Health monitoring of engineering systems using neural networks

Scaife, Mark W.

Award date:
1994

Awarding institution:
University of Bath

Health monitoring of engineering systems using neural networks

submitted by Mark W. Scaife

for the degree of PhD

of the University of Bath

March, 1994

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Summary

In both the marine and power industries there is now a choice of *off-the-shelf* condition monitoring systems available that utilise artificial intelligence techniques to analyse engine performance data. It is accepted that such systems are a valuable aid to optimising performance and reducing down-time by assisting with maintenance planning. They rely on careful monitoring of an engine's performance, for instance speed, fuelling, boost pressure, turbine inlet pressure, turbocharger speed, and exhaust temperature. They often utilise interpolation and pattern recognition algorithms to compare the data with previously recorded data stored in look-up tables.

This thesis describes a test environment that was developed at the University to investigate alternative diagnostic approaches. A Diesel engine test cell has been configured to give ready access to both real engine data and data generated by a complex thermodynamic simulation. This simulator was modified to run on a powerful multi-processor computer so that entire performance maps can be generated quickly.

It is proposed that a neural network approach could be used as an alternative approach for the diagnostic problem, or to enhance current systems. Neural networks greatly reduce the need for large lookup tables and complex pattern recognition programs currently used. It is shown that neural networks make more efficient use of computer resources than do conventional expert methods in other applications.

Alternative approaches for constructing diagnostic systems are discussed. Demonstration models are used to investigate the alternatives. The first set of models uses experimental data supplied by British Gas, taken from a gas powered combined heat and power (CHP) plant. A spreadsheet environment was used to construct a simulation of this plant at the University. The simulator was used to generate data representing a healthy, and a faulted, system. This information was used to train a variety of networks to recognise a number of different hypothetical faults in the system. The models were tested on both simulated data and the real data supplied by British Gas.

The test facility at the University was used to develop a more complex on-line model. A detailed thermodynamic simulator of the test cell's turbo-charged

The test facility at the University was used to develop a more complex on-line model. A detailed thermodynamic simulator of the test cell's turbo-charged Diesel engine was constructed and calibrated give a good match with the test engine's base-line results. The simulator was then used to generate entire performance maps of the engine under a number of different fault conditions. This data was used to train a diagnostic neural network that was run on a PC. The neural network software was linked to the data acquisition software monitoring the test engine. This system was able to identify a limited number of faults artificially introduced on the test engine while it was operating anywhere in its performance envelope.

The models show that neural networks can respond well to real engine test data when trained on simulated results The neural networks are shown to be capable of learning how the performance of a complex system, such as a Diesel engine, varies with the introduction of a limited number of fault conditions. They can then be used to monitor the engine and indicate when a fault appears and identify what fault is present. The demonstration networks work well even if the load and ambient conditions are changed. They do, however, need a great deal of training data and do not respond well to conditions outside the domain of this data. The networks are also able to cope with the presence of multiple faults even though trained on individual fault data.

This thesis shows that neural network technology has the potential of simplifying the computational requirements of conventional performance analysis systems. It proposes that a complex diagnostic system, based on neural network technology, can be easily integrated with the control system of engineering machinery such as the modern Diesel engine and it suggests the architecture for such a system.

Acknowledgements

I would like to gratefully ackowledge the help given by the following:

Lloyds Register of Shipping for their financial support of this project.

British Gas for their financial support of further neural research

Derek Tilley for taking over as supervisor and his contribution to making this document complete.

Peter Prest and his team for their technical assistance with the control and instrumentation of the test facility.

Chris Mobley, of our instrumentation department, for his assistance with the black art of programming for ***windows®***.

Mike Wilson, for his assistance with the somewhat temperamental ***Meiko*** computer system

Tony Elley, Paul Frith and all the other technicians for their invaluable help with maintaining and running the test cell.

Most thanks, however, must go to ***Dr Steve Charlton*** for his guidance and support that has been so vital to this research project.

Table of contents

Chapter 3

Introduction to neural network techniques

Chapter 4

The test facilities used for this project

Chapter 5

The integrated software environment

Chapter 6

Simulation of the CHP plant and the Diesel engine

Chapter 7

Alternative approaches to fault diagnostics using neural networks (CHP plant)

Chapter 8

An on-line neural network diagnostic system (Diesel engine)

Chapter 9

Recommendations for future research

Chapter 10

Overview and conlusions

Appendix A

Mathematical Basis of the Back Propogation Method

Appendix B

Performance calculations used on British Gas CHP data

Appendix C

Advanced data acquisition techniques for multi-tasking personal computers

Appendix D

Performance calculations used on TL-11 data

Appendix E

Mathematical Basis of the Engine Simulation

Appendix F

Example input & output data files from the SPICE II engine simulation

Appendix G

Calculation of valve effective area for SPICE II model of the TL-11 Diesel engine

References

List of figures

List of tables

Glossary

The following terms and expressions have been used in this thesis:

Activation	The output level of a neuron in and artificial neural network
ADC	Analogue to digital convertor
ANN	Artificial neural network
Bias	Control parameter of a neuron's threshold function in an artificial neural network
CHP	Combined heat and power plant
CPU	Central processing unit of a computer.
DDE	Dynamic data exchange, this is a data passing protocol used in *windows 3.1*
DLL	Dynamic link library is a *windows* protocol for joining active program units
DT2811	Data Translation data acquisition card that plugs into the PC BUS
Epoch	A training iteration cycle for artificial neural networks.
FMEA	Failure mode effects analysis is a method of determining potential faults in a new system
Forward pass function	The threshold function that makes up the neural processing unit of an artificial neural network
FSD	Full scale deflection.
GUI	Graphical user interface is the generic term for front end programs such as *windows 3.1* and *X-windows*.
HX	Heat exchanger
MFlOP	Million floating point operations per second. This is a performance indicator for computer processors

Neuron / neural node	The processing element of an artificial neural network
NPT	Normal pressure and temperature, that is 1 bar absolute pressure and 25 C (or in SI units 10^5 N/m^2, 298 K)
PC	Personal computer
PDP	Parallel distributed program is the term for parallel processing techniques such as artificial neural networks.
Pseudo differential input	This is a means of connecting single ended amplifiers to minimise signal loss.
SI	International system of standard units
SPICE (II)	Simulated performance of internal combustion engines, a Fortran subroutine library for simulating CI engines by S.J.Charlton
Synapse	The connection between neurons
Unimap	A graph plotting package for mainframe computers.
Weight	The connection strength of a synapse in an artificial neural network

Notation

a	activation level of neural network	[-]
A	area	[m^2]
bmep	brake mean effective pressure	[N/m^2]
bsfc	brake specific fuel consumption	[g/kWhr]
$\mathbf{C_d}$	discharge coefficient	[-]
ch	channel number	
$\mathbf{C_p}$	Specific heat	[J/kgK]
d	diameter	[m]
h	specific enthalpy	[kJ/kg]
H	enthalpy	[kJ]
I	input vector of a layer of neurons	[-]
I	moment of inertia	[kgm^2]
k1,k2,k3,k4	governor and actuator coefficients	[-]
L	valve lift	[m]
m	mass of gas in the control volume	[kg]
mfp	mass flow parameter	[$kg\sqrt{K}m^2/N$]
O	ouput vector of a layer of neurons	[-]
p	pressure in the control volume	[N/m^2]
Q	heat added or rejected	[kJ]
r	coolant mixture & gas composition mass ratios	[-]
R	specific gas constant	[kJ/kgK]
t	time	[s]
T	stagnation temperature	[K]
u	specific internal energy	[kJ/kg]
U	internal energy	[kJ]
V	output activation of a neuron	[-]
V	volume of the control volume	[m^3]
w	weight coefficient of neural function	[-]
W	work done	[kJ]
x	rack position, displacement	[mm]

ε	effectiveness of a heat exchanger	[-]
ε	speed error	[rad/s]
η	efficiency	[-]
φ	bias of neural function	[-]
φ	crank angle	[deg]
μ	efficiency	[-]
λ	equivalent fuel-air ratio	[-]
ρ	density	[kg/m^3]
τ	torque	[Nm]
υ	efficiency	[-]

SUBSCRIPTS

a	ambient (pressure)
a	air (mass flow)
c	compressor
co	compressor out
f	fuel
e	flow out of a control volume
em	exaust manifold
i	flow into a control volume
ic	intercooler
it	inlet to turbine
im	inlet manifold
g	governor
o	initial rack position
o	upstream
r	fuel pump rack
t	turbine
to	turbine outlet
vol	volumetric
w	combustion chamber wall

Chapter 1

Introduction

1.1. The need for advanced monitoring and diagnostic systems.

All products of modern engineering, from bridges to satellites, require some form of monitoring to ensure they are performing their task satisfactorily. This monitoring may be vital for safety reasons or it may be used to improve operating efficiency by reducing energy consumption or to optimise system usage by assisting with maintenance planning.

Traditionally this monitoring has been performed by a person who has expert training which allows them to inspect and make judgements as to the system's state of health. In many cases this manual approach is now no longer suitable. This may be for a variety of reasons:

- the system may be too complex for an expert to understand fully.
- the mean time between failures may be so long that it is not economic to employ a specialist on site to watch over it.
- there may be no specialists available.
- the system may be too new for the expert to have gained sufficient experience of it.
- the consequence of a failure is so great that the possibility of human error in the diagnosis cannot be allowed
- the operating environment may be too hostile for human inspection.

Many methods have been developed in order to assist in the monitoring and inspection of engineering systems. Some aid the expert by providing information that helps them to make a judgement, others replace the expert completely by monitoring and analysing the system automatically. All major industries are actively involved in developing new monitoring techniques. Particularly active are the nuclear and aerospace industries because the consequence of a major failure is so catastrophic. The shipping industry is also very active in order to improve safety, maintenance planning, efficiency

and to reduce on-board manning levels. The automotive industry is under pressure to improve efficiency and reduce toxic emissions from their engines and so there is a need to monitor and control them in much more detail than has been previously required.

Although the details of each specific system will be different depending on its application, the general principles of monitoring and diagnosis are essentially the same. A diagnostic system must have:

- A set of measuring devices to show the current operating performance of the system.
- A reference set of data that shows the expected performance.
- A list of possible faults that may occur and their effect on the normal operating state.
- Some analysis techniques to interpret the above data.
- A system environment to control the whole operation.

1.2. The systems investigated by diagnostic techniques in this project.

This study is based on experiments which investigate diagnosis of two types of engineering system these are:

- a combined heat and power (CHP) plant based on a 2 litre natural gas engine,
- an 11 litre turbocharged Diesel engine.

These IC engine based systems are ideal for general demonstration for several reasons.

- They have, at their hearts, a complex multi-input multi-output non-linear system, the IC engine.
- IC engine performance has been studied extensively over many years, both theoretically and experimentally, and so its characteristics are well known.
- It is a suitable system for realistic, economic laboratory analysis.
- There are many systems in use world wide.

1.3. An overview of the limitations of current IC engine diagnostic systems.

There have been many attempts to design diagnostic tools to analyse sub systems of the IC engine and several that take a more holistic approach which are able to monitor an entire installation.

Conventional diagnostic systems fall into three catagories.

1) ***Direct measurement of the fault.*** Techniques, such as vibration analysis fall into this catagory because they measure the effect of a particular fault, such as a worn bearing, directly.

2) ***Rule based systems.*** This type is based on generating sets of rules that relate differences in cause and effect of different fault conditions. The database may be composed of logical statements such as *A fault in the oil pump will cause low oil pressure* and *Low oil pressure will cause high bearing temperature.* The diagnostic system uses heuristic reasoning to relate the observed deviation in performance, such as high bearing temperature, to the fault, in this case the oil pump. The rules can be generated by previous experimental, or field, observations or can come from theoretical analysis.

3) ***Pattern recognition systems.*** These rely on matching the currently observed performance with recorded data sets that represent the performance of the system under all healthy and faulted states.

For complex multi-component systems, such as the IC engine, there are too many components to make direct measurement practical and there can be great difficulty in establishing what the expected output should be for a given input state because there are so many variables that are involved. The size of the databases that are required for the rule and pattern based systems increases exponentially as the numbers of measuring points and fault states increase. The introduction of extra scenarios such as multiple faults, trend analysis or instrument failure may overload these traditional expert systems completely. Comprehensive diagnostic systems for IC engines have therefore been limited in their application to very large installations, such as found in power stations and ships, where the expense of the necessary processing power and instrumentation can be justified.

For the reasons mentioned above it is widely accepted that a new approach to data processing and analysis is required before health diagnostic systems can be applied on a wider scale to complex systems such as IC engines in cars.

1.4 The aims of this research project

This research investigates how artificial neural networks can be applied to the performance diagnosis of an IC engine based system.

It has been shown in other areas of research, how artificial neural networks are capable of recreating complex, non-linear, relationships with relatively small data processing requirements. This project assesses the ability of such techniques to extrapolate the state of health of IC engine based systems directly from measurements. Different network architectures and training algorithms are proposed and studied.

Real neural networks, for example the human brain, are able to recognise patterns such as faces and sounds despite corruption and fragmentation of input data. It is therefore felt that diagnostic tools working on the same processing principles should be able to cope with multiple faults and data corruption caused by instrument failure more easily than traditional pattern matching systems. This ability is therefore investigated in the study.

The specific aims of this project are:

-to develop an environment suitable for the analysis of diagnostic systems.

-to study diagnostic techniques currently in use.

-to investigate different ways of enhancing these methods using neural network techniques.

-to research the abilities of any methods proposed.

-to demonstrate a working model of a complete diagnostic system using this novel approach.

Chapter 2

Review of literature

2.1. Introduction

This thesis examines the potential of artificial neural network techniques applied to diagnostic systems. To put the work into context it is necessary to describe why these systems are needed and to review the current diagnostic techniques that are available and look at some commercial applications that apply them. It is also beneficial to review the history of artificial neural networks. The theory and techniques of artificial neural processing will be looked at in more detail later on.

2.2. The need for *intelligent* health monitoring and diagnostic systems

It is widely accepted that considerable savings in the running costs of large engineering systems can be achieved by using condition monitoring techniques to assist with the performance optimisation and maintenance planning. Nolden [1] states that typically 20% of direct maintenance costs can be saved this way. Fox [2] demonstrates how these savings can be achieved in the trials conducted by the National Bus Company for the DTI. In this case diagnostic equipment was installed into a fleet of 20 buses. It was shown that it can be up to ten times more expensive to conduct remedial maintenance after catastrophic failure than it is to carry out preventive maintenance in a planned way using condition monitoring techniques to detect deterioration of operating components. In this example, the condition monitoring system had the added benefit of being able to show the effectiveness of conventional servicing operations. For instance the system highlighted problems of the staff over-filling the lube oil systems and with calibration problems of the air brake system test rig.

However, it is suggested [3] that the large amounts of data that typical condition monitoring systems now generate are difficult to interpret manually. It states that expert systems can greatly assist the engineer with the task of extracting useful diagnostic information from this data. Expert

systems require a great deal of development and hardware to run. Neale [4], however, suggests that companies are prepared to invest up to 1% of a machine's capital value in monitoring devices. This has made it economically viable for suppliers of large power plant to develop quite complex systems dedicated to the health monitoring and performance optimisation of the plant. As hardware costs reduce and processing power increases these expert systems are becoming more powerful and are now viable propositions for a much larger number of plant users.

2.3. A review of health monitoring and diagnostic techniques

The main techniques used for data processing in diagnostic systems include:

- Trend analysis
- Spectral Analysis (e.g. vibration)
- Chemical composition analysis of working fluids
- Failure Mode Effects Analysis
- Rule based expert systems
- Fuzzy logic
- Multi Level Flow Models
- Neural networks

Each of these techniques will now be looked at in more detail:

2.3.1. Trend analysis

Trend analysis is one of the simplest and most fundamental condition monitoring techniques. By relating measured parameters to a time base it is possible to extract much more information than is possible from instantaneous observation. The rate of change of performance of a system can give useful information that can assist in the prediction of a failure. Typical trend analysis is done manually, most monitoring systems store a history of measured values so that the operator can view the trends, as illustrated in many of the examples below. Various mathematical techniques have been utilised to project trends to anticipate the time to failure. Most methods however are limited by the effects of noisy data and the highly non-linear nature of system failures. As will be discussed later, neural network techniques have been effectively applied to the trend analysis problem.

2.3.2. Vibration Spectral Analysis

Spectral Analysis is another method of extracting information from the time domain of measurements taken from from dynamic systems. Spectral analysis is most commonly used for diagnosing information from vibration and noise transducers. However, for automotive applications, the technique is also used for the more specific role of extracting diagnostic information from crankshaft speed and torque measurements from engines.

The general methods for spectral analysis are described by Haddad [5] where he shows that vibration analysis can be conducted at various levels:

- Level 1; overall vibration can give some indication as to the general state of health of the plant.
- Level 2; third octave frequency analysis gives a clear indication of the magnitude of the major frequencies that indicate most events occurring within an engine, such as piston movement and bearing rotation. Current operating states can be compared with historical data of healthy systems to identify the increased contribution of individual components to the overall vibration.
- Level 3; narrow band analysis can be conducted by filtering out all but specific frequencies. This can be used to investigate known frequencies at which specific faults will cause vibrations.
- Level 4; normalised narrow band analysis can be used to produce output proportional to the amount of vibration present in certain components. Detailed knowledge of the transfer functions defining the response of the components at a certain operating point are needed for this technique to work.

A method that can perform diagnostic operations on IC engines by analysis of speed and torque wave forms is described by Mauer and Watts [6]. They describe the filtering needed to reduce the signal-noise ratio in order to extract useful information from the torque and speed wave forms. Digital techniques using low pass Fourier filters were used. Analysis of the filtered results can show the power contributions made by each cylinder as it compresses and fires. This information is used to identify cylinder to cylinder faults. The technique can be used to differentiate between compression and combustion faults but is currently limited to engines with less than 6 cylinders.

2.3.3. Chemical analysis of working fluids

Chemical analysis of the lubricating fluids can be an effective means of studying the degradation of components that are too inaccessible for normal inspection techniques. Skinner [7] indicates that this method can be used to discriminate between wear of the various moving parts in an engine such as the piston rings and bearings because of their different chemical composition. The technique can be enhanced by fingerprinting various components by incorporating unique compounds in their wear surfaces.

Reasons for the wear can also be determined to a limited degree. Skinner shows that chemical analysis can differentiate between dirt, sand and sea water contamination. The technique can also highlight harmful contamination of the fuel by cracking catalysts.

The oil analysis is usually conducted off-line. The engine operator sends samples to the oil company for analysis. Results are typically returned within 2-3 days. The potential of on-line analysis is being investigated by many institutions and companies.

2.3.4. Failure Mode Effects Analysis (FMEA)

FMEA is a diagnostic approach that has arisen from the design phase of a system. The various modes of failure of each component are assessed during the design stage. The consequence of each mode is then projected throughout the whole system so that the effects can be assessed. Statistical methods can be used to determine the probability of failure of each component and the consequence to the whole system. The information can be used for calculating estimated mean time between failures and for tracing faults when they occur.

Clearly this method becomes very numerically intensive as the number of components increase.

2.3.5. Rule based expert systems

Rule based systems have been the mainstay of expert systems to date as is illustrated in the example of commercial systems below. These systems are based on encoding a set of rules that define the relationships between parameters and between components and sub assemblies of the system. The rules can be generated from service histories, interrogation of human experts and from FMEA described above. Logic processing algorithms are used to establish which rules are valid at any one time. Many high level languages

have been written to simplify the design and application of the knowledge data bases.

Knowledge data bases rapidly grow in size as the number of components and fault scenarios increase and so these methods are very memory intensive and rely heavily on very efficient data base interrogation procedures to be effective.

2.3.6. Fuzzy logic

Fuzzy logic systems operate on fuzzy set theory defined by Zadeh in 1965. The logic systems operate with similar logic to the rule based systems described above. However, instead of operating on absolute values, fuzzy systems operate on a range of values defined by a membership set. This allows the processing of more abstract rules, for instance a rule could be: IF the oil pump of an engine is *worn* THEN the oil pressure will be *low* AND the oil temperature will be *high.* Although fuzzy system research started in the US, Munakata [8] describes how the Japanese were the first to commercially exploit this technology in practical systems. Many Japanese appliances include fuzzy logic controllers. The development of these systems in Japan have been assisted by the Laboratory for International Fuzzy Engineering Research founded in Japan in 1989.

2.3.7. Multi-Level Flow Models (MFM)

This technique is described by Lind [9]. Using MFM a system is represented as a set of inter-related flow structures describing the flow of mass, energy and information on different abstraction levels. This technique allows the integration of various categories of knowledge about the plant in a natural way. The flow structures are defined as a set of abstract functions that have certain goals which are used as inputs to other functions. For instance an oil pump has the goal of achieving *oil mass flow* and has inputs of *oil tank level* and *pump motor power.* Oil mass flow will be one of the inputs to the lubrication model. MFM provides a structure for interrogating the causes and effects of various disturbances in these subsystems. MFM gives a more structured means of tracking the effects of various faults than does conventional heuristic reasoning. This allows a more efficient method of conducting the monitoring and identification tasks of diagnostic systems.

2.3.8. Neural networks

Neural networks are data structures that are designed to store and operate knowledge in a highly interconnected network of simple processing nodes. This technique has been developed to mimic the way that the brain processes data. The methods by which neural networks work are described in detail in the next chapter.

2.4. An overview of techniques used in current commercial diagnostic systems

This thesis is primarily concerned with diagnosis of engine installations, there are several commercial diagnostic systems currently available for these installations. This section reviews the most significant. As can be seen from this review most are aimed at very large installations such as ship and power station engines and large processing plants. This is due to the complex, hence expensive, processes that they employ.

2.4.1. Ricardo Consulting Engineers - Diesel Engine Intelligent Monitoring System (DEIMOS)

Richards [10] describes the demonstration Diesel Engine Intelligent Monitoring System (DEIMOS II). This demonstration model was developed by a consortium lead by Ricardo consulting engineers as part of the UK Government's Alvey programme for advanced information systems. DEIMOS uses a blackboard architecture to combine various different modular information processing systems. The tasks that these sub systems are responsible for include:

- Live data acquisition from the engine.
- Data retrieval from historical databases to provide reference data with which the current operating state can be compared.
- Numerical simulation. This is used to provide reference data so that the live data can be qualitively assessed despite changes in ambient conditions. For instance, values that are higher or lower than the simulation results are used by the expert modules for diagnosis.
- Feature finding. This system is used to look for significant groups of performance deviation, such as temperatures and pressures either side of a compressor.

- Set comparison. Pattern recognition techniques are used to investigate the deviation results.
- Inference mechanism uses rule based algorithms to link cause and effects of potential known faults.
- Truth maintenance. This is used to arbitrate when different conclusions are met by different sub systems. If assumptions have been made in some of the diagnostic solutions then the truth maintenace system checks the validity of the assumption and withdraws the solution if the justification is weak.
- Reporting. A separate system is used to present the combined diagnostic solution in a structure that can easily be understood by the operator. The reporting system includes relevent information taken from the data acquisition system for future reference.
- Operator interface. This allows access for the operator. The operator can view the current operating performance directly and also look at the results of the diagnostic system. The operator can interrogate the reporting system to determine the methods that have been used to form the diagnosis.

The whole system is controlled by a stratagist system that has access to all the data on the blackboard and controls the requests for action by all the sub systems. For instance it will first request live data from the engine, it will then ask for simulation results at the current ambient state extracted from this data. These two sets of results are compared by the feature extraction module. Results from this may cause the strategist to request the set comparison system and inference system to suggest reasons for any discrepency. If contradictory results are fed back the the truth maintenance system will attempt to find the most plausible cause. The operator is then alerted and the cause is reported.

2.4.2. GEC Alsthom Regulateurs Europa - Fault detection and diagnosis system (FDDS)

This system is described by Poppe [11, 12] of GEC. It is a first generation prototype developed by a consortium including GEC Alsthom, Ricardo Consulting Engineers and the MOD. It is a development of Ricardo's own Diesel Engine Intelligent Monitoring System (DEIMOS) described above. It is a PC-UNIX based monitoring system that uses heuristic algorithms enhanced with fault symptom matrix (FSM) techniques. This enhancement is used to

give quantitative rather than qualitative output. The diagnosis is based on the values of about 40 performance parameters. The rules that define the expected relationships between performance parameters and fault states are established from the knowledge of human experts with the help of computer simulation. It is finely tuned, or fingerprinted, with test data from the actual engine to which the system is being applied.

The engine operator uses the system as a tool to aid their understanding of the current health of the engine. The operator can select different ways of viewing the data. This way they are able to see past, present and predicted parameter values, trend graphs and plots of parameter deviations from healthy conditions. This information can be viewed in tabular or mimic form. It is also possible to see the results of the fault diagnosis system including the justification of any diagnosis.

2.4.3. Asea Brown Boveri - CYLDET cylinder pressure monitoring for Diesel and Gas engines

Cyldet[13] is a PC based system used for the continuous monitoring of the combustion chamber presssure of Diesel and gas engine installations. The system is capable of monitoring up to 20 cylinders per engine for a maximum of 10 engines. The system can display the cylinder pressure *vs.* crank angle for each combustion chamber. From this pressure data, the system calculates the following performance parameters:

- start of combustion
- ignition delay
- mean indicated pressure (MIP)
- pressure at top dead centre
- peak pressure
- expansion pressure at 36^{o} after TDC
- indicated cylinder power

The system also monitors scavenge air pressure and various parameters of the fuel injection system. These include:

- pressure of fuel oil after the non-return valve
- crank angle at fuel pump start
- pressure at fuel pump opening

- angle at fuel valve opening
- maximum fuel pressure
- duration of fuel injection in degrees of crank angle
- quantity of fuel injected per stroke

Cyldet checks that all the values are within acceptable limits set by the operator. If a parameter goes out of limits the operator is warned and a signal can be sent to the engine management system to automatically alter the engine operating state. For instance MIP can be used to signal engine overloading.

All the data is stored and can be used for trend analysis using graphical displays of the historical data.

An auto diagnostic system is avaliable that compares current values with theoretical values for the current operating state. The system uses heuristic logic to provide reasons for poor combustion and to give advice on how to remedy the situation.

2.4.4. Marconi Command and Control Systems Ltd & Lloyds Register of Shipping - Condition Monitoring and Predictive Systems (CPMPS)

Described by Davies [14] and Katsoulakos et al [15], the CPMPS project was established to develop an on-line marine engine monitoring and management system. The project was a co-operative venture between Marconi, Lloyds Register of Shipping, the University of Newcastle and Humberside College of Higher Education. The sponsor was the DTI. The system was to have 5 main functions:

- To assess the condition of the engine
- To diagnose faults
- To monitor performance and efficiency
- To optimise performance
- To plan maintenance activities

The heart of the CPMPS system is a pattern matching system that relates current operating performance with simulated reference data for both healthy and faulted cases. Causal analysis is used to ensure that the faults, or combination of faults, that are proposed are possible and logical. The reference data is provided by a polynomial based engine model reference

generator for the healthy data and a re-configurable thermodynamic simulation for the fault states. The expert system can request the simulation of certain faults at the current operating point for accurate comparison with the real engine data.

The heuristic rules defining the cause and effects of different faults were written based on historical data provided by Lloyd's Register of Shipping. The actual performance data was gathered from a test engine at Newcastle University. The rules that make up the expert knowledge base were then generated using an automatic rule induction process.

The development tool used to integrate the various expert system techniques was MUSE. This system provides a *blackboard architecture* that allows each knowledge source to make its own contribution to a central decision making process.

The experimental system was constructed using a variety of hardware including Sun workstations, a DEC VAX system and PC's. A complete integrated system has however been designed to conduct on-board sea trials. This is a shared memory system that uses a Motorola 68020 processor to run the expert system and thermodynamic simulation and includes an IBM 286 processor to run data acquisition tasks.

2.4.5. Power Group International Ltd - Remote condition monitoring of standby Diesel generators

This system is discussed by Perryman[16]. The system described is used to remotely monitor standby generators in order that they may start reliably when called upon to provide emergency power.

The system comprises of three 80C196 micro-controllers on the generator set. These are connected to a display board and modem. The boards have various analog and digital inputs and outputs connected to transducers and actuators on the generator set in order to monitor and control it. The system stores the data that is collected at rates that vary between 1 sample per second to 1 per hour depending on the operating state of the plant. Each time the plant is run, or if a signal goes out of limits, the data is communicated via a modem and telephone lines to a central monitoring station.

The central monitoring station is equipped with a network of Apollo workstations. The workstations can be connected to several generating sets at any one time. The system is capable of monitoring up to 100 hundred sets in total. At the workstation the operator is able to interrogate the current

operating state of a particular generator set by logging onto it via the modem link. It is also possible to look at the entire performance history of the unit as all the performance logs are kept at the central monitoring station. Performance patterns and trends can therefore be investigated. The operator also has the ability to remotely start and run the generator set to ensure that all is well.

The group has suggested that considerable savings in operating costs can be made if a condition based maintenance strategy is employed rather than the preventative maintenance methods conventionally employed on emergency generator sets. The large amounts of data that their system gathers on the generator sets should allow reasonably accurate prediction of faults. A separate PC has been implemented at the central monitoring point for analysis of the large data histories that are accumulated on the generator sets using expert system techniques.

The PC system uses conventional heuristic reasoning methods to determine the causes of performance deviations. The expert system input information is based on the patterns of the discrete measured values including digital alarm states (e.g. on/off) and categorised analogue values (e.g. high/normal/low).

The group highlights that these techniques are limited on their dependence of predetermined rules laid down by human experts. Perryman suggests that this constrains the system to the human level of expertise. The group's system therefore supplements the standard rule base with new rules that are made up using statistical *machine learning* algorithms whenever a new pattern of behaviour is encountered. The operator is still needed to provide an explanation of the new pattern but the implementation of the new rule in the system is automatic.

2.4.6. Wartsilla - Fault avoidance Knowledge System (FAKS)

FAKS [17] is an Apple Macintosh based diagnostic system for the VASA 32 and the VASA 46 marine Diesel installations. Every 15 minutes the system automatically measures various performance parameters such as temperatures and pressures around the engine. about 10 transducers per cylinder are used. These measurements are categorised into bands depending on the current operating state (e.g. high/normal/low). The data can be viewed in mimic form and normalised trends can be shown. Fault diagnosis can also be performed either manually or automatically. This mode warns the operator if the probability of a particular fault exceeds a pre-defined limit.

The diagnosis is based on heuristic analysis of the patterns of banded deviation of the performance data. The system has a knowledge base of the cause and effect of about 1000 different faults. Diagnosis can also be assisted with satellite links to company headquarters so that diagnostic information can be exchanged. The system has been evaluated on board the *Railship III* with a system that monitors the 5 engine installations using about 300 transducers.

2.4.7. Renk AG, MAN, B&W, AG and AEG, - ADES, Modis Geadit and GEDIS

ADES [18] is a comprehensive expert diagnostic system that integrates individual diagnostic systems such as the Diesel engine diagnostic system, Modis Geadit, and the gearbox diagnostic system GEDIS. Modis Geadit is a PC based diagnosis and trend analysis program for large marine Diesel installations. The PC monitors data captured by on-line monitoring plant, or off-line, by manual input by the operator. A major part of this system is combustion analysis conducted on pressure traces taken from the combustion chambers. GEDIS is an expert system that uses knowledge of the important components such as gear teeth and bearings, coupling and oil system and of operating parameters such as temperatures vibration and noise. ADES takes all the information from these systems and uses inference mechanisms based on a separate knowledge and rule database to perform fault diagnosis, trend analysis functions and to produce maintenance instructions.

2.4.8. MaK, - DICARE

DICARE [19], is a PC based expert diagnostic system currently installed on about 68 engine installations. The system monitors the performance values of speed, power and load as well as pressure and temperatures around the system. The program relates deviations of performance, adjusted according to ambient conditions, with fault causes using heuristic analysis. Data is stored every 2 seconds and the historical data can be used for manual trend analysis by plotting graphs.

2.4.9. Cummins Engine Company Inc., - Compuchek

Compuchek [20] is a workshop based Diesel engine diagnostic tool developed by Cummins for their 6 cylinder 14 litre Diesel engines. It is capable of identifying up to 50 faults in areas of cylinder performance, fuel pump set up, injector set up, combustion air delivery, lube and cooling

systems. The system recommends maintenance procedures based on analysis of direct measurements taken from pressure and temperatures around the engine using simple causal logic. It is able to infer in-cylinder performance by comparing measured crank shaft speed wave forms with predicted values generated by a computer model. This technique allows cylinder to cylinder comparisons in power contribution, compression powers, torque contribution and piston blow-by.

The system has gained wide commercial acceptance and is installed in 90 locations in North America.

2.5. The history of neural networks and their application into diagnostics

Neurologists have been using computer and electronic methods of simulating brain cell behaviour since the 1950s. The earliest work includes Rosenblatt's work on a device he named the perceptron and the development of a system called Adaline by Widrow. These systems tried to mimic the highly parallel nature of the brain. The structure and programming of these systems were based on the Hebbian principles of learning [21] established in 1949. The brain's power lies in the transmission of signals through a highly inter-connected network of neurons. Hebb proposed that repeated stimulation of inter-neuron connections will increase the sensitivity of transmission of the signal between them. By altering the sensitivity of the neurons the brain is able to produce different responses to different patterns of input.

The non-linear nature of the neural nodes and their highly interconnected nature meant that the techniques are able to cope well with non-linear relationships. This property has meant that they were applied to engineering problems from the outset although there was great scepticism because of the difficulty in numerically analysing the behaviour of the large analogue networks. Freeman [22] states that general research in the subject was stifled by the sceptical analysis of the perceptron by Minsky and Papart in 1969 [23]. In 1986 the subject was considerably revitalised by the publishing of Parallel Distributed Processing by Rummelhart and McClelland [24]. Since then research in the field has flourished and many commercial systems are now coming into operation that utilise neural network techniques. One of the first markets for the technology was in the finance industry where it was used to interpret the apparently chaotic systems of exchange rates and stock values. The Chase Manhatten Bank introduced a system to monitor personal bank

accounts to spot unusual spending patterns to detect credit card fraud. Neural networks have been applied to the problems of character recognition from the outset and Hall [25] describes the rapidly developing market for chip based products such as those capable of intelligent character recognition. The latest generation of notepad computers employ neural techniques to learn the owners' hand writing so that data entry can be performed by free-hand writing with an electronic pen. Similar hand writing recognition systems have been developed to enhance fax machines so that incoming faxes can be turned into computer ready format.

The process control industry has been the first to seriously utilise the technology in the engineering world but there are now many prototype and commissioned systems in use today.

The Science Applications International Corporation (SAIC) is one of the first companies to commercially use neural networks in an engineering diagnostic role with their steam turbine boiler monitoring system [26]. SAIC have been investigating applications for neural network technologies since 1986 [27] The applications include:

- data verification of measured signals used for a power station control system.
- chip based neural networks for a variety of embedded applications, including chips with on-chip learning
- data enhancement on an explosives detector for flight baggage checking
- signal separation tasks for submarine acoustic signals and satellite communications
- form reading applications
- intelligent active vibration damping systems that learn the counter motion necessary to reduce vibration

The boiler monitoring system is required to intelligently process the large quantity of monitoring data available so that only data that is currently useful to the operator is passed on. The system is designed to give prior warning of faults, improve efficiency, avoid operation in high wear states and advise on operating procedures in unfamiliar situations.

A neural network that is trained on historical data is used to provide reference data to compare with the current sensor values. Deviations are

passed to a conventional heuristic system used to determine the cause of the fault.

The neural network utilises data from 30 monitoring sensors. The model was designed to include a time domain by using 5 consecutive samples as separate inputs for each sensor.

Wu et al [28] describe how neural networks can be used for diagnostics by using them in a pattern recognition role to produce corrective remedies to faults observed in injection moulded products. A three layer network was trained to relate 20 known faults to 15 known causes. The neural approach achieved very promising results in cause identification. This approach was adopted because of its flexibility and speed of adaptation to a new situation. The network also proved to be able to generalise well around scarce training data.

Aylward et al [29] describe the advantages of a neural network approach to diagnostics on a F-15 fighter plane over conventional expert systems that have been used. They state that neural networks offer more rapid development as the complex rule data bases do not need to be made up. They also state that the neural network system coped better with the noisy data common in the aircraft's diagnostic system. The low memory requirement of the networks were also highlighted as an advantage. In addition to straight forward pattern recognition the networks included trend information by using consecutive values of the measured parameters as different inputs to the network. The networks were trained purely on simulated data but the authors propose to train future systems on data from a variety of sources including;

a) Systems integration bench test data

b) Flight components avionics and hydraulic test data

c) Flight simulation test data

d) Subsystem qualification data

e) Failure mode effects analysis

The networks produced outputs indicating both the presence of individual faults and the amount of fault. The quantitative results being a great improvement over mathematical methods that had been tried previously. The authors have also proposed methods of allowing the networks to be

expanded to include new faults while in service, as new components are added, whilst still retaining all their old knowledge.

Guillot [30] describes a diagnostic network that also operates on time domain data by performing trend analysis. The network tested had 20 inputs and 1 output with 5 intermediate nodes. The input nodes consisted of the 20 most recent samples of data extracted from a machine tool dynamometer. The output signals whether the tool is OK, a value of 0 suggests that it is healthy and 1 shows that the tool has broken. Traditional methods of monitoring machine tool equipment have proved unreliable or ineffective because of the noise that is present on measurements from such machinery. The network was trained on data that represented different feed rates of cutting and data showing the transition as the tool enters and exits the cut. The network approach yielded excellent results when tested on simulations of various tool failures.

Wasserman [31] investigated the use of neural networks to analyse vibration data to detect bearing wear and crack propagation in drive shafts. For the bearing analysis, vibration data was taken from a test rig fitted with new bearings and with worn bearings. The vibration signature is first filtered and then digitised. 8 features were extracted using time, frequency and cepstral domain techniques. These 8 features were used as the inputs to the network. 280 samples were taken from the rig in total. The network was trained using a combination of back propagation and probabilistic neural network techniques. The network achieved 100% accuracy in recognition of poor bearings in this test although field trials on other machinery have not yet been applied. The crank shaft network was also trained on real data taken from a shaft in which a crack was allowed to propagate. By regular inspection, parameters extracted from the vibration of the shaft were related to crack size by the network. The crack size was represented by dividing the output into 10 values each related to a specific crack depth. The system managed to identify cracks deeper than 4% of the shaft diameter.

Lin [32] describes a system adopted on the Space Shuttle and F8 and F100 fighter aircraft which utilise neural networks to mimic the system being monitored in order to detect sensor failures. On the Space Shuttle example, the network monitors all the sensors on the main engine. It produces an output of expected sensor values that is compared with actual sensor results in order to detect failures. A hidden layer of the network is smaller than the input and output layers so that each output can not be directly connected to

its respective input. The hidden layer therefore represents a data compression stage. the output that is reconstructed is therefore influenced by all the input values. This means that the indicated value for each sensor is dependent on the values of all the sensors that are being monitored not just the original value of the sensor itself. This system therefore allows the reconstruction of expected values of a sensor even if the actual sensor has failed. In the example given 10 sensors were monitored and so the network has 10 inputs and 10 outputs. It has 3 hidden layers of which the middle layer is the bottleneck and has only 4 nodes. Either side of the bottle neck are layers of 20 nodes each used for the data compression and regeneration. The network was trained on simulated and test data representing the engine operating over its entire operating envelope. The system proved very effective at maintaining valid sensor data for the complex non-linear control system even when 2 sensors out of 10 were corrupted. Improved performance was achieved by feeding back the predicted output of suspected faulty sensors into the input layer of the network instead of the measured signal itself.

In the automotive field, Bacon & Shayler [33] have recognised the potential of neural networks for replacing the large multidimensional look-up tables currently in use in modern electronic engine control systems (EEC's). They describe three systems to determine spark advance and injection duration from engine speed, load and air mass flow and temperature. The first system is a total neural network control system. The network was trained on data taken from an experimental engine operating a conventional EEC system. A fairly course sample of training points was used (19 points) and the network had some difficulty in interpolating between the test points in the desired manner. Some of the verification points involved extrapolation beyond the original map and the network produced poor results in this region. In an attempt to reduce these errors, the second system adopted a hybrid approach which uses neural networks to produce a correction factor to add to an output generated by a simple mathematical polynomial. This approach reduced the errors considerably. The third method was another hybrid system similar to the second system. However, in this case, the contribution from the neural network was limited to a proportion of the estimated value from the polynomial. This prevented the total output from the system being dominated by the corrective function. The authors state that the systems showed potential for actual application but further work was needed.

All the systems described so far are based on digital simulations of the neural networks. A neural node is, however, an analogue device and there have

been several developments in devising analogue circuitry containing physically connected programmable analogue devices. These can be purely electronic, as is Edinburgh University's [34] device, or photonic as described by Farhat [35]. The introduction of these devices will allow the development of neural systems that can be connected directly to a system without the need for conventional computers which require signal translation into digital format.

2.6. Conclusions drawn from the literature review

With the increased complexity of modern machinery and increased demands on them in terms of performance and efficiency, there is clearly a demand for condition monitoring equipment. It has also been widely demonstrated that there can be considerable advantages in applying artificial intelligence techniques to analyse the large amounts of data that can be extracted from such equipment.

However, only a fraction of the known AI technology has found itself onto widely used commercial diagnostic systems, most advanced systems have not progressed much further than the prototype stage. This is not only due to the time lag of production but is due in part to the complexity of the systems themselves. Diagnostic systems are becoming more complex than the plant they are monitoring and subsequently just as likely to fail !. Wood [36] identifies this as the main problem that has prevented the introduction of much needed diagnostics of Armoured Fighting Vehicles in the UK, USA and Germany.

Most systems use causal reasoning and pattern recognition techniques to interpret the signals from the monitoring system. For anything other than simple alarm signalling this requires the establishment of large knowledge data bases and very complex control algorithms for interrogation and interpolation of these data bases. All developers of heuristic based diagnostic systems accept that it is essential for such a system to be able to present its reasoning for a particular diagnosis to the operator for their interpretation of whether the logic is justifiable. This indicates that there is still a lack of confidence in the results and that these systems are still far from perfect. Neale [37] confirms this with his recommendation that at least two sources of information are required because it is relatively easy to obtain spurious readings and conclude incorrectly that either a major fault is occurring or alternatively that important trends are ignored.

This thesis suggests that there is a considerable amount to be gained from the simplification and rationalisation of the diagnostic knowledge database and control systems by the application of artificial neural network techniques. This technology has been taken up in other fields where traditional processing techniques have failed to produce results or are too slow to process large amounts of data in the required time, as in character recognition systems.

The adoption of this technology could open the door to the replacement of complex computer systems, that need a great deal of signal conditioning before digital calculations can be performed, with analogue devices that could be connected directly to the transducers, thereby greatly simplifying the overall system complexity and reducing response times.

Chapter 3

Introduction to neural network techniques

3.1. Introduction

Neural Networks are an artificial intelligence technique otherwise known as Parallel Distributed Processing (PDP). This technique has evolved from studies of the brain and its ability to solve complex problems, even though it is made up of cells that are very limited in their individual processing power. PDP simulates the brain by using an array of simple processing units to translate input data into output results. This is in contrast to conventional programming techniques that involve solving mathematical equations with a single complex processing module, see figure 3.1.

$$x = f_1(a,b,c) \quad y = f_2(a,b,c) \quad z = f_3(a,b,c)$$

Outputs x,y,z described by 3 mathematical functions

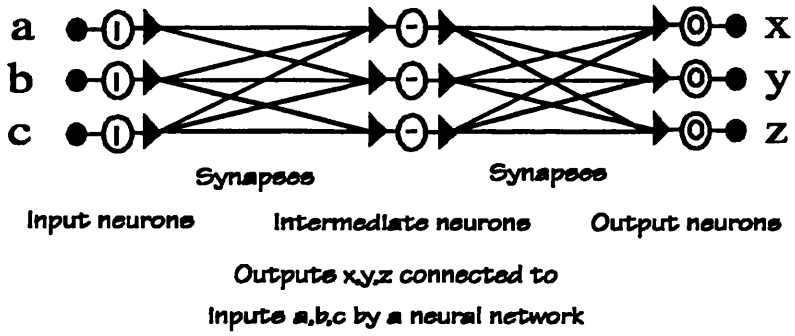
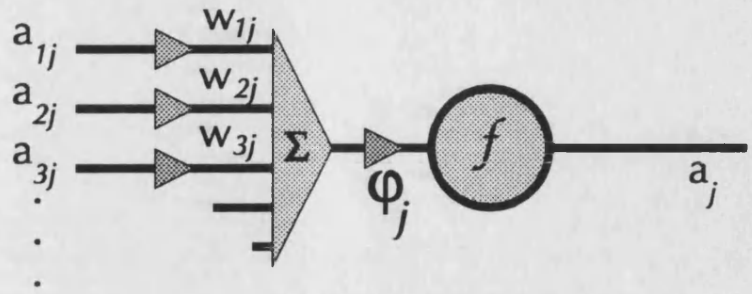


Figure 3.1 **The conventional mathematical approach vs. a neural network**

3.2. The threshold function

The ***neurons, or neural nodes,*** within the network amplify the sum of the signals at their input side. The amount of amplification is determined by a function, known as the ***forward pass function,*** the characteristic of which is controlled by a parameter called the ***bias***. This function is usually a non-linear (sigmoid) relationship as illustrated in figure 3.2. The output of each

neuron, known as its ***activation***, is transmitted to other neurons by the interconnecting ***synapses***. A synapse can amplify or reduce the strength of the signal it carries by an amount determined by a factor known as its ***weight***.



$$a_j = \frac{1}{1 + e^{(\Sigma a_{ij} w_{ij} + \varphi_j)}}$$

a = activation
w = weighting factor
φ = bias

Figure 3.2 An example forward path threshold function

By setting different biases and weights for all the neurons and synapses, different relationships between inputs and outputs can be achieved.

3.3. Complex relationships by multiple neurons

A parallel layer of multiple neurons is able to produce a non-linear output as shown in figure 3.3. An individual neuron's bias will affect the level of input required before it starts to have an effect on the output. The weight will determine the amount of influence that an individual neuron will have on the output after that point. Very complex relationships can be developed by fully connecting several layers of these parallel structures together. Formal analysis of the capability of an artificial neural network (ANN) to generate non-linear output has proved impossibly complex using formal mathematical methods. The size and architecture of a network necessary for a particular problem is therefore usually based on experimentation.

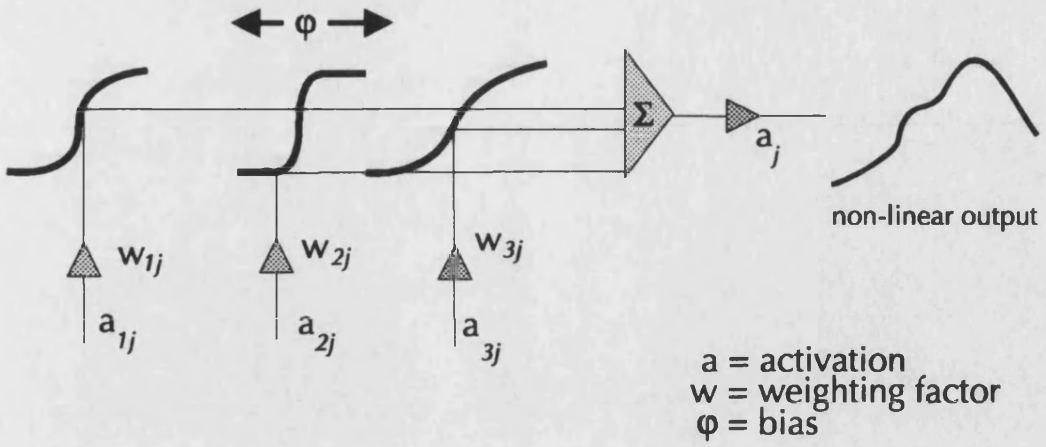


Figure 3.3 Complex output generated by multiple neurons

3.4. Training of networks using the back propagation method

The programmer does not have to specify the bias and weight of all the neurons and connections because the network calculates these during a ***training*** process.

The most common training method currently used for multi-layer ANNs is the ***Back Propagation Method***. This method is described in detail by Rumelhart[38] and the mathematical basis of this method is detailed in Appendix A. The method relies upon presenting the network with data vectors containing known inputs and corresponding outputs. The inputs are fed into the network and the errors in the outputs are calculated. These errors are monitored during the training process to assess how well the network is learning the relationship. All the weights and biases are re-calculated to reduce the mean error, this iteration is known as a ***training epoch***. The input vectors are presented repeatedly until the ***training error*** is satisfactory.

3.5. Limitations of the standard back propagation method and some potential solutions

Each iteration of the back propogation method relies on reducing the training error. A multi- level ANN is highly non-linear, therefore the training process is also highly non-linear. The training can therefore get locked into a local minimum. The back propagation will not allow the increase in error necessary to climb out of this situation. This situation is usually overcome by randomising the network and starting the training process again.

Because the standard back propagation method is iterating towards the minimum error, it may overshoot the desired minimum error solution and become unstable. For this reason there is control of the amount of adjustment allowed per iteration. The parameter used to control this is known as the learning rate. It is up to the programmer to set this rate. This is usually determined by experimentation.

Variants of the standard back propagation method have been designed to tackle some of these problems. These methods include; the ***stochastic back propagation*** method, the ***quick propagation*** method, proposed by Scott Fahlman, and the ***Haffner modified back propagation*** method. The stochastic method adjusts the weights and biases after each individual input vector is presented rather than accumulating the errors from all the input vectors of the training data. The order in which the sets are presented is changed each cycle in a random way. This method should stop the solution becoming trapped or unstable. The Haffner and Scott Fahlman methods monitor the training progress and are therefore able to adjust the learning rates dynamically. These rates are designed to progress to a stable solution as quickly as possible.

The most appropriate method for a particular solution is usually found by experimentation.

3.6. The benefits of neural networks for diagnostics

Neural networks are able to store and regenerate complex non-linear relationships very simply and efficiently. They are ideal for use when historical data is readily available and where conventional mathematical methods are too complex or cumbersome to be used on-line. Neural networks have found many successful applications, ranging from money transaction analysis to process control, as described in the previous chapter.

The mechanism of neural net processing means that the system naturally interpolates between the known data points and will produce output patterns even if portions of the input data are missing. This ability has proved particulary useful in the control industry where neural networks have coped well with loss of instrumentation used in the control loop. They have also been shown to cope well with the noisy data that is common in this industry. This was highlighted by Aylward in his experiences with experimentation with neural networks on the F15 fighter aircraft. It is however accepted that a networks ability to interpolate and cope with noisy data is limited if training

is allowed to progress too far. The ideal amount of training necessary is again an amount determined by experience and experimentation.

Neural network techniques have proved to be a very efficient way of storing information. The human brain seems to be capable of storing a lifetime of experiences in a finite number of cells. An ANN can be defined by a simple list of weights, biases and interconnection architecture of all of the neurons. This takes up much less space than the training data set that the network has learnt. These techniques are now used in data compaction systems because of this reason. In the previous chapter it is shown that neural networks were used to predict instrumentation failure on the space shuttle because it was the most efficient and simplest way of storing the knowledge of what values would be expected at any particular operating point.

ANNs are also very versatile and can be reconfigured to a new set of knowledge very quickly. This has proved particularly useful in the control and diagnostic applications.

The relationship of the performance of complex systems, such as the combined heat and power plant and Diesel engine, to its state of health is a complex problem ideally suited to neural net processing. It is not always known exactly what the cause and effect of a particular fault, or group of faults might be. Also the data that is used for the analysis is often subject to limitations in experimental accuracy and noise. Prohibitively large rule bases would be needed to describe the effects of all known faults at all known ambient conditions at all operation points using conventional heuristic methods. Mathematical simulations that can be used to generate the effects under the known conditions tend to be extremely complex and difficult to control. Neural networks offer the opportunity to do complex performance analysis with limited processing resources.

Chapter 4

The test facilities used for this project

4.1. Introduction

This chapter describes the equipment used for this research project to evaluate the diagnostic tool's ability to deal with data from real engineering plant. Test data was collected from an experimental combined heat and power plant (CHP) at the British Gas Midland Research Centre and a Diesel engine test cell located at the University of Bath. The chapter details the type of engine, the instrumentation and the signal conditioning. It also describes the computer data acquisition hardware used, although the software is discussed in the next chapter.

4.2. The British Gas CHP test unit

This experimental plant is located at the British Gas West Midlands Research station in Solihull. It consists of an integrated unit containing a 2 litre Ford Sierra engine converted to run on natural gas. This produces useful electrical power via a 3-phase electric generator which is patched into the site electrical system. The engine coolant system is modified to retrieve as much heat as possible from the engine and its exhaust and oil using a set of heat exchangers. The heat is passed from the plant to the site water via a water-water heat exchanger. Figure 4.1 shows a schematic view of the system and also shows the location of transducers used to monitor the system. A full list of transducers is given in table 4.1.

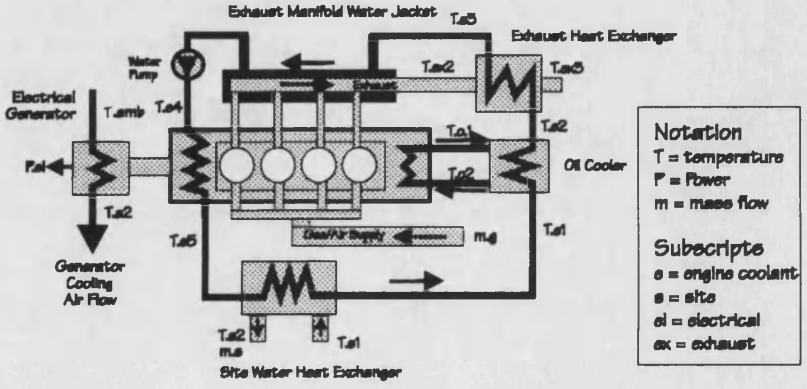


Figure 4.1 Schematic view of the CHP showing measuring points

Label	Description	Type
T.e1	Engine coolant temp after site HX	PRT
T.e2	Engine coolant temp after oil cooler	PRT
T.e3	Engine coolant temp after exhaust HX	PRT
T.e4	Engine coolant temp after exhaust manifold jacket	PRT
T.e5	Engine coolant temp after engine block	PRT
T.ex2	Exhaust gas temp after manifold	Thermocouple
T.ex3	Exhaust gas temp after exhaust HX	Thermocouple
T.s1	Site water entering site HX	PRT
T.s2	Site water leaving site HX	PRT
T.amb	Ambient air temperature	PRT
T.a2	Generator coolant air temperature	PRT
T.o1	Oil temperature entering cooler	PRT
T.o2	Oil temperature entering block	PRT
m.g	Gas mass flow	
P.e	Electrical power out of generator	

Table 4.1 Transducers used on the CHP plant

Data is gathered by a PC fitted with an analogue and digital data acquisition card with built in signal conditioning. The data is gathered hourly and represents an average value for the hour. The data is stored in standard ASCII files on the PC. The processing of this data is described in the next section on software. This system can only analyse historical data and therefore, for this project, is used to investigate different neural network architectures.

4.3. The Leyland TL-11 Diesel test cell

At the University a Diesel engine test facility has been developed by the author for the validation of computer simulation techniques and the development of diagnostic systems.

4.3.1. The TL-11 test engine

The test facility is based around a highly instrumented turbo-charged and inter-cooled six cylinder, direct injection, Diesel engine. The engine is a

Leyland TL-11 truck and bus engine, rated at 190 kW at 2100 r/min. It is illustrated in figure 4.2 below.

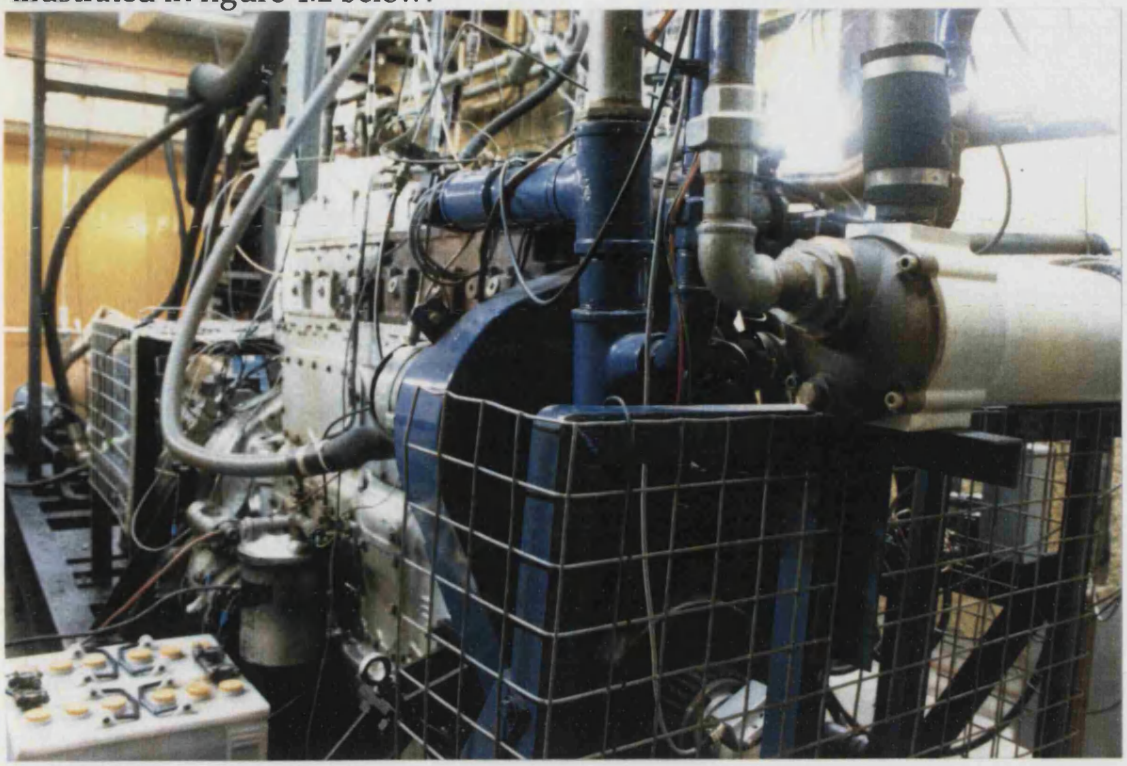


Figure 4.2 The Leyland TL-11 diesel engine

The test engine has had several modifications made to it to increase its flexibility as a research engine. These include:

Turbo-charger. An instrumented turbo-charger with known performance characteristics has been supplied by *Garrett Air Research*. The compressor's nose cone and casing have been drilled through to allow optical measurement of turbine speed.

The mechanical fuel injection pump. This has been altered to include a timing advance/retard facility. This enables active control of the timing to +/- 6^{o} either side of the standard 22^{o} before top dead centre (TDC) injection point.

The inter-cooler. This has been replaced by a high capacity cooler with a very low air resistance. Water and air flow through it can be controlled by gate valves. This allows complete control of the inter-cooler characteristics. The effectiveness can be altered over the range of 0-99%. The pressure drop at full load can be varied from about 15 mbar to ∞.

Engine cooling. This is acheived using a thermostatically controlled heat exchanger with controllable water flow.

Air Inlet and Exhaust Outlet. These have butterfly valves on them to allow control of inlet depression and exhaust back pressure.

Dynamometer. The hydraulic dynamometer has a control system that allows the engine to be run in 3 modes:

- **Constant speed mode**
- **Constant torque mode**
- **Windage mode** in which torque increases as a square of the speed.

Windage mode simulates the resistive loading that a vehicle would experience as its speed increases. The gain of the torque - speed relationship can be altered.

These modifications allow the controlled simulation of various faults that occur in real engines in service, such as fouled, or leaking, air and water passages, and poor fuel injection timing.

4.3.2. The instrumentation on the TL-11

The test cell is fitted with a large number of transducers that measure the thermodynamic processes occurring within the engine. The measuring points are shown in the schematic figure 4.3 and are detailed in table 4.2.

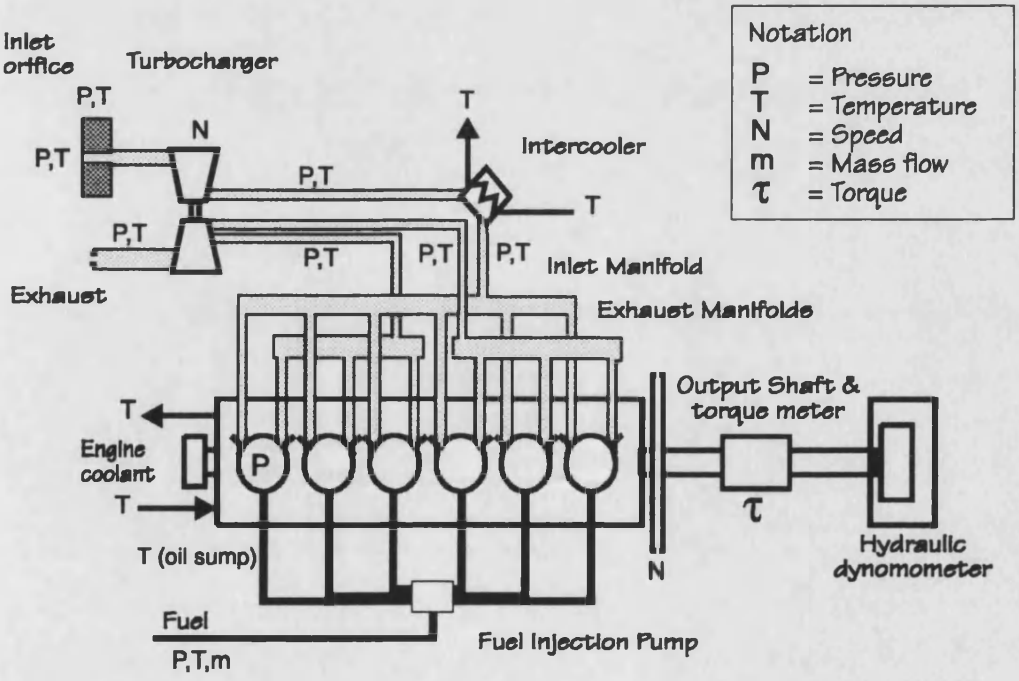


Figure 4.3 A schematic view of the Leyland TL-11 Diesel engine

Parameter	Transducer type	Range
Crank Speed	AVL 3060C/600 Optical encoder 0.2°/pulse	2200 rpm
Turbo Speed	Optical couter 2 pulses/rev and Orbit tachometer	100000 rpm
Load Torque	Vibrometer brushless torque meter	0-1000 Nm
Fuel mass & volume flow & density	EXAC harmonic metering device EX8100	0-20 g/sec
Timing advance/retard	Potentiometer feedback	+/- 6°
In-cylinder pressure (cyls 3 & 6)	Kistler 6121 Piezo-electric and 5007 charge amp	0-200 bar
Needle lift (cyls 3 & 6)	Bentley nevada inductive cap. proximeter	
Ambient pressure	Druck absolute pressure transducer	0-1.6 bar
Inlet depression	PDCR 920/7WL differential transducer	0-175 mbar
Differential air flow orifice pressure	Furnese differential transducer FC040 Mk2	0-50 mbar
Compressor outlet pressure	Druck gauge pressure transducer FC040 Mk2	0-1.6 barg
Differential intercooler pressure	Furnese differential transducer	0-50 mbar
Inlet manifold pressure	Druck gauge pressure transducer	0-1.6 barg
Front exhaust manifold pressure	Druck gauge pressure transducer	0-1.6 barg
Exhaust manifold differential pressure	Furnese differential transducer FC040 Mk2	0-25 mbar
Turbine outlet pressure	PDCR 920/7WL differential transducer	0-175 mbar
Front exhaust temperature	K-type thermocouple	0-1000 C
Rear exhaust temperature	K-type thermocouple	0-1000 C
Compressor outlet temperature	K-type thermocouple	0-1000 C
Turbine uutlet temperature	K-type thermocouple	0-1000 C
Exhaust stack temperature	K-type thermocouple	0-1000 C
Dyno oil cooler temperature	K-type thermocouple	0-1000 C
Dyno oil tank temperature	K-type thermocouple	0-1000 C
Engine coolant temperature into block	K-type thermocouple	0-100 C
Engine coolant temperature out of block	K-type thermocouple	0-100 C
Air box temperature	K-type thermocouple	0-100 C
Cell temperature	K-type thermocouple	0-100 C
Fuel pump outlet temperature	K-type thermocouple	0-100 C
Sump oil temperature	K-type thermocouple	0-100 C
Inlet Manifold temperature	K-type thermocouple	0-100 C
Oil cooler water outlet temperature	K-type thermocouple	0-100 C
Air cooler outlet temperature	K-type thermocouple	0-100 C
Fuel tank temperature	K-type thermocouple	0-100 C
Intercooler water inlet temperature	K-type thermocouple	0-100 C

Table 4.2 Transducers used on the TL-11 test engine

4.3.3. The signal conditioning and accuracy of the test data

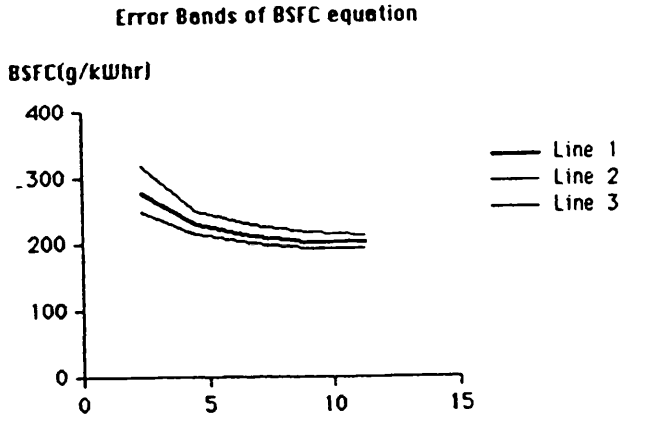
The signals are conditioned to a standard +/-5 V level outside the test cell using high integrity instrumentation amplifier boards built at the University. All channels achieve a total accuracy of +/- 1% of full scale deflection using this system.

The importance of keeping instrumentation errors as low as possible is highlighted in the MPhil thesis by the author [39]. The thesis shows how the accuracy of parameters dependent on several inputs can be degraded by compounded instrumentation errors, especially when the instruments are measuring at the lower limits of their range. Figure 4.4 shows a typical result of this analysis. The figure shows the range of the potential error in the

calculation of brake specific fuel consumption (BSFC), a parameter that is dependant on 3 inputs; *torque, speed* and *fuel mass flow*. The figure shows the actual value of BSFC calculated at 5 operating points over the engine's entire operating range plus the maximum and minimum values that could be expected due to instrumentation errors compounded in the calculation. The errors included in the analysis are of the three types outlined below:

- *Gain* where the error is proportional to value
- *Linear* where the value deviates within a fixed band either side of the true value
- *Offset* where the error is constant over the whole range.

As can be seen in the figure, the accumulated effect on the final solution can be quite significant even though very small instrumentation errors are assumed.



Initial Value Table for BSFC equation

	x-axis				
BMEP (Bar)	2.341	4.522	6.783	8.998	11.26
	y-axis				
Torque (Nm)	204.9	395.8	593.7	787.6	985.5
Speed (rpm)	1501	1502	1501	1500	1502
Fuel Flow (g/s)	2.5	3.987	5.52	7.04	8.74

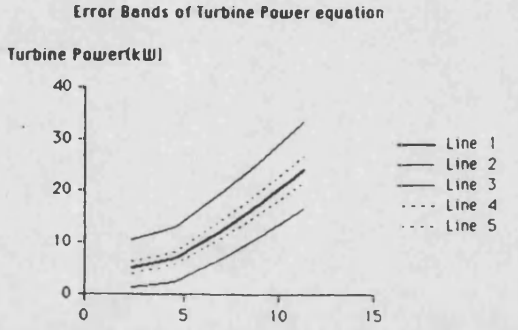
Error Matrix for BSFC equation

	E-gain (%)	E-linear	E-offset
BMEP (Bar)	0	0	0
Torque (Nm)	1	5	14
Speed (rpm)	.03	0	0
Fuel Flow (g/s)	2	0	0

Figure 4.4 **Possible errors in a typical derived performance parameter due to instrumentation errors**

The BSFC example is only dependent on 3 inputs. A parameter such as *turbine power* is based on 12 inputs and the compounded effect of small errors is quite dramatic, as can be seen in figure 4.5. In reality the probability of all these errors compounding in one direction is very slight, they will tend to occur in random directions. The dotted line in this figure shows the expected probable error band. These examples do, however, highlight the problems of post-processing of experimental data and the importance of keeping instrumentation errors as low as possible.

In order to keep the errors as low as possible the test cell was regularly re-calibrated and tested and any deviations logged for probability analysis.



Initial Value Table for Turbine Power equation

	x-axis				
BMEP (Bar)	2.341	4.522	6.783	8.998	11.26
	y-axis				
P.amb (Bar)	.97	.97	.97	.97	.97
P.inlet.dep.g (mBarg)	-.1254	-.2258	-.2007	-.3261	-.4767
P.air.diff (Bar)	.00512	.006259	.008325	.0106	.01317
T.inlet.dep (C)	26.99	29.53	27.94	28.65	29.32
gamma.air (-)	1.4	1.4	1.4	1.4	1.4
T.ex.man (C)	239	345	405	466	520
P.ex.man.g (Barg)	.2538	.3496	.5215	.6884	.8659
T.post.turb (C)	213.2	313.4	357.5	403.9	445
P.post.turb.g (Barg)	.00687	.01367	.01565	.01881	.02422
gamma.ex (-)	1.33	1.33	1.33	1.33	1.33
CP.ex (J/kgK)	1135	1135	1135	1135	1135
Fuel Flow (g/s)	2.5	3.987	5.52	7.04	8.74

Error Matrix for Turbine Power equation

	E-gain (%)	E-linear	E-offset
BMEP (Bar)	0	0	0
P.amb (Bar)	1	0	0
P.inlet.dep.g (mBarg)	1	0	.35
P.air.diff (Bar)	1	0	.00175
T.inlet.dep (C)	0	2	7
gamma.air (-)	1	0	0
T.ex.man (C)	0	2	7
P.ex.man.g (Barg)	1.5	.019	.01
T.post.turb (C)	0	2	7
P.post.turb.g (Barg)	1	0	.00035
gamma.ex (-)	1	0	0
CP.ex (J/kgK)	.1	0	0
Fuel Flow (g/s)	2	0	0

Figure 4.5 **Possible errors in a typical derived performance parameter due to instrumentation errors**

4.3.4. The data acquisition and communication system

For this research the output signals from the test cell were logged by a data acquisition card located in an IBM compatible 486 Personal Computer system. This acquisition system can be seen in figure 4.6 together with the signal conditioning and control rack.

The acquisition card used was a Data Translation DT 2811. This has a 25 kHz analogue to digital converter (ADC). In this application the ADC is configured to be multiplexed giving 16 channels of pseudo differential input. This gives a sampling rate of 1.5 kHz per channel. Four of these channels are externally multiplexed a further 16 ways per channel by the thermocouple signal conditioning units. This means that there are 16 x 4 = 64 channels available for temperature measurement, each giving a sample rate of 97 Hz. Control of the external multiplexing is via digital output from the DT 2811 card.

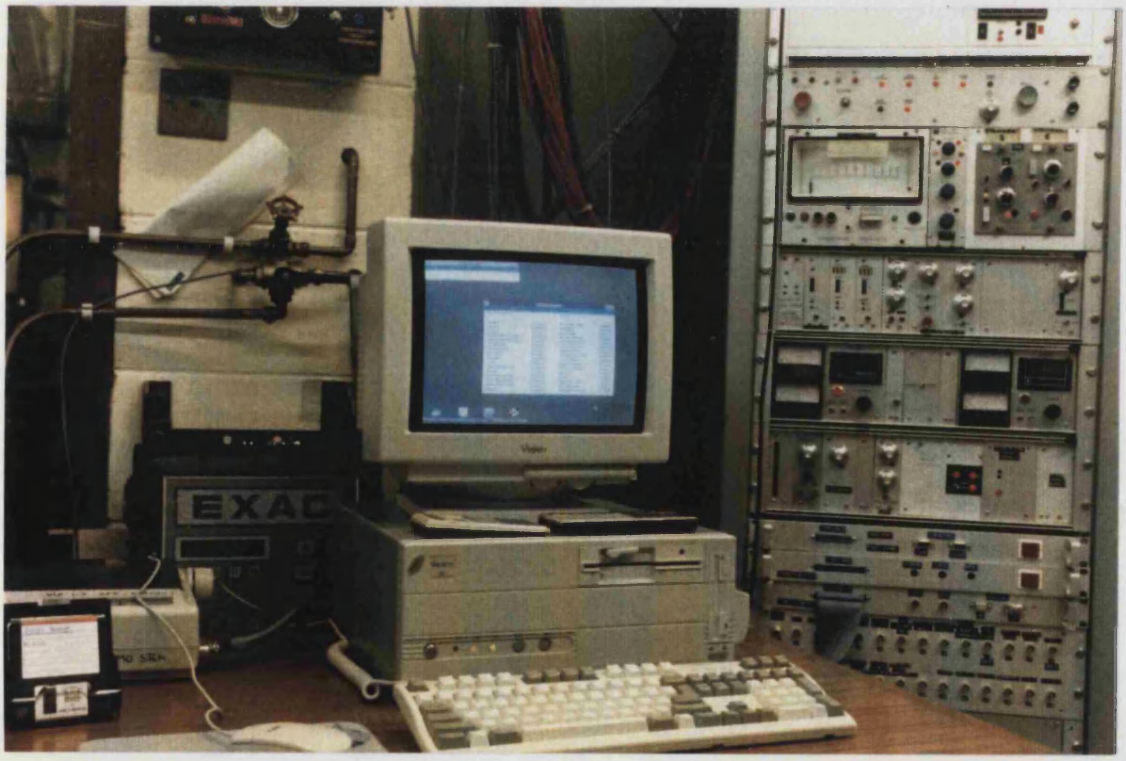


Figure 4.6 The control and acquisition system for the TL-11 test cell

The computer was running the pseudo multi-tasking graphical user interface ***windows 3.1.*** Using software written for the purpose by the author, the PC was able to communicate concurrently with the data acquisition cards, the neural network software and the high speed computer system used for engine simulation as shown in figure 4.7. This software is described in the

next chapter. The entire test facility is therefore capable of modelling a real on-line diagnostic system and process live data coming from the engine.

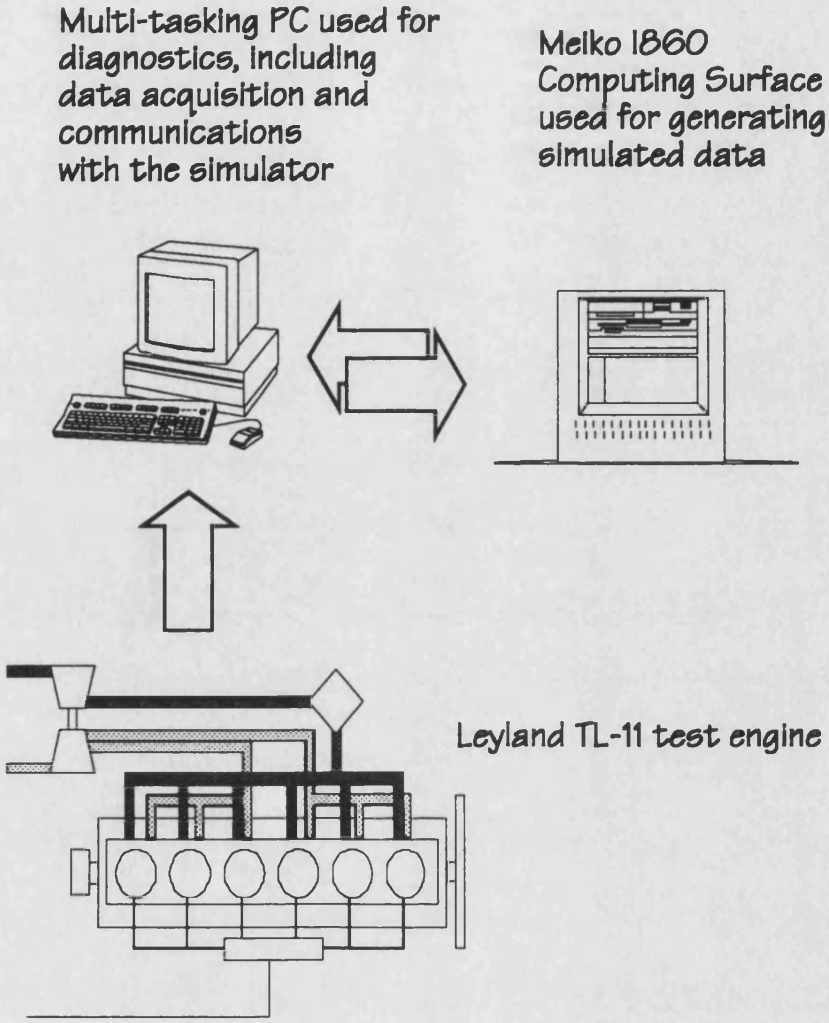


Figure 4.7 **A schematic view of the test facility**

Chapter 5

The integrated software environment

5.1. Introduction

The first aim of this project was to develop an environment for the investigation of diagnostic techniques. This section describes how a PC based multi-tasking system has been configured to make a flexible operating environment for this investigation. It describes the operating system chosen and the various software packages used to build the complete system. These include commercial packages such as ***Excel*** and ***NeuralDesk*** and a package written for this project, ***EngSpy***, that was used for the on-line acquisition of engine data from the test cell.

5.2. The *windows 3.1* environment and Microsoft *Excel v4*

The popular multi-tasking graphical user interface ***windows 3.1*** was chosen to be the platform for this research environment. This was chosen for the following reasons.

- It allows several processes to run concurrently.
- It provides a clear, well defined means of communication and data transport between packages. This allows the easy integration of several packages to work together without extra low-level programming.
- It is inexpensive.
- The hardware necessary is readily available and relatively inexpensive.
- It is well supported by compatible software packages.

5.3. Microsoft *Excel v4*

Microsoft's spread sheet package, *Excel v4,* was chosen to be the main program for data handling for this project because:

- It has a comprehensive list of built in and programmable mathematical functions
- It has an internal multi-page workbook file structure which allows simple and clear cross referencing of data.
- It is totally compatible with the ***windows 3.1*** dynamic data transfer protocols of ***dynamic data exchange (DDE)*** and ***dynamic link libraries (DLL)*** which allows live connection to external packages.
- It is user friendly, allowing rapid development of new ideas.

5.4. The data processing of the British Gas CHP data

For this research the data is imported into a spreadsheet page of an *Excel* workbook for processing. The data is first reduced to SI units and then represented in several configurations on subsequent pages. Secondary calculations, such as power transfer around the system are calculated on these pages. These are listed in Table 5.1 and detailed in Appendix B. Table 5.2 shows the typical output indicating the temperature and power distribution around the heat exchanger circuit.

Performance parameter	Units
Engine coolant mass flow	kg/s
Site water coolant mass flow rate	kg/s
Gas mass flow rate	kg/s
Air mass flow rate	kg/s
Density of engine coolant	kg/m^3
Density of site coolant	kg/m^3
Specific heat of engine coolant	J/kgK
Specific heat of site coolant	J/kgK
Specific heat of exhaust gases	J/kgK
Density of gas	kg/m^3
Calorific value of gas	J/kg
Power into plant from gas	W
Power transfer in each heat exchanger	W
Heat exchanger effectiveness for each	%
Heat loss from generator	W
Generator efficiency	%

Table 5.1 **Secondary calculations performed in *Excel* spreadsheet on data supplied by British Gas**

		Engine Coolant (tube side data)							Secondary circuit (shell side)							Effectiveness
Name	Type	m flow	Cp	m.Cp	Inlet T	Outlet	dT	Power	m flow	Cp	m.Cp	Inlet T	Outlet T	dT	Power	E
		kg/s	J/(kgK)	W/K	K	K	K	W	kg/s	J/(kgK)	W/K	K	K	K	W	-
Data from data logger																
1 Water heater(glycol)	water-water HX	1.455	3728	5424.24	361.9	347.8	-14.1	-76,482	1.6617	3998	6642.5	327.4	338.5	11.1	73,732	0.39
2 Oil cooler	oil-water HX	1.455	3728	5424.24	347.8	348.6	0.8	4,339	0.8900	1900	1691.0	369.6	367.0	-2.6	-4,397	0.12
3 Exhaust pipe	air-water HX	1.455	3728	5424.24	348.6	353.9	5.3	28,748	0.0497	1222	60.8	875.0	400.0	-475.0	-28,862	0.90
4 Exhaust manifold	water jacket	1.455	3728	5424.24	353.9	356.8	2.9	15,730	0.0497	1279	63.6	1125.0	875.0	-250.0	-15,898	0.32
5 Pump		1.455	3728	5424.24	356.8	356.8	0.0	0								
6 Engine		1.455	3728	5424.24	356.8	361.9	5.1	27,664								

Table 5.2 Example of processed data from British Gas CHP test rig

Simulated results were also generated for the CHP plant in the *Excel* workbook, this simulator is described in the next chapter.

5.5. The data acquisition from the TL-11 using *EngSpy*

Data is collected from the test engine by data acquisition cards described in the previous chapter. Software to handle this data dynamically was not available at the time this research began and so a software package ***EngSpy*** was written for this project. This software allows the control of high and low speed time coded digital and analogue data into any ***windows*** package that supports *dynamic data exchange* (DDE). The timing of data collection is set by the clock on the data acquisition card. The software uses interrupt routines to ensure that data is collected at regular intervals despite other processor demands that may be present in the multi-tasking environment. ***EngSpy*** also drives the external multiplexing of the thermocouple units using the digital output port of the DT2811 card. Because of the way the hardware is configured, 16 samples of data are taken from the standard ADC channels for every thermocouple measurement. ***EngSpy*** accumulates these values and calculates the average. ***Engspy*** also calculates the crank and turbo-charger speed from the digital counter-timer cards that are used to measure these parameters. How the data flow is controlled in the multi-tasking environment by the software is described in detail in the paper by Mobley and the author [40] given in Appendix C.

For this project ***EngSpy*** is used to import data into a spreadsheet in an ***Excel*** workbook from where it is processed in a similar way to data from the British Gas rig - except that the data in this case is live, and so is continuously updated from the engine test cell. The data is first coverted to SI units. Various performance parameters are then calculated. These are listed below in table 5.3 and detailed in Appendix D.

Parameter	Units
Brake Power	W
Brake mean effective pressure	N/m^2
Brake thermal efficiency	-
Brake specific fuel consumption	g/kWhr
Air mass flow rate	kg/s
Air/fuel ratio	-
Volumetric efficiency	-
Compressor pressure ratio	-
Turbine pressure ratio	-
Compressor density ratio	-
Turbine density ratio	-
Compressor mass flow parameter	$kg\sqrt{K}m^2/N$
Turbine mass flow parameter	$kg\sqrt{K}m^2/N$
Compressor efficiency	-
Turbine efficiency	-
Compressor power	W
Turbine power	W
Intercooler pressure ratio	-
Intercooler density ratio	-
Intercooler ffectiveness	-

Table 5.3 **Secondary calculations performed in *Excel* spreadsheet on data acquired from the TL-11 Diesel engine**

5.6. NeuralDesk v2.0

NeuralDesk is a proprietary collection of programs[41] that allows the construction and execution of artificial neural networks in the ***windows*** environment. This package was chosen because:

- It is fully compatible with ***windows 3.1*** protocols
- It allows easy construction of different neural architectures using the graphical ***NeuModel*** program.
- It can allow the training and interrogation of networks using the program ***NeuRun*** either via the self contained spreadsheet and control program ***NeuDesk*** or externally via dynamic data exchange from packages such as ***Excel.***

5.7. The integrated environment

As described above, the ***windows*** GUI allows communication between several packages running concurrently. Figure 5.1 shows how these packages

were configured for this research project. The figure shows the highly integrated system and how control is centralised into the *Excel* spreadsheet

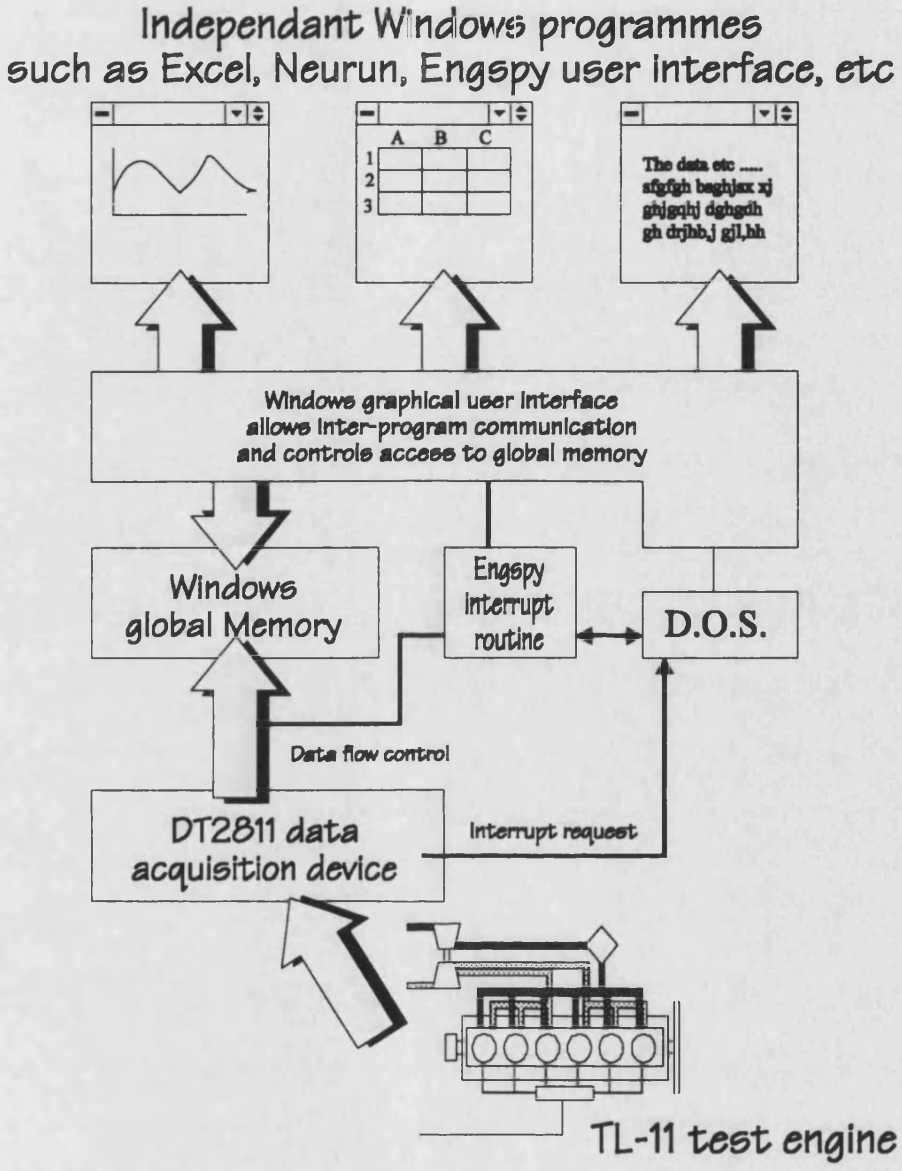


Figure 5.1 Schematic view of the multi-tasking environment

Live dynamic linking is used to pass the data from *EngSpy* into an *Excel* spreadsheet. This updates automatically as soon as new data is collected. The workbook environment is used to extract relevent data onto a separate spreadsheet for the neural processsing. Macros are then used to send the data to the neural network package *Neurun* via dynamic data exchange. These macros are instigated by *pushing* software buttons on the spreadsheet using the computer's mouse. The results of the neural processing are returned to a different area of the *Excel* workbook and are be displayed graphically in an *Excel* chart.

Chapter 6

Simulation of the CHP plant and the Diesel engine

6.1. Introduction

A neural network needs to be trained on examples of inputs and their corresponding outputs so that they can learn the relationship between them. For fault recognition systems this data needs to represent both healthy and faulted systems. For the CHP plant, the inputs needed would define the current ambient conditions and operating performance of the system. For instance the speed and power output of the engine and generator would be needed together with the temperature changes of the coolant and exhaust gases as they pass through the heat exchangers. For several reasons the availability of this data is limited. For instance, data sets can be incomplete because of the unavailability of instrumentation or it may not be possible to measure some fault states on a test rig due to the permanent damage that would be incurred. For this reason there is a need to supplement real data with data generated by simulation.

Simulators can vary greatly in complexity. They may simply be used to predict missing values in data sets or they can be a complete free standing model of the whole system.

This chapter describes two simulators used for this research. The first is a spreadsheet based model of the heat and power distributions around the CHP plant. The second is a comprehensive thermodynamic model of the IC engine. This was modified to run on a high power multi-processor machine to generate large data sets representing entire engine performance maps. The chapter describes the principles of these simulators and illustrates their effectiveness.

6.2. The CHP simulator

Data was available for this study from the plant described in the previous chapter. Much of this data, however, was insufficient due to incomplete instrumentation, or failure of existing transducers. Test data was also only available for the healthy state as the plant could not be compromised because

of other research programmes, although a known fault did develop giving one real fault state to analyse. A simple simulator was therefore written to fill in the gaps and to generate information that could mimic several faulted conditions.

The simulator is based around an interactive spreadsheet operating in the PC ***windows 3.1*** environment. It can communicate directly with the spreadsheets containing the actual engine data and the neural network package.

The simulator generates the temperature distribution around the CHP plant based on the electrical and heat loads on the system. It contains models of the heat exchangers, the engine and the alternator. It produces a solution by iteration from a set of assumed initial conditions that define the operating parameters of the models together with the power added to the system by gas and the power removed by the site water and electrical connections.

The heat exchanger models are based on their effectiveness and the mass of coolant flow through them according to the following relationship, based on analysis by Stoeker [42];

$$\varepsilon = \frac{q_{\text{actual}}}{(\dot{m}C_p)_{\min}(t_{hot,in} - t_{cold,in})} \quad \text{..... 6.1}$$

from which the exit temperature of working fluid 2, $t_{2,out}$, can be determined as

$$t_{2,out} = -\frac{\dot{m}_1 Cp_1 (t_{1,out} - t_{1,in})}{\dot{m}_2 Cp_2} + t_{2,in} \quad \text{..... 6.2}$$

where fluid suffix 1 has the lower product of $\dot{m}Cp$ of the two fluids passing through the heat exchanger.

The fluid temperature at exit from one heat exchanger is used as the input to the next.

The engine model is based on the the 1st law of thermodynamics defining the energy conservation of the engine and the mass flow of air and fuel through it such that:

$$P_{total} = P_{work} + P_{exhaust} + P_{coolant} + P_{oil} = P_{gas.in} \quad \text{.....6.3}$$

the ratios $\frac{P_{work}}{P_{total}}, \frac{P_{exhaust}}{P_{total}}, \frac{P_{coolant}}{P_{total}}, \frac{P_{oil}}{P_{total}}$ are used as inputs to the model.

The alternator model is also based on its energy conservation, in this case between heat generated and electrical power out. The air temperature rise across it is then calculated from an assumed mass flow generated by the fan using the relationship:

$$P = \dot{m} Cp \Delta T \quad \text{.....6.4}$$

The variable parameters of the model were adjusted to give a good match with the data available from the British Gas test engine.

A comparison of real and simulated data showing the temperatures around the heat exchanger circuit is given below in table 6.1.

		Engine Coolant (tube side data)							Secondary circuit (shell side)							Effectiveness
Name	Type	m flow	Cp	m.Cp	Inlet T	Outlet	dT	Power	m flow	Cp	m.Cp	Inlet T	Outlet T	dT	Power	E
		kg/s	J/(kgK)	W/K	K	K	K	W	kg/s	J/(kgK)	W/K	K	K	K	W	-
Data from data logger																
1 Water heater(glycol)	water-water HX	1.455	3728	5424.24	361.9	347.8	-14.1	-76,482	1.6617	3998	6642.5	327.4	338.5	11.1	73,732	0.39
2 Oil cooler	oil-water HX	1.455	3728	5424.24	347.8	348.6	0.8	4,339	0.8900	1900	1691.0	369.6	367.0	-2.6	-4,397	0.12
3 Exhaust pipe	air-water HX	1.455	3728	5424.24	348.6	353.9	5.3	28,748	0.0497	1222	60.8	875.0	400.0	-475.0	-28,862	0.90
4 Exhaust manifold	water jacket	1.455	3728	5424.24	353.9	356.8	2.9	15,730	0.0497	1279	63.6	1125.0	875.0	-250.0	-15,898	0.32
5 Pump		1.455	3728	5424.24	356.8	356.8	0.0	0								
6 Engine		1.455	3728	5424.24	356.8	361.9	5.1	27,664								
Simulated data																
1 Water heater	water-water HX	1.455	3728	5424.24	361.2	347.3	-13.8	-75,024	1.6700	3998	6675.8	325.7	337.0	11.2	75,024	0.39
2 Oil cooler	oil-water HX	1.455	3728	5424.24	347.3	348.1	0.8	4,186	0.8900	1900	1691.0	368.0	365.5	-2.5	-4,186	0.12
3 Exhaust Heater	air-water HX	1.455	3728	5424.24	348.1	353.5	5.3	28,953	0.0497	1223	60.8	878.0	401.6	-476.3	-28,953	0.90
4 Exhaust manifold	water jacket	1.455	3728	5424.24	353.5	356.4	2.9	15,767	0.0497	1279	63.6	1125.9	878.0	-248.0	-15,767	0.32
5 Pump				0			0.0									
6 Engine		1.455	3728	5424.24	356.4	361.4	5.0	27,314								
%age difference of real to simulated data																
1 Water heater	water-water HX	0.0%	0.0%	0.0%	-0.2%	-0.1%	-1.9%	-1.9%	0.5%	0.0%	0.5%	-0.5%	-0.5%	1.2%	1.7%	-1.0%
2 Oil cooler	oil-water HX	0.0%	0.0%	0.0%	-0.1%	-0.1%	-3.7%	-3.7%	0.0%	0.0%	0.0%	-0.4%	-0.4%	-5.0%	-5.0%	1.9%
3 Exhaust Heater	air-water HX	0.0%	0.0%	0.0%	-0.1%	-0.1%	0.7%	0.7%	0.0%	0.1%	0.0%	0.3%	0.4%	0.3%	0.3%	0.0%
4 Exhaust manifold	water jacket	0.0%	0.0%	0.0%	-0.1%	-0.1%	0.2%	0.2%	0.0%	0.0%	0.0%	0.1%	0.3%	-0.6%	-0.6%	0.1%
5 Pump																
6 Engine		0.0%	0.0%	0.0%	0.0	0.0	0.0	-1.3%								

Table 6.1 A comparison of real and simulated data

The various efficiencies and power distribution balances can be altered to mimic faults in sub systems of the plant. The simulator then calculates the new temperature distributions around the system.

For this research ten fault states on the CHP plant were investigated. These were:

- Healthy
- Timing advanced
- Timing retarded
- Mixture rich
- Mixture lean
- Alternator fault
- Site heat exchanger fault
- Oil heat exchanger fault
- Exhaust heat exchanger fault
- Exhaust manifold heat exchanger fault

The first four faults affecting timing and mixture were simulated by changing the gas flow, air/fuel ratio and power balance of the model as determined by prior experimentation and given in table 6.2 below

	Mass flow	AFR	Power to shaft	Power to exhaust	Power to coolant	Power to Oil	Power to air
Healthy	0.00284	16.5	30 %	38.5 %	20 %	3.3 %	8.2 %
Timing Advanced	0.002561	16.5	31 %	35 %	22 %	3.3 %	8.7 %
Timing Retarded	0.002561	16.5	27 %	42 %	22 %	3 %	8 %
Mixture rich	0.002561	15	28 %	36%	24 %	3.3 %	8.7 %
Mixture lean	0.002561	18	26 %	41 %	22 %	3 %	8 %

Table 6.2 Assumed power balance effect of various faults

The alternator fault was simulated by changing the power balance of the generator from 14% (power lost to heat) to 20%.

The heat exchanger faults were simulated by changing the effectivenesses by the amounts shown in table 6.3:

	Healthy	Faulted
Site heat exchanger	39 %	30 %
Oil heat exchanger	12 %	6 %
Exhaust heat exchanger	89.9 %	50 %
Ex manifold heat exchanger	32.1 %	20 %

Table 6.3 **Assumed heat exchanger effectiveness changes due to faults**

6.3. The SPICE II Diesel engine simulator

The simulator used in this research is the thermodynamic *filling and emptying* model, SPICE II [43]. The mathematics of this model are described briefly in Appendix E.

Simulation codes such as SPICE II have been used in Diesel engine design for many years and have proved to be reasonably accurate, provided sufficient data is available to describe the system. Data, such as poppet valve effective area and turbo-machinery characteristics, has to be known if a realistic prediction of performance is to be made. In this study the research engine is an 11 litre Leyland TL-11 truck and bus engine, described in Chapter 4, the characteristics of which are well established. A schematic diagram of the SPICE II model of this engine is given in Figure 6.1. An example of the input files used to define the TL-11 and its turbo-charger are given in Appendix F.

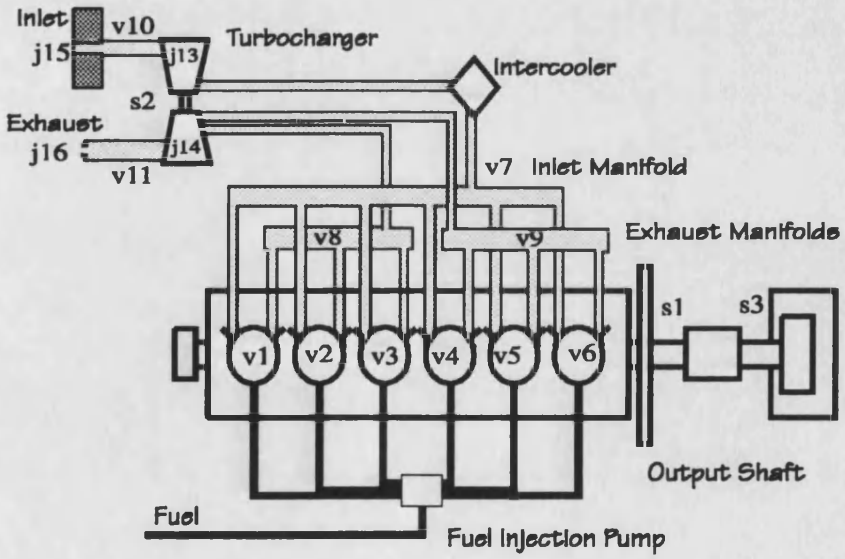


Figure 6.1 **A schematic view of the SPICE II model of the Leyland TL-11 Diesel Engine**

The model is comprised of six variable control volumes representing the cylinders, five constant control volumes representing the manifolds and the inlet and exhaust pipes. The volumes are interconnected by 16 flow junctions that represent inlet and exhaust orifices, the engine's valves and the twin entry turbocharger.

An example of the output file from SPICE II simulator is given in Appendix F. The model represents the engine with good accuracy as may be seen in figure 6.2, although a great deal of development was needed to get a good match over the entire performance envelope, as described later.

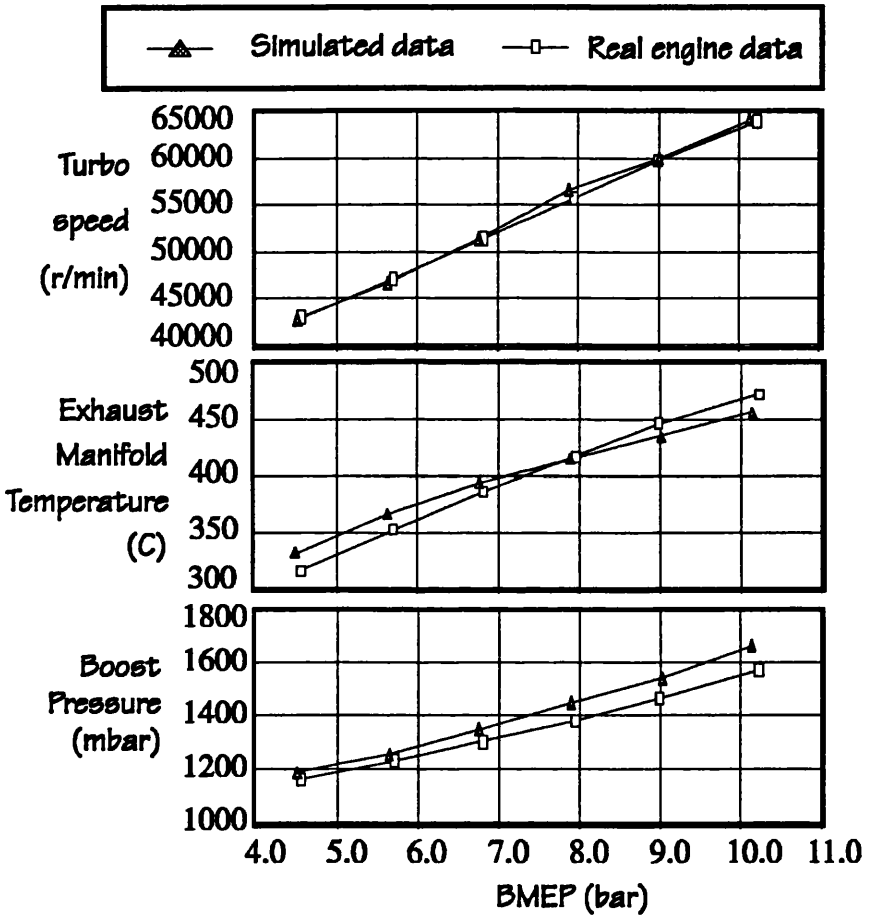


Figure 6.2 A comparison of real and simulated data at 1600 r/min

6.3.1. Multiprocessor SPICE II

Because of the large CPU requirements of the SPICE II code it is a very time consuming process to generate the large amounts of data needed to train neural networks using the standard PC version of the program. For this research the core program was adapted to run on a UNIX based multi-processor computer. The computer used was a Computing Surface *Meiko*

which is based on 16 Intel i860 processors. Each of these processors is capable of 1 MFloP performance. A suite of programs had to be written by the author to co-ordinate the batch processing operations on this machine. The programs are responsible for acquiring resources from the operating system and the generation and distribution of files to be run.

The first program constructs input files from a template that defines the engine's geometry and a separate control file which contains operating point information for each point on the map to be processed. The program saves each new input file generated and adds its name to a list of files to be processed by the batch processing controller.

The batch processing controller is a separate program that monitors processor activity and sends the next input file on the list for processing as soon as one becomes available.

Significant performance enhancement was achieved using this adaptation. A single operating point would take approximately 20-30 minutes to simulate on a 386 type machine. An entire 35 point map could be simulated in around 10-15 minutes on the multiprocessor version.

Two post processing programs were written for the multi-processor system to extract relevant data from the large amount of output that SPICE produces. The SPICE program is set to save the end of cycle results after each iteration of the simulation. The post processing packages searches for the last complete cycle and extracts the performance parameters that are also available from the real engine using the data acquisition package. The first program puts the data into a format that can be compared directly with the real data within the PC's ***Excel*** environment. The second program puts data into a format suitable for the graphics program, Unimap. Macros were written for Unimap so that 3D graphs of performance parameter *vs.* load & speed can be plotted. Some examples of these maps are shown in the next section.

6.3.2. Simulation of performance maps using SPICE II

Until this research SPICE II had only been used to simulate isolated points, or regions, of an engine's performance envelope and had not been tested over entire operating maps. The integrated environment developed for this research allowed the rapid comparison of entire performance maps and this enabled the development of a comprehensive simulation of the real engine.

SPICE II is dependent on a great deal of input data defining the engine's geometry as the input data set in Appendix F illustrates. Not all of this data is known precisely, values such as valve, and orifice, discharge coefficients can only be estimated, or determined through experimentation. It is often necessary to adjust these to get a good base-line match with real results. It proved very difficult to develop a single template that achieved a good match over the whole operating range of the engine and some additions to the basic model were needed. An example of this is with the inlet and outlets of the system. The standard practice is to use a model in which the inlet and outlet to the engine are defined as the orifice junctions of the turbo-charger. It proved impossible to establish fixed discharge coefficients for these junctions that allowed the simulated turbo-charger performance, such as speed and pressure ratio, to accurately match experimental results at all four corners of the map. The problem was overcome by adding the additional volumes (v10 & v11) and junctions (j15 & j16). These represent the air box and exhaust stack. These volumes created a more realistic simulation as the correct inlet depressions and back pressures were set up. The turbo-charger is very sensitive to these values in both the simulation and on the real engine, especially at low loads and speeds when the turbo-charger is not working hard and is basically free-wheeling. The schedule of effective areas of inlet and outlet valves also had to be carefully calculated because the geometry and discharge coefficients change as they open. The calculations used to generate the valve schedules for this model are given in Appendix G. The flow through these valves is critical on a real engine. The simulator proved equally sensitive and this makes it very important to get the correct values.

The final model simulates the real engine with great accuracy over the entire map as can be seen in the following examples of simulated *vs* real data.

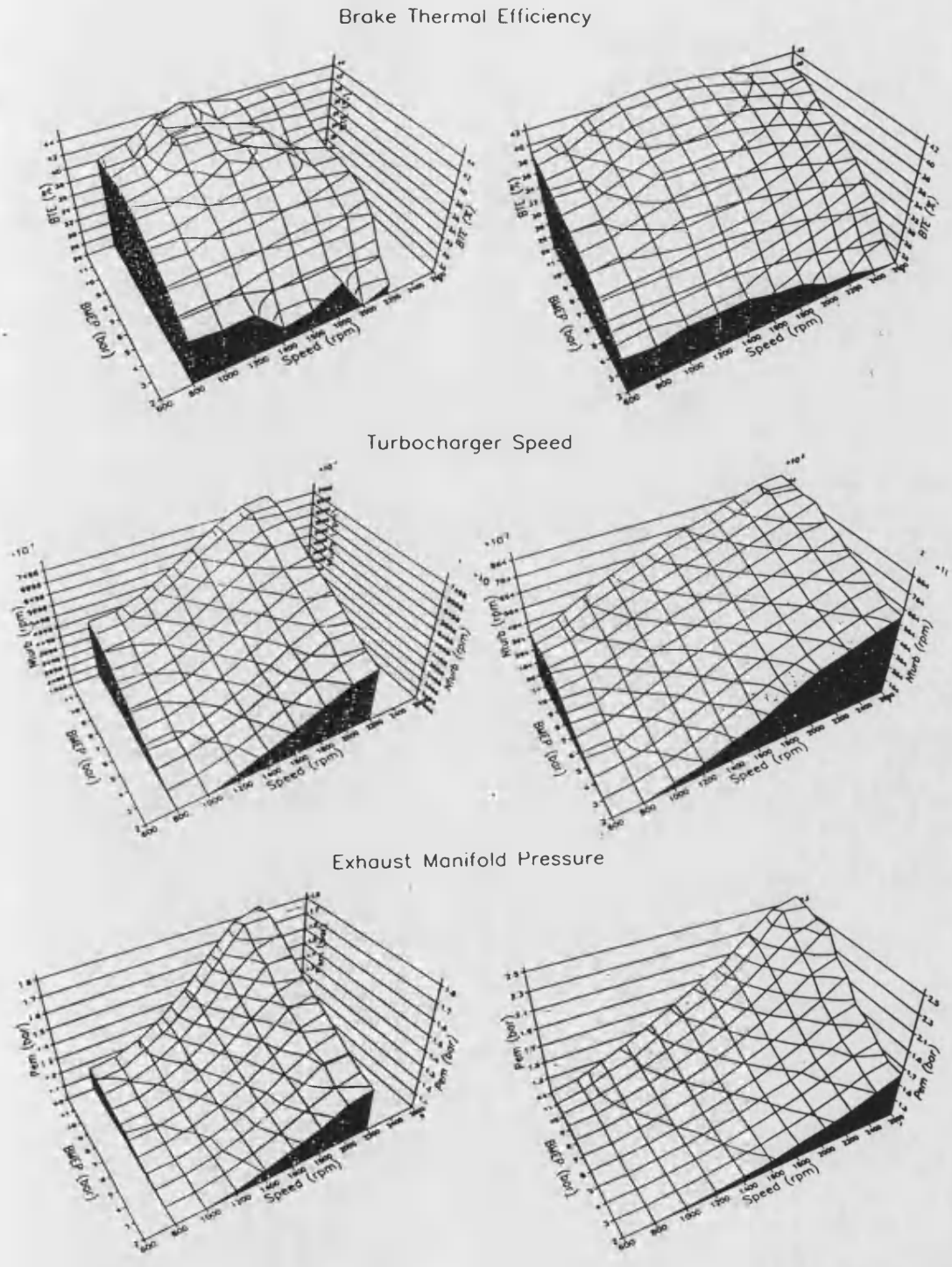


Figure 6.3 **Example performance maps of real data *vs* simulated results for the TL-11 engine**

6.3.3. Simulation of performance maps of a faulty engine using SPICE II

In this study SPICE II has been used to generate the response of the system to a range of faults over the entire load and speed range. Five faults that effect the airflow were selected. These were chosen because they may be replicated on the research engine, they were:

- Intercooler fault
- Exhaust restriction
- Exhaust manifold leak
- Inlet manifold leak
- Exhaust valve leak.

These faults will all tend to reduce air flow but the actual effect on the various performance parameters is very hard to predict, particularly those relating to the turbo-machinery, therefore if the diagnostic system is able to distinguish between them it will be a good test. These faults can also be applied to the engine without major modification. The faults were introduced into the simulation as follows:

- The ***intercooler fault*** was simulated by reducing the effectiveness from 88 % to 50 %. (The water flow can be restricted on the real engine)
- The ***exhaust restriction*** was simulated by reducing the effective area of exhaust junction (j16) from 0.01 m^2 to 0.005 m^2. (The exhaust butterfly valve can be closed on the real engine)
- The ***exhaust manifold leak*** was simulated by introducing a new orifice junction of effective area $47x10^{-6}$ m^2 (7.7mm dia) into the manifold (v9) connecting it to atmosphere. This orifice size is equivalent to about 1 % total manifold area. (A plug can be removed from the manifold on the real engine)
- The ***inlet manifold*** (v7) was altered in the same way as the exhaust manifold.

- The ***exhaust valve leak*** was simulated by introducing a new orifice junction between the exhaust manifold (v8) and the cylinder volume (v1). The effective area of this junction was 132 mm^2 which is equivalent to a permanent valve lift of 1 mm. (This can be acheived by tighteneing the tappets on the real engine, although this was not actually tried because of the potential risk of damage due to the low bumping clearances of the engine)

Figure 6.4 compares typical maps of predicted turbocharger speed for the healthy engine and the engine with a simulated leak in the exhaust manifold. It can clearly be seen in this figure how the simulator is sensitive to this change because the turbocharger speed has been considerably reduced in the higher speed and load region with the fault present. The accuracy of the simulation of the faults was tested at isolated points but not over entire maps due to risk of damage to the engine. The comparisons yielded reasonable results as shown in the later sections where the neural nets are used to compare real and simulated fault data.

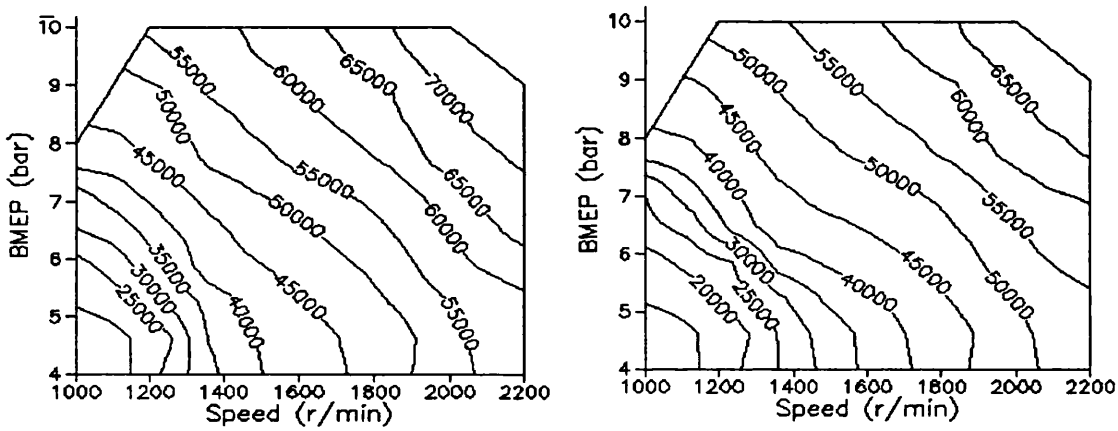


Figure 6.4 **A map of simulated turbocharger speed for a healthy engine compared with that with a leaking exhaust manifold**

Chapter 7

Alternative approaches to fault diagnostics using neural networks (CHP plant)

7.1. Introduction

This section demonstrates alternative ways that were investigated to apply neural networks to the diagnostic problem. The networks are trained on simulated data and tested using both simulated and real data. The results of these models are analysed and then the networks that would enhance the basic models are discussed and demonstrated.

7.2. The Basic Diagnostic models

Five models are tested. The input to each is the data taken from the CHP plant as shown in table 7.1. The values are normalised to the range 0-1 depending on their maximum and minimum expected values.

Gas flow
Electrical power out
Temperatures of the engine coolant water leaving:
the site heat exchanger (T. site HX out)
the oil cooler (T. Oil cooler HX out)
the exhaust heat exchanger (T Ex HX out)
the exhaust manifold jacket (T Ex Man out)
the engine block (T Engine out)
Temperatures of the site water:
entering the site water heat exchanger (T site water in)
leaving the site water heat exchanger (T site water out)
Temperatures of the engine oil:
entering the oil cooler (T Oil in)
leaving the oil cooler (T Oil out)
Temperatures of the engine exhaust:
leaving the engine (T Ex engine out)
leaving the cooled manifold (T Ex man out)
leaving the exhaust heat exchanger (T Ex HX out)
Ambient temperature
Temperature of the air leaving the alternator

Table 7.1 Input parameters for the diagnostic models

From the table above it can be seen that the parameters used for inputs to the networks are restrained to those that can be directly measured as opposed to performance parameters requiring secondary calculation. A human expert will tend to work with these secondary results because it helps him understand the system behaviour. It is felt, however, that the compounding of measurement errors through these calculations can seriously jeopardise the integrity of the diagnostic system. It has been shown in analysis by the author [39] which is highlighted in Chapter 4, that measurement errors of less than 1% FSD can produce errors in some performance calculations of well over 10%. Also if one instrument fails it may corrupt the results of several performance calculations which will lead to a confused diagnosis. Restraining the inputs to direct measurements therefore increases accuracy and helps in the tracking of instrumentation faults.

The diagnostic models fall into two categories depending on the output they produce. These are:

Health Status Recognition models	These have one output that indicates that the plant is either healthy or faulted
Fault Recognition models	These models have several outputs, each one indicating that a particular fault is present when active

The difference between the models in each category is in the network architecture and the methods used to train them.

Each model is discussed in more detail below.

7.2.1. Description of the basic diagnostic models

7.2.1.1. Model 1 - Health Status Recognition type

This model is shown in figure 7.1. The 16 inputs monitoring each measured parameter are connected to the single output, indicating system health, via one intermediate layer of 5 neurons. Different numbers of layers and neurons were tested for the intermediate levels and this configuration proved optimal for training purposes. Increasing the complexity of the hidden layers tends to lead to problems of the training getting trapped in local minima, a problem discusssed in Chapter 3. Too few neurons means that the network cannot learn the complex relationships between the performance parameters.

The network is trained on simulated healthy engine data representing several different ambient and operating conditions. These included differences in:

- **Gas mass flow** (equivalent to differing electrical demand) from 0.00276 to 0.00292 kg/s
- **Ambient temperature**, from 275 to 290 K
- **Site water power demand** from 70 to 78 kW
- **Gas calorific value**, from 4.79 to 4.83 MJ/kg

The network must have data in the training set that represents the plant not in the healthy state otherwise the connection weights will tend to increase to their maximum value during training meaning that the output will just go high for any value of input, signalling that the plant is healthy for any level of input.

For this first model the maximum and minimum values of 0 and 1 are used in the input vectors to represent the unhealthy condition. This approach assumes that the network should generate an output that peaks when the input vectors that represent the healthy state are encountered. The output should tend towards 0 as the values of the input vector move away from these recognised states.

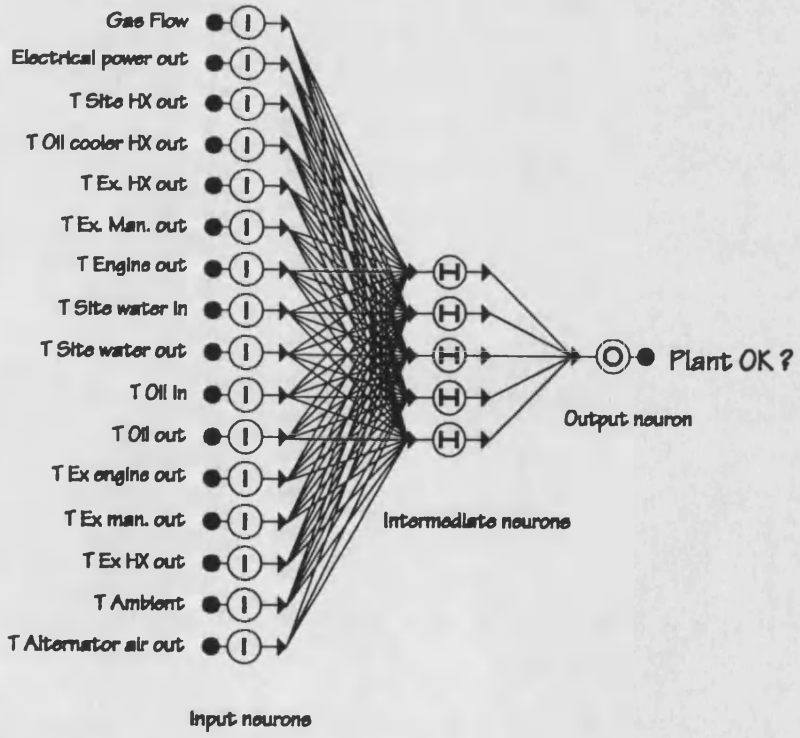


Figure 7.1 The health recognition neural network models 1 and 2

7.2.1.2. Model 2 - Health Status Recognition type

This network has the same structure as model 1. The training data used however is different. The same ambient and operating points are used to represent the healthy state but the unhealthy state in this case is represented by examples of actual simulated faults. This approach should make the network more sensitive to the subtle changes in performance that faults will incurr.

7.2.1.3. Model 3 - Diagnostic

This network is shown in figure 7.2. It has the same 16 inputs as the previous two models but has 10 outputs. The neural network in this case is trained to identify faults. 9 of these are to indicate the presence of each of the 9 test fault and one output indicates that the plant is OK, similar to the output of models 1&2. In this case only one output should be active at any one time but if all go low then this indicates that an unidentifiable fault is present.

A single intemediate layer of 5 neurons proved to be optimal for this model as well.

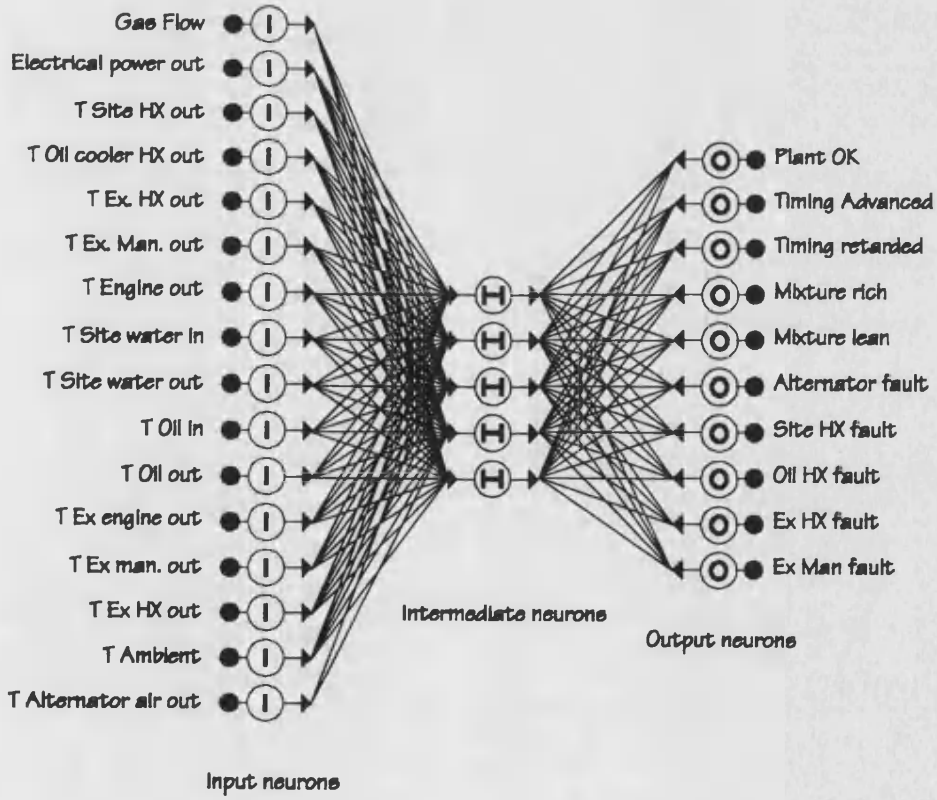


Figure 7.2 The fault recognition neural network model 3 and 4

7.2.1.4. Model 4 - Diagnostic

This model has the same structure as the previous model. The input data is however modified. The input values used are the deviations from a known healthy state at one ambient condition. Because the neurons work on an activation level between 0 and 1 the deviations are normalised using maximum and minimum anticipated deviations. This means that zero deviation from the healthy state is represented by an activation of 0.5. The healthy state of the system is therefore when the activation of all the inputs equals 0.5. An activation of 1 shows that the parameter is very much above normal and a value of 0 is very much below.

This approach should give much more obvious patterns of change due to the introduction of the various faults. It also makes it easier for the operator to understand what inputs are influencing the network as he can inspect the input data and look for values that are not close to 0.5.

7.2.1.5. Model 5 - Diagnostic

This model is a modification of the previous example. Each of the inputs are split into two and so there are 32 inputs to this network. Each performance parameter has a pair of inputs associated with it. One input represents that the value is higher than normal and the other shows that the value is lower.

The performance data is pre-processed in a similar way to the previous example. The input deviations from the expected performance are first calculated. These values are filtered so that, if the deviation value is over 5% above the normal value, the high input neuron for that parameter is set to 1. If the value is less than 5% below the expected value the low input neuron is set to 1. The healthy case is therefore represented by zero values at all inputs.

This approach gives much stronger input patterns than any of the previous models as it is basically a binary system, all input values are either 0 or 1. The level of filtering is clearly critical because if it is set too high then a lot of subtle deviation information is lost, too low and the system becomes susceptible to noise.

This pre-processing is similar to that necessary for most rule based expert systems. The performance of this network should therefore compare directly with conventional techniques.

This model is shown in figure 7.3. In this case it was found that an extra intermediate neuron was needed.

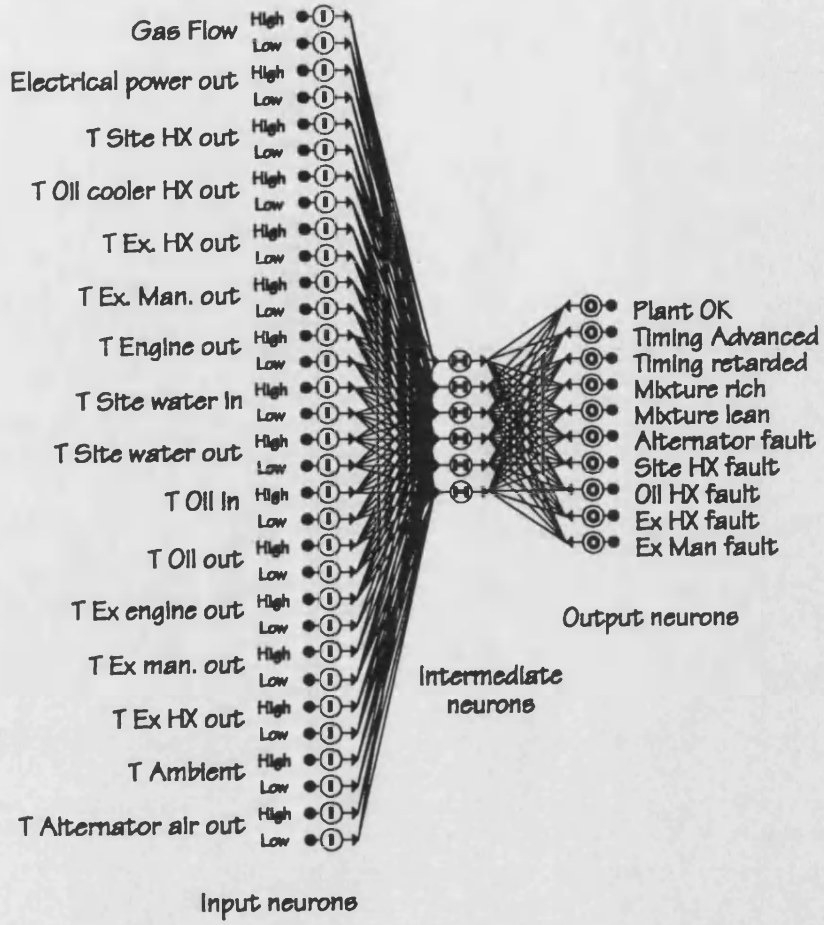


Figure 7.3 The fault recognition neural network model 5

7.2.2. A comparison of the training of the models

The training of each network was logged and the results are shown in figure 7.4. The figure shows the maximum training error after each training epoch as described in Chapter 3.

The different training requirements of each model can clearly be seen. The figure does not relate directly to time because extra incidental factors also effect this. These include the complexity of the network and the number of other processes running concurrently with the neural network package. The times indicated on the figure are therefore very approximate. They are the times achieved on a Viglen 486 33 MHz machine with 8 Mbytes of RAM.

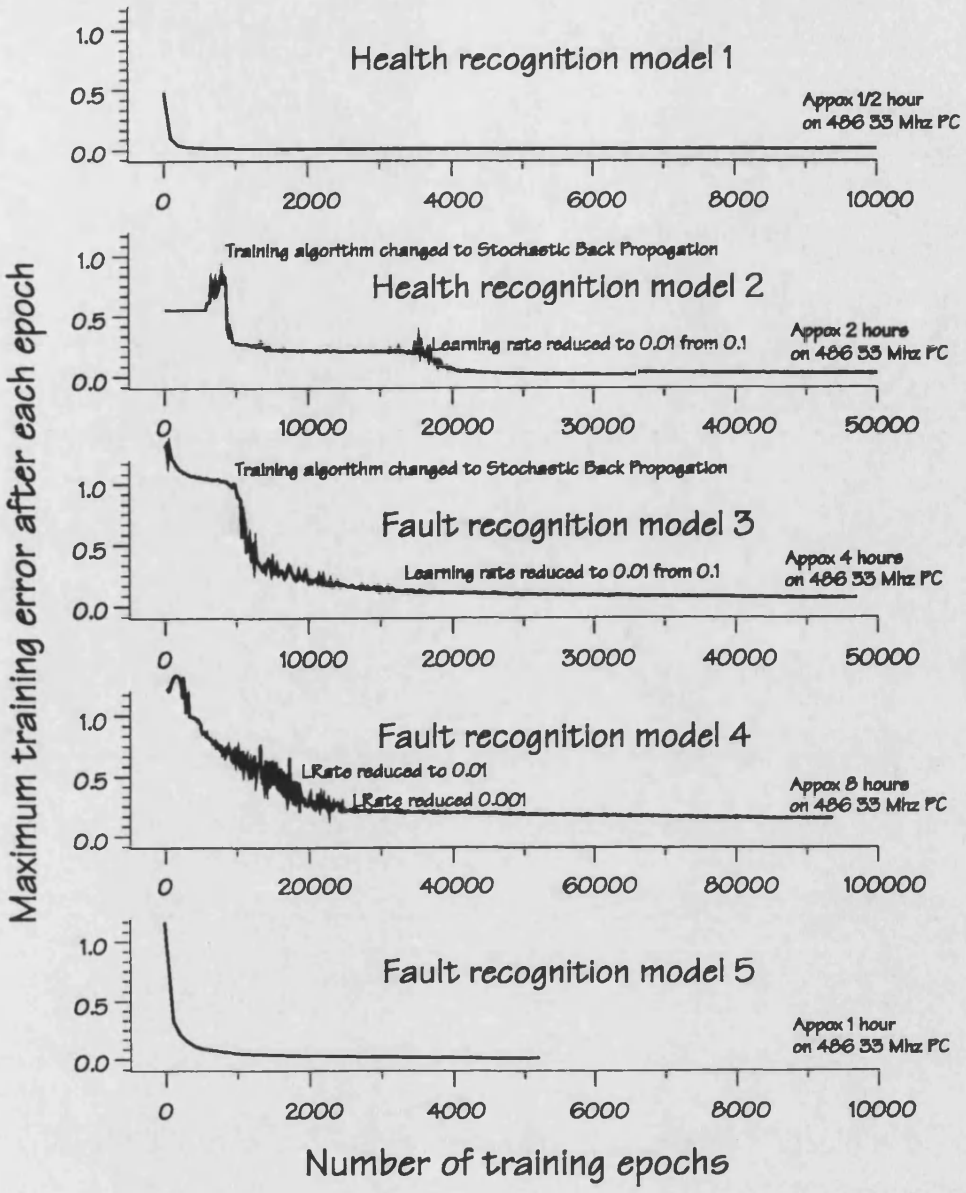


Figure 7.4 **The training error logs for the basic diagnostic networks.**

What can be seen is that the simpler binary type networks of models 1 and 5 train very rapidly, converging to the desired accuracy of less than 1% maximum training error in less than 1000 iterations (epochs). The more sensitive networks, however, take much longer and are prone to some interesting irregularities, as can be seen in the figure.

Model 2 was trained initially using the ***standard back propogation*** method, but there was clearly no reduction in training error even after about 4000 epochs. At this point the training method was changed to the ***stochastic back propogation*** method. This method presents the vectors of the training data set in a random order and calculates the new *weights* and *biases* after each

individual vector is presented rather than calculating after all the vectors of the set have been presented. This method is therefore slower but, in this case, it manages to break the training process out of the deadlock. In this example, after the new method is adopted the training becomes unstable for a short period and then stabilises at a much lower training error, gradually reducing for a further 15,000 epochs. It then suddenly becomes very unstable again. At this point the learning rate was reduced from 0.1 to 0.01 so that the *weights* and *biases* would not be changed too much each iteration. This allowed the network to stabilise again and train steadily until a satisfactory error level was reached after a further 30,000 epochs.

These results show that interactive training is necessary when the networks are trying to learn complex relationships. The *NCS* package was very useful for this because it gives a good clear display of how the training is progressing and allows control of the model while it is working. As the next two models show, a certain amount of interference was necessary to achieve the desired training accuracy.

In several cases in the course of this research it was impossible to break out of a training deadlock, but if the network was re-randomised it managed to train without further problems.

7.2.3. Storage requirements of the basic models

The data defining the network is stored on a file that take up very little space, less than 6 kBytes of storage for all the example networks described here, as can be seen in the table 7.2 below.

Unlike conventional knowledge based systems, this size does not need to be increased if extra information needs to be *learnt*. A conventional rule based system needs more space to store new rules, a neural network only needs retraining with the new information.

Model name	File name	Size (Bytes)
Healthy recognition model 1	HLTH1REC.NCS	1962
Healthy recognition model 2	HLTH2REC.NCS	1962
Fault recognition model 3	FAULTREC.NCS	2934
Fault recognition model 4	DEV1REC.NCS	2934
Fault recognition model 5	DEV2REC.NCS	5436

Table 7.2 Storage requirements the diagnostic models

7.2.4. Assessment of the performance of the basic diagnostic models

To compare each model a set of standard tests were developed and applied to all networks. The tests were designed to investigate the networks' ability to learn the performance data and deviations due to faults, and also to test the sensitivity to problems that a real system might have, such as changes in ambient conditions, noisy data or the occurance of unknown and multiple faults. The tests were as follows:

i) Interrogation with simulated data representing different normal and ambient conditions.

ii) Interrogation with simulated data that varied from the normal data set by a random amount +/- 1%. This is to represent sensitivity to measurement errors in the test cell.

iii) Interrogation with a set of the 9 simulated faults listed above.

iv) Interrogation with data representing a progressive fault in the exhaust heat exchanger

v) Interrogation with simulated data representing a dual fault, one in the alternator and one in the exhaust heat exchanger

vi) Interrogation with real data from the healthy CHP plant and from the plant with a defective exhaust heat exchanger. (effectiveness reduced to 77% from 89%)

Each test was constructed in an *excel* workbook. The test data were passed to the *neurun* module via the *windows* DDE protocol using *excel* macros, as described in Chapter 5. The results of the interrgoation were returned to the *excel* workbook where they were presented in the form of a bar graph of the activation level for each output. The entire interrogation process takes well under a second to perform for all the models examined. This *interrogation time* contrasts sharply with the *training times* of up to 8 hours described previously.

The results of each of the tests are described below:

7.2.4.1. Interrogation with simulated data representing different normal and ambient conditions.

This test is designed to investigate whether the networks can cope with changes in ambient conditions. The outputs of the health recognition models

should equal 1 in all cases. Only the *plant OK* output should be active for the diagnostic models.

Figures 7.5 shows the response of the two healthy recognition models to different ambient conditions. Clearly both of these models indicate that the plant is healthy.

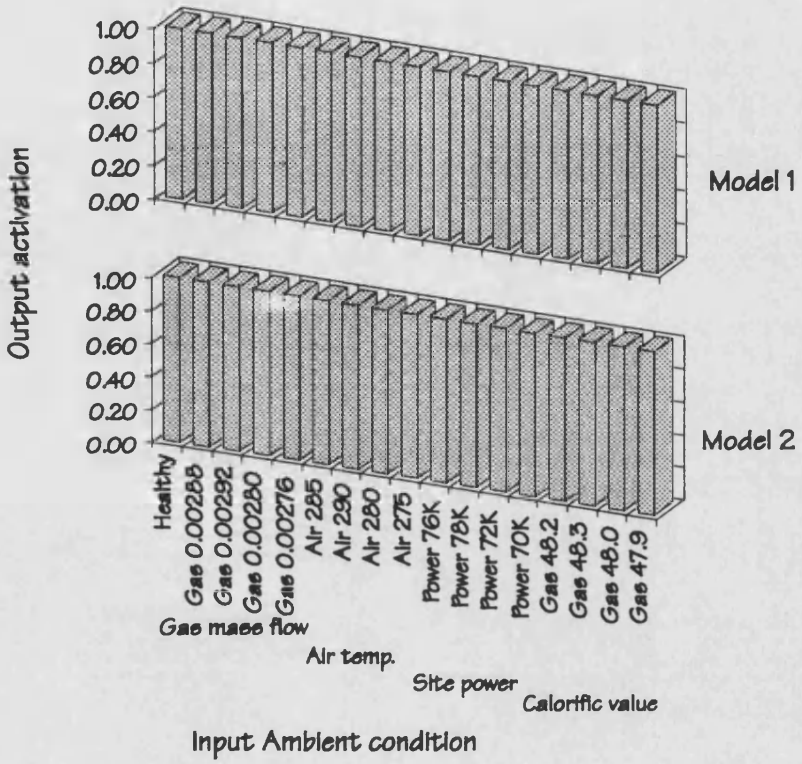


Figure 7.5 A comparison of the output of the healthy recognition models to data representing differing ambient conditions

The response from the fault recognition plants in figure 7.6 clearly shows that all three diagnostic models indicate that the plant is healthy under all ambient conditions.

There is a small *blip* on the model 5 output for the case when low power is being absorbed into the site water. The model still shows that the plant is healthy but there is some doubt. What is interesting about this feature is that this is the model that has binary inputs. As can be seen in this example the output is not restricted to binary levels, if the input pattern does not match one of the training patterns exactly it gives an output that reflects the most likely contributers to the deviation. This ability to give proportional outputs clearly makes this approach more flexible than conventional pattern recognition techniques analysis.

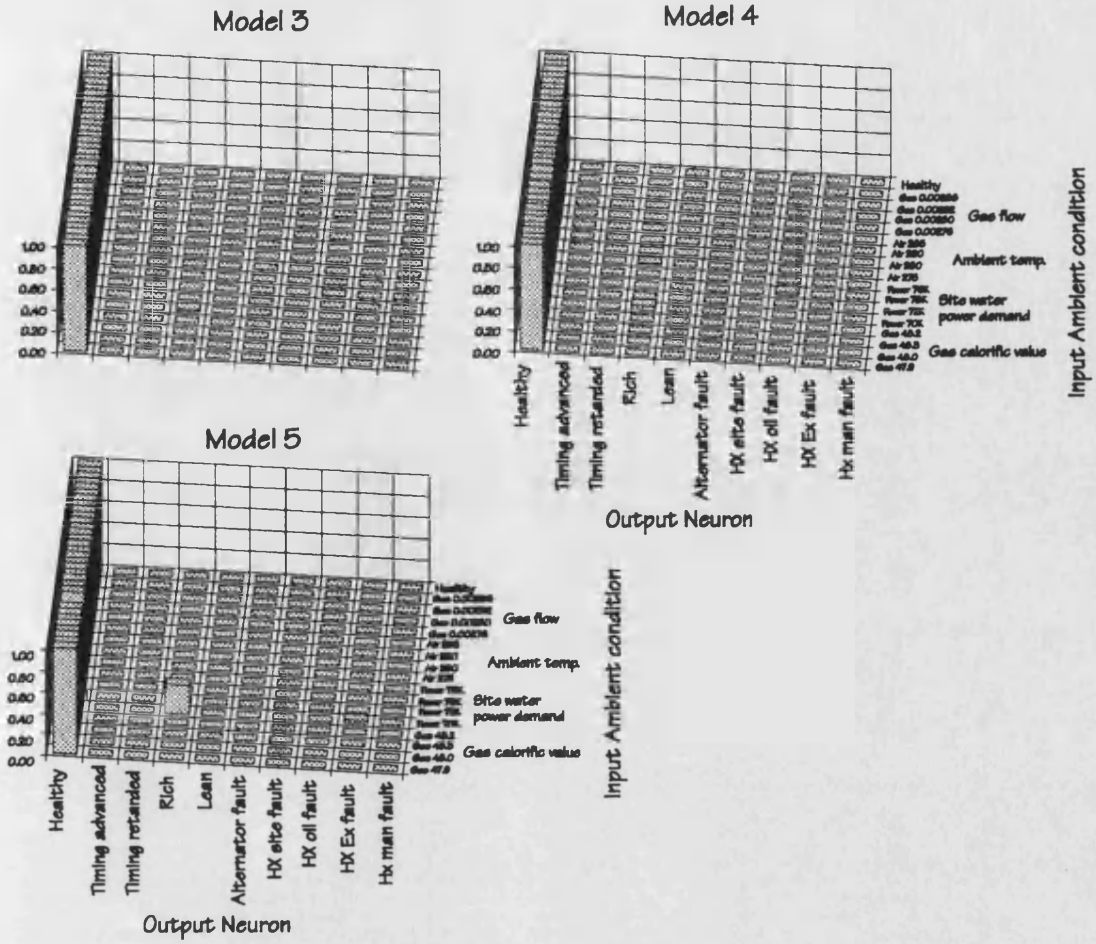


Figure 7.6 **A comparison of the output of the fault recognition models to data representing differing ambient conditions**

7.2.4.2. Interrogation with simulated data that varied from the normal data set by a random amount up to +/- 1%.

This test is to assess the diagnostic model's sensitivity to typical measurement errors. The measurement error of modern instrumentation quality transducers and signal conditioning is about 1% FSD and so this was the level set for this test. 10 different sets of random data were used to ensure that the networks were exposed to a reasonably diverse set of examples. This test should give the same healthy outputs as the previous one if the networks are insensitive to these errors.

The output of the healthy recognition networks shown in figure 7.7 and clearly both of these models indicate that the plant is healthy.

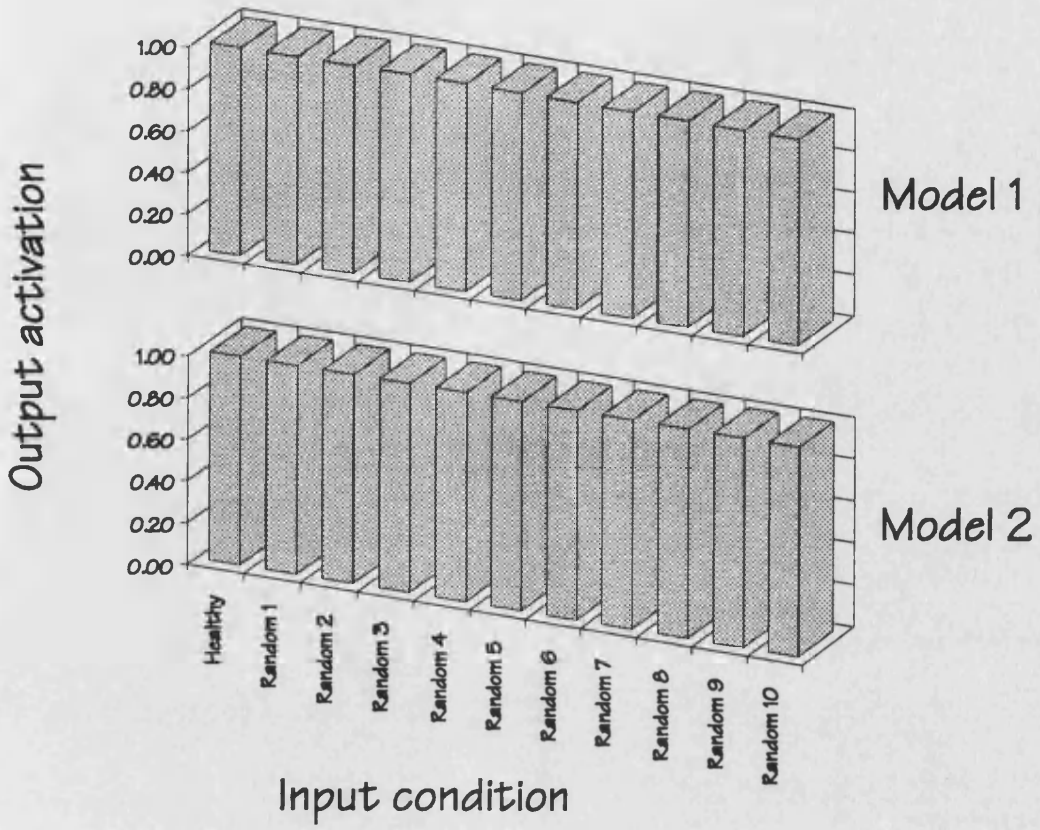


Figure 7.7 **A comparison of the sensitivity of the healthy recognition models to random deviations of up to +/- 1%**

Figure 7.8 shows that the fault recognition models are also insensitive to the random deviations. They all clearly indicate that the plant is healthy.

It should be noted that the 1% level is also the level set for the maximum training error for the networks and is below the filtering threshold value for model 5. When the networks were allowed to train further, or if the threshold is reduced, then they do become sensitive to these random fluctuations. Over training makes the networks very sensitive to the actual patterns of the orignal data set and spoils their ability to generalise.

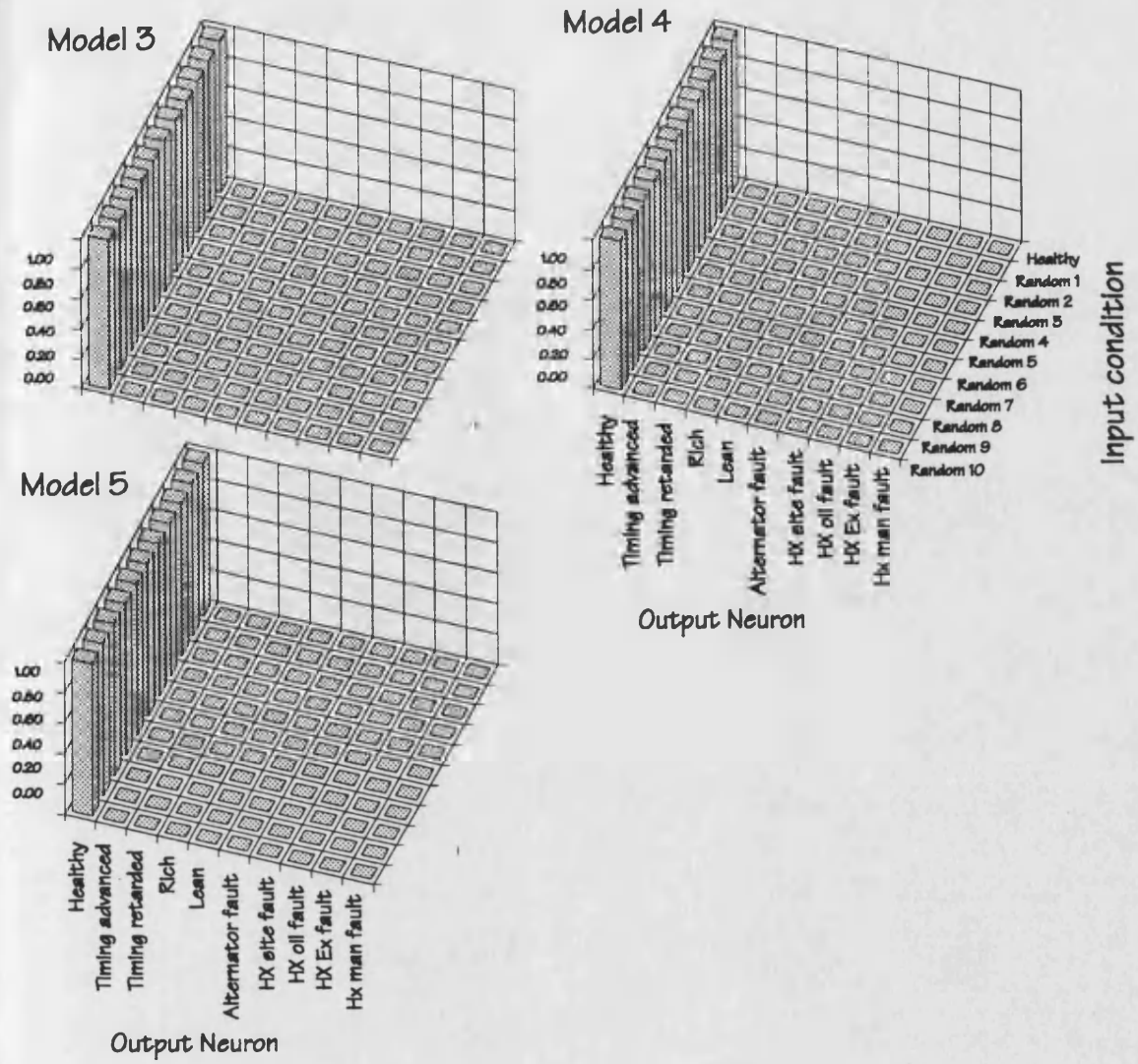


Figure 7.8 **A comparison of the sensitivity of the fault recognition models to random deviations of up to +/- 1%**

7.2.4.3. Interrogation with a set of the 9 simulated faults listed above.

This test is designed to see if the networks can pick up the presence of a fault and, in the case of the diagnostic models, identify it. The output of the health recognition models should therefore be 0. For the diagnostic models, all outputs should be 0 apart from the output representing the fault that is present.

Figure 7.9 shows the response of the health recognition models. Model 1 does not respond to the faults at all, wrongly indicating that the plant is still healthy. Model 1 has had no exposure to fault information in its training and obviously the data looks enough like the healthy plant data that it was

trained on for it to respond in this way. Model 2 performs better indicating the 8 faults that it had previously experienced. The last fault that was excluded from the training data set is, however, misrepresented as healthy.

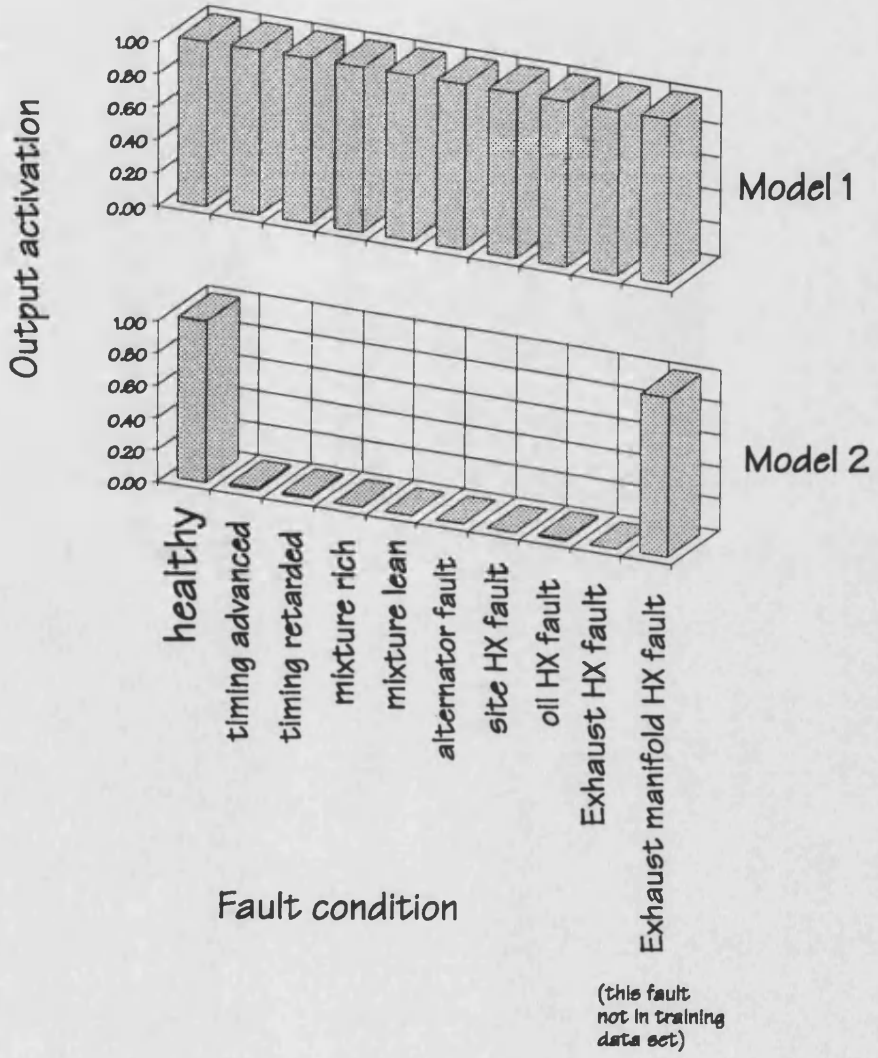


Figure 7.9 **The response of the healthy recognition models to 9 simulated faults**

All three fault recognition models perform very well on this test clearly indicating the presence of each fault, as can be seen in figure 7.10.

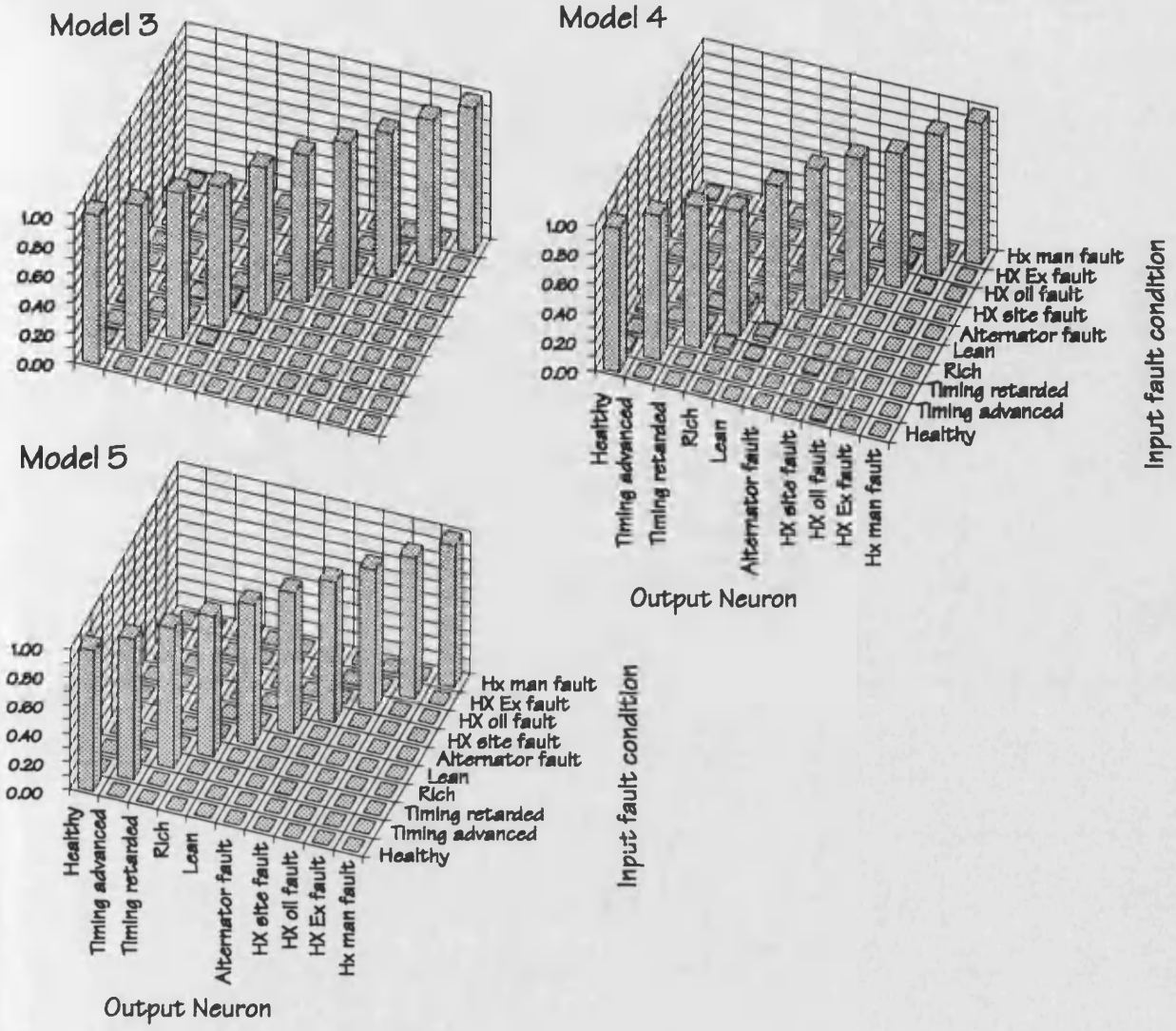


Figure 7.10 **The response of the fault recognition models to 9 simulated faults**

7.2.4.4. Interrogation with data representing a progressive fault in the exhaust heat exchanger

This test is designed to investigate at what levels the network begins to respond to a new fault. In the training data a faulted exhaust heat exchanger was defined as one with an effectiveness of 50% as opposed to 89% for the healthy version. The CHP was therefore simulated with varying levels of heat exchanger effectiveness and this data passed through each network. The output should therefore indicate reducing levels of health and highlight the exhaust heat exchanger as the cause.

Figure 7.11 shows again that the first health recognition model is insensitive to all levels of this fault. The second model responds well showing a clear reduction in confidence of the state of health of the plant, even with only a 9% reduction in heat exchanger effectiveness.

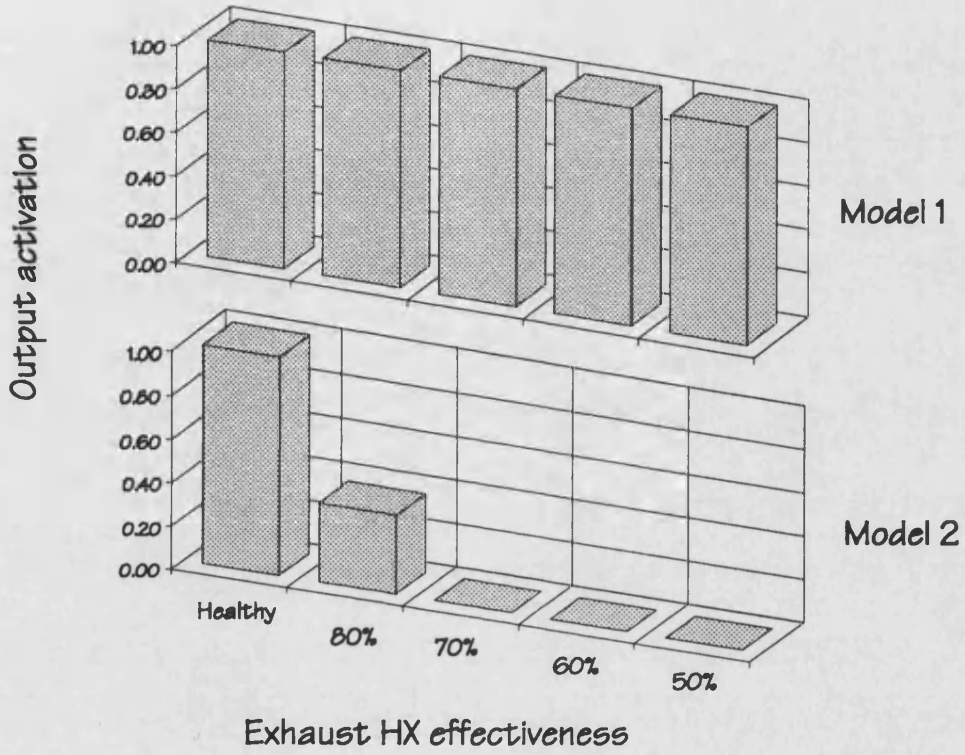


Figure 7.11 The response of the healthy recognition models to a progressive introduction of a single fault

There is very clear responses from the fault recognition models as can be seen in figure 7.12.

Diagnostic Model 3 is clearly very sensitive and shows that the plant is not healthy with only a 9% reduction in heat exchanger effectiveness present. At this stage, however it can not identify the cause of the fault. By the time the effectiveness has reached 70% it is clearly indicating that the exhaust heat exhanger is the cause, this output having an activation of about 50%. Once the effectiveness has been reduced to 60% the output activation for this fault has increased to over 90%.

Model 4 is just as sensitive to the presense of the fault but is not so certain about the cause, indicating that is is either the exhaust heat exchanger or the oil cooler until the 60% effectiveness is reached.

Model 5 takes much longer to respond and also gives two potential causes until the final 50% effectiveness is reached. This insensitivity is due to the filtering of the data that is occuring before the network sees it. Important information about small deviations is therefore being lost because of this filtering. The network does respond well to the patterns when the information starts getting through. Clearly the level of filtering is very important for this approach to be as effective as the ones that are connected directly to the transducers.

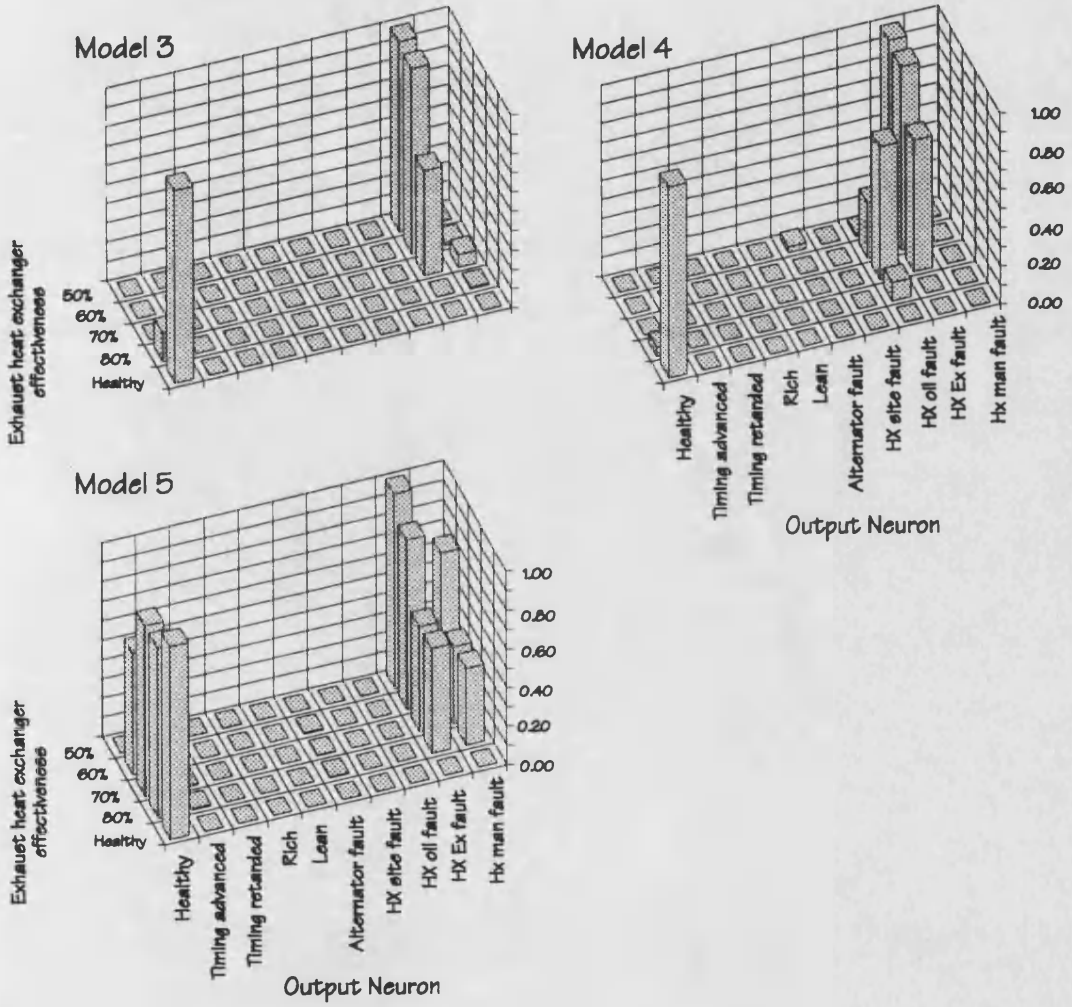


Figure 7.12 **The response of the healthy recognition models to a progressive introduction of a single fault**

7.2.4.5. Interrogation with simulated data representing a dual fault, one in the alternator and one in the exhaust heat exchanger

Many diagnostic systems become confused by the effects of several faults present at the same time. This test is designed to investigate how the networks respond to performance data from the CHP plant when two faults are present, even though they were trained on each fault individually.

All the models, except model 1, generally give a good response in this test despite being trained on the faults independently as can be seen in figure 7.13. The healthy recognition model 2 clearly indicates the presence of a fault as does the fault recognition model 3. Fault recognition models 4 and 5 show the presence of a combined fault but model 5 also suggests a timing fault is also present.

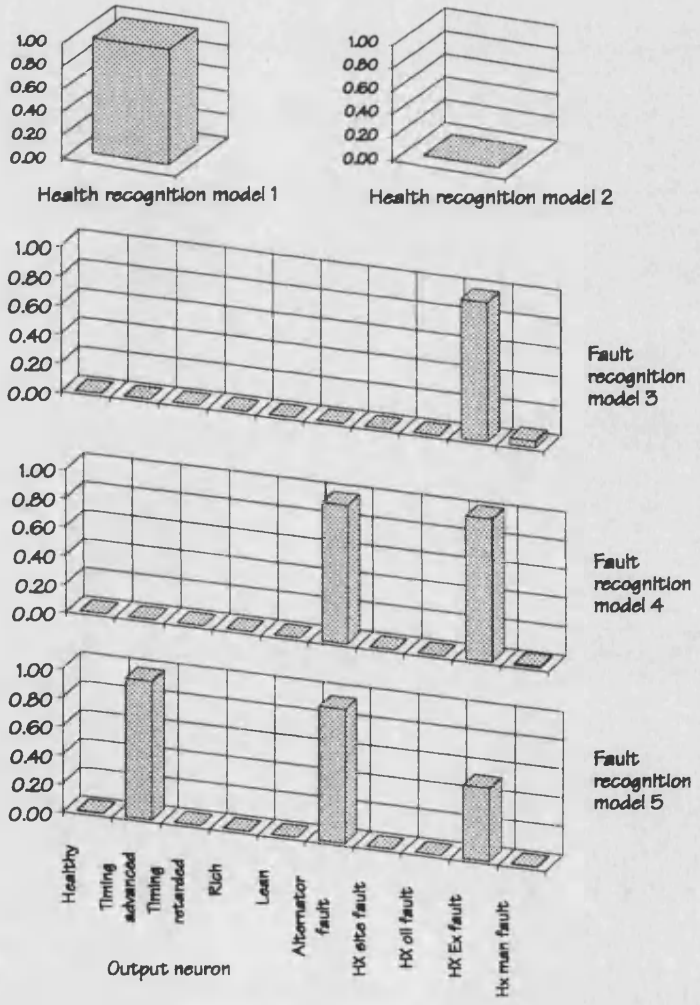


Figure 7.13 The response of the models to a combined alternator and heat exhaust exchanger fault

7.2.4.6. Interrogation with real data from the CHP plant.

For this test data from the British Gas experimental plant was used. Data was available for both a healthy system and the system with a known fault in the exhaust heat exchanger. It is known from performance analysis of the data that the fault reduced the effectiveness from 89% to 77%.

Again all but the first health recognition model performed well with the real data as can be seen in figure 7.14. They all show the data from the real healthy plant to be OK and show the presence of a fault with the data from the faulted plant. The fault recognition model 4 correctly diagnoses the fault, although it does suggest a combined fault. Models 3 and 5 show that the installation is faulty but wrongly diagnose the cause.

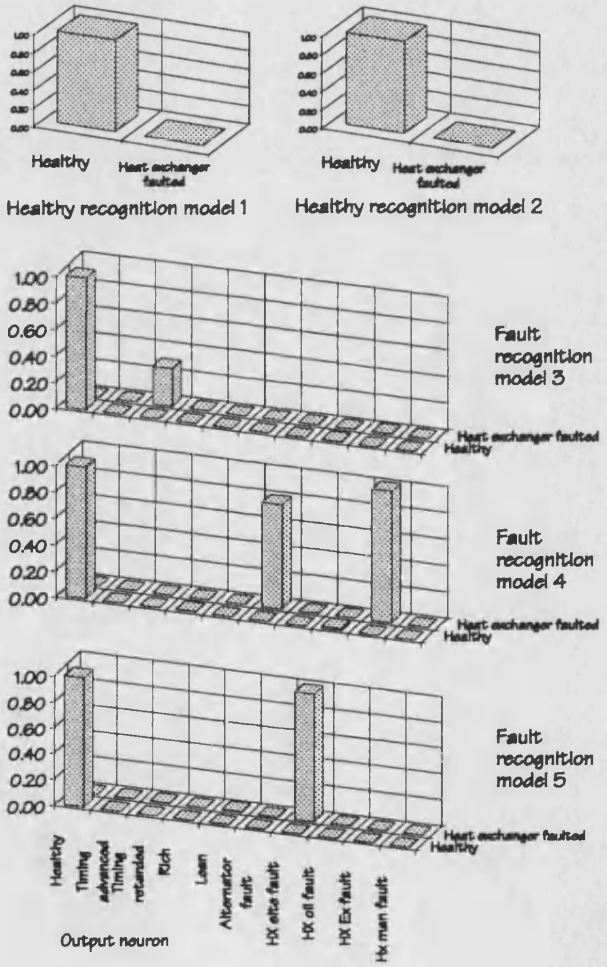


Figure 7.14 **The response of the models to real CHP data from both healthy plant and plant with a known exhaust heat exchanger fault.**

7.3. Enhancements to the basic models

The previous results show promising results despite their simplistic methods.

The examples show there would be benefits from knowing what performance is expected at a certain operating point under the specific ambient conditions. The effects of ambient conditions could then be removed leaving clearer patterns for the networks to work on. A separate network could be used to predict the performance expected under particular operating conditions. The measured performance could then be subtracted from the output to generate deviations for a diagnostic model.

The third fault recognition model is able to filter out misleading deviations that could be caused by small ambient fluctuations or measurement error. This model, however, requires the banding of the deviation into different streams according to the amount of deviation. The second enhancement looked at in this section shows how neural networks can be used to stream the deviation into 2 or more channels.

How the different units described here could be combined to make a complete system is described in the discussion of the results later on.

7.3.1. The performance predictor model

This model has 4 inputs representing the different parameters that determine the system's operating point. It has 16 outputs that represent the 16 measuring points on the plant. The network is illustrated in figure 7.15.

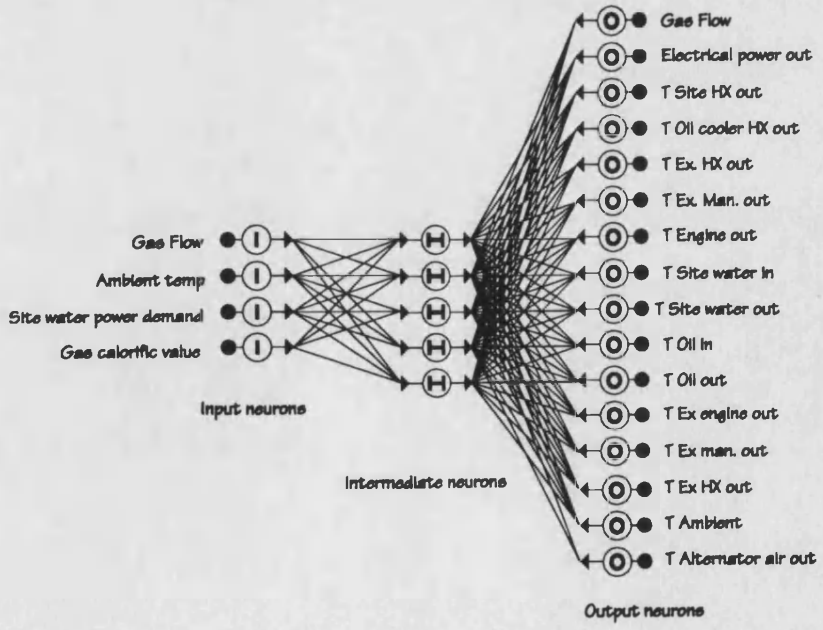


Figure 7.15 The performance predictor neural network.

The network was trained on an operating map generated for the tests above. The trained network was able to reproduce the original data very well as can be seen in figure 7.16 showing the absolute error of each output at each training point.

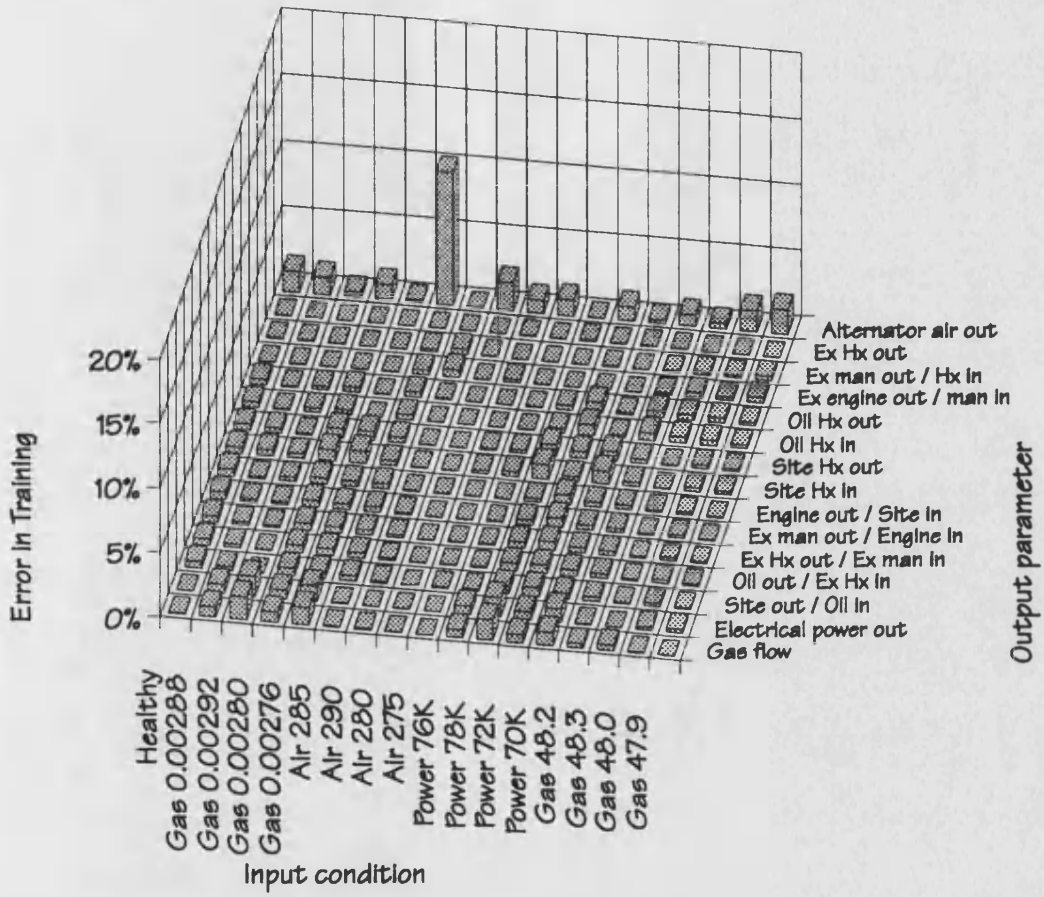


Figure 7.16 The error in response of the performance predictor neural network.

7.3.2. The deviation filter

Two neural filters are shown in this section. The networks can be seen in figure 7.17 and the output in figure 7.18.

The first streams the input value into 2 bands representing *low* and *high* values. If the input value is less than -0.05 then the *high output* = 0 and the *low output* = 1, indicating that the input is low. If the input is between -0.05 and + 0.05 then both outputs are = 0, indicating that the input is neither high nor low. If the input is above +0.05 then the *high output* = 1 and the *low output* = 0, indicating that the input is high.

Clearly this simple network could provide very effective streaming of data comparing expected and actual performance suitable for generating inputs for a banded deviation diagnostic network such as the demonstration model 5 described previously, or for activating warning lights.

Model 5 proved to be insensitive to small changes because of the loss of information with coarse filtering. The other filter shows the potential for streaming the data into more than two bands to give the network more information to base its diagnosis on. In this case the outputs show that the input is *very low, low, high* or *very high*. This could be used for a more comprehensive banded deviation diagnostic network.

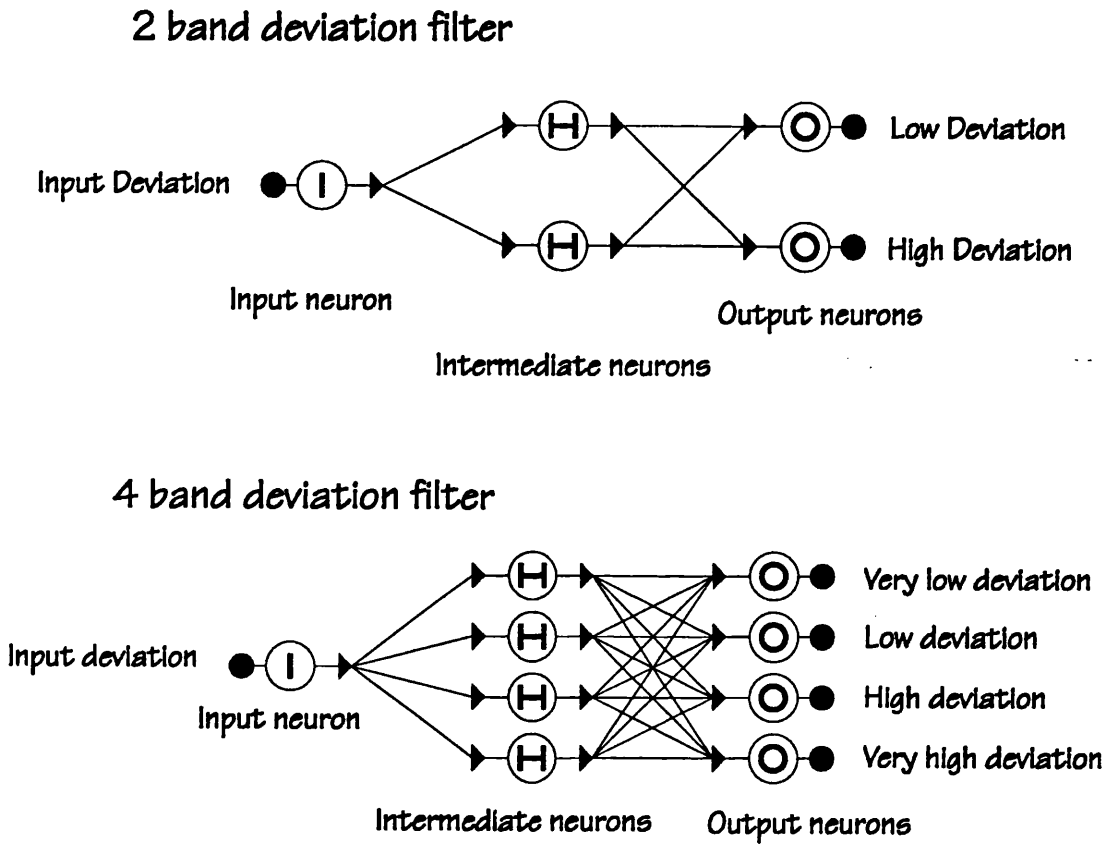


Figure 7.17 The neural network deviation filters.

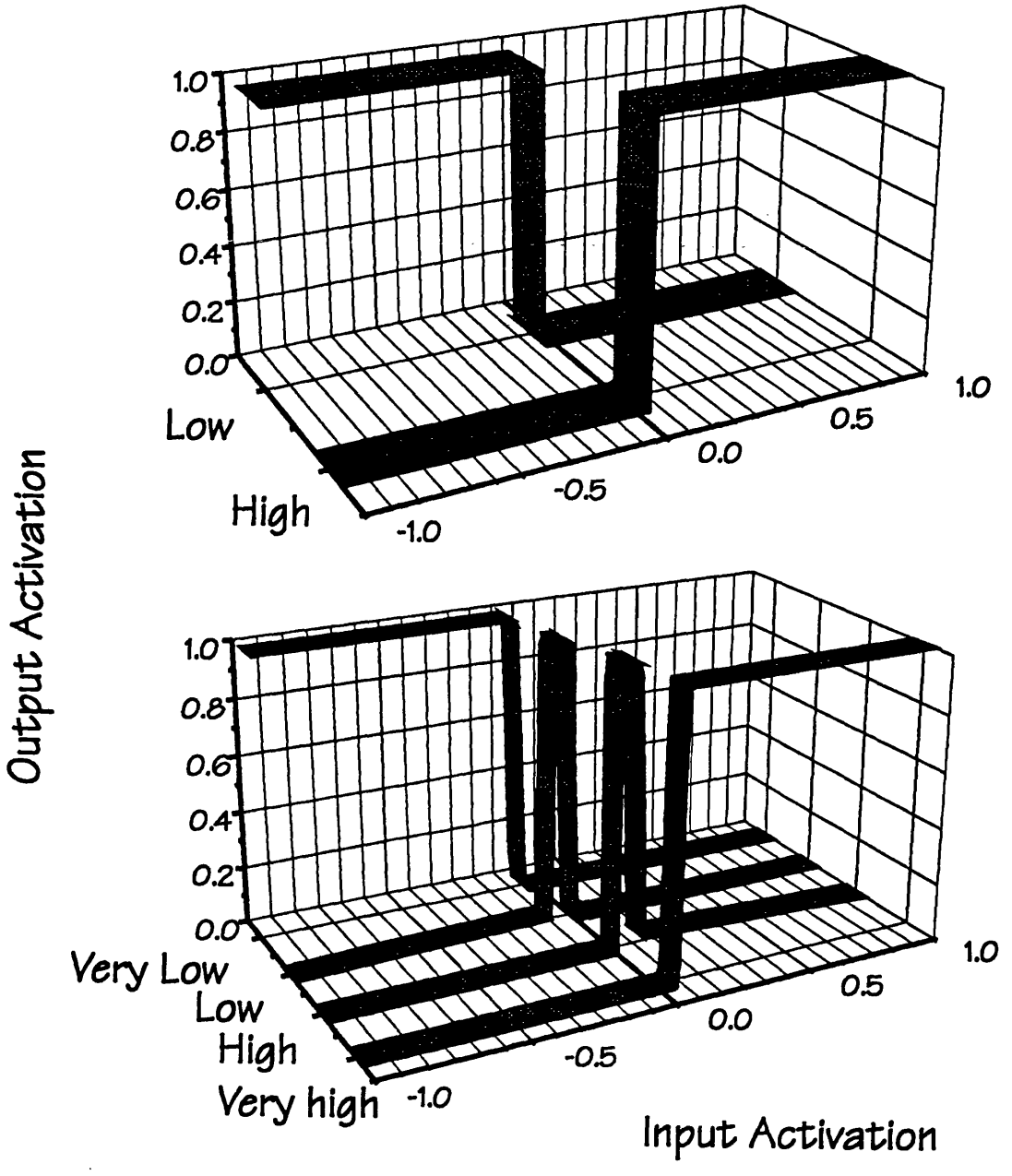


Figure 7.18 **The response of the neural network deviation filters.**

Chapter 8

An on-line neural network diagnostic system (Diesel engine)

8.1. Introduction

The previous chapter shows alternative approaches to constructing a neural network diagnostic system. It demonstrates how a simulation can be used to generate the data necessary for training the network. The example uses a very simple simulator but does produce reasonable results with the real data available. This chapter describes a more comprehensive system that was built to run on-line with the TL-11 diesel test engine. For this the complex simulation described earlier was used to generate entire performance maps of both the healthy and faulted engine. This was used to train a neural network off-line that was then connected to the real engine via the software described earlier. This system was able to produces a live diagnosis of the engine as it ran.

8.2. TL 11 Diesel Engine diagnostic model

The working model shown in figure 8.1 has been developed to demonstrate how a neural network may be used for condition monitoring.

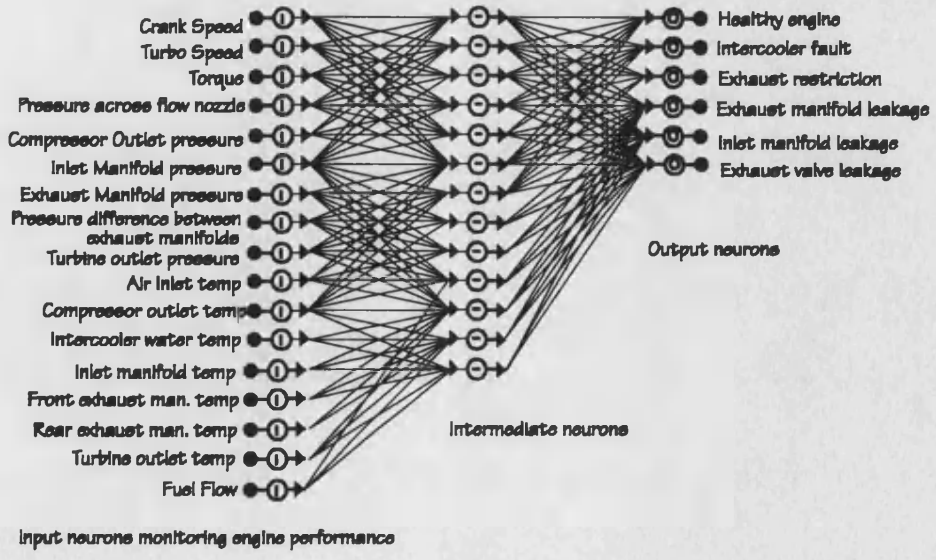


Figure 8.1 The neural network for the diagnostic model

The network is comprised of 17 inputs that represent key performance parameters and 6 outputs that represent the engine's state of health; A high value of the first output value indicates that the engine is healthy, the second value indicates that fault 1 is present, etc. One intermediate layer of 13 neurons proved adequate for this example.

The network is first trained on simulated data for the healthy and faulted engine for all speeds between 1000 and 2200 rpm (at 200 rpm intervals) and for all bmep's between 2 and 10 bar (at 2 bar intervals). An example of the training data is given in figure 8.2. This shows the monitored performance parameters for the healthy engine compared with each of the different fault conditions at one speed and load point. The values have been normalised with reference to their maximum expected value for ease of viewing.

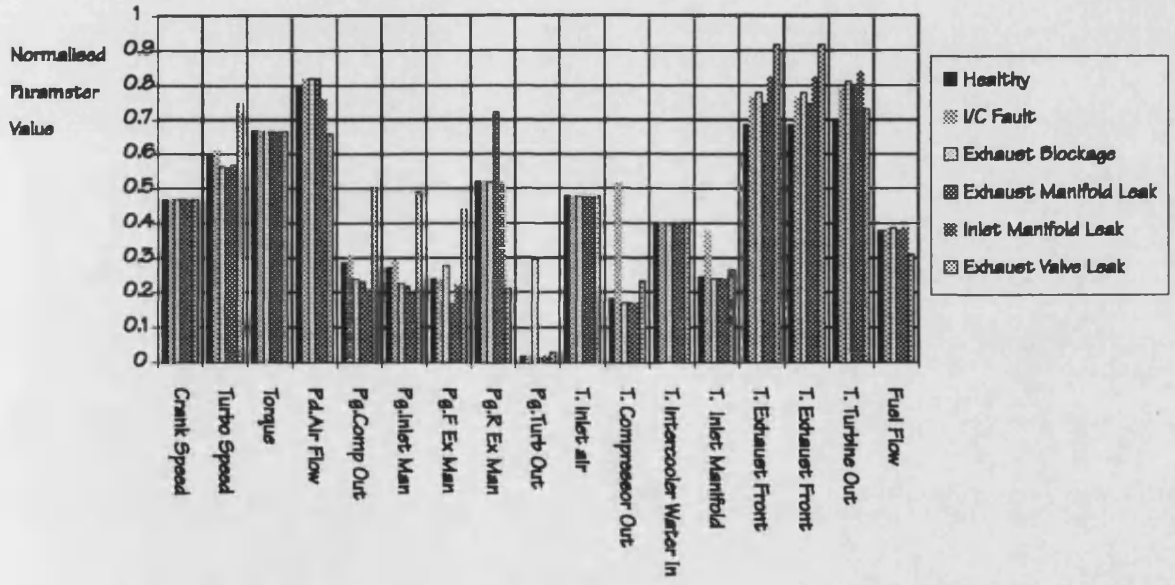


Figure 8.2 An example of the training data set at one load and speed point

It can be seen that this data is all from one operating point as the load and speed values are constant height for all fault states. This figure shows how complex the pattern of change is for the various faults. Parameters such as turbine speed decreases with some faults yet increases with others. It is these patterns for all operating points that the network is trained to recognise.

As discussed in Chapter 3, the training process is an iterative one, whereby the entire data set is repeatedly passed through the network. The errors in the output are used at each iteration to modify the neural functions. The software

used for the neural processing uses the standard back propagation method. The iteration is allowed to continue until the average error has reached an acceptable level. The network described above is able to achieve an average 96% convergence over all the data sets given. It can therefore be said that the network is able to learn the relationship between the engine performance and its state of health, with regard to the faults described above.

To illustrate the type of output that can be gained from the network, Figure 8.3 shows how the networks outputs respond to the introduction of a simulated leak of various sizes in the inlet manifold, at one load and speed point.

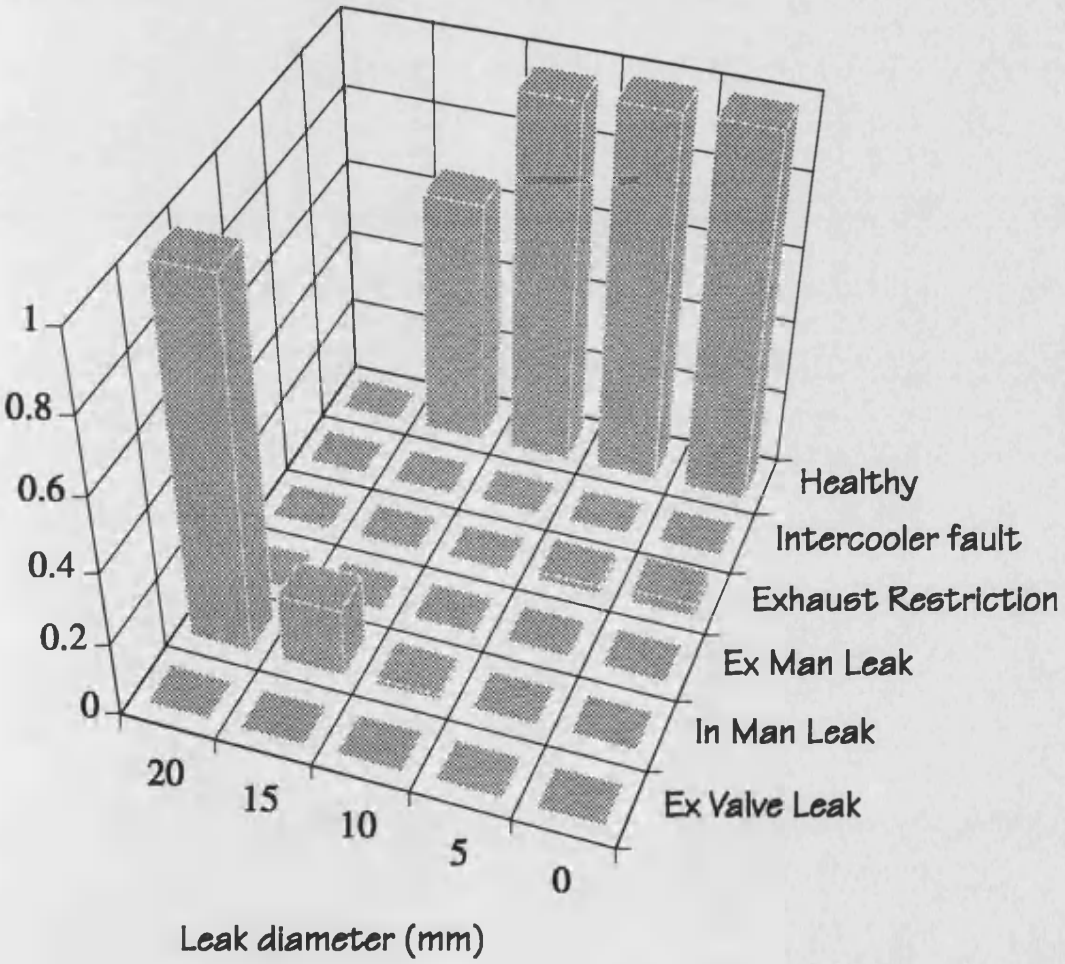


Figure 8.3 **Neural response to increasing manifold leakage at 2000 r/min, 8 bar bmep**

From this figure it can be seen that the output indicating health has the highest activation until the hole reaches 15mm diameter At this point this activation is reduced to about 0.7 indicating that a fault may be present. Of all

the other outputs only the *inlet manifold leak* output has increased, but only to about 0.2 suggesting that this may be the cause. Once the hole has reached 20mm the network is confident that an inlet manifold leak must be present.

Figure 8.4 shows in more detail how the two key outputs respond to the introduction of the leak. The figure shows maps of the response of the healthy output and the inlet manifold leak output at all speeds at 8 bar bmep.

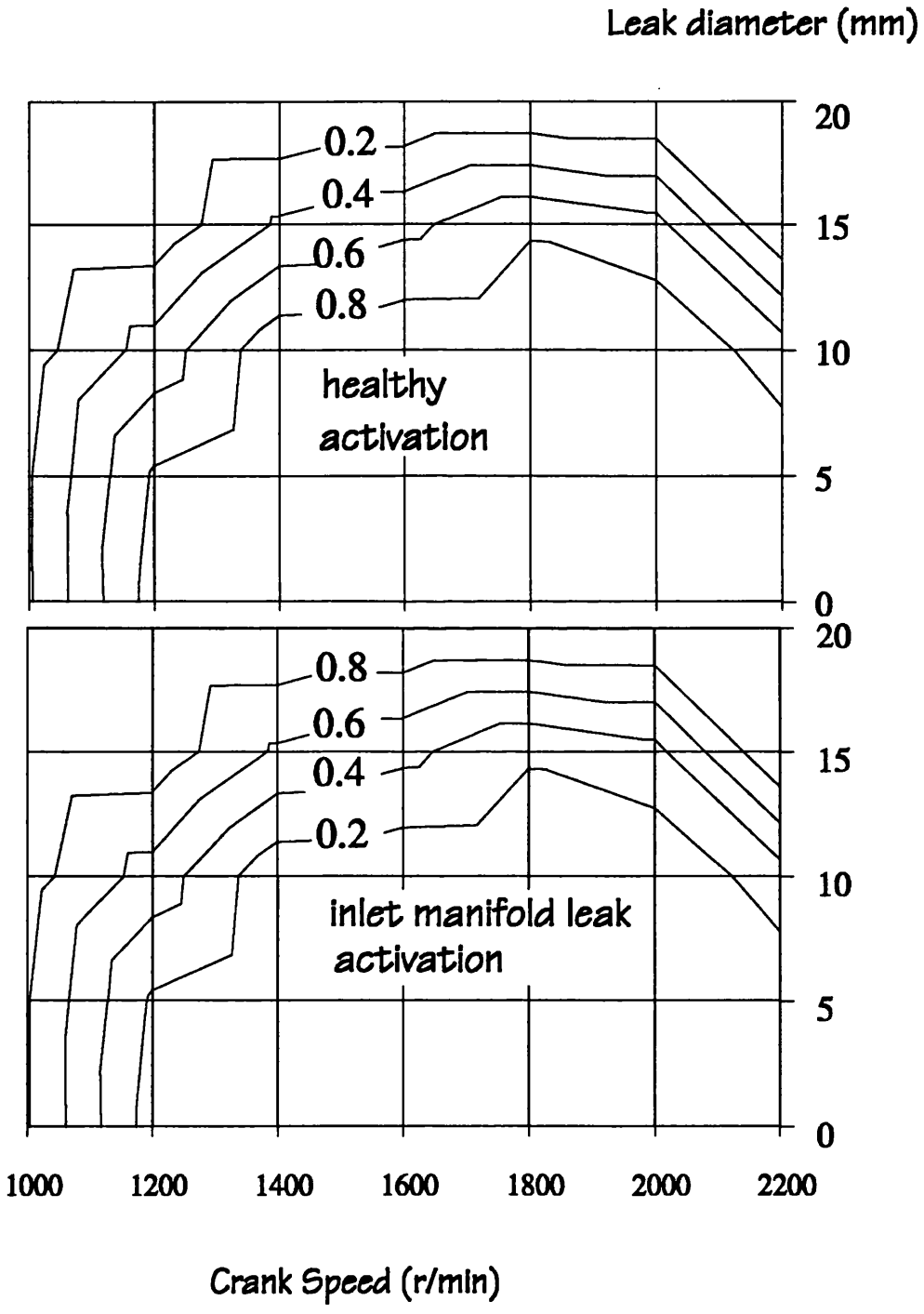


Figure 8.4 **Response of the healthy and inlet manifold leak outputs to the introduction of a hole into the inlet manifold**

The above examples used simulated data. Figure 8.5 shows the response of the network to real engine data whilst running at 1600 rpm 800 Nm. The figure shows the results for a healthy engine plus its response after the introduction of several faults. Typically an interrogation by the network takes less than a second to complete.

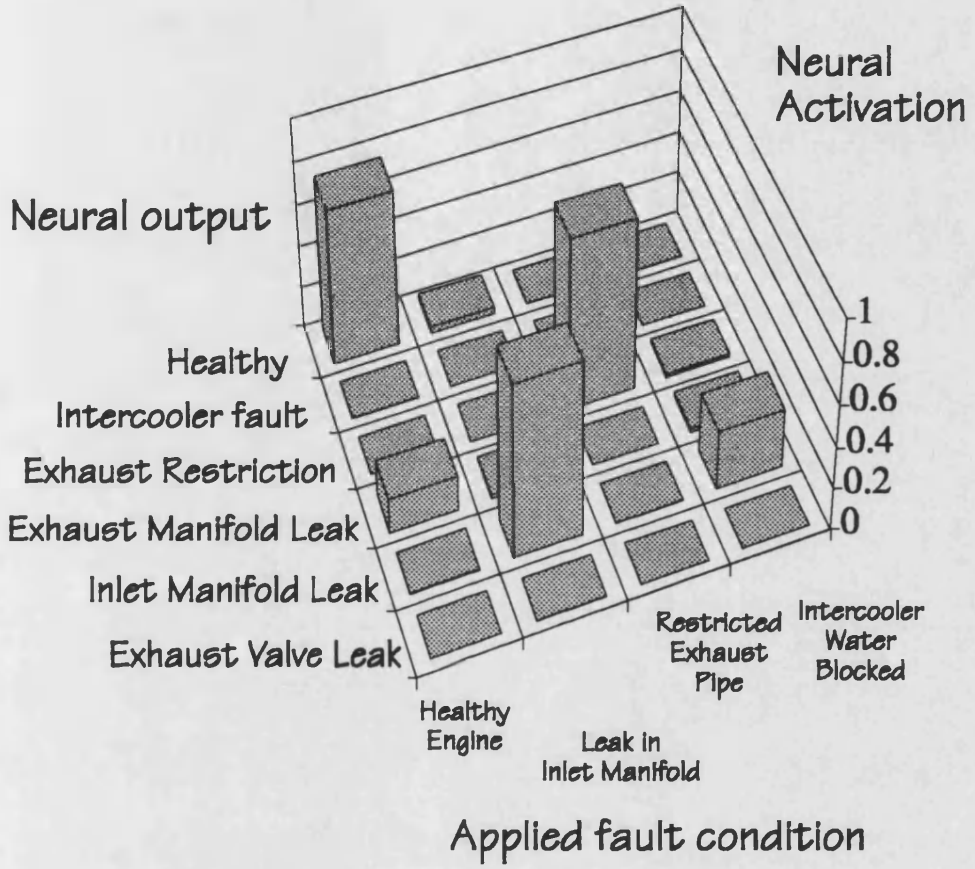


Figure 8.5 **Response of the neural network to real engine data with the introduction of various faults**

The first case on the chart shows that the network is 70% certain that the engine is healthy. The activation of all the other outputs indicating faults are minimal. If the engine has faults introduced, the network responds accordingly. The network does fail to distinguish the restriction of the water supply to the intercooler as an intercooler fault. It shows that a fault is present, indicating this by giving very low activation's on all output channels including healthy. The inlet manifold fault has the largest activation of 28%. This may mean that the method of simulation for the intercooler fault does not tie in with the response from the real engine and needs further investigation.

The example given concentrates on one operating point, but the network does perform equally well on all areas of the map. It even managed, with limited success, to respond to operating points that lie outside the limits of the original data sets. This problem would be very difficult to resolve for conventional pattern recognition techniques.

Chapter 9

Recommendations for future research

9.1. Introduction

The example networks in the previous chapters show that this technique clearly has potential in the field of diagnostics. The examples do, however, demonstrate that many areas need further research before a commercial system could be employed. This section highlights the areas of research that the author considers to be of primary concern. Various considerations for verifying such a system are also considered.

9.2. Specific areas requiring further research

The areas of study can be classified into the following areas.

- The effect of different training algorithms.
- The effect of training convergence.
- Neural network architectures.
- Input data requirements.
- The development of a detailed simulator.
- The potential for analogue solutions

9.2.1. The effect of different training algorithms.

For this research only the *standard back propagation* and the *stochastic back propagation* were used. In some cases the network failed to converge whilst training and would not train further. This problem can sometimes be overcome by simply resetting the network parameters to random values and starting again or by selecting a different algorithm. A range of available algorithms should be assessed for a prospective application.

9.2.2. The effect of training convergence.

During this research it was found that the best results for specific examples were achieved by letting the network train to very low average errors. This

was found, however to compromise the models generality. The ideal extent to which networks should be trained needs to be assessed. The addition of random noise to the training data to avoid over training is a technique that should be investigated in this context.

9.2.3. Neural network architectures.

Various network architectures have been demonstrated in this study with varying degrees of success. For a commercial system these architectures will need to be extended. Because the network is a very dynamic system the relationship between network architecture and training algorithm needs to be carefully considered.

Other applications have shown that the use of feed back paths can greatly enhance the sensitivity of ANN's. The possibility of using these types of architectures, particularly to reduce the effect of broken instrumentation needs to be investigated.

9.2.4. Input data requirements.

The more data available to a diagnostic system the more detailed the diagnosis can be. Extensive instrumentation is, however, very expensive. Further research is clearly needed to analyse what data is most useful and the most cost effective way of retrieving it.

9.2.5. The development of a detailed simulator.

The examples given in this study are based on great simplifications of the real CHP plant. It is, however, very clear from all these examples that large amounts of data are required in order to train any diagnostic system. Computer simulation may be the only cost effective way to generate this data for the diverse range of equipment to which such diagnostic systems could be applied. Any simulator needs to be very comprehensive to be able to accurately represent the plant under the wide range of operating and ambient conditions that are possible.

9.2.6. The potential for analogue solutions

In the literature survey the concept of analogue devices that are being developed for building neural systems is discussed. The potential of these techniques should be investigated for this application.

9.3. Verification methods for a diagnostic system

As with all non-linear systems, neural networks are extremely complicated to investigate analytically and so much of the work above would need to be conducted experimentally.

A neural diagnostic system could only be verified by extensive field trials. A systematic method of testing would therefore need to be developed so that the different methods can be compared fairly. Several methods have been used in this study. The most fundamental technique is the interrogation of the trained network with the input data set used for training. This is to see how well the network can reconstruct the original output relationship. Tests with random numbers and data sets not previously encountered have also been used and could be refined.

Chapter 10

Overview and conlusions

10.1. Discussion of the experimental results

The systems described in the previous chapters clearly show that artificial neural networks (ANNs) are capable of learning the complex relationships between performance parameters of both the CHP plant and the Diesel engine test cell. This in itself can be useful as it means that an ANN can be used to predict what values are expected at a particular operating point. This is illustrated in the performance predictor example. This technique can be used to produce base-line values from which changes in expected performance can be calculated. These changes can be used to diagnose system faults from the pattern of change or to highlight individual transducers that are giving unexpected readings. This may help identify transducers that may be faulty.

Diagnostic networks have been demonstrated that can relate performance directly with state of health. It is shown that these networks work significantly better on fault diagnosis problems than health recognition whereby they can differentiate between known data sets rather than responding to one known data set amongst several unknown ones.

The examples show that it is valid to train such networks on simulated data because they are to able to analyse real results effectively. The research does show that great care is needed in the development and fine tuning of the individual simulation before accurate results can be established over the entire operating envelope of an engine.

It is demonstrated that it is much more difficult to train networks to recognise absolute values than it is to train them to recognise deviations from the normal. It is clearly shown that they train easiest with binary type patterns of input. The advantage of the neural network approach for binary pattern recognition over other conventional mathematical pattern recognition techniques is that the output is not restrained to binary values, a value approaching 1 will be given if the input pattern is almost correct. The

advantages of the ease of training of the binary type input networks will become significant as the networks become more complex as more faults are introduced.

There can be problems of setting error levels desired during training. If they are set too low then the network becomes too sensitive to the orignal data and are sensitive to noisy data. Too high and they do not produce clear outputs. The noise problem can be overcome with filtering but then there are problems with setting filter levels.

As the examples show, the neural networks show potential in identifying multiple faults without extra programming, this is a particularly difficult problem for conventional pattern recognition methods.

The on-line example shows that a diagnostic system using neural networks can be run effectively on a simple computer platform such as a PC. The generation of training data required extensive computer resources, test cell experimentation and time. However, the implementation of the acquired knowledge using the neural network takes only a fraction of a second on the PC.

10.2. Proposed neural network system

The models shown in the previous chapters are clearly very simple and as such are very limited for practical application in their present form. They do, however, show great potential and should provide a useful basis for designing a practical diagnostic system. By combining ideas from several models discussed here a viable system could be constructed. A suggested configuration is illustrated in figure 10.1 below.

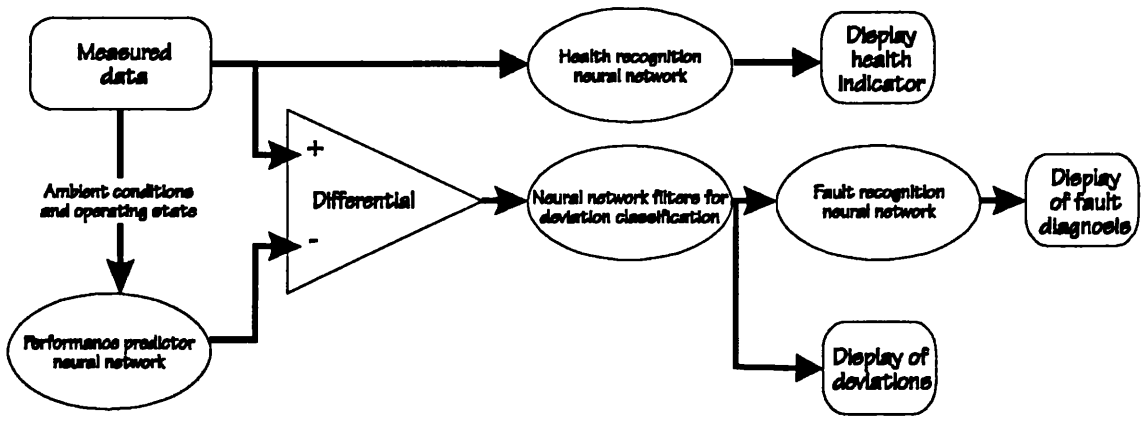


Figure 10.1 An integrated diagnostic system.

In this proposed system, the measured ambients and engine load settings are passed though a performance predictor network to get the expected performance. This is subtracted from the measured data to form a deviation. This is streamed by a set of neural filters into several discrete channels depending on the amount of deviation. This data is then passed through a fault identification network to formulate a diagnosis.

The benefits of a modular approach such as this is that each part can be trained and tested individually. The data passing between each section can also be interrogated to assist with the diagnosis - the banded data could be used directly to drive indicator lights that may be of use to an operator. The data could also be divided up and shared amongst several specialist networks.

10.3. Conclusions

This thesis describes the limitations in the methods used in current commercial diagnostic systems. Conventional expert system techniques are very susceptible to mis-diagnosis due to many reasons, including; instrumentation failure, noisy data and the presence of multiple faults. They are also extremely demanding in terms of the hardware necessary to store and process the large knowledge databases that they employ. Neural networks have been used very effectively in other knowledge processing systems. These systems have proved to be very efficient on resources and behave well when used on real data.

Several experimental systems are described that could form the basis of a practical engine fault recognition and diagnosis system. The requirements for on-line computer hardware and software are shown to be much less than for existing condition monitoring systems. Processing of operating data is very rapid, taking a fraction of a second on the model systems described.

The artificial neural networks have been trained using computer simulated engine data. The generation of this data is demanding but the data generation and training functions would be carried out off-line when the system is configured for a particular application. The simulation and training tasks for some of the examples described would require an equivalent of several days computing on a conventional 486 based PC.

In order to calibrate the simulation it has been shown to be necessary to obtain experimental data from the engine in its healthy state. Such data would normally be available from the test bed evaluation of the particular

engine prior to delivery, or from data generated during the engine development programme. The chief role of the simulator is to generate engine responses for a wide range of possible faults, which could not possibly be generated experimentally due to potenial risk to the engine and the time that such tests would take.

The thesis describes the methods that could be used to make and train a practical neural network diagnostic system and the different ways in which such a system could be implemented. The further work that would be needed before such a system could be implemented commercially has been highlighted in the preceeding Chapter.

The artificial neural network industry is rapidly expanding and the thesis points towards the types of technology that may be available to assist this approach to knowledge based systems in the near future.

The neural network techniques are shown to give not only a means of processing complex data but a simple environment in which several processing systems can be combined to form a complete intelligent system.

Appendix A

Mathematical Basis of the Back Propogation Method

A.1 Introduction

This algorithm detailed by Rumelhart [38] is used for solving multilayer neural networks as shown in figure.

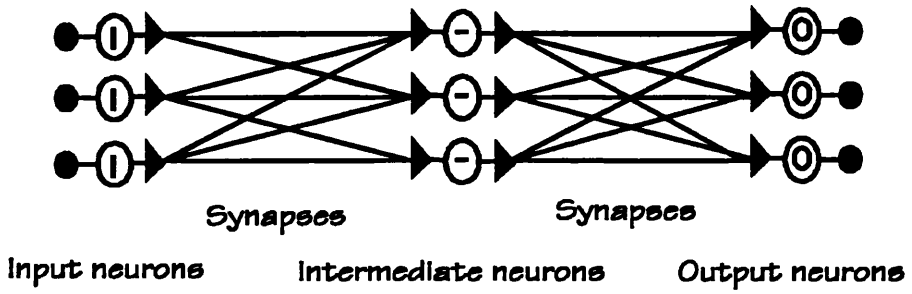


Figure A.1 Schematic model of the multi-layer neural network

The output of the *i*th neuron in layer m, V_i^m is given by the sigmoid function:

$$V_i^m = \frac{1}{1+e^{\left(\sum_j w_{ij}^m V_j^{m-1} + \varphi_i\right)}} \qquad \text{..... A.1}$$

A.2 Algorithm

1 Set all weights to small random values

2 Present an input vector **I** and desired output vector **O**. Apply **I** to the input layer (m=0) so that $\mathbf{V^0}$=**I**.

3 For other layers, namely m=1,...,M, perform the forward computation:

$$V_i^m = f\left(\sum_j w_{ij}^m V_j^{m-1}\right) \quad \text{.....A.2}$$

where w_{ij}^m represents the connection weight from V_j^{m-1} to V_i^m

4 Compute the errors in the output layer:

$$\delta_i^M = V_i^M\left(1-V_i^M\right)\left(O_i - V_i^M\right) \quad \text{.....A.3}$$

5 Compute the back propogation errors for the preceeding layers M-1, ..., 1

$$\delta_i^{m-1} = V_i^{m-1}\left(1-V_i^{m-1}\right)\sum_j w_{ji}^m \delta_j^m \quad \text{.....A.4}$$

6 Adjust the weights:

$$w_{ij}^m(t+1) = w_{ij}^m(t) + \eta\delta_i^m V_j^{m-1} \quad \text{.....A.5}$$

where η is the gain parameter. Thresholds are adjusted in a way similar to weights

7 Repeat by going to step 2

The algorithm continues until the overall error, which is the mean square difference between the desired and actual outputs for all training patterns is reduced to an acceptable level.

Appendix B

Performance calculations used on British Gas CHP data

B.1 Introduction

This section outlines the performance clacultions that were used to analyse the results from the British Gas experimental combined heat and power plant. These include calculations of:

- Engine coolant mass flow rate kg/s
- Site water coolant mass flow rate kg/s
- Gas mass flow rate kg/s
- Air mass flow rate kg/s
- Density of engine coolant kg/m^3
- Density of site coolant kg/m^3
- Specific heat of engine coolant J/kgK
- Specific heat of site coolant J/kgK
- Specific heat of exhaust gases J/kgK
- Density of gas kg/m^3
- Calorific value of gas J/kg
- Power into plant from gas W
- Power transfer in each heat exchanger W
- Heat exchanger effectiveness for each %
- Heat loss from generator W
- Generator efficiency %

Each of these calculations will be detailed below.

B.2 Engine coolant mass flow

Engine coolant mass flow, $\dot{m}_{ec}$(kg/s) is derived from the measured volume flow rate, $\dot{V}_{ec}$(m^3/s), and the calculated density, ρ_{ec} (kg/m^3).

$$\dot{m}_{ec} = \dot{V}_{ec} \cdot \rho_{ec} \quad \text{.....B.1}$$

B.3 Site water coolant mass flow

Site coolant mass flow, $\dot{m}_{sc}$ (kg/s), is derived from the measured volume flow rate, $\dot{V}_{sc}$ (m^3/s), and the calculated density, ρ_{sc} (kg/m^3)..

$$\dot{m}_{sc} = \dot{V}_{sc} \cdot \rho_{sc} \quad \text{.....B.2}$$

B.4 Gas mass flow

Gas mass flow, $\dot{m}_g$ (kg/s) is derived from the measured volume flow rate $\dot{V}_g$ (m^3/s) and the calculated density ρ_g (kg/m^3) at the measured temperature and pressure (see below).

$$\dot{m}_g = \dot{V}_g \cdot \rho_g \quad \text{.....B.3}$$

B.5 Air mass flow

Air mass flow, $\dot{m}_a$ (kg/s) is derived from the calculated gas mass flow, $\dot{m}_g$ (kg/s), and the fixed air fuel ratio λ_f. This ratio is assumed to be fixed because it is measured and controlled by the CHP control system.

$$\dot{m}_a = \dot{m}_g \cdot \lambda_f \quad \text{.....B.4}$$

B.6 Density of engine and site coolants

The density of the engine and site coolant, ρ (kg/m^3), is calculated from the density, ρ (kg/m^3), and mass fraction, r (-), of the two constituents; water (suffix H_2O) and ethylene glycol (suffix eg).

$$\rho = r_{H_2O} \cdot \rho_{H_2O} + r_{eg} \cdot \rho_{eg} \quad \text{.....B.5}$$

B.7 Specific heat of engine coolant

The Specific heat of the engine and site coolant, Cp (J/KgK), is calulated from the specific heats, Cp (J/kgK), and mass fraction, r (-), of the two constituents; water (suffix H_2O) and ethylene glycol (suffix eg).

$$Cp = r_{H_2O}.Cp_{H_2O} + r_{eg}.Cp_{eg} \quad \text{.....B.6}$$

B.8 Cp of exhaust gases

The specific heat of the exhaust gas $Cp_{exhaust}$ (J/kgK) is determined from the measured gas temperature $T_{exhaust}$ (K) and the air/fuel ratio λ using the polynomial:

$$Cp_{air} = c_1 + T_{ex}.(c_2 + T_{ex}.c_3) + \frac{1}{\lambda}.(c_4 + T_{ex}.(c_5 + .(c_6 + T_{ex}.c_7))) \text{ where,}$$

$$c_1 = 920.7128$$

$$c_2 = 280.5702e^{-3}$$

$$c_3 = -54.0389e^{-6}$$

$$c_4 = 101.4663$$

$$c_5 = 2.522374$$

$$c_6 = -1.054534e^{-3}$$

$$c_7 = 175.6886e^{-9}$$

....B.7

B.9 Density of gas

The gas density, ρ_g (kg/m^3), is determined by the density of the gas at NPT (10^5 N/m^2, 298 K), $\rho_{g\,NPT}$ (kg/m^3), which is found from the mass fraction, r (-), and densities, ρ (kg/m^3), of the constituent gases at the measured gas temperature, T_g (K), and pressure, P_g.(N/m^2)

$$\rho_{g_{NPT}} = \sum_{gases} \rho.r \quad \text{.....B.8}$$

$$\rho_g = \rho_{g_{NPT}} \cdot \frac{p_g}{p_{NPT}} \cdot \frac{T_{NPT}}{T_g} \quad \text{.....B.9}$$

The gases and mass fractions are given in table B.2

Gas	Formula	Mass fraction %age	Density at NPT kg/m3	Density contribution kg/m3
Methane	CH_4	85.64%	0.7160	0.6637
Ethane	C_2H_6	7.52%	1.3420	0.0582
Propane	C_3H_8	2.36%	1.9670	0.0183
Butanes	n-C_4H_{10}	0.97%	2.5930	0.0075
Pentanes	C_5H_{12}	0.33%	3.2190	0.0026
Nitrogen	N_2	1.81%	1.1600	0.0130
Carbon dioxide	CO_2	1.37%	1.8000	0.0097
Totals		**100.00%**		**0.7731**

Table B.1 Density contributions of natural gas fuel

B.10 Calorific value of gas

The calorific value, Q_{HV} (J/kg), is determined by the mass fraction, r (-), and calorific values, Q_{HV} (J/kg), of the constituent gases given in table B.2 below according to the relationship

$$Q_{HV} = \sum_{gases} Q_{HV} . r \quad \text{.....B.10}$$

The gases and mass fractions are given in table B.2

Gas	Formula	Mass fraction %age	Calorific value MJ/kg	contribution MJ/kg
Methane	CH_4	85.64%	50.01	42.83
Ethane	C_2H_6	7.52%	47.49	3.57
Propane	C_3H_8	2.36%	46.35	1.09
Butanes	n-C_4H_{10}	0.97%	45.73	0.44
Pentanes	C_5H_{12}	0.33%	45.37	0.15
Nitrogen	N_2	1.81%	0.00	0.00
Carbon dioxide	CO_2	1.37%	0.00	0.00
Totals		**100.00%**		48.09

Table B.2 Calorfic value of natural gas fuel

B.11 Power into plant from gas

Power into the plant from the gas, Q_g (W), is determined from the calculated mass flow rate, $\dot{m}_g$ (kg/s), (see above) and the calculated calorific value, Q_{HV} (J/kg) (see above)

$$Q_g = \dot{m}_g . Q_{HV} \quad \text{.....B.11}$$

B.12 Power transfer in each heat exchanger

Power transfer across the heat exchangers, Q (W), is determined by the calculated mass flow rates, $\dot{m}$ (kg/s), the calculated specific heats, Cp (J/kgK), and the measured temperature differential, ΔT (K), across the heat exchanger according to the relationship:

$$Q = \dot{m}.Cp.\Delta T \quad \text{.....B.12}$$

B.13 Heat exchanger effectiveness

The heat exchanger effectiveness, ε (-), is calculated from the calculated power transfer, Q_{actual} (W), and the theoretical maximum transfer possible determined from the measured inlet temperatures, T (K), and the lowest product of calculated specific heat, Cp (J/kgK), and mass flow rate, $\dot{m}$ (kg/s), of the two working fluids.

$$\varepsilon = \frac{Q_{\text{actual}}}{(\dot{m}.C_p)_{\min}(T_{hot,in} - T_{cold,in})} \quad \text{.....B.13}$$

B.14 Heat loss from generator

The heat lost from the generator, Q (W), is calculated from the measured temperature rise of the cooling air, ΔT (K), and an assumed air mass flow, $\dot{m}$ (kg/s) determined from previous experimentation using the equation:

$$Q = \dot{m}.Cp.\Delta T \quad \text{.....B.14}$$

The specific heat of the air, Cp_{air} (J/kgK), is determined from the measured ambient temperature, T_{air} (K) using the polynomial:

$$Cp_{air} = c_1 + T_{air}.(c_2 + T_{air}.c_3) \quad \text{where,}$$

$$c_1 = 920.7128 \qquad \text{.....B.15}$$

$$c_2 = 280.5702e^{-3}$$

$$c_3 = -54.0389e^{-6}$$

B.15 Generator efficiency

The generator efficiency, $\upsilon_{generator}$ (-), is determined from the measured electrical power, W (W), and the calculated heat loss, Q (W).

$$\upsilon_{generator} = \frac{W_{electrical}}{Q_{heat}} \qquad \text{.....B.16}$$

Appendix C

ADVANCED DATA ACQUISITION TECHNIQUES FOR MULTI-TASKING PERSONAL COMPUTERS

IMechE Seminar - *PC's in Engineering*, 1993

C.G.Mobley, M.W.Scaife, P.H.Prest, S.J.Charlton.
School of Mechanical Engineering, University of Bath.

SYNOPSIS

With the recent introduction of multi-tasking operating systems the personal computer can now offer the engineer a complete set of powerful tools to aid data analysis. The aim of this paper is to demonstrate how separate software packages can be combined in this environment for the specialist activity of data acquisition and post-processing. The paper describes a complete hardware and software system designed to aid the engineer in the capture and analysis of complex, real-time waveforms and signals. The system utilises both standard data acquisition hardware and hardware developed at the University for very high speed, high accuracy data capture on multiple channels. The research uses commercial software, such as spread-sheets and graph plotting packages, but the binding agent which integrates these parts together is software developed at the University.

NOTATION

ADC	Analogue to Digital Converter
DDE	Dynamic data exchange
DMA	Direct memory access
GUI	Graphical user interface
KSPS	Kilo samples per second
MPU	Microprocessor unit
MSPS	Mega samples per second
OLE	Object linking and embedding
PC	Personal computer (IBM PC-AT)
RAM	Random access memory
SRAM	Static random access memory

1. INTRODUCTION

The arrival of the new generation of powerful desk-top personal computers (PCs) has provided a platform for conducting the advanced data processing common in modern engineering. The type of data processing that previously necessitated the use of main frames can now be conducted on the desk, or portable PC.

Previously software had to be written with the hardware in mind, the speed of execution and the size of memory were critical to performance. Software could not easily serve the needs of the user because of the limitations of the hardware.

Improvements in PCs have allowed the software to become better matched to the needs of the user. The advances in graphical displays have allowed better presentation

and hence made packages easier to understand and interact with.

The increasing power of the micro processor used in PCs has meant that multi-tasking operating systems are now being used []. These allow multiple software packages to run concurrently on a single machine. Standard protocols have been developed in these operating systems that allow software packages to exchange data easily. This development has meant that the user is no longer restricted to the limitations of a single software package because he can easily create links to transfer data to other packages if required.

To take full advantage of this new technology the research described here has re-investigated the concepts of data acquisition and a new strategy has been developed.

The focus of this work has been on the flow of information through multi-tasking systems and its interaction with the user.

The work was first developed in conjunction with research into the processing of data from internal combustion engines by neural networks [2]. In this work, data was collected from a Diesel engine by commercial data acquisition cards with software written at the University. The information is passed to a spreadsheet for processing. Specific data items are extracted from the spreadsheet and conveyed to a neural network package for subsequent analysis. All these activities run concurrently in the *Windows 3.1* environment.

Another application that has been used to investigate these techniques has been the high speed data capture of information from a moving vehicle used in research. This work utilised special data acquisition hardware developed at the University connected to a laptop PC in the vehicle. Special software allowing the control of the high data flow rates has been written to work in the *Windows* environment.

The paper describes how the software and hardware integrates with the multi-tasking operating system and the commercial software packages used for data presentation.

2. DATA FLOW - A DIFFERENT APPROACH

The main purpose of instrumentation is to provide a detailed and accurate set of data on the performance of the equipment under observation. This data is critical to the correct understanding of the equipment and its operation. The integrity of the data, from the sensors, through the computer and to the operator is of vital importance for error free results.

The flow of data through a system will dictate the design of the software, and is the basis of the new operating environments. The basic concept is to deal with data as a fundamental resource which is then acted upon by various tools. These act on the data in different forms. The data sets are referred to as *objects* and the techniques of manipulating them is called ***object linking and embedding*** (OLE) [3]. The data flow through the acquisition system is no longer regarded as an intangible part of a complex process, but rather as a dynamic stream of information which can be presented in any form chosen by the user.

3. MULTI-TASKING PCS AND THE GRAPHICAL USER INTERFACE

The *Graphical User Interface* (GUI) is now standard on PCs and provides a platform for running multiple software packages, including new data acquisition systems. However, there are some criticism of these GUIs. response is very dependent on the number of processes running and the internal architecture of the software. Straight conversions of long serial programs designed for conventional operating systems to the new object orientated environment do not work efficiently. However, many of the speed associated problems can often be overcome by hardware upgrades.

There are several multi-tasking operating systems available for PCs but *Windows 3.1* is currently the most widely used. *Windows* incorporates a *graphical user interface*, it conforms to the *object linking and embedding* philosophy. It also provides built in functions to allow data exchange between software packages. One such method is

dynamic data exchange (DDE) which is a protocol widely used to link software packages. The ***Windows*** operating environment therefore provides a powerful framework for the integration of different products. Because of the popularity of this environment there is also very good software and hardware support.

For the above reasons the University has chosen ***Windows 3.1*** as the platform for the recent development of acquisition systems.

4. APPLICATIONS

Engineering experimentation involves the collection two types of data. The first is ***quasi-static***, in which the data is steady or slowly fluctuating during a test. The second is ***high speed*** which is captured in a 'snapshot'. For example, in IC engine research both sets of data are present. The engine coolant temperature remains constant, cycle to cycle, but is variable over longer periods, whereas in-cylinder pressures vary through the cycle and a snapshot of data is required. Acquisition systems need to provide for both types of data.

For the quasi-static data, samples are taken at slow intervals (up to 100 Hz). At these rates they can be presented live to a spread-sheet via the ***dynamic data exchange*** protocol.

Large quantities of fast moving data dramatically reduce system response and resources, so for high speed data capture alternative methods need to be used. These include; ***direct memory access*** (DMA) and ***buffer memory***. DMA is a hardware device which is used to transfer data direct from the data acquisition card to the PC's main memory avoiding any restrictions caused by the execution of slow programs. The advantage with DMA is cost, the PC has an in-built capability and so is cheap to implement. However, system contentions can limit the maximum data rate, and there are problems with data integrity.

Buffer memory is additional memory that resides on the acquisition card and can be written to at any time and speed by the analogue to digital converters (ADC's). The information can be transferred to the PC when convenient. In contrast to DMA the *buffer memory* approach offers virtually unlimited acquisition rates but suffers from the cost of the additional memory required. However, the cost of memory is continually reducing and so this option is becoming cheaper. It is therefore this option which was chosen for research.

5. DATA ACQUISITION CARDS

The PC interfaces to the outside world by using ***expansion cards*** which plug into the computer's system bus. These cards are widely available to fulfil a range of functions. For low speed data standard multi-function acquisition cards may be used.

For higher data rates it is felt that *buffer memory* cards provide the best solution in terms of simplicity and convenience. There are several suitable acquisition cards with on-board memory available, however in the area of ultra high speed data acquisition, over 1 million samples per second (MSPS), the choice is very limited. Therefore, a card was specifically designed at the University.

6. THE UNIVERSITY OF BATH DATA ACQUISITION CARD

This card was designed to fulfil the needs of high speed data capture from Diesel engines. It can provide 40 Mega samples per second (MSPS) of 10 bit data capture. There is 1 MByte of on board memory, which provides sufficient space to capture two channels of data from an engine over many cycles at maximum speed. The design is described in appendix A.

7. DRIVER SOFTWARE

To enable an efficient system to be created there must exist a flexible link between the expansion cards, the PC and the user software. Via this link, data from the expansion card is fed to the user software.

If this link is slow and unpredictable, data gathered may be of little value due to errors and missing blocks. Many expansion cards are available with links into the standard PC disk operating system (MS-

DOS), but few provide the necessary links into the *Windows 3.1* environment. MS-DOS drivers provide the facility to transfer data to a single MS-DOS program, but *Windows* drivers must work in a multi-tasking environment and be able to transfer data to multiple programs. The unpredictable nature of the operating state of multi-tasking environments and the complexity of the system mean that the drivers written for MS-DOS cannot be used for environments such as *Windows.*

The driver software must provide all the necessary controls and links to enable the user to be connected to the data stream. These links are transparent to the user but a driver window is provided to control the acquisition of data and other relevant test parameters.

8. A USER INTERFACE FOR IC ENGINES

A system for work with IC engines has been constructed and tested to record data from an engine and continuously variable transmission (CVT) installed in a car. The data collected has both low and high speed content and is captured during a 30 second run.

The system consists of three data acquisition devices;

(i) a low speed multi-function card,

(ii) a high speed card,

(iii) a timer/counter card.

There is also driver and user interface software. The architecture of the system is illustrated in Figure 1.

The multi-function card provides temperature and slow speed pressure information and monitors various status parameters. The engine speed, CVT primary and secondary shaft speeds are measured by the timer/counter card.

This data is presented to the PC at 0.5 second intervals using the interrupt lines to request the attention of the computer's operating system. When called, the driver's interrupt routine transfers the data to system memory (and disk) and posts a *Windows* protocol message to secondary packages such as spreadsheets to inform them of new data. As the live data updates the spreadsheet, the system under test can be analysed and the data stream examined. Figure 2 shows a typical *quasi-static* live output using Microsoft's spread-sheet ***Excel.***

The high speed card is activated by the driver software under user control and provides a *snapshot* of the data stream. This snapshot is then transferred from the card's own *buffer memory* to the PC's disk and to the display software ***Signal Spy,*** described below. In this example, the high speed data card records hydraulic oil pressure transients in the CVT transmission.

On completion of the test, the high speed data can be read into a spreadsheet. Time varying graphs can then be generated and examined. Figure 3. shows a typical output from ***Origin,*** a *Windows* based scientific graphing package.

9. SIGNAL SPY

To view the large amounts of data generated by high speed cards a new display package has been developed at the University called ***Signal Spy***.

This package is designed to view long data streams, up to the limit of acquisition memory. The package integrates into the data stream through the driver software and provides a method for visualising high speed data as it is generated. ***Signal Spy*** has been designed specifically to display 4 channels of data with unlimited sample size, subject to the amount of system memory available.

At present the package is still in the development phase and plans for the future include full OLE compatibility.

10. FUTURE DEVELOPMENTS

As the cost of high speed memory decreases the ability to place data analysis and signal processing on the expansion cards becomes more cost effective. New RISC microprocessors are becoming available and there are great benefits gained by placing a RISC processor, fast memory and high speed A/D converters on the acquisition card. These include the ability to condense the data from raw information to a

higher form on the acquisition card. Hence the quantity of data transferred to the PC is substantially reduced but the content is enhanced.

In this area, a set of ultra high speed acquisition cards is being designed at the University. These cards provides an entire sub-processing system which can enable high speed Digital Signal Processing. The system consists of three expansion cards which reside on the PC-AT Bus and communicate together across their own separate high speed bus.

The system can provide 120 MSPS of 12 bit data capture. This is supported by 1 Mega word of acquisition memory.

This system is still in the design phase but is partly constructed.

Second generation true multi-tasking operating systems such as ***Windows New Technology (NT)*** will be available soon and this will make the dominance of the PC-AT architecture weaker. (***Windows 3.1*** is only *pseudo* multi-tasking as it still uses MS-DOS as a sub shell). The basis of ***NT*** is to provide a common GUI for any machine irrespective of make and model. Indeed, ***Windows NT*** Beta versions are already available for both Intel and R4000 RISC architectures. This may dramatically affect the market for expansion cards and acquisition systems.

The built in features of NT for networking can be exploited to enable a common acquisition system to exist across many machines. The data acquisition system described above is being developed for use with networks and the PC-AT bus, to cover the needs of ***Windows 3.1*** and ***NT***.

11. CONCLUSIONS

In this paper work has been presented which attempts to provide a set of useful tools for the engineer. This work has proved to be successful for a number of different research projects. The research into continuously variable transmission has shown that a fully integrated acquisition systems can provide the system required to capture, analyse and present various data types required by the project.

There are strong indications that the future of computing will be based around distributed autonomous multi-tasking PCs and mini-computers that will be using ***Windows NT*** as the main operating system. Networking is a standard part of this operating system and will be used to share common resources such as printers and extra storage space. Thus future work will concentrate on providing data acquisition systems that will fit in to this complex area.

12. ACKNOWLEDGEMENTS

Borland are gratefully acknowledged for their software support. Kudos Thame Ltd., Advanced Micro Devices and Microprocessor & Memory Distribution Ltd. are also acknowledged for their technical and financial help.

13. REFERENCES

1 Scaife M.W. ***An Experimental Facility for The Development of Intelligent Engine Diagnostics,*** M.Phil dissertation, University of Bath, Bath, UK. 1990.

2 Scaife M.W., Charlton S.J, Mobley C.. ***Engine Fault Diagnosis using a neural network approach,*** SAE international congress 1993

3 Klemond P. ***Taking the Bull by the Horns: Investigating Object Linking and Embedding, Part 1,*** Microsoft Systems Journal, Vol. 1 No 2 April 1992.

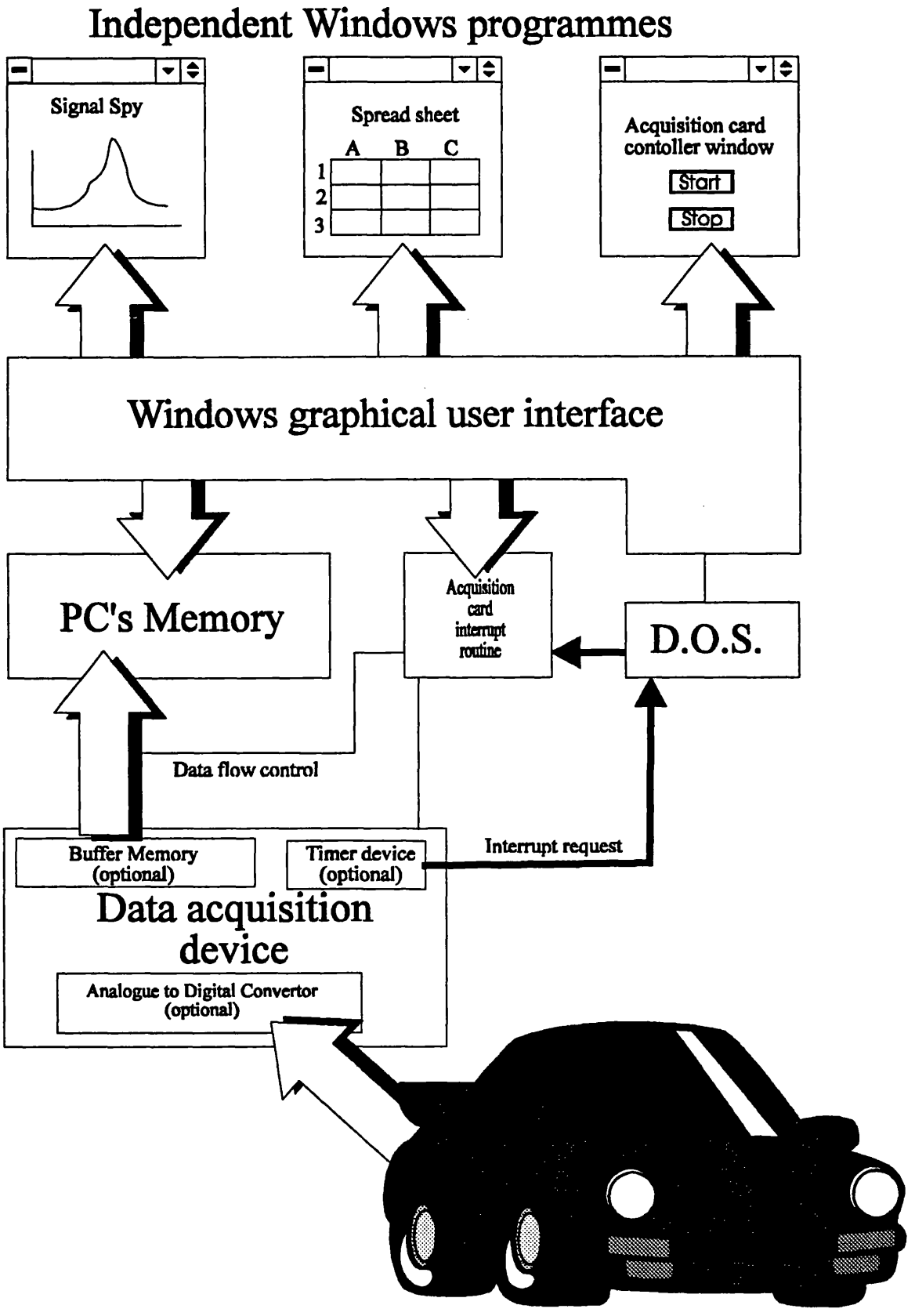


Figure 1. DIAGRAM OF THE ACQUISITION SYSTEM.

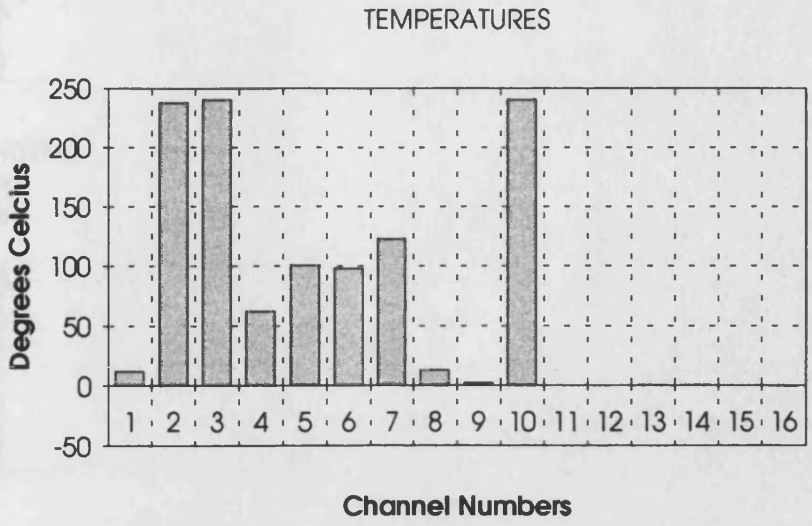


Figure 2. BAR CHART FROM EXCEL

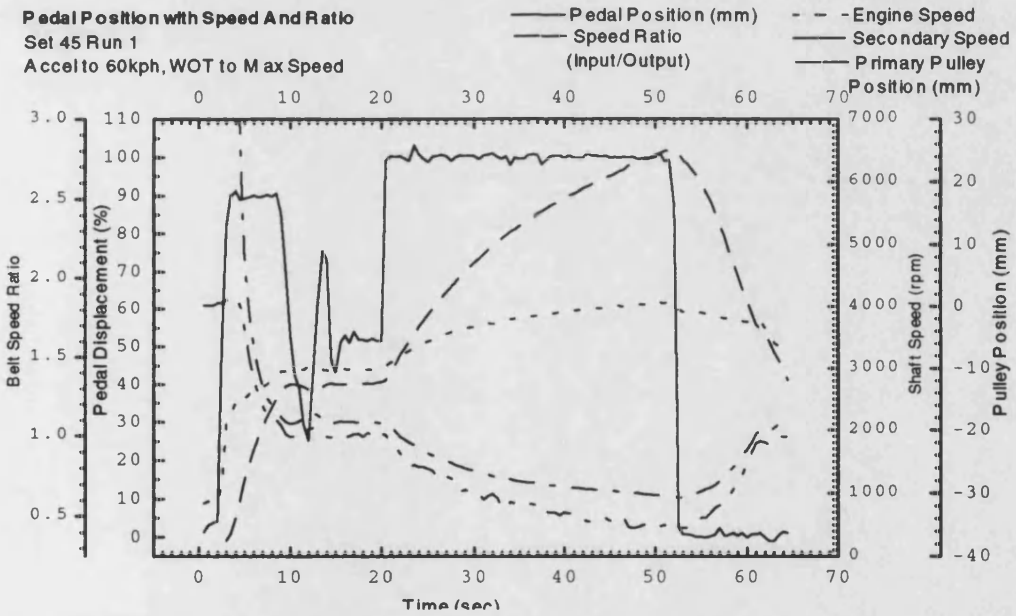


Figure 3. CVT OUTPUT FROM ORIGIN

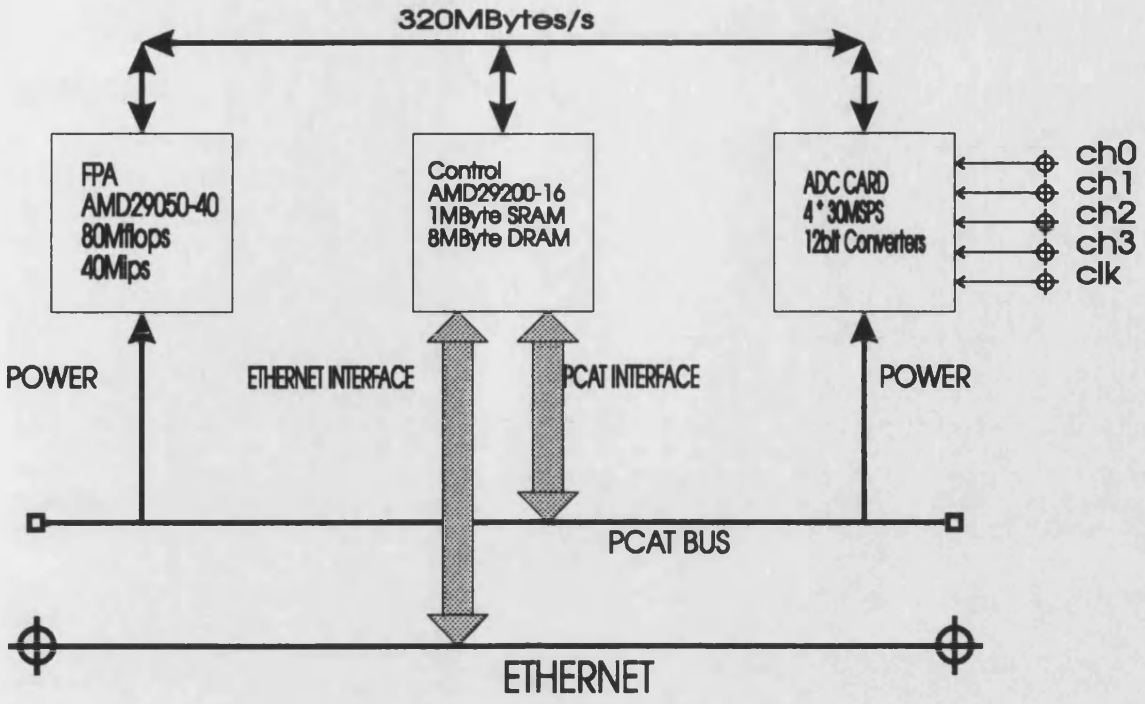


Figure 4. BLOCK DIAGRAM OF HIGH SPEED 3 CARD ACQUISITION SYSTEM.

APPENDIX A. ULTRA HIGH SPEED ACQUISITION CARD

The design of the card was based around two 10 bit 20 MSPS a/d converters and 256 KWords of *static random access memory* (SRAM). To enable continious conversion at the 20 MHz rate the SRAM needed an access time of 20 ns. Hitachi 256k by 1 - 20 ns SRAM'S were chosen and to provide 2 * 10 bits, 20 devices were needed. To enable automatic data capture a method of RAM addressing was required. These addresses was generated by an 18 bit counter which was implemented in a 15 ns EPLD - Intel 85C090-15.

The time available between samples is 50 ns, the SRAM need 20 ns for access, and the address generator needs 15 ns to generate the next address. This leaves a safe 15 ns for signal propagation and settle time. The system could achieve a higher data rate by;

1. using faster SRAM 10 ns is available.
2. a faster EPLD 12 ns and
3 the address time can be masked by the access time of the SRAM, this is shown in Figures 5 and 6.

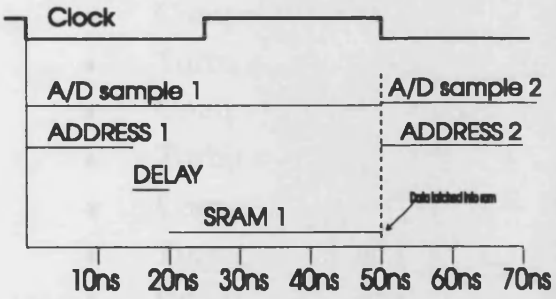


Figure 5.

Once a trigger is received the system automatically starts generating addresses at a rate governed by the external clock, usually fed from an encoder. This address is presented to the SRAM and the data from the flash converters is stored at that location. Once the counter reaches 2^17 the trigger is released and the system returns to the inactive state.

While the board is in this inactive state the driver software is able to access the memory. To read the data the driver software places an address in the interface and then reads the contents of the memory accessed by that address. It must do this for all 256k locations to read all the data.

If the trigger is activated before the memory is transferred the old data is overwritten.

The prototype acquisition board was wire wrapped on to a blank expansion card and has been tested to 10 MHz. For full performance tests to be generated the design needs migrating to a PCB.

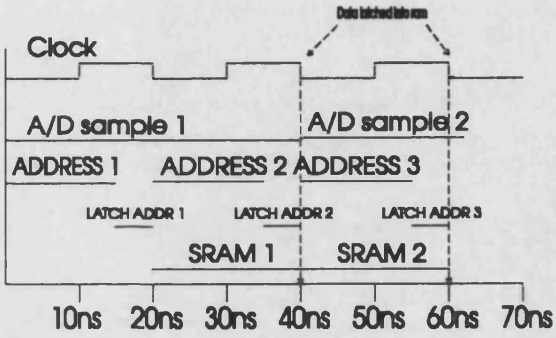


Figure 6.

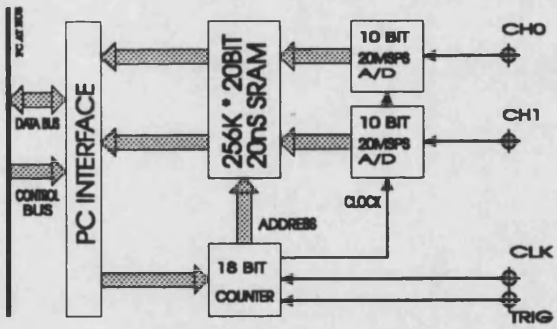


Figure 7 BLOCK DIAGRAM OF THE HIGH SPEED DATA ACQUISITION CARD

Appendix D

Performance calculations used on TL-11 data

D.1 Introduction

This section outlines the performance calculations that are used to analyse the results from the TL-11 Diesel test cell at the University of Bath. These include calculations of:

- Brake Power W
- Brake mean effective pressure N/m^2
- Brake thermal efficiency -
- Brake specific fuel consumption g/kWhr
- Air mass flow rate kg/s
- Air/fuel ratio -
- Volumetric efficiency -
- Compressor pressure ratio -
- Turbine pressure -
- Compressor density ratio -
- Turbine density ratio -
- Compressor mass flow parameter $kg\sqrt{K}m^2/N$
- Turbine mass flow parameter $kg\sqrt{K}m^2/N$
- Compressor efficiency -
- Turbine efficiency -
- Compressor power W
- Turbine power W
- Iintercooler pressure ratio -
- Intercooler density ratio -
- Intercooler effectiveness -

Each of these will be detailed below.

D.2 Brake power

Engine brake power, W_{brake} (W) is calculated from the measured brake torque, T_{brake} (Nm), and the measured speed, N (r/min).

$$W_{brake} = T_{brake} . N . \frac{2\pi}{60} \quad \text{..... D.1}$$

D.3 Brake mean effective pressure

Engine brake mean effective pressure, *bmep* (N/m^2), is derived from the calculated brake power, W_{brake} (W) , the measured speed, N (r/min) and the known engine geometry including: stroke, s (2 or 4), displacement volume of each cylinder Vc (m^3) and the number of cylinders n_c.

$$bmep = W_{brake} . \frac{60}{N} . \frac{s}{2 . n_c . V_c} \quad \text{..... D.2}$$

D.4 Brake thermal efficiency

The brake thermal efficiency, η_{brake} (-), is derived from the the measured fuel mass flow rate, $\dot{m}_f$ (kg/s), the calculated brake power, W_{brake} (W), and the assumed calorific value of the fuel, Q_{HV} (J/kg)

$$\eta_{brake} = \frac{W_{brake}}{\dot{m}_f . Q_{HV}} \quad \text{..... D.3}$$

D.5 Brake specific fuel consumption

The brake specific fuel consumption, *bsfc* (g/kWhr), is derived from the calculated brake power, W_{brake} (W), and the measured fuel mass flow rate, $\dot{m}_f$ (kg/s)

$$bsfc = \frac{\dot{m}_f . 60 . 60 . 10^6}{W_{brake}} \quad \text{..... D.4}$$

.

D.6 Air mass flow rate

The air mass flow rate, $\dot{m}_a$ (kg/s), is derived from the following parameters:

Measured:	p_a	Ambient absolute pressure	(N/m^2)
	p_{og}	Inlet gauge pressure	(N/m^2)
	p_{ot}	Pressure difference across orifice	(N/m^2)
	T_o	Upstream temperature	(K)
Calculated	p_o	Upstream absolute pressure	(N/m^2)
	p_t	Orifice throat absolute pressure	(N/m^2)
Assumed	R	gas constant (= 287.1)	(J/kgK)
	Cd	Disharge coefficient (= 0.9867)	(-)
	A_t	Orifice throat area (= 0.0045603)	(m^2)
	γ	Cp/Cv (= 1.4)	(-)

The calculations used are as follows:

$$p_o = p_a + p_{og}$$

$$p_t = p_o + p_{ot}$$

$$m_a = \frac{Cd.A_t}{\sqrt{R}} \cdot \sqrt{\frac{2\lambda}{\gamma - 1}} \cdot \frac{p_o}{\sqrt{T_o}} \cdot \left(\frac{p_t}{p_o}\right)^{\frac{1}{\gamma}} \cdot \sqrt{1 - \left(\frac{p_t}{p_o}\right)^{\frac{\gamma-1}{\gamma}}} \quad \text{.....D.5}$$

D.7 Air/fuel ratio

The air fuel ratio, λ (-), is derived from the calculated air mass flow rate, $\dot{m}_a$ (kg/s), and the measured fuel mass flow rate, $\dot{m}_f$ (kg/s).

$$\lambda = \frac{\dot{m}_a}{\dot{m}_f} \quad \text{.....D.6}$$

D.8 Volumetric efficiency

The volumetric efficiency, η_{vol} (-), is determined from the calculated air mass flow rate, $\dot{m}_a$ (kg/s), the measured crank speed, N (r/min), the measured inlet manifold pressure, P_{im} (N/m^2), and temperature, T_{im} (K), and the geometric values of stroke, s (2 or 4), cylinder displacement volume, V_C (m^3), and the number of cylinders, n_C.

$$\eta_{vol} = \frac{\dot{m}_a}{V_c . n_c \frac{2}{s} . \frac{N}{60} . \frac{P_{im}}{R.T_{im}}} \quad \text{..... D.7}$$

D.9 Compressor pressure ratio

Compressor pressure ratio, r_C (-), is derived from the measured ambient absolute pressure, p_a (N/m^2), and the measured inlet gauge pressure, p_o (N/m^2), and the measured compressor outlet gauge pressure, p_{co} (N/m^2)

$$r_c = \frac{p_a + p_{co}}{p_a + p_o} \quad \text{..... D.8}$$

D.10 Turbine pressure ratio

Turbine pressure ratio, r_t (-), is derived from the measured ambient absolute pressure, p_a (N/m^2), and the measured exhaust manifold gauge pressure, p_{em} (N/m^2), and the measured turbine outlet gauge pressure, p_{to} (N/m^2)

$$r_t = \frac{p_a + p_{em}}{p_a + p_{to}} \quad \text{..... D.9}$$

D.11 Compressor density ratio

The compressor density ratio, r_{ρ_c} (-), is derived from the measured ambient absolute pressure, p_a (N/m^2), and the measured inlet gauge pressure, p_o (N/m^2), the measured compressor outlet gauge pressure, p_{co} (N/m^2), and the measure inlet and outlet temperatures, T_o & T_{co} (K).

$$r_{\rho_c} = \frac{p_a + p_{co}}{T_{co}} . \frac{T_o}{p_a + p_o} \quad \text{..... D.10}$$

D.12 Turbine density ratio

The turbine density ratio, r_{ρ_t} (-), is derived from the measured ambient absolute pressure, p_a (N/m^2), and the measured exhaust manifold gauge pressure, p_{em} (N/m^2), the measured turbine outlet gauge pressure, p_{to} (N/m^2), and the measure inlet and outlet temperatures, T_{em} & T_{to}.(K).

D.12 Turbine density ratio

The turbine density ratio, r_{ρ_t} (-), is derived from the measured ambient absolute pressure, p_a (N/m^2), and the measured exhaust manifold gauge pressure, p_{em} (N/m^2), the measured turbine outlet gauge pressure, p_{to} (N/m^2), and the measure inlet and outlet temperatures, T_{em} & T_{to} (K).

$$r_{\rho_t} = \frac{T_{to}}{p_a + p_{to}} \cdot \frac{p_a + p_{em}}{T_{em}} \quad \text{..... D.11}$$

D.13 Compressor mass flow parameter

The compressor mass flow parameter, mfp_C (kg$\sqrt{\text{K}}$m^2/N), is derived from the calculated air mass flow rate, $\dot{m}_a$ (kg/s), the measured ambient pressure, p_a (N/m^2), the measured inlet gauge pressure, p_0 (N/m^2), and the measured inlet temperature T_0 (K).

$$mfp_c = \dot{m}_a \cdot \frac{\sqrt{T_o}}{p_a + p_o} \quad \text{..... D.12}$$

D.14 Turbine mass flow parameter

The turbine mass flow parameter, mfp_t (kg$\sqrt{\text{K}}$m^2/N), is derived from the calculated air mass flow rate, $\dot{m}_a$ (kg/s), the measured fuel mass flow, $\dot{m}_f$ (kg/s), the measured ambient pressure, p_a (N/m^2), the measured exhaust manifold gauge pressure, p_{em} (N/m^2), and the measured exhaust manifold temperature T_{em} (K).

$$mfp_t = (\dot{m}_a + \dot{m}_f) \cdot \frac{\sqrt{T_{em}}}{p_a + p_{em}} \quad \text{..... D.13}$$

D.15 Compressor efficiency

The compressor efficiency, η_c (-), is derived from the measured inlet temperature, T_0 (K), the measured compressor outlet temperature, T_{co} (K),

the calculated compressor pressure ratio, r_C (-), and the assumed ratio of specific heats for the air,γ_a (-).

$$\mu_c = T_o.\frac{r_c^{\left(\frac{\gamma_a-1}{\gamma_a}\right)}-1}{(T_{co}-T_o)} \quad \text{..... D.14}$$

D.16 Turbine efficiency

The turbine efficiency, η_t (-), is derived from the measured exhaust manifold temperature, T_{em} (K), the measured turbine outlet temperature, T_{to} (K), the calculated turbine pressure ratio, r_t (-), and the assumed ratio of specific heats for the exhaust gas mixture,γ_{ex} (-).

$$\mu_t = \frac{T_{em}-T_{to}}{T_{em}.\left(1-\frac{1}{r_t^{\left(\frac{\gamma_{ex}-1}{\gamma_{ex}}\right)}}\right)} \quad \text{..... D.15}$$

D.17 Compressor power

The compressor power, Q_C (W), is derived from the measured inlet temperature, T_0 (K), the calculated specific heat of the air, Cp_a (J/kgK), (this is calculated as in the processing of the British Gas data in Appendix A), the calculated air mass flow rate, $\dot{m}_a$ (kg/s), the calculated compressor pressure ratio, r_C (-), the calculated compressor efficiency, η_t (-), and the assumed ratio of specific heats for the air,γ_a (-).

$$Q_c = \frac{\dot{m}_a.Cp_a.T_o.\left(r_c^{\left(\frac{\gamma_a-1}{\gamma_a}\right)}-1\right)}{\eta_c} \quad \text{..... D.16}$$

D.18 Turbine power

The turbine power, Q_t (W), is derived from the measured exhaust manifold temperature, T_{em} (K), the calculated turbine efficiency, η_t (-), the calculated specific heat of the exhaust gas mixture, Cp_{ex} (J/kgK), (this is calculated as in the processing of the British Gas data in Appendix A), the calculated mass flow rates of air and fuel, $\dot{m}_a$ & $\dot{m}_f$ (kg/s), the calculated turbine pressure ratio, r_t (-), and the assumed ratio of specific heats for the exhaust gas mixture, γ_{ex} (-).

$$Q_t = (\dot{m}_a + \dot{m}_f).Cp_{ex}.T_{em}.\eta_t.\left(1 - \frac{1}{r_t^{\left(\frac{\gamma_{ex}-1}{\gamma_{ex}}\right)}}\right) \quad \text{..... D.17}$$

D.19 Intercooler pressure ratio

Intercooler pressure ratio, r_{ic} (-), is derived from the measured ambient absolute pressure, p_a (N/m^2), and the measured compressor outlet gauge pressure, p_{co} (N/m^2), and the measured inlet manifold gauge pressure, p_{im} (N/m^2)

$$r_t = \frac{p_a + p_{im}}{p_a + p_{co}} \quad \text{..... D.18}$$

D.20 Intercooler density ratio

Intercooler density ratio, $r_{\rho_{ic}}$ (-), is derived from the measured ambient absolute pressure, p_a (N/m^2), and the measured compressor outlet gauge pressure, p_{co} (N/m^2), the measured inlet manifold gauge pressure, p_{im} (N/m^2), and the measured temperatures into and out of the intercooler, T_{co} & T_{im} (K).

$$r_{\rho_{ic}} = \frac{p_a + p_{im}}{T_{im}} . \frac{T_{co}}{p_a + p_{co}} \quad \text{..... D.19}$$

D.21 Intercooler effectiveness

Intercooler effectiveness, ε_{ic} (-), is derived from the measured temperatures into and out of the intercooler, T_{co} & T_{im} (K), and of the coolant entering the intercooler, $T_{water.in}$ (K)

$$\varepsilon_{ic} = \frac{T_{co} - T_{im}}{T_{co} - T_{water,in}} \quad \text{..... D.20}$$

Appendix E

Mathematical Basis of the Engine Simulation

E.1 Introduction

The mathematical model presented here describes the variation of the thermodynamic state of an engine system during steady-state or transient operation. The computer code created for this investigation makes use of the SPICE II subroutine library[43], which provides infrastructure, such as gas properties, piston motion and numerical integration routines.

The model makes use of the concept of the thermodynamic control volume as shown schematically in Figure E.1. The equations which determine the behaviour of the system are derived from the principles of mass and energy conservation for the thermodynamic control volume and momentum conservation for the crank shaft and turbocharger rotor. The application of each conservation principle leads to a first order ordinary differential equation (ODE). The complete set consists of 31 coupled ODEs. The development of the basic equations for a general thermodynamic control volume may be found in reference [44] The set of ODEs is given below.

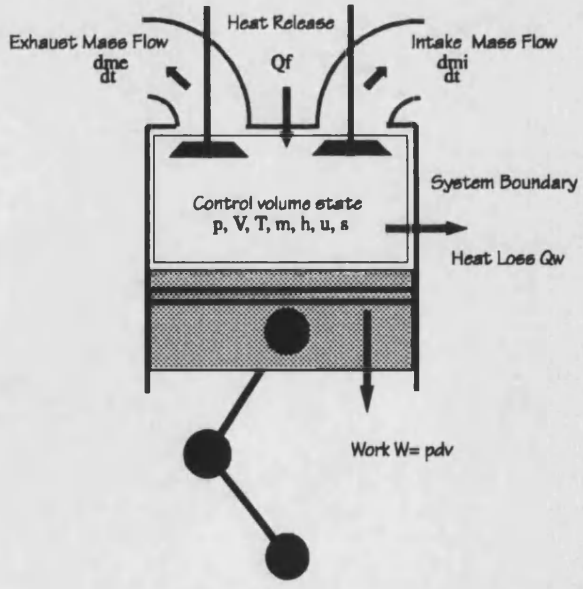


Figure E.1 Schematic model of the SPICE II control volume

E.2 Mass Conservation

In order to satisfy the principle of mass conservation for an individual control volume, and thereby the system as a whole, the sum of the flows entering and leaving should be equal to the rate of change of mass in the control volume. Allowance is made for the additional mass of the fuel at the time of heat release.

$$\frac{dM}{dt} = \sum_i \frac{dm_i}{dt} - \sum_e \frac{dm_e}{dt} + \sum \frac{dm_f}{dt} \quad \text{.....E.1}$$

E.3 Energy Conservation

The principle of conservation of energy is applied through the first law of thermodynamics...

$$\Delta U = \sum Q - \sum W + \left(\sum H_i - \sum H_e\right) \quad \text{.....E.2}$$

Wherein the change in internal energy is a simple function of the net heat transfer at the system boundary (SQ), the work done on the surrounding (SW) and the enthalpy (SH) entering and/or leaving the control volume. When each term is broken down into its component parts the equation for the rate of change of stagnation temperature with time may be obtained.

$$\frac{dT}{dt} = \frac{1}{m\frac{\partial u}{\partial T}} \left[\begin{array}{l} \sum \frac{dQ_w}{dt} + \frac{dQ_f}{dt} + \sum_i h_i \frac{dm_i}{dt} + \sum_e h_e \frac{dm_e}{dt} \\ -m\frac{\partial u d\lambda}{\partial \lambda dt} - \frac{mRT}{V}\frac{dV}{dt} - u\frac{dm}{dt} \end{array} \right] \quad \text{.....E.3}$$

E.4 Mixing of air with the products of combustion

The physical properties of the working fluid are assumed to depend on both temperature and composition. Composition is derived from the ratio of dry air to products of combustion. Atmospheric humidity and high temperature dissociation effects are ignored. Thus the working fluid in any part of the system consists of nitrogen, carbon dioxide, gaseous water and oxygen. The relative proportions of each component can be deduced from the fuel:air ratio

at which the original heat release took place. Hence, in each of the control volumes in the system a "fuel: air ratio" (l) is computed....

$$\frac{d\lambda}{dt} = \frac{(1+\lambda)}{m}\left[\frac{dm_f}{dt} + \sum_i \frac{dm_i}{dt}\frac{(\lambda_i - \lambda)}{(1+\lambda_i)}\right] \quad \text{.....E.4}$$

The mixing process between the contents of a control volume and an incoming flow, or the products of combustion, is assumed to occur instantaneously; the so-called perfect mixing assumption. The gas properties are then computed as a function of temperature and composition from 4th and 5th order polynomials.

E.5 Flow between Control Volumes

Figure E.1 showed the assembly of thermodynamic control volumes in the configuration of a six cylinder turbocharged diesel engine.

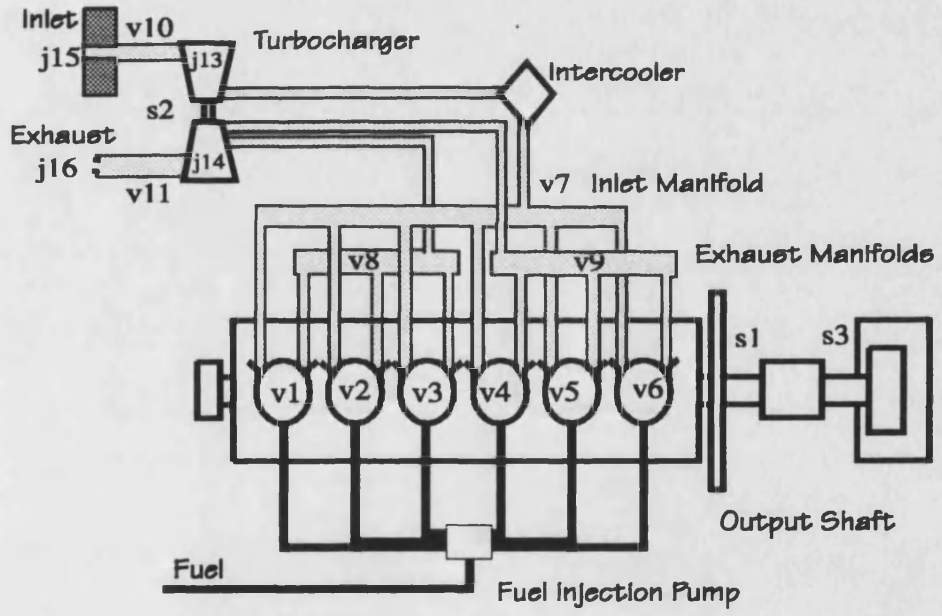


Figure E.1 Schematic view of the SPICE II model of the Leyland TL-11 Diesel engine

It will be seen that control volumes (v1 to v11) are interconnected by flow elements (j1 to j16). Flow elements have properties dependent upon their function in the model. Poppet valves have a schedule of effective area with crank angle, derived from the cam profile and the flow characteristics of the valve-seat-port combination. The instantaneous rate of flow through the valve is determined from the prevailing pressure ratio across the valve using one-

dimensional quasi steady-state compressible flow theory. The kinetic energy developed at the throat is assumed to be dissipated fully in the downstream control volume, that is without pressure recovery. Turbocharger compressors and turbines are represented by performance maps, usually obtained from steady-state rig tests. The performance maps relate rotational speed and pressure ratio to mass flow and isentropic efficiency. The calculation of turbomachine power assumes quasi steady-state flow, in which the work transfer is equal to the enthalpy change.

E.6 Crank Shaft Dynamics, Fuel Pump and Speed Control

The crank shaft model consists of a concentrated moment of inertia with a number of applied torques. The acceleration is given by ...

$$\frac{d^2\varphi}{dt^2} = \frac{\sum \tau}{I} \quad \text{.....E.5}$$

which, when integrated will yield both the instantaneous angular velocity (dj/dt) and the position (j). The engine under test was controlled by a hydromechanical fixed speed governor, with many non-linearities, for example, due to the flow of hydraulic fluid through ports and viscous friction. Consequently, it was decided not to attempt to develop a detailed physical model of the governor. For this study a 3 term - proportional, integral and derivative - transfer function was used to represent the governor, with a first order lag being used to represent the fuel pump rack actuator. The governor equation is as follows ...

$$x_g = k_1\varepsilon(t) + k_2\int_0^{\infty}\varepsilon(t)dt + k_3\frac{d\varepsilon(t)}{dt} \quad \text{.....E.6}$$

The actuator equation is ..

$$x_r = x_g\left(1 - e^{-k_4 t}\right) \quad \text{.....E.7}$$

The combined equation becomes ..

$$x_r = (x_0 - x_g)e^{-k_4 t} + x_g \quad \text{.....E.8}$$

Appendix F

Example input & output data files from the SPICE II engine simulation

F.1 The SPICE II engine definition file for the TL-11 baseline model

```
* File: tl11.base.05.pro  Version: 05  Date: 6/8/92

* control data
2                                   * mode 2
3.0   125.0E-6                      * duration / step size

C nv nj ns nh ng nht nval nshl ngov nfp nss ntci
   11 16  3  1  10   1    2    1    1   1   1    0

* print and plot controls
1 -2  0  -2
1 -2 0 0 0 0 0

* volume data; firing order of TL11 is 1,5,3,6,2,4
0 1 1 1 1 360 4 1 21 11 1 * CYL #1            v1
0 1 1 1 1 600 4 1 21 11 1 * CYL #2            v2
0 1 1 1 1 120 4 1 21 11 1 * CYL #3            v3
0 1 1 1 1 480 4 1 21 11 1 * CYL #4            v4
0 1 1 1 1 240 4 1 21 11 1 * CYL #5            v5
0 1 1 1 1   0 4 1 21 11 1 * CYL #6            v6
1 2 0 0 1   0 0 0  0 0 0 * INLT MFLD  #1      v7
1 3 0 0 1   0 0 0  0 0 0 * EXHST MFLD #1      v8
1 4 0 0 1   0 0 0  0 0 0 * EXHST MFLD #2      v9
1 7 0 0 1   0 0 0  0 0 0 * Inlet pipe         v10
1 9 0 0 1   0 0 0  0 0 0 * Exhaust pipe       v11


* junction data
1 1 0 7  0 0 0  1 1 * INLT VALVE CYL #1         j1
1 2 0 1  0 0 0  8 1 * EXHST VALVE CYL #1        j2
1 1 0 7  0 0 0  2 2 * INLT VALVE CYL #2         j3
1 2 0 2  0 0 0  8 2 * EXHST VALVE CYL #2        j4
1 1 0 7  0 0 0  3 3 * INLT VALVE CYL #3         j5
1 2 0 3  0 0 0  8 3 * EXHST VALVE CYL #3        j6
1 1 0 7  0 0 0  4 4 * INLT VALVE CYL #4         j7
1 2 0 4  0 0 0  9 4 * EXHST VALVE CYL #4        j8
1 1 0 7  0 0 0  5 5 * INLT VALVE CYL #5         j9
1 2 0 5  0 0 0  9 5 * EXHST VALVE CYL #5        j10
1 1 0 7  0 0 0  6 6 * INLT VALVE CYL #6         j11
1 2 0 6  0 0 0  9 6 * EXHST VALVE CYL #6        j12
2 0 0 10  0 0 0  7 2 * turbocharger compressor j13
3 0 2 8  9 0 0  11 2 * turbocharger turbine    j14
0 8 0 0  0 0 0  10 0 * Inlet Orifice           j15
0 10 0 11  0 0 0  0 0 * Exhaust Orifice        j16


* shaft data
2 4.0      0.00                     * crankshaft
2 1.4E-4   1.00                     * turbocharger rotor
2 1.0      1.00                     * output shaft
```

```
* heat release data sets
0   0 125 42.8E6    0    0.7 1 0 0


* geometric data sets
   0.127     0.146 15.75    0.2667 0.0     * cylinder geometry
   0.00488   0.000  0.0     0.00   0.0     * intake manifold volume
   0.0015    0.1   28.0   292.26   0.0     * exhaust manifold volume 8
   0.0011    0.08  28.0   292.26   0.0     * exhaust manifold volume 9
-LOAD_TORQUE 0.0    0.0     0.0    0.0     * speed - torque coefficients
  30.0e-9    0.0    5.0e-9  0.0    0.0     * governor coefficients
   0.0029    0.0    0.0     0.0    0.0     * Inlet pipe
   0.004996  0.0    0.0     0.0    0.0     * BS Orifice
   0.5       0.0    0.0     0.0    0.0     * Exhaust Pipe
   0.008     0.0    0.0     0.0    0.0     * Exhaust Orifice dia (130mm Cd 0.6)

* chamber wall heat transfer properties
5
1000 350 0.025 160 0 0.0135 544 0.300
1000 350 0.012       40 0 0.0123 616       0.000
1000 350 0.012       40 0 0.0175 592       0.045
1000 350 0.012       40 0 0.0175 427       0.090
1000 350 0.012       40 0 0.0175 387       0.136

* poppet valve definitions

17 0 1 1E-6                    * inlet valve (data calculated 29/1/91)
355 370 385 400 415  430  445  460  475  490  505 520 535 550 565 580 595
  0 108 366 628 863 1025 1143 1143 1133 1115 1026 870 645 361 130  15   0

18 0 1 1E-6                  * exhaust valve (ditto)
130 145 160 175 190  205  220  235  250  265  280  295 310 325 340 355 370 390
  0  48 218 504 791 1002 1134 1206 1207 1207 1163 1046 855 599 300 100  19   0

* shaft load definitions  / type / shaft / 'geom' data set / spare / spare / spare
1  3  5  0  0  0                     * load torque on the output shaft

* shaft system definitions
2
1  3                                 * group shafts 1 and 3 in fixed ratio

* fuel pump
2               / number of speeds (horizontal curve)

7               / control volume used for boost control (not used)

1.00e+5  2.0e+5       / boost control pressure range

500 rev/min
0.0020     0.010        0.010     rack(m)
20e-6      120e-6       120e-6    fuel(kg/shot)
698        698          698       timing (22 deg BTDC, manufacturer's data)

3000 rev/min
0.0020     0.010        0.010     rack(m)
20e-6      120e-6       120e-6    fuel(kg/shot)
698        698          698       timing


* speed governor
1  1  1  6  DEMAND_SPEED

* transient control instruction
s
* entry and exit conditions
99E+3 300 99E+3 300
```

F.2 The SPICE II compressor definition file for the TL-11 model

```
* File:comp.dat; Version:2; Date:4/9/91; Compressor Map for Garrett Comp T51
* CONTROL DATA :- NO SPEED LINES, SCALING FACTORS: PR, N, EFF, MF.
7 1 0.0575 0.0092 1E-4
* AFTERCOOLER DATA
2600 290 0.88
* TABULATED COMPRESSOR MAP, PR, MF, EFF.
40900
1.28 0.05 60.0
1.26 0.14 70.0
1.23 0.23 70.0
1.20 0.28 65.0
1.18 0.32 60.0
1.02 0.48 40.0
*
56300
1.54 0.10 60.0
1.54 0.24 73.0
1.52 0.32 75.0
1.48 0.38 73.0
1.40 0.46 65.0
1.10 0.60 40.0
*
67900
1.82 0.19 65.0
1.84 0.35 75.0
1.80 0.48 75.0
1.75 0.53 73.0
1.69 0.57 70.0
1.24 0.72 40.0
*
77900
2.18 0.28 70.0
2.21 0.43 75.0
2.12 0.60 75.0
1.99 0.68 70.0
1.80 0.74 60.0
1.36 0.81 40.0
*
87100
2.50 0.48 73.0
2.60 0.55 75.0
2.54 0.67 75.0
2.35 0.77 70.0
2.08 0.81 60.0
1.44 0.85 40.0
*
95100
3.02 0.59 70.0
2.89 0.77 73.0
2.72 0.82 70.0
2.52 0.84 65.0
2.34 0.85 60.0
1.60 0.85 40.0
*
102700
3.44 0.67 70.0
3.44 0.74 70.0
3.40 0.78 70.0
3.24 0.82 70.0
2.88 0.87 65.0
1.60 0.86 40.0
*
```

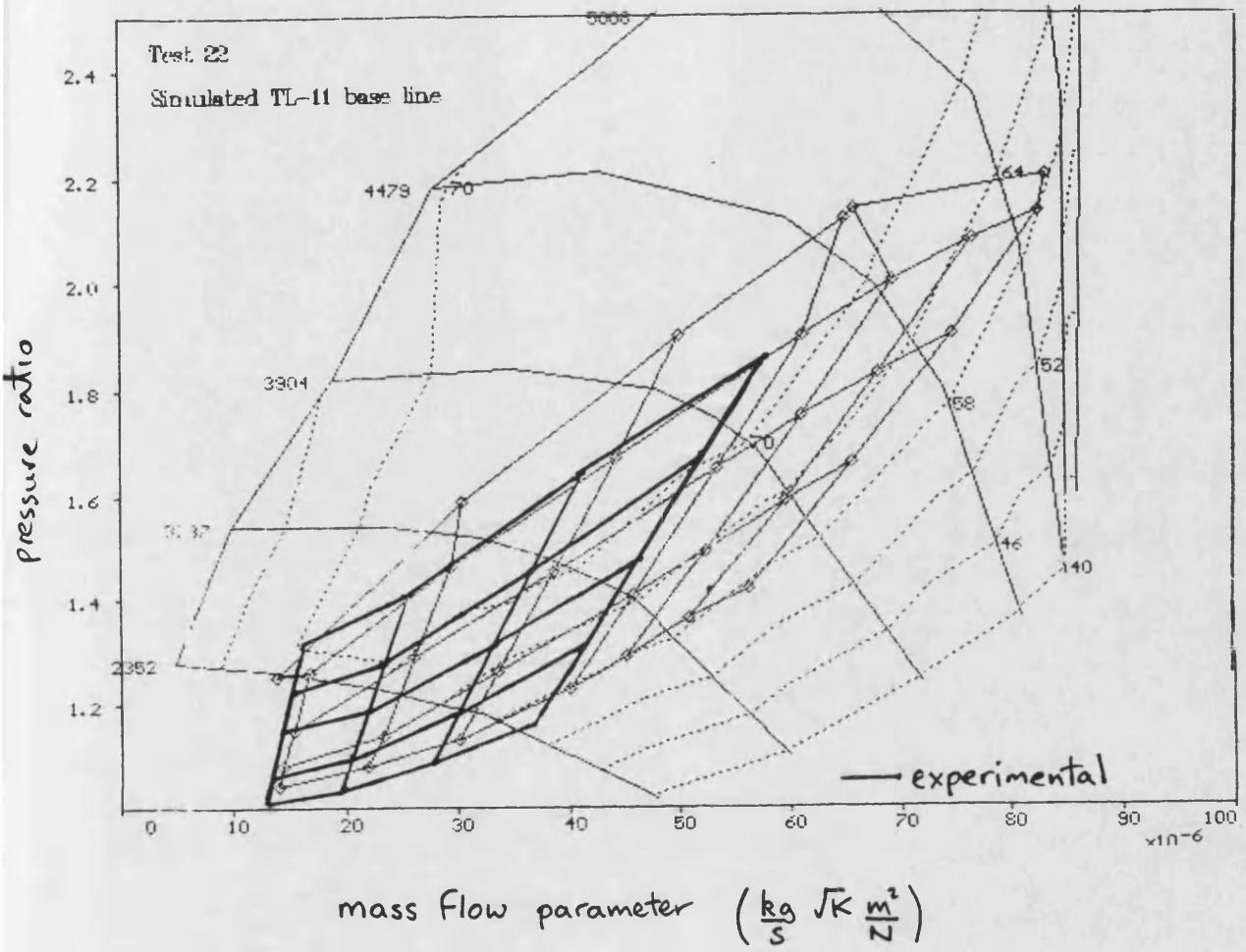


Figure F.1 **Compressor map for the TL-11 turbocharger**

F.3 The SPICE II turbine map definition file for the TL-11 model

```
* File:turb.dat; Version:04; Date: 5/8/92 Turbine map for Garrett Air 1.3469 Turbine
* CONTROL DATA :- NO SPEED LINES, ROTOR DIA., SCALING - PR, N, EFF, MF
8 0.075 1.0 0.0589 0.0096 1.0E-4
* TABULATED TURBINE MAP :- PR, MF, EFF,
22900
1.05 0.200 50.0
1.13 0.295 63.0
1.18 0.350 68.0
1.22 0.380 67.0
1.24 0.405 66.0
1.26 0.425 65.0
*
31200
1.07 0.200 47.0
1.20 0.350 68.0
1.30 0.420 70.0
1.36 0.460 69.5
1.44 0.495 68.5
1.52 0.520 66.0
*
37600
1.09 0.200 43.0
1.20 0.330 64.0
1.40 0.460 70.0
1.50 0.505 70.5
1.64 0.545 70.0
1.74 0.570 68.0
*
43300
1.10 0.200 38.0
1.20 0.310 60.0
1.40 0.400 67.0
1.60 0.515 70.0
1.74 0.550 71.0
2.00 0.590 69.0
*
48400
1.12 0.200 37.0
1.20 0.300 57.0
1.40 0.420 64.0
1.60 0.500 68.0
2.04 0.580 70.0
2.26 0.593 68.0
*
52600
1.13 0.200 36.0
1.40 0.400 62.0
1.60 0.475 66.0
2.00 0.560 68.0
2.36 0.580 68.0
2.54 0.580 68.0
*
56900
1.15 0.200 35.0
1.60 0.440 64.0
2.00 0.530 67.0
2.40 0.560 67.0
2.70 0.565 65.0
2.80 0.565 65.0
*
70000
1.18 0.200 33.0
1.40 0.330 57.0
1.80 0.460 64.0
2.40 0.520 65.0
3.00 0.530 62.0
3.40 0.520 60.0
*
```

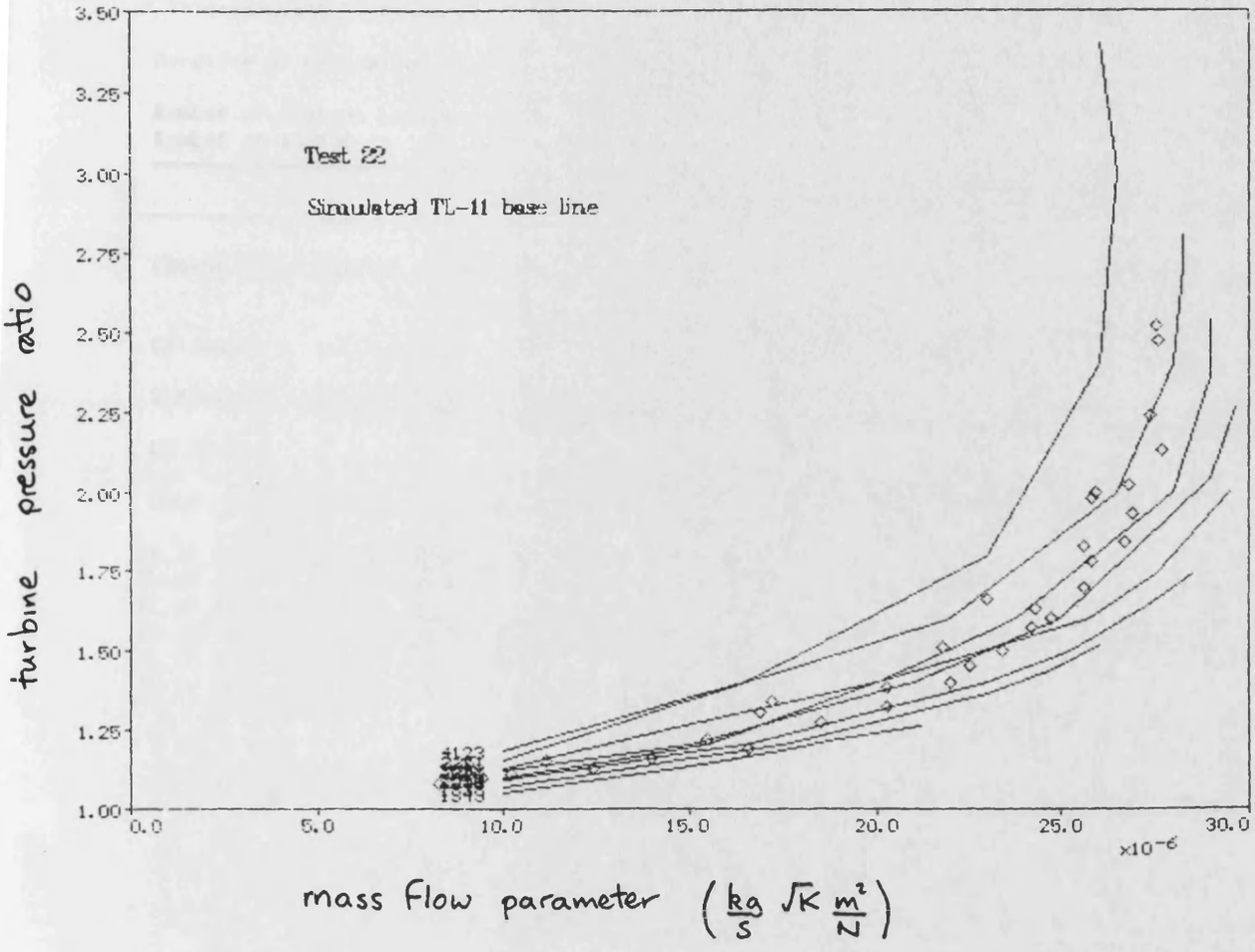


Figure F.2 **Turbine map for the TL-11 turbocharger**

F.4 An example of a SPICE II results file for the TL-11 model

```
*******************************************************************************
**** SPICE II    ENGINE SYSTEM SIMULATOR ************** UNIVERSITY OF BATH ****
**** Version 1.10      MAY 1990 ************************************************
*******************************************************************************


 Title :
* File: tl11.base.05.2200.883.eng; Speed: 2200 rpm; Load: 883 Nm; Mark Scaife 19
* File:comp.dat; Version:2; Date:4/9/91; Compressor Map for Garrett Comp T51
* File:turb.dat; Version:04; Date: 5/8/92 Turbine map for Garrett Air 1.3469 Tur

 Duration of simulation (s)=   3.000000  Integration time step  (s)=   0.000125

 Number of control volumes =   11        Number of flow junctions  =   16
 Number of shafts          =    3
 ------------------------------------------------------------------------------


 END-OF-CYCLE SUMMARY OF RESULTS {CYCLE 53}
 ------------------------------

 Cylinders :  1 2 3 4 5 6

 SIMULATION TIME   2.970625 [SEC]     ELAPSED TIME   0.00 [MIN]

 CYLINDERS
 ---------
 IMEP  BMEP FMEP PMEP  ISFC BSFC  EFFICIENCY  POWER [kW]  VOL.EFFICIENCY SCAV.
       [BAR]           [g/kWh]   b.Th  MECH  IND.  BRAKE  TRAPD O/ALL  RATIO
11.76 10.0 1.75-0.96 191.9 225.5 37.30 85.11  39.7  33.8  85.9  86.2   1.0037
11.80 10.0 1.75-0.96 191.8 225.2 37.35 85.16  39.9  34.0  86.0  86.3   1.0036
11.75 10.0 1.75-0.95 192.3 226.0 37.22 85.10  39.7  33.8  85.9  86.2   1.0035
11.69  9.9 1.75-0.99 192.4 226.3 37.16 85.02  39.5  33.6  86.0  85.9   0.9992
11.78 10.0 1.75-0.98 192.2 225.8 37.25 85.14  39.8  33.9  86.0  86.4   1.0038
11.73 10.0 1.75-0.99 188.3 221.4 37.99 85.07  39.6  33.7  86.0  86.3   1.0037

 CYLINDERS
 ---------
 CYL. Pmax & ANG AFtr  Tmax Tmean hmean    Qw1   Qw2   QW3    QW4   QW5   Qinj
      [BAR]            [K]  [K] [W/m2K]          [J/cycle]
  1   130.   7.8  31.1 1747.1046.1 631.5   172.  215.  162.    0.    0.   4962.
  2   130.   7.0  31.1 1750.1048.0 631.8   173.  216.  164.    0.    0.   4976.
  3   130.   7.1  31.2 1748.1046.2 631.4   171.  215.  163.    0.    0.   4970.
  4   130.   5.8  31.4 1743.1043.1 631.1   171.  213.  161.    0.    0.   4944.
  5   130.   6.6  31.2 1749.1047.9 632.4   173.  216.  164.    0.    0.   4978.
  6   130.   6.5  31.2 1748.1043.9 628.6   173.  212.  160.    0.    0.   4855.

 COMBUSTION SUMMARY
 ------------------
 VOL. PmID  TmID   ID    ID   Lpipe CAstat  CAinj CAcomb  BETA  MODEL
     [bar]  [K]  [ms]  [deg]  [m]   [deg]  [deg]  [deg]        TYPE
  1   62.0  852.  0.61   8.08  0.70  698.0  704.5  712.6 0.2092  21
  2   62.0  852.  0.61   8.08  0.70  698.0  704.5  712.6 0.2111  21
  3   62.0  852.  0.61   8.08  0.70  698.0  704.5  712.6 0.2092  21
  4   62.0  852.  0.61   8.07  0.70  698.0  704.5  712.6 0.2096  21
  5   62.0  851.  0.61   8.08  0.70  698.0  704.5  712.6 0.2104  21
  6   62.1  852.  0.61   8.07  0.70  698.0  704.5  712.6 0.2117  21
```

HEAT TRANSFER SUMMARY New surface temperatures [K]

CYL.	PISTON	CYL.HD	LINER			ADDITIONAL AREAS				
			UPPER	MID	LOWER					
1	585.4	663.8	629.9	444.7	397.9	0.0	0.0	0.0	0.0	0.0
2	586.2	664.8	630.6	444.8	397.8	0.0	0.0	0.0	0.0	0.0
3	585.5	663.8	629.9	444.1	397.5	0.0	0.0	0.0	0.0	0.0
4	584.3	662.4	628.8	444.5	397.4	0.0	0.0	0.0	0.0	0.0
5	586.3	664.9	630.6	445.1	398.0	0.0	0.0	0.0	0.0	0.0
6	584.0	662.0	629.9	444.9	397.8	0.0	0.0	0.0	0.0	0.0

ADDITIONAL SYSTEM DATA

CYL	SHAFT	SPEED	VALVE FLOW				INLET...MANIFOLDS...EXHAUST					
		[rev/min]	INLET	[kg/cycle]	EXHAUST		NO.	P[bar]	T[K]	NO.	P[bar]	T[K]
1	1	2199.88	1	0.00362	2	0.00373	7	2.00	306.27	8	2.13	813.19
2	1	2199.88	3	0.00362	4	0.00373	7	2.00	306.27	8	2.13	813.19
3	1	2199.88	5	0.00362	6	0.00373	7	2.00	306.27	8	2.13	813.19
4	1	2199.88	7	0.00361	8	0.00374	7	2.00	306.27	9	2.15	808.85
5	1	2199.88	9	0.00363	10	0.00373	7	2.00	306.27	9	2.15	808.85
6	1	2199.88	11	0.00362	12	0.00374	7	2.00	306.27	9	2.15	808.85

CONSTANT VOLUMES - Mean Values

VOL	Pmn	Tmn	Tmn	AFRm	Qin	Qconv	Qrad	Vol.	Asurf
	[bar]	[K]	[C]			[J/cycle]		[litres]	[cm2]
7	2.004	306.	33.14	803.44	0.00	0.00	0.00	4.88	0.00
8	2.129	813.	540.	31.21	0.00	4272.71	0.00	1.50	1000.00
9	2.151	809.	536.	31.29	0.00	3377.07	0.00	1.10	800.00
10	0.961	300.	27.	0.00	0.00	0.00	0.00	2.90	0.00
11	1.018	733.	460.	31.16	0.00	0.00	0.00	500.00	0.00

JUNCTIONS - Mean Values

JUN.	FLOW	MN.MACH	MN.VEL.	MN.POWER	EFFIC	P-RATIO	MFParam	U/C	JUNC.TYPE
	[kg/s]		[m/s]	[kW]					
1	0.06643	0.2745	88.1						VAR.AREA
2	0.06850	0.5951	321.4						VAR.AREA
3	0.06644	0.2689	85.7						VAR.AREA
4	0.06851	0.5965	322.3						VAR.AREA
5	0.06641	0.2750	88.2						VAR.AREA
6	0.06851	0.5965	322.0						VAR.AREA
7	0.06618	0.2757	88.5						VAR.AREA
8	0.06858	0.6064	326.0						VAR.AREA
9	0.06653	0.2721	87.1						VAR.AREA
10	0.06842	0.6083	327.5						VAR.AREA
11	0.06650	0.2779	89.5						VAR.AREA
12	0.06863	0.6073	326.7						VAR.AREA
13	0.39848			45.87	0.6376	2.112	0.0000718		COMP
14	0.41120			45.88	0.6458	2.222	0.0000536	0.5262	TURB
15	0.39848	0.2050	71.2						ORIFICE
16	0.41156	0.2058	108.9						ORIFICE

SHAFTS - Mean Values

SHAFT	TORQUE VALUES [Nm]			POWER TRANSMITTED [kW]			MN.SPEED	MN.ACCEL
NO.	NET	INPUT	OUTPUT	NET	INPUT	OUTPUT	[rev/min]	[rev/min/s]
1	884.1	1947.8	1063.7	203.64	448.73	245.10	2199.9	0.2051E+01
2	0.0	5.4	5.4	0.01	45.88	45.87	81074.4	0.1045E+03
3	-883.0	0.0	883.0	-203.42	0.00	203.42	2199.9	0.2051E+01

SYSTEM MASS BALANCE

AIR ENTERING SYSTEM	= 0.02171732
FUEL ENTERING SYSTEM	= 0.00069359
EXHAUST LEAVING SYSTEM	= 0.02242994
MASS BALANCE	= 0.9992

SYSTEM ENERGY BALANCE

	[J]	[%]
ENERGY OF FUEL	29685.7	100.00
ENERGY TO EXHAUST	10836.7	36.50
ENERGY TO COOLANT	3294.0	11.10
USEFUL WORK	11098.8	37.39
FRICTION WORK	1942.9	6.54
INTERCOOLERS	2395.2	8.07

```
                                    SHAFT LOSSES              0.0       0.00
                                    ENERGY BALANCE    -       1.0040

INITIAL CONDITIONS - FOR RESTART
--------------------
* Initial Fuel Pump Rack Positions
  9.8389797E-03
* Initial Shaft Speeds
   2193.357
   81291.13
   2193.357
* Initial Pressure / Temperature / Fuel-Air Ratio
   234483.3        697.0046       3.2032412E-02
   262765.8        387.1266       1.1503361E-03
   727517.4        1057.534       3.2081291E-02
   174690.6        349.3085       1.5067367E-03
   277420.2        809.6897       3.2084908E-02
  1.2203441E+07    1398.634       1.2358830E-02
   200610.6        306.3836       6.4474443E-05
   166414.1        727.1592       3.2036260E-02
   261241.0        874.4745       3.2053337E-02
   96100.66        299.9735       0.0000000E+00
   101911.4        733.6933       3.2083690E-02
```

Appendix G

Calculation of valve effective area for SPICE II model of the TL-11 Diesel engine

G.1 Introduction

It is demonstrated in the text that it is very important to get accurate values for the schedule of effective areas for input to the SPICE II simulator.

This section shows the calculations and assumptions used to calculate these values for the inlet and exhaust valves of the TL-11 model.

G.2 Calculation of effective area of a poppet valve as it opens

For a poppet valve with a 30° face angle, as shown in figure Figure G.1 below, the cross-sectional flow area is given as:

$$A_v = \pi d_1^2 \left(0.866 + 0.375 \frac{L}{d_1} \right) \frac{L}{d_1} \quad \text{..... G.1}$$

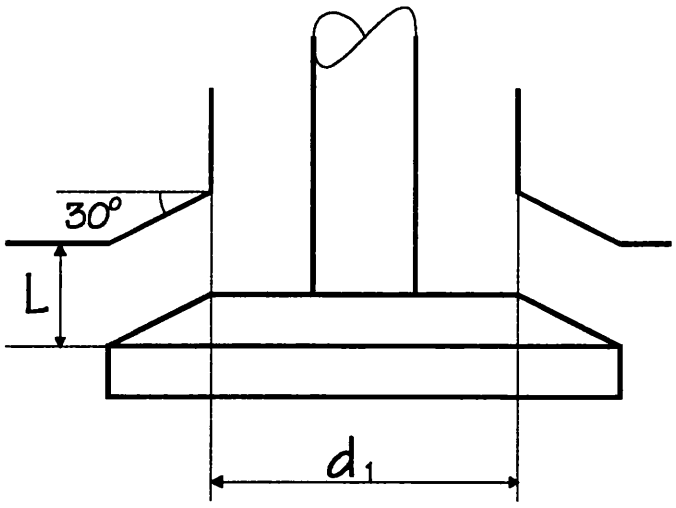


Figure G.1 Poppet valve geometry

For the TL-11 the values of d_1 are:

Inlet valve $d_1 = 49.64$ mm

Exhaust valve $d_1 = 42.02$

The effective flow area is given as:

$$A_e = A_v C_D \quad \text{..... G.2}$$

where C_D is assumed to vary linearly between the following two values

Valve open $C_D = 0.4$

Valve closed $C_D = 0.8$

The valve lift for the TL-11 were measured on the real engine and the effective areas were calculated. These are sown in figure G.2.

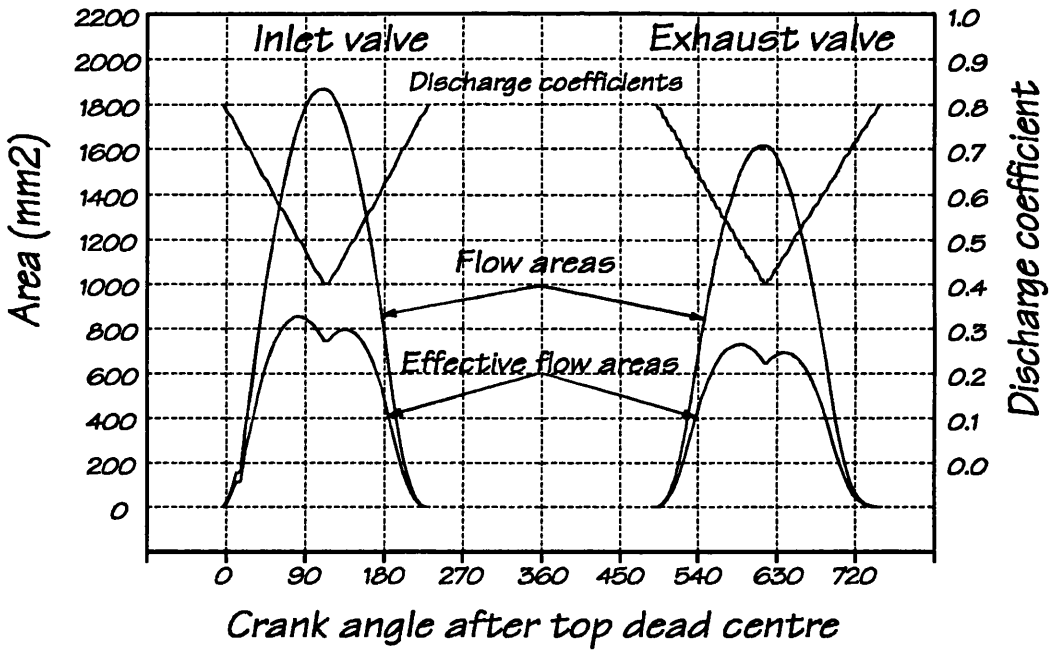


Figure G.2 Calculated inlet and exhaust valve effective areas

References

1 *Nolden C.;* **Predictive Maintenance (1987);** Plant Engineering - File 5543

2 *Fox T.B.;* **Operating Experience with a Vehicle Fault Diagnosis System (1985);** Proc IMechE C44/85

3 *Anon;* **Diagnostics: How Expert Systems can Pinpoint Faults (1988);** MER August 1988

4 *Neale M.J.;* **Experience with condition monitoring in other industries (1985);** Proc IMechE C36/85

5 *Haddad S.D.;* **Optimum maintenance approach using condition monitoring and fault diagnosis (1987);** Noise and Vibration Control Worldwide, June 1987.

6 *Mauer G.F. & Watts R.J.;* **On-line cylinder diagnostics on Combustion Engines by Noncontact Torque and speed measurements (1989);** IMechE paper 890485

7 *Anon;* **Lube Oil Analysis Gets to the Heart of the Machine (1988);** MER August 1988.

8 *Munakata T.;* **Practical AI: Where it's been and where it is now (1993);** IEEE expert

9 *Lind M.;* **Diagnosis Using Multi-Level Flow Models (1988);** Institute of Automated Control Systems

10 *Richards R.A.;* **Expert Systems for Condition Monitoring of Diesel Engines (1988);** Proc IMarE

11 *Poppe R.H.;* **Expert Condiion Monitoring of Diesel Engines (1992);** Diesel and Gas Turbine Worldwide

12 *Poppe R.H. & Rigby D.J.;* **Personal correspondence (1993)**

13 *Sales literature;* **Asea Brown Boveri (1992),** Harness Road Altens Aberdeen AB9 1RZ

14 *Davies A.J.;* **Diesel Engine and Performance Monitoring 1991;** Naval Architect Journal

15 *Katsoulakos P.S. Newland J., Stansfield J.T. & Ruxton T.;* **Monitoring, Databases and Expert Systems in the Development of Engine Fault Diagnostics (1988);** British Journal of NDT

16 *Perryman R, Reynolds M, Strudwick S;* **Remote condition monitoring and control of standby Diesel generators (1992).** Measurement and Control Vol 25 July/August 1992

17 *Frantzen A,* **Private corresponence (1992)**

18 *Lindemann. U.;* **Comprehensive Diagnostic System for Ships Propulsion and Power Machinery. (1992);** Diesel and Gas Turbine Worldwide

19 *Anon;* **Diagnostics: How Expert Systems can Pinpoint Faults (1988);** MER August 1988

20 *Kinclaid C.L.;* **Computerised Diagnostics for Diesel Engines (1985);** Proc ImechE C47/85

21 *Hebb;* **Organisation of Behaviour (1949)**

22 *Freeman J.A.,& Skapura D.M;* **Neural networks : Algorithms, Applications and Programming Techniques *(1991);*** Addison-Wesley Publishing Company.

23 *Minsky M. & Papert S.;* **Perceptrons (1969);** MIT press

24 *McClelland J. & Rumelhart D.;* **Parallel Distributed Processing Vols 1 and 2 (1986);** MIT press

25 *Hall C;* **Neural net technology: Ready for Prime Time? (1992);** IEEE Expert December 1992

26 *Kraft T. et al;* **Hybrid Neural Net and Rule Based System for Boiler Monitoring and Diagnosis (1991);** Proc American Power Conference

27 *SAIC Intelligent Systems Capabilities Statement (1992);* **SAIC,**

28 *Wu H.J. Liou C.S. Pi H.H.;* **Fault Diagnosis of Processing Damage in Injection Moulding via Neural Network Approach (1991);** Proc Artificial Neural Networks in Engineering Conference ASME Press.

29 *Aylward S., Sheckler L., Anderson R., Urnes J.;* **Application of neural networks to In-Flight Diagnostics (1991);** Proc Artificial Neural Networks in Engineering Conference ASME Press.

30 *Guillot M. & Ouafi A.E.;* **On-line Identification of tool breakage in Metal Cutting Processes by use of Neural Networks. (1991);** Proc Artificial Neural Networks in Engineering Conference ASME Press.

31 *Wasserman P.D., Unal A., & Haddad S.;* **Neural network On-line Machine Condition Monitoring. (1991);** Proc Artificial Neural Networks in Engineering Conference ASME Press.

32 *Lin C.S. & Wu I.C.;* **Neural Networks for Sensor Failure Detection and Data Recovery. (1991);** Proc Artificial Neural Networks in Engineering Conference ASME Press.

33 *Bacon A, Shayler P.J. & MA T.;* **Potential for Engine Control using Neural networks (1992);** Proc IMechE C448/057

34 *MacLoid A.;* **Analogue IC Makes Perfect Neural Net; (1993);** New Electronics

35 *Farhat N. H.;* **Photonic Neural Networks and Learning Machines (1992);** IEEE Expert

36 *Wood J.R., Bethell J.J.;* **The Development of Automotive Diagnostic Systems for Armoured Fighting Vehicles in the British Army (1985);** Proc ImechE C41/85

37 *Neale M.J.;* **Experience with Condition Monitoring in other Industries (1985);** Proc IMechE C35/85

38 *Rumelhart D.E., McClelland J.L. et al;* **Parallel Distributed Programming;** MIT Press

39 *Scaife M.W.; An Experimental Facility for the Development of Intelligent Engine Diagnostics* 1990. MPhil thesis, University of Bath

40 *Mobley C., Scaife M.W., Prest P. & Charlton S.;* **Advanced data acquisition techniques for multi-tasking personal computers (1993);** IMechE Seminar PC's in Engineering

41 *Neural Computer Sciences;* **User Guide,** NCS, Totton, Southampton

42 *Stoeker W.F.;* **Design of Thermal systems (1989);** McGraw Hill

43 *Charlton S.J;* **SPICE II User manual (1990);** University of Bath

44 *Zinner C.;* **Supercharging the Internal Combustion Engine (1978);** Springer-Verlag