Intelligent Techniques for Improved Engine Fuel Economy

S.H.Lee
BEng (Hons), MSc, AMIMechE

A thesis submitted in partial fulfilment of the requirement of the University of Brighton for the degree of Doctor of Philosophy

April 2011

School of Computing, Engineering & Mathematics
University of Brighton

Abstract

This thesis presents an investigation into a novel method of estimating the trajectory (future direction and elevation) of a vehicle, and subsequently influencing the control of an engine. The technique represents a convenient and robust method of achieving road prediction, to form a fuzzy system that 'looks ahead', leading potentially to improved fuel consumption and a consequent reduction in exhaust emissions. The work described in this thesis brings together two modern technologies, Neuro-fuzzy techniques and Global Positioning System, and applies them to engine/vehicle control.

The intelligent GPS-based control system presented in this thesis utilises information about the current vehicle position and upcoming terrain in order to reduce vehicle fuel consumption as well as improve road safety and comfort. The development of such in-vehicle control systems has provided static and dynamic road information. The vehicle running parameters have been mathematically defined whilst the engine control algorithms were derived from a custom-built engine test-rig. As the vehicle travelled along a particular route, the road information such as gradient and position was stored with the past trajectory using a Neuro-fuzzy technique. This road information was continuously updated and replaced by new data as the vehicle moved along, thereby adjusting the engine control parameters to reflect the actual current vehicle running data. The control system essentially used a fuzzy logic derived relief map of the test route and this was further validated and corrected based on the past trajectory from the in-vehicle GPS sensor. The simulation model demonstrated the feasibility and robustness of the control system for motor vehicle control applications.

Contents

Table of Figures

List of Tables

Acknowledgements

I would like to express my sincere thanks to Prof. R. J. Howlett and Dr S. D. Walters for their academic and intellectual guidance, advice, assistance and support during the course of this research program, and the preparation of this thesis. I should also like to thank my family for their understanding, encouragement, support and immense patience during this period, when they were in fact, deprived of my time and attention.

Declaration

I declare that the research contained in this thesis, unless otherwise formally indicated within the text, is the original work of the author. The thesis has not been previously submitted to this or any other university for a degree, and does not incorporate any material already submitted for a degree.

S.H.Lee

December 2010

Chapter 1 : Introduction

The environmental challenges of the 21st century will require tremendous technological advancements in the automotive industry. The next generation of power unit in these vehicles has to be much cleaner and more efficient than the current ‘conventional’ internal combustion engines. Ultimately, zero emission power units will be required, pushing the development of electric and fuel cell technologies. However, at the moment, fuel cells are complex and currently too expensive for widespread application. The advancement of battery and fuel cell technology still limits the potential of both developments. Hence, internal combustion (IC) engines are still widely used and available in automotive applications.

1.1 Brief and Motivation

Europe, the United States, and much of the rest of the world, have legislative controls which govern the permissible levels of pollutants in the exhausts of IC engines. Maintaining these standards in current engines demands strict control of operational parameters using a microprocessor-based Engine Management System (EMS) or Engine Control Unit (ECU) and an increasingly comprehensive array of sensors.

Engine management technology for controlling automotive engines is well established. The EMS implements control strategies which aim to achieve optimum efficiency and high output power when required, whilst also maintaining low emission levels. At the same time, in many cases, the EMS must operate the engine in a region favourable to the functioning of a three-way catalytic converter, which further reduces the harmful content of the exhaust. The EMS governs the ignition timing and the amount of fuel admitted to the engine; also, in some cases the amount of exhaust gas recirculation (EGR), and other parameters in advanced engine designs, for example, the valve timing. It selects values for these parameters from measured quantities such as engine speed, load torque, air mass flow rate, inlet manifold air pressure, temperatures at various points, and throttle position. The EMS has a further role, in that legislation in the US and now in Europe demands that automotive engines possess an on-board diagnostic (OBD) system. The OBD system must indicate when emissions do not conform to standards, or when fault conditions occur that could lead to excessive emissions. In modern automotive engines, the EMS and associated sensor technology forms a complex electronic system.

The principal air pollutants emitted by small gasoline engines are HC and CO; total NOx emissions from these engines are insignificant. The European Commission Directorate-General for the Environment, the Auto Oil II study concluded that there was no strong environmental need for further controls on CO emissions [1]. Hydrocarbons contribute to ground level ozone formation leading to risk of damage to human respiratory systems. In addition some hydrocarbons are carcinogenic. Carbon monoxide is harmful to human health, interfering with the ability of the blood to supply oxygen to the body's tissues. However airborne CO concentrations are now sufficiently low as not to be a general concern.

The legislative pressures that influence automotive engine manufacturers have not in the past been applied to small off-road engines in Europe. For this reason, complex electronic control, of the sort found in automotive engines, has seldom proved cost-effective. However, current European emission standards are now forcing manufacturers to limit the emissions of IC engines [2]. In addition, there are increasing pressures from the environmental lobby in this direction. This will make electronic control necessary, even for small engines where previously it has not been economical.

The low-cost of small passenger vehicles will necessitate that control systems which are applied to them are themselves of minimum cost, with a requirement for a small number of inexpensive sensors, and sensing and control algorithms that can be executed in real-time on simple microcontrollers.

The UK and Europe have many major cities and surrounding villages which have excessive level of traffic, often making travel around the cities difficult. The exhaust emissions and noise of the increasing number of vehicles going in and around the city add to the problem. In urban areas, due to their beneficial effects on the environment, the use of small and fuel-efficient passenger vehicles could be a medium term solution for the improvement of traffic and more particularly for a healthier environment.

A small vehicle is typically accommodating the driver and one or two passengers. Such vehicles are designed and produced for economic purposes when the use of materials, resources and fuel are the major consideration. The economy of operating such a small car has often been helped by simply-constructed three-wheeled carriages with very small engines. Two-stroke internal combustion engines are widely used in the early versions of such vehicles. Many evolve from a small motorcycle design, of the kind found in parts of developing countries, which currently use unsophisticated gasoline or diesel engines, and produce high levels of emissions.

The Southeast Asian versions of such small cars, known as tuk-tuks are a form of urban transport. They are particularly popular where traffic congestion is a major problem, such as in Bangkok, Thailand and other Thai cities. These vehicles have a small cabin for the driver in the front and seating for three in relative comfort in the rear. They are very manoeuvrable and can turn around in one lane of traffic with room to spare. Tuk-Tuks were introduced in Brighton, England, in 2006, where a fleet of twelve, known as ‘TucTuc’ here, operated using compressed natural gas, as the first motorised rickshaw service in Europe, between Brighton Marina and Hove, via Brighton railway station. The fleet is also planning to expand into London.

These vehicles spend most of the time on journeys around town which involve a lot of stop-and-go traffic. Most of these journeys are repeated throughout their service life, for example between the railway station and the town centre. The current advancement and availability of vehicle positioning and sensor technology, i.e. Global Positioning System (GPS) could be used to accurately pinpoint the location of the vehicle. This information provides a reference of the road ahead and thus be used to predict the future direction and elevation of the vehicle; this information is expected to be of use for dynamically tuning the engine and/or giving an inference command to the EMS.

This project aims to benefit from the latest research project co-funded by the INTERREG IIIa European research programme, and carried out by the University of Brighton's Centre for Smart Systems on the development of Intelligent Vehicle On-board Systems (VBIS). It has focused on the development of an innovative motive power control system from the fusion of externally-acquired positioning data and internal vehicle operating parameters; also from analysis and optimisation of these data using intelligent techniques and tools for application to small cars. These facilitate the reduction of exhaust emissions and fuel consumption through precise control of the combustion process.

1.2 Project Aim

The aim of this research project was to investigate the monitoring and control of small cars using intelligent techniques to achieve reduced emissions and improved fuel economy.

1.3 Research Objectives

The project was focussed on the development of engine control systems using the fusion of externally-acquired positioning data and engine operating parameters; also from analysis and optimisation of these data using intelligent techniques and tools for application to motor vehicles. These have facilitated the reduction of exhaust emissions and fuel consumption through precise control of the combustion process.

- To review small internal combustion engines, instrumentation and related global positioning technology;
- To conduct a survey of current intelligent control techniques in automotive systems;
- To devise a low-cost road mapping system and a computationally efficient intelligent control algorithm for small passenger vehicles;
- To implement and evaluate the low-cost engine control system that has been devised.

1.4 Rationale

A considerable amount of research and development work has been expended on power units for automotive applications. However, there has been comparatively little work done on motive power units for small passenger vehicle applications mainly outside the mainstream automotive sector. In the longer term, fuel cells are likely to provide an efficient and clean method of converting fuel (hydrogen, methane, alcohol, diesel, etc) directly into electricity, which in turn may be converted into motive power. In the short to medium term, efficient small IC engine units form a compromise technology which offers benefits over un-regulated small engines.

These efficient power units are expected to be better than un-regulated ones in terms of average fuel consumption, emissions and performance without compromising consumer expectations with respect to performance, comfort, safety, quality and cost of ownership. In order to achieve these goals, it is very important to optimise the architecture and components of the small engine, but, just as important is the control algorithm that is used to control the complete engine system.

The prime motive power source of most small cars is a small internal combustion engine, fuelled using conventional diesel, petrol, liquefied petroleum gas (LPG) or natural gas. The on-board EMS governs the control strategies aiming to achieve optimum efficiency and high output power.

With the help of GPS, intelligent techniques and low-cost sensors, information such as the gradient of the road and the past trajectory can be obtained and used to predict future possible control actions. An optimisation algorithm can be devised based on the captured information to find the best inputs with respect to the vehicle operating parameters, such as speed and load. Figure 1.1 shows the schematic diagram of a GPS acquisition system in conjunction with a fuzzy mapping technique developed for this project. The system is able to produce a relief map of a route derived from fuzzy logic processed GPS data where route parameters such as road gradient and curvature are stored.

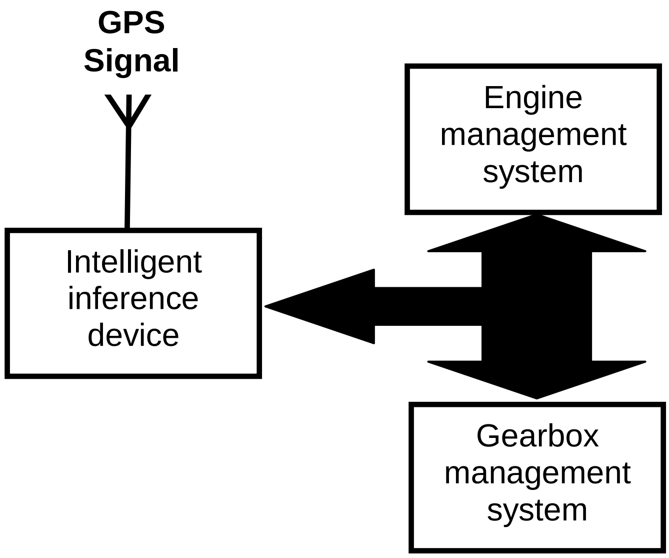

Figure 1.1: Extra-vehicular data sources for engine/vehicle control

An essential component of such a complicated system will be a computer control system and software. The system will be able to combine the mapping data and the operating data from the vehicle. A major part of the work is to further consolidate and analyse these data; i.e.: vehicle, engine operating parameters and operating characteristic of the vehicle. The system will execute algorithms that optimise the operation of the engine, GPS sensor and other system components to ensure minimal use of energy and minimised emissions.

1.5 Outline of the Thesis

This thesis describes a novel method of mapping a route subsequently to be used for controlling a gasoline powered internal combustion engine. This is accomplished by analysing the effectiveness of the devised control system using Matlab/Simulink model.

Chapter 1 (this chapter) has introduced the background, motivation and need for a low-cost control system in small city passenger vehicles.

Chapter 2 gives the reader some background information on the four-stroke cycle of an internal combustion engine.

Chapter 3 gives an introduction to fuzzy logic and Neuro-fuzzy techniques.

Chapter 4 contains a literature review of relevant research being carried out by others in the area of engine and vehicle control.

Chapter 5 details two previous experiments carried out at the beginning of this project. The work describes here was to ascertain two different fuzzy-derived techniques for controlling small internal combustion engine and modelling fuel spray penetration in the cylinder of a diesel internal combustion engine. The outcome and experience gained allows for the exploration of GPS vehicle control.

Chapter 6 introduces a new and original method of Neuro-fuzzy road mapping technique. It also explains how the relief map is being derived from fuzzy logic processed GPS data and how it was further used as a reference for the vehicle/engine control.

Chapter 7 continues by detailing the experimental set up and the experimental work performed on an engine test-rig.

Chapter 8 contains the results and general and specific conclusions drawn from the work carried out.

Chapter 9 discusses further research work and recommendations are made in a number of areas.

The Appendices contain samples of the program code which was developed and papers which were published by the author during the course of the investigation.

Chapter 2 : Engine Technology

There are two types of internal combustion engines commonly used today, spark ignition (SI) or gasoline engines and compression ignition (CI) or diesel engines. In SI engines, the air and fuel mixture is ignited by an electrical spark. They are also known as Otto cycle engines after the name of its inventor, Nikolaus August Otto who invented the four-stroke engine in 1876. In CI engines, the rise in temperature due to compression pressure is sufficient to cause spontaneous combustion of the fuel without the need for a spark. The fuel diesel is named after the inventor of the engine, Rudolf Diesel who invented the diesel cycle engine in 1893.

2.1 The Internal Combustion Engine

The engine is a device for converting the internal energy stored in its fuel into mechanical energy. An internal combustion engine is an engine in which the combustion, or rapid oxidation, of gas and air occurs in a confined space called a combustion chamber. A supply of air and fuel mixture is fed to the inside of the cylinder where it is compressed and then burnt. This internal combustion releases heat energy which is then converted into useful mechanical work as the high gas pressure generated forces the piston to move along its stroke in the cylinder.

To enable the piston movement to be harnessed, the driving thrust on the piston is transmitted by means of a connecting-rod to a crankshaft whose function is to convert the linear piston motion in the cylinder to a rotary crankshaft movement as shown in Figure 2.1. The piston can thus be made to repeat its movement to and fro, due to the constraints of the crankshaft crankpin’s circular path and the guiding cylinder.

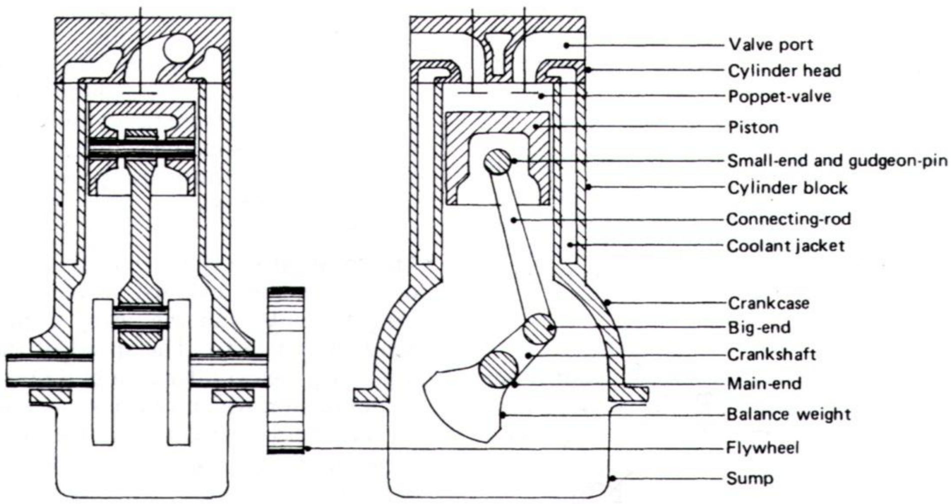

Figure 2.1: Sectioned view of the basic engine

2.1.1 Operating Principles

The four-stroke cycle of an internal combustion engine is the cycle most commonly used for automotive and industrial purposes today, for example, cars and trucks, small motive units, etc. The first engine to operate successfully on the four-stroke cycle was constructed in 1876 by Nicolaus August Otto. In this type of engine a sequence of events is continuously repeated all the time it is running; this sequence of operation and the associated pressure & volume graph are shown in Figures 2.2 to 2.5.

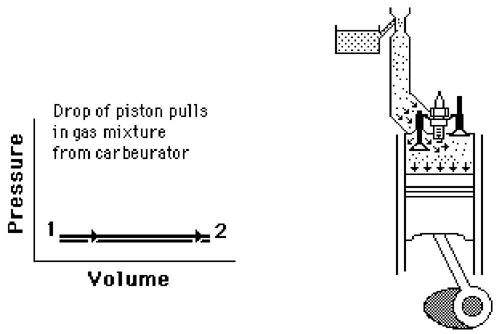

Figure 2.2: Induction stroke

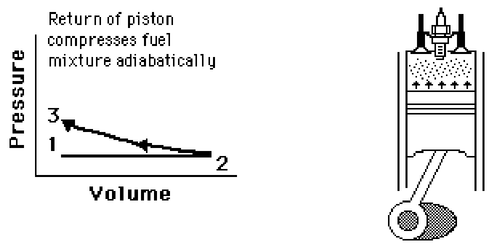

Figure 2.3: Compression stroke

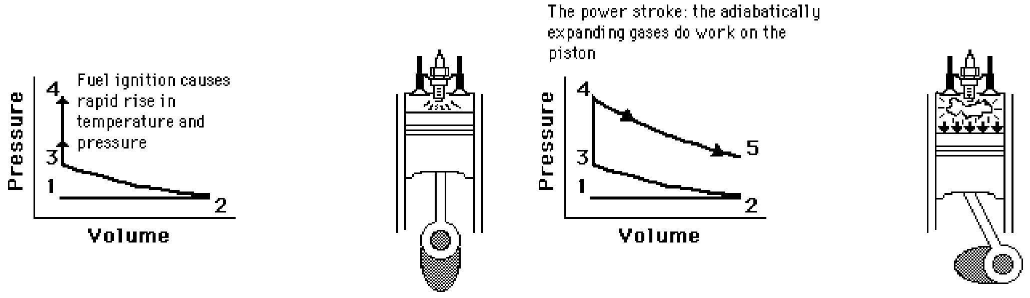

Figure 2.4: Power stroke

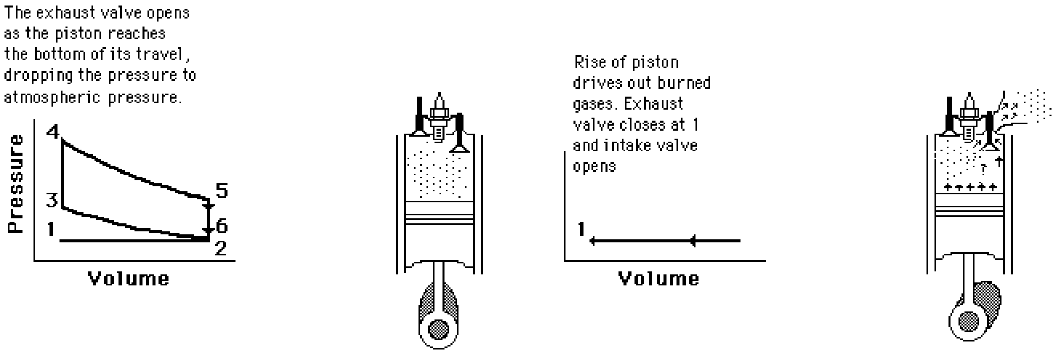

Figure 2.5: Exhaust stroke

- The induction stroke, during which the combustible charge of air and fuel is drawn into the combustion chamber and cylinder, as a result of the partial vacuum or depression created by the retreating piston.
- The compression stroke, which serves to raise both the pressure and temperature of the combustible charge as it is compressed into the lesser volume of the combustion chamber by the advancing piston.
- The power stroke, immediately preceding which the combustible charge is ignited by the sparking plug and during which the gases expand and perform useful work on the retreating piston.
- The exhaust stroke, during which the products of combustion are purged from the cylinder and combustion chamber by the advancing piston, and discharged into the exhaust system.

It thus follows that one complete cycle of operation occupies two complete revolutions of the engine crankshaft. Since energy is necessarily required to perform the initial induction and compression strokes of the engine piston before firing occurs, an electrical starter motor or a pull-start device (in the case of small utility engines) is used for preliminary cranking of the engine. Once the engine is running the energy required for performing subsequent induction, compression and exhaust strokes is derived from the crankshaft and flywheel system, by virtue of its kinetic energy of rotation.

2.1.2 Fuel Systems

A fuel system for a gasoline engine stores, transfers and filters the gasoline required either by a carburettor or by the pressure regulated circuit of a fuel injection system.

2.1.2.1 Carburation

The function of the carburettor is to mix the fuel with the incoming air in the correct proportions to form a mixture which is combustible under engine operating conditions. Gasoline is a liquid fuel derived from crude petroleum, whose major constituent elements are carbon and hydrogen but also contains three minor elements: sulphur, nitrogen and oxygen. The mixing of gasoline with air in a carburettor is achieved by introducing the liquid gasoline into a rapidly moving air stream which suspends and breaks up the liquid into very tiny droplets. This process is known as atomising the fuel.

2.1.2.2 Air and Fuel Mixture Strengths

The air-fuel ratio (AFR) is often defined in terms of the excess air factor, or lambda (λ). Lambda is defined such that a lambda factor of unity corresponds to an AFR of 14.7:1 at normal temperature and pressure. This is termed the stoichiometric ratio, corresponding to the proportions of air and fuel which are required for complete combustion. A greater proportion of fuel gives a lambda of less than unity, termed a rich mixture, while a greater proportion of air gives a lambda of greater than unity, termed a weak or lean mixture. Maximum power is obtained when lambda is approximately 0.9 and minimum fuel consumption occurs when lambda is approximately 1.1. The AFR essentially sets the operating point of the engine, and in conjunction with the ignition timing angle, determines the output power and the resulting levels of emissions.

A rich mixture containing more than the optimum amount of gasoline, usually produces more power than optimum and lean mixtures, the engine power generally being at its maximum when the mixture is about 15 to 20% rich, that is with the air-fuel ratio of between 12 to 1 and 13 to 1. The exhaust products of these rich mixtures normally have an excess of carbon monoxide and are visually observed as dark cloudy exhaust smoke. Prolonged running with a very rich mixture will result in sooting up of the combustion chamber and of the spark plug electrodes.

A lean mixture, however, containing less than the optimum amount of gasoline, usually produces less power than optimum and rich mixtures, but fuel economy is normally much better than those for the other conditions. For minimum fuel consumption, the mixture can be 15 to 20% weak, that is, with the AFR of between 17 to 1 and 18 to 1. Burning is generally slow, and misfiring, overheating, and incomplete combustion will result if sufficient ignition timing advance is not provided to compensate for this prolonged combustion period [3].

2.1.3 Fuel Injection Systems

The function of a fuel injection system is to monitor the engine's operating parameters, to transfer this information to a metering controller, then to discharge and atomise the fuel into the incoming air stream. The position where the fuel is injected into the air charge considerably influences the performance of the engine.

The advantage of this system is accurate control of the fuel quantity injected into the engine. The idea is that if gasoline is supplied to an electrically controlled fuel injector, at a constant differential pressure, then the amount of fuel injected will be directly proportional to the injector open time termed 'fuel pulse width'. These systems are electronically controlled, allowing the operation of the injection system to be very closely matched to the requirements of the engine. This matching process is carried out during development on test beds. The ideal operating data for a large number of engine operating conditions are stored in a memory in the ECU. Close control of the fuel quantity injected allows the optimum setting of mixture strength. Further advantages of electronic fuel injection control are that overrun cut-off can easily be implemented, fuel can be cut at the engines rpm limit and information on fuel used can be supplied to an on-board computer.

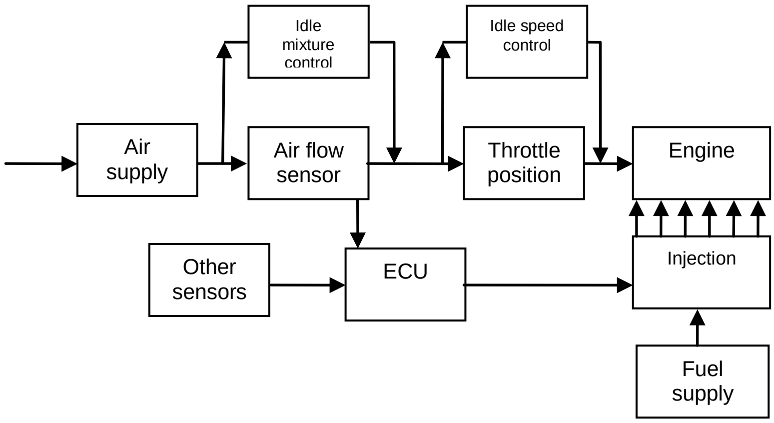

Figure 2.6: Typical control layout for a fuel injection system

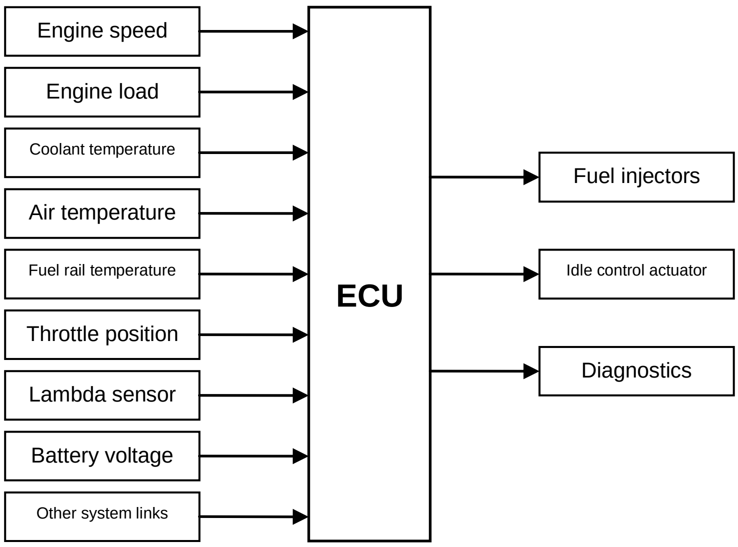

Figure 2.7: Block diagram of input and output parameters

Figure 2.6 shows a typical control layout for a fuel injection system; a block diagram in Figure 2.7 shows the inputs and outputs of a common system. The two most important input parameters to the system are speed and load. The basic fuelling requirement is determined from these inputs in a similar way to the determination of ignition timing.

The development of fuel injection in general and the reduced manufacturing cost have now started to make the carburettor obsolete. As the emission regulations continue to become more stringent, engine manufacturers are being compelled to adopt fuel injection, even on small engines. The adoption by larger engine manufacturers will, in turn, bring the price of the systems down, making them comparable to carburetion techniques on price but better in performance.

2.1.3.1 Mapping

A three-dimensional map shown in Figure 2.8 is used to represent how the information on an engine's fuelling requirements is stored. The information is held in memory in the ECU. When the ECU has determined the look-up value of the fuel pulse width required, corrections to this quantity can be added for engine temperature, throttle opening rate or position and fuel cut-off according to the control strategy of an ECU.

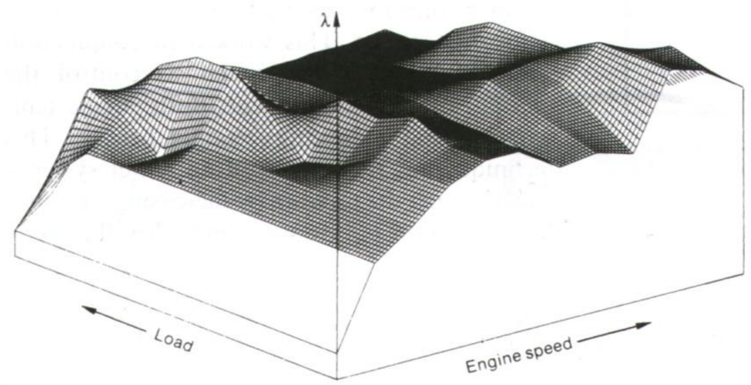

Figure 2.8: Three-dimensional map showing the variation of λ with engine load and speed

Idle speed and fast idle are generally controlled by the ECU and a suitable actuator. It is also possible to have a form of closed-loop control with electronic fuel injection. This involves a lambda sensor to monitor exhaust gas oxygen content. This allows very accurate control of the mixture strength, as the oxygen content of the exhaust is proportional to the air-fuel ratio. The signal from the lambda sensor is used to adjust the fuel pulse width in modern automotive engines.

2.1.4 Ignition Systems

The primary function of the ignition system is to supply a spark inside the cylinder of an internal combustion engine, near the end of the compression stroke, to ignite the compressed charge of air-fuel mixture. For a spark to jump across an air gap of 0.6 mm under normal atmospheric conditions (1 bar), a voltage of 2 to 3 kV is required [4]. In order to generate a high-voltage, an ignition coil, which consists of two coupled coils known as the primary and secondary windings, is fitted. The two coils are wound on to the same iron core, so any change in magnetism of one coil will induce a voltage into the other. This happens when a current is switched on and off to the primary coil. This transformer action is the basic principle of all ignition systems.

2.1.4.1 Magneto

A magneto is a type of ignition system, it provides pulses of electrical power to the spark plugs in some gasoline-powered internal combustion engines where batteries are not available. Magneto are most commonly found on two-stroke and four-stroke engines used in small motorcycles, lawnmowers and chainsaws, and thus serve a similar function as the coil-type ignition system found in cars. In these cases, the magneto’s advantage is in its compact nature and simple reliable function. They are used in most small aircraft, some racing cars and in older tractors. In aircraft, typically each cylinder has two spark plugs, each driven from a separate magneto. This arrangement provides redundancy in the event of a failure of one of the magnetos, and two sparks burn more efficiently than one.

Magnetos combine the functions of a dynamo, contact breaker points and coil into one unit. The engine rotates a coil of wire between the poles of a permanent magnet to provide a basic source of electrical energy. On each revolution, a cam opens the contact breaker one or more times, interrupting the current, causing the voltage in the secondary winding of the coil to reach a very high figure; enough to produce an arc across the electrodes of the spark plug. Because no battery or other source of energy is required, the magneto is a rugged, reliable and self-contained solution to providing ignition of the air-fuel mixture. In some modern small engine designs, an electronic switch replaces the contact breaker.

Since the magneto is a single integrated unit providing its own power source, a vehicle using one has no means of providing electrical power for other needs, such as lighting, which would readily be available from an external dynamo or alternator. Most importantly, there is no provision for a starter motor, so that an alternative starting means will be required for a machine using a magneto. Such means may be a kick-start, starting crank or pull-start.

2.1.4.2 Ignition Timing

For optimum engine operating efficiency the ignition advance facilitates maximum combustion pressure to occur about 10º after top dead centre (TDC). The ideal ignition timing is dependent on two main factors, engine speed and engine load. An increase in engine speed requires the ignition timing to be advanced. The cylinder charge, of air-fuel mixture, requires a certain time to burn. At higher engine speeds the time taken for the piston to travel the same distance reduces. Advancing the time of the spark ensures complete combustion is achieved.

A change in timing due to engine load is also required as the weaker mixture used on low-load conditions burns at a slower rate. In this situation, further ignition advance is necessary. Greater load on the engine requires a richer mixture, which burns more rapidly. In this case some retardation of timing is necessary. In summary, under any condition of engine speed and load an ideal advance angle is required to ensure maximum pressure is achieved in the cylinder just after top dead centre. The ideal advance angle may be further refined by engine temperature and the need to reduce any risk of detonation or knocking.

Spark advance is achieved in a number of ways. The simplest of these is the mechanical system comprising a load sensing device. The manifold air pressure

(MAP) is directly proportional to engine load. A microprocessor-based engine management system may adjust the timing in relation to the temperature as well as speed and load. The values of all ignition timing functions are combined either mechanically or electronically in order to determine the ideal ignition point. The ignition timing also has a significant effect on fuel consumption, torque, and exhaust emissions. Because ignition timing is critical to engine performance, controlling it precisely through all operating conditions has become a major application of digital electronic engine control.

2.1.5 Engine Operating Conditions

When the engine changes from one operating speed condition to another, additional fuel mixture is necessary to compensate for the inertia lag of the heavier fuel injected into the relatively light incoming air stream, and this temporarily produces an overweak mixture. These operating conditions will have to be considered towards the development of intelligent control strategy.

2.1.5.1 Acceleration Response

Enrichment for acceleration is achieved when the throttle is opened suddenly. The rapid rush of intake air through the air flow-meter causes an increased fuel demand signal to be passed to the ECU. If the engine is warming up, the overswing enrichment may have to be supplemented with additional fuel, which signalled to the ECU by the air speed from the air flow sensor.

2.1.5.2 Cold Start

During engine starting up, the very low crankshaft speed of rotation does not produce sufficient air velocity to atomise and support the fuel droplets in the air stream while some of the fuel which has atomised will then condense in the induction ports and on the cylinder walls. To compensate for the loss of effective air-fuel mixture reaching the cylinder, a cold start facility is provided to supply extra fuel only when the engine is being started.

2.1.5.3 Warm-up Period

When starting a cold engine, between 30% and 60% more fuel is required depending on the surrounding temperature. Once the engine is operating, a smaller additional amount of fuel to that normally supplied is still necessary to counteract mixture condensation, and the strength of the mixture should be gradually decreased as the engine warms up, until only the normal operating quantity of fuel is injected into the induction manifold. Information on the operating temperature of the engine is obtained by the temperature sensor which relays a signal voltage to the ECU to enable a mixture correction to be made.

2.1.6 Engine Management

A modern engine management system is essentially a combination of ignition, fuel management and OBD. As the stringent requirements for lower emissions continue, together with the need for better performance under all engine operating conditions, other areas of the engine are constantly being monitored. This control is becoming even more important with the carbon dioxide emission being included in regulations. As more and more systems are integrated then the cost of the electronics necessary will reduce. The computing power required for this type of development is increasing, and use of 32-bit microcontrollers is becoming the norm. The down-side of using a single ECU to control the entire engine or vehicle is the replacement cost of the unit. At present, even a single system ECU can cost a significant amount. However, a comprehensive control unit its advantages such as allowing the expansion of OBD to cover any other system, potentially saving repair time and running costs.

2.1.7 Sensory Systems

In an EMS, sensors and actuators are in many cases the vital components for determining system performance. The sensors and actuators that are available to an EMS are not always what the actual system needs, because the ideal device may not be commercially available at acceptable cost, especially in the case of small engine applications. Often the signal interface or conditioning circuits are designed to adapt to an available sensor or actuator, or the control system is designed in a specific way to fit available sensors or actuators.

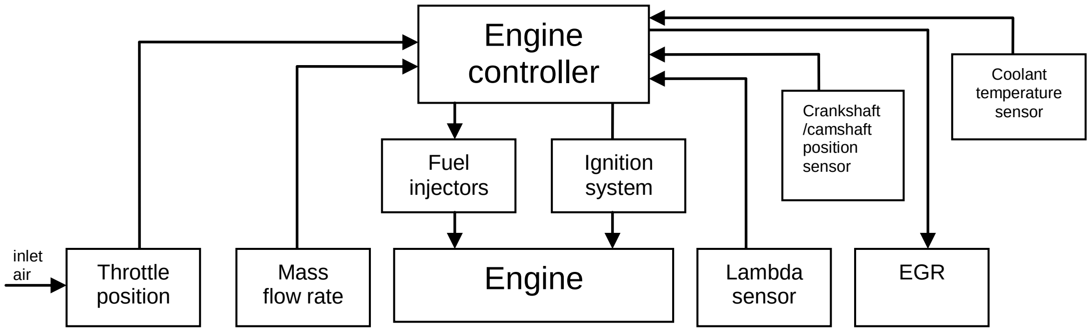

Figure 2.9: Typical engine sensory systems

There are many subsystems in modern engine control systems that operate with sensors and actuators. Figure 2.9 shows a block diagram of a typical electronic engine control system illustrating most of the relevant sensors used for engine control. The position of the throttle, determined by the throttle position sensor (TPS), directly regulates the air flow into the engine, thereby controlling output power. A set of fuel injectors delivers the correct amount of fuel to a corresponding cylinder during the intake stroke under control of the electronic engine controller. The ignition control system fires each spark plug at the appropriate time under control of the electronic engine controller. The EGR is controlled by another output from the engine controller. All vital engine control parameters are based on measurements made by various sensors connected to the engine. Computations made within the engine controller based on these inputs yield output signals to the actuators according to their own control strategy and algorithm used. However, part of the investigation here will be focused on the sensors and actuators that are likely to be used in small engine control. Variables to be measured in a small internal combustion are engine speed, throttle position, engine temperature, air temperature, manifold air pressure, exhaust gas temperature and engine load.

2.1.7.1 Thermistors and Thermocouples

Engine temperature measurement is carried out by a simple thermistor, and in many cases the same sensor is used for the operation of the temperature gauge and to provide information to the fuel control system. The mass of air drawn into the engine depends on the air density, which varies directly with its temperature. The colder the air, the denser it becomes. Therefore, for a given throttle opening, a greater mass of air will enter the cylinder as the temperature of the air intake rises, however its density decreases so that less air is drawn into the cylinder. A separate memory map is used to correct the ideal timing settings. Timing may be retarded when the engine is cold to constitute a more rapid warm up.

The principle of measurement is that a change in temperature will cause a change in resistance of the thermistor, and hence an electrical signal proportional to the measurand can be obtained. Thermocouples on the other hand have two different metals joined together at two junctions. If one junction is at a higher temperature than the other junction, then this will be registered on the engine controller. They are commonly used for measuring high temperature, e.g. exhaust and turbo charger temperature.

2.1.7.2 Engine Speed and Position Sensors

An engine speed sensor consists of a permanent magnet, a winding and a soft iron core. They tend to work on the basic principle of electrical induction and are mounted in proximity to a reluctor disc. The disc has 34 teeth, spaced at 10° intervals around the periphery of the disc. The disc also has two teeth missing, 180° apart, one of which is at a known position before top dead centre (BTDC). Many engines use this technique with minor variations. As a tooth from the reluctor disc passes the core of the sensor, the reluctance of the magnetic circuit is changed. This induces a voltage in the winding, the frequency of the waveform being proportional to the engine speed. The missing tooth causes a ‘missed’ output wave and hence the engine position can be determined. The output voltage of such type of sensors approximates to a sine wave. The amplitude of this signal depends on the rate of change of flux induced by the sensors winding. The most common way of converting this output waveform is to pass it through a Schmitt trigger circuit. This produces constant amplitude but a variable frequency square wave. The analogue signal of the engine speed can be determined by using a suitable signal conditioning circuit.

2.1.7.3 Variable Resistance Sensors

An application for a variable resistance sensor is the throttle position sensor. A device known as a potentiometer produces a voltage proportional to the throttle position. The device can also be used to indicate the rate of change of throttle position. This information is used when implementing acceleration enrichment or inversely, over-run fuel cut-off.

2.1.7.4 Manifold Pressure Sensors

The engine load is proportional to inlet manifold pressure in that high-load conditions produce high pressure and lower load conditions produce low pressure such as cruising. These load sensors are therefore pressure sensors. These pressure sensors are either a gauge or differential package. They are based around a power active element piezo-resistive bridge construction which has been laser-trimmed to enhance device performance. The ‘gauge’ sensors use atmospheric pressure as a reference whereas the ‘differential’ sensors will accept two independent pressure sources simultaneously. The manifold vacuum pressure sensors use atmospheric pressure as a reference; they can either be mounted in the ECU or as a separate unit, and are connected to the inlet manifold with a pipe. These sensors are very responsive to change in throttle position and engine speed, and also they are robust and low cost which make them ideal for small engine control applications.

Chapter 3 : Intelligent Techniques

Intelligent systems, i.e. software systems incorporating artificial intelligence, have shown many advantages in engineering system control and modelling. They have the ability to rapidly model and learn characteristics of multi-variate complex systems, exhibiting advantages in performance over more conventional mathematical techniques. This has led to them being applied in diverse applications in power systems, manufacturing, optimisation, medicine, signal processing, control, robotics, and social/psychological sciences [5, 6]. In industrial automation and process control, fuzzy logic technologies enable the efficient and transparent implementation of human control expertise. For example, an individual control loop of a single industrial process has variables mostly remaining controlled by conventional models such as proportional-integral-derivative (PID). The fuzzy logic system then gives the set values for these controllers based on the process control expertise put in the fuzzy logic rules. In Japan, Germany and France, cars with intelligently controlled components are quite common; the reasons being the control systems in cars are complex and involve multiple parameters. The optimisation of these systems is based on engineering expertise rather than mathematical models. Criteria such as ride-comfort and handling are optimisation goals that cannot be defined mathematically.

3.1 Fuzzy Logic

Fuzzy logic is a problem-solving technique that derives its power from its ability to draw conclusions and generate responses based on vague, ambiguous, incomplete and imprecise information. To simulate this process of human reasoning it applies the mathematical theory of fuzzy sets first defined by Zadeh, in 1965 [7]. Fuzzy inference is the process of formulating a mapping from a given input value to an output value using fuzzy logic. The mapping then provides a basis from which decisions can be made, or patterns discerned. It has been proved that the system can effectively express highly non-linear functional relationships [8]. Fuzzy inference systems (FIS) have been successfully applied in fields such as automatic control, data classification, decision analysis, expert systems and computer vision. Fuzzy control has been shown to provide a convenient calibration procedure that has the potential to lower costs through reducing the effort required in calibrating the engine.

Fuzzy logic provides a practical way to understand and manually influence the mapping behaviour. In general, fuzzy logic uses simple rules to describe the system of interest rather than analytical equations, making it easy to implement. For advantages, such as robustness and speed, fuzzy logic technique is one of the best solutions for system modelling and control. An FIS contains three main components, the fuzzification stage, the rule base and the defuzzification stage. The fuzzification stage is used to transform the so-called crisp values of the input variables into fuzzy membership values. Then, these membership values are processed within the rule-base using conditional 'if-then' statements. The outputs of the rules are summed and defuzzified into a crisp analogue output value. The effects of variations in the parameters of a FIS can be readily understood and this facilitates calibration of the model.

The system inputs, which in this case are the cylinder pressure and the air density, are called linguistic variables, whereas 'high and 'very high' are linguistic values which are characterised by the membership function. Following the evaluation of the rules, the defuzzification transforms the fuzzy membership values into a crisp output value. The complexity of a fuzzy logic system with a fixed input-output structure is determined by the number of membership functions used for the fuzzification and defuzzification and by the number of inference levels. The block diagram of a general fuzzy logic system is shown in Figure 3.1 where x1, x2,…xn stand for n crisp input and y is the crisp output.

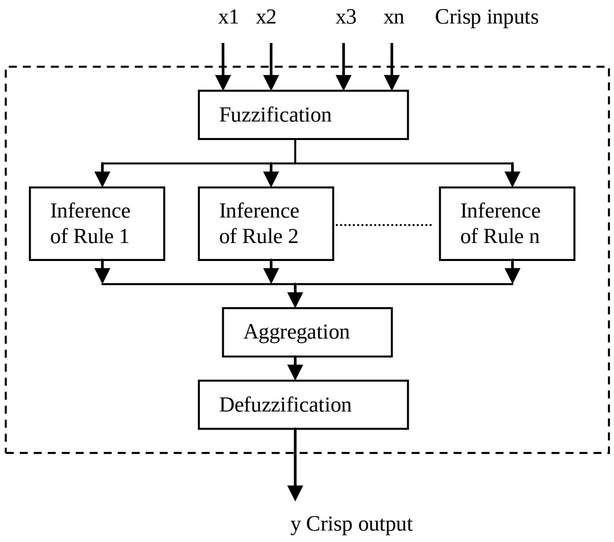

Figure 3.1: Block diagram of a general fuzzy logic system

A fuzzy system of this kind requires that a knowledgeable human operator initialises the system parameters e.g. the membership function bounds. The operator must then optimise these parameters to achieve a required level of mapping accuracy of the physical system by the fuzzy system. While the visual nature of a fuzzy system facilitates the optimisation of the parameters, the need for it to be accomplished manually is a disadvantage.

3.2 Neuro-fuzzy Systems

Adaptive Neuro-Fuzzy Inference Systems (ANFIS), developed in the early 1990s by Jang [9], combines the concepts of fuzzy logic and neural networks to form a hybrid intelligent system that enhances the ability to automatically learn and adapt. Hybrid systems have been used by researchers for modelling and prediction in various engineering systems. The basic idea behind these neuro-adaptive learning techniques is to provide a method for the fuzzy modelling procedure to learn information about a data set, in order to automatically compute the membership function parameters that best allow the associated FIS to track the given input/output data. The membership function parameters are tuned using a combination of least squares estimation and backpropagation algorithm for membership function parameter estimation. These parameters associated with the membership functions will change through the learning process similar to those of a neural network. Their adjustment is facilitated by a gradient vector, which provides a measure of how well the FIS is modelling the input/output data for a given set of parameters. Once the gradient vector is obtained, any of several optimisation routines can be applied in order to adjust the parameters so as to reduce error between the actual and desired outputs. This allows the fuzzy system to learn from the data it is modelling. The approach has the advantage over the pure fuzzy paradigm that the need for the human operator to tune the system by adjusting the bounds of the membership functions is removed.

ANFIS largely removes the requirement for manual optimisation of the fuzzy system parameters. A neural network is used to automatically tune the system parameters, for example the membership function bounds, leading to improved performance without operator intervention. In addition to a purely fuzzy approach, an ANFIS was also developed for the estimation of diesel spray penetration because the combination of neural network and fuzzy logic enables the system to learn and improve its performance based on past data. The Neuro-fuzzy system with the learning capability of neural network and with the advantages of the rule-base fuzzy system can improve the performance significantly and can provide a mechanism to incorporate past observations into the classification process. In a neural network the training essentially builds the system. However using a Neuro-fuzzy scheme, the system is built by fuzzy logic definitions and then it is refined using neural network training algorithms.

3.2.1 Architecture

The initial membership functions and rules for the FIS are designed by employing human knowledge about the target system to be exploited. ANFIS can then refine the fuzzy if-then rules and membership functions to describe the input/output behaviour of a complex system. In practical applications Sugeno type FISs have been considered most suitable for constructing fuzzy models due to their more compact and computationally efficient representation of data than the Mamdani fuzzy systems. A typical zero-order Sugeno fuzzy system has the form:

If x is A and y is B then z = c

where A and B are fuzzy sets and z is a crisply defined function. A singleton spike is often completely sufficient to cater for a given problem's needs. Alternatively a more general first-order Sugeno can be used by setting the consequent to a higher order function, for example z = px + qy + c.

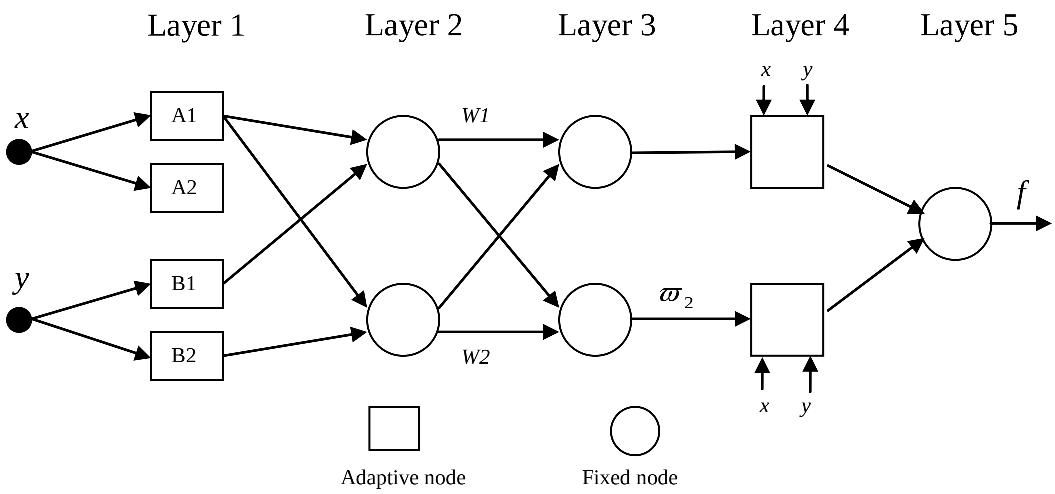

Figure 3.2: ANFIS Sugeno fuzzy system.

However a higher-order system often adds an unwarranted level of complexity because of the algorithm needed to optimise the parameters. For this reason a zero-order Sugeno FIS is used in this investigation. Figure 3.2 shows the equivalent ANFIS architecture which consists of five layers [9]. The nodes in the input layer are adaptive. Any appropriate membership functions can be used. In this experiment generalised bell-shaped membership functions were chosen to describe the input parameters because of their smoothness and concise notation. Variables x and y form input values of A1, B1 and A2, B2 respectively. A1, A2, B1 and B2 are the linguistic labels (small, large, etc.) used in the fuzzy theory for dividing the membership functions. The membership relationship between the output and input in this layer can be expressed as:

(1)

$$O_{1,j} = \mu_{B_j}(y), \quad j = 1,2 \tag{2}$$

where and represent the output functions, μ and μ_{B_j} are the membership functions.

Layer 2, sometimes referred to as the rule layer, consists of two fixed nodes which represent the fuzzy strengths of each rule. The product rules can be used to calculate the weighting function for the fuzzy operator 'AND' of a Sugeno FIS. The outputs W1 and W2 are the weight functions for the next layer. The input and output relationship in this layer is:

$$O_{2,i} = W_i = \mu_{A_i}(x)\mu_{B_i}(y) \quad i = 1,2 \tag{3}$$

where is the output of layer 2.

The third layer is the normalised layer and its function is to normalise the weight function.

$$\frac{\omega}{\omega \quad \omega} \tag{4}$$

where is the layer 3 output.

The fourth layer containing adaptive nodes is the defuzzification layer. The output from this layer is:

ϖ where , and are the consequent parameters of the node.

The input and output relationship in this layer can be defined as:

(5)

where is the layer 4 output.

The fifth layer consists of a single fixed node, it is the summation of the weighted output of the consequent parameters in layer 4. The output layer is given by:

$$\varpi \qquad \frac{\omega}{\omega} \tag{6}$$

Although any feedforward network can be used in an adaptive network-based fuzzy inference system, Jang [9] implemented a hybrid learning algorithm that converges much faster than using the gradient descent method alone. During the forward pass, the node outputs advance until the output membership function layer, where the consequent parameters are identified by the least-squares method. The backward pass uses a backpropagation gradient descent method to update the premise parameters, based on the error signals that propagate backward. Under the condition that the premise parameters are fixed, the consequent parameters determined are optimal. This reduces the dimension of the search space for the gradient descent algorithm, thus ensuring faster convergence.

Fuzzy logic and neural networks are two different kinds of mathematical tools. But in reality, there are a number of similarities. Neural networks are models of brain architecture. They have simple processing elements that collectively form a complex network structure. Fuzzy logic, however, is based on the way the brain deals with imprecise information. Fuzzy systems combine fuzzy sets with fuzzy rules to produce overall complex structure too. Both neural networks and fuzzy systems have the capability of modelling complex non-linear problems to some degree of accuracy. In light of these similarities, both fuzzy logic and neural networks are suitable for solving many similar problems. The combined techniques have been proven to improve accuracy [9]. Internal combustion engine controllers are exactly the types of problems and issues for which an AI approach appears to be most applicable and has the potential for making better, quicker and more accurate predictions than traditional methods.

Chapter 4 : Literature Survey

4.1 Intelligent techniques in Automotive Systems

AI consist of five major categories, i.e. neural network, genetic algorithms (GA), expert systems and various hybrid systems which are the combination of two or more of the categories. Expert systems are based on rule-based inference, in which previous knowledge are used to process data. Neural network are collections of individually interconnected processing nodes. Information is passed between these nodes along interconnections. The output of the node is a function of the summed value. The network is being trained with respect to the data sets to perform a specific task. Once they are trained, new patterns may be presented to them for prediction or classification.

GA are inspired by the way living organisms adapt to the harsh realities of life in a hostile world, i.e. by evolution and inheritance. The algorithm imitates in the process the evolution of population by selecting only fit individuals for reproduction. Therefore, a GA is an optimum search technique based on the concepts of natural selection and survival of the fittest. It works with a fixed-size population of possible solutions of a problem, called individuals, which are evolving in time. A GA utilises three principal genetic operators: selection, crossover, and mutation.

Fuzzy logic is a powerful way to put engineering expertise into products in a short period of time. Therefore, it is highly beneficial in automotive engineering, where many system designs involve the experience of research and development engineers as well as test drivers. In an automotive engine with electronic control, the amount of fuel that is supplied to the engine is controlled by an ECU. This is a microprocessor based system that controls the frequency and width of the control pulses supplied to the fuel injector. The AFR is important in the combustion and calibration processes. These ECUs use three-dimensional mappings (3-D maps), in the form of look-up tables, to represent the non-linear behaviour of the engine in real-time. A modern automotive ECU can contain up to 50 or more of these maps to realise complex functions. These real-time control applications and the use of 3-D maps are very common in the area of engine modelling and simulation. In addition the engine will be equipped with a wide range of sensors to gather input data for the control system. A major disadvantage of the look-up table representation is the time taken to determine the values it should contain for optimal engine operation; a process known as calibration of the ECU. These 3-D maps are typically manually calibrated or tuned, using an engine dynamometer to obtain desired levels of power, emissions and efficiency. The calibration process is an iterative one that requires many cycles of engine measurements and is very time consuming. Techniques that reduce the time and effort required for the calibration process are of considerable interest to engine manufacturers. This is especially the case where the engine is a small capacity non-automotive engine. These engines are particularly price sensitive and any additional cost, including the cost of extended calibration procedures, is likely to make the engine un-economic to manufacture. For similar economic reasons, any control strategy intended for application to a small engine has to be achievable using only a small number of low-cost sensors.

From the description of the various applications presented in this chapter, one can see that intelligent techniques have been applied in a wide range of fields for modelling, prediction and control in automotive systems. What is required for setting up such a system is data that represents the past history and performance of the real system and a selection of a suitable model. The selection of this model is done empirically and after testing various alternative solutions. The performance of the selected models is tested with the data of the past history of the real system.

Certainly, the number of applications presented here is neither complete nor exhaustive but merely a sample of applications that demonstrate the effectiveness and possible applications of intelligent techniques. Same as to all other approximation techniques, intelligent techniques have relative advantages and disadvantages. There are no rules as to when this particular technique is more or less suitable for an application. Based on some previously works and survey presented here it is believed that intelligent techniques offer an alternative method, which should not be underestimated. Automotive engineering is competitive on an international scale. A technology such as fuzzy logic that proves a competitive advantage will soon be commonly used in both automotive and non-automotive applications.

4.1.1 Braking System

Boeing developed the first mechanical antilock braking systems (ABS) in 1947 for aeroplanes. Today, ABS is standard equipment on most cars. A microcontroller and electronic sensor measure the speed of every wheel and control the fluid pressure for the brake cylinders hence regulating the braking force on each road wheel. Although mathematical models for a car's braking system exist, the interaction of the braking system with the road is far too complex to model adequately. Hence, today's ABS contains the engineering experience and knowledge of years of testing in different roads and climates.

ABS also benefits from the high computational efficiency of fuzzy logic. During a control loop time of 2-5 ms, the controllers must fetch all sensor data, pre-process it, compute the ABS algorithm, drive the by-pass valves, and conduct the test routines. Any additional function thus has to be computationally efficient. Most ABS systems use 16-bit controllers, which can compute a medium-size fuzzy logic system in about 0.5ms, using only about 2-KB-ROM of memory space [10]. Faster times can be achieved by using different microcontrollers [11].

Nippondenso, the automotive components manufacturer implemented fuzzy logic in ABS design; experiments showed that a prototype with just six fuzzy logic rules improved performance significantly [12]. On a test track alternating from snowy to wet roads, the fuzzy ABS detected the road-surface changes even during braking.

4.1.2 Transmission

The vehicle transmission is a gear system that adjusts the ratio of engine input speed to output speed. The transmission essentially enables the engine to operate within its optimal performance range regardless of the load or speed. It provides a gear ratio between the engine speed and vehicle speed such that the engine provides adequate power to drive the load at any speed. A transmission (either manual or automatic) is crucial to deliver the power from an internal combustion engine to the wheels. Maintenance of both high fuel efficiency and high performance are becoming important issues; therefore, the control of an automatic transmission has become much more complex.

Modern automatic gear changes are electronically-controlled, rather than hydraulically-controlled, as with previous conventional gearboxes. One of the advantages of electronic management is to provide a faster gearchange response. The control unit of the transmission employs fuzzy logic to determine the gear up-shift and down-shift points. Instead of having predetermined points for up-shift and down-shift, the control unit takes into account several influencing factors before deciding to shift up or down. These factors include engine speed, driving resistance or engine load, brake pedal position, throttle position, and the rate at which the throttle pedal position is changed. This results in an almost infinite number of shift points, which the control unit can tailor to match the driving style, be that sporty or economical.

4.1.3 Heating, Ventilating and Air Conditioning

Fuzzy logic design technologies are well-established in heating ventilation and air conditioning (HVAC) in buildings. Many car manufacturers also use fuzzy logic in their HVAC system design. While most car manufacturers work on these systems, very little of their work is being published. The control approach in general and hence the use of fuzzy logic in the design, differs significantly for each manufacturer.

The fundamental goal of HVAC in cars is to make vehicle occupants comfortable. Human comfort however is a complex reaction, involving physical, biological and psychological responses to the given conditions. The performance criterion ‘comfort’, is not some well-defined mathematical formula but an inconsistent and empirically determined goal. In a typical HVAC system, temperature sensors measure cabin temperature, ambient temperature, sun heating load, humidity, and the engine operating conditions. Typical actuators are variable speed blowers, means for varying air temperature, ducting and internal flaps to control the direction of air flow, and the ratio of fresh to re-circulated air. This multiple input, multiple output control problem doesn’t fall into any convenient category of traditional control theory. Figure 4.1 shows the control surface of part of an HVAC system [13]. The blower speed is determined by the temperature error i.e. the in-car temperature minus the set-point temperature, and the engine coolant temperature. It also shows how the two parameters of engine temperature and temperature error affect the blower speed.

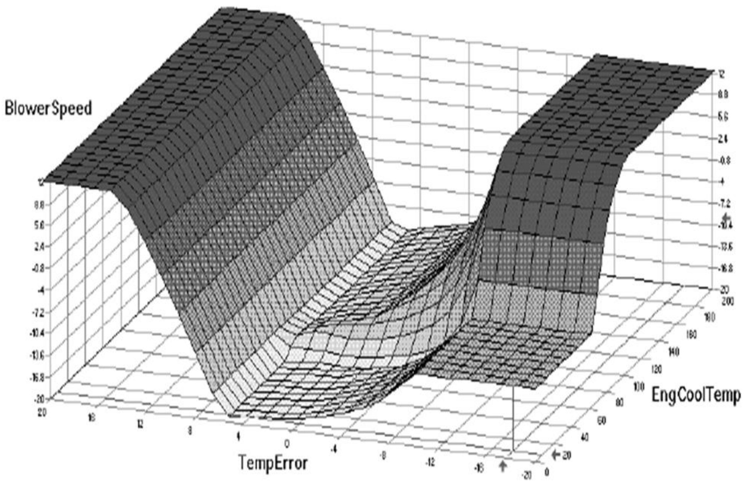

Figure 4.1: Blower control surface

The rule-base of the fuzzy logic blower speed control in Table 1 demonstrates that when the temperature error is zero, low blower speed is desired. If the in-car temperature is high i.e. positive temperature error, high blower speed is needed. When the temperature error is negative, indicating that it is too cold inside, and the engine is cold, little blower speed is needed for defrost. However, if the temperature error is negative but the engine is warm, high blower speed is needed to heat up the cabin.

Matrix	IF		THEN
Utilities	EngCoolTemp	TempError	BlowerSpeed
1		zero	low
2	high	negative	high
3		positive	high
4	low	negative	med_low

Table 1: Blower control – rule-base

4.1.4 Engine Control

The control of internal combustion engines is becoming increasingly complex with more stringent emission standards and constant effort to gain higher fuel efficiency. Twenty years ago, fuel and ignition control systems were purely mechanical i.e. carburettor, distributor and contact breaker.

Much of the research in the past has been done in the area of engine modelling [14-17]. A large portion of that research was prompted by the automotive industry in search of accurate engine models that are needed to develop sophisticated controllers. Internal combustion engines exhibit inherently non-linear characteristics under different operating conditions. These are also characterised by a time delay associated with the combustion cycle of the engine. Time delays in the engine model tend to degrade the performance of controllers by introducing overshoot and oscillations. Modern engines employ microcontroller-based systems to control fuel injection and ignition. The control strategy for an engine depends heavily on the current operating point, for example engine speed, throttle position, etc., therefore linear control models over classical methods such as PID are not suitable. On the other hand, no mathematical model fully describing the complete behaviour of an engine exists. Most engine controllers use look-up tables to present the control strategy instead. Currently, engine tuning, generation and interpretation of these look-up tables is typically performed by a skilled technician who manually adjusts the operating parameters for each zone of the engine look-up map on a dynamometer or while testing. The engine is tuned to the desired specification for each zone in which the engine is expected to be operated. This process is iterative and time consuming, creating the potential for significant time savings, additional repeatability and more accurate results via fuzzy logic control. Engine tuning is a well-known art, documented in a variety of texts on the topic of internal combustion engine performance. Prior research in the area of intelligent control of small IC engines and tuning is limited, with most research in the area of engine tuning on modifying specific engine functions to enhance performance. No references were found specifically addressing small engine control processes, this is partly because engine manufacturers are reluctant to publish any details on a fuzzy logic engine control solution. This secretiveness is due to the fact that the rules of a fuzzy logic system make the entire engine control knowledge of the company completely transparent. They are afraid competitors will learn too much about the solution by disassembling the fuzzy logic rules.

A conventional engine controller consists of a closed loop PID, and an open loop control. A feedforward controller provides the theoretical injection time, necessary to obtain a value of lambda equal to 1 at the output of the engine. It consists of a two-dimensional look-up table that contains the necessary injection times to achieve lambda equal to 1 on steady state, as a function of the throttle position and the engine speed.

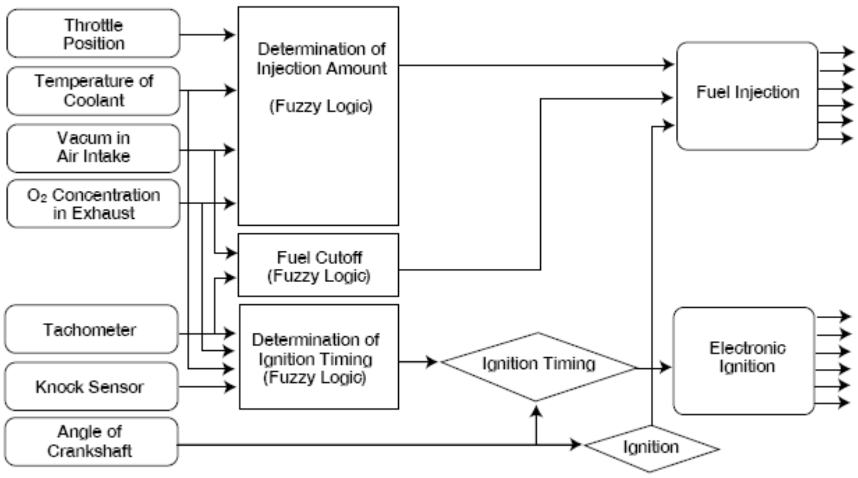

Figure 4.2: The engine controller of NOK Corporation contains three fuzzy logic modules

Fuzzy logic is known to be more robust than classical control schemes, resulting in less susceptibility to nonlinearities and modelling uncertainties, making it a viable option for control of ICEs [18]. Several pieces of work on fuzzy logic engine controllers [19-23] have demonstrated improved performance over the method of PID control. Some of this work was directed towards idle speed control [19-21] and some to the tuning of PID control parameters [22].

The major components of a fuzzy engine controller which contains three fuzzy logic modules are shown in Figure 4.2. The system first notes the engine's operational condition by the linguistic variable 'situation'. This variable has the linguistic terms in Table 2. Each linguistic term represents a typical operating point. Because each term is represented as a fuzzy logic membership function, the linguistic variable can also classify all other operation points. The determination of 'situation' is a state estimation of the operation point. Because 'situation' is a linguistic variable, more than one term can be valid at the same time, so combinations of the operating points can be expressed as defined by the terms. A possible value of 'situation' could be (0.8; 0; 1; 0; 0; 0.3). Linguistically this value represents the driving condition 'engine started a short while ago, normal drive condition at medium or low load, slightly accelerating'.

Situation 1	Start	Control strategy is that the cold engine runs smooth, ignition is timed early, and the mixture is rich
Situation 2	Idle	Control ignition timing and fuel injection depending on engine temperature to ensure that the engine runs smoothly
Situation 3	Normal drive, low/medium load	Maximise fuel efficiency by controlling mixture, monitor knocking
Situation 4	Normal drive, high load	Rich mixture and ignition advance to maximise performance, limited by the permitted emission boundaries
Situation 5	Cruising	Fuel cut-off, depending on situation
Situation 6	Acceleration	Depending on load, increasing the mixture strength

Table 2: Linguistic variable situation of a fuzzy engine controller

Similarly to ABS, engine control needs a very rapid computation speed. Some systems are as fast as 1ms for an entire control loop. Some manufacturers design the system using fuzzy logic but then translate it into a look-up table for faster processing. Although a look-up table computes rapidly, memory requirements may prohibit its use. A look-up table with two inputs and one output, all of 8-bit resolution, already requires 64KB of ROM. Restricting the resolution of the input variables to 6-bits each, the look-up table still requires 4KB. A table with three inputs and one output, all inputs 6-bit resolution, requires 250KB. Some researchers have implemented a look-up table with a limited resolution and used an interpolation algorithm; however, the interpolation needs about as much computing time as the fuzzy logic system itself.

4.1.5 Development Tools for Fuzzy Systems

Fuzzy development software is a collection of functions built in a fuzzy computing environment. It provides tools to create and edit fuzzy inference systems within a framework. Functions are provided for many common fuzzy logic methods, including fuzzy clustering and adaptive Neuro-fuzzy learning. The software provides a convenient facility for modelling complex system behaviours using simple logic rules, and then implements these rules in a fuzzy inference system. Most of these tools can be used as a stand-alone fuzzy inference engine. Some even can be used in a simulation environment and these simulate the fuzzy systems within a comprehensive model of the entire dynamic system.

4.1.5.1 Fuzzy Development Environment

The Fuzzy Development Environment (FDE) was developed in-house by the Intelligent Systems & Signal Processing Laboratories at the University of Brighton. The software enables an FIS to be developed and evaluated at the same time. The import and export functions allow for connection to machines and for developing web-based applications. Figure 4.3 shows a sequence of screenshots of the FDE. The top figure illustrates an example of a FIS with two input fuzzy sets and one output fuzzy set, this is a process of formulating the mapping from a given input to an output using fuzzy logic. The figure on the right shows a selected input fuzzy set, the graphic editor facilitates the comprehension of membership functions whilst the rule-base editor allows for the creation of if-then rules. This software has been used extensively during the course of this project and has proven extremely effective and convenient in developing dedicated FIS.

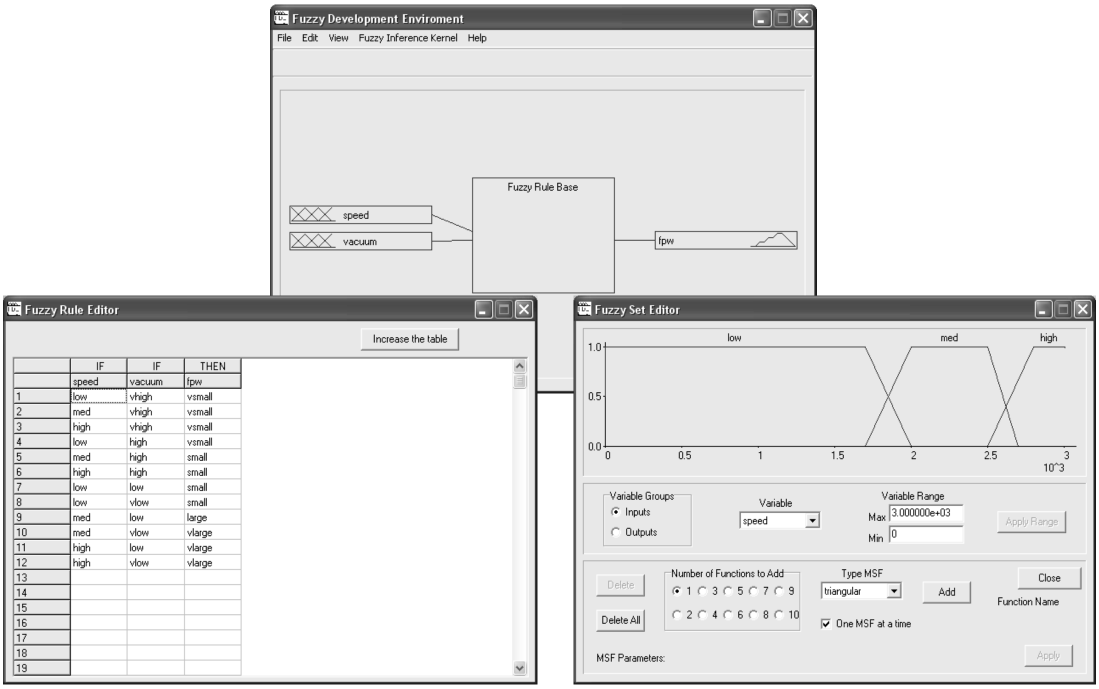

Figure 4.3: The Fuzzy development environment

4.1.5.2 Matlab/Simulink

Similar to the FDE, the fuzzy logic toolbox in Matlab developed by Mathworks can be customised in its own developing environment. It also offers the facilities to inspect algorithms, modify source code, and add customised membership functions or defuzzification techniques. Some of its key features include specialised graphical user interfaces for building fuzzy inference systems, viewing and analysing results; provide membership functions for creating fuzzy inference systems; support for: AND, OR, and NOT logic in user-defined rules; automated membership function shaping through neuro-adaptive and fuzzy clustering learning techniques; ability to embed a fuzzy inference system in a Simulink model and generate embeddable C code or stand-alone executable fuzzy inference engines. The package has a wide selection of dynamic systems for modelling, analysing and simulation. It offers a user interface for creating block diagram models. A system is configured in terms of block diagram representation from a library of standard components. During the course of the simulation, algorithms and parameters can still be changed to get intuitive results, thus providing the user with a readily accessible knowledge tool for simulating many of the operational problems found in the real world. It also provides immediate access to the mathematical, graphical and programming capabilities.

4.2 Embedded Control Design

Embedded control design has been widely applied in automotive systems such as vehicle and engine control. Such a system is dedicated to specific tasks; design engineers can optimise, reducing the size and cost of the product, and/or increasing the reliability and performance. Some embedded systems are mass-produced, benefiting from economies of scale. Typically, several complex algorithms are running in this embedded system most of them are based on operating information from the vehicle. Thus a reliable estimation of running parameters is very important in determining that the control regime consequently improves vehicle performance. The vehicle mass, road gradient and air drag are essentially the major factors which influence a vehicle’s performance. These parameters are experimentally significant in the case of buses due to their weight and particularly large frontal area. Many modern vehicle control systems consist of engine, transmission, brake and auxiliary functions. There are large numbers of dedicated algorithms in these sub-systems ranging from pure control tasks to running resistance estimation. To develop such a system with precision requires time and knowledge, increasing complexity in some systems making design and development of such system even more difficult. Hardware-in-the-Loop (HIL) simulation is a technique that is used in the development and testing of complex real-time embedded systems. HIL simulation provides an effective platform by adding the complexity of the system under control to the test platform. The complexity of the system under control is included in test and development by adding a mathematical representation of all related dynamic systems.

4.3 GPS-based Engine/Vehicle Control

The use of externally acquired information such as GPS data is believed to be useful in engine and vehicle control. It has been increasingly used in real-time tracking of vehicles, especially when GPS is integrated with ever increasingly powerful Geographic Information System (GIS) technologies. The accuracy and reliability of low-cost, stand-alone GPS receivers can be significantly improved to meet the technical requirements of various transportation applications of GPS, such as vehicle navigation, fleet management, route tracking, vehicle arrival/schedule information systems (bus/train) and on-demand travel information. Systems that were previously only intended for fixed installation in vehicles are gradually being replaced on the market by portable systems that require no connection to the vehicle other than the power supply. To an increasing extent, GPS navigation is becoming a software product that can also be installed on handheld computers, laptops and mobile phones.

Global positioning determination is based primarily on the use of GPS. Stand-alone systems, such as handheld computers, use this exclusively, whereas fixed installation systems also run ‘dead reckoning’ if they have additional in-vehicle sensors. Dead reckoning ensures exact position determination even if no GPS signals can be received, e.g. in tunnels. To measure the distance travelled, all that is needed is a speedometer output signal. The change of direction is ascertained by a rotation rate sensor or gyroscope. Hence, the absolute direction of travel can be determined by the Doppler effect of the GPS signals [24]. The levels of accuracy that can be achieved is in the range of 3 to 5m, and 10 to 20m in the case of measuring altitude relative to sea level. With the autonomous European Satellite Navigation System Galileo, an opportunity of a joint system ‘GPS + Galileo’ with more than 50 satellites will provide many advantages for civil users and vehicle systems, in terms of availability, reliability and accuracy [25]. Vehicle drive cycle simulation model incorporating vehicle telematics systems can provide vehicle control system with the information of the road topography [26]. The study demonstrated an effective approach for design and analysis of powertrain. It is shown that simulation is easy to set up and provided consistent results. The simulation of the drive cycle and road profile was useful especially in situation when developing engine and powertrain control strategies. In addition, the use of telematics could cut vehicle running costs up to 10% according to the report by Zurich Financial Services Group. The company examines how a combination of technology and driver-development programmes can help to improve vehicles' safety, reduce their effect on the environment and cut their running cost. The study showed a vehicle which integrate GPS sensors with wireless communications not only can give companies vital information on location, behaviour and performance of the drivers but also save up to 11% on fuel bills and 10% on operation costs [27].

Future GPS may not only be used to guide the vehicle but information from the system may also be used to control or influence the engine, through given control parameters in a safe and cost-effective manner. A GPS receiver provides reliable reference position data which can be manipulated to provide more significant road information such as gradients or even road traffic congestion updates when it is combined with the vehicle telematics. It is a technology integrated with computers and mobile communications technology in vehicle navigation systems. This information can be used to not only inform the driver but also to enhance the control of several systems of the vehicle. Ultimately, for example, the vehicle speed, gear selection and even the application of brakes could be appropriately chosen and strategically designed. The idea is to provide the control system with this essential information that the driver normally uses when driving. Good driving requires consideration of several inputs; it can be a complex, exhausting and demanding task, even for commercial vehicle drivers and thus supporting control functionality is of great interest. It is believed to be even valuable to obtain road information beyond the line of sight of the driver. Whilst all of these driving decisions have to be made manually by the driver in the interest of comfort and fuel efficiency, the newly intelligent vehicle controller aims to address these tasks.

This work, in pursuit of sustainable transportation, will lend to dramatically reduced fuel consumption and emissions. This project has addressed the issues and challenges imposed by legislation and guidelines with the aim of facilitating the reduction of exhaust emissions and fuel consumption through precise control of the vehicle.

Techniques include the fusion of data from sources that are external as well as internal to the vehicle; also from analysis of these data using special intelligent systems techniques and tools. The resulting system essentially used a fuzzy logic derived relief map of the test route, and this was further validated and corrected based on the past trajectory from the GPS sensor. The information was then processed and translated in order to estimate the future elevation of the vehicle. Similar techniques based on predictive parameters have been proven useful and achieved better results. Model Predictive Control (MPC) is an optimisation algorithm which has shown that a 2.5% reduction of fuel consumption can be achieved by controlling the speed of a vehicle. The control signals were; percentage of throttle opening, activation of brakes and gear selection. The control algorithm there was tuned and optimised according to some criteria, e.g. the main issues were to minimise costs, time and fuel consumption [28]. Similar work described in another publication has been designed and simulated on cruise control [29]. The simulation showed that a reduction of fuel consumption in the range of 1.5 to 3.4% was achieved. It used a dedicated logic in a finite number of simulated driving situations, given that the topography of the road such as gradient was a known input to the system. Control of the vehicle powertrain has been undertaken by DaimlerChrysler; the research suggested usage of a three-dimensional digital road map in order to let the cruise control replicate a skilled driver [30]. A reduction in fuel consumption of 4.1 to 5.2% was attained. Furthermore, cruise control has now been incorporated with radar technology to record the distance and speed relative to the vehicle in front as well as additional data such as position of other vehicles in the vicinity. The system used such information to regulate the time gap between vehicles. The interface was developed under the framework of the European project MAPS&ADAS to obtain the map data from the on-board data provider [30]. This is a predictive system which adapts the speed to the surrounding vehicles and keeps a safe distance.

All in all, a number of approaches have been researched. A substantial amount of work has been carried out on how the interface between vehicle control system and the GPS system should be designed. The investigation was focused on information retrieval and processing. Location data could be available to the vehicle control unit in a variety of formats, resolutions and temporal accuracies. Data processing and fusion forms the main part of this project. This information was made available and able to combine with other sensory data of the vehicle.

The simulation model in Chapter 6 generated using Matlab/Simulink showed the effectiveness of the system. Simulink has a wide selection of dynamic systems for modelling, analysing and simulating. It offers a graphical user interface for creating block diagram models. A system is configured in terms of block diagram representation from a library of standard components. In the middle of a simulation, algorithms and parameters can still be changed to get intuitive results, thus providing the user with a readily accessible learning tool for simulating many of the operational problems found in the real world. It also provides immediate access to the mathematical, graphical, and programming capabilities of Matlab. The effective engine/vehicle control system devised using Simulink could potentially be used in vehicle control for reduced fuel consumption and emissions.

Chapter 5 : Engine Control and Modelling

The experimental work was conducted to determine the performance and characteristic of two different fuzzy-derived techniques for controlling small combustion engine and modelling fuel spray penetration in the cylinder of a diesel internal combustion engine.

The first study was to derive a cost-effective fuzzy control system applied to a small spark-ignition internal combustion engine to achieve regulation of the fuel injection system. The control system determined the amount of fuel required from a fuzzy algorithm using engine speed and manifold air pressure as input values which has led to improved fuel regulation, and a consequent reduction in exhaust emissions. The aim of the second study was to demonstrate the effectiveness of an ANFIS for the prediction of diesel spray penetration length in the cylinder of a diesel internal combustion engine. The technique involved extraction of necessary representative features from a collection of diesel engine spray data. A comparative evaluation of two fuzzy-derived techniques for modelling fuel spray penetration was also described.

5.1 Engine Fuel Injection Control using Fuzzy Logic

A fuzzy control system (FCS) was applied to a small engine to achieve regulation of the fuel injection system. It was demonstrated that intelligent systems can be used for the computer control of the fuel supply of a small internal combustion engine. The technique represented a convenient and quick method of achieving engine control unit (ECU) calibration; conventionally, ECUs use three-dimensional mappings (3-D maps), in the form of look-up tables, to represent the non-linear behaviour of the engine in real-time. As discussed, the major disadvantage of the look-up table representation is the time taken to determine the values it should contain for optimal engine operation; the calibration process is an iterative one that requires many cycles of engine measurements.

5.1.1 Fuzzy Feedforward Control

The aim of the control strategy here was to govern the value of AFR in the engine, keeping it at a desired optimal value, and minimising the influence of changes in speed and load. Engine load was estimated indirectly by measurement of the inlet manifold air pressure (MAP). The parameters of the FCS and rule-base contents in the FCS were determined during test-rig trials and implanted as a system reference into the control unit. The details of the creation of such a control algorithm for this experiment are explained in the next section of the paper. The minor drawback of this feedforward control is lack of feedback information; factors such as wear and spark plug deterioration will detract from optimum fuel injection quantity in what is still effectively an open-loop system. Feedback control of AFR is often provided in automotive engines, but this is seldom economic on small engines.

A suitable model was created to predict throttle position by using the MAP and the engine rotational speed. The feedforward fuzzy control scheme was used in order to reduce deviations in lambda value. The scheme also has the benefit of reducing the sensitivity of the system to disturbances which enter the system outside the control loop. This fuzzy model offers the possibility of identifying a single multi-input single-output non-linear model covering a range of operating points [31].

Figure 5.1 shows the block diagram of the test system. The system determines the amount of fuel required from a fuzzy algorithm that uses the engine speed and MAP as input values. These input variables were converted to digital form and the crisp digital signals were then applied to a fuzzy algorithm implemented in the C programming language on a PC. The crisp output from the algorithm was equal to the width of the pulse applied to the fuel injector, the fuel pulse width (FPW). The parameters of this fuzzy control paradigm were a collection of rules and fuzzy-set membership functions. These were intuitively comprehensible by the operator. This facilitated the calibration process, leading to relatively quick and convenient tuning.

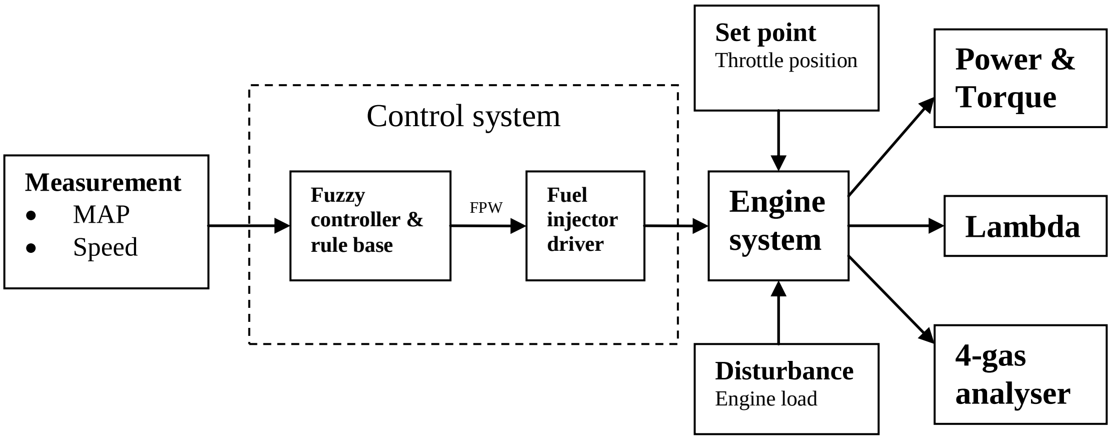

Figure 5.1 Block diagram for fuzzy logic control scheme

5.1.2 Engine Load Estimation

In a spark-ignition engine the induction manifold pressure varies with engine speed and throttle opening according to a non-linear mapping. Figure 5.2 shows the three dimensional relationship between these operating parameters for the Bosch Suffolk engine. By measuring these two variables, the engine load/throttle position can be determined. A conventional look-up table can be used, although in the case of this work fuzzy logic was used to represent the non-linear relationship between functions.

5.1.3 Fuzzy Control Algorithm

The FCS was devised using the Fuzzy Development Environment (FDE). Although the commercial software, Matlab provides a similar facility in creating and evaluating fuzzy systems, the FDE provided a better developing platform for this work. Fuzzy sets, membership functions and rule sets for this project were all created, and modified where required, using the FDE. Parameters derived from the FDE, specific to the particular set-up devised, were transferred to an execution module, known as the Fuzzy Inference Kernel (FIK). The FIK is a module programmed in C++ code. To make it possible to embed the FIK directly into an ECU, the code was compiled to .obj format, and incorporated into the rest of the control code by the linker.

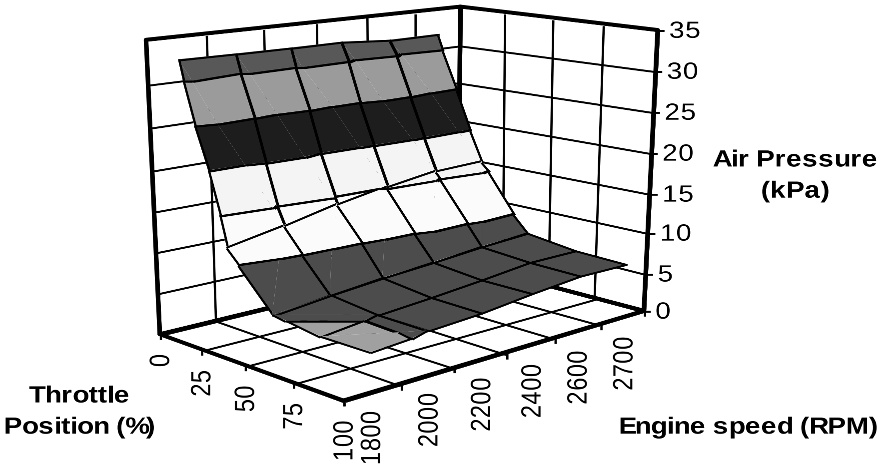

Figure 5.2: Variation of MAP with speed and throttle opening

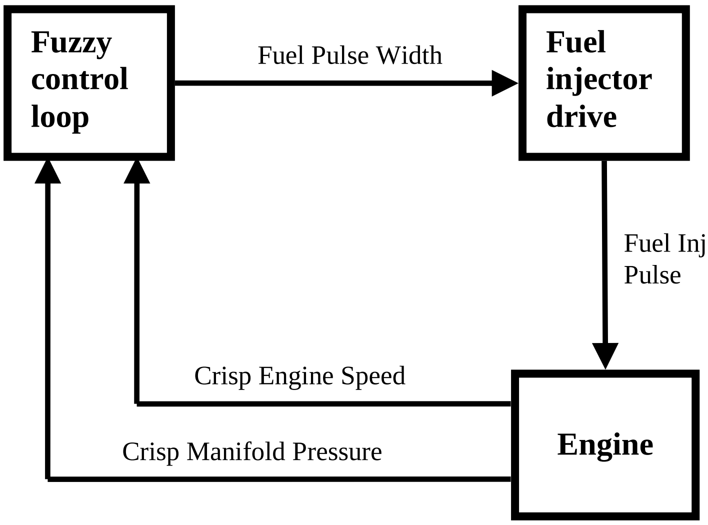

Figure 5.3: Air-fuel ratio fuzzy control loop

The fuzzy control loop illustrated in Figure 5.3 was implemented in order to optimise the AFR. To determine the effectiveness of the control loop, the AFR was monitored using a commercial instrument, a Horiba Lambda Checker. The engine speed was determined by an optical sensor while the MAP was measured by a pressure sensor located in the intake manifold. These instruments sampled individual parameters and through the medium of signal conditioning circuitry provided analogue output voltage levels proportional to their magnitude. These were converted to digital form and the crisp digital signals were then applied to a fuzzy algorithm implemented in the C programming language on a PC. The crisp output from the algorithm was the width of the pulse applied to the fuel injector (the FPW).

The fuzzy sets shown in Figures 5.4 and 5.5 were used in the fuzzy controller. The engine speed fuzzy set used three trapezoidal membership functions for classes low, medium and high. The MAP fuzzy set consisted of four trapezoidal membership functions for classes Very Low, Low, High, Very High. Experimental adjustment of the limits of the membership classes enabled the response of the control kernel to be tailored to the physical characteristic of the engine.
The optimisation of the rule-base and the membership functions underwent experimental refinement as part of the calibration process. The final set of rules contained in the rule-base is shown in Figure 5.6.

The fuzzified values for the outputs of the rules were classified into membership sets similar to the input values. An output membership function of output singletons, illustrated in Figure 5.7, was used. This was defuzzified to a crisp value of FPW.

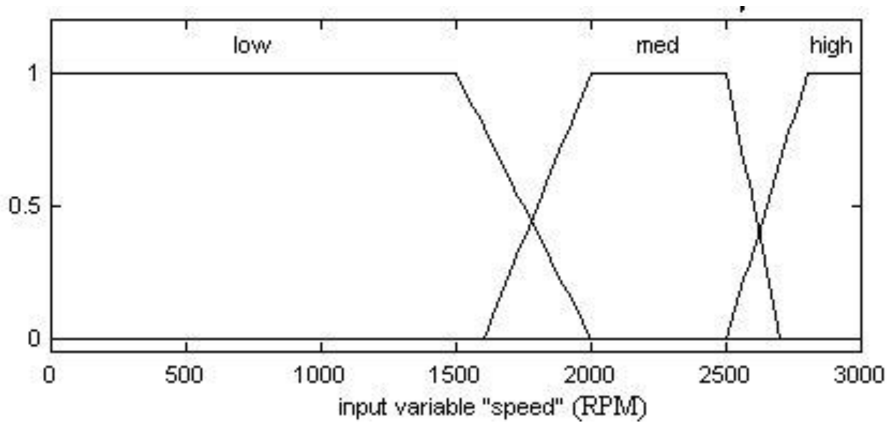

Figure 5.4: Fuzzy input set – engine speed

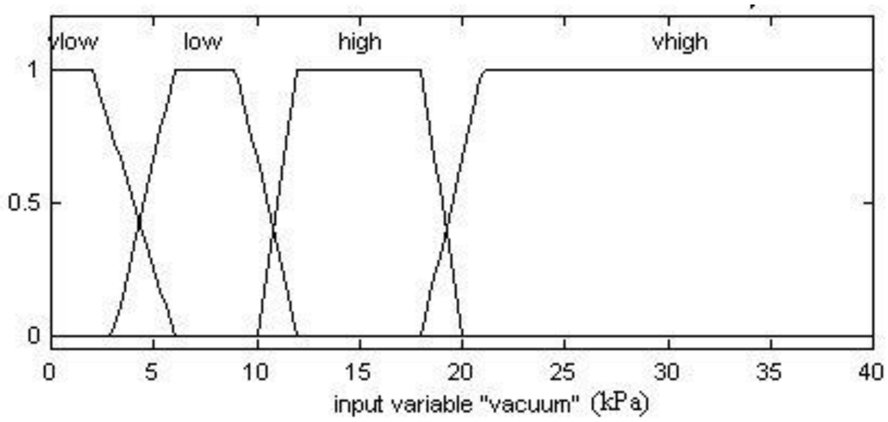

Figure 5.5: Fuzzy input set – vacuum pressure

Rule 1:	If the speed is low and the vacuum is very high then fuel pulse width is very small
Rule 2:	If the speed is medium and the vacuum is very high then fuel pulse width is very small
Rule 3:	If the speed is high and the vacuum is very high then fuel pulse width is very small
Rule 4:	If the speed is low and the vacuum is high then fuel pulse width is very small
Rule 5:	If the speed is medium and the vacuum is high then fuel pulse width is small
Rule 6:	If the speed is high and the vacuum is high then fuel pulse width is small
Rule 7:	If the speed is low and the vacuum is low then fuel pulse width is small
Rule 8:	If the speed is low and the vacuum is very low then fuel pulse width is small
Rule 9:	If the speed is medium and the vacuum is low then fuel pulse width is large
Rule 10:	If the speed is medium and the vacuum is very low then fuel pulse width is very large
Rule 11:	If the speed is high and the vacuum low then fuel pulse width is very large
Rule 12:	If the speed is high and the vacuum is very low then fuel pulse width is very large

Figure 5.6: The fuzzy rule base

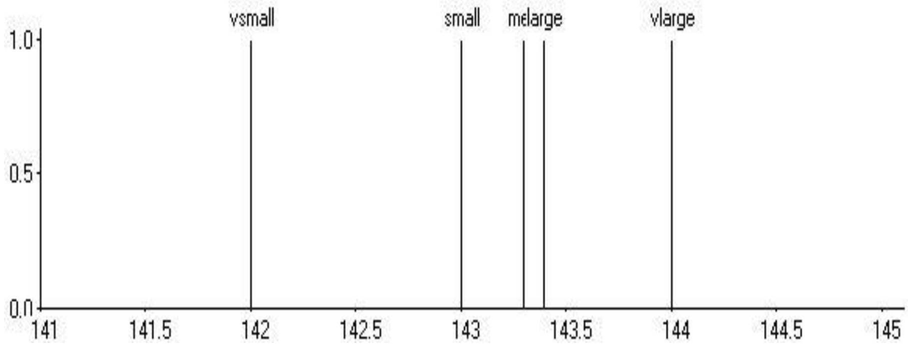

Figure 5.7: Fuzzy output set – FPW (ms)

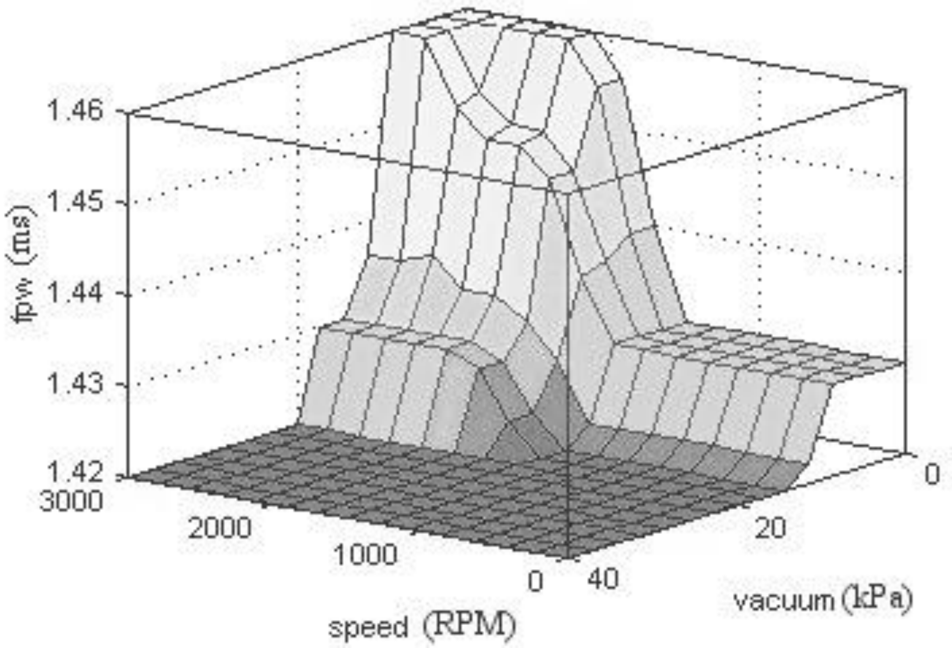

Figure 5.8: Three-dimensional FCS map

5.1.4 The Mapping

Engine control typically requires a two-dimensional plane of steady state operating points with engine speed along the horizontal axis and throttle position along the vertical axis. The control surface in Figure 5.8 shows the crisp value of FPW at different combinations of speed and vacuum pressure using FCS. Each of these intersection points indicates the differing requirement for fuel, which is determined by the design of fuzzy sets and membership functions. The control surface acts as a means of determining the FPW needed for each combination of speed and MAP value.

5.1.5 Results and Discussion

Figures 5.9 and 5.10 illustrate that the power produced by the engine with the FCS exhibited an increase of between 2% and 21% with an average of approximately 12% compared with the original mechanical fuel delivery system.

A corresponding improvement in output torque also resulted from the use of the fuel injection system with the FCS compared to when the original fuel delivery system was used. With its injector located at the entrance to the air intake just ahead of the throttle butterfly-value. Air/fuel mixing and atomisation take place at low injection pressure upstream of the throttle. The carburettor system however relies heavily on negative pressure created by the venture to induce the metered fuel to enter the incoming air stream, the pressure drop in a carburettor also impairs the volumetric efficiency of an engine and reduces its power output whereas the FCS injected fuel into an unrestricted air stream above atmospheric pressure. Figures 5.11 to 5.12 show that the average torque exhibited an increase of between 2% and 20% with an overall average of 12%. These increases in engine performance are partly due to the improvement in charge preparation achieved by the fuel injection process; the improvement in fuel metering also results in improved combustion efficiency hence increased engine power.

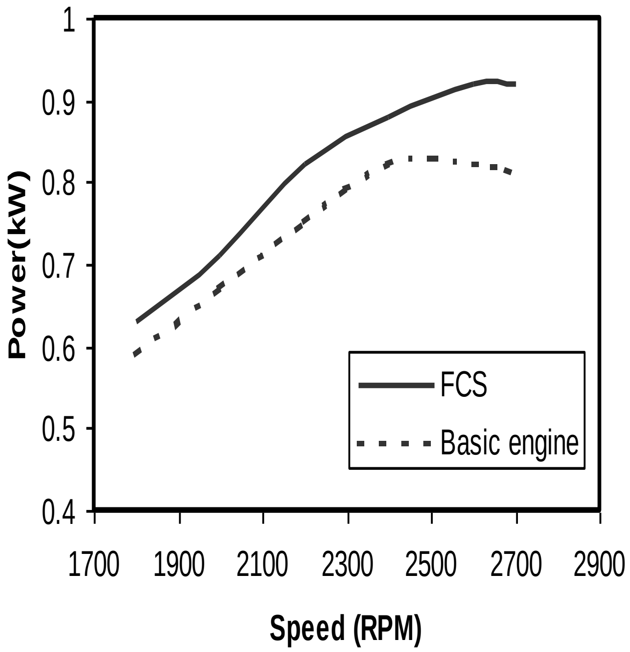

Figure 5.9: Engine power when throttle=50%

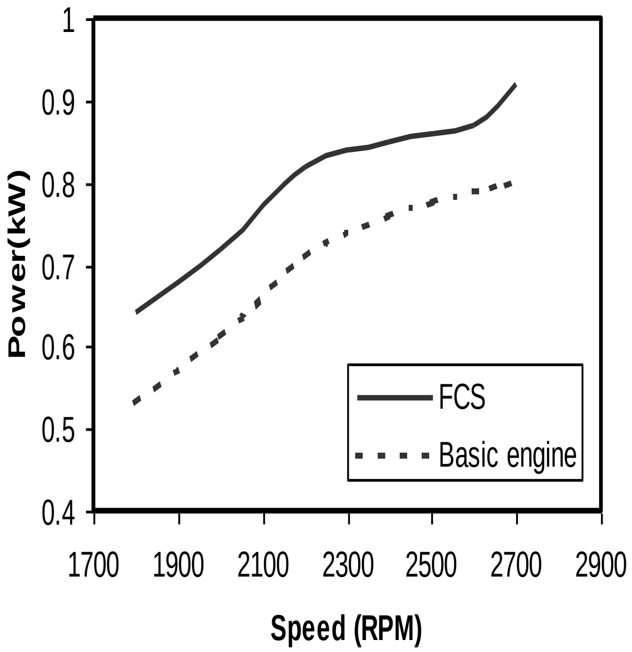

Figure 5.10: Engine power when throttle=75%

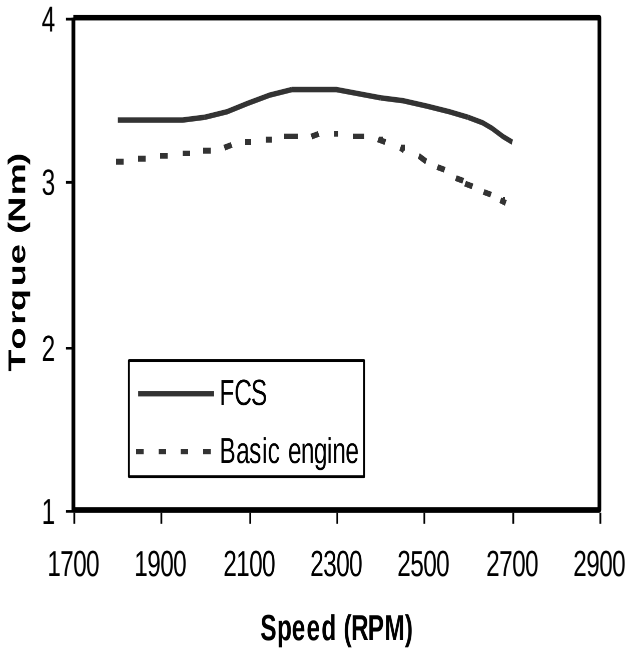

Figure 5.11: Engine torque when throttle=50%

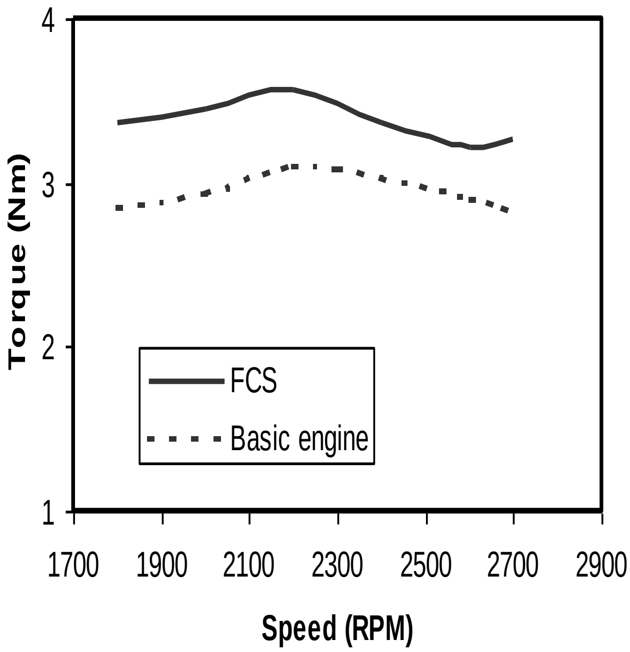

Figure 5.12: Engine torque when throttle=75%

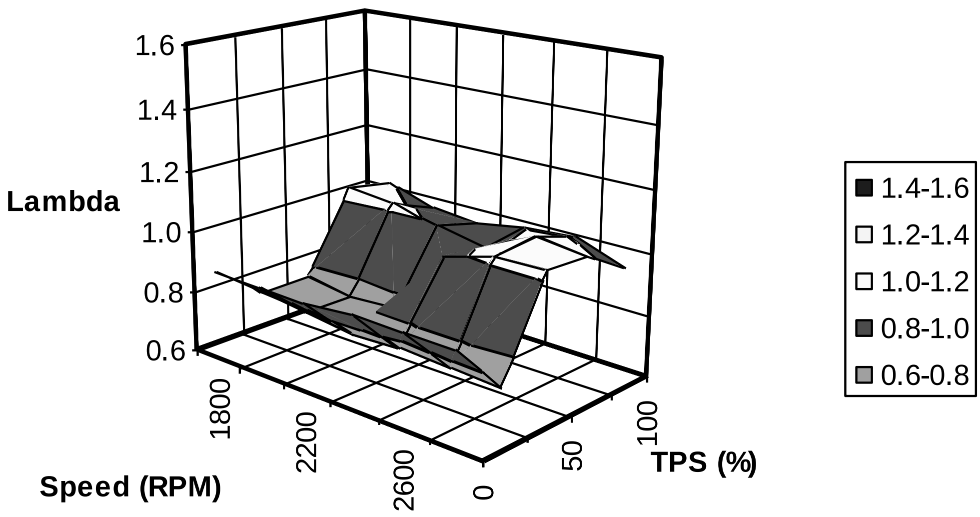

Figure 5.13: Variation in lambda with original fuel regulation system

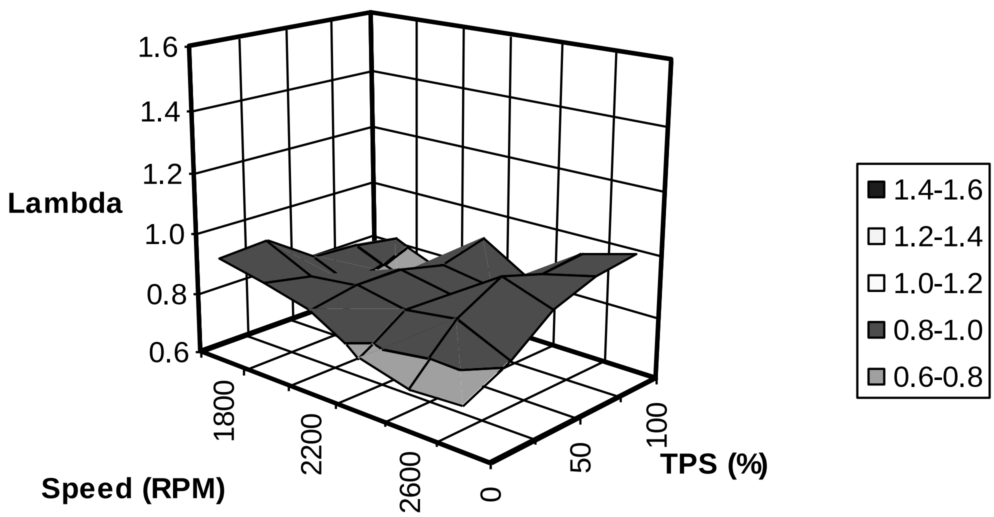

Figure 5.14: Variation in lambda with fuzzy-controlled fuel-injection system

Figures 5.13 and 5.14 illustrate how the value of λ varied with different combinations of speed and throttle position using the original fuel regulation system and the fuzzy-controlled fuel-injection system, respectively. Figure 5.13 shows that wide variations in λ occurred when the original fuel regulation system was used, this being due to non-linearities in the characteristic of the carburettor. This resulted in an excessively rich mixture at small throttle openings and an excessively weak mixture when the throttle opening was large. The large variations in λ suggested poor combustion efficiency and higher, harmful, exhaust emissions. An improved and refined contour was found to occur when the FCS was employed, Figure 5.14. Reasonable regulation of λ was achieved, the value being maintained between 0.8 and 1.0 in approximately 90% of the experimental operating region.

The complexity of such a fuzzy logic system with a fixed input-output structure is determined by the number of membership functions used for the fuzzification and defuzzification and by the number of inference levels. Clearly a fuzzy system of this kind requires that a knowledgeable human operator initialises the system parameters e.g. the membership function bounds. The operator must then optimise these parameters to achieve a required level of accuracy of mapping of the physical system by the fuzzy system.

5.2 Fuzzy and Neuro-fuzzy Modelling

This section describes another linked piece of work on fuel spray penetration in the cylinder of a diesel internal combustion engine. The aim is to demonstrate the effectiveness of an ANFIS for the prediction of diesel spray penetration length in the cylinder of a diesel internal combustion engine.

The first model was implemented using a conventional fuzzy-based paradigm, where human expertise and operator knowledge were used to select the parameters for the system. The second model used an ANFIS, where automatic adjustment of the system parameters was effected by a neural network, based on prior knowledge.

A large collection of diesel spray data was generated using a Ricardo Proteus test engine. These data comprised images depicting the spray patterns of diesel injection processes, under selected conditions of relative pressure, nozzle size and type and in-cylinder air temperature. The images representing time-varying spray patterns under each relative pressure condition were examined and processed using a thresholding technique; each image representing the instant of maximum penetration length was then determined, yielding a maximum penetration value which could be linked with its corresponding relative pressure across the injector. The collected maximum spray penetration values and corresponding relative pressures then formed labelled data to be modelled by the FIS.

Two engine operating parameters were used as inputs to the model; namely in-cylinder pressure and air density. Spray penetration length was modelled on the basis of these two inputs. The models derived using the two techniques were validated using test data that had not been used during training; please see Figure 5.15.

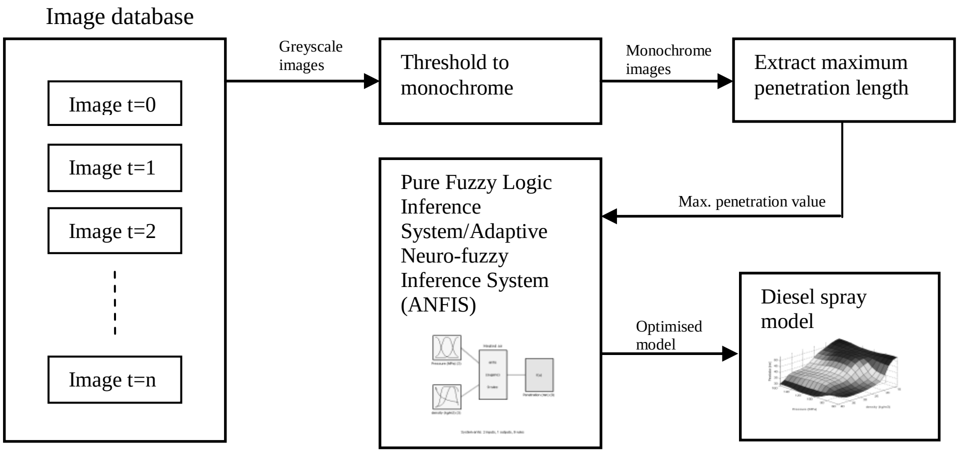

Figure 5.15: Schematic diagram of FIS modelling

The generalised bell-shaped membership functions were used for classes: low, medium and high; this was empirically selected, based on the features of all data under consideration, although in many cases membership functions are fixed and somewhat arbitrarily chosen. The process was carried out by examining the ranges of all data sets to determine where the majority of points were located. The functions were also created to have an approximately equal amount of overlap between each membership curve. Experimental adjustment of the limits of the membership classes enabled the response of the model to be tailored to the experimental output from the experimental data. The rule structure was essentially predetermined by the user's interpretation of the characteristics of the input parameters in the model. The contents of these rule-base and membership functions undertake many modifications as part of the process of heuristic optimisation and in many cases it is a continuing process.

A second model was devised using Matlab based application, ANFIS. A neuro-adaptive learning technique facilitated the learning of dataset information by the fuzzy modelling procedure; it was then possible to compute the membership function parameters that best allowed the associated FIS to track the given input/output data, rather than choosing the parameters associated with a given membership function arbitrarily.

5.2.1 Pre-processing

Raw penetration lengths were plotted against time under each relative pressure and density condition. Polynomial fitting was employed to produce best fitted curves where maximum penetration values can be depicted. As an example, Figure 5.16 shows a selected plot when relative pressure is 60MPa and air density is 14kg/m^3.

Figure 5.16: Selected polynomial curves fitting & maximum spray penetration seeking

Various values of relative pressure and density were selected and the resulting maximum penetration was recorded. These were combined into a vector with which were used in the training of the ANFIS, as illustrated in Table 3. During training in ANFIS, 6 sets of pre-processed data were used to conduct 180 cycles of learning.

Data set	Parameters		Measured penetration (mm)
	Relative pressure (MPa)	Density (kg/m3)	
1	60	14	53
2	60	35	32
3	100	14	52
4	100	35	38
5	160	14	54
6	160	35	36

Table 3: Training data sets and results

Figure 5.17: Pure fuzzy logic model – surface plot

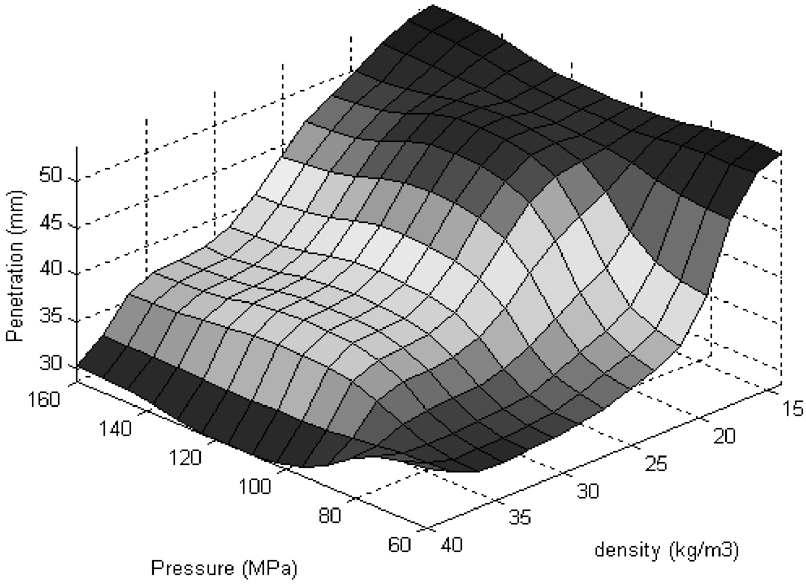

Figure 5.18: ANFIS surface plot

The control surface in Figure 5.17 shows the crisp value of penetration depth at different combinations of in-cylinder pressure and air density, using a pure fuzzy logic model. Each of these intersection points indicates the differing predicted value of spray penetration depth, which is determined by the design of the fuzzy sets, the rule-base and the membership functions. The surface plot acts as a practical means of determining the output needed for each combination of input parameters.

Figure 5.18 depicts a three-dimensional plot that represents the ANFIS mapping from relative pressure and air density to spray penetration length. As the relative pressure and air density increase, the predicted penetration length increases in a non-linear piecewise manner, this being largely due to non-linearity of the characteristic of the input vector matrix derived from the raw image data. This assumes that these raw image data are fully representative of the features of the data that the trained FIS is intended to model.

5.2.2 Results and Discussion

The validation data in Table 5 was used as checking data to see how well the FIS model could predict the corresponding penetration length. Figure 5.19 shows a scatter plot of the measured and FIS-modelled penetration lengths utilising six sets of testing data. These two sets of data demonstrate that the predicted values are close to the experimentally-measured values, as many of the data points fall very close to the diagonal (dotted) line, indicating good correlation. Figure 5.20 shows similar comparisons between the FIS-modelled and measured values of the penetration length using the same testing data. Clearly the model created by ANFIS has a better agreement than the pure fuzzy logic model. The correlation coefficient also suggested identical findings.

Data set	**Parameters**		**Penetration (mm)**		
	Relative pressure (MPa)	*Density (kg/m3)*	*Measured*	*Pure Fuzzy Paradigms*	*ANFIS*
1	60	28	33	30	33
2	60	40	35	28	35
3	100	28	40	40	41
4	100	40	29	23	29
5	160	28	40	39	40
6	160	40	30	21	30
		Correlation coefficient		0.971	0.997

Table 4: Testing data

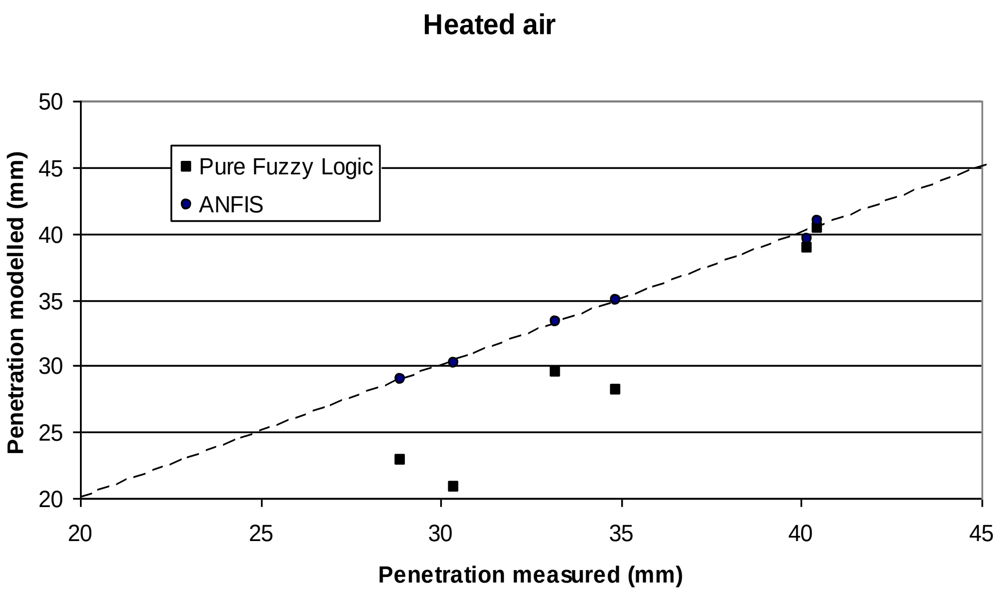

Figure 5.19: Scatter plot of measured penetration and predicted penetration

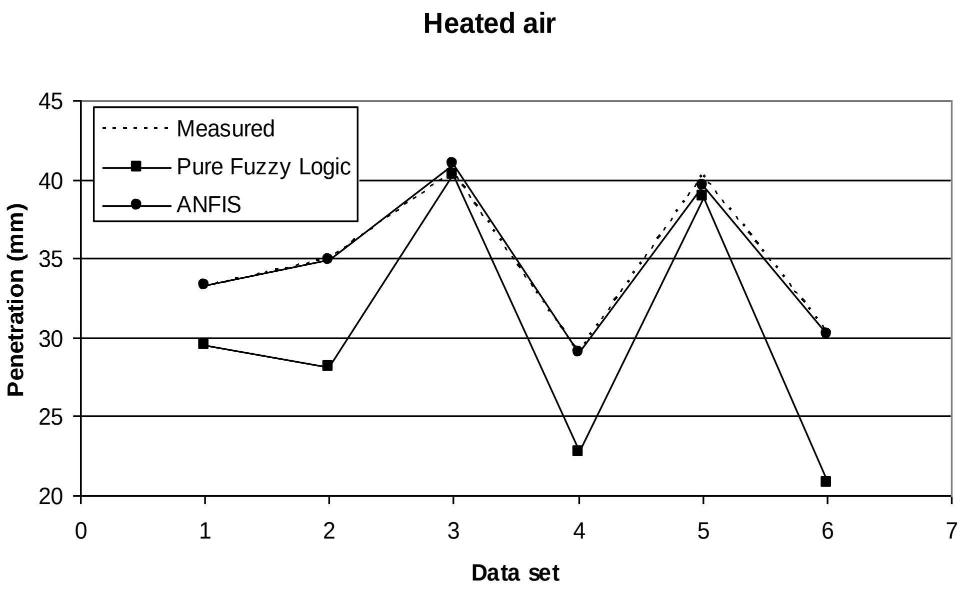

Figure 5.20: Comparisons between predicted and measured penetration

This technique ANFIS has several advantages when assigned to applications in which only partial knowledge of the system characteristic exists, as is typically the case with engineering systems. Additionally, the ANFIS can rapidly identify important characteristics of the data, which is an important and useful feature of models used for estimation purposes in internal combustion engines research. In the experiment, an ANFIS was used to predict changes in diesel spray penetration depth as a potential means to evaluate impending changes in combustion chamber and fuel injector design. As an initial step toward modelling and prediction with an ANFIS for this particular application, it has proven very useful for short-term prediction of penetration depth using engine operating parameters as the input. Both models performed fairly well and approximated the output function to a reasonable extent whilst the ANFIS model exhibited improved performance in this respect. Pure fuzzy logic models were conveniently constructed whilst the ANFIS performed well in cases where the input to output relationships became more complex.

Chapter 6 : Experimental Work

This chapter describes an intelligent GPS based control system utilising information about the current vehicle position and upcoming terrain in order to reduce vehicle fuel consumption as well as improve road safety and comfort. The development of the in-vehicle control systems has provided static and dynamic road information. The vehicle running parameters have been mathematically defined whilst the engine control algorithms were derived from a custom-built engine test-rig. As the vehicle travelled over a particular route, road information such as gradient and position was stored with the past trajectory using a Neuro-fuzzy technique. This road information was continuously updated and replaced by new data as the vehicle moved along, thereby adjusting the engine control parameters to reflect the actual current vehicle running data.

6.1 Vehicle Model

The model developed using Simulink was based on a Volkswagen Golf with a standard gasoline-powered internal combustion engine. This model has formed a core part of the simulation, intending to predict the amount of torque required to balance the loads exerted on the vehicle. This model consisted of two major input components, the engine speed and the predictive gradient of the road. The outputs from the model are the ignition timing and the engine torque. The ultimate objective here is to control the ignition timing so that the engine is operated in its optimised condition. Ignition timing in an internal combustion engine is the process of setting the time that a spark will occur in the combustion chamber, relative to piston position and crankshaft angular velocity. Setting the correct ignition timing is crucial in the performance of an engine. The ignition timing affects many variables including engine longevity, fuel economy and engine output.

6.1.1 Vehicle dynamics

Consider a vehicle with a mass ‘m’ travelling on a road with an incline ‘θ’ shown schematically in Figure 6.1. The resultant force acting on the vehicle was simplified to the sum of the driving force generated by the engine, the air drag and the gravitational force. The rolling resistance between the road and the tyres was assumed to be zero. The car was travelling in a straight line and must maintain constant speed with change in θ.

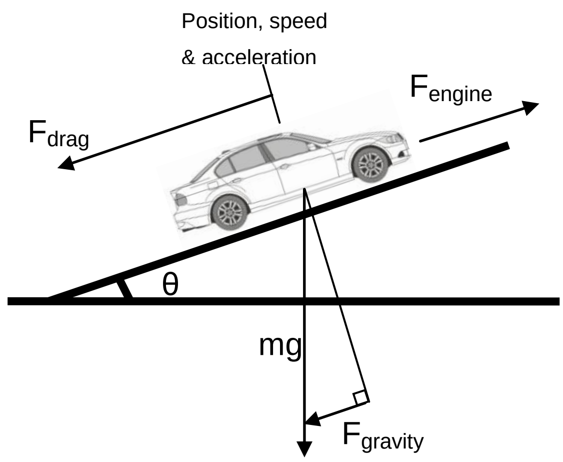

Figure 6.1 Schematic force diagram

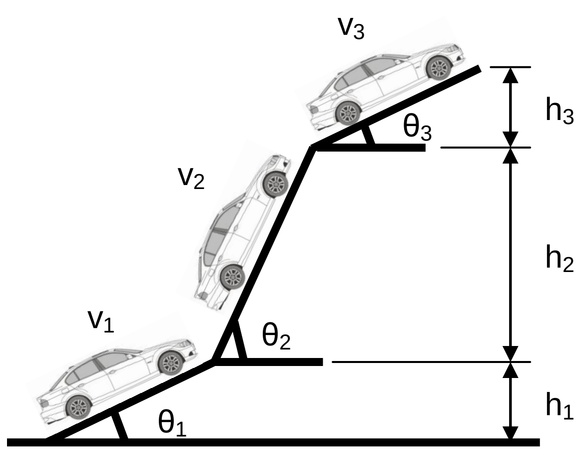

Figure 6.2: Uphill scenario

Applying Newton's second law, the resultant motive force, Fm on the vehicle is given by the resolved component parallel to the slope:

$$F_m = ma = F_{engine} - F_{drag} - mg\sin\theta \tag{1}$$

where 'm' is the vehicle mass, 'a' is the acceleration of the vehicle, 'g' is the gravity, Fengine is the driving force produced by the engine and Fdrag is the resistance due to aerodynamic drag.

The engine controller was to maintain the vehicle at a constant speed regardless of the change in road gradient, as shown in Figure 6.2. In the case of an uphill scenario, the controller adjusted the ignition timing according to the loads i.e. air drag and gravity, and associated with the advance road gradient derived from the predictive algorithm.

For a vehicle travelling at constant speed, i.e. a=0, equation (1) is reduced to:-

$$F_{engine} = F_{drag} + mg\sin\theta \tag{2}$$

The vehicle drag force is given by:-

$$F_{drag} = \frac{1}{2}\rho v^2 A C_d \tag{3}$$

The gross indicated power, Ip is given by:

$$I_{p=}\left(\frac{GIMEP \times V_h}{2\pi}\right)\dot{\theta} \tag{4}$$

By rearranging and substituting relevant parameters in the equation (2), (3) and (4), the power balance of the system is given by:-

$$\left(\frac{GIMEP \times V_h}{2\pi}\right)\dot{\theta} = \left(\frac{1}{2}\rho v^2 A C_d + mg\sin\theta\right)v \qquad (5)$$

where GIMEP is the gross indicated mean effective pressure of the engine, Vh is the engine capacity, $\dot{\theta}$ is the engine speed, v is the speed of the vehicle, ρ is the air density, A and Cd are the frontal area and the drag coefficient of the vehicle, respectively.

The term Ip, gross indicated power of an engine is the theoretical power of an internal combustion engine, given that it is completely efficient in converting the energy contained in the expanding gases in the cylinders. The term GIMEP is effectively torque without losses. Losses and efficiency can be built into a more complex model. An essential part of this vehicle model however is to obtain optimised engine operating parameters in order to achieve reduced fuel consumption and emissions. As the GIMEP is directly related to the engine control parameters 'ignition timing' and 'engine operating speed', a series of engine tests were carried out so that the optimised control map could be obtained and included in the simulation model.

6.1.2 Test engine

A single-cylinder Ricardo Mk I Hydra engine was built using a production B230 Volvo cylinder head, cut from a multi-cylinder head. Modifications to the camshafts, the oil and water systems were carried out. An intake manifold was fabricated from the multi-cylinder manifold (including the injector boss) to fit the cylinder head, throttle body and intake plenum. The fuel injector was directed down the intake port. The angle of the injector to the machined gasket face of the cylinder head was the same as the production engine configuration. A low-pressure fuel rail was manufactured to fit the injector. An exhaust manifold was fabricated to fit the cylinder head with bosses for a lambda sensor and thermocouple. The basic engine specifications are given in Table 5.

Bore	92 mm
Stroke	80 mm
Number of cylinders	1
Head	one cylinder of B230 4 valve multi-cylinder
Compression ratio	10:1 (nominal)
Maximum valve lift	9.8 mm
Ignition system	Brighton with Mitsubishi coil-on-plug
Coil charge duration	3, 4 ms
Spark plug	NGK BP8EVX
Spark plug gap	0.85 mm
Fuel injection system	Brighton with Bosch injector
Fuel pressure	3.5 bar
Injection timing	90 CA BTDC firing (F)

Table 5: Engine specification

6.1.2.1 Test bed and installation

The engine was installed on a test bed in the Sir Harry Ricardo Laboratories at the University of Brighton. The facility was equipped with a Plint dynamometer and electrically-driven pumps for oil, coolant and fuel supplies. The oil and coolant temperatures were maintained at 80°C and 90°C ± 2°C respectively. Oil, coolant, fuel, intake air and exhaust gas temperatures were recorded using type-K thermocouples. Intake manifold pressure was recorded using two, 2 bar absolute pressure transducers (Kistler 4045A2, Druck DPI 201). In-cylinder pressure was recorded using a gauge pressure transducer (Kistler 6125).

The rotational speed of the engine was measured using an optical encoder (Leine and Linde) with a resolution of 720ppr directly coupled to the crankshaft. The engine speed was maintained to an accuracy of ± 5 rpm. Mass flow of air through the engine was measured using a thermal mass flow meter (Endress and Hauser, AT70F). The minimum flow measurement (and the greatest uncertainty) was approximately 4 kg/hr. It was not possible to record the mass flow of air for engine speeds less than or equal to 1000rpm. The throttle valve was driven by a geared stepper motor. The throttle position was controlled with a multi-turn potentiometer.

AFR was measured close to the exhaust port using a calibrated wide-range lambda sensor (ETAS LA3). A greater proportion of fuel gives a lambda of less than unity, termed a rich mixture, while a greater proportion of air gives a lambda of greater than unity, termed a weak or lean mixture. The calibration of the sensor was checked periodically against a Horiba MEXA 7170DEGR exhaust gas analyser.

The fuel rig comprised of production automotive components (tank, regulator, and pump) integrated within a standalone unit with provision for fuel cooling. The low pressure part of the circuit was used for the Port Fuel Injection (PFI) injector. The fuel rail was fabricated from a modified Bosch production fuel rail. The fuel used throughout was pump grade BP 95 RON unleaded gasoline.

An AVL INDISET 620 data acquisition system and INDICOM V1.5 software were used to record data for combustion analysis over 400 consecutive cycles with a resolution of 0.5°CA. The in-cylinder gauge pressure was not pegged to the intake manifold absolute pressure conditions. The AVL thermodynamic correction was applied to in-cylinder pressure. In addition to in-cylinder pressure, the data acquisition system was used to record AFR, intake manifold absolute pressure, air mass flow rate, ignition timing signal, injection timing signal and engine coolant, air, exhaust and oil temperatures. The greatest error in the engine load condition was ± 0.04 bar recorded at the lowest speed condition.

All instrumentation was calibrated prior to engine testing and periodically throughout the programme. Before each test run, a hot, motored TDC determination was performed. The engine test installation is shown in Figure 6.3.

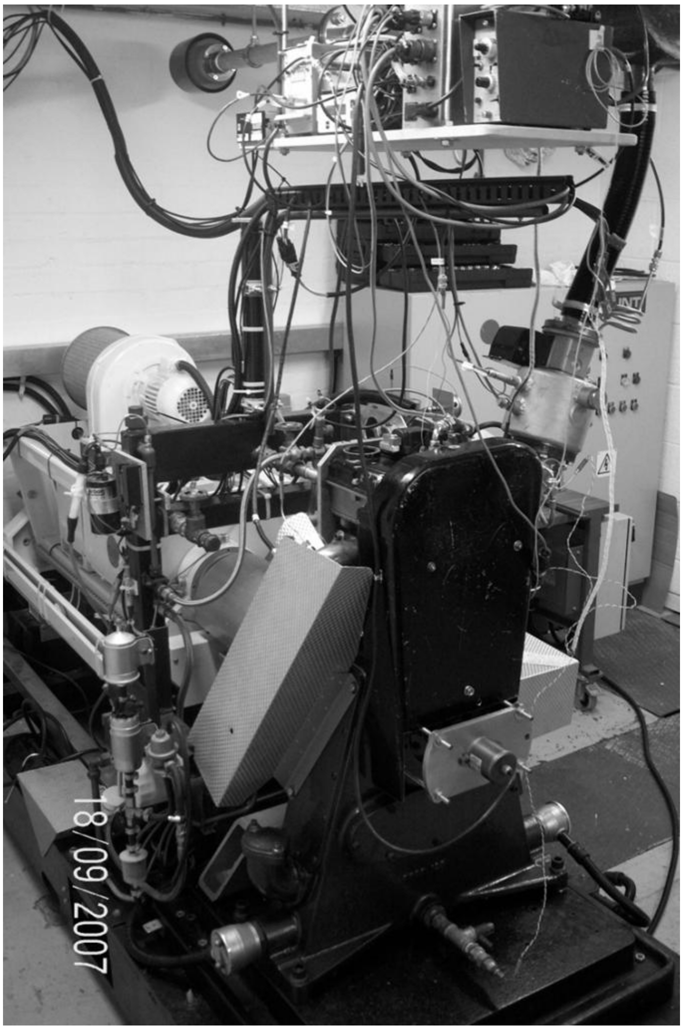

Figure 6.3: Engine test installation

6.1.2.2 Programme of work

The operating points for engine speed and load were selected to be representative of the conditions typically encountered during city driving between 1st and 4th gear. The engine was tested using the baseline production operating parameters for three speed and part load conditions: 1000 rpm and 1.0 bar GIMEP, 1500 rpm and 1.5 bar GIMEP and 1800 rpm and 1.8 bar GIMEP. At each part load condition, a mixture response swing was completed using MBT ignition timings (minimum ignition advance for best torque). Fuel was injected at 90°CA BTDC firing (CVI - Closed-valve injection). The spark plug gap was optimised for the ignition system used.

6.1.3 Engine mapping

The data obtained from the instruments were manipulated and analysed. An engine control map was generated using a Neuro-fuzzy modelling approach whereby two input and one output parameters were configured in the FIS. This Neuro-fuzzy system automatically adjusted the parameters of the basic fuzzy logic system very efficiently and identified the unknown process mapping from input to output data. The basic idea behind these neuro-adaptive learning techniques was to provide a method for the fuzzy modelling procedure to learn information about the data set, in order to automatically compute the membership function parameters that best allowed the associated FIS to track the given input/output data. The fuzzy membership function parameters were tuned using a combination of least squares estimation and a backpropagation algorithm for membership function parameter estimation. These parameters associated with the membership functions changed through the learning process similar to those of a neural network. Their adjustment was facilitated by a gradient vector, which provided a measure of how well the FIS was modelling the input/output data for a given set of parameters. Once the gradient vector was obtained, any of several optimisation routines could be applied in order to adjust the parameters so as to reduce the error between the actual and desired outputs. This allowed the fuzzy system to learn from the data it is modelling. The approach had the advantage over the pure fuzzy paradigm that the need for the human operator to tune the system by adjusting the bounds of the membership functions was removed.

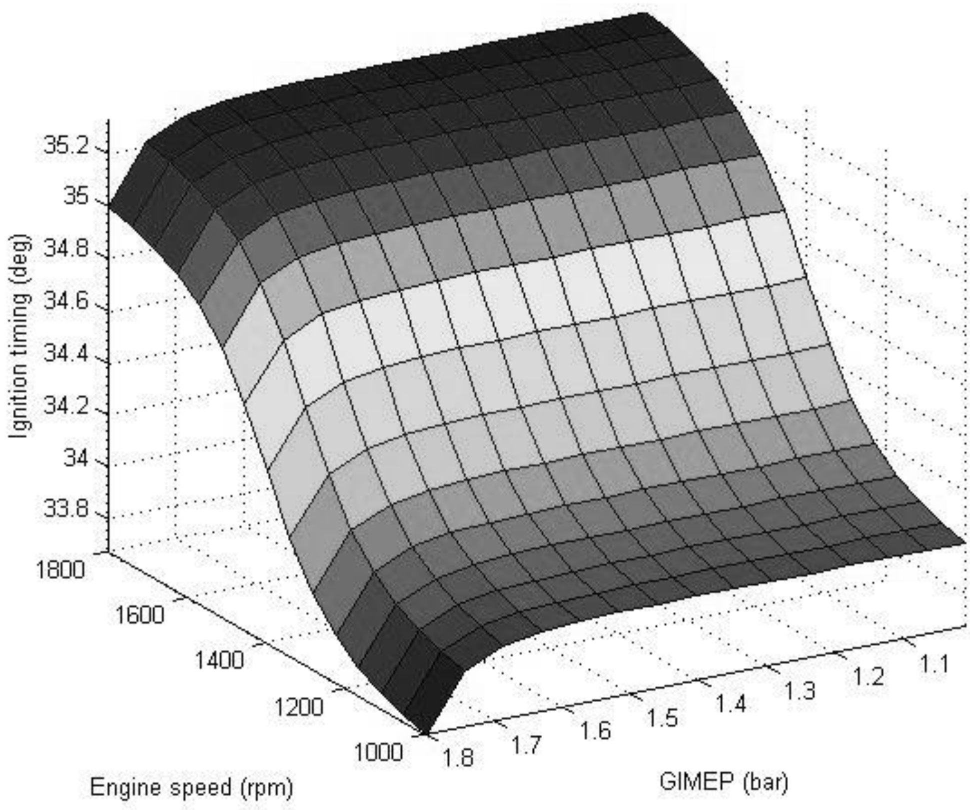

Figure 6.4: Engine control map devised using Hydra test-rig

Engine parameters were collected from 1000 rpm to 1800 rpm; ignition timing was recorded at each operating point where the AFR was monitored and maintained at λ=1.0 throughout, ensuring optimal combustion efficiency [32]. Small reductions in AFR can optimise power output but may lead to sizeable increases in emissions. Figure 6.4 depicts a three-dimensional plot representing the resulting Neuro-fuzzy mapping of engine speed, GIMEP and ignition timing. It can be seen that as the engine speed and GIMEP increased, the ignition timing increased in a non-linear piecewise manner, this being largely due to non-linearity of the characteristic of the input vector matrix derived from the raw engine data. This assumed that these raw engine data are fully representative of the features of the data that this Neuro-fuzzy FIS was intended to model.

By combining the optimised engine operational map with equation (3), the primary aim of the controller was to maintain the ignition timing with respect to the predictive gradient of the road, θ. This has formed a major part of this engine sub-system in the simulation.

6.2 Fuzzy Predictive Model

6.2.1 Techniques

A challenge of the project was how to meet higher safety standard requirements and obtain reduced fuel consumption through the use of live GPS road information for vehicle control. The approach was to use GPS to track the vehicle, and also to create the base map. At all other times GPS readings were used to validate or correct the base map when a reliable signal was available and of sufficient accuracy. The correct vehicle position was achieved by tracing this GPS signal received at a predetermined time interval.

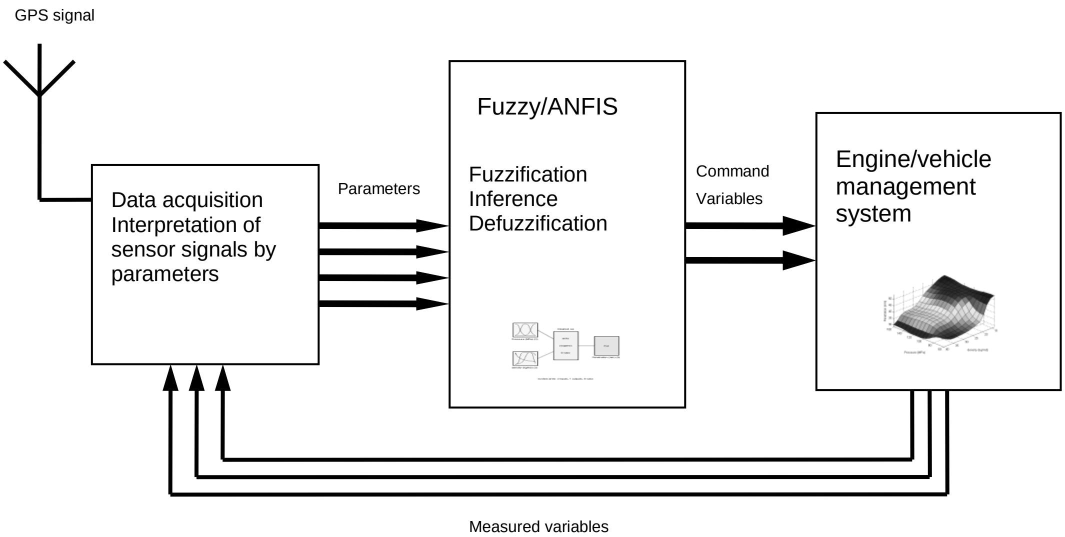

Figure 6.5: Engine/vehicle control using internally and externally acquired data

The conceptual overview of this model is shown in Figure 6.5. The control system acquired the position data through the serial interface, so that it could be used to improve the operation of several sub-systems in the vehicle, e.g. controlling a series of actuators or settings. Distance travelled and vehicle speed were recorded along the way. A 10 m distance span was used initially and this was maintained by on-board timers. The Neuro-fuzzy technique was used to derive a relief map of the test track, and here the position data was translated and represented by two input and one output membership functions together with twelve rules as part of the optimisation routine. The relief map that was devised under the scheme was used for future gradient prediction. The chosen intelligent technique involved extraction of necessary representative features from a series of data points. An experiment using this Neuro-fuzzy derived technique for modelling fuel spray penetration was described in Chapter 5 and achieved good results.

6.2.2 Experimental setup

The experiments were performed on a small passenger vehicle. A test route was established on the outskirts of Eastbourne in East Sussex, UK. A stand-alone laptop with a handheld GPS device was used throughout the experiment. The devised system was not connected to the vehicle control system. Therefore, an external GPS receiver was used and data were logged together with the time from the on-board clock through a serial Bluetooth interface.

Figure 6.6: Researcher capturing live GPS data

The aim of this investigation was to use a GPS receiver in conjunction with custom-written Matlab software to collect and store three-dimensional vehicle position data. The incoming stream of data was used to estimate the future elevation of the vehicle; this data was also expected to be of further use for dynamically influencing the control of an engine. The programme flowchart in Figure 6.7 showed how these algorithms integrated together to form a fuzzy predictive control system.

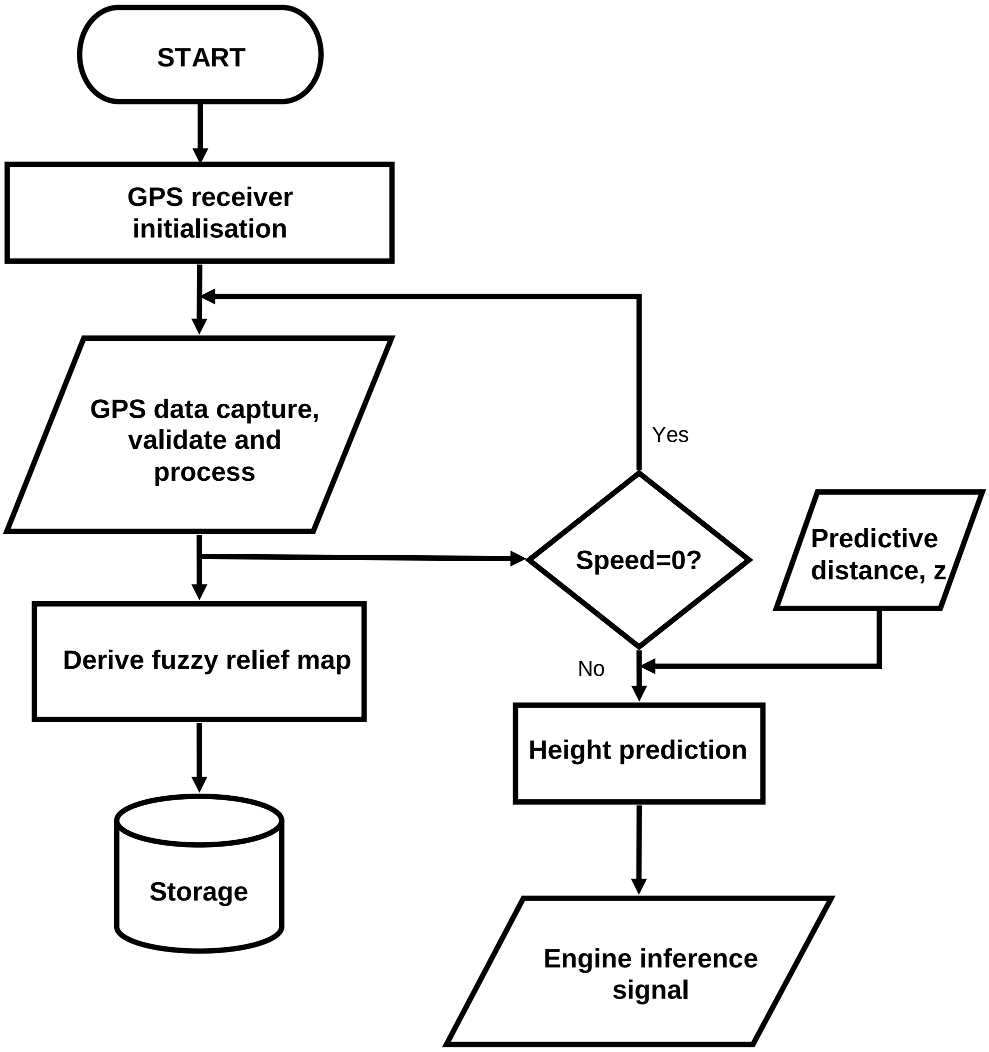

Figure 6.7: Predictive fuzzy inference system

The main software was divided into several functional modules, each of which performed its own set of calculations. The optimisation was performed by the Neuro-fuzzy module, the generated FIS was stored in the computer memory, whilst the timing of all these activities was governed by the on-board clock and timers.

6.2.3 Road gradient estimation

The fuzzy predictive control scheme is shown in Figure 6.8. The operation was triggered by a start signal and the status of the GPS data. A few given set points were needed i.e. predictive distance and sampling rate. The initial position data of the vehicle was registered and as a result a reference trajectory could be designed. From the reference trajectory, the next reference position was obtained according to the preset distance span. Meanwhile, the predictive algorithm calculated the next position of the vehicle using the current speed gathered from the GPS receiver. Based on the difference between the current and the predicted position, the fuzzy controller deduced the height at a set distance ahead and subsequently calculated the gradient.

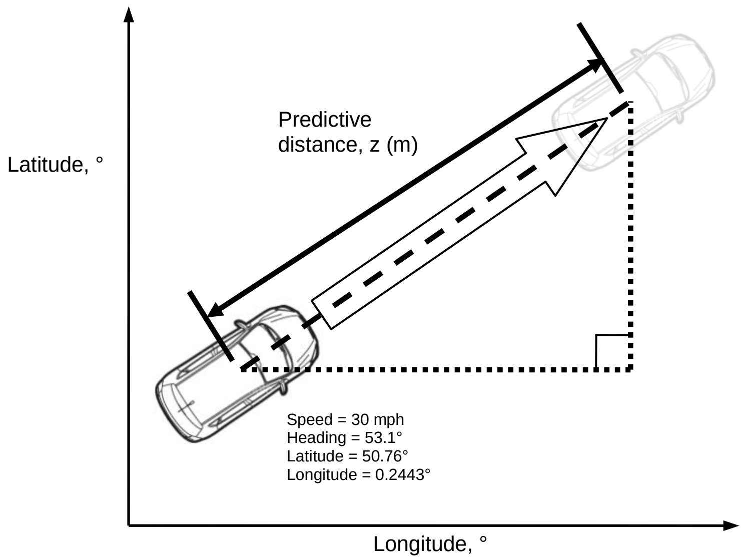

Figure 6.8: Road gradient predictive scheme

To calculate the distance between two points on the Earth required the use of the Great Circle Distance Formula [33].

$$Distance\,(m) = r \times \arccos\begin{bmatrix}\sin(lat1)\times\sin(lat2)+\\ \cos(lat1)\times\cos(lat2)\times\cos(long2-long1)\end{bmatrix} \qquad (6)$$

where r was the radius of the earth, 6378.7km. The variables lat1, long1 and lat2, long2 were the current position and predicted position, respectively.

The software was capable of handling double-precision floating point as this formula required a high level of floating point mathematical accuracy. The future location deduced using the described algorithms, was of particular interest since it provided information about the condition of the road ahead, in order to realise the appropriate control signal.

6.2.4 Validation

A set of measurements is shown in Figure 6.9. The solid line represents the height data; this was used to train the Neuro-fuzzy network to produce a base relief map of the route, a second run was performed at a variable sampling rate i.e. speed dependent sampling. These data were used as test data; the predictive algorithm was applied where future gradient estimation was computed. The result was compared and automatically logged for off-line analysis.

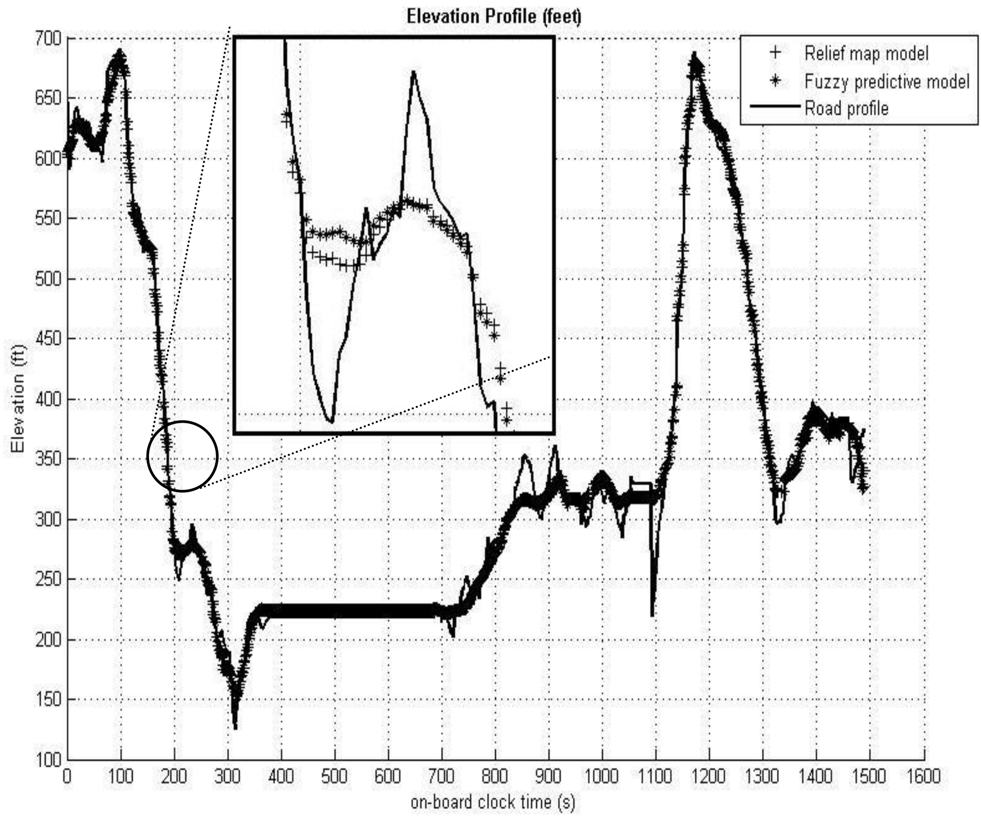

Figure 6.9: Measurement of road elevations

Chapter 7 : Simulation and Results

Simulation was carried out using Simulink; the status window was presented by an interface shown in Figure 7.1. The simulation model can be divided into four sub-systems, and each sub-system contained its own algorithms and mathematical functions. The vehicle GPS block essentially was a collection of vehicle running data, this was recorded and consolidated to be used and linked to the rest of the sub-systems. The predictive system shown in Figure 7.2 was the core function of the fuzzy predictive model derived in the previous section. This sub-system primarily used a Neuro-fuzzy technique to learn and model the route and this was continuously updated and replaced by new data as the vehicle moved along. A fuzzy logic derived relief map was generated and this was stored and used as a reference for the gradient prediction.

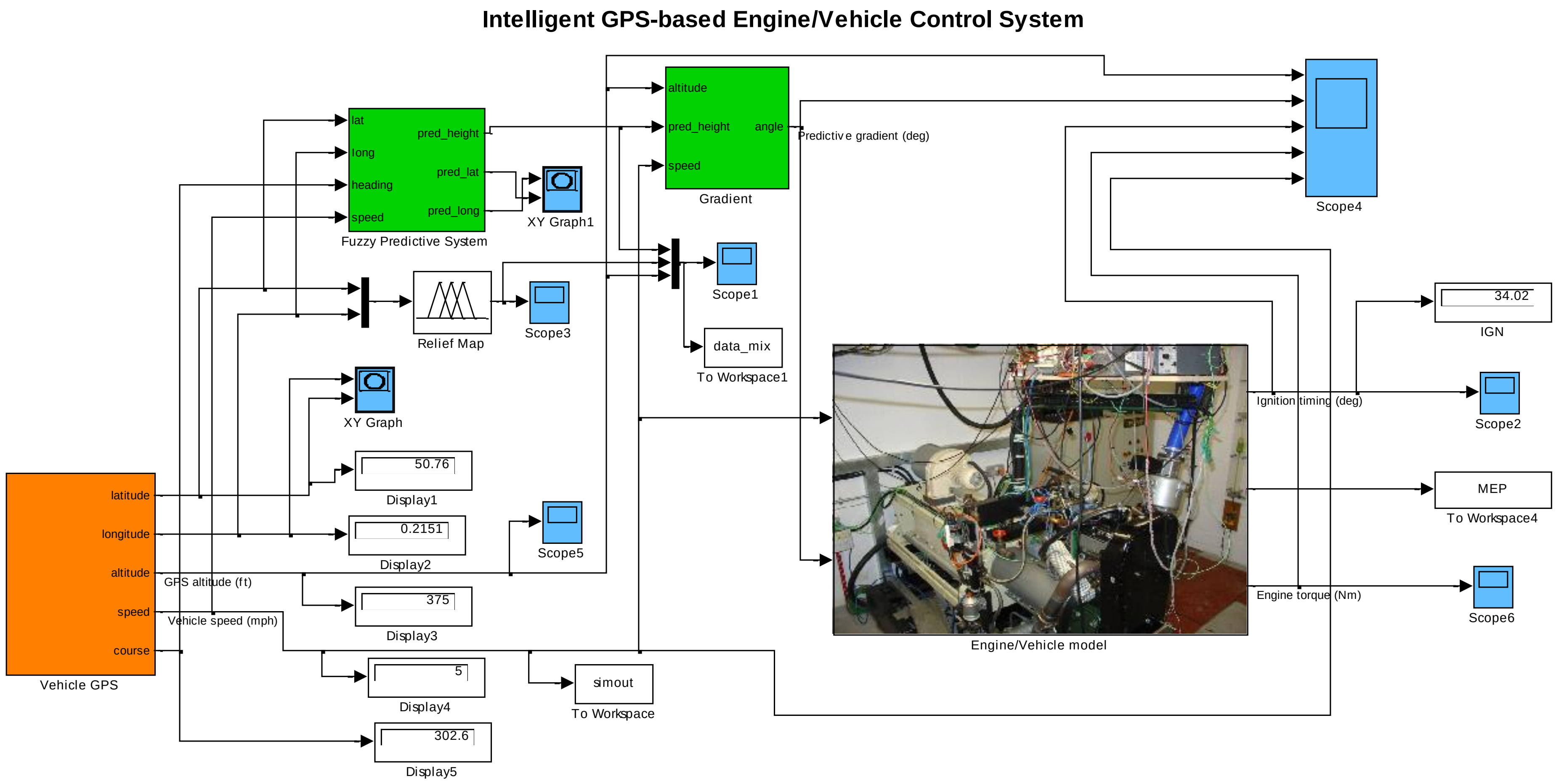

Figure 7.1: An overview of the simulation model

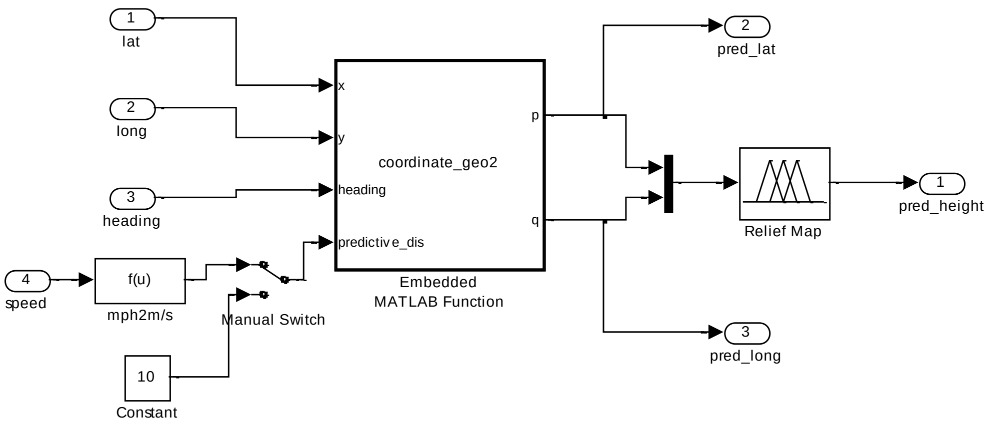

Figure 7.2: Fuzzy predictive system

The gradient block shown in Figure 7.3 was a series of mathematical functions. These were employed to convert predictive height into road gradient, taking into account the instantaneous vehicle speed and altitude. The engine/vehicle block consisted of real engine running data and the dynamic vehicle model previously explained in section 6.1.1. Each sub-block was associated with variables such as vehicle mass, frontal area, drag coefficient where each individual value could be changed if required. Figure 7.4 shows the connections and the mathematical relationship between parameters. This simulation provided an effective platform by adding the complexity of the system under control to the test platform.

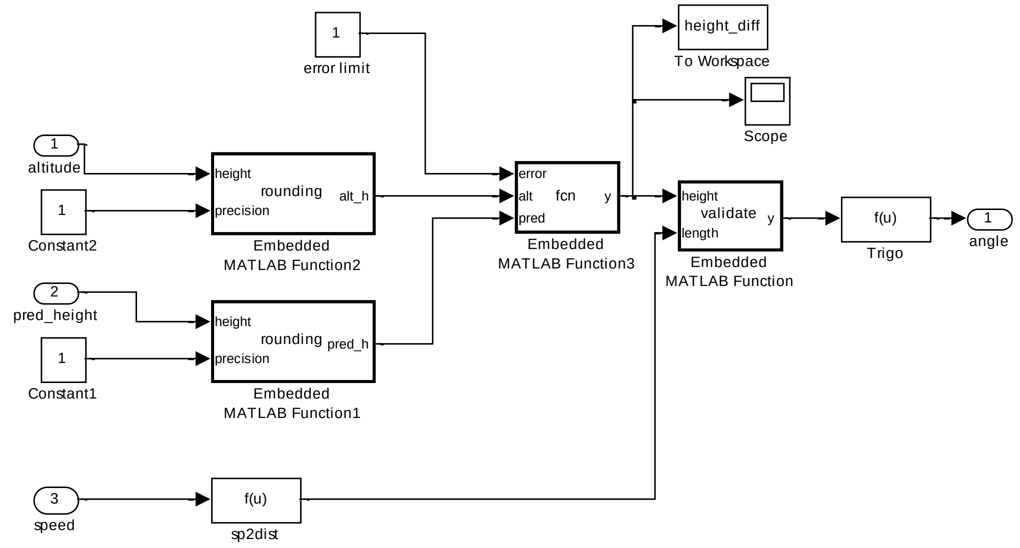

Figure 7.3: Gradient conversion block

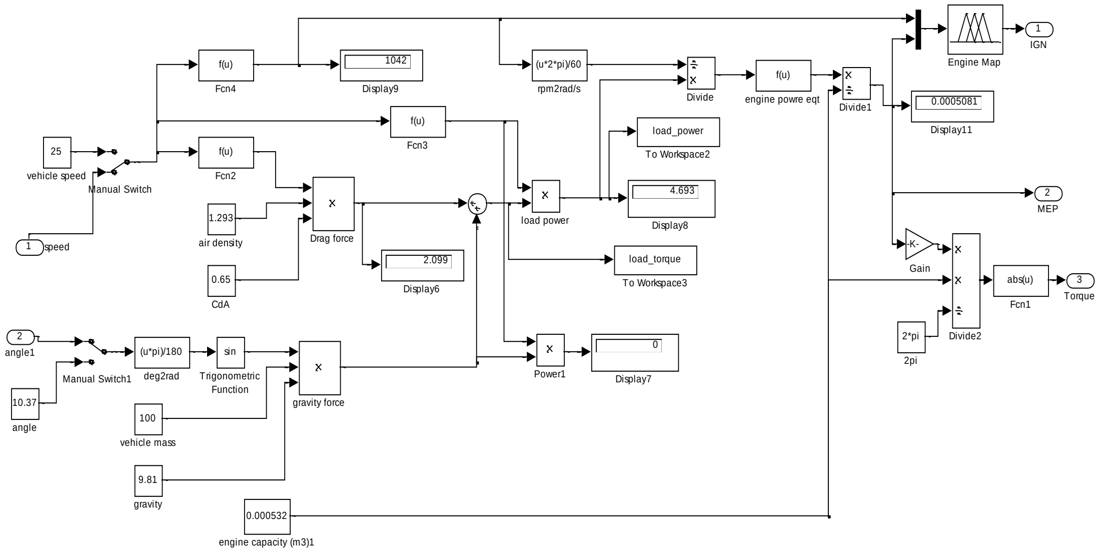

Figure 7.4: Engine/vehicle model

At each simulation time step, the controller action was to retrieve data from the GPS database, predict the height, compute the gradient with reference to the relief map and apply it through to the engine and vehicle dynamic model. The adjustable two-second time-step was selected to match the sampling speed of the GPS receiver and was utilised by the rest of the blocks. A series of plots and graphs was generated alongside, to monitor and validate the results. Figure 7.5 shows the results from a simulation run.

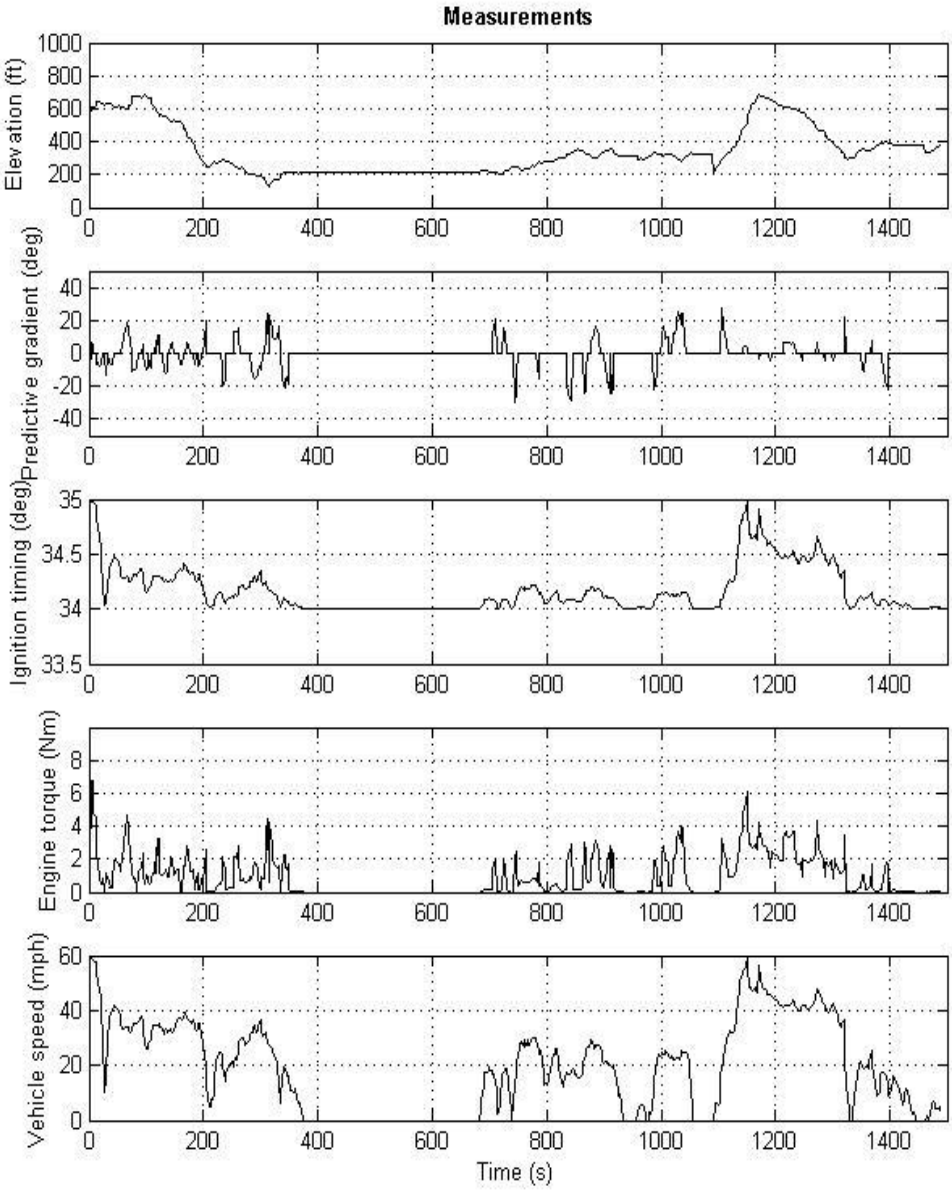

Figure 7.5: Simulation results

Each of these graphs indicates the corresponding results from each block and/or combination of blocks in the simulation. There were 754 GPS data points in the set, covering 7.8 miles of test track. The whole journey took just under 25 minutes. The elevation plot shows the height profile of the test route. The predictive gradient plot was based on the height profile of a training run. The controller obtained this predictive information and adjusted the ignition timing according to the loads. Notice that the change of engine torque is affected by the level of loads acting on the vehicle, i.e. mass, gradient and speed. Consequently, as speed increases, the drag resistance increases exponentially. These graphs demonstrate the effectiveness of the system and how it responded to different loading conditions and road gradient, derived from the fuzzy logic relief map. The ignition timing plot illustrates the optimal value at each time step, when the AFR was maintained in order to achieve minimum fuel consumption and emissions.

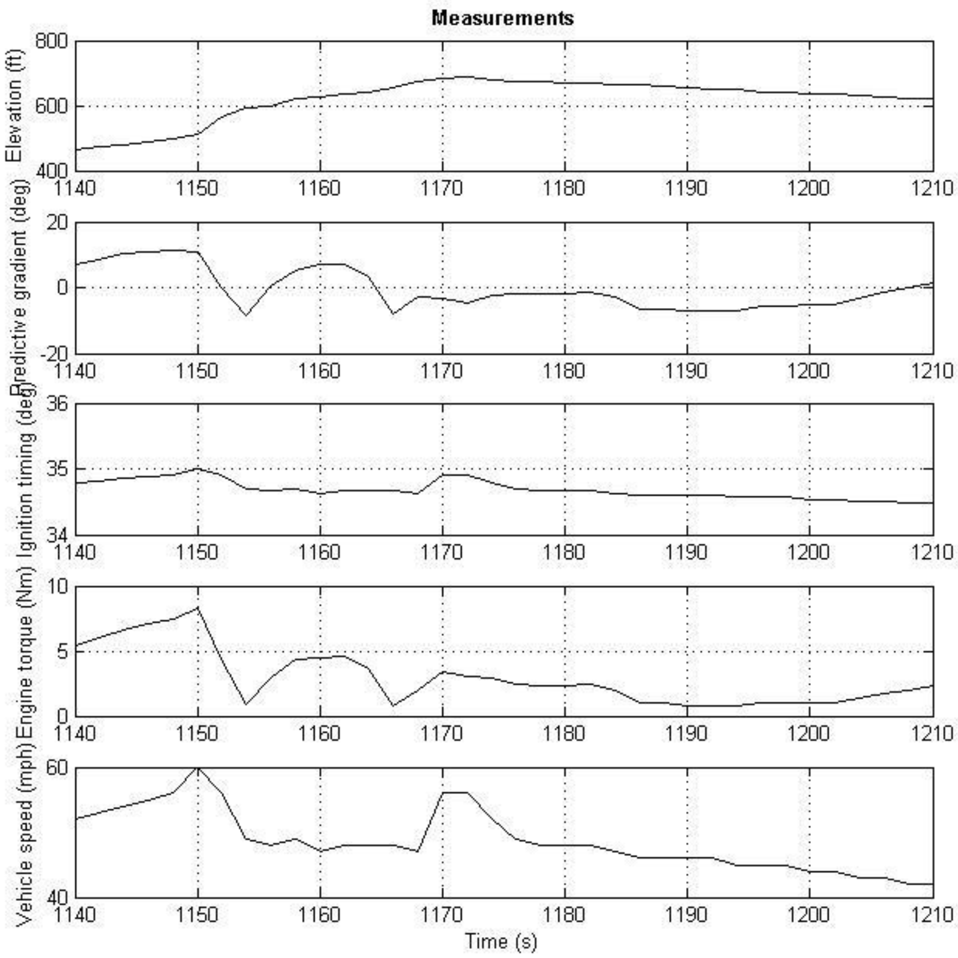

Figure 7.6: Magnified plot between 1170 to 1195 seconds

The chosen test route included a few successive up and down hill sections, a magnified graph in Figure 7.6 depicting one of them. In general, the predictive algorithm was able to distinguish the trend of a section except a glitch between 1150 to 1170 seconds. This is mainly due to the altitude resolution of the GPS receiver. Considering that the relief map information is continuously updated and replaced by new data as the vehicle moves along, thus this error might be diminished. However, the vehicle dynamic and the engine model worked well in response to the changes in vehicle speed and predictive gradient. The downhill section from 1172 seconds demonstrated the effect of decreasing loads due to both decreasing speed and gradient. The reduced loads on the vehicle reflected a reduction in engine torque which was shown in the ignition timing plot.

7.1 Model Validation

The simulated engine control parameters generated from the Simulink model shown in Figure 7.4 were electronically recorded and coordinated for offline Software-in-the-Loop (SIL) testing. Some key input parameters such as ignition timing, engine speed and engine torque used to run the engine at its optimal combustion efficiency were used as offline running data for the engine test-rig. The validation run was carried out at each operating point where the AFR was monitored throughout. The test was to check how well the Simulink model could predict the engine torque required for different road conditions and terrains.

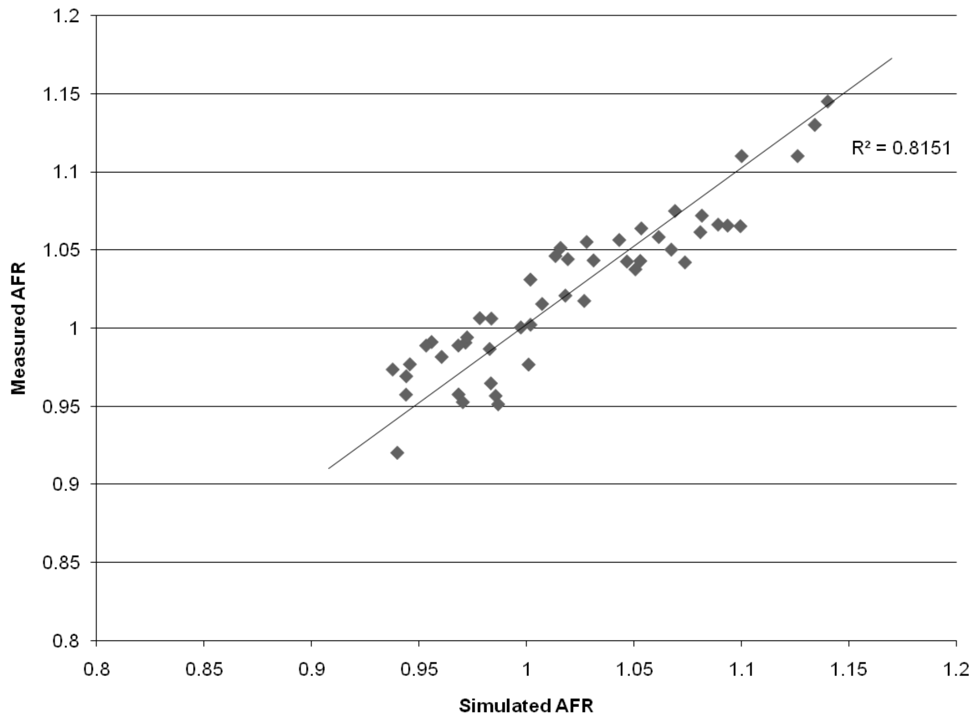

Figure 7.7: Scatter plot of measured and simulated AFR

Figure 7.7 shows the scatter plot of the AFR from two separate engine test run utilising the measured and the Simulink modelled dataset. The diagram demonstrated that the simulated values are close to the experimentally-measured values. The coefficient of determination, R^2 is 0.8151, indicating reasonably good correlation between the measured and simulated parameters.

Chapter 8 : Conclusions

Intelligent control is a promising field in modern control technology typically dedicated to highly complex and uncertain systems. This project has demonstrated two different techniques for representing fuzzy algorithms as well as discussing the relevance to neuro-fuzzy adaptive modelling through two separate previous studies. Pure fuzzy implementations which store the relational information and set definitions at discrete points have been used within the control community, whereas the neuro-fuzzy system has demonstrated its adaptability; the system automatically adjusted the parameters of the basic fuzzy logic system very efficiently and identified the unknown process mapping from input to output data. The latter technique has gained in popularity due to its strong links with neural networks.

Experimental results suggested neuro-fuzzy paradigm can be applied to control a small internal combustion engine. The establishment of an improved neuro-fuzzy paradigm for adaptive, fast and accurate control of small internal combustion engines is promising drawing from experience gained in intelligent engine control and modelling. The neuro-fuzzy algorithm that governs the control process is designed to automatically optimise the engine control parameters for each operating zone, to achieve performance in accordance with user-defined specifications. The best parameters for the neuro-fuzzy engine controller can be determined by using this ANFIS methodology together with the engine operating parameters. A test rig has been used to validate the tracking ability and insensitivity to engine conditions/load changes experimentally. The optimal fuzzy control strategy should be inexpensive to set up and computationally efficient. These combined or hybrid topological techniques are likely to be highly beneficial in future engine control research.

The successful and promising outcomes of the control and modelling work have led to the exploration of engine/vehicle control using GPS data. The simulated model demonstrated that intelligent systems can be used for predictive control of a vehicle. The technique represented a convenient and robust method of achieving road prediction, to form a fuzzy system that ‘looks ahead’ leading to improved fuel consumption and a consequent reduction in exhaust emissions. A new algorithm was demonstrated, which integrates live GPS data with the existing fuzzy logic derived relief map; matching software was developed and successfully implemented. This Neuro-fuzzy paradigm utilised simple map matching criteria, determining the gradient ahead based on current GPS position, and subsequently influenced the control of an engine. The GPS data observations were combined with fuzzy logic derived position to provide vehicle height information every two seconds.

Experimental results demonstrated the feasibility and advantages of this predictive fuzzy control on the trajectory tracking of a vehicle. Over 900 vehicle positions were generated and computed on each 7.8 mile test run using the newly devised algorithms. A similar number of test data were collected and compared to the height information generated by the predictive algorithm. The results showed that a good agreement was achieved between the predictive and the actual position data. The correlation coefficient of the elevation estimated by the Nero-fuzzy technique is 0.996, indicating good correlation.

The technique developed in road height estimation performs well and has been simulated using Simulink. This was combined with the technique of HIL in the development and test of a complex real-time system. The results showed how each system responded and a predictive height algorithm was used throughout the simulation. The work also demonstrated the potential effectiveness of the system for use in developing a simplistic vehicle control system for reduced fuel consumption and emissions. Due to the fact that the method is tested and used on known and repeated routes, the system is intended for and ideal on buses or fleet vehicles.

Chapter 9 : Further Work

There is considerable potential for further research in a number of areas that this work has encompassed. The system can be further improved with new, low-cost GPS receiver technology and integration with in-vehicle sensors as well as with the engine operating parameters. An inclinometer or accelerometer can be found in many current vehicles; this essential fitment not only provides active safety in vehicles, i.e. the stability control, the braking system and automatic gearbox, but could also improve the accuracy of the height predictive algorithm. An inclinometer prototype has been constructed and tested after the project. The standalone device was interfaced with the laptop via a USB link, enabling live road gradient and GPS data to be acquired. Preliminary tests showed the new setup capable of recording the inclination, potentially to be used to improve the accuracy and stability of the model.

Furthermore, the integration of the real-time testing environment for Simulink models can be achieved by connecting the host computer, a target computer, and the engine hardware under test enables access and interactively controls the engine hardware. It allows for validation of the models by comparing the performance and data obtained from an actual vehicle/engine test rig with similar configuration as the simulation. Tuning the Fuzzy parameter within the Simulink models helps to obtain high-fidelity representation of the physical plant of the vehicle. The development time can be significantly reduced as the result and any design changes can be directly and quickly implemented. Such approach provides a cost-effective solution by identifying and resolving problem in the laboratory instead of in the field.

In addition, the availability of more engine data will allow for the expansion of the engine model to include engine cold-start performance and subsequently lower the exhaust emissions, a new proposed fuzzy control loop illustrated in Figure 9.1 can be incorporated with the devised algorithm. This can be done via the use of low-cost temperature sensor together with an additional fuzzy set associated with a set of rule-base membership functions.

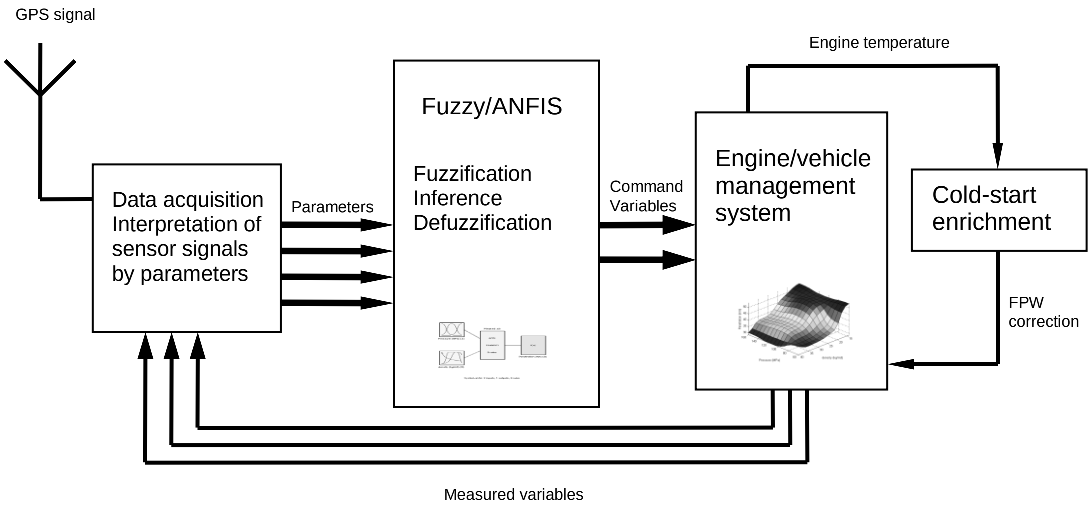

Figure 9.1: Engine/vehicle control with cold-start enrichment

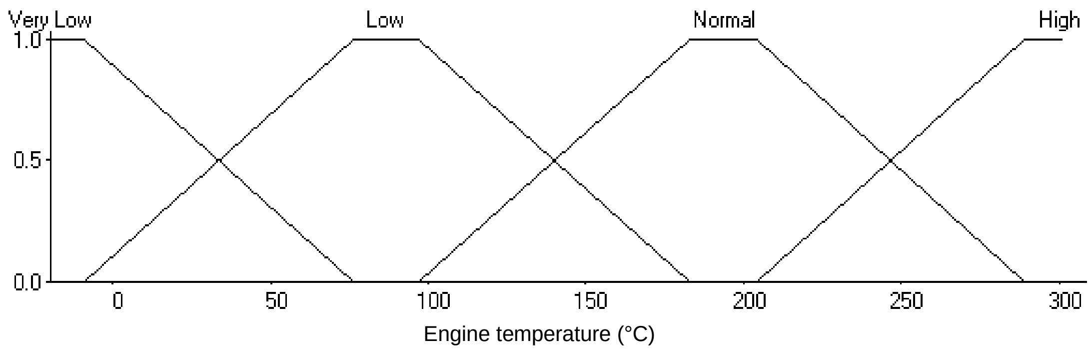

Figure 9.2: Cold-start fuzzy input set – engine temperature

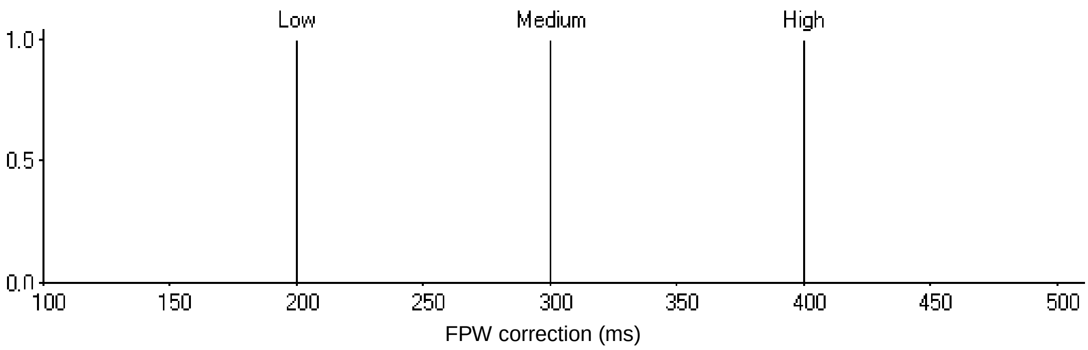

Figure 9.3: Cold-start fuzzy output set – FPW correction

Rule 1:	If the engine temperature is very low then the FPW enrichment is high
Rule 2:	If the engine temperature is low then the FPW enrichment is medium
Rule 3:	If the engine temperature is normal then the FPW enrichment is low
Rule 4:	If the engine temperature is high then the FPW enrichment is low

Figure 9.4: The fuzzy rule base – cold-start

The engine temperature fuzzy set shown in Figure 9.2 will be used in the fuzzy controller. It consisted of three simple trapezoidal membership functions for classes Very Low, Low, Normal and High. The optimisation of the rule-base shown in Figure 9.4 enables experimental adjustment of the limits of the membership classes and an experimental refinement will form a part of the calibration process. The output fuzzy set in this case could be a series of FPW singletons, illustrated in Figure 9.3, which combines with the developed FCS to facilitate air/fuel mixture enrichment when the engine temperature is below its optimum level.

Similarly, other running parameters such as vehicle load can be included in the model. This may be accomplished by the use of load sensor attached directly to the suspension parts. This arrangement can be found in most commercial vehicles and vehicles fitted with air-suspension level-control system. Such a system can be easily incorporated to the developed FCS by adding an additional fuzzy set, say, Vehicle Load. The system will need to be optimised with the rule-base and output fuzzy set. This essentially sets the operating point of the engine, and in conjunction with the ignition timing angle, determines the output power and the resulting level of emissions.

In general, the addition of these sensors and rule-base will certainly improve the accuracy of the model, the HIL testing offers the facility to change and record all these activities in real-time and make fine-tuning possible in the simulation. The prospect and experience will not only give rise to the use of intelligent techniques in improving engine and vehicle efficiency but also to demonstrate the combination of GPS, level and inclination sensors can further improve vehicle safety by providing the driver with an optimal speed recommendation in the form of a visual driving aid embedded in the instrumentation cluster for instance. Future work will be focused on system integration that can be cost effectively developed.

References

[1] The Auto Oil II web site created by the European Commission Directorate-General for the Environment, http://ec.europa.eu/environment/air/autooil.htm.

[2] European emission standards, the implementation of the renewed strategy 2007, http://ec.europa.eu/reducing_co2_emissions_from_cars/index_en.htm

[3] Heywood J.B., Internal Combustion Engine Fundamentals. McGraw-Hill, 1988.

[4] Denton T., Automobile Electrical and Electronic Systems, 2nd edition, 2000.

[5] Kalogirou S.A. Applications of Artificial Neural-Networks for Energy Systems, Appl. Energy 67 pp.17–35, 2000.

[6] Xu K., Luxmoore A.R., Jones L.M., Deravi F., Integration of Neural Networks and Expert Systems for Microscopic Wear Particle Analysis, Knowledge-Based Systems 11 pp.213–227, 1998.

[7] Zadeh L.A. Fuzzy sets, Information Control 8 pp.338–353, 1965.

[8] Wang L.X. Fuzzy Systems are Universal Approximators. Proceedings of the IEEE International conference on Fuzzy Systems, pp.1163-1170, 1992.

[9] Jang J. ANFIS: Adaptive Network-based Fuzzy Inference Systems, IEEE Transactions on Systems, Man, and Cybernetics 23, pp.665-685, 1993.

[10] Intel, Fuzzy Anti-Lock braking System, developer.intel.com/design/MCS96/DESIGNEX/2351.htm, 1996.

[11] Benchmark Suites for Fuzzy Logic, www.fuzzytech.com.

[12] Matsumoto N., Expert Antiskid System, IEEE IECON'87, pp.810-816, 1987.

[13] Davis L.I., Fuzzy Logic for Vehicle Climate Control, 3rd IEEE International Conference on Fuzzy Systems, pp.530-534, 1994

[14] Cook J.A. & Powell B.K., Modelling of an Internal Combustion Engine for Control Analysis, IEEE Control Systems Magazine, pp.20-26, 1988

[15] Morris R.L., Hopkins H.G. & Borcherts R.H., An Identification Approach to Throttle-torque Modelling, Society of Automotive Engineers (SAE),

International Congress and Exposition, SP-487, Detroit, MI, paper 810448, 1981.

[16] Dobner D.J., Dynamic Engine Models for Control Development – Part 1: Non-linear and Linear Model Formulation, International Journal of Vehicle Design, SP4, 1983.

[17] Kiencke U., A View of Automotive Control Systems, IEEE Control Systems Magazine, pp.11-19, 1988.

[18] Pedrycz W., Fuzzy Control and Fuzzy Systems, John Wiley & sons Inc., 1993.

[19] Vachtsevanos G. et al., Fuzzy Logic Control of an Automotive Engine, IEEE Control Systems Magazine, pp.63-68, 1993.

[20] Martinez A.S. & Jamshidi M., Design of Fuzzy Logic Based Engine Idle-speed Controllers, Symposium on Circuits and Systems, 1993.

[21] Alashe W.D. & Bedi J.S., Engine Idle Quality Improvement using a Fuzzy Logic Controllers, Midwest Symposium on Circuits and Systems, 1994.

[22] Nishizawa H. et al., Fuzzy Auto-tuning for Gasoline Engine's Electronic speed Controller, Trans of the Japan Society of Mechanical Engineering, Part C, 1994.

[23] Matsumoto H., Morita S. & Takiyama T., Application of Fuzzy Control to Internal Combustion Engines, Japan Society of Mechanical Engineering, 1994.

[24] Automotive Electrics Automotive Electronics, Robert Bosch GmbH. 5th Edition, Wiley, (2007)

[25] Directorate-General Energy and Transport, Galileo European Satellite Navigation System, http://ec.europa.eu/dgs/energy_transport/galileo/index_en.htm, (2007)

[26] Andes Froberg, Efficient Simulation and Optimal Control for Vehicle Propulsion, Linkopings University, Sweden, 2008

[27] Insight: Telematics and Fleet Risk Management, Zurich Financial Services Group, November 2010.

[28] Hellstrom, Erik: Explicit use of road topography for model predictive cruise control in heavy trucks. Master thesis. Linkoping University, Sweden, (2005)

[29] Wingren, Anna: Look Ahead Cruise Control: Road Slope Estimation and Control, Linkoping University, Sweden. LiTH-ISY-EX-05/3644-SE, (2005)

[30] Lattemann, Frank, Konstantin Neiss, Stephan Terwen & Thomas Connolly: The predictive cruise control – a system to reduce fuel consumption of heavy duty trucks, SAE Technical paper 2004-01-2616, (2004)

[31] Copp D. G., Burnham K. J., Locket F. P., Model Comparison for Feedforward Air/fuel Ratio Control. KACC International Conference on Control 9, pp. 670-675, 1988.

[32] Taylor, C.F.: The internal combustion engine in theory and practice, MIT press, Cambridge, Massachusetts, (1985)

[33] Sinnott R.W.: Virtues of the Haversine, Sky and Telescope, vol. 68, no. 2, , pp. 159 (1984)

[34] Loewenau, J.P., W.Richter, C.Urbanczik, L.Beuk, T.Hendriks, R.Pichler & K.Artmann: Real time optimization of active cruise control with map data using standardised interface, Proceedings of the 12th World congress on ITS, San Francisco, CA, USA, paper 2015, (2005)

[35] Powers, William F., Nicastri, Paul R.: Automotive vehicle control challenges in the 21st century, Control Engineering Practice 8, pp. 605-618, (2000)

[36] Bai H.S., Ruy J and Gerdes J.: Road grade and vehicle parameter estimation for longitudinal control using GPS, Proceedings of the 4th IEEE Conference on intelligent transportation systems, San Francisco, CA, (2001)

[37] Lee S.H., Howlett R.J., Walters S.D. and Crua C., Fuzzy Logic and Neuro-fuzzy Modelling of Diesel Spray Penetration, 2005. Ninth International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES2005), September, Melbourne, Australia, 2005

[38] Lee S.H., Walters S.D. and Howlett R.J., An Adaptive Neuro-fuzzy Modelling of Diesel Spray Penetration, 2005. Seventh International Conference on Engines for Automobile (ICE2005), September, Capri, Napoli, Italy, 2005

[39] Walters S.D., Lee S.H., Crua C., Howlett R.J., Neural Network Classification of Diesel Spray Images, Paper No. k06is-368, Prceedings, 10th International Conference on Knowledge-based Intelligent Information and Engineering Systems (KES2006), Bournemouth International Conference Centre, 9th-11th October 2006.

[40] Lee S.H., Howlett R.J. & Walters S.D., Fuzzy and Neuro-fuzzy Techniques for Modelling and Control, Paper No. k06is-613, Prceedings, 10th International Conference on Knowledge-based Intelligent Information and Engineering Systems (KES2006), Bournemouth International Conference Centre, 9th-11th October 2006.

[41] Lee S.H., Howlett R.J., Walters S.D. & Crua C., Modelling and Control of Internal Combustion Engines using Intelligent Techniques, International Journal of Cybernetics and Systems, Volume 38, Issue 5 & 6 June 2007, pp 509-533.

[42] Small Engine Control by Fuzzy Logic, S.H.Lee, R.J.Howlett and S.D.Walters, International Journal of Intelligent & Fuzzy Systems, Volume 15, Numbers 3, 4, 2004, pp.207-217.

[43] Lee S.H., Howlett R.J., Crua C., Walters S.D., Fuzzy Logic and Neuro-fuzzy Modelling of Diesel Spray Penetration: a comparative study, International Journal of Intelligent & Fuzzy Systems, Volume 18, Number 1 (2007) pp.43-56.

[44] Intelligent GPS-based Vehicle Control for Improved Fuel Consumption and Reduced Emissions, S.H.Lee, S.D.Walters & R.J.Howlett, 12th International Conference on Knowledge-based Intelligent Information and Engineering Systems (KES2008), Zagreb, Croatia, 3rd-5th September 2008.

[45] Priest M.W., Williams D. J., Bridgman H.A., (2000). Emission from In-use Lawn-mowers in Australia, Atmospheric Environment, 34, p657-664.

[46] Holzmann H., Halfmann Ch., Isermann R. (1997). Representation of 3-D Mappings for Automotive Control Applications using Neural Networks and Fuzzy Logic. IEEE Conference on Control Applications Proceedings, pp. 229-234.

[47] Diesel-engine management. - 4th ed. completely revised and extended. - Plochingen : Robert Bosch Gmbh; Chichester : Wiley, 2005

[48] Chin L., Mital D. P. (1998). Fuzzy Logic And Neural Networks, IEEE Region 10th Annual International Conference, Proceedings/TENCON, pp. 195-199.

[49] Trebi-Ollennu A, Dolan J M, Khosla P K. Adaptive Fuzzy Throttle Control for an All-terrain Vehicle, Institution of Mechanical Engineers, 2001, Vol 215 Part 1, pp. 189-198.

[50] Lee S.H., Howlett R.J., Walters S.D. (2003). A Fuzzy Control Systems for a Small Gasoline Engine. International Conference on Knowledge-based Information Engineering Systems and Allied Technologies, Proceedings, (KES2003) part 1, pp722-732.

[51] Ross, Timothy J. Fuzzy logic with engineering applications / Timothy J. Ross . - 2nd ed . - Chichester : Wiley, 2004

[52] Docquier Nicolas, Candel Sebastien (2002). Combustion Control and Sensors: A Review. Progress in Energy and Combustion Science 28, pp.107-150.

[53] Lee C.C., Fuzzy logic in control systems: fuzzy logic controller - Parts 1 and 2, IEEE Trans. System Man and Cybernet. 20 (2) pp.404-435, 1990.

[54] Howlett R.J., de Zoysa M.M., Walters S.D. & Howson P.A. (1999). Neural Network Techniques for Monitoring and Control of Internal Combustion Engines. International Symposium on Intelligent Industrial Automation.

[55] Odry P., Diveki S., Csasznyi A., Burany N., Fuzzy Logic Motor Control with MSP430x14x', Texas Instruments, 2005.

[56] Jager R., Adaptive fuzzy control, in: L. Boullart and A. Krijgsman, Eds., Application of Artificial Intelligence in Process Control, Pergamon Press, Oxford, pp.368-387. 1992.

[57] Taylor, George: Intelligent positioning: GIS-GPS unification / George Taylor and Geoff Blew, Chichester: Wiley, (2006)

[58] Brown M. and Harris C.J., Neuro-fuzzy, Adaptive Modelling and Control, Prentice Hall, Hemel Hempstead, 1994.

[59] Sugeno M. & Nishida M., Fuzzy Control of a Model Car, Fuzzy Sets and Systems 16 pp.103-113, 1985.

[60] Applied fuzzy systems / edited by Toshiro Terano, Kiyoji Asai, and Michio Su . - Boston; London : AP Professional, 1994

[61] Sutton R. & Towill D.R., An Introduction to the Use of Fuzzy Sets in the Implementation of Control Algorithms, J. Inst. Elec. Radio Eng. 55 (10) pp.357-367, 1985.

[62] Dorf, Richard C. Modern control systems / Richard C. Dorf, Robert H. Bishop . - 10th. ed. . - Upper Saddle River : Pearson Prentice Hall, 2005

[63] Wang L.X. & Mendel J.M., Fuzzy Basis Functions, Universal Approximation, and Orthogonal Least-squares Learning, IEEE Trans. Neural Networks 3 (5) pp.807-814, 1992.

[64] Sigmon, Kermit MATLAB primer / Kermit Sigmon, Timothy A. Davis . - 6th ed. - Boca Raton, Fla.; London : Chapman & Hall/CRC, 2002

[65] Magrab, Edward Barry: An engineer's guide to MATLAB : with applications from mechanical, aerospace, electrical, and civil engineering, 2nd edition, Upper Saddle River, NJ : Pearson Prentice Hall, 2005.

Appendix A: Published Papers

A number of papers were published during the course of this research project. Twelve papers were produced in conjunction with other researchers.

Three papers by the author are included in this Appendix as examples of the published work [42-44].

Volume 15, Numbers 3-4/2004
Engineering applications of Computational Intelligence
IOS Press
ISSN 1064-1246

Small Engine Control by Fuzzy Logic

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted] [redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

3.2

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted] part of the calibration process. The final set of rules contained in the rule-base is shown in Figure 6.

[redacted]

[redacted]

[redacted]	[redacted]
[redacted]	[redacted]
[redacted]	[redacted]
[redacted]	[redacted]
[redacted]	[redacted]
[redacted]	[redacted]
[redacted]	[redacted]
[redacted]	[redacted]
[redacted]	[redacted]
[redacted]	[redacted]
[redacted]	[redacted]
[redacted]	[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

Abbreviations

Volume 18, Numbers 1/2007
Engineering applications of Computational Intelligence
IOS Press
ISSN 1064-1246

Fuzzy Logic and Neuro-fuzzy Modelling of Diesel Spray Penetration: a comparative study

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

$$O_{1,j} = \mu_{B_j}(y), \quad i = 1,2$$

μ_{B_j}

$$O_{2,i} = \omega_i = \mu_{A_i}(x)\mu_{B_i}(y) \quad i = 1,2$$

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

Table 2: Fuzzy rule-base

[redacted]

Heated air

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

Ignac Lovrek Robert J. Howlett
Lakhmi C. Jain (Eds.)

Knowledge-Based Intelligent Information and Engineering Systems

12th International Conference, KES 2008
Zagreb, Croatia, September 3-5, 2008
Proceedings, Part III

[redacted]

[redacted]

[redacted]

[redacted] [redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]zy controller deduces the height at a set distance ahead and subsequently calculates the gradient

Appendix B: Matlab Program Code

The work described in this thesis involved the creation of approximately 1,000 lines of code written in the Matlab programming language. Two section of code are included in this Appendix as examples. The first is the program ‘timercap8.m’. This is a software implementation of the GPS data capturing and processing algorithms and the Fuzzy Predictive algorithm described in Section 6.2.2 of the thesis. A series of associated custom-written functions was devised in order to capture and validate the raw GPS data. The operation of these routines was explained in Section 6.2; the program-code for the road gradient estimation routines ‘manipulate3.m’form the second section of code included in this Appendix. The core part of the operation was to translate the three-dimensional vehicle position data to Neuro-fuzzy derived relief map. This was used demonstrated in the engine/vehicle simulation described in Section 7.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% File: timercap8.m
% Data acquisition/processing, Predictive Fuzzy inference functions
% Software: Matlab/Simulink version 7 or higher
%
% Input: data capturing parameters   Output: Vehicle tracking data
%
% Executable: timercap8.m
% Associated functions:
% Timercap8.m
% summary2.m
% readpacket4.m
% display2.m
% data_reg2.m
% analysis2.m
% deg2dec.m
% Decimal2Degrees.m
% Rounding.m
% report.m
% rt.m
%
% S.H.Lee
% Revision: 8.0
% Date: 11/8/2007
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

gpson=false;
fixed=false;
port='com10';
gps=serial(port);
set(gps,'BaudRate',38400);

while (gpson==false)
    disp('Switch on the GPS receiver and press anykey to continue');
    pause;
    gpson=true;
end
fopen(gps);

t = timer('StartDelay', 1,'Period', 2,'TasksToExecute', inf,...
          'ExecutionMode','fixedRate');
%t.StartFcn = {'my_callback_fcn', 'My start message',};
t.StopFcn = 'disp(''End of data capturing''); summary2; fclose(gps);
delete(gps);';
%t.StopFcn = { @my_callback_fcn, 'My stop message'};
t.TimerFcn = 'disp(''Timer fire!''); readpacket4; make_train;
display2';

%Global variables%
t_data=[];
trainfile = 't_track.mat';
raw_gps=[];
log_count=0;
lat='0';
long='0';
alt='0';

figure;
```

```
subplot(2,2,1);
axis off;
title('GPS Data', 'fontweight', 'bold');
text(0, .85, ['Initialising........']);

load t_track
start(t);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% File: summary2.m
% Data acquisition/processing, Predictive Fuzzy inference functions
% Software: Matlab/Simulink version 7 or higher
%
% Input: subroutine of timercap8.m   Output: url, Neuro-fuzzy
training % file
%
% Executable: via timercap8.m
% Associated functions:
% Timercap8.m
% summary2.m
% readpacket4.m
% display2.m
% data_reg2.m
% analysis2.m
% deg2dec.m
% Decimal2Degrees.m
% Rounding.m
% report.m
% rt.m
%
% S.H.Lee
% Revision: 2.0
% Date: 28/8/2007
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if log_count>0

[distance, seconds] = analysis2(track);
% Covert to hours
time = seconds / 3600;
if (time > 0)
   average_speed = distance / time;
end

figure;
axis off;
title('Summary', 'fontweight', 'bold');
% Distance
text(0, .95, ['Track length: ' num2str(distance) ' miles']);
% Time
    if (time <= 0)
        etime = 'Data not available.';
    else
        etime = [num2str(time) ' hours.'];
    end
text(0, .85, ['Elapsed time: ' etime]);
```

```
% Average speed
    if (time <= 0)
        aspeed = 'Data not available.';
    else
        aspeed = [ num2str(average_speed) ' miles/hour.'];
    end

text(0, .75, ['Average speed: ' aspeed]);

    % Start and end points
    [deg min2 sec dir] =
Decimal2Degrees(track.waypoint(1).latitude_dec, 'latitude');
    point = sprintf('%d\\circ %d" %4.2f'' %s', deg, min2, sec, dir);
    text(0, .55, ['Start:']);
    text(.1, .45, point);
    [deg min2 sec dir] =
Decimal2Degrees(track.waypoint(1).longitude_dec, 'longitude');
    point = sprintf('%d\\circ %d" %4.2f'' %s', deg, min2, sec, dir);
    text(.1, .35, point);

    %n = length([track.waypoint(1).latitude])-1;
    n = length(track.waypoint);
    [deg min2 sec dir] =
Decimal2Degrees(track.waypoint(n).latitude_dec, 'latitude');
    point = sprintf('%d\\circ %d" %4.2f'' %s', deg, min2, sec, dir);
    text(0, .25, ['End:']);
    text(.1, .15, point);
    [deg min2 sec dir] =
Decimal2Degrees(track.waypoint(n).longitude_dec, 'longitude');
    point = sprintf('%d\\circ %d" %4.2f'' %s', deg, min2, sec, dir);
    text(.1, .05, point);


end

save(trainfile, 't_track')
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% File: readpacket4.m
% Data acquisition/processing, Predictive Fuzzy inference functions
% Software: Matlab/Simulink version 7 or higher
%
% Input: GPS strings   Output: raw GPS location data
%
% Executable: via timercap8.m
% Associated functions:
% Timercap8.m
% summary2.m
% readpacket4.m
% display2.m
% data_reg2.m
% analysis2.m
% deg2dec.m
% Decimal2Degrees.m
% Rounding.m
% report.m
% rt.m
%
% S.H.Lee
% Revision: 4.0
% Date: 28/8/2007
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

tic
fixed=false;         % assume no gps fix signal
speed_flag=false;
while (fixed==false) || (speed_flag==false)
    output = fscanf(gps);
    comm=strread(output,'%s', 'delimiter', ',');
 switch comm{1}
   case '$GPGGA'
      if length(comm)==15
       GGA = textscan(output, '%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s',
'delimiter', ',');
         if str2num(comm{7})==1 % change to 1 when it is running
             fixed=true;
             %data_reg2;
             disp('Position fixed');
             status='Position fixed';
         end
      end
      %break;
   case '$GPRMC'
      if length(comm)==12
       RMC = textscan(output, '%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s',
'delimiter', ',');
         if comm{3}=='A' % change to 'A' when it is running
             % knot to mph

ground_speed=num2str(str2num(cell2mat(RMC{1,8}))*1.150779);
             course=cell2mat(RMC{1,9});
             speed_flag=true;
         end
       end
      %disp('Output is $GPRMC')
```

```
   case '$GPGSV'
      %disp('Output is $GPGSV')
   %otherwise
   %   disp('Unknown output.')

 end

end
data_reg2;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% File: display2.m
% Data acquisition/processing, Predictive Fuzzy inference functions
% Software: Matlab/Simulink version 7 or higher
%
% Input: GPS variables   Output: url GPS location data/plots
%
% Executable: via timercap8.m
% Associated functions:
% Timercap8.m
% summary2.m
% readpacket4.m
% display2.m
% data_reg2.m
% analysis2.m
% deg2dec.m
% Decimal2Degrees.m
% Rounding.m
% report.m
% rt.m
%
% S.H.Lee
% Revision: 2.0
% Date: 28/8/2007
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clf;
subplot(2,2,1);
axis off;
title('GPS Data', 'fontweight', 'bold');
text(0, .95, ['GPS status: ' status]);
text(0, .85, ['Computer time: ' t_now ' ']);

if (fixed==true)
text(0, .75, ['UTC Time: ' time_vec]);
text(0, .65, ['Latitude: ' lat ' ']);
text(0, .55, ['Longitude: ' long ' ']);
text(0, .45, ['Altitude: ' alt ' ']);
text(0, .35, ['Speed: ' ground_speed ' mph']);
text(0, .25, ['Course: ' course ' deg']);
text(0, .15, ['Elapsed time: '  ' hours']);
text(0, .05, ['Distance travelled: '  ' miles']);

for counter=1:length(track.waypoint)
    latitude(1,counter)=track.waypoint(1,counter).latitude_dec;
    longitude(1,counter)=track.waypoint(1,counter).longitude_dec;
    altitude(1,counter)=str2num(track.waypoint(1,counter).altitude);
    speed(1,counter)=str2num(track.waypoint(1,counter).speed);
```

```
end
% Display the bird's eye view
subplot(2,2,2);
plot(longitude, latitude);
title('Overhead (north up)', 'fontweight', 'bold');
xlabel('Longitude');
ylabel('Latitude');

% Display the elevation profile
subplot(2,2,4);

plot(altitude);
title('Elevation Profile (feet)', 'fontweight', 'bold');
xlabel('Waypoint number');
ylabel('Feet');

% Display the 3D data
subplot(2,2,3);
plot3(longitude, latitude, altitude);
axis tight;
xlabel('Longitude');
ylabel('Latitude');
zlabel('Feet');

end
t1(log_count)=toc;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% File: data_reg.m
% Data acquisition/processing, Predictive Fuzzy inference functions
% Software: Matlab/Simulink version 7 or higher
%
% Input: GPS variables   Output: track data
%
% Executable: via timercap8.m
% Associated functions:
% Timercap8.m
% summary2.m
% readpacket4.m
% display2.m
% data_reg2.m
% analysis2.m
% deg2dec.m
% Decimal2Degrees.m
% Rounding.m
% report.m
% rt.m
%
% S.H.Lee
% Revision: 2.0
% Date: 28/8/2007
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

log_count=log_count+1;
sattime=cell2mat(GGA{1,2});
[hh,mm,ss] = strread(sattime,'%2c %2c %6c', 1);
time_vec=[hh,':',mm, ':',ss];
```

```
lat=cell2mat(GGA{1,3});
lat_dir=cell2mat(GGA{1,4});
long=cell2mat(GGA{1,5});
long_dir=cell2mat(GGA{1,6});
% Altitude, metre to feet
alt=num2str(str2num(cell2mat(GGA{1,10}))*3.281);
t_now=datestr(now);
t_data=[t_data; time_vec ' ' t_now];
track.waypoint(log_count).pctime=t_now;
track.waypoint(log_count).pc_serial_date=now;
track.waypoint(log_count).ulc_time=time_vec;
track.waypoint(log_count).latitude=lat;
track.waypoint(log_count).latitude_dec = deg2dec(lat, lat_dir);
track.waypoint(log_count).longitude=long;
track.waypoint(log_count).longitude_dec = deg2dec(long, long_dir);
track.waypoint(log_count).altitude=alt;
track.waypoint(log_count).speed=ground_speed;
track.waypoint(log_count).course=course;
disp(time_vec)
disp(t_now)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% File: analysis2.m
% Data acquisition/processing, Predictive Fuzzy inference functions
% Software: Matlab/Simulink version 7 or higher
%
% Input: track data   Output: Spherical distance, time
%
% Executable: via timercap8.m
% Associated functions:
% Timercap8.m
% summary2.m
% readpacket4.m
% display2.m
% data_reg2.m
% analysis2.m
% deg2dec.m
% Decimal2Degrees.m
% Rounding.m
% report.m
% rt.m
%
% S.H.Lee
% Revision: 2.0
% Date: 9/6/2007
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [ distance, seconds ] = analysis2(track)
distance = 0;
time = 0;
for i=1:length(track.waypoint)-1
    distance = distance + SphericalDistance(track.waypoint(i),
track.waypoint(i+1));
end
% Time stamp stored as seconds, simply subtract end from beginning to
% get duration.
seconds = etime(datevec(track.waypoint(i+1).pc_serial_date),
datevec(track.waypoint(1).pc_serial_date));
```

```
% Covert to hours
%time = seconds / 3600;

function d = SphericalDistance(point1, point2)
% SPHERICALDISTANCE Compute the distance between two points on the
Earth's
% surface Earth's radius:
%   3437.74677 statue miles
%   6378 kilometers
%   3963 normal miles

% Convert to radians.
deg2rad = (180/pi);
lat1 = point1.latitude_dec / deg2rad;
long1 = point1.longitude_dec / deg2rad;
lat2 = point2.latitude_dec / deg2rad;
long2 = point2.longitude_dec / deg2rad;

% Compute distance in miles
r = 3963;
d = acos( sin(lat1) * sin(lat2) + ...
          cos(lat1)*cos(lat2)*cos(long2-long1) ) * r;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% File: deg2dec.m
% Data acquisition/processing, Predictive Fuzzy inference functions
% Software: Matlab/Simulink version 7 or higher
%
% Input: position data, heading   Output: position (degree)
%
% Executable: via timercap8.m
% Associated functions:
% Timercap8.m
% summary2.m
% readpacket4.m
% display2.m
% data_reg2.m
% analysis2.m
% deg2dec.m
% Decimal2Degrees.m
% Rounding.m
% report.m
% rt.m
%
% S.H.Lee
% Revision: 1.0
% Date: 8/6/2007
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function x = deg2dec(position, dir)
% DEGREES2DECIMAL Convert degrees, minutes, seconds to a decimal
degree
% position=3723.2475;
% dir='N';
pos=str2num(position)/100;
deg=floor(pos);
min=(pos-deg)*100;
```

```
x=deg+(min/60);

if (strcmp(dir, 'S')) || (strcmp(dir, 'W'))
    x=x*-1;
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% File: Decimal2Degrees.m
% Data acquisition/processing, Predictive Fuzzy inference functions
% Software: Matlab/Simulink version 7 or higher
%
% Input: position (degree)   Output: position (deg, min, sec)
%
% Executable: via timercap8.m
% Associated functions:
% Timercap8.m
% summary2.m
% readpacket4.m
% display2.m
% data_reg2.m
% analysis2.m
% deg2dec.m
% Decimal2Degrees.m
% Rounding.m
% report.m
% rt.m
%
% S.H.Lee
% Revision: 1.0
% Date: 8/6/2007
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [deg, min2, sec, dir] = Decimal2Degrees(x, type)
% DECIMAL2DEGREES Convert a decimal degree to degrees, minutes,
seconds

deg = floor(x);
min2 = 60 * (x - floor(x));
sec = 60 * (min2 - floor(min2));
min2 = floor(min2);

if (deg > 0)
    if (strcmp(type, 'latitude'))
        dir = 'N';
    elseif (strcmp(type, 'longitude'))
        dir = 'W';
    end
else
    if (strcmp(type, 'latitude'))
        dir = 'S';
    elseif (strcmp(type, 'longitude'))
        dir = 'E';
    end
end

deg = abs(deg);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% File: Rounding.m
% Data acquisition/processing, Predictive Fuzzy inference functions
% Software: Matlab/Simulink version 7 or higher
%
% Input: altitude, precision   Output: rounded altitude
%
% Executable: via timercap8.m
% Associated functions:
% Timercap8.m
% summary2.m
% readpacket4.m
% display2.m
% data_reg2.m
% analysis2.m
% deg2dec.m
% Decimal2Degrees.m
% Rounding.m
% report.m
% rt.m
%
% S.H.Lee
% Revision: 1.0
% Date: 29/8/2007
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function h = rounding(height, precision)
% rounding off to the nearest precision

h1=height/precision;
h2=floor(h1);
h3=h1-h2;
if h3>=0.5
    h2=h2+1;
end
h=h2*precision;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% File: report.m
% Data acquisition/processing, Predictive Fuzzy inference functions
% Software: Matlab/Simulink version 7 or higher
%
% Input: track data   Output: data/plot to url
%
% Executable: via timercap8.m
% Associated functions:
% Timercap8.m
% summary2.m
% readpacket4.m
% display2.m
% data_reg2.m
% analysis2.m
% deg2dec.m
% Decimal2Degrees.m
% Rounding.m
% report.m
% rt.m
%
% S.H.Lee
% Revision: 1.0
% Date: 29/8/2007
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure;
latitude=[];
longitude=[];
altitude=[];
speed=[];

for counter=1:length(track.waypoint)
    latitude(1,counter)=track.waypoint(1,counter).latitude_dec;
    longitude(1,counter)=track.waypoint(1,counter).longitude_dec;
    altitude(1,counter)=str2num(track.waypoint(1,counter).altitude);
    speed(1,counter)=str2num(track.waypoint(1,counter).speed);
end

%Display speed vs waypoints
subplot(2,2,1);
plot(speed);
title('Speed Profile (MPH)', 'fontweight', 'bold');
xlabel('Waypoint number');
ylabel('MPH');


% Display the bird's eye view
subplot(2,2,2);
plot(longitude, latitude);
title('Overhead (north up)', 'fontweight', 'bold');
xlabel('Longitude');
ylabel('Latitude');

% Display the elevation profile
subplot(2,2,4);
plot(altitude);
title('Elevation Profile (feet)', 'fontweight', 'bold');
```

```
xlabel('Waypoint number');
ylabel('Feet');

% Display the 3D data
subplot(2,2,3);
plot3(longitude, latitude, altitude);
axis tight;
xlabel('Longitude');
ylabel('Latitude');
zlabel('Feet');


[distance, seconds] = analysis2(track);
% Covert to hours
time = seconds / 3600;
if (time > 0)
   average_speed = distance / time;
end

figure;
axis off;
title('Summary', 'fontweight', 'bold');
% Distance
text(0, .95, ['Track length: ' num2str(distance) ' miles']);
% Time
    if (time <= 0)
        etime = 'Data not available.';
    else
        etime = [num2str(time) ' hours.'];
    end
text(0, .85, ['Elapsed time: ' etime]);

% Average speed
    if (time <= 0)
        aspeed = 'Data not available.';
    else
        aspeed = [ num2str(average_speed) ' miles/hour.'];
    end

text(0, .75, ['Average speed: ' aspeed]);

    % Start and end points
    [deg min2 sec dir] =
Decimal2Degrees(track.waypoint(1).latitude_dec, 'latitude');
    point = sprintf('%d\\circ %d" %4.2f'' %s', deg, min2, sec, dir);
    text(0, .55, ['Start:']);
    text(.1, .45, point);
    [deg min2 sec dir] =
Decimal2Degrees(track.waypoint(1).longitude_dec, 'longitude');
    point = sprintf('%d\\circ %d" %4.2f'' %s', deg, min2, sec, dir);
    text(.1, .35, point);

    %n = length([track.waypoint(1).latitude])-1;
    n = length(track.waypoint);
    [deg min2 sec dir] =
Decimal2Degrees(track.waypoint(n).latitude_dec, 'latitude');
    point = sprintf('%d\\circ %d" %4.2f'' %s', deg, min2, sec, dir);
    text(0, .25, ['End:']);
```

```
    text(.1, .15, point);
    [deg min2 sec dir] = 
Decimal2Degrees(track.waypoint(n).longitude_dec, 'longitude');
    point = sprintf('%d\\circ %d" %4.2f'' %s', deg, min2, sec, dir);
    text(.1, .05, point);

figure;
plot(longitude, latitude, 'r+');
title('Overhead (north up)', 'fontweight', 'bold');
xlabel('Longitude');
ylabel('Latitude');


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% File: rt.m
% Data acquisition/processing, Predictive Fuzzy inference functions
% Software: Matlab/Simulink version 7 or higher
%
% Input: track data   Output: speed and altitude with precision
%
% Executable: via timercap8.m
% Associated functions:
% Timercap8.m
% summary2.m
% readpacket4.m
% display2.m
% data_reg2.m
% analysis2.m
% deg2dec.m
% Decimal2Degrees.m
% Rounding.m
% report.m
% rt.m
%
% S.H.Lee
% Revision: 1.0
% Date: 23/8/2007
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% rounding off routine applied to speed, altitude
% precision:  speed=5mph, altitude=25ft
for counter=1:length(track.waypoint)

speed_rounded(1,counter)=rounding(str2num(track.waypoint(1,counter).s
peed),5);

altitude_rounded(1,counter)=rounding(str2num(track.waypoint(1,counter
).altitude),25);
end
figure;
plot(speed_rounded);
title('Speed Profile (feet)', 'fontweight', 'bold');
xlabel('Waypoint number');
ylabel('MPH');

figure;
plot(altitude_rounded);
title('Elevation Profile (feet)', 'fontweight', 'bold');
```

```
xlabel('Waypoint number');
ylabel('Feet');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% File: manipulate3.m
% Data acquisition/processing, Predictive Fuzzy inference functions
% Software: Matlab/Simulink version 7 or higher
%
% Input: track data   Output: variables, fuzzy relief map
%
% Executable: [var1, var2, var3,…..var n] = manipulate3(track);
%
% S.H.Lee
% Revision: 3.0
% Date: 20/2/2008
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function[tfile, fismat1, time_fuzzy, fuzzy_altitude, time_predictive,
pred_fuzzy_altitude, time, altitude, gps_time, gps_altitude,
gps_TimeInSec, course]=manipulate3(track)

[tfile]=MakeTrainSA(track, 10);
predictive_dis=10;

%tfile=track;
for i=1:length(tfile.waypoint)
    trnfile(i,1)=tfile.waypoint(i).latitude_dec;
    trnfile(i,2)=tfile.waypoint(i).longitude_dec;
    trnfile(i,3)=rounding(str2num(tfile.waypoint(i).altitude), 25);
end

numMFs=[5 5]; % num of MF, input output
inmfType=str2mat('gbellmf','gbellmf');
outmfType=str2mat('constant');
fismat=genfis1(trnfile,numMFs,inmfType,outmfType); % generate fis
fismat.input(1).name='latitude (deg)';
fismat.input(2).name='longitude (deg)';
fismat.output(1).name='height (ft)';

trnOpt=[400 0 0.01 0.9 1.1];
dispOpt=[0 0 0 0];
[fismat1,error1,stepsize]=anfis(trnfile, fismat, trnOpt, dispOpt);

figure; gensurf(fismat1);
figure; plotfis(fismat1);
title 'Relief map';
figure; subplot(2,1,1); plotmf(fismat1,'input',1);
subplot(2,1,2); plotmf(fismat1,'input',2);

figure;
subplot(2,2,1); plotmf(fismat1,'input',1);
subplot(2,2,2); plotmf(fismat1,'input',2);
subplot(2,2,3); gensurf(fismat1);
subplot(2,2,4); plot(error1);

lat_range=fismat1.input(1,1).range;
long_range=fismat1.input(1,2).range;
pred_fuzzy_altitude=[];
```

```
time_predictive=[];
fuzzy_altitude=[];
time_fuzzy=[];
gps_altitude=[];
gps_time=[];
TimeInSec_fuzzy=[];
gps_TimeInSec=[];
TimeInSec_predictive=[];
timer=0;

for counter=1:length(track.waypoint)
    latitude(1,counter)=track.waypoint(1,counter).latitude_dec;
    longitude(1,counter)=track.waypoint(1,counter).longitude_dec;
    altitude(1,counter)=str2num(track.waypoint(1,counter).altitude);
    speed(1,counter)=str2num(track.waypoint(1,counter).speed);
    time(1,counter)=track.waypoint(1,counter).pc_serial_date;
    course(1,counter)=str2num(track.waypoint(1,counter).course);
    gps_TimeInSec = [gps_TimeInSec timer];

    p=track.waypoint(1,counter).latitude_dec;
    q=track.waypoint(1,counter).longitude_dec;
    if (p>=lat_range(1,1)) && (p<=lat_range(1,2)) &&
(q>=long_range(1,1)) && (q<=long_range(1,2))
        %fuzzy_altitude(1,counter)= rounding(evalfis([p q], fismat1),
25);
        fuzzy_altitude = [fuzzy_altitude evalfis([p q], fismat1)];
        time_fuzzy = [time_fuzzy time(1,counter)];
        TimeInSec_fuzzy = [TimeInSec_fuzzy timer];
        %plot(counter,fuzzy_altitude(1,counter),'r+');
        gps_altitude = [gps_altitude altitude(1, counter)];
        gps_time = [gps_time time(1, counter)];
    end

    if (speed(1,counter)>0)
    [p(1,counter), q(1,counter)]=coordinate_geo1(latitude(1,counter),
longitude(1,counter), course(1,counter), predictive_dis);
    if (p(1,counter)>=lat_range(1,1)) &&
(p(1,counter)<=lat_range(1,2)) && (q(1,counter)>=long_range(1,1)) &&
(q(1,counter)<=long_range(1,2))
        %fuzzy_altitude(1,counter)= rounding(evalfis([p q], fismat1),
25);
        pred_fuzzy_altitude=[pred_fuzzy_altitude
evalfis([p(1,counter) q(1,counter)], fismat1)];
        time_predictive=[time_predictive datenum([0 0 0 0 0
(predictive_dis/(speed(1,counter)*0.27778))])+ time(1,counter)];
        TimeInSec_predictive = [TimeInSec_predictive timer];
        %plot(time_predictive(1,counter),
pred_fuzzy_altitude(1,counter),'g*');
    end
    end
timer=timer+2;
end
figure;
grid on;
hold on;
plot(time_fuzzy, fuzzy_altitude, 'r+');
plot(time_predictive, pred_fuzzy_altitude, 'g*');
plot(time,altitude,'-');
```

```
title('Elevation Profile (feet)', 'fontweight', 'bold');
xlabel('on-board clock time (s)');
ylabel('Elevation (ft)');
legend('Relief map model','Fuzzy predictive model','Road profile');
hold off;

figure;
grid on;
hold on;
plot(TimeInSec_fuzzy, fuzzy_altitude, 'r+');
plot(TimeInSec_predictive, pred_fuzzy_altitude, 'g*');
plot(gps_TimeInSec, altitude,'-');
title('Elevation Profile (feet)', 'fontweight', 'bold');
xlabel('on-board clock time (s)');
ylabel('Elevation (ft)');
legend('Relief map model','Fuzzy predictive model','Road profile');
hold off;
```