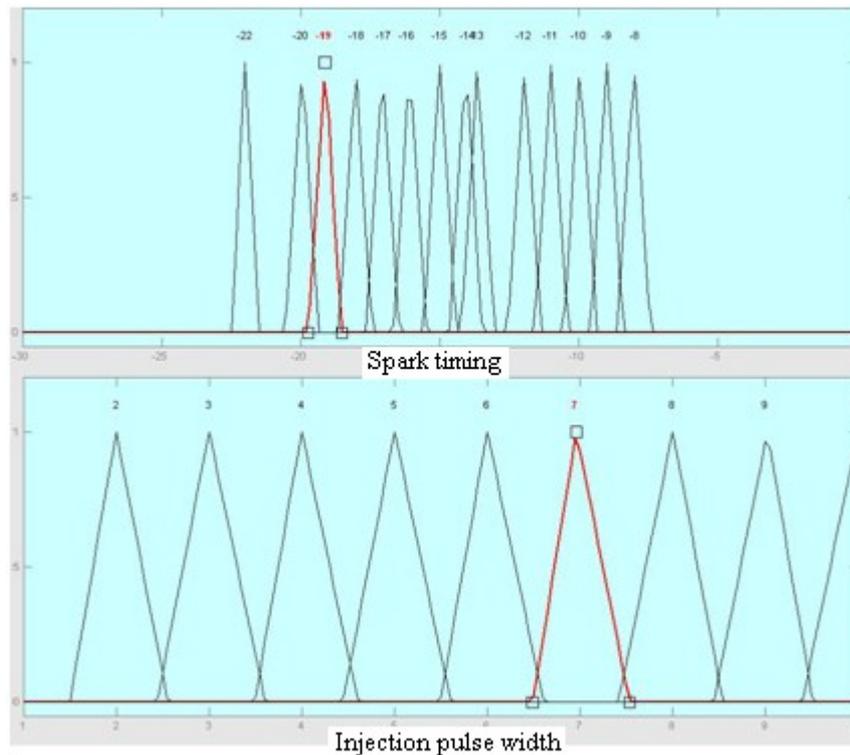
Figure 6.9 Universe of discourse for outputs spark timing and fuel injection pulse width

The nonlinear dynamics of the engine can be modelled through the shape of the MFs, how they overlap and how they are combined through the rule base. Figure [6.9], shows the universe of discourse for the outputs spark timing and injection pulse width. By mapping the input MF to these output MFs it is possible to build a nonlinear model of the engine behaviour. By allowing the input MFs to overlap it has allowed for interpretation for any value within the input universe of discourse to be mapped to one or more output MFs. The level of activation through the rule-base determines the level of membership to each output.

Table 6.6 Sample of rules from fuzzy rule table for spark timing and fuel pulse width control outputs

1	If (RPM is 1500) and (throttle is 20) then (Spark timing is -10)
2	If (RPM is 1500) and (throttle is 22) then (Spark timing is -9)
3	If (RPM is 1500) and (throttle is 25) then (Spark timing is -9)
4	If (RPM is 1500) and (throttle is 20) then (fuel pulse is 7)
5	If (RPM is 1500) and (throttle is 22) then (fuel pulse is 8)
6	If (RPM is 1500) and (throttle is 25) then (fuel pulse is 8)

Table [6.6], represents a sample of the rule base between the inputs of engine speed and throttle position and outputs spark timing and injection pulse width. The rules are based

upon simple if then statements and a number of rules can be active at any one time depending on input activation. The level of activation of a rule and therefore output, is determined by the level of activation of the input MF. The shape, location and overlap of the input MF in relation to the output membership are critical for the calibration of the fuzzy dynamic model.
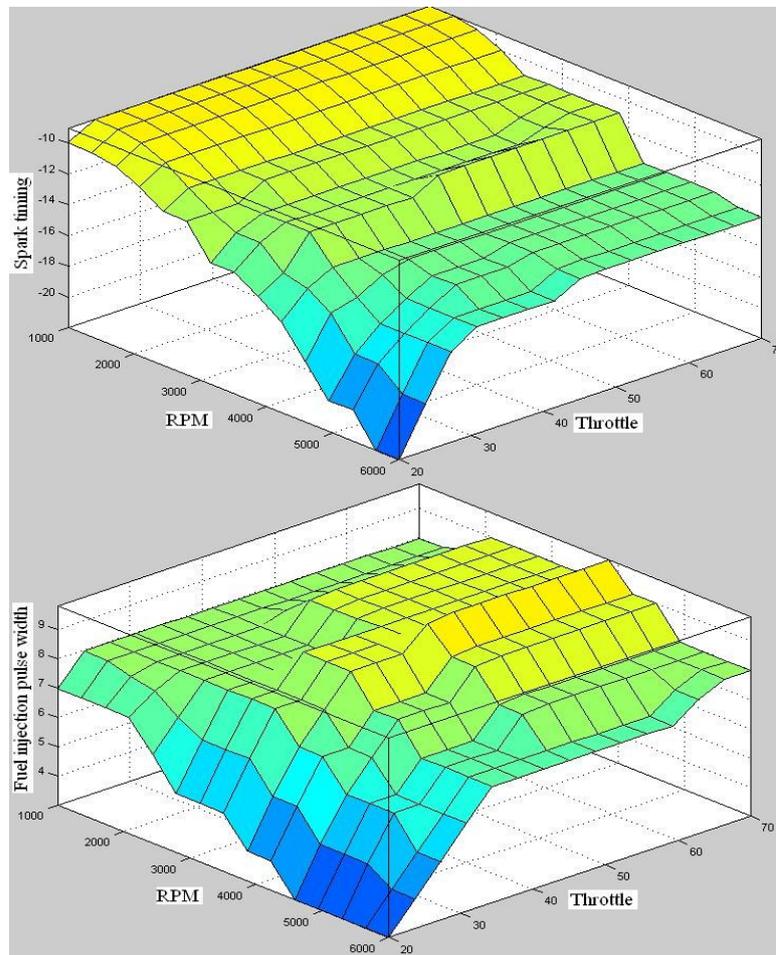
Figure 6.10 Fuzzy logic nonlinear control model for spark timing and fuel injection pulse width

Figure [6.10] shows the resulting spark timing and fuel injection control maps from the combination of rules and input/output MFs. It can be noted at this stage that to create a self-calibrating fuzzy system the position, shape and overlap of the MFs could be automated through the use of ANNs to give desired outputs. From comparison between figure [6.10] and [6.6],[6.7], the basic fuzzy logic control maps represent similar shapes to that of the maps found for optimal performance at CR16.5 for the GT-power model. The initial fuzzy logic model has not been designed for optimal performance but to more simply locate peak pressure position at 10 degrees ATDC and AFR at 14.5. The rough fuzzy map has sharp edges

due to the extensive use of triangular membership functions and limited overlap. If these required smoothing, the Gaussian or bell MF would be more applicable. The shape of the map has been reproduced using smaller quantity of reference operating points due to the nature of the FLC mapping.

6.2.3 FUZZY FEEDBACK SYSTEM

The feedback control is based on the error and rate or change of the error and also requires configuration to the dynamics of the engine. If a proportional and derivative controller were being designed for this task the gains for the two components would be optimized to give the desired response of the system. With the FLC structure the gain can be considered as the ratio between the range of the input and output universe of discourse. As MFs are used to separate the input and output space, the response of the system does not need to be linear. Through MF shape, location and size, any nonlinear function can be modeled allowing non-linear response.
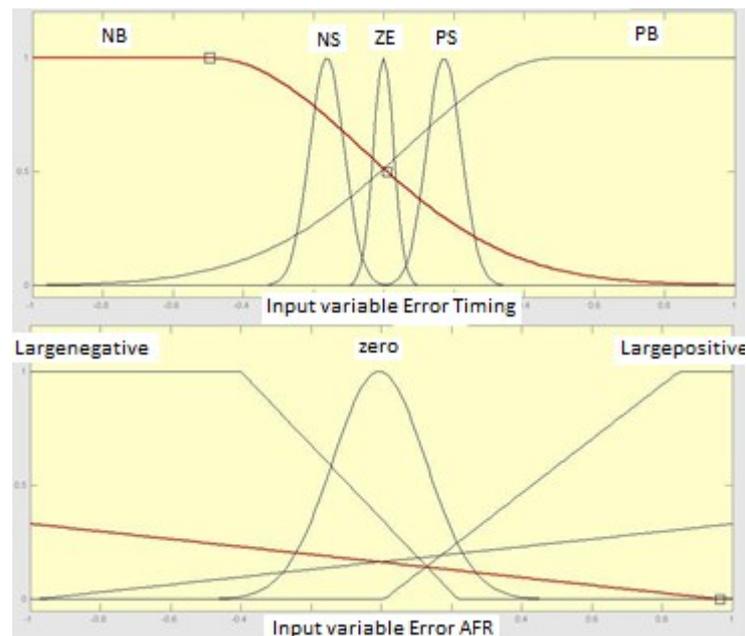
Figure 6.11 Fuzzy logic input universe of discourse for timing and AFR error

Figure [6.11], represents the input universe of discourse for combustion timing and AFR error. The first stage of their design was to create a linear response. This can be easily achieved through having two diagonal MFs across the universe of discourse. The angle of the diagonal MF and the range of the universe of discourse dictate the level of linear response. Once a basic linear response can be achieved, nonlinearities in the response can be

incorporated by altering the shape of the MF to nonlinear functions, or addition of new MF to adjust the response over finite areas of the input range. Through the nonlinearities it is possible to achieve different response rates depending on the level of error. For example a large error will require a large response; whereas small errors around a sensitive operating point may require a finer adjustment.
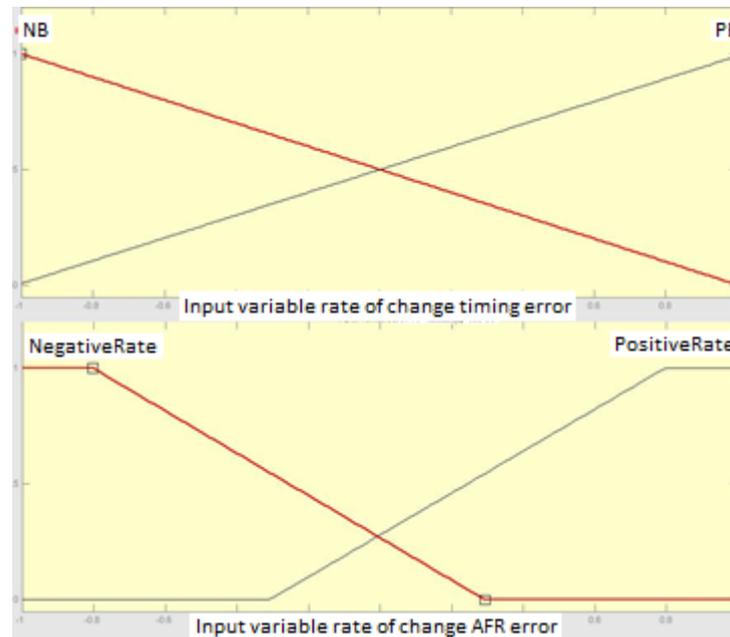
Figure 6.12 Fuzzy logic input universe of discourse for rate of change of timing and AFR error

The fuzzy control response for the error terms was tuned through adjusting the shape and location of the MFs as well as the range of the input and output space. The response to error was increased until overshoot was exhibited. This overshoot allows the control to have a fast response. The level of overshoot and any oscillation can be damped through the addition of response to rate of change of error. Figure [6.12] shows the input universe of discourse for rate of change of error for combustion timing and AFR. The same procedure was used for developing the design of the input space.
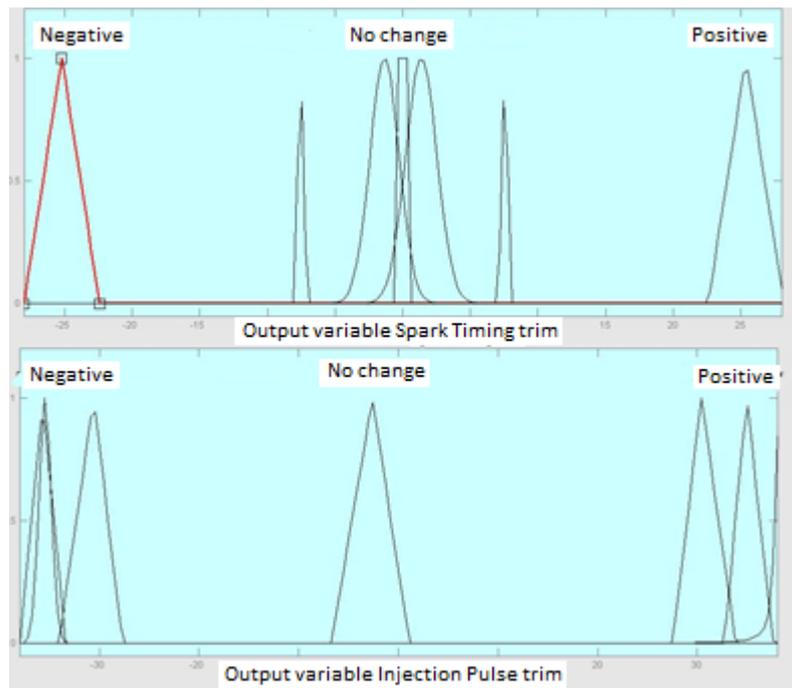
Figure 6.13 Fuzzy logic output universe of discourse for outputs spark timing trim and injection pulse width trim

Figure [6.13], shows the output universe of discourse for spark timing trim and injection pulse trim. By using MFs to only cover finite parts of the output space it allows for tuning of the control response. By widening or moving a particular MF will adjust the effect and weighting it will have on the associated rules and final defuzzified output.

Table 6.7 Sample of rules from fuzzy rule table for feedback fuzzy control

1	If (AFR error is Zero) then (Inj pulse trim is no change)
2	If (AFR error is Negative) then (Inj pulse trim is Positive)
3	If (AFR error is Positive) then (Inj pulse trim is Negative)
4	If (AFR rate timing error is negative) then (Inj pulse trim is rate damp positive)
5	If (AFR rate timing error is positive) then (Inj pulse trim is rate damp negative)

Table [6.7], demonstrates a sample of rules for the feedback fuzzy control. Again the format of if, then rules was used. By keeping the timing error rules separate to the error rules it allows fewer rules to be written overall. A combination of active SISO rules is equivalent to a MIMO rule. The interaction of the SISO is controlled through the fuzzy reasoning process.
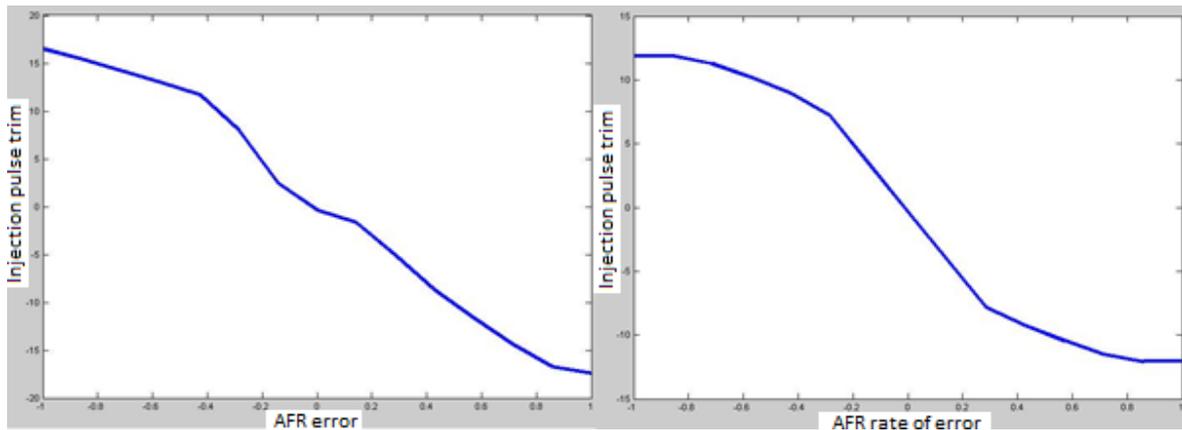
Figure 6.14 Fuzzy logic control mapping between AFR error , rate of AFR error and ouput injection pulse width trim

Figure [6.14], demonstrates the nonlinear mapping between the input error and rate of error with output fuel injection pulse trim. When the AFR error is small there are smaller response changes to reduce sudden changes in rate of change of error and allow for smaller sensitive changes to be made. At larger errors, the nonlinear function allows for a large response to be achieved. By allowing a larger response for a given error allows a greater influence over the control and will drive the error back to set point at a faster rate.
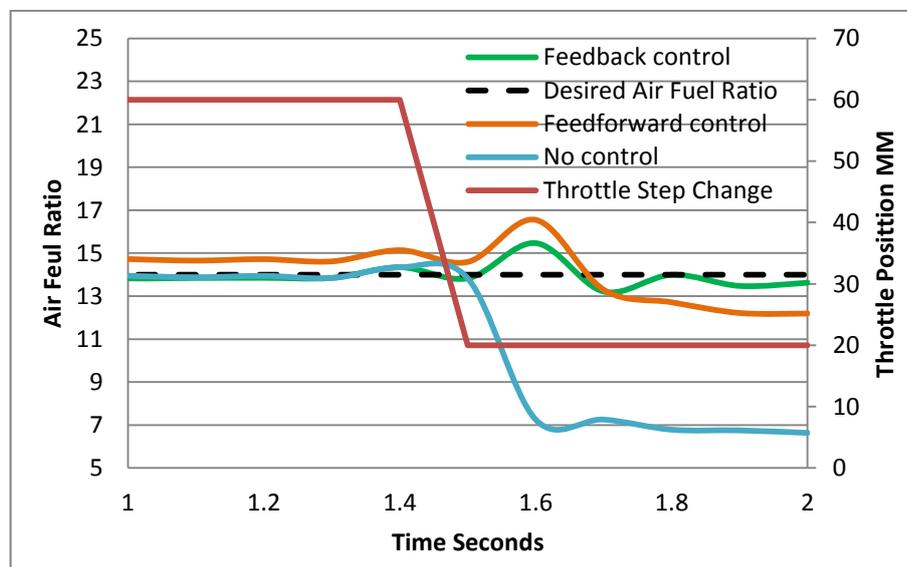
Figure 6.15 Comparison of no control, feed-forward control and combined feedback control for a sudden step change in throttle at constant engine speed.

Figure [6.15], demonstrates a comparison of three AFR responses for a step change in throttle at constant speed for no control, feed-forward only and combined feedback control. For the sharp step change in throttle orifice diameter form 60mm to 20mm, under no control the AFR becomes increasingly rich and lead to misfire and instability. Feed-forward control allowed the AFR to remain within acceptable limits but includes offset error and large oscillation from the throttle change. By combining the feed-forward control with the feedback components allows for the error to be reduced eliminating error offset while also reducing oscillation from the sharp throttle change.

In a standard ECU transient AFR control is achieved through a look-up table. The fuel map is a table of injector pulse widths covering all the operating points which may include intake manifold pressure, engine speed, mass airflow, throttle positions, intake and coolant temperatures. Generating the fuel maps can typically be an expensive process, as it takes many hours of calibration on expensive equipment to must be repeated for every engine model variation (73). Fuel maps can include some form of adaption to changes from vehicle to vehicle or changes over time, but are generally static in nature. Therefore, a feedback control system can offer an alternative method that could potentially save time and money during development.

6.3 NEURAL NETWORK INITIALIZATION AND OFFLINE TRAINING

When training ANNs there is a great deal of importance to the correct choice of inputs and the spread of the training data. The ANN is not a magic box, it cannot find relationships where there are none to be found. Likewise, if the ANNs are given too much information it is possible for incorrect relationships to be found.

The inputs to each ANN needed to be carefully considered. To be able to predict spark timing from a given peak pressure position there is some basic information that is needed for training.

- Spark timing.
- Peak pressure position.
- Engine speed.
- Mass air flow.

Using the described input data allows an ANN to predict spark timing as an output for information on peak pressure position, engine speed and mass airflow. If there is a change in any of the inputs it would create a change in the output value for predicted spark timing. For the GT–power engine model this is the only information required to predict spark timing.

For fuel injection pulse width the following data was used:

- Air fuel ratio.
- Fuel injection pulse width.
- Engine speed.
- Throttle position.

The fuel injection is directly related to the ratio of air to fuel and the mass of air currently being inducted into the engine. Engine speed and throttle position give the ANN information regarding relationships to airflow, these relationships are then linked to fuel requirement for a given air fuel ratio. Mass air flow was first used as an input instead of throttle and engine speed, but using engine speed and throttle position provided better ANN training results.

The spread of data was an important consideration. The initial training of an ANN should be on a wide spread of data. If the ANN is trained on a specific area of data its training knowledge will be limited and its predictions for other operating conditions will have large inaccuracies. However, being able to generate the training data can become problematic when considering problems with a large number of variables. If we consider the spark timing control map, it has three variables, spark timing, load site and speed site. This allows for a three dimension lookup table to be created. But with the inverse model we require additional information on spark timing for a range of peak pressure positions, not just at optimal engine operational locations. Therefore, the three dimensional table has become more complex in a fourth dimension. At each of the load and speed sites there would need to be a two dimension table for spark timings for different peak pressure positions. Collecting this data for the operating range of the GT-power model require an excessive memory demand and increase in computational effort. Lookup tables are limited to two dimensions before their size becomes computationally intensive. This causes a problem, if we cannot collect all the relevant training data how can the ANN be trained over the entire operating range of the GT-power model?

The ANNs have several key advantages for modelling systems. The complete training data is not required as the ANNs have an ability to generalise their knowledge between training sets. Therefore, as long as the training data can encompass the full operating range, the ANN can

generalise the relationship between the gaps. This allows for a reduced amount of training data to be collected and stored without problems in computer memory. Another advantage of the ANNs is their ability to continually train online. This is an advantage that the proposed control architecture exploits. The control architecture allows training online to improve the original training knowledge. By training online it allows new knowledge to be added and it allows the ANNs to alter the memory to dynamic relationships that can change with time.

The key to achieving accurate training is a balance between:

- Correct input output data choices.
- Accurate input output data.
- Training frequency.
- Number of epochs or passes of data in training runs.
- Size of ANN in relation to dynamics of problem.

Each of these points has to be carefully considered for each application of ANNs. If any of these are incorrect then ANN performance will suffer.

6.3.1 COLLECTING INPUT OUTPUT TRAINING DATA USING DESIGN OF EXPERIMENT

To determine the training data for the ANNs the statistical tool Design of Experiments (DOE) was used. The DOE tool allows a test program to be designed for the collection of training data for ANNs. The DOE tool provides a method for test procedure design that can generate training data that encompass the systems dynamics from a reduced number of test points.
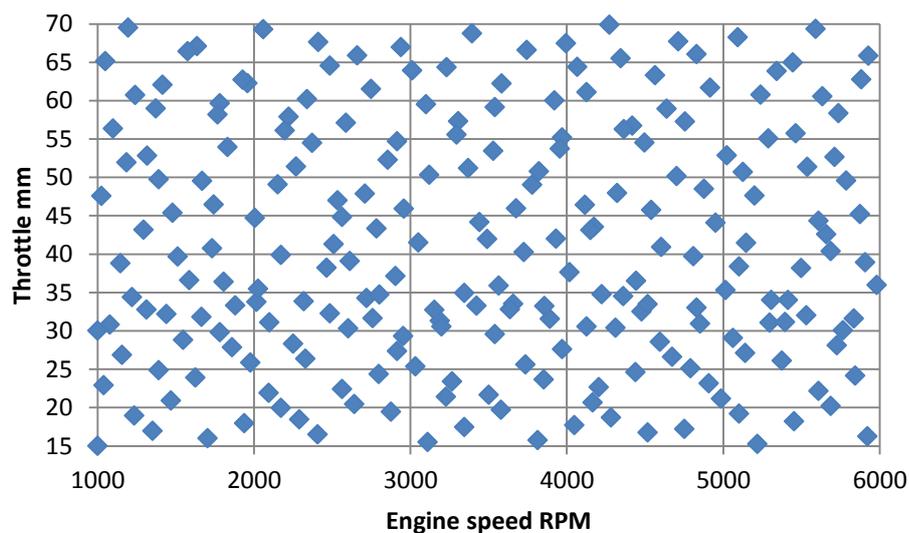
Figure 6.16 Design of experiment test points generated through space filling design for engine speed and throttle position

Figure [6.16], demonstrates the test points for the global variables engine speed and throttle. By defining the range of the global variables it is possible to use a space filling design to create a test plan. The space filling designs do not assume a particular model form. The aim is to spread the points as evenly as possible around the operating space. These designs literally fill out the *n*-dimensional space with points that are in some way regularly spaced, for this DOE 230 points where created using a Halton Sequence design, with a heavier weighting towards the 15mm to 35mm throttle positions. Using only 230 test points allows for the full operating window of the engine to be investigated while significantly reducing the number of operating points. A heavier weighting of points at the lower throttle positions allows for increase training data resolution over this region where larger nonlinear dynamics are exhibited in the airflow.
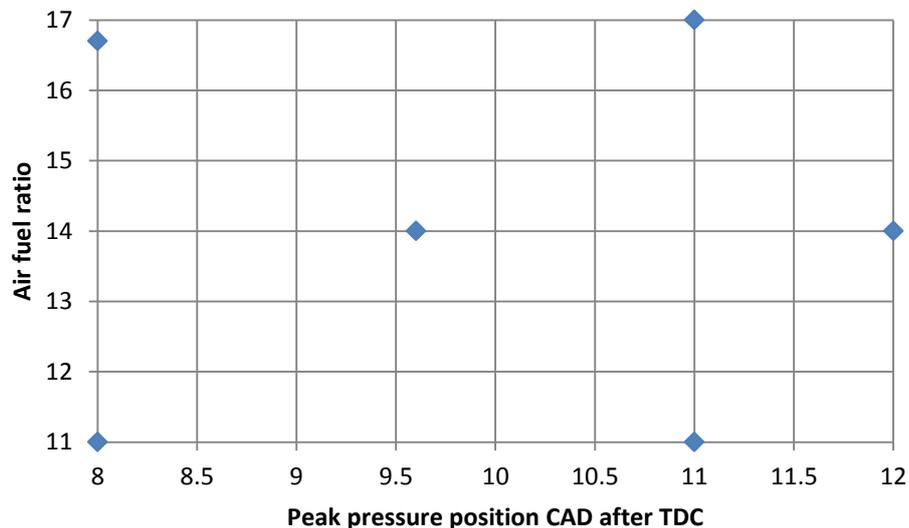
Figure 6.17 Design of experiment test points generated through space filling design for AFR and peak pressure position

Figure [6.17], demonstrates the test points for the local variables peak pressure position and air fuel ratio. These test points where designed using the same methods as above but for only 6 test sites. At each of the load and speed sites in the first space filling design there are numerous spark ignition timings and injector pulse width depending on the required combustion. To enable the ANNs the ability to predict what the ideal control variables would be for a given engine state, the ANNs also require some level of knowledge of what incorrect control variables are. If the ANNs were only ever trained on data for ideal operating

conditions their predictions would deteriorate if the engine operated outside of this ideal data. By allowing the ANNs to train from ideal operating conditions as well as incorrect operating conditions it allows a more complete inverse model of the engine to be created. By having data based on incorrect data allows the ANN to generalize the knowledge of the different controls at different operating conditions for different AFR and combustion timing to predict what the ideal should be to achieve the desired set point control. The training data was therefore collected for each of the AFR and peak pressure positions at each of the throttle and engine speed test points.

6.3.2 INVERSE NEURAL NETWORK MODEL CALIBRATION

Like the FLC the ANNs also need calibration. The number of layers, nodes and MF type are important to the performance of the ANN. However, to the authors knowledge there are currently no set procedures for matlab that guarantee the correct choice of these variables and the process is often a trial and error iterative method. For this reason a novel procedure was developed for calibrating the ANNs, the method is based upon the idea of growing the ANNs. The procedure begins with a simple neuron, this is configured with standard sigmoid activation function and trained, the performance is evaluated and if insufficient the ANN increases in neuron size. This procedure is followed until performance improvements decrease. This offers a method for finding the smallest ANN that can offer acceptable performance. Once a single layer ANN is investigated the same process can be extended to see the benefit of additional hidden layers. This approach was applied in a matlab script to offer a simple training tool for the ANN development. This approach will not guarantee optimal choice of ANN parameters but offers a logical approach to investigating ANN development. For the inverse model ANNs a structure of two hidden layers, with 20 neurons in the first layer and 10 neurons in the second layer was used. Increasing the number of neurons or layers past this point only offered small improvements on predictive accuracy. All the hidden layer neurons are of the hyperbolic tangent sigmoid activation function type. The tangent sigmoid function is a popular standard activation function for ANNs. The function is mathematically simple and exhibits a good trade-off between speed and training accuracy of the shape of the systems transfer function (142). The outputs nodes of the inverse models were of the saturated linear activation function type. This activation function is again mathematically simple, it also allows the outputs of the ANNs to be constrained between limits. This is an attractive feature for the inverse models once they are configured for online training. During online training, if an incorrect weight update did occurs the ANN could become unstable. Under that scenario if the output activation function was not saturated it could output any number to infinity. Using the saturated linear activation function offers a

level of safety. If the ANN becomes unstable its outputs can only be as large or small as the outer limits.

6.3.3 NETWORK TRAINING ALGORITHM

For offline training of the ANNs the Levenberg-Marquardt learning algorithm was used. The Levenberg-Marquardt algorithm (143), is the most widely used optimization algorithm. It outperforms simple gradient descent and other conjugate gradient methods for a wide variety of problems. The simple gradient descent suffers from convergence problems due to local minima or having too large a learning step and can miss global minima. Another problem is curvature of the error surface. For example, if there is a long and narrow valley in the error surface, the component of the gradient in the direction that points to the base of the valley is very small while the component along the valley walls is quite large. This results in motion more in the direction of the walls even though we have to move a long distance along the base and a small distance along the walls. Newton's method improved convergence through the use of curvature as well as gradient information. Newton's method has a rapid convergence, but the convergence is highly sensitive to the linearity of the start location. The Levenberg-Marquardt method is a blend of the gradient decent and Newton method to combine their advantages while reducing disadvantages. The curvature is calculated through the approximate Hessian matrix and is used to scale the gradient decent. Large steps are taken in directions of low curvature and small steps taken in directions with high curvature. The resulting training reduces convergence problems associated with error valley or nonlinearity of start location. It also facilitates variable step size for reducing problems with missing global minimum or getting stuck in a local minimum (143).

After the offline training and configuration procedure, figure [6.18] and [6.19] show the final inverse ANN models prediction of 100 test data points for spark timing control and injection pulse width. To evaluate an ANNs training performance often a separate data set to that of training data is presented to the ANN and the predicted outputs are compared to the desired results. In the comparison of the results in figure [6.18] and [6.19], a correlation coefficient calculation was used. The correlation coefficient helps determine the level of relationship between the desired results and the predicted results. The coefficient can be any number between -1 and 1. If the value is 1, then the prediction is perfectly correlated to the desired outputs. If the value is -1, then the prediction is perfectly inversely correlated, if one rises the other falls. When the correlation is equal to 0, then the prediction has no correlation or relation.

Equation [6.4] is used for calculating the correlation coefficient :

$$R = \frac{n\sum xy - (\sum x)(\sum y)}{\sqrt{n(\sum x^2)-(\sum x)^2}\sqrt{n(\sum y^2)-(\sum xy)^2}}$$

(6.4)

Where R is the correlation coefficient, x and y the data sets and n the number of data pairs. When correlation values are greater than 0.8 they are generally described as strong, whereas correlation values less than 0.5 are generally described as weak.
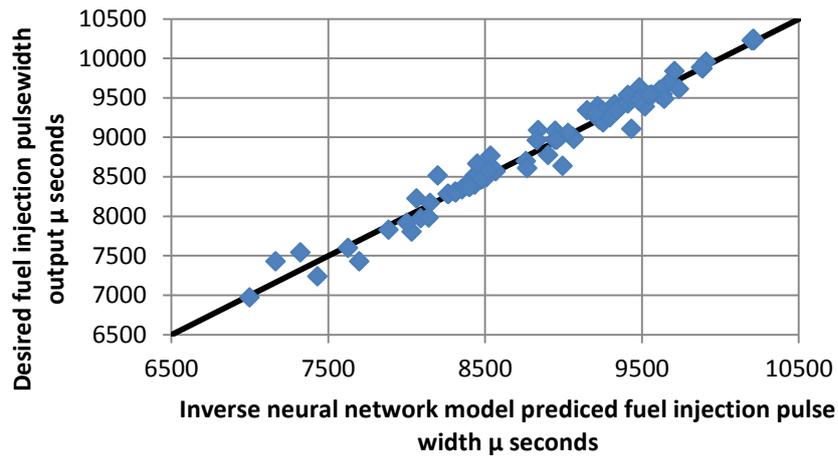
Figure 6.18 Neural network inverse model prediction for fuel injection control
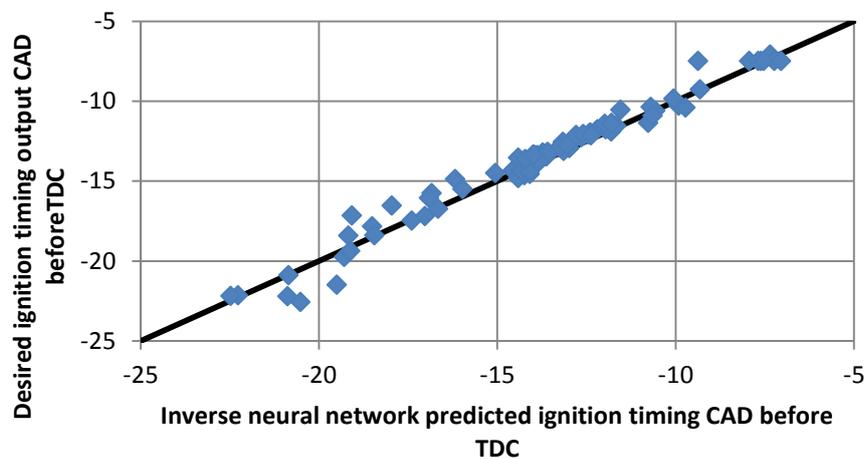
Figure 6.19 Neural network inverse model prediction for spark timing

Figures [6.18] and [6.19] show the ANNs predictions for fuel injection pulse width and ignition timing respectively. The ANNs have a correlation coefficient of 0.98 with the desired output test data for fuel injection pulse width and 0.97 with desired outputs for ignition

timing. These strong correlation coefficients show that the ANNs have successfully trained on the DOE data to represent the inverse dynamics of the GT-power engine.

6.3.4 FEED-FORWARD CONTROL NEURAL NETWORK

The feed forward control ANN also requires the same calibration and initial offline training procedure as the inverse models. However the feed forward control ANN relies on the inverse ANN models for training data. If the inverse ANNs where trained badly then the performance of the feed forward control ANN would suffer. This therefore emphasizes the importance of the preliminary offline training and ANN sizing. It may be possible to rely entirely on the online training procedure for ANN training, however it would be difficult to ensure the ANN structure and size chosen was capable of modelling the inverse dynamics of the engine. For the feed forward control ANN the following input and output data is of interest:

- Engine speed RPM.
- Mass air flow.
- Spark timing feed-forward error component.
- Fuel injection pulse width feed-forward error component.
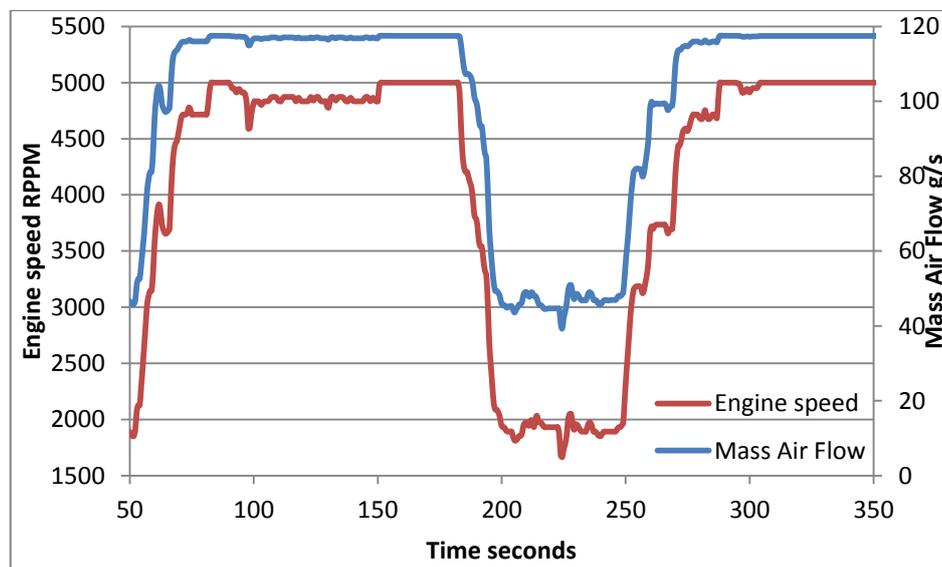
Figure 6.20 Short transient engine speed drive cycle

The range of data of interest for the feed-forward control ANN is around the desired operating set-points. To generate the initial offline training data a short transient drive cycle was simulated for changes in engine speed, figure [6.20].
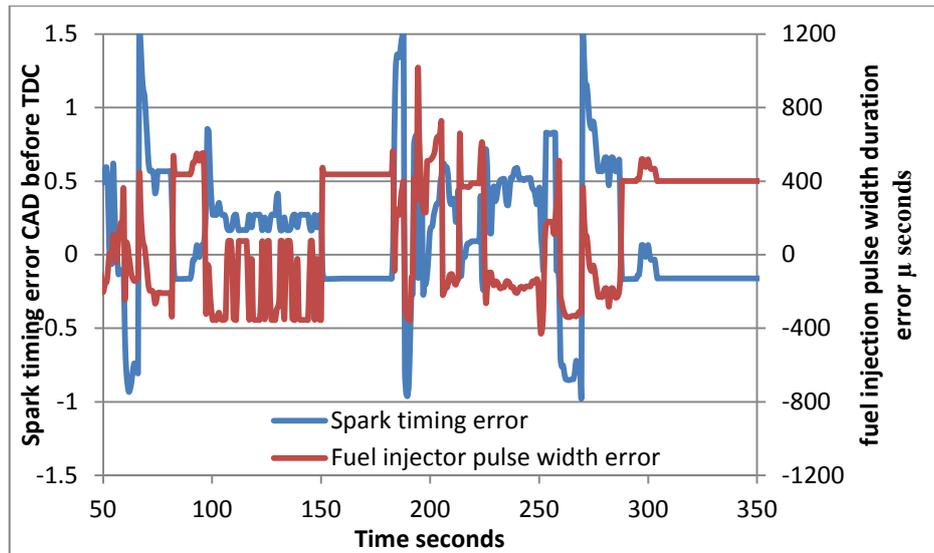
Figure 6.21 Control signal error for spark timing and fuel injection pulse width

The short transient engine speed drive cycle allows the collection of enough training data to initiate the feed forward control ANNs. The corresponding control error calculated from the difference between the current control signal and the predicted ideal signal of the inverse ANN models is shown in figure [6.21]. The control signal error in figure [6.21] is from two main components. The main proportion of the error is the difference between the current control signal and the ideal control signal predicted by the inverse ANN models. Depending on the accuracy of the inverse model there would also be error from the prediction. By allowing the inverse models and the feed-forward model to train continuously online the inverse model error can be minimized for time varying dynamics while the feed-forward ANN can minimize control error.

Through following the same procedure as with the inverse ANN models, the feed forward ANN structure was found to be of three hidden layers of 30, 15 and 10 neurons, with all tangent sigmoid activation function. The output layer was again constructed from saturated linear activation functions. After training was completed a 100 sample test set of data was presented to the ANN.
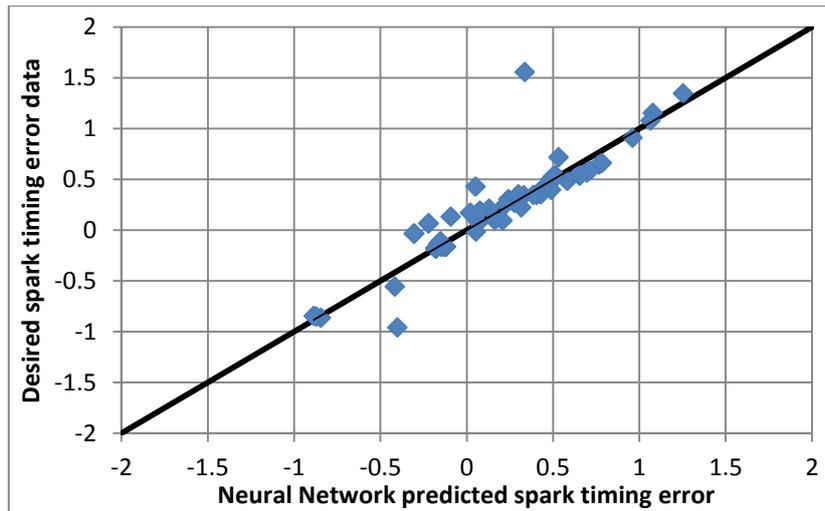
Figure 6.22 Correlation of neural network prediction of spark timing control error and desired spark timing control error data
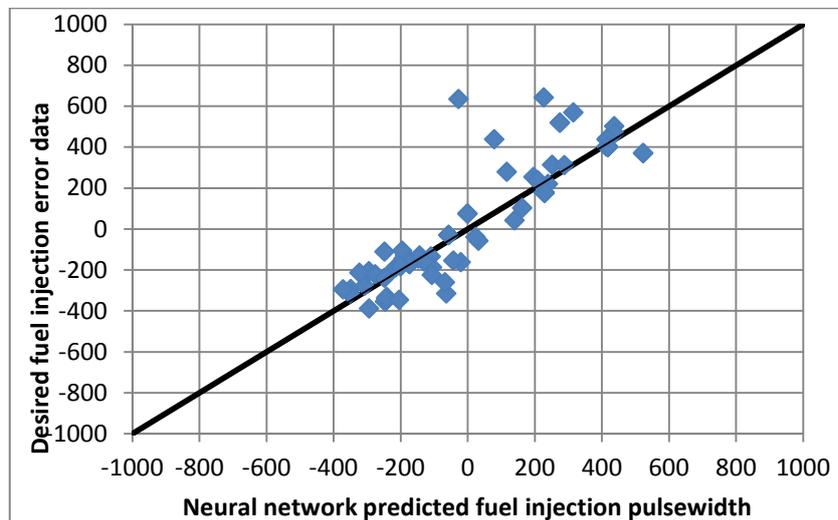
Figure 6.23 Correlation of neural network prediction of fuel injection pulse width error and desired fuel injection pulse width error data

Figure [6.22] and [6.23], show the comparison between the ANN prediction of spark timing error and fuel injection error compared to desired outputs. The ANN outputs have a correlation coefficient of 0.92 with the desired output test data for spark timing error and 0.90 with the desired output data for fuel injection pulse width error. Again these strong correlation coefficients show that the feed-forward ANN has successfully trained on the drive cycle data to capture the difference between the current control model and the inverse model of the engine.

6.3.5 ONLINE ADAPTION

Training the inverse ANN models and feed forward control ANN offline allows the control to begin with a reasonable level of knowledge about the current system. Without the offline training the ANNs would need to start from random weights. With random weights the ANNs would output random predictions until a level of knowledge is achieved. The time it would take to achieve this knowledge would depend on the spread of data in each training interval. As the ANNs used already have a level of knowledge we do not need to train over the entire operating range at each training interval but only need to adapt parts of the ANNs knowledge that are incorrect. To only adapt finite parts of an ANN's knowledge requires a different training method to that used in the offline training. In the offline training the Levenberg-Marquardt back propagation algorithm was used for its robustness against local minimum. The Levenberg-Marquardt back propagation method is a batch learning method. A batch learning method presents the entire set of training data but weight update only occurs after all the data points are processed. Using this method online would still allow for online training, but it has the potential to lose previous trained knowledge. When using this method all weights and biases are updated once the full data set is presented, therefore the entire ANN memory changes after each training set. For online adaption we only desire to adjust locations within the memory so that previous training knowledge is preserved within the other weights. The original basic back propagation algorithm is susceptible to problems with local minimum. But if the ANN has already been trained through Levenberg-Marquardt back propagation, the non linear transfer function has already been modeled. Therefore, if the standard back propagation algorithm is used after for adaption, the risk of being stuck at a local minimum is reduced as the global minimum has already been determined and modeled. Using the standard back propagation algorithm allows an online training method to be used where each data set is presented to the ANN individually and a weight update is made after each data pair. This method therefore only adjusts the weights and biases of the activated neurons for that given data set, the remaining inactive weights and bias values are preserved.

For a comparison between training online with Levenberg-Marquardt method and the standard back-propagation for previous training knowledge retention a simple test was conducted.
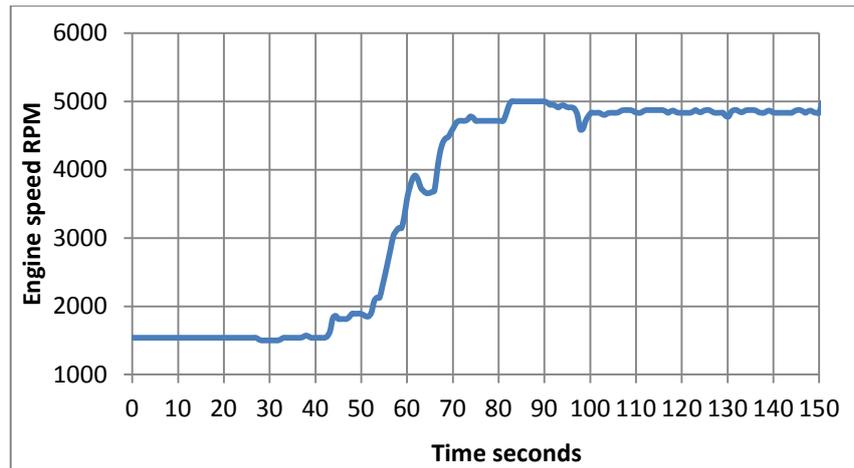
Figure 6.24 Online adaption development test cycle for transient in engine speed
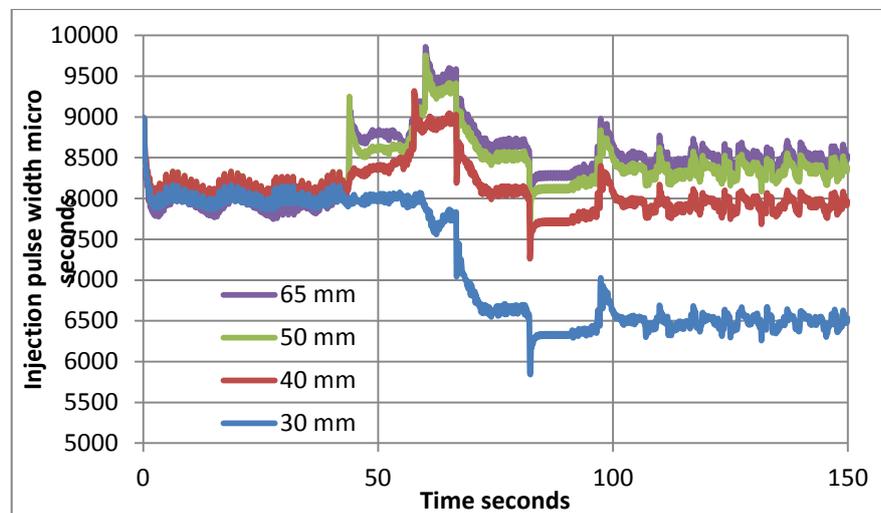
Figure 6.25 Injection pulse width requirement for constant AFR 14.5 under varying throttle for the online adaption development test cycle

The test comprised of an offline inverse model being trained on a transient speed cycle such as figure [6.24], at constant throttle positions 65, 50, 40 and 30mm. The training data at each of the different throttle positions will have different mass air flow dynamics; this creates different fuel injection pulse width requirements to maintain a constant AFR, figure [6.25]. By sequentially training the ANN on the different throttle position data while attempting to predict the original training set allows us to investigate the ANNs ability at memory retention during adaption. As the dynamics are different for each training data set the ANNs will need to adjust their weights and bias values to model the new dynamics. If a ANN training algorithm does not preserve previous trained knowledge its performance at predicting the

original training data will decrease. Table [6.8] shows the comparison of correlation coefficients after four training sets.

Table 6.8 Comparison of correlation coefficients for online training with Levenberg-Marquardt technique and Online adapt back propagation technique after four training sets

Throttle position training data before 30mm prediction	Levenberg-Marquardt technique, correlation with 30 mm training data	Online adapt back propagation technique, correlation with 30 mm training data
30 mm	R= 0.99	R= 0.97
40 mm	R= -0.79	R= 0.98
50 mm	R= 0.56	R= 0.98
65 mm	R= -0.83	R= 0.80

Using the Levenberg-Marquardt method produces a more accurate offline training algorithm, but once online training begins the ability to retain previous trained knowledge is lost. The back propagation algorithm using an online adapting technique of presenting data sets individually can improve prediction performance of the original data set. With all correlation maintaining values above 0.8 the ANN is able to maintain original knowledge while adding new knowledge. For this reason the standard back propagation algorithm was implemented for online adaption.

6.4 DRIVE CYCLE

For testing and evaluating the control architecture it was important to generate tests for steady state conditions but also for transients. The controller will need to model the engine behaviour over the full operating range of the engine. Simple steady state tests allow for an investigation of the architectures ability to adapt and predict to changing dynamics at a single location. However, they would be insufficient to test the ANNs for maintaining control and learning the full range of dynamics of the engine. The following drive cycle will be used for testing transient control behaviour. The data was obtained from the Non Road Transient Cycle (NRTC) published by the American Environmental Protection Agency (EPA) (93). By using a real world drive cycle it is possible to test the ANNs and control architecture over realistic operating range of the engine, with realistic transient time steps.
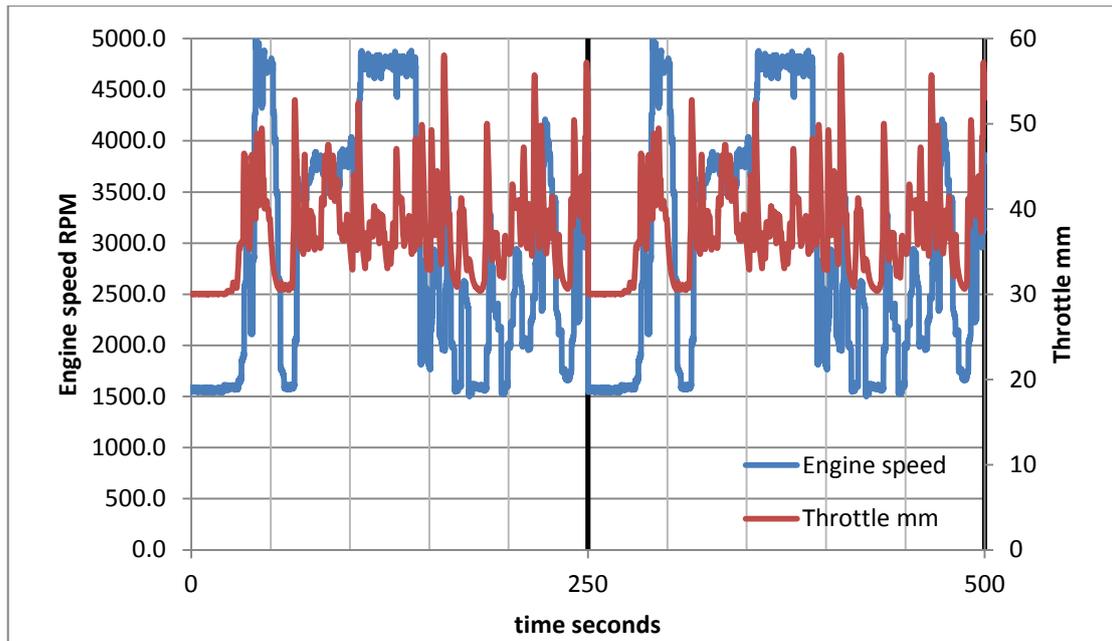
Figure 6.26   Sample of the non-road transient drive cycle

The transient cycle in figure [6.20], is comprised of a 250 second sample from the NRTC drive cycle. The sample is repeated once to offer a 500 second drive cycle in total. The purpose of repeating the sample is to investigate if the dynamics of the drive cycle in the first 250 second can be captured by the control architecture to help improve the control over the final 250 seconds. If the ANNs in the architecture are unable to improve their knowledge of the systems dynamic behaviour and range the control performance from 250-500 seconds will not improve over the first 250 seconds. A section of the NRTC drive cycle was used over on road drive cycles primarily due to its aggressive transient nature. As developing adaptive control is the main interest of this work and not emission testing and modelling, the content of the cycle is of major interest rather than its target vehicle application. A second benefit of the NRTC is the data is in the format of normalised engine speed and load. Often on road drive cycles are in the format of vehicle speed and load data. The engine speed and load format is more useful with the GT power engine model then vehicle speed and load data. To use vehicle speed and load data a transmission model would also need to be considered.

7. GT-power Engine Simulation Control Results

7.1 Multi Input Single Output Steady State Test using GT-Power Engine Model

In the development of the control structure the fuzzy logic and ANNs have been tested for a MISO control problem. Using a MISO control problem at first allowed the basic control architecture to be developed and the experimental methods for weight update and online training to be tested. By keeping the initial control problem simple, it allowed for any errors in the programming or architecture to be quickly identified and fixed. The first control variable used was spark timing. Through feedback of cylinder peak pressure position, the controller was required to phase the peak pressure position at a set point of 10 degrees past TDC. The initial test was for set point tracking under steady state conditions. To evaluate the control performance and the ANNs training ability an immeasurable disturbance was introduced to force the dynamics of the system to change.
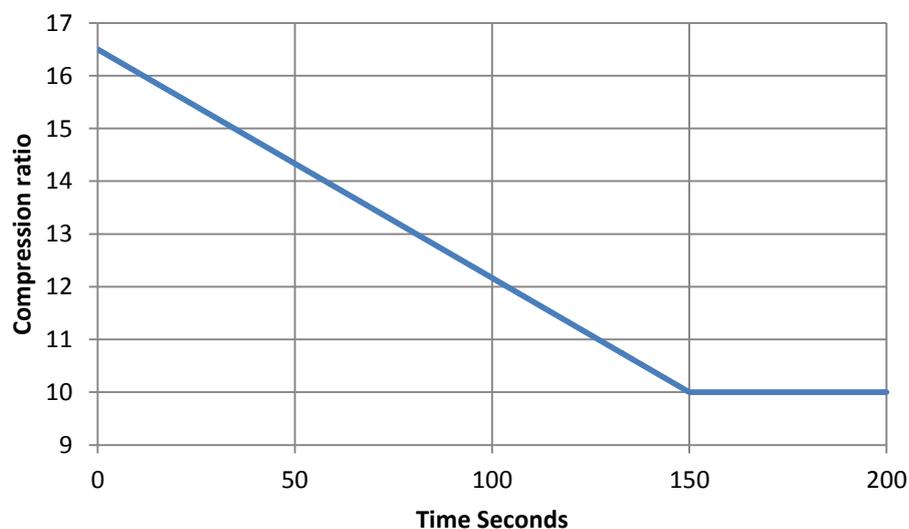
Figure.7.1 Dynamic shift in engine compression ratio during operation

Figure [7.1] shows a change in CR from 16.5 to 10 at a steady decrease over 150 seconds. Such a drastic CR change will have a dramatic effect on the ideal control for spark timing to maintain the peak pressure position at the desired set point of 10 degrees past TDC.
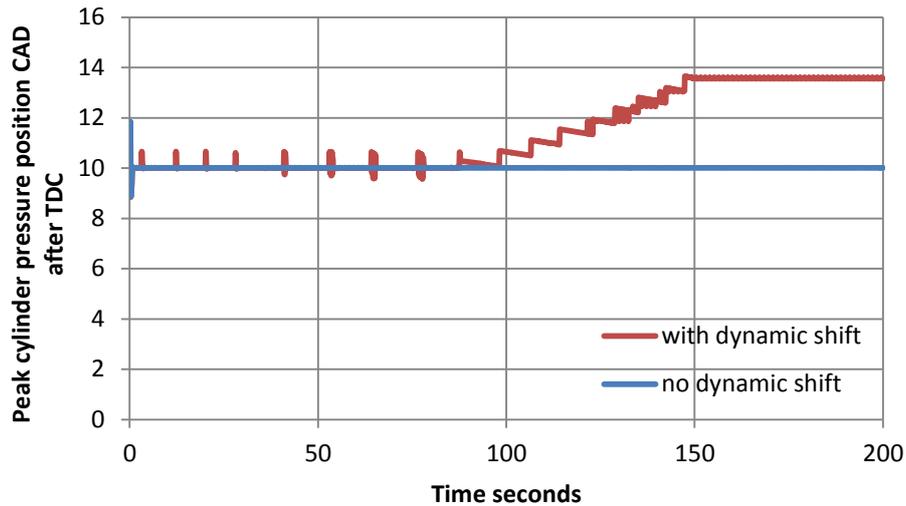
Figure 7.2 Control response for peak pressure position at steady state conditions with and without dynamic shift in compression ratio

Figure [7.2] shows the control response for peak pressure position at a steady state condition of 3500 rpm and throttle 50mm, with and without the dynamic shift. Without the dynamic shift, the control can adequately maintain the set point control at 10 degrees past TDC. However, during the dynamic shift the control is able to compensate for the dynamic shift through feedback control until 100 seconds, but then becomes saturated. Once the feedback control has become saturated it is unable to offer any further influence over the increasing control error.
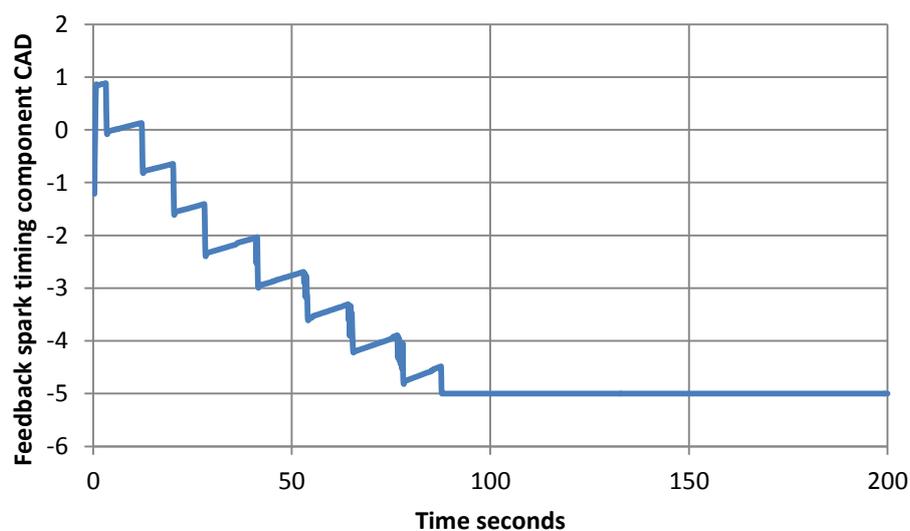
Figure 7.3 Feedback control component at steady state condition with dynamic shift in compression ratio

The spark timing control required to maintain peak pressure position at 10 degrees after TDC at CR 10 is significantly different to that at CR 16.5. The fixed dynamic control map within the fuzzy system becomes mismatched to the system dynamics forcing the feedback control to increase its component to counteract the model/system mismatch, figure [7.3]. Once the mismatch becomes too large the control saturates.

Distinct 'steps' can be seen in the control signal as the CR changes from 16.5 to 10. This behaviour could be due to the design of the input membership of discourse for timing error. As there is a 'thin' MF for zero error corresponding to an output MF for no change, this would have an effect of limiting the rate of response of the controller to small changes in error. Only once the level of error increased past or to a smaller level of activation of the zero error would the controller rate of response be increased. As the adjacent MF for negative and positive error would have larger level of activation as the zero error MF blends out, it could create a sudden step change in rate of response. In future work, the width of Zero MF and blend between zero, positive and negative error MFs could be investigated to achieve a smoother transition. An improved design could potentially reduce these distinct 'steps'.
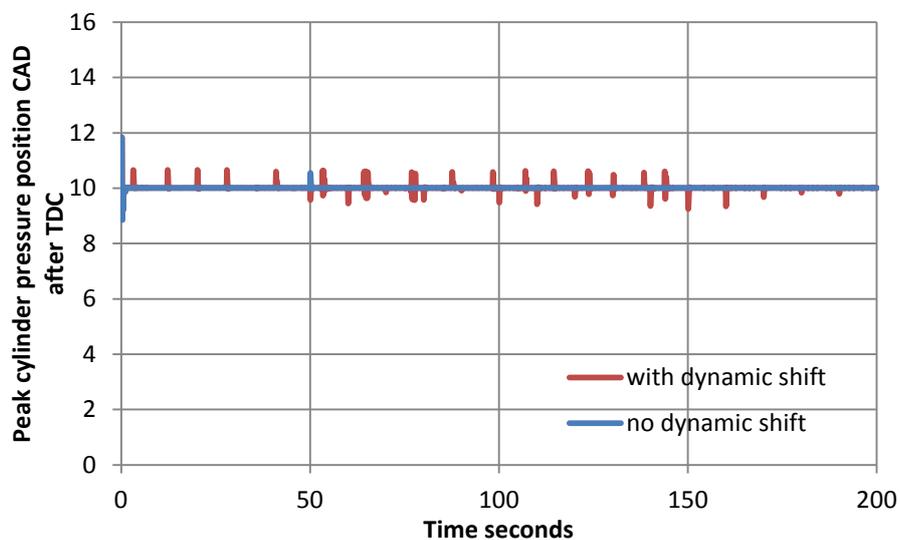
Figure 7.4 Adaptive control response for peak pressure position at steady state conditions with and without dynamic shift in compression ratio

By allowing the ANN inverse models to adapt to the change in dynamics, the difference between the FLC model and the changing engine dynamics can be captured in the feed-forward ANN. Figure [7.4] shows the control response for peak pressure position for the

same dynamic shift in CR with online adaption. The ANNs are able to adapt to the change in dynamics and allow the control to maintain peak pressure position at the desired set point.
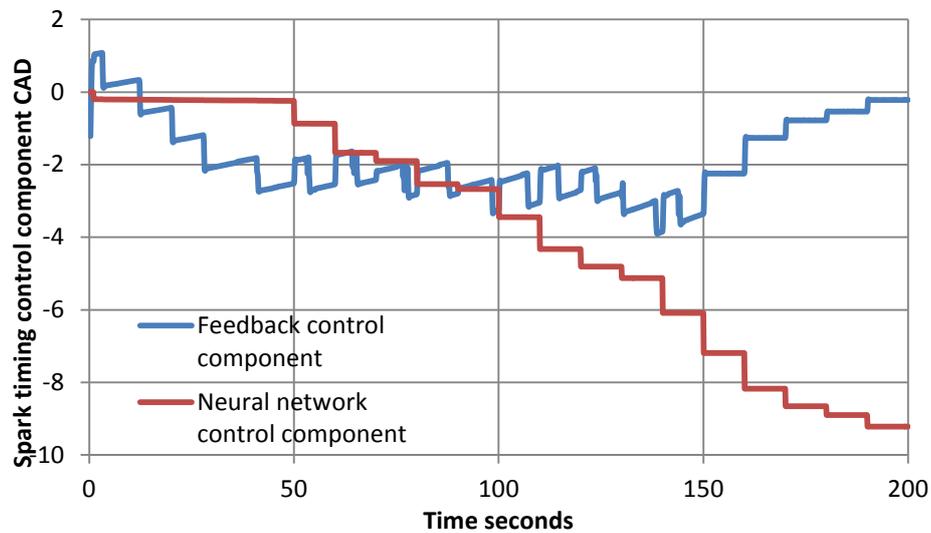
Figure 7.5 Feedback and adapting neural network control components for steady state test during dynamic shift

Figure [7.5] shows the control components for the fuzzy feedback control and the feed forward adaptive ANN. The ANNs begin training at 50 seconds and then every following 10 second interval. The first 50 seconds allows a data buffer to collect enough training data for the ANN adaption. The buffer has a rolling overlap of 40 seconds worth of data, for every training increment 10 seconds of old data is replaced with 10 second of new data. As the feed forward ANN is adapted using an online method each pair of training data is presented individually to preserve previous training knowledge. Using a buffered set of training data offers the ANN a number of data sample pairs at each training interval. Presenting multiple data pairs at each training interval allows the ANN to adapt a number of times. By adapting multiple times reduces the chance of an incorrect training due to an error prone data pair. If the majority of training pairs are of satisfactory accuracy, their training will dilute the effect of error prone data pairs. This can offer some robustness against incorrect training before the final weight update are made to the Simulink model. Each data pair is also only presented to the ANNs twice per training cycle. If the training data pairs are not corrupt through error their relationships will be repeated more often within the data set, increasing their effect on the ANN training. Corrupted data pairs with error however will be diverse depending on the level of error or noise and will be less repeatable. Therefore by using a rolling buffer the effects of error or noise on the training data will be averaged and reduced. The buffer size

was determined through increasing the number of pairs until an acceptable training effort was achieved. If the buffer was too large it could have detrimental effect on training speed or response of training to changes in dynamics. Allowing the buffer to be to small increased the chance of training to error or noise and not true global changes in dynamics.

From figure [7.5], we can see that as the dynamic shift in CR occurs the feedback control begins to increase its component to reduce the error between the fuzzy control model and the inverse dynamics of the engine similar to figure [7.3].However, once the ANN begins to adapt it reduces the mismatch between the fuzzy control model and the system inverse allowing the feedback control component to reduce. The results show that the developed control is able to adapt online at a stationary steady state condition for an immeasurable dynamic shift on the engine dynamics. The online adaption through continuous ANN training allows the control to self-calibrate a feed forward ANN to the difference between the current control and the inverse system dynamics.

7.2 MULTI INPUT SINGLE OUTPUT TRANSIENT DRIVE CYCLE TEST USING GT-POWER ENGINE MODEL

Under the steady state condition, the control was only required to adapt to a single operating point of the control model. To achieve adaptive control over the transient cycle is more complex. The adaptive ANNs need to adapt for the multiple speed and load conditions of the transient drive cycle during the dynamic shift. This will test the ANNs ability to adapt to only the system immeasurable global time varying dynamics during transient drive cycle operation.
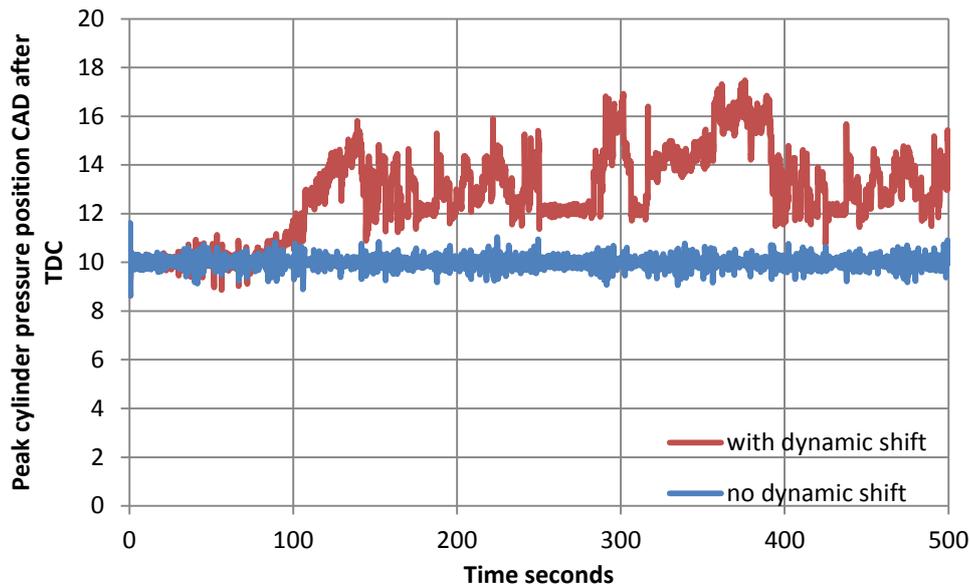
Figure 7.6 Control response for peak pressure position during a sample from the NRTC drive cycle with and without dynamic shift in compression ratio

Figure [7.6] shows the control response for peak cylinder pressure position during the transient drive cycle with and without an immeasurable dynamic shift in CR. Without the dynamic shift the FLC is able to maintain peak cylinder pressure position close to the desired set point. When the immeasurable dynamic shift is introduced the FLC model's mismatch with the systems dynamics increases as with the steady state tests.
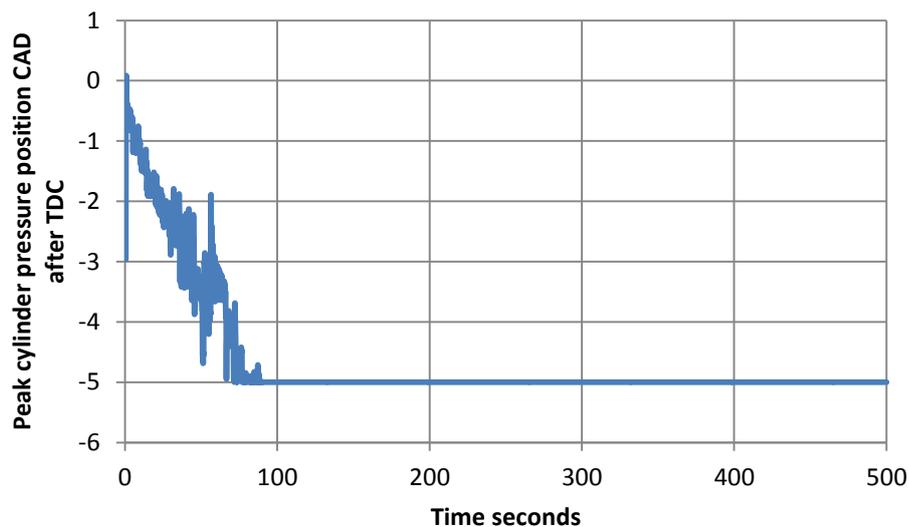
Figure 7.7 Feedback control component during a sample of the NRTC drive cycle with dynamic shift in compression ratio

As the model/plant mismatch increases, the feedback control saturates, figure [7.7]. Once the feedback control becomes saturated the control relies on the inaccurate feed-forward model offering poor set point tracking and no response to disturbances leading to an increase in control error, figure [7.6].
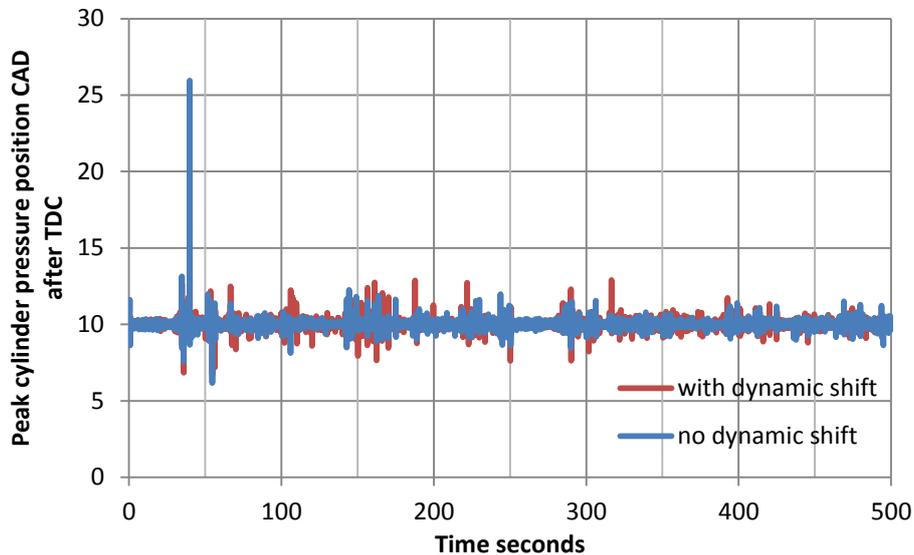
Figure 7.8 Adaptive control response for peak pressure position during a sample from the NRTC drive cycle with and without dynamic shift in compression ratio

Figure [7.8] shows the adaptive control response for peak cylinder pressure position during the transient drive cycle with and without the dynamic shift in CR. The result shows that the adaptive ANNs are able to learn the difference between the current control and the system inverse prediction for ideal control, during a transient drive cycle. With the addition of the adaptive control component, the control is maintained around the desired set point. Training of the feed forward ANN was conducted in the same manner as the steady state tests, with training beginning at 50 seconds. Before the first training interval there is a large spike of error that was not present during the non-adaptive control response of figure [7.6]. Figure [7.9] shows the control components for the feed forward ANN and the FLC. The large spike of error was caused from a large spike in the feed forward ANN component. In the first 50 seconds of the drive cycle the ANNs only have offline trained knowledge. The offline training of the feed-forward ANN was on a limited drive cycle. Therefore, a potential reason the ANN component produced a large spike is due to inputs that were outside of the trained knowledge. When an ANN prediction is outside of the trained knowledge, it can be difficult for the ANN to draw reasonable relationships. This can lead to the saturation of the outputs. Once the feed forward ANN began its online training, the knowledge of the system dynamics

improved. As the drive cycle is formed from a 250 second sample from the NRTC drive cycle repeated once, the same system dynamics are repeated between 250 and 300 seconds. During the second half of the drive cycle, there are no large error spikes. The feed-forward ANN has successfully adapted the weights/biases to incorporate the dynamic range of the drive cycle between 0-250 seconds.
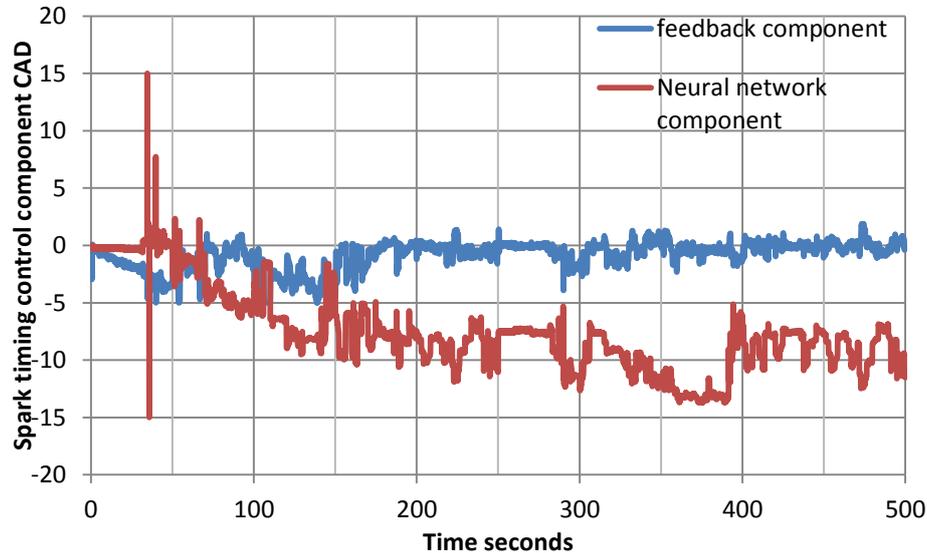
Figure 7.9 Feedback and adapting neural network control components for transient drive cycle during dynamic shift

Therefore from figure [7.9], the feedback ANN has adapted to the dynamic range of the drive cycle as well as the immeasurable change in dynamics from CR. The adaptive feed forward ANN allowed the FLC feedback control component to reduce, offering better response to future disturbances.

Overall the adaptive control architecture is capable of adapting a MISO control system to an immeasurable change in dynamics during a transient drive cycle. The control was able to calibrate the feed forward control model to the difference between the current control and the ideal control predicted from the inverse ANN model of the engine. This online adaption has been successful at calibrating the control over a wide operation of speed and load conditions covered by the sample from the NRTC drive cycle test. The control was also capable of continually adapting the MISO control during the transient operation of the engine without causing control instability during ANN updates.

7.3 MULTI-INPUT MULTI- OUTPUT STEADY STATE TEST USING GT-POWER ENGINE MODEL

The MIMO control structure was configured for control over combustion phasing and AFR. The desired set point for the AFR control was 14.5. By increasing the number of control variables the ANN adaption and control modelling becomes a more complex problem. When an ANN has multiple outputs the weights and bias values need to be able to represent the nonlinear relationship information for both outputs. To assess the control architectures ability for MIMO control, modelling and online adaption, it was useful to begin with a steady state test. The steady state test was at constant speed 3500 rpm and throttle 50mm. The change in CR was used for the immeasurable dynamic shift in combination with an additional change in fuel rate from 6 to 7.5 grams per second, figure [7.10].
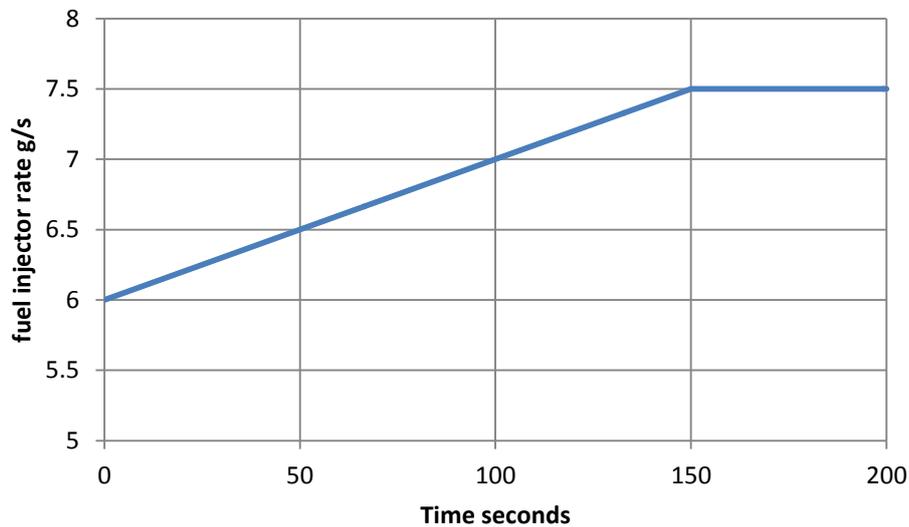
Figure 7.10 immeasurable dynamic shift in injector fuel rate from 6 to 7.5 g/s

The dynamic change in the fuel delivery rate of the injector model will alter the amount of fuel delivered for a given injection pulse width. Therefore, the FLC model will no longer represent the fuel pulse width required to maintain an AFR of 14.5.
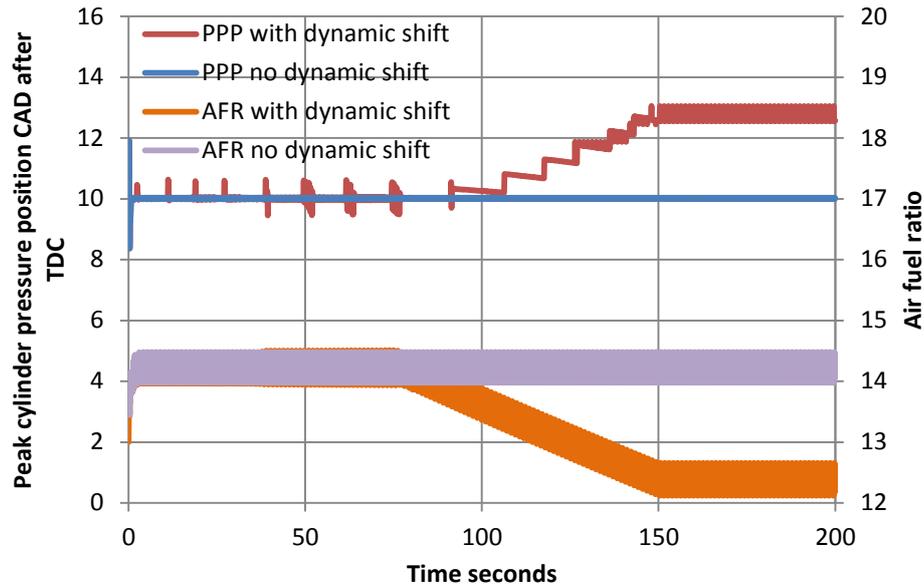
Figure 7.11 Control response for peak pressure position and AFR at steady state condition with and without dynamic shift in compression ratio and fuel rate.

Figure [7.11] shows the control response for peak cylinder pressure position and AFR during the steady state test, with and without the immeasurable dynamic shifts in CR and fuel injector delivery rate. Without the dynamic shift, the FLC is able to maintain the desired set point targets for both peak cylinder pressure position and AFR. Therefore the FLC has been able to model the stationary nonlinear dynamics for MIMO engine control. However with the dynamic shifts, the fixed FLC system cannot maintain the desired set points.

The AFR feedback displays a continuous uniform oscillation. This behaviour is believed to be due to where the feedback was taken from within the GT-Power engine model. The AFR feedback was taken from a parameter of a flow component for the manifold. As the flow in the manifold is effected by the pattern of exhaust valve openings it will cause this oscillation behaviour. In future work, averaging the signal or obtaining feedback further down the exhaust system could potentially decrease the oscillations observed.
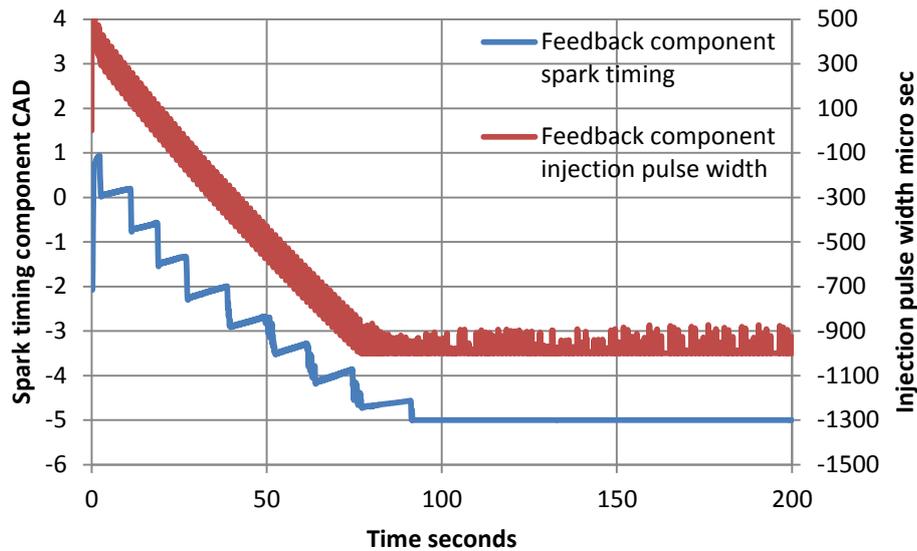
Figure 7.12 Feedback control component during steady state conditions with dynamic shift in compression ratio and fuel injector flow rate.

Figure [7.12], shows the feedback components for spark timing and injection pulse width. Like the MISO tests the fixed feed forward model becomes inaccurate as the systems dynamics change with time. As the model/system mismatch increases the feedback control compensates until saturation occurs. Once the feedback control has become saturated it can offer no further control over disturbances or model/system mismatch. These results show that the immeasurable dynamic shifts in CR and fuel injector can cause the fixed FLC to saturate and become ineffective at control.
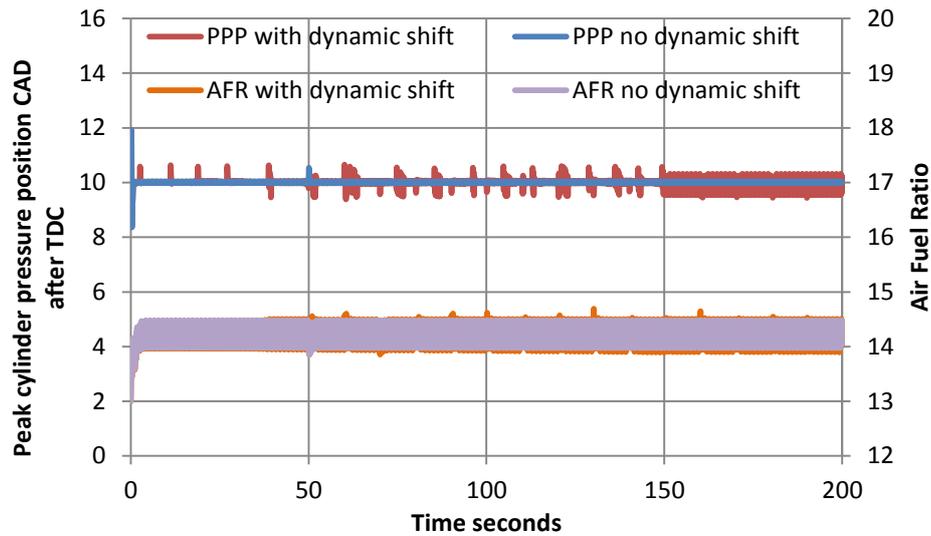
Figure 7.13 Adaptive control response for peak pressure position and AFR at steady state condition with and without dynamic shift in compression ratio and fuel rate

Figure [7.13], shows the adaptive control architectures control for cylinder peak pressure position and AFR under steady state condition, with and without the immeasurable dynamic shifts in CR and fuel injector flow rate. The adaptive control architecture has been able to maintain AFR and cylinder peak pressure position to the desired set points for tests with and without immeasurable dynamic shifts. The ANNs in the adaptive architecture are capable of modeling the inverse dynamics of a MIMO engine control problem. The architecture is also able to predict appropriate feed-forward control components, allowing the MIMO feed-forward ANN to maintain set point tracking for two control variables during an immeasurable dynamic shift.
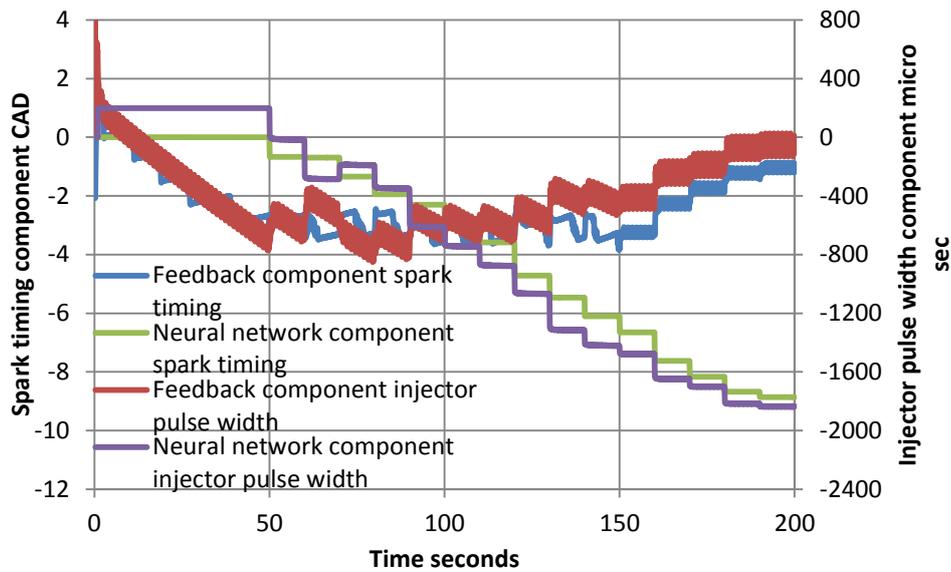
Figure 7.14 Feedback and adaptive neural network control component during steady state conditions with dynamic shift in compression ratio and fuel injector flow rate.

Figure [7.14] shows the FLC feedback and adaptive feed forward ANN control components during the immeasurable dynamic shift at steady state conditions. The FLC feedback begins to counteract the model/system mismatch during the start of the dynamic shift. Once the ANN online adaption begins at 50 seconds it reduces the model/system mismatch. The ANN adaption allows the feedback components for spark timing and injection pulse width to reduce for greater response against future disturbances.

The FLC was capable of modelling and control for the static dynamic control over a MIMO engine at steady state conditions. When immeasurable dynamic shift where introduced the FLC was not capable of maintain the desired set point tracking. The FLC model became inaccurate to the changing dynamics of the system.

The adaptive architecture was capable of adapting and calibrating the control model for immeasurable dynamic shifts in CR and fuel injector flow rate. The inverse ANNs modelled the changing system dynamics allowing for predictions on ideal control. From the prediction, the feed forward ANN was able to adapt the two feed forward control components at the steady state operating condition to maintain set point tracking. The ANNs have demonstrated the potential for online modelling and adaption for MIMO engine control problems without causing instability.

7.4 MULTI-INPUT MULTI-OUTPUT TRANSIENT CYCLE TEST USING GT-POWER ENGINE MODEL

The MIMO steady state tests demonstrated the adaptive control architectures ability to maintain control and adapt for MIMO systems with immeasurable time varying dynamics at a single operating condition. By evaluating the control during the transient drive cycle will allow an investigation of the adaptive control architecture performance for MIMO control at multiple operating conditions and for online adaption during transient behaviour. The NRTC drive cycle sample combined with the immeasurable dynamic shifts in CR and fuel injector flow rate offer a significantly demanding test. The adaptive control architecture will need to accurately model the MIMO inverse dynamics of the system over the drive cycle and predict ideal control. The online adaption and calibration of the feed forward ANN will also need to take place without causing instability during the transients of the drive cycle.
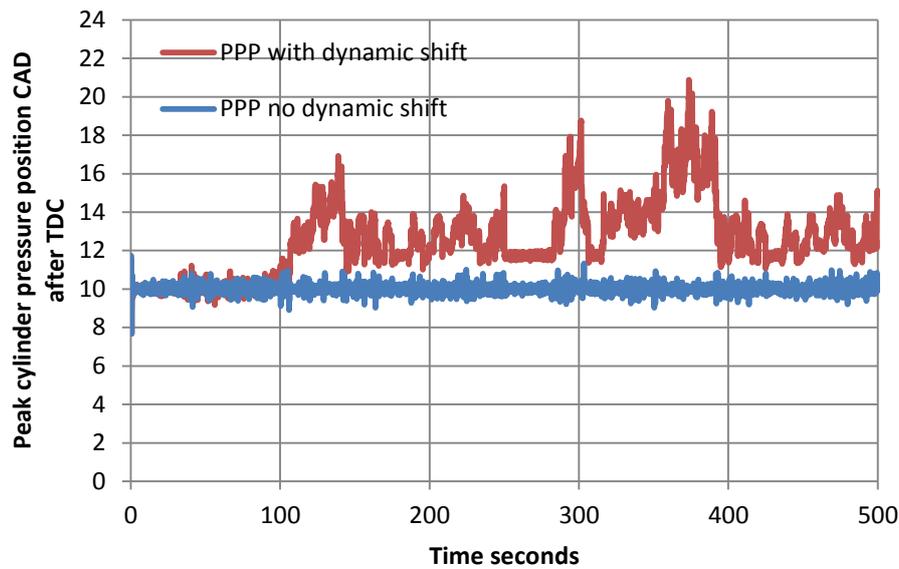
Figure 7.15 MIMO Control response for peak pressure position during a sample from the NRTC drive cycle with and without dynamic shift in compression ratio and injector flow rate
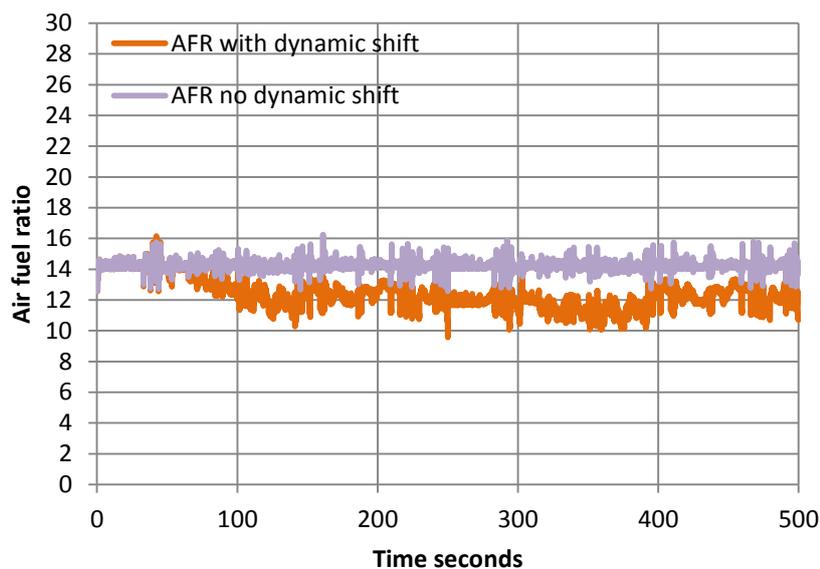
Figure 7.16 MIMO Control response for AFR during a sample from the NRTC drive cycle with and without dynamic shift in compression ratio and injector flow rate

Figure [7.15] and [7.16] show the MIMO Control response for peak cylinder pressure position and AFR respectively during the transient drive cycle, with and without immeasurable dynamic shift in CR and injector flow rate. The fixed FLC was able to model the stationary dynamics for the MIMO control problem accurately to maintain set point tracking during the transient drive cycle. However, once immeasurable dynamic changes caused from CR and fuel

injector flow rate are introduced the fixed FLC becomes incapable of maintaining the desired set point tracking for peak cylinder pressure position or AFR.
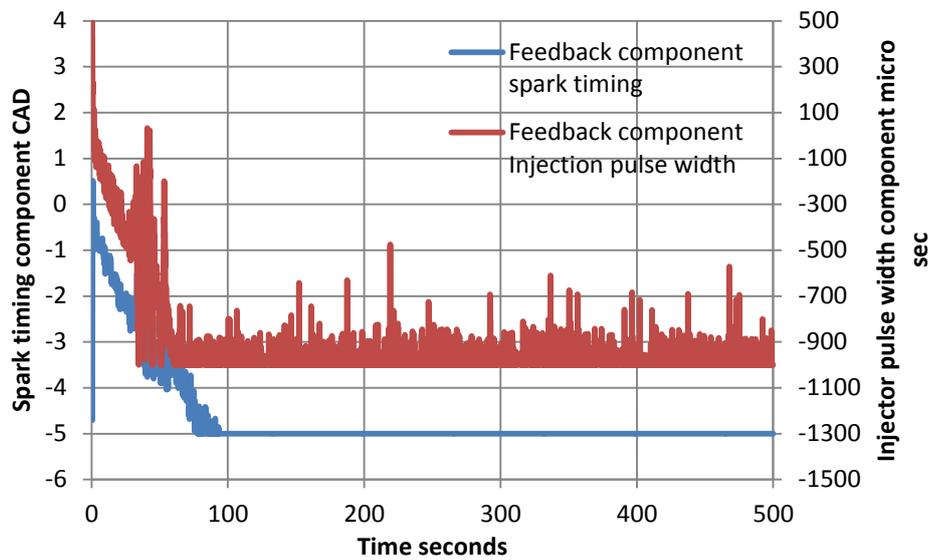
Figure 7.17 Feedback control components during NRTC drive cycle conditions with dynamic shift in compression ratio and fuel injector flow rate

Figure [7.17], shows the feedback control component for the fixed FLC system during the transient drive cycle with immeasurable dynamics shifts in CR and injector flow rate. Both the feedback control components become saturated due to the increased model/system mismatch between the fixed FLC and the changing system dynamics.
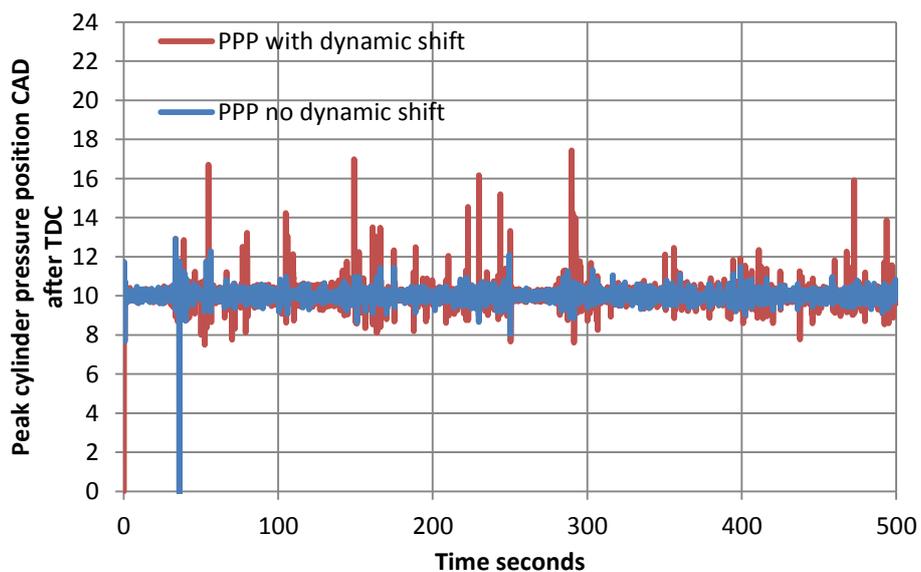
Figure 7.18 Adaptive control response for peak pressure position during NRTC drive cycle conditions with and without dynamic shift in compression ratio and fuel rate
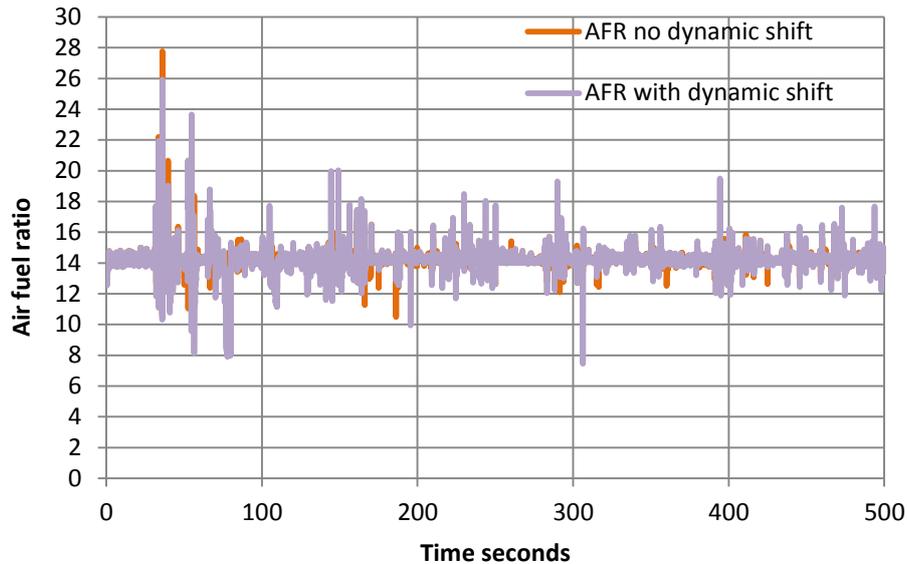
Figure 7.19 Adaptive control response for AFR during NRTC drive cycle conditions with and without dynamic shift in compression ratio and fuel rate

Figures [7.18] and [7.19] show the adaptive control response for peak cylinder pressure position and AFR during the transient drive cycle with immeasurable dynamic shifts in CR and injector flow rate. The adaptive control is capable of maintaining control to the desired set point for peak cylinder pressure position and AFR. However, there are some significant errors in both the peak cylinder pressure position and the AFR in the first 250 seconds of the drive cycle. When the simulation is initiated, the ANNs only have knowledge from their offline training. The NRTC drive cycle sample is a continuous transient cycle in both load and speed. During the first pass (0 -250 seconds) of the NRTC drive cycle sample, the ANNs are adapting their knowledge to the systems operating range and also to the dynamics of this range as they change with time due to the immeasurable effects of CR and injector flow rate. In the second pass of the NRTC sample (250-500 seconds), the error peaks in peak cylinder pressure position and AFR control are reduced. As the immeasurable changing dynamics due to the CR and injector flow rate become stable, the ANN are able to calibrate to the stationary non-linear dynamics and allow the control to reduce the set point error.
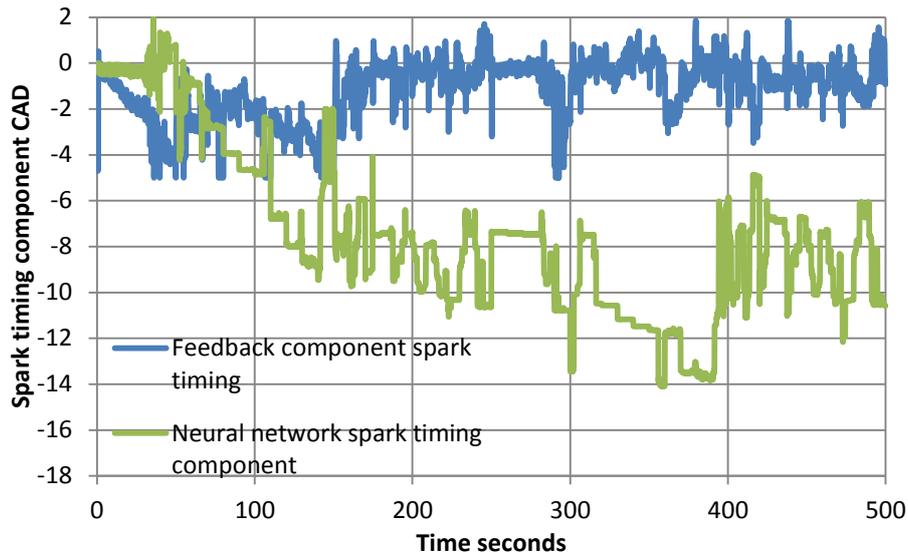
Figure 7.20 Feedback and adaptive neural network control component for spark timing during NRTC drive cycle with dynamic shift in compression ratio and fuel injector flow rate
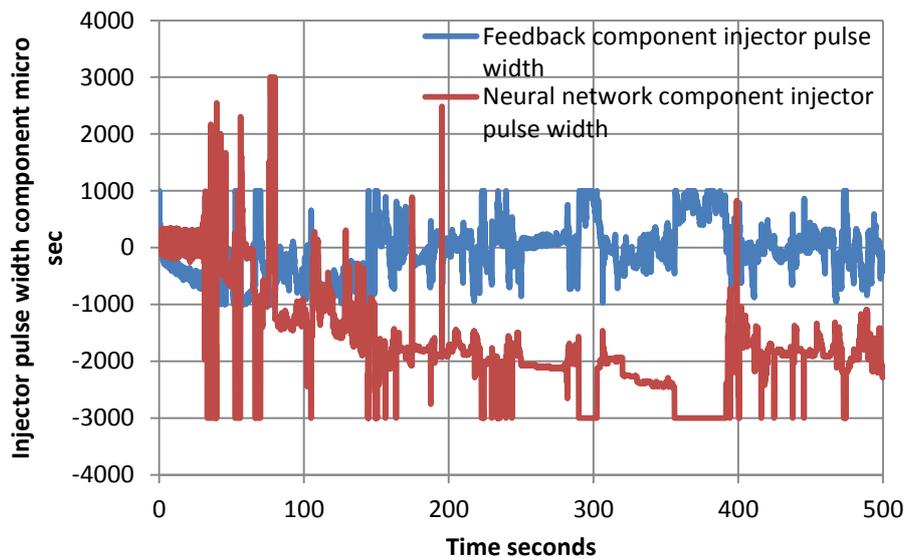
Figure 7.21 Feedback and adaptive neural network control component for injector pulse width during NRTC drive cycle with dynamic shift in compression ratio and fuel injector flow rate

Figure [7.20] and [7.21], show the FLC feedback and feed forward ANN component during the transient drive cycle with the immeasurable dynamic shift in CR and injector flow rate. As the MIMO feed forward ANN adapts the control component to the changing system dynamics, the FLC feedback component is able to avoid saturation and maintain improved set point control during the transient behaviour of the drive cycle. At the beginning of the ANN adaption the feed forward ANN appears to train badly for the fuel injection component, figure

[7.21]. During the first 100 seconds of the drive cycle there are large oscillations in the ANNs output. The oscillations show that the trained knowledge is inaccurate. As the ANN further adapts the knowledge improves decreasing the undesirable oscillations and allowing the desired control to be achieved. The result from the first 250 seconds of the cycle could potentially be improved if the offline training of the feed-forward ANN encompassed more of the operating range present in the NRTC drive cycle. However, the results found have provided evidence that the architecture is capable of learning new stationary dynamic range for the measurable control variables while also accounting for the immeasurable changes that are causing the stationary dynamics to change with time. As the ANNs are able to extend its predictive operating range through online training it shows the control architecture is capable of online calibration.

Overall the results have provided evidence that the adaptive control architecture is capable of maintaining desired control set points for a MIMO engine control problem during a complex transient cycle in speed and load. The adaptive control architecture is capable of adapting the ANN inverse model of the engine online to an accuracy that allows prediction of ideal control for a MIMO system during the transient drive cycle. The combined feed forward FLC and ANN model provides an online adaptive feed forward control component based on the inverse dynamics of the engine. The control can adapt either a single operating location of the control model or the entire operating range depending on the influence of immeasurable dynamic changes that can alter the engines dynamics with time. The control architecture allows control to be maintained without instability through ANN updates or control system conflict for time varying dynamic engine control under steady state and transient conditions.

8. Initial Experimental Feasibility Tests with Ricardo Variable Compression Ratio E6 Engine

Figure 8.1 Ricardo E6 engine at Loughborough University

 The Ricardo E6 research engine figure [8.1], was originally developed over sixty years ago by Sir Harry Ricardo as a variable CR diesel engine. The E6 engine continues to be used in current research and is often equipped with modern fuel injection systems to facilitate operation in SI or CI combustion modes with a variety of fuels. A major benefit of the E6 engine is that the CR can also be altered during engine operation.

For the work presented, the E6 engine is configured for Gasoline SI operation and has upgraded electronic control for port fuel injection and spark ignition timing. The basic parameters of the E6 engine can be seen in table [8.1].

Table 8.1 parameters of the E6 engine

Parameter	Value
Bore	76.2 mm
Stroke	111.2 mm
Connecting rod length	240.5 mm
Displacement volume	536 cm3
Compression Ratio	4.5-23
IVO	8 BTDC
IVC	36 ATDC
EVO	43 BBDC
EVC	7 ATDC

The E6 engine can be used in this initial practical study to validate the developed adaptive control architecture for combustion timing control at steady state conditions with ionisation current signal as a feedback and adaption to immeasurable dynamic changes during steady state operation of a real engine.
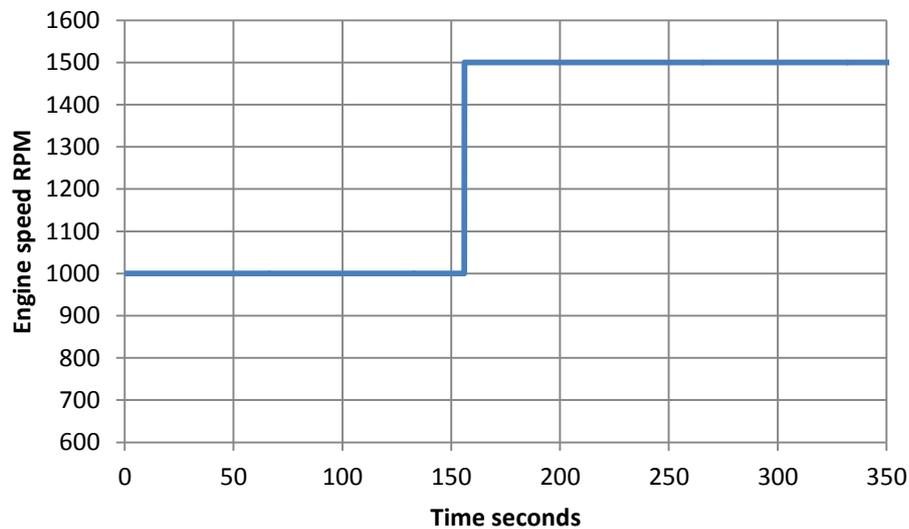
Figure 8.2 Engine speed change test for E6 engine control

Two tests are conducted for evaluating the control architecture performance for online adaption and calibration with a real engine using the experimental feedback technique of ionisation. The first test figure [8.2], will investigate the architectures ability to adapt the control and calibrate the model to two single steady state speeds at 50% throttle using

ionisation current signal as a feedback. The control architecture has no prior knowledge of the E6 engine dynamics. The control model is based on the simulation work with GT-power; it will offer a rough starting point based on SI combustion dynamics. For the combustion to be phased to the desired set point, the control will need to use information from the ionisation feedback signal and adapt the control model online to calibrate the control signal. Once the control is achieved at the steady state position, the engine speed will be changed to force the combustion to be incorrectly timed. The control architecture should then re-calibrate the control for the new speed and load position.
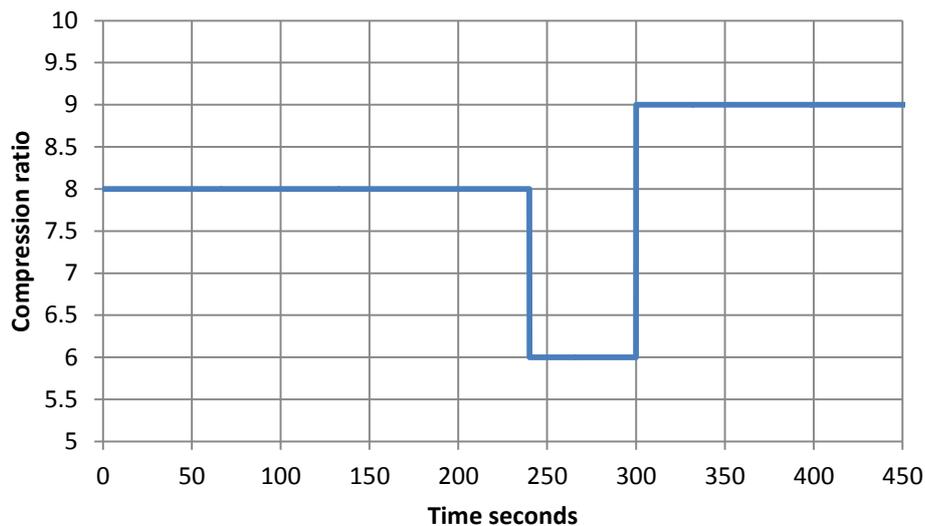
Figure 8.3 Compression ratio change test for E6 engine control

The second test will be based at a single speed of 1500 rpm and 50% throttle position. Once the combustion control set point is achieved the engine dynamics will be altered through a series of changes in CR, from 8 to 6 then return to 9, figure [8.3]. This will evaluate the control architecture for adaption to immeasurable time varying dynamics with a real engine using ionisation as a feedback. By changing the direction of the CR change, the control will need to adapt for advanced and retarded ignition timings.

8.1 CONTROL IMPLEMENTATION WITH RICARDO E6 ENGINE

The control architecture has been developed using Matlab and Simulink. In order to validate the control performance for a practical application it requires implementation with the E6

engine's feedback and control signals. National instruments Labview software offers a simulation interface toolbox that can pass signals between Labview and Simulink. The toolbox is primarily designed to allow Labview to be a front panel for simulations based in Simulink. However, labview also offers a platform that allows for data acquisition, signal processing and creation of digital output signals. By manipulating the simulation interface toolbox, it is possible to configure a Labview Virtual Instrument (VI) to perform as an intermediate step between the Simulink control and the engine. Labview can perform data acquisition of the ionisation current signal, process the relevant data and pass it back to Simulink. The Simulink control, can then output a desired control response back to the Labview VI. Once the desired control signal is returned to Labview, it can be manipulated into a digital output pulse width for control. The Simulink and Labview method will allow for initial validation testing for the evaluation of the performance of the control architecture for control and adaption with a real engine using ionisation as feedback.
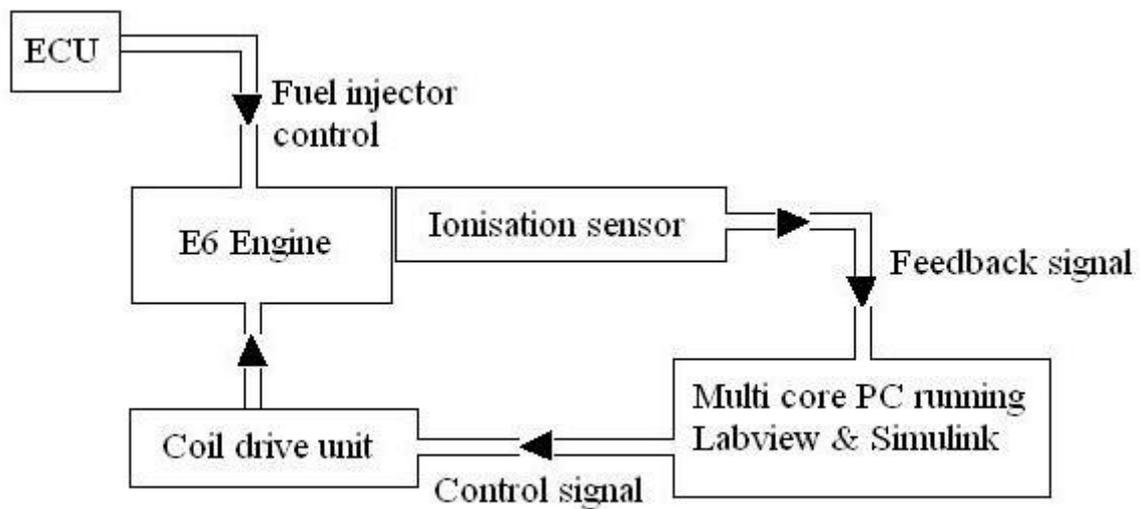
Figure 8.4 Flow diagram of the E6 engine control test

Figure [8.4], shows a flow diagram of the data during the control procedure. The E6 engine ECU maintains control over the fuel injector, while the Labview/Simulink programs maintain control for spark ignition timing. The ionisation feedback can be acquired for every engine cycle. For the feedback control, 30 engine cycles are averaged and the 75 % of peak value position in CAD was chosen as the feedback parameter to be returned to Simulink. In SI engines the ionisation feedback signal can have different characteristics to that seen for HCCI combustion. SI engine ionisation can often display two peaks, as opposed to a single peak for

HCCI combustion. The first peak is considered to be from the flame front and the second peak to be from post flame ionisation (138). It has been described that most timing information is given by the location of the start of the signals first peak [12]. However, as found with the HCCI ionisation investigation in chapter 4, the start location of ion current signal can be unreliable for feedback due to noise. Therefore, taking a location along the rising edge of the first peak offers a location less effected by noise and can be more accurately averaged. The 75% location along the rising edge was found to be accurate for the HCCI investigation and was used again as a starting point for SI ion current signal feedback. This location may not be optimal for SI engine timing feedback, but the evidence suggests it can offer the relevant information. The optimal ionisation feedback location for SI engine could be of interest and investigated in future work.

For every Simulink time step, a control signal is passed back to the Labview program in the form of desired spark ignition timing in CAD. In Labview it is possible to create a digital output based on an on-chip counter. For the digital output for spark timing, the counter was configured to count crank degree markers. By triggering the digital output task from TDC it was also possible to synchronise the signal to a fixed location within the combustion cycle.
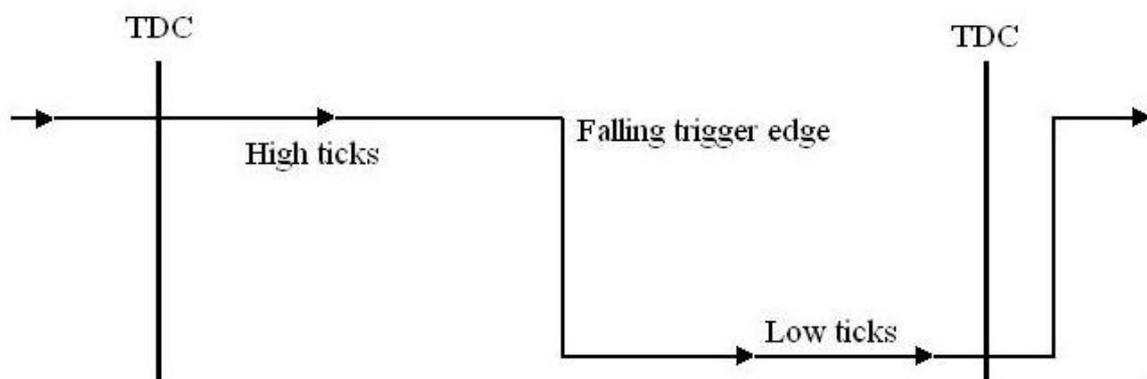
Figure 8.5 Example of Labview digital output for falling edge trigger

Figure [8.5], shows an example of the digital output control signal from Labview. Through the digital output task it is possible to configure the output signal to be high, or low for a number of counter ticks. To activate the ignition coil drive unit, only a falling signal is needed to be presented for each combustion cycle. The location of the falling edge can be controlled through adjusting the number of counter ticks to maintain high and low voltage. The Labview program calculates the number of high and low counter ticks depending on the desired ignition timing control signal from Simulink. Applying this method allows for a control signal

for each cycle to be phased with the engine and synchronised to TDC. Although the control signal will be output for each cycle, the Simulink model runs within its own simulation clock. The data will be passed between the two programs on the completion of each time step of the Simulink simulation. When the data is passed between the programs the values are stored within hold blocks until new data becomes available. Therefore, if the Simulink control becomes idle while the program calculates ANN adaption, the output control signal will not be interrupted.

8.3 ENGINE SPEED CHANGE CALIBRATION TEST RESULTS
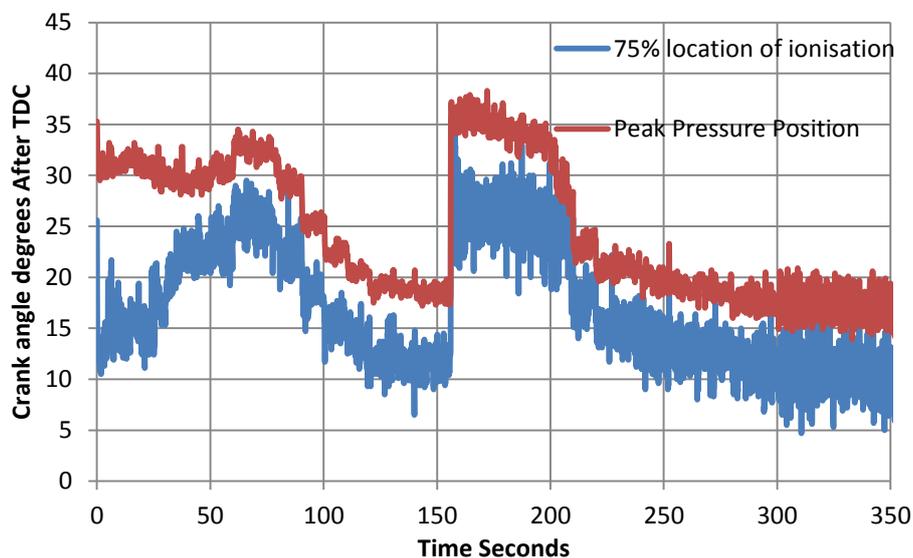
Figure 8.6 Feedback signals for 75% of peak ionisation current signal location and peak pressure position during change in speed test for E6 engine

Figure [8.6] shows the 75 % of peak ionisation current signal location and peak cylinder pressure position feedback signals during the change in engine speed test. The ionisation feedback signal has been successfully combined with the developed control architecture. Through using the ionisation feedback signal the control architecture has been able to control to the desired set point combustion timing. The control architecture has also been able to adapt online for the real engine dynamics and noisy feedback signal for both engine speed conditions.
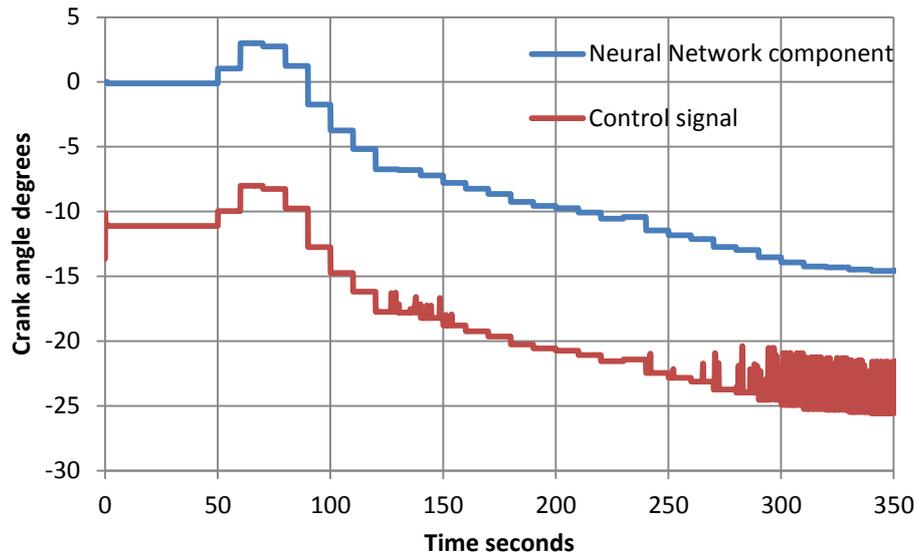
Figure 8.7 Neural network component and overall control for change in speed test for E6 engine

Figure [8.7] show the feed forward ANN control component and overall control component from the control system during the change in engine speed tests. The feed forward ANN adaption compensates for the FLC model/system mismatch at both speed positions. The inverse model ANNs within the control architecture are able to learn the system dynamics form the noisy variable feedback signal. As the model/system mismatch is reduced the FLC feedback component becomes unsaturated. This can be seen evident at 140-150 seconds, then again at 290 seconds onwards. From figure [8.7] the initial two feed forward ANN adaptations are incorrect. All the models in the architecture, FLC and ANN, are based on the simulation engine dynamics and pressure trace as feedback. This has allowed for a rough starting point for the control system. It must adapt to obtain more optimal control. As the models are initially incorrect it could potentially explain why the initial adaptations are incorrect. It is also difficult for the control to predict a direction if the online training data is only for a single control signal output condition, time 0-50 seconds on figure [8.7]. Once incorrect adaption has taken place, the training data will be based on several control conditions, 20-70 seconds on figure [8.7]. As there have been incorrect conditions the error has increased allowing the control to find a gradient to train. It is difficult to predict a direction if only a single point of reference is available.

Overall the change in speed test has provided evidence the adaptive control architecture can operate with the experimental ionisation feedback signal for combustion phasing control at a steady state condition. The ionisation current signal was also able to provide information on combustion phasing to allow the adaptive control architecture to adapt the control model to calibrate to the two separate speeds and throttle operating conditions.

8.4 COMPRESSION RATIO CHANGE ADAPTION TEST RESULTS
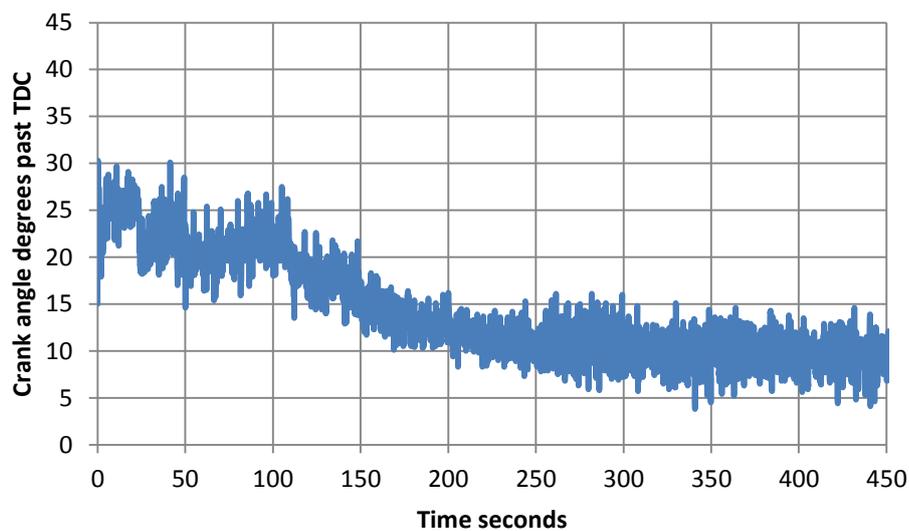
Figure 8.8 Feedback signal for 75% of peak ionisation current signal location during CR change test for E6 engine

At the steady state position of 50% throttle and engine speed 1500 RPM the control architecture has been able to adapt to the two immeasurable dynamic changes in CR and maintained combustion timing control, figure [8.8]. The changes in CR had only small effect on the combustion timing control when location of 75% peak ionisation current signal is used for feedback.
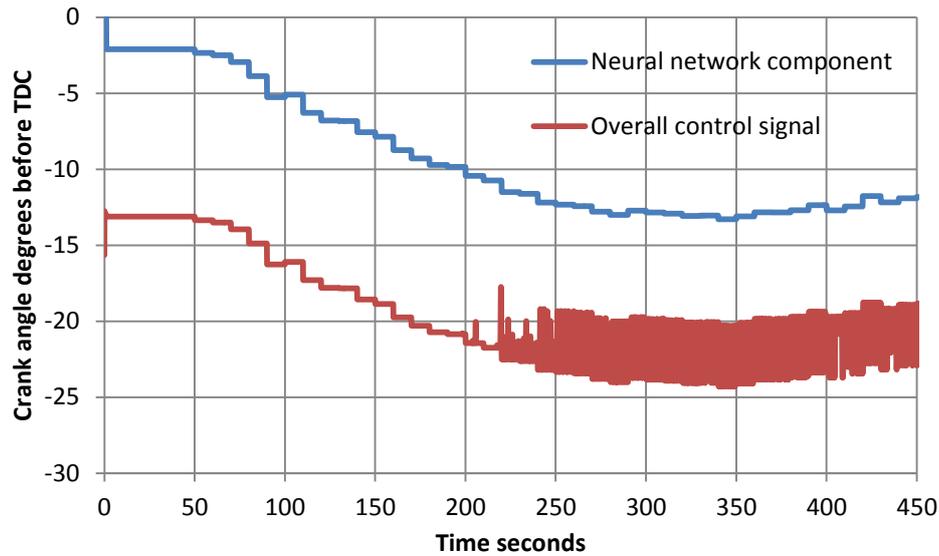
Figure 8.9 Neural network component and overall control for change in compression ratio test for E6 engine

The combustion timing is maintained through the CR changes while the ANNs adaption improved the control models accuracy for both advanced and retarded timing conditions, figure [8.9]. The ionisation feedback signal is directly related to the combustion reaction products, and as the combustion is initiated through spark ignition the initial ignition timing and location is relatively constant. As the CR changes the cylinder charge temperature and pressure will be affected. Igniting a mixture at a higher CR will lead to a faster burn rate and a more efficient combustion of the charge. Although the rate of combustion will change, the product of the reaction will be present in the cylinder as soon as the reactions begin. Within the range of CR used in the test, there would be only small changes in the timing difference for 75% of peak free ions in the cylinder.
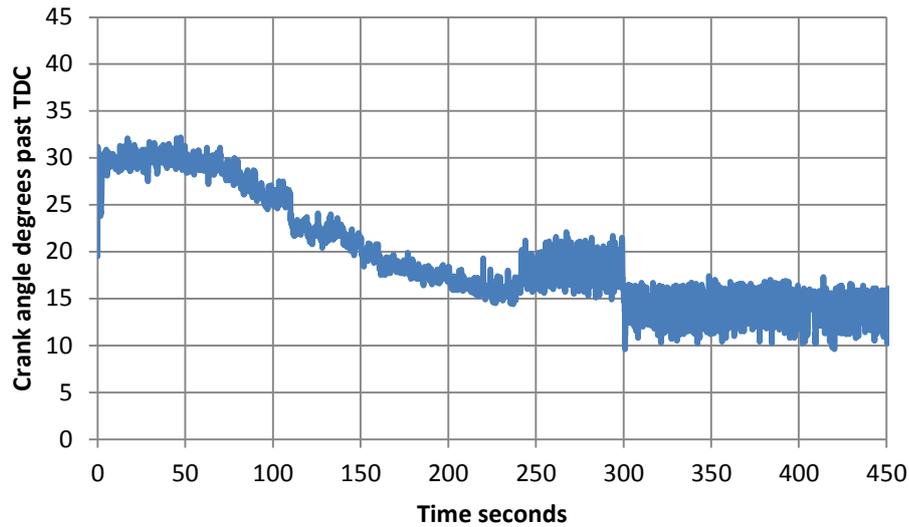
Figure 8.10 Feedback signal for peak pressure position during CR change test for E6 engine

However, the peak cylinder pressure position is much more effected by the CR change, figure [8.10]. In the first 200 seconds, there is a gradual decrease in timing of peak cylinder pressure after TDC. This time period is at a constant CR of 6, the gradual decrease is due to the ANN adapting and controlling 75% of peak ionisation current signal location to 10 degrees after TDC. During the CR changes at 240 and 300 seconds, clear step can be seen in the timing of peak cylinder pressure position. The peak cylinder pressure position is a product of the temperature and pressure history of the cylinder. At the different CRs the combustion efficiency and duration will alter.
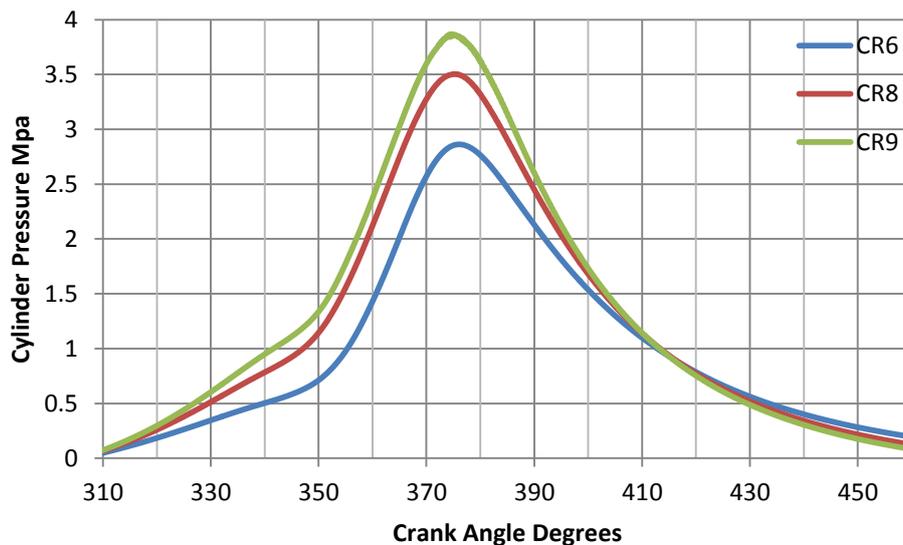
Figure 8.11 Typical 30 cycle averaged cylinder pressure (Mpa) for Ricardo E6 SI engine at Compression Ratio 6, 8 and 9 during experiment with fixed fuel injection pulse width.

Combustion occurring under a higher CR will have a compound effect on peak cylinder pressure position. At the higher cylinder pressure rate and efficiency of combustion will increase causing increased heat release. The additional heat release causes a rise in peak cylinder pressure. This will cause peak cylinder pressure position to be closer to TDC for constant combustion timing. Figure [8.11] provides typical 30 cycle averaged cylinder pressure traces for the Ricardo E6 engine at the three CRs. As the CR increases the peak cylinder pressure rises and the timing becomes closer to TDC as expected. If the peak cylinder pressure position timing is desired instead of start of combustion timing then more feedback information would be required from the ionisation current signal. For example information on the combustion duration or rate would be beneficial for peak cylinder pressure position timing.

These results show the adaptive control architecture is able to control ignition timing for the E6 single cylinder research engine, to a desired set point at steady state conditions using ionisation as a feedback signal. The adaptive control architecture is able to adapt online for calibration of the control to speed and load conditions as well as immeasurable changes in the dynamics of the system caused through the CR changes. Using ionisation current signal as feedback can potentially offer more accurate control over start of combustion timing then feedback from cylinder pressure. However, if combustion timing is not the primary interest and phasing peak cylinder pressure is, then it would be beneficial to have more information from ionisation on rate or duration of combustion.

9. Conclusions and Future Development

The goal of this work has been to investigate methods to simplify the control of advanced combustion strategies such as downsize SI, GDI and HCCI. The advanced combustion strategies used today have several inherent control problems due to their significant increase in complexity.

All advanced combustion control strategies require an increase in number of actuators. The control designs ideally need to be MIMO with their descriptive control models capturing the dynamics and nonlinearities of the system. Using traditional map based control is limited to two dimensions and often requires vast amounts of work to determine optimal setting for specific engine applications.

Many of the useful internal state variables are not measured due to the difficulties in justifying the cost of additional sensors. Current feedback systems are generally of indirect states throughout the engine system such as manifold air pressure or exhaust lambda. Some of these feedback systems can have long time delays causing further complication with control feedback.

In addition, some of the advanced combustion strategies and their sub-systems can be sensitive to uncontrollable immeasurable influences. This can be in the form of aging, environmental effects, manufacturing tolerances and in cylinder or cooling system deposits. If these immeasurable influences can alter the systems dynamics with time then any fixed control system will become mismatched to the systems true dynamics. A feedback control strategy can add a level of robustness against some level of mismatch, but if significant changes occur the feedback control can become saturated.

This work has investigated methods for reducing the modelling requirement for nonlinear MIMO control. Methods for in-cylinder feedback using cost effective components and methods for combining MIMO control with experimental feedback in an architecture that can allow for online adaption. Allowing the MIMO control to adapt on-line can aid in self-calibration of the control model and to adapt the control model to changes in the engine dynamics with time.

The result of this work is a universal ionisation sensor that can be used on a range of engines, and the development of the initial foundations for adaptive control architecture for MIMO

automotive engine control. The combination of the two systems offers tools that can be used to further investigate adaptive control for advanced combustion strategies.

9.1 UNIVERSAL MOBILE IONISATION SYSTEM

The developed universal mobile ionisation system allows for feedback of ions from a dedicated probe or the firing spark plug of the engine. The system was designed with variable sensor voltage so that it maybe tuned and applied to different engines. The system can also work alongside existing spark ignition control systems with falling or raising trigger signals for control of the integrated coil for ignition timing.

The ionisation current signal from the mobile ionisation system was evaluated for feedback on combustion characteristics on the Lotus Omnivore two stroke HCCI engine. Ionisation current and the rate of ionisation current signals both showed relationships with combustion timing. Feedback from the 75% location of peak ionisation current signal had the best correlation with the location of peak cylinder pressure position. With the strong relationship with peak cylinder pressure position the ionisation feedback shows good potential for combustion timing control.

There was also correlation between the duration of ionisation current signal and the duration of the MFB. As the MFB increases and decreases with cycle-to–cycle variation, the ionisation current signal follows the trend. This relationship could potentially offer feedback between ionisation duration and combustion duration.

The rate of ionisation current signal had shown strong spikes within the peak of the signal, this represents rapid changes in the rate of ionisation. This characteristic maybe linked with combustion occurring at multiple sites of the cylinder, typical of HCCI combustion. This relationship could give the ability to detect knock, and auto ignition combustion. The peak value of the rate of ionisation current signal tends to show an inverse relationship with timing location of peak pressure. This shows a potential relationship with combustion duration.

During operation with leaner AFR the magnitude of ionisation current signal decreased as well as duration. This relationship could offer a potential in-cylinder feedback for AFR control. However, consideration would be needed as the peak magnitude of ionisation current signal is also affected through density of the charge near the sensor.

Overall the mobile ionisation feedback system has been validated for feedback information on relationships with; combustion timing, rate of combustion, knock, combustion duration and AFR for a two stroke HCCI engine. The mobile ionisation system is able to offer a wealth of feedback information and can be used in further work as a tool for investigating advanced combustion control.

9.2 CONTROL ARCHITECTURE

The developed adaptive control architecture uses a combination of fuzzy logic and ANN techniques to offer a control structure that can reduce the effort of modelling the nonlinear dynamics of IC engines.

The FLC offers an alternative method for nonlinear MIMO control. The control is based on linguistic rules created through expert knowledge of the system. This allows expert knowledge about the interactive behaviour of engine dynamics to be integrated into a control structure. The choice of rules and MF allow for an intuitive method for modelling non-linear dynamics. By using increase number of MFs in regions of high non linearity allows for greater precision of modelling, while less MF can be used in more linear areas. This allows for accurate nonlinear modelling while keeping the complexity low. All the MFs used in FLC are mathematically simple. The combination of all the MFs allows a mathematically simple nonlinear control model.

The ANNs within the control structure are also built up of many mathematically simple functions. Through the training of the weighting of these functions it is possible to complement the fuzzy control model with an adaptive component.

To the Author's knowledge there is no current method for the offline calibration and training of matlab based ANN that allow an optimum choice of MF and size of network to be chosen. Often the choice of MFs and net size is a process of trial and error. In this work a logical approach was adopted that was based around the idea of growing the ANNs. A simple code was written to increase the size of the ANN until further increase offered little improvement. This method will not guarantee an optimal solution but it did offer a logical approach to the creation of ANNs.

The ANNs implementation with the control architecture allows online calibrate of the control model to reduce model/system mismatch and to allow the adaption to systems that have

dynamics that change with time. By combining the fuzzy logic controller and the ANNs in an architecture based on the work of Yuan et al (77), it has allowed for a structure that can continuously adapt online without undesirable interactions occurring between the fixed controller and the ANNs during updates. The fixed fuzzy controller and the ANNs work together to model and control the engine. By using the architecture proposed by Yuan et al, and combining with the FLC it has extended the original control structure from MISO control for robot arm control to nonlinear MIMO control for engine control.

The adaptive control architecture was developed using Matlab, Simulink and a GT-power SI engine simulation. The GT-power engine simulation offered a low risk method for testing the control architecture's ability to adapt to immeasurable time varying dynamics. By altering the CR and fuel injector flow rates during simulation it was possible to simulate a theoretical engine that's non-linear control dynamics change with time.

The results from the GT-power model validated that the developed control architecture was able to maintain stable control to the desired set points for both MISO and MIMO control for both steady state and transient conditions. When the dynamic shifts were introduced, the control architecture was capable of continuous online adaption without causing instability between the control architecture components during the ANN updates for both steady state and transient drive cycles. The architecture design allows the fuzzy logic and ANNs to work harmoniously to reduce the control error.

9.3 ADAPTIVE CONTROL WITH IONISATION FEEDBACK

The adaptive control architecture and the mobile ionisation feedback system were combined through implementation with the single cylinder Ricardo E6 research engine. The practical implementation allowed for the validation of the control architecture for control with ionisation current signal as feedback and for control of a real engine.

The ionisation current signal was seen to be less sensitive to CR change then peak pressure position. Changing CR has a compound effect on cylinder pressure through increased pressure and temperature from geometric compression and increased heat release and heat release rate from improved combustion efficiency. Ionisation current signal is proportional to the free ions from the reactions of combustion. Using ionisation current signal can offer an accurate feedback for combustion timing. However, if phasing of peak pressure position were of primary interest over combustion phasing, additional feedback information from

ionisation such as rate or duration of combustion would be required. A future investigation using feedback from 75% ionisation current signal location and peak value of rate of ionisation current signal would be beneficial to investigate the potential of ionisation for phasing peak cylinder pressure timing.

The control architecture was able to maintain control to the desired set point at steady state condition using the 75% location of peak ionisation current signal as feedback. The ionisation feedback signal from the single cylinder engine had significant variation. However, the control architecture was still able to use this feedback information to model the inverse dynamics of the engine at the three different CRs. Therefore, the control architecture with the ionisation feedback was able to adapt and self-calibrate online for the steady state speed and throttle positions during immeasurable time varying dynamics. These results validate that the mobile ionisation sensor can offer a feedback signal for control. The ionisation feedback signal can offer a reliable method for maintaining combustion timing and when combined with the developed adaptive control architecture, can offer online adaption during immeasurable changes in the engines dynamics.

9.4 FUTURE DEVELOPMENT

Matlab and Simulink offer a programming platform that has numerous pre-programmed functions and facilitates the fast development of control architecture through the high level programming language. The Matlab and Simulink package is also recognized by many other software and hardware companies as a leading software solution for developing simulation and control systems. As a result, these third party companies offer methods to link their products with Simulink to reduce development time and improve quality of work. The development and validation of the adaptive control architecture was achievable through the Matlab Simulink link with third party software, GT-power for engine simulation. For the practical validation with ionisation feedback, Simulink was linked with third party software National Instruments Labview. Although Matlab and Simulink has offered an excellent platform for the initial development, future work will require real time applications. The Simulink connection to National instruments Labview allows an interface between Simulink and the engine that can continually output a given control signals until a new control is available. This allows the control program to run in Simulink under its own simulation clock, therefore not real time. Although this method has facilitated the initial validation of the

controller with an engine using ionization feedback, it will be insufficient for more complex test such as MIMO transient cycles.

The adaptive control architecture has been validated for control and adaption to time varying dynamics for MISO and MIMO control through engine simulation for steady state and transient drive cycle. It has also been validated for control and adaption to time varying dynamics for MISO control for a real engine with ionisation feedback for steady state conditions. This has allowed for the initial development of an adaptive control architecture that can work with ionisation as feedback. This developed adaptive control architecture can be used as a foundation control tool for future development. The basics aspects and theory of the controller and feedback method have been validated. Future work should extend the applications of the control by using it as a tool that can be applied to different systems. It would be useful to extend the work to investigate the adaptive control with ionisation feedback for the control of the advanced combustion strategies GDI and HCCI.

The AFR control during lean burn GDI combustion can be complex due to the averaged feedback signal and significant time delay. With advanced combustion modes there are often an increase in key combustion technologies. This leads to an increase in control variables that can have linked nonlinear behaviour. Accurate modelling and calibration of the resulting MIMO control systems is an expensive time consuming task. The ionisation feedback can potentially offer information on AFR. To the author's knowledge this has not been investigated for GDI lean burn AFR control. During the lean burn mode the ionisation feedback could potentially be used for phasing the combustion and controlling the AFR in the rich region surrounding the firing plug. If the ionisation can be used, then the developed adaptive control from this work can be applied to simplify the modelling and calibration requirement for the MIMO advanced combustion strategy.

HCCI combustion has no direct ignition trigger and relies on a number of key combustion technologies to simultaneously maintain combustion timing, rate and duration. The MIMO control for this combustion is complicated further due to its linked behaviour with cycle history and nonlinear dynamics that can cause a number of immeasurable influences that can change with time. The ionisation feedback is capable of offering combustion timing, combustion duration and rate of combustion information for two stroke HCCI. The developed adaptive control architecture from this work can be applied with the ionisation feedback for the control of HCCI combustion. The adaptive control architecture can be used to reduce the modelling requirement. The HCCI MIMO nonlinear dynamics are complicated to model accurately. Once they have been modelled the systems dynamics can change with time

causing large inaccuracies for what could have been an ideal model. Using the fuzzy logic method allows the nonlinear dynamics to be modelled through expert knowledge of the system. This allows for an intuitive control system to be designed. Any error between the developed fuzzy model and the engines dynamics can be calibrated through the online adaption. If the HCCI combustion dynamics do change with time, then the architecture can adapt to maintain model accuracy.

To allow the developed control architecture to be applied to these advanced combustion systems it ideally needs to be converted for real time application for MIMO control. As the control has been developed using Simulink it is possible to link with software and hardware from a company dSPACE. dSPACE offers third party products and services for high performance rapid control prototyping. The dSPACE prototype system allows for optimization of the real controlled system without manual programming. Any modifications needed to the control design, such as building the fuzzy logic rule base, can be carried out on the spot. The Simulink control design is automatically implemented on the dSPACE hardware via code generated by real time workshop. dSPACE also offer a wide range of input/output interfaces to connect the control design to the real MIMO engine systems.

REFERENCES

1. J. C. Paul. Emissions and Efficiency Targets: Advanced Technology Reciprocating Internal Combustion Engines. California Advanced Reciprocating Internal Combustion engines Collaborative (ARICE) Workshop ed. Ricardo, Inc, 10 July 2001.

2. J. B. Heywood. Internal Combustion Engine Fundamentals. McGraw-Hill Book company, 1988. ISBN 0071004998.

3. J. J. J. Louis. Well-to-Wheel Energy use and Greenhouse Gas Emissions for various Vehicle Technologies. S.A.E Techical Paper Series, 2001, vol. SP-1589. no. 2001-01-1343.

4. M. Fujiwara, et al. Development of a 6-Cylinder Gasoline Engine with New Variable Cylinder Management Technology. S.A.E Techical Paper Series, 2008, vol. SP-2180. no. 2008-01-0610.

5. G. Lumsden, et al. Development of a Turbocharged Direct Injection Downsizing Demonstrator Engine. S.A.E Int. J. Engines. 2009-01-1503 ed., 2008, vol. 2, no. 1. pp. 1420-1432.

6. C. Guillaume, et al. Neural Control of Fast Nonlinear Systems-Application to a Turbocharged SI Engine with VCT. IEEE Transactions on Neural Networks. 1045-9227 ed., 2007, vol. 18, no. 4. pp. 1101-1114.

7. J. E. Kirwan, et al. 3-Cylinder Turbocharged Gasoline Direct Injection: A High Value Solution for Low CO2 and NOx Emissions. S.A.E Int. J. Engines. 2010-01-0590 ed., 2010, vol. 3, no. 1. pp. 355-371.

8. L. Wang et al. On-Line Identification and Adaption of LNT Models for Improved Emission Control in Lean Burn Automotive Engines. American Control Conference. 0-7803-5519-9 ed., 2000. pp. 1006-1010.

9. F. Zhang, et al. Linear Parameter-Varying Lean Burn Air-Fuel Ratio Control. IEEE Conference on Decision and Control. 0-7803-9568-9 ed., 2005. pp. 2688-2693.

10. J. Sun, et al. Modeling and Control of Gasoline Direct Injection Stratified Charge (DISC) Engines. IEEE International Conference on Control Applications. 0-7803-5446-X/99 ed., 1999. pp. 471-477.

11. K. Epping, et al. The Potential of HCCI Combustion for High Efficiency and Low Emissions. S.A.E.Transactions., 2002, vol. SP-1721. no. 2002-01-1923.

12. D. Panousakis. Ion current sensing for controlled auto ignition in internal combustion enignes, PhD Thesis of Loughborough University, Department of Aeronautical and Automotive Engineering, May 2009.

13. D. Yap, et al. Natural Gas HCCI Engine Operation with Exhaust Gas Fuel Reforming. Science Direct International Journal of Hydrogen Energy, 25 July 2005, 2006, vol. 31. pp. 587-595.

14. T. Ohmura, et al. A Study on Combustion Control by using Internal and External EGR for HCCI Engines Fuelled with DME. S.A.E.Transactions., 2006, no. 2006-32-0045.

15. A. Tsolakis, et al. Application of Exhaust Gas Fuel Reforming in Diesel and Homogeneous Charge Compression Ignition (HCCI) Engines Fuelled with Biofuels. Science Direct Energy, 14 June 2007, 2008, vol. 33. pp. 462-470.

16. D. Kawano, et al. Ignition and Combustion Control of Diesel HCCI. S.A.E.Transactions., 2005, no. 2005-01-2132.

17. L. Koopmans, et al. A Four Stroke Camless Engine, Operated in Homogeneous Charge Compression Ignition Mode with Commercial Gasoline. S.A.E.Transactions., 2001, vol. SP-1642. no. 2001-01-3610.

18. R. Chen, et al. A Review of Experimental and Simulation Studies on Controlled Auto Ignition Combustion. S.A.E Technical Paper Series, 2001, no. 2001-01-1890.

19. A. Bhave, et al. Evaluating the EGR AFR Operating Range of a HCCI Engine. 2005-01-0161 S.A.E Technical Paper Series, 2005.

20. P. Strandh, et al. Variable Valve Actuation for Timing Control of a Homogeneous Charge Compression Ignition Engine. S.A.E Technical Paper Series,2005, no. 2005-01-0147.

21. M. Christensen, et al. Supercharged Homogenous Charge Compression Ignition (HCCI) with Exhaust Gas Recirculation and Pilot Fuel. S.A.E Technical Paper Series. 2000. no. 2000-01-1835.

22. J. M. Grenda. Numerical Modeling of Charge Stratification for the Combustion Control of HCCI Engines. S.A.E Technical Paper Series. 2005. no. 2005-01-3722.

23. J.W.G Turner, et al. Project Omnivore: A Variable Compression Ratio 2-Stroke Engine for Ultra-Wide-Range HCCI Operation on a Variety of Fuels. SAE 2010 World Congress. Detroit, Michigan, USA , April, 2010, no. 2010-01-1249.

24. M Lida, et al. The Effects of Intake Air Temperature, Compression Ratio and Coolant Temperature on the Start of Heat Release in an HCCI (Homogeneous Charge Compression Ignition) Engine. S.A.E Technical Paper Series. 2001. no.2001-01-0418.

25. Z. Chen, et al. How to Put the HCCI Engine to Practical use: Control the Ignition Timing by Compression Ratio and Increase the Power. S.A.E Technical Paper Series. 2003. no. 2003-01-1832.

26. L. Koopmans, et al. A Four Stroke Camless Engine, Operated in Homogenous Charge Compression Ignition Mode with Commercial Gasoline. S.A.E Technical Paper Series.2001.no. 2001-01-3610.

27. N. Milovanovic, et al. Influence of the Variable Valve Timing Strategy on the Control of a Homogeneous Charge Compression Ignition (HCCI) Engine. S.A.E Technical Paper Series.2004.no. 2004-01-1899.

28. G. Kontarakis, et al. Demonstrating of HCCI using a Single-Cylinder, Four-Stroke SI Engine with Modified Valve Timing. S.A.E Technical Paper Series. 2000 .no. 2000-01-2870.

29. B. Thirouard, et al. Investigation of Mixture Quality Effects on CAI Combustion. S.A.E Technical Paper Series.2005. no. 2005-01-0141.

30. C. D. Marriott, et al. Experimental Investigation of Direct Injection Gasoline for Premixed Compression Ignition Combustion Phasing Control. S.A.E Technical Paper Series.2002 .no. 2002-01-0418.

31. K. Ingaki, et al. Duel Fuel PCI Combustion Controlled by in Cylinder Stratification of Ignitability. S.A.E Technical Paper Series.2006 .no. 2006-01-0028.

32. M. Sjoberg, et al. An Investigation of the Relationship between Measured Intake Temperature, BDC Temperature and Combustion Phasing for Premixed and DI HCCI Engines. S.A.E Technical Paper Series.2004.no.2004-01-1900.

33. M. Sjoberg, et al. Comparing Enhanced Natural Thermal Stratification Against Retarded Combustion Phasing for Smoothing HCCI Heat Release Rates. S.A.E Technical Paper Series.2004. no. 2004-01-2994.

34. S. Zhong, et al. Experimental Investigation into HCCI Combustion using Gasoline and Diesel Blended Fuel. S.A.E Technical Paper Series. 2005. no.2005-01-3733.

35. J. Yang et al. Robustness and Performance Near the Boundary of HCCI Operation Regime of a Single Cylinder OKP Engine. S.A.E Technical Paper Series. 2006. no. 2006-01-1082.

36. G. Shibata et al. The Interaction between Fuel Chemical and HCCI Combustion Characteristics Under Heated Intake Air Conditions. S.A.E Technical Paper Series.2006. no. 2006-01-0207.

37. M. Christensen, et al. Influence of Mixture Quality on Homogeneous Charge Compression Ignition. S.A.E Technical Paper Series. 1998. no. 982454

38. F. Zhao, et al. Homogenous Charge Compression Ignition (HCCI) Engines Key Research and Development Issues. Society of Automotive Engineers Inc, March 2003, 2003. ISBN 0-7680-1123-x.

39. H. Santoso, et al. Managing SI/HCCI Duel Mode Operation. S.A.E Technical Paper Series. 2005. no.2005-01-0162.

40. J. Franz, et al. Closed Loop Control of an HCCI Multi-Cylinder Engine and Corresponding Adaptation Strategies . S.A.E Technical Paper Series, 2009, no. 2009-24-0079.

41. O. A. Guralp, et al. Characterizing the Effect of Combustion Chamber Deposits on a Gasoline HCCI Engine. S.A.E Technical Paper Series, 2006, no. 2006-01-3277.

42. D. L. Flowers, et al. Operation of a Four Cylinder 1.9 L Propane Fueled Homogeneous Charge Compression Ignition Engine: Basic Operating Characteristics and Cylinder-to-Cylinder Effects . S.A.E Technical Paper Series, 2001, no. 2001-01-1895.

43. J. Misztal, et al. Cylinder-to-Cylinder Variations in a V6 Gasoline Direct Injection HCCI Engine. Journal of Engineering for Gas Turbines and Power, 2009, vol. 131, no. 042801.

44. M. Lida, et al. The Effect of Intake Air Temperature, Compression Ratio and Collent Temperature on the Start of Heat Release in an HCCI (Homogeneous Charge Compression Ignition) Engine. S.A.E Technical Paper Series, 2001, no. 2001-01-1880.

45. J. Hyvonen, et al. Balanceing Cylinder to Cylinder Variation in a Multi Cylinder VCR HCCI Engine. S.A.E Technical Paper Series, 2004, vol. SP-1884. no. 2004-01-1897.

46. M. Gafvert, et al. Control of GDI Engines using Torque Feedback Exemplified by Simulations. Science Direct Control Engineering Practice. 0967-0661/03 ed., 2004, vol. 12. pp. 165-180.

47. J. Olsson, et al. Closed-Loop Control of an HCCI Engine. S.A.E Technical Paper Series, 2001, vol. SP-1623. no. 2001-01-1031.

48. G. Haraldsson, et al. HCCI Combustion Phasing with Closed-Loop Combustion Control using Variable Compression Ratio in a Multi Cylinder Engine. S.A.E Technical Paper Series, 2003, no. 2003-01-1830.

49. G. Haraldsson, et al. HCCI Closed-Loop Combustion Cotnrol using Fast Thermal Managment. S.A.E Technical Paper Series, 2004, vol. SP-1819. no. 2004-01-0943.

50. N. Zhou, et al. Study on Layered Close Loop Control of 4-Stroke Gasoline HCCI Engine Equipped with 4VVAS. S.A.E Technical Paper Series, 2008, vol. SP-2182. no. 2008-01-0791.

51. F. Agrell, et al. Integrated Simulation and Engine Test of Closed Loop HCCI Control by Aid of Variable Valve Timings. S.A.E Technical Paper Series, 2003, vol. SP-1742, no. 2003-01-0748.

52. F. Agrell, et al. Control of HCCI during Engine Transients by Aid of Variable Valve Timings through the use of Model Based Non-Linear Compensation. S.A.E Technical Paper Series, 2005, vol. SP-1963, no. 2003-01-0131.

53. A. Kwiatkowski, et al. Application of LPV Gain Scheduling to Charge Control of a SI Engine. IEEE Conference on Computer Aided Control Systems Design. 0-7803-9797-5 ed., 2006. pp. 2427-2331.

54. M. Hillion, et al. Active Combustion Control of Diesel HCCI Engine: Combsution Timing. S.A.E Technical Paper Series, 2008, vol. SP-2159, no. 2008-01-0984.

55. O. Grondin, et al. Torque-Oriented Control of a Homogeneous Charge Compression Ignition Vehicle. IEEE Conference on Decision and Control, 2008, vol. ThC16.1, no. 978-1-4244-3124-3. pp. 5608-5614.

56. R. Pfeiffer, et al. System Identification and LQG Control of Variable-Compression HCCI Engine Dynamics. IEEE International Conference on Control Applications, 2004, no. 0-7803-8633-7/04/. pp. 1442-1447.

57. G. Haraldsson, et al. Transient Control of a Multi Cylinder HCCI Engine during a Drive Cycle. S.A.E Technical Paper Series, 2005, vol. SP-1963, no. 2005-01-0153.

58. G. M. Shaver, et al. Cycle to Cycle Control of HCCI Engines. Proceedings of IMECE 03 ASME International Mechanical Engineering Congress and Exposition, 2003, vol. IMECE2003-41966.

59. G. M. Shaver, et al. Physics-Based Closed-Loop Control of Phasing, Peak Pressure and Work Output in HCCI Engine Utilizing Variable Valve Action. IEEE Proceeding of the 2004 American Control Conference, 2004, vol. 0-7803-8335-4/04/.

60. D. Blom, et al. HCCI Engine Modeling and Control using Conservation Principles. S.A.E Technical Paper Series, 2008, vol. SP-2182, no. 2008-01-0789.

61. J. R. Popp, et al. Model-Based Feed-Forward Control of Diesel HCCI Engine Transients. S.A.E Technical Paper Series, 2009, no. 2009-01-1133.

62. G. M. Shaver, et al. A Physics-Based Approach to the Control of Homogeneous Charge Compression Ignition Engines with Variable Valve Actuation. International Journal of Engine Research, 2005, vol. 6 number 4, no. 10.1243/146808705X30512. pp. 361-375.

63. J. Kang, et al. Concept and Implementation of a Robust HCCI Engine Controller . S.A.E Technical Paper Series, 2009, no. 2009-01-1131.

64. J. M. Maciejowski. Predictive Control with Constraints. Pearson Education, 2002. ISBN 0201398230.

65. A. Dutka, et al. Model Based Nonlinear Multivariable Engine Control. American Control Conference. 1-4244-0989-6 ed., 2007. pp. 3671-3677.

66. P. Strandh, et al. Variable Valve Actuation for Timing Control of a Homogenous Charge Compression Ignition Engine. S.A.E Techical Paper Series, 2005, vol. SP-1963, no. 2005-01-0147.

67. W. H. Kwon, et al. Stabilizing State Feedback Design Via the Moving Horizon Method. International Journal of Control, vol. 37. pp. 631-643.

68. S. Karra, et al. Adaptive Model Predictive Control of Multivariable Time Varying Systems. Ind. Eng. Chem. Res, 2008, vol. 47. pp. 2708-2720.

69. S. A. Assadi. Neural Network Based Model Reference Adaptive Control for Electronic Throttle Systems. S.A.E Technical Paper Series, 2007, vol. SP-2111, no. 2007-01-1628.

70. K. M. Hornik, et al. Multilayer Feedforward Networks are Universal Approximators. Neural Networks, 1989, vol. 2, no. 5. pp. 359-366.

71. I. Arsie, et al. Nonlinear Recurrent Neural Networks for Air Fuel Ratio Control in SI Engines. S.A.E Technical Paper Series, 2004, vol. SP-1822, no. 2004-01-1364.

72. M. Mirhassani, et al. On Control of HCCI Combustion Neural Network Approach. Proceedings of the 2006 IEEE International Conference on Control Applications, 2006, vol. 0-7803-9796-7/06/. pp. 1669-1674.

73. T. Wiens, et al. A Survey of Literature Applicable to On-line Learning for Transient Control of Fuel/Air Ratio on SI Engines Powered by Gaseous fuels. Dept.of Mechanical Engineering University of Saskatchewan Saskatoon. Proceedings of CSME Forum 2006, Kananaskis, 2006.

74. W. T. Miller III. Sensor-Based Control of Robotic Manipulators using a General Learning Algorithm. IEEE Journal of Robotics and Automation, 1987, vol. RA-3, no. No 2. pp. 157-165.

75. W. T. Miller III, et al. Real Time Dynamic Control of an Industrial Manipulator using a Neural Network Based Learning Controller. IEEE Transactions on Robotics and Automation, 1990, vol. 6, no. 1. pp. 1-9.

76. W. T. Miller et al. CMAC: An Associative Neural Network Alternative to Back propagation. Proceedings of the IEEE, 1990, vol. 10. pp. 1561-1567.

77. Y. Yuan, et al. Advances in Neural Networks - ISNN 2004Heidelberg: Springer Berlin, 2004. Online Learning CMAC Neural Network Control Scheme for Nonlinear Systems, pp. 37-50. ISBN 978-3-540-22843-1.

78. H. Shiraishi, et al. CMAC Neural Network Controller for Fuel Injection Systems. IEEE Transactions on Control Systems Technology, 1995, vol. 3.

79. M. Majors, et al. Neural Network Control of Automotive Fuel-Injection Systems. IEEE International Symposium of Intelligent Control, 1993.

80. S. Ganesan, et al. An Idle Speed Controller using Analystically Developed Fuzzy Logic Control Law. S.A.E Technical Paper Series, 2002, no. 2002-01-0138.

81. A. H. Shamekhi, et al. Fuzzy Control of Spark Advance by Ion Current Sensing. S.A.E Technical Paper Series, 2005, vol. SP-1975, no. 2005-01-0079.

82. C. H. Wang, et al. Self Learning FNN (SLFNN) with Optimal on Line Tuning for Water Injection Control in a Turbo Charged Automobile. IEEE, 2005. pp. 878-882.

83. J. Bengtsson, et al. Closed-Loop Combustion Control of Homogeneous Charge Compression Ignition (HCCI) Engine Dynamics. International Journal of Adaptive Control and Signal Processing, 2004, vol. 18, no. 2. pp. 167-79. ISSN 08906327; 08906327.

84. D. Panousakis, et al. Using Ion Current Sensing to Interpret Gasoline HCCI Combustion Processes. S.A.E Technical Paper Series, 2006, vol. SP-2005, no. 2006-01-0024.

85. D. Panousakis, et al. Ion Current Signal Interpretation Via Artificial Neural Network for Gasoline HCCI Control. S.A.E Technical Paper Series, 2006, vol. SP-2005, no. 2006-01-1088.

86. T. Tanaka, et al. Ion Current Measurement in a Homogeneous Charge Compression Ignition Engine. ImechE Int. J. Engine Res, 2005, vol. 6, no. JER01505. pp. 453-463.

87. P. Attard, et al. Ion Current Combustion Technology for Controlled Auto Ignition Gasoline Engines. Int. J. Engine Res, 2007, vol. 8, no. JER03604. pp. 429-437.

88. D. Blundell, et al. Design and Evaluation of the ELEVATE Two Stroke Automotive Engine. S.A.E Technical Paper Series, 2003, no. 2003-01-0403.

89. K. James, et al. Ionisation and Ionisation Rate of a Two-Stroke HCCI Engine Fuelled with E85 for Control Feedback, SAE 2010 World Congress. Detroit, Michigan, USA , April, 2010, no. 2010-01-1247.

90. P. Strandh, et al. Ion Current Sensing for HCCI Combustion Feedback. S.A.E Technical Paper Series, 2003, no. 2003-01-3216.

91. Autopressnews Fords Lean Burn GDI. Available from: http://www.autopressnews.com/2009/01/volvo/direct_injection_ecoboost.jpg. Data accessed 09/10/2011.

92. D. Panousakis, et al. Analysis of SI Combustion Diagnostics Methods using Ion-Current Sensing Techniques. S.A.E Technical Paper Series, 2006, vol. SP-2015, no. 2006-01-1345.

93. Emission test cycles, 2 Sept 2011. http://www.dieselnet.com/standards/cycles Date accessed 10/09/2011

94. K. Schnauffer. Engine Cylinder Flame Propagation Studies by New Methods. S.A.E Technical Paper Series, 1934, vol. 29, no. 340074. pp. 17-24.

95. J. Lawton, et al. Electrical Aspects of Combustion. Clarendon Press Oxford, 1965. ISBN0198553412

96. S. Yoshiyama, et al. Fundamental Study on Combustion Diagnostics using a Spark Plug as Ion Probe. S.A.E Technical Paper Series, 2000, no. 2000-01-2828.

97. A. Saitzkoff, et al. An Ionization Equilibrium Analysis of the Spark Plug as an Ionization Sensor. S.A.E Technical Paper Series, 1996, vol. SP-1149, no. 960337.

98. A. Franke, et al. Analysis of the Ionization Equilibrium in the Post-Flame Zone. S.A.E Technical Paper Series, 2003, no. 2003-01-0715.

99. R. Reinmann, et al. Local Air-Fuel Ratio Measurments using the Spark Plug as an Ionization Sensor. S.A.E Technical Paper Series, 1997, no. 970856.

100. S. Byttner et al. Estimation of Combustion Variability using in-Cylinder Ionization Measurements. S.A.E Technical Paper Series, 2001, vol. SP-1638, no. 2001-01-3485.

101. P. Einewall, et al. The Potential of using The Ion Current Signal for Optimizing Engine Stability - Comparisons of Lean and EGR (Stoichiometric) Operation. S.A.E Technical Paper Series, 2003, vol. SP-1749, no. 2003-01-0717.

102. S. Byttner, et al. An Ion Current Algorithm for Fast Determination of High Combustion Variability. S.A.E Technical Paper Series, 2004, vol. SP-1822, no. 2004-01-0522.

103. S. Byttner, et al. Using Multiple Cylinder Ion Measurements for Improved Estimation of Combustion Variability. S.A.E Technical Paper Series, 2005, vol. SP-1975, no. 2005-01-0042.

104. A. Saitzkoff, et al. In Cylinder Pressure Measurements using the Spark Plug as an Ionization Sensor. S.A.E Technical Paper Series, 1997, vol. SP-1236, no. 970857.

105. N. Wickstrom, et al. Estimating Pressure Peak Position and Air-Fuel Ratio using the Ionization Current and Artificial Neural Networks. IEEE, 1998, no. 0-7803-4269-0. pp. 972-977.

106. M. Hellring, et al. An Ion Current Based Peak Finding Algorithm for Pressure Peak Position Estimation 2000-01-2829 ed. Society of Automotive Engineers, 2000.

107. M. Hellring, et al. A Comparison of Ion Current Based Algorithms for Peak Pressure Position Control. 2001-01-1920 ed. Society of Automotive Engineers, 2001.

108. Y. Moudden, et al. Indirect Measurement of Cylinder Pressure Peak Position in an Internal Combustion Engine. IEEE Instrumentation and Measurement, 2001, no. 0-7803-6646-8. pp. 772-777.

109. A. Gazis, et al. Computationally Inexpensive Methods of Ion Current Signal Manipulation for Predicting the Characteristics of Engine in-Cylinder Pressure. ImechE Int. J. Engine Res, 2006, vol. 7, no. 10.1243/14680874JER04005. pp. 271-282.

110. N. Rivara, et al. Peak Pressure Position Control Fo Four Cylinders through the Ion Current Method. S.A.E Technical Paper Series, 2009, no. 2009-01-0235.

111. G. Dong, et al. A Method of Torque Estimation Utilizing Ionization Sensing Technology in Internal Combustion SI Engines. IEEE, 2006, no. 1-4244-0759-1. pp. 32-36.

112. M. Hellring, et al. 1999-01-1161 Robust AFR Estimation using the Ion Current and Neural Networks. Elecronic Engine Controls 1999: Neural Networks, Diagnostic and Electronic Hardware, and Controls SP-1419, 1999. ISSN 0148-7191.

113. H. Klovmark, et al. Estimating the Air/Fuel Ratio from Gaussian Parameterizations of the Ionization Currents in Internal Combustion SI Engines. S.A.E Technical Paper Series, 2000, vol. SP-1501, no. 2000-01-1245.

114. X Wu, et al. Investigation of Air-Fuel Ratio Control using Ionic Current Signal. J. Automotive Engineering, 2007, vol. 221, no. part D. pp. 1139-1146.

115. A. A. Abdel-Rehim, et al. Impact of A/F Ratio on Ion Current Features using Spark Plug with Negative Polarity. S.A.E Technical Paper Series, 2008, no. 2008-01-1005.

116. D. Schneider, et al. Real Time Air/Fuel Ratio Control in a Small SI Engine the Ionic Current Signal. S.A.E Technical Paper Series, 1999, vol. P-348, no. 1999-01-3323.

117. M. Hellring, et al. Spark Advance Control using the Ion Current and Neural Soft Sensors. S.A.E Technical Paper Series, 1999, vol. SP-1419, no. 1999-01-1162.

118. G. W. Malaczynski, et al. Real Time Digital Signal Processing of Ionization Current for Engine Diagnostics and Control. 2003-01-1119 S.A.E Technical Paper Series, 2003.

119. N. Milovanovic, et al. The Advance Combustion Control in a Hybrid SI/HCCI Enigne by using Ion Current Sensing. S.A.E Technical Paper Series, 2006, no. 20065415.

120. N. Cavina, et al. Individual Cylinder Combustion Control Based on Real Time Processing of Ion Current Signals. S.A.E Technical Paper Series, 2007, vol. SP-2087, no. 2007-01-1510.

121. A. H. Shamekhi. et al. Fuzzy Control of Spark Advance by Ion Current Sensing. J. Automotive Engineering, 2007, vol. 221, no. Part D. pp. 335-342.

122. M. Kaiadi, et al. Closed Loop Combustion Control using Ion Current Signal in a 6 Cylinder Port Injected Natural Gas Engine. S.A.E Technical Paper Series, 2008, no. 2008-01-2453.

123. S. Yuichi, et al. S.I. Engine Combustion Flame Propagation Measurement and Knocking Analysis by Ion Current Probes Including Moving Intake and Exhaust Valve Faces. S.A.E Technical Paper Series, 2007, vol. SP-2084, no. 2007-01-1420.

124. Abhijit et al. Ionization Signal Response during Combustion Knock and Comparison to Cylinder Pressure for SI Engines. S.A.E Technical Paper Series, 2008, no. 2008-01-0981.

125. E. A. VanDyne, et al. Misfire Detection from Ionization Feedback Utilizing the SmartFire Plasma Ignition Technology. S.A.E Technical Paper Series, 2000, no. 2000-01-1377.

126. D. Lundstrom, et al. Misfire Detection for Pre-chamber SI Engines using Ion Sensing and Rotational Speed Measurements. S.A.E Technical Paper Series, 2001, vol. SP-1586, no. 2001-01-0993.

127. T. Jungmann. The Technology of the V12 Engine for the Maybach. AutoTechnology, 2002, available from: http://www.atzonline.com/index.php;do=show/site=a4e/sid=7034247474a5f4f3ef1210160970781/alloc=1/id=4379. Data accessed: 10/09/2011.

128. S. Yoshiyama, et al. Ion Current in a Homogeneous Charge Compression Ignition Engine. S.A.E Technical Paper Series, 2007, no. 2007-01-4052.

129. M. Larsson et al. Ion Current Sensing in an Optical HCCI Engine with Negative Valve Overlap. S.A.E Technical Paper Series, 2007, no. 2007-01-0009.

130. A Vressner, et al. Multiple Point Ion Current Diagnostics in a HCCI Enigne. S.A.E Technical Paper Series, 2004, no. 2004-01-0934.

131. Delphi Corporation. Deliphi Ionization Current Sensing Ignition Subsystem. Delphi corporation , 2009, available from: http://delphi.com/shared/pdf/ppd/pwrtrn/ionization-current-sensing-ignition-susbystem.pdf. Data accessed 10/09/2011

132. L. Koopmans, et al. Location of the First Auto Ignition Sites for Two HCCI Systems in a Direct Injection Engine. S.A.E Technical Paper Series, 2004, vol. 2004-01-0564, no. SP-1819.

133. P.J. Baugh. Oxford University press, Gas Chromatography: A Practical Approach 1993. Generating Alaytical Results from Time Slice Data, pp. 65-66. ISBN 0199632715

134. S. Onishi, et al. Active Thermo Atmosphere Combustion (ATAC) A New Combustion Process for Internal Combustion Engines. S.A.E Technical Paper Series,1979. no. 790501.

135. J. S .R. Jang. Neuro Fuzzy and Soft Computing a Computational Approach to Learning and Machine Intelligence. Prentice Hall inc, 1997. ISBN: 0132610663

136. R. Rojas. Neural Networks. Berlin: Springer-Verlag, 1996. ISBN: 3540605053

137. dSPACE GmBH. Embedded Success dSPACE. dSPACE GmBH. 2011 Available from: http://www.dspace.de/de/gmb/home.cfm. Data Accessed: 10/09/2011

138. J Forster, et al. Ion Current Sensing for Spark Ignition Engines. S.A.E Technical paper series. Electronic Engine Control 1999: Sensors,Actuators and Development Tools SP_1418. 1999-01-0204

139. T. Morel, et al. Model for Heat Transfer and Combustion in Spark Ignition Engines and its Comarison with Experiments. S.A.E Techical Paper Series, 29-01-1988, 1988, vol. 880198.

140. N. C. Blizard, et al. Experimental and Theoretical Investigation of Turbulent Burning Model for Internal Combustion Engines. S.A.E. Transactions, 25/02/1974, 1974, vol. 83, no. 740191.

141. S. D. Hires, et al. The Prediction of Ignition Delay and Combustion Intervals for Homogeneous Charge, Spark Ignition Engine. S.A.E. Transactions, 27/02/1978, 1978, vol. 87, no. 780232.

142. MATHWORKS, Inc. Documentation- Neural Network Toolbox Tansig. The MathWorks, Inc. Available from: http://www.mathworks.co.uk/help/toolbox/nnet/ref/tansig.html Date Accessed 10/09/2011.

143. A. Renganathan. The Levenberg-Marquardt Algorithm. Cronos.Rutgers.Edu, 2004.

144. SAE, HCCI combustion drawing. Available from http://www.sae.org/dlymagazineimages/6320_6209_ART.jpg Date Accessed 18/01/2011

145. MATHWORKS, inc diagram of the fuzzy inference system. Available from: http://www.mathworks.com/help/toolbox/fuzzy/fp243dup9.html Date Accessed 18/01/2011.

146. Greenhouse Gas Reduction Strategies in the Transport Sector: Preliminary Report, OECD/ITF, 2008. Page iii.

147 "Christiaan Huygens." Encyclopædia Britannica. Encyclopaedia Britannica Online. Encyclopaedia Britannica, 2011.
http://www.britannica.com/EBchecked/topic/277775/Christiaan-Huygens. Date Accessed 26/07/2011

145 "The cost of vehicle tax for cars, motorcycles, light goods vehicles and trade licences" Directgov, 2011.
http://www.direct.gov.uk/en/Motoring/OwningAVehicle/HowToTaxYourVehicle/DG_10012524.Date Accessed 29/07/2011

146 " Nissan leaf electric car, the basic range" Nissan US, 2011,
http://www.nissanusa.com/leaf-electric-car/index#/leaf-electric-car/theBasicsRange/index. Date Accessed 29/07/2011

147 An Independent Report on the Future of the Automotive Industry in the UK. New Automotive Innovation and Growth Team (NAIGT). Department for business enterprise & regulatory reform . May 2009.

148 R. Potenza et al, Multicylinder engine pressure reconstruction using NARX neural networks and crank kinematics. International Journal of Engine Research. 2007, vol. 8 no.6 499-518. doi: 10.1243/14680874JER01207

149 R. Villarino et al. Fast In-Cylinder Pressure Reconstruction from Structure-Borne Sound using EM Algorithm. IEEE international conference on Acoustics, Speech, and Signal Processing, 2003. Vol. 6 597-600. 0-7803-7663-3

APPENDICES

APPENDIX A. OVERALL MOBILE IONISATION CIRCUIT DIAGRAM
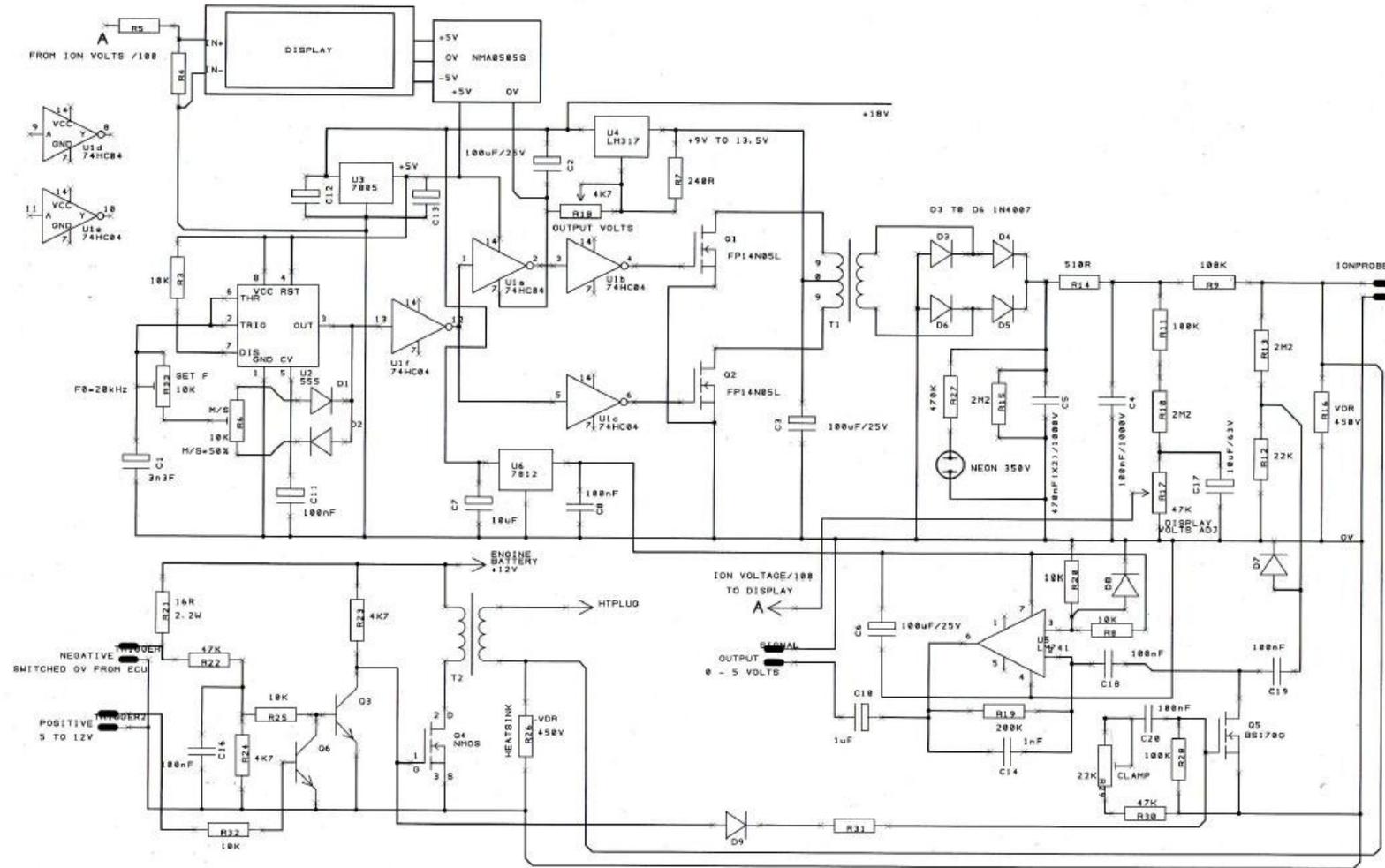
APPENDIX B. MATLAB M-FILE S-FUNCTIONS FOR MIMO CONTROL

M-FILE S-FUNCTION FOR INVERSE ANN MODELS

```matlab
function train2-2(block)
%MSFUNTMPL_BASIC A template for a Leve-2 M-file S-function
%   The M-file S-function is written as a MATLAB function with the
%   same name as the S-functtrainion. Replace 'msfuntmpl_basic' with the
%   name of your S-function.
%
%   It should be noted that the M-file S-function is very similar
%   to Level-2 C-Mex S-functions. You should be able to get more
%   information for each of the block methods by referring to the
%   documentation for C-Mex S-functions.
%
%   Copyright 2003-2007 The MathWorks, Inc.


%%
%% The setup method is used to setup the basic attributes of the
%% S-function such as ports, parameters, etc. Do not add any other
%% calls to the main body of the function.
%%
setup(block);

%endfunction

%% Function: setup ===================================================
%% Abstract:
%%   Set up the S-function block's basic characteristics such as:
%%   - Input ports
%%   - Output ports
%%   - Dialog parameters
%%   - Options
%%
%%   Required         : Yes
%%   C-Mex counterpart: mdlInitializeSizes
%%
function setup(block)

% Register number of ports
block.NumInputPorts  = 21;
block.NumOutputPorts = 12;

% Setup port properties to be inherited or dynamic
block.SetPreCompInpPortInfoToDynamic;
block.SetPreCompOutPortInfoToDynamic;

% inputs for spark timing inverse model

%Peak pressure position
% Override input port properties
block.InputPort(1).SamplingMode = 'frame';
block.InputPort(1).Dimensions       = [1000 1];
block.InputPort(1).DatatypeID  = 0;  % double
block.InputPort(1).Complexity  = 'Real';
block.InputPort(1).DirectFeedthrough = true;

%Engine speed RPM
```

```matlab
% Override input port properties
block.InputPort(2).SamplingMode = 'frame';
block.InputPort(2).Dimensions       = [1000 1];
block.InputPort(2).DatatypeID  = 0;  % double
block.InputPort(2).Complexity  = 'Real';
block.InputPort(2).DirectFeedthrough = true;


%Spark timing control signal
% Override input port properties
block.InputPort(3).SamplingMode = 'frame';
block.InputPort(3).Dimensions       = [1000 1];
block.InputPort(3).DatatypeID  = 0;  % double
block.InputPort(3).Complexity  = 'Real';
block.InputPort(3).DirectFeedthrough = true;


%weights matrix IW
% Override input port properties
block.InputPort(4).SamplingMode = 'frame';
block.InputPort(4).Dimensions       = [20 3];
block.InputPort(4).DatatypeID  = 0;  % double
block.InputPort(4).Complexity  = 'Real';
block.InputPort(4).DirectFeedthrough = true;


% Weights matrix LW
% Override input port properties
block.InputPort(5).SamplingMode = 'frame';
block.InputPort(5).Dimensions       = [10 20];
block.InputPort(5).DatatypeID  = 0;  % double
block.InputPort(5).Complexity  = 'Real';
block.InputPort(5).DirectFeedthrough = true;


% Weights matrix LW2
% Override input port properties
block.InputPort(6).SamplingMode = 'frame';
block.InputPort(6).Dimensions       = [10 1];
block.InputPort(6).DatatypeID  = 0;  % double
block.InputPort(6).Complexity  = 'Real';
block.InputPort(6).DirectFeedthrough = true;


%weights matrix b1
% Override input port properties
block.InputPort(7).SamplingMode = 'frame';
block.InputPort(7).Dimensions       = [20 1];
block.InputPort(7).DatatypeID  = 0;  % double
block.InputPort(7).Complexity  = 'Real';
block.InputPort(7).DirectFeedthrough = true;


%weigths matrix b2
% Override input port properties
block.InputPort(8).SamplingMode = 'frame';
block.InputPort(8).Dimensions       = [10 1];
block.InputPort(8).DatatypeID  = 0;  % double
block.InputPort(8).Complexity  = 'Real';
block.InputPort(8).DirectFeedthrough = true;


%weigths matrix b3
% Override input port properties
block.InputPort(9).SamplingMode = 'frame';
block.InputPort(9).Dimensions       = [1 1];
block.InputPort(9).DatatypeID  = 0;  % double
```

```matlab
block.InputPort(9).Complexity  = 'Real';
block.InputPort(9).DirectFeedthrough = true;




%inputs for fuel pulse width inverse model

% air fuel ratio
block.InputPort(10).SamplingMode = 'frame';
block.InputPort(10).Dimensions        = [1000 1];
block.InputPort(10).DatatypeID  = 0;  % double
block.InputPort(10).Complexity  = 'Real';
block.InputPort(10).DirectFeedthrough = true;

% Mass air flow
block.InputPort(11).SamplingMode = 'frame';
block.InputPort(11).Dimensions        = [1000 1];
block.InputPort(11).DatatypeID  = 0;  % double
block.InputPort(11).Complexity  = 'Real';
block.InputPort(11).DirectFeedthrough = true;

% fuel injection pulse width

block.InputPort(12).SamplingMode = 'frame';
block.InputPort(12).Dimensions        = [1000 1];
block.InputPort(12).DatatypeID  = 0;  % double
block.InputPort(12).Complexity  = 'Real';
block.InputPort(12).DirectFeedthrough = true;

%weights matrix IW
% Override input port properties
block.InputPort(13).SamplingMode = 'frame';
block.InputPort(13).Dimensions        = [20 3];
block.InputPort(13).DatatypeID  = 0;  % double
block.InputPort(13).Complexity  = 'Real';
block.InputPort(13).DirectFeedthrough = true;

% Weights matrix LW
% Override input port properties
block.InputPort(14).SamplingMode = 'frame';
block.InputPort(14).Dimensions        = [10 20];
block.InputPort(14).DatatypeID  = 0;  % double
block.InputPort(14).Complexity  = 'Real';
block.InputPort(14).DirectFeedthrough = true;

% Weights matrix LW2
% Override input port properties
block.InputPort(15).SamplingMode = 'frame';
block.InputPort(15).Dimensions        = [10 1];
block.InputPort(15).DatatypeID  = 0;  % double
block.InputPort(15).Complexity  = 'Real';
block.InputPort(15).DirectFeedthrough = true;

%weights matrix b1
% Override input port properties
block.InputPort(16).SamplingMode = 'frame';
block.InputPort(16).Dimensions        = [20 1];
block.InputPort(16).DatatypeID  = 0;  % double
block.InputPort(16).Complexity  = 'Real';
```

```matlab
block.InputPort(16).DirectFeedthrough = true;

%weigths matrix b2
% Override input port properties
block.InputPort(17).SamplingMode = 'frame';
block.InputPort(17).Dimensions      = [10 1];
block.InputPort(17).DatatypeID  = 0;  % double
block.InputPort(17).Complexity  = 'Real';
block.InputPort(17).DirectFeedthrough = true;


%weigths matrix b3
% Override input port properties
block.InputPort(18).SamplingMode = 'frame';
block.InputPort(18).Dimensions      = [1 1];
block.InputPort(18).DatatypeID  = 0;  % double
block.InputPort(18).Complexity  = 'Real';
block.InputPort(18).DirectFeedthrough = true;


%throttle
block.InputPort(19).SamplingMode = 'frame';
block.InputPort(19).Dimensions      = [1000 1];
block.InputPort(19).DatatypeID  = 0;  % double
block.InputPort(19).Complexity  = 'Real';
block.InputPort(19).DirectFeedthrough = true;


%boulian spark
block.InputPort(20).SamplingMode = 'sample';
block.InputPort(20).Dimensions      = [1 1];
block.InputPort(20).DatatypeID  = 0;  % double
block.InputPort(20).Complexity  = 'Real';
block.InputPort(20).DirectFeedthrough = true;
% boulian fuel
block.InputPort(21).SamplingMode = 'sample';
block.InputPort(21).Dimensions      = [1 1];
block.InputPort(21).DatatypeID  = 0;  % double
block.InputPort(21).Complexity  = 'Real';
block.InputPort(21).DirectFeedthrough = true;

%% outputs for spark timing inverse model
% weights matrix IW
% Override output port properties
block.OutputPort(1).SamplingMode = 'frame';
block.OutputPort(1).Dimensions      = [20 3];
block.OutputPort(1).DatatypeID  = 0; % double
block.OutputPort(1).Complexity  = 'Real';


% weights matrix LW
% Override output port properties
block.OutputPort(2).SamplingMode = 'frame';
block.OutputPort(2).Dimensions      = [10 20];
block.OutputPort(2).DatatypeID  = 0; % double
block.OutputPort(2).Complexity  = 'Real';


% weights matrix LW1
% Override output port properties
block.OutputPort(3).SamplingMode = 'frame';
block.OutputPort(3).Dimensions      = [1 10];
block.OutputPort(3).DatatypeID  = 0; % double
block.OutputPort(3).Complexity  = 'Real';
```

```matlab
%weights matrix b1
% Override output port properties
block.OutputPort(4).SamplingMode = 'frame';
block.OutputPort(4).Dimensions     = [20 1];
block.OutputPort(4).DatatypeID  = 0; % double
block.OutputPort(4).Complexity  = 'Real';


%weights matrix b2
% Override output port properties
block.OutputPort(5).SamplingMode = 'frame';
block.OutputPort(5).Dimensions     = [10 1];
block.OutputPort(5).DatatypeID  = 0; % double
block.OutputPort(5).Complexity  = 'Real';


%weights matrix b3
% Override output port properties
block.OutputPort(6).SamplingMode = 'frame';
block.OutputPort(6).Dimensions     = [1 1];
block.OutputPort(6).DatatypeID  = 0; % double
block.OutputPort(6).Complexity  = 'Real';




%% outputs for fuel pulse width model

% weights matrix IW
% Override output port properties
block.OutputPort(7).SamplingMode = 'frame';
block.OutputPort(7).Dimensions     = [20 3];
block.OutputPort(7).DatatypeID  = 0; % double
block.OutputPort(7).Complexity  = 'Real';


% weights matrix LW
% Override output port properties
block.OutputPort(8).SamplingMode = 'frame';
block.OutputPort(8).Dimensions     = [10 20];
block.OutputPort(8).DatatypeID  = 0; % double
block.OutputPort(8).Complexity  = 'Real';


% weights matrix LW1
% Override output port properties
block.OutputPort(9).SamplingMode = 'frame';
block.OutputPort(9).Dimensions     = [1 10];
block.OutputPort(9).DatatypeID  = 0; % double
block.OutputPort(9).Complexity  = 'Real';


%weights matrix b1
% Override output port properties
block.OutputPort(10).SamplingMode = 'frame';
block.OutputPort(10).Dimensions     = [20 1];
block.OutputPort(10).DatatypeID  = 0; % double
block.OutputPort(10).Complexity  = 'Real';


%weights matrix b2
% Override output port properties
block.OutputPort(11).SamplingMode = 'frame';
block.OutputPort(11).Dimensions     = [10 1];
block.OutputPort(11).DatatypeID  = 0; % double
block.OutputPort(11).Complexity  = 'Real';
```

```
%weights matrix b3
% Override output port properties
block.OutputPort(12).SamplingMode = 'frame';
block.OutputPort(12).Dimensions     = [1 1];
block.OutputPort(12).DatatypeID  = 0; % double
block.OutputPort(12).Complexity  = 'Real';


% Register parameters
block.NumDialogPrms     = 0;

% Register sample times
%  [0 offset]            : Continuous sample time
%  [positive_num offset] : Discrete sample time
%
%  [-1, 0]               : Inherited sample time
%  [-2, 0]               : Variable sample time
block.SampleTimes = [6 0];

% Specify the block simStateComliance. The allowed values are:
%    'UnknownSimState', < The default setting; warn and assume
DefaultSimState
%    'DefaultSimState', < Same sim state as a built-in block
%    'HasNoSimState',   < No sim state
%    'CustomSimState',  < Has GetSimState and SetSimState methods
%    'DisallowSimState' < Error out when saving or restoring the model sim
state
block.SimStateCompliance = 'DefaultSimState';

%% ----------------------------------------------------------------
%% The M-file S-function uses an internal registry for all
%% block methods. You should register all relevant methods
%% (optional and required) as illustrated below. You may choose
%% any suitable name for the methods and implement these methods
%% as local functions within the same file. See comments
%% provided for each function for more information.
%% ----------------------------------------------------------------

block.RegBlockMethod('Outputs', @Outputs);     % Required
block.RegBlockMethod('Terminate', @Terminate); % Required

%end setup

%%
%% Outputs:
%%   Functionality    : Called to generate block outputs in
%%                      simulation step
%%   Required         : Yes
%%   C-MEX counterpart: mdlOutputs
%%
function Outputs(block)

boulian1 = (block.InputPort(20).data);
boulian2 = (block.InputPort(21).data);
```

```matlab
%%network 1
%ppp
i(1,:)=(block.InputPort(1).data');
%eng speed
i(2,:)=(block.InputPort(2).data');
%MAF
i(3,:)=(block.InputPort(11).data');
%spk timing
o=(block.InputPort(3).data');

% randomize sequence

sze = size(i);
col=randperm(sze(1,2));
i=i(:,col);
o=o(:,col);

ps.name='mapminmax';
ps.xrows=1;
ps.xmax= 30;
ps.xmin= 0;
ps.xrange=30;
ps.yrows=1;
ps.ymax=1;
ps.ymin=-1;
ps.yrange=2;

ps1.name='mapminmax';
ps1.xrows=1;
ps1.xmax= 6000;
ps1.xmin= 1000;
ps1.xrange=5000;
ps1.yrows=1;
ps1.ymax=1;
ps1.ymin=-1;
ps1.yrange=2;

ps2.name='mapminmax';
ps2.xrows=1;
ps2.xmax=0;
ps2.xmin=-30;
ps2.xrange=30;
ps2.yrows=1;
ps2.ymax=1;
ps2.ymin=-1;
ps2.yrange=2;

ps4.name='mapminmax';
ps4.xrows=1;
ps4.xmax= 130;
ps4.xmin= 0;
ps4.xrange=130;
ps4.yrows=1;
ps4.ymax=1;
ps4.ymin=-1;
ps4.yrange=2;

i(1,:)=mapminmax('apply',i(1,:),ps);
i(2,:)=mapminmax('apply',i(2,:),ps1);
i(3,:)=mapminmax('apply',i(3,:),ps4);
```

```matlab
o=mapminmax('apply',o,ps2);


i=con2seq(i);
o=con2seq(o);


weights1 =block.InputPort(4).data;
weights2=block.InputPort(5).data;
weights3=block.InputPort(6).data';
bias1=block.InputPort(7).data;
bias2=block.InputPort(8).data;
bias3=block.InputPort(9).data;


p=i;
t=o;


%set input range
inputr=[-1 1; -1 1;-1 1];
outputr=[-1 1];


net=newff(inputr,outputr,[20 10],{'tansig' 'tansig' 'satlins'});


%set weights
net.IW{1,1}=weights1;
net.LW{2,1}=weights2;
net.LW{3,2}=weights3;
net.b{1,1}=bias1;
net.b{2,1}=bias2;
net.b{3,1}=bias3;
 %turn off GUI
 %net.trainParam.showWindow = false;
 if  boulian1 == 1;
     1
net.adaptparam.passes=2;
% train network
net=adapt(net,p,t);


% simulate data
yy= sim(net,p);
end
% correlation of simulation
R = corrcoef(cell2mat(t),cell2mat(yy));


%output new weights and R value
IW1=net.IW;
II=IW1(1,1);
II=cell2mat(II);
block.OutputPort(1).Data =  II ;


LW= net.LW(2,1);
LW=cell2mat(LW);
block.OutputPort(2).Data =  LW ;



LW2= net.LW(3,2);
LW2=cell2mat(LW2);
block.OutputPort(3).Data =  LW2 ;


b1= net.b(1,1);
b1=cell2mat(b1);
```

```matlab
block.OutputPort(4).Data =  b1 ;


b2= net.b(2,1);
b2=cell2mat(b2);
block.OutputPort(5).Data =  b2 ;


b3= net.b(3,1);
b3=cell2mat(b3);
block.OutputPort(6).Data =  b3 ;



%% network 2

%AFR
in(1,:)=(block.InputPort(10).data');


%rpm
in(2,:)=(block.InputPort(2).data');


%throttle
in(3,:)=(block.InputPort(19).data');


%Fuel pulse width
out=(block.InputPort(12).data');



% randomize sequence
%
sze2 = size(in);
col2=randperm(sze2(1,2));
in=in(:,col2);
out=out(:,col2);

% range of inputs

%AFR range

ps3.name='mapminmax';
ps3.xrows=1;
ps3.xmax= 19;
ps3.xmin= 9;
ps3.xrange=10;
ps3.yrows=1;
ps3.ymax=1;
ps3.ymin=-1;
ps3.yrange=2;



% RPM range

ps4.name='mapminmax';
ps4.xrows=1;
ps4.xmax= 6000;
ps4.xmin= 1000;
ps4.xrange=5000;
ps4.yrows=1;
ps4.ymax=1;
ps4.ymin=-1;
```

```matlab
ps4.yrange=2;

% throttle range

ps6.name='mapminmax';
ps6.xrows=1;
ps6.xmax= 70;
ps6.xmin= 20;
ps6.xrange=50;
ps6.yrows=1;
ps6.ymax=1;
ps6.ymin=-1;
ps6.yrange=2;

% range of output

ps5.name='mapminmax';
ps5.xrows=1;
ps5.xmax=12000;
ps5.xmin=4000;
ps5.xrange=8000;
ps5.yrows=1;
ps5.ymax=1;
ps5.ymin=-1;
ps5.yrange=2;

in(1,:)=mapminmax('apply',in(1,:),ps3);
in(2,:)=mapminmax('apply',in(2,:),ps4);
in(3,:)=mapminmax('apply',in(3,:),ps6);

out(1,:)=mapminmax('apply',out(1,:),ps5);

in=con2seq(in);
out=con2seq(out);

weights11 =block.InputPort(13).data;
weights22=block.InputPort(14).data;
weights33=block.InputPort(15).data';
bias11=block.InputPort(16).data;
bias22=block.InputPort(17).data;
bias33=block.InputPort(18).data;

p2=in;
t2=out;
% initialise network
%limmits 2

%set input range
inputr=[-1 1; -1 1; -1 1];
outputr=[-1 1];

net2=newff(inputr,outputr,[20 10],{'tansig' 'tansig' 'satlins'});

% set parameters for network

net2.IW{1,1}=weights11;
 net2.LW{2,1}=weights22;
net2.LW{3,2}=weights33;
 net2.b{1,1}=bias11;
```

```matlab
 net2.b{2,1}=bias22;
net2.b{3,1}=bias33;
 %turn off GUI
% net2.trainParam.showWindow = true;

if boulian2 == 1;
    2
 net2.adaptparam.passes=2;
 net2=adapt(net2,p2,t2);

yy2= sim(net2,p2);
 R = corrcoef(cell2mat(t2),cell2mat(yy2));

end

% weight outputs second network
IW11=net2.IW;
II1=IW11(1,1);
II1=cell2mat(II1);
block.OutputPort(7).Data =  II1 ;

LW1= net2.LW(2,1);
LW1=cell2mat(LW1);
block.OutputPort(8).Data =  LW1 ;


LW21= net2.LW(3,2);
LW21=cell2mat(LW21);
block.OutputPort(9).Data =  LW21 ;

b11= net2.b(1,1);
b11=cell2mat(b11);
block.OutputPort(10).Data =  b11 ;

b22= net2.b(2,1);
b22=cell2mat(b22);
block.OutputPort(11).Data =  b22 ;

b33= net2.b(3,1);
b33=cell2mat(b33);
block.OutputPort(12).Data =  b33 ;

%end Outputs

%%
%% Terminate:
%%   Functionality    : Called at the end of simulation for cleanup
%%   Required         : Yes
%%   C-MEX counterpart: mdlTerminate
%%
function Terminate(block)

%end Terminate
```

M-FILE S-FUNCTION FOR FEED FORWARD ANN CONTROL MODEL

```matlab
function tr2-2(block)
%MSFUNTMPL_BASIC A template for a Leve-2 M-file S-function
%   The M-file S-function is written as a MATLAB function with the
%   same name as the S-functtrainion. Replace 'msfuntmpl_basic' with the
%   name of your S-function.
%
%   It should be noted that the M-file S-function is very similar
%   to Level-2 C-Mex S-functions. You should be able to get more
%   information for each of the block methods by referring to the
%   documentation for C-Mex S-functions.
%
%   Copyright 2003-2007 The MathWorks, Inc.


%%
%% The setup method is used to setup the basic attributes of the
%% S-function such as ports, parameters, etc. Do not add any other
%% calls to the main body of the function.
%%
setup(block);

%endfunction

%% Function: setup ===================================================
%% Abstract:
%%   Set up the S-function block's basic characteristics such as:
%%   - Input ports
%%   - Output ports
%%   - Dialog parameters
%%   - Options
%%
%%   Required         : Yes
%%   C-Mex counterpart: mdlInitializeSizes
%%
function setup(block)

% Register number of ports
block.NumInputPorts  = 14;
block.NumOutputPorts = 8;

% Setup port properties to be inherited or dynamic
block.SetPreCompInpPortInfoToDynamic;
block.SetPreCompOutPortInfoToDynamic;

% Override input port properties RPM
block.InputPort(1).SamplingMode = 'frame';
block.InputPort(1).Dimensions        = [4000 1];
block.InputPort(1).DatatypeID  = 0;  % double
block.InputPort(1).Complexity  = 'Real';
block.InputPort(1).DirectFeedthrough = true;


% Override input port properties error PPP
block.InputPort(2).SamplingMode = 'frame';
block.InputPort(2).Dimensions        = [4000 1];
block.InputPort(2).DatatypeID  = 0;  % double
block.InputPort(2).Complexity  = 'Real';
block.InputPort(2).DirectFeedthrough = true;
```

```matlab
% Override input port properties error AFR
block.InputPort(3).SamplingMode = 'frame';
block.InputPort(3).Dimensions      = [4000 1];
block.InputPort(3).DatatypeID  = 0;  % double
block.InputPort(3).Complexity  = 'Real';
block.InputPort(3).DirectFeedthrough = true;


% Override input port properties MAF
block.InputPort(4).SamplingMode = 'frame';
block.InputPort(4).Dimensions      = [4000 1];
block.InputPort(4).DatatypeID  = 0;  % double
block.InputPort(4).Complexity  = 'Real';
block.InputPort(4).DirectFeedthrough = true;


% Override input port properties Error spark control
block.InputPort(5).SamplingMode = 'frame';
block.InputPort(5).Dimensions      = [4000 1];
block.InputPort(5).DatatypeID  = 0;  % double
block.InputPort(5).Complexity  = 'Real';
block.InputPort(5).DirectFeedthrough = true;


% Override input port properties Error injection control
block.InputPort(6).SamplingMode = 'frame';
block.InputPort(6).Dimensions      = [4000 1];
block.InputPort(6).DatatypeID  = 0;  % double
block.InputPort(6).Complexity  = 'Real';
block.InputPort(6).DirectFeedthrough = true;


% Override input port properties IW
block.InputPort(7).SamplingMode = 'frame';
block.InputPort(7).Dimensions      = [30 2];
block.InputPort(7).DatatypeID  = 0;  % double
block.InputPort(7).Complexity  = 'Real';
block.InputPort(7).DirectFeedthrough = true;


% Override input port properties LW 1
block.InputPort(8).SamplingMode = 'frame';
block.InputPort(8).Dimensions      = [15 30];
block.InputPort(8).DatatypeID  = 0;  % double
block.InputPort(8).Complexity  = 'Real';
block.InputPort(8).DirectFeedthrough = true;


% Override input port properties LW 2
block.InputPort(9).SamplingMode = 'frame';
block.InputPort(9).Dimensions      = [10 15];
block.InputPort(9).DatatypeID  = 0;  % double
block.InputPort(9).Complexity  = 'Real';
block.InputPort(9).DirectFeedthrough = true;


% Override input port properties LW 3
block.InputPort(10).SamplingMode = 'frame';
block.InputPort(10).Dimensions      = [2 10];
block.InputPort(10).DatatypeID  = 0;  % double
block.InputPort(10).Complexity  = 'Real';
block.InputPort(10).DirectFeedthrough = true;


% Override input port properties bias 1
block.InputPort(11).SamplingMode = 'frame';
```

```matlab
block.InputPort(11).Dimensions      = [30 1];
block.InputPort(11).DatatypeID  = 0;  % double
block.InputPort(11).Complexity  = 'Real';
block.InputPort(11).DirectFeedthrough = true;

% Override input port properties bias 2
block.InputPort(12).SamplingMode = 'frame';
block.InputPort(12).Dimensions      = [15 1];
block.InputPort(12).DatatypeID  = 0;  % double
block.InputPort(12).Complexity  = 'Real';
block.InputPort(12).DirectFeedthrough = true;

% Override input port properties bias 3
block.InputPort(13).SamplingMode = 'frame';
block.InputPort(13).Dimensions      = [10 1];
block.InputPort(13).DatatypeID  = 0;  % double
block.InputPort(13).Complexity  = 'Real';
block.InputPort(13).DirectFeedthrough = true;


% Override input port properties bias 4
block.InputPort(14).SamplingMode = 'frame';
block.InputPort(14).Dimensions      = [2 1];
block.InputPort(14).DatatypeID  = 0;  % double
block.InputPort(14).Complexity  = 'Real';
block.InputPort(14).DirectFeedthrough = true;

% OUTPUTS


% Override output port properties IW
block.OutputPort(1).SamplingMode = 'frame';
block.OutputPort(1).Dimensions      = [30 2];
block.OutputPort(1).DatatypeID  = 0; % double
block.OutputPort(1).Complexity  = 'Real';

% Override output port properties LW 1
block.OutputPort(2).SamplingMode = 'frame';
block.OutputPort(2).Dimensions      = [15 30];
block.OutputPort(2).DatatypeID  = 0; % double
block.OutputPort(2).Complexity  = 'Real';

% Override output port properties LW 2
block.OutputPort(3).SamplingMode = 'frame';
block.OutputPort(3).Dimensions      = [10 15];
block.OutputPort(3).DatatypeID  = 0; % double
block.OutputPort(3).Complexity  = 'Real';

% Override output port properties LW 3
block.OutputPort(4).SamplingMode = 'frame';
block.OutputPort(4).Dimensions      = [2 10];
block.OutputPort(4).DatatypeID  = 0; % double
block.OutputPort(4).Complexity  = 'Real';

% Override output port properties b1
block.OutputPort(5).SamplingMode = 'frame';
block.OutputPort(5).Dimensions      = [30 1];
block.OutputPort(5).DatatypeID  = 0; % double
block.OutputPort(5).Complexity  = 'Real';
```

```matlab
% Override output port properties b2
block.OutputPort(6).SamplingMode = 'frame';
block.OutputPort(6).Dimensions    = [15 1];
block.OutputPort(6).DatatypeID  = 0; % double
block.OutputPort(6).Complexity  = 'Real';


% Override output port properties b3
block.OutputPort(7).SamplingMode = 'frame';
block.OutputPort(7).Dimensions    = [10 1];
block.OutputPort(7).DatatypeID  = 0; % double
block.OutputPort(7).Complexity  = 'Real';


% Override output port properties b4
block.OutputPort(8).SamplingMode = 'frame';
block.OutputPort(8).Dimensions    = [2 1];
block.OutputPort(8).DatatypeID  = 0; % double
block.OutputPort(8).Complexity  = 'Real';


% Register parameters
block.NumDialogPrms     = 0;


% Register sample times
%  [0 offset]            : Continuous sample time
%  [positive_num offset] : Discrete sample time
%
%  [-1, 0]               : Inherited sample time
%  [-2, 0]               : Variable sample time
block.SampleTimes = [10 0];


% Specify the block simStateComliance. The allowed values are:
%    'UnknownSimState', < The default setting; warn and assume
DefaultSimState
%    'DefaultSimState', < Same sim state as a built-in block
%    'HasNoSimState',   < No sim state
%    'CustomSimState',  < Has GetSimState and SetSimState methods
%    'DisallowSimState' < Error out when saving or restoring the model sim
state
block.SimStateCompliance = 'DefaultSimState';


%% ----------------------------------------------------------------
%% The M-file S-function uses an internal registry for all
%% block methods. You should register all relevant methods
%% (optional and required) as illustrated below. You may choose
%% any suitable name for the methods and implement these methods
%% as local functions within the same file. See comments
%% provided for each function for more information.
%% ----------------------------------------------------------------




block.RegBlockMethod('Outputs', @Outputs);      % Required
block.RegBlockMethod('Terminate', @Terminate); % Required


%end setup
```

```matlab
%%
%% Outputs:
%%   Functionality    : Called to generate block outputs in
%%                       simulation step
%%   Required         : Yes
%%   C-MEX counterpart: mdlOutputs
%%
function Outputs(block)


%inputs and outputs
% RPM
i(1,:)=(block.InputPort(1).data');

% MAF
i(2,:)=(block.InputPort(4).data');

% NN ouput control error spark timing component
o(1,:)=(block.InputPort(5).data');
% NN output control error fuel injection component
o(2,:)=(block.InputPort(6).data');

sze = size(i);
col=randperm(sze(1,2));
i=i(:,col);
o=o(:,col);



%% set range of inputs and outputs

% eng speed range

ps.name='mapminmax';
ps.xrows=1;
ps.xmax= 6000;
ps.xmin= 1000;
ps.xrange=5000;
ps.yrows=1;
ps.ymax=1;
ps.ymin=-1;
ps.yrange=2;



% MAF range

ps3.name='mapminmax';
ps3.xrows=1;
ps3.xmax= 130;
ps3.xmin= 0;
ps3.xrange=130;
ps3.yrows=1;
ps3.ymax=1;
ps3.ymin=-1;
ps3.yrange=2;

% Spark timing error range
```

```matlab
ps4.name='mapminmax';
ps4.xrows=1;
ps4.xmax=15;
ps4.xmin=-15;
ps4.xrange=30;
ps4.yrows=1;
ps4.ymax=1;
ps4.ymin=-1;
ps4.yrange=2;

% fuel pulse width error range

ps5.name='mapminmax';
ps5.xrows=1;
ps5.xmax=3000;
ps5.xmin=-3000;
ps5.xrange=6000;
ps5.yrows=1;
ps5.ymax=1;
ps5.ymin=-1;
ps5.yrange=2;


i(1,:)=mapminmax('apply',i(1,:),ps);
i(2,:)=mapminmax('apply',i(2,:),ps3);

o(1,:)=mapminmax('apply',o(1,:),ps4);
o(2,:)=mapminmax('apply',o(2,:),ps5);

i=con2seq(i);
o=con2seq(o);

% weights

weights1 =block.InputPort(7).data;
weights2=block.InputPort(8).data;
weights3=block.InputPort(9).data;
weights4=block.InputPort(10).data;
bias1=block.InputPort(11).data;
bias2=block.InputPort(12).data;
bias3=block.InputPort(13).data;
bias4=block.InputPort(14).data;
% prep inputs

p=i;
t=o;

%% network setup
inputr=[-1 1; -1 1 ];
outputr=[-1 1; -1 1];

net=newff(inputr,outputr,[30 15 10],{'tansig' 'tansig' 'tansig'
'satlins'});

 %set weights
 net.IW{1,1}=weights1;
  net.LW{2,1}=weights2;
  net.LW{3,2}=weights3;
  net.LW{4,3}=weights4;
```

```matlab
 net.b{1,1}=bias1;
 net.b{2,1}=bias2;
 net.b{3,1}=bias3;
 net.b{4,1}=bias4;
 %turn off GUI
 net.trainParam.showWindow = false;
net.adaptParam.passes=1;
% train network
net=adapt(net,p,t);
% simulate data
yy= sim(net,p);
% correlation of simulation
R = corrcoef(cell2mat(t),cell2mat(yy))


%output new weights and R value

II=net.IW(1,1);
II=cell2mat(II);
block.OutputPort(1).Data =  II ;

LW1= net.LW(2,1);
LW1=cell2mat(LW1);
block.OutputPort(2).Data =  LW1 ;

LW2= net.LW(3,2);
LW2=cell2mat(LW2);

block.OutputPort(3).Data = LW2  ;

LW3= net.LW(4,3);
LW3=cell2mat(LW3);

block.OutputPort(4).Data = LW3  ;

b1= net.b(1,1);
b1=cell2mat(b1);
block.OutputPort(5).Data =  b1 ;

b2= net.b(2,1);
b2=cell2mat(b2);
block.OutputPort(6).Data =  b2 ;

b3= net.b(3,1);
b3=cell2mat(b3);
block.OutputPort(7).Data =  b3 ;

b4= net.b(4,1);
b4=cell2mat(b4);
block.OutputPort(8).Data =  b4 ;


%end Outputs
```

```
%%
%% Terminate:
%%   Functionality    : Called at the end of simulation for cleanup
%%   Required         : Yes
%%   C-MEX counterpart: mdlTerminate
%%
function Terminate(block)

%end Terminate
```

APPENDIX C. SIMULINK TOP LEVEL FOR MIMO ENGINE CONTROL
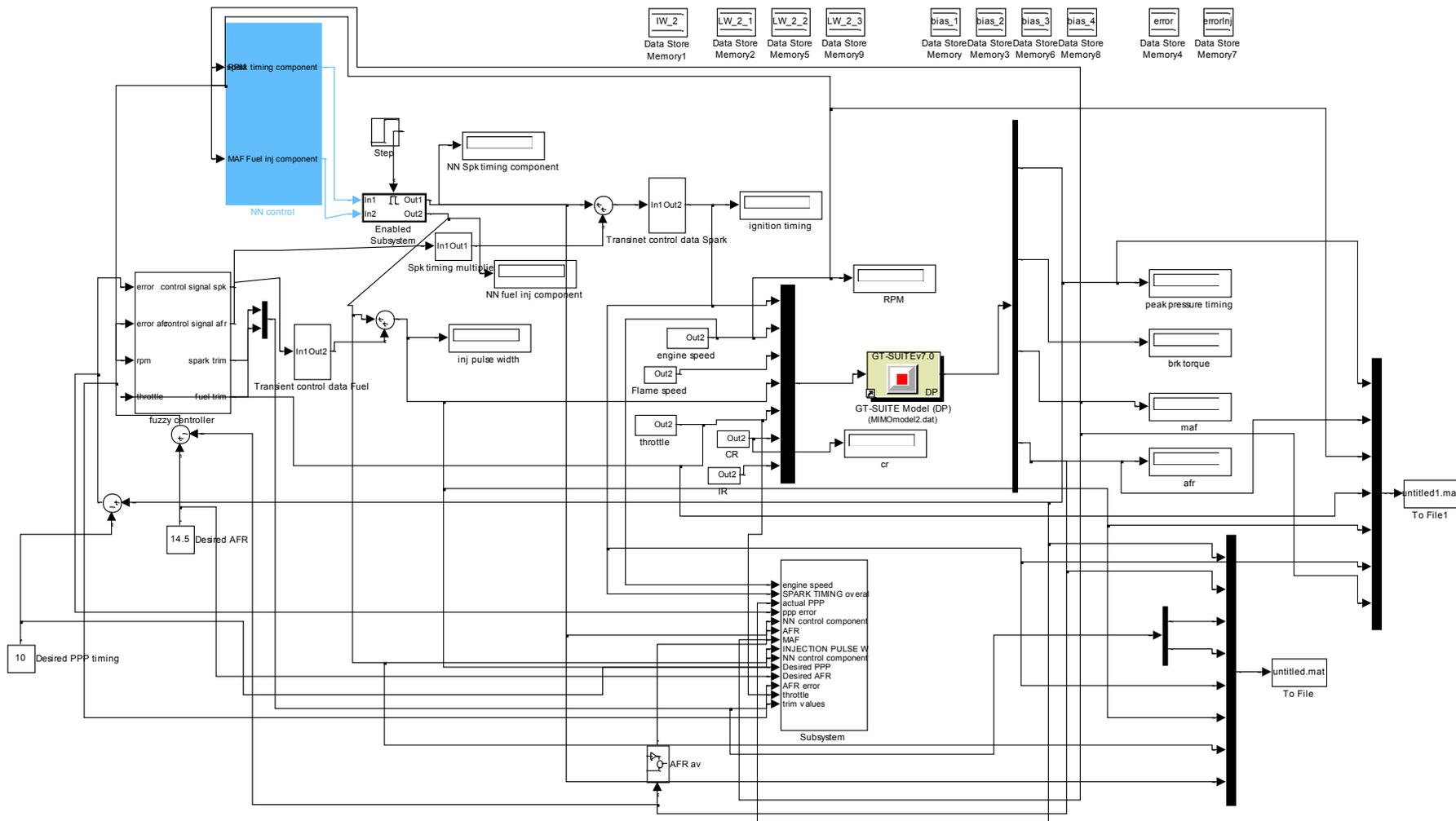
PUBLICATIONS

Keith James; Rui Chen and James Turner. Ionisation and Ionisation Rate of a Two-Stroke HCCI Engine Fuelled with E85 for Control Feedback, *SAE 2010 World Congress*. Detroit, Michigan, USA , April, 2010, no. 2010-01-1247.

Keith James and Rui Chen, Review on the Engine Control Strategies for Homogenous charge compression Ignition Combustion. 10th International Symposium on Advanced Vehicle Control, AVEC10. August 2010.