Liverpool John Moores University

(School of Engineering)

Ph.D. Thesis

INTELLIGENT FAULT DIAGNOSIS FOR AUTOMOTIVE ENGINES AND REAL DATA EVALUATION

Mahavir Singh Sangha

(Director of Studies: Prof. Dingli Yu)

May 2008

DAMAGED TEXT IN ORIGINAL

ABSTRACT

Fault detection, isolation (FDI) and accommodation is one of the most demanding and developing areas in the automobile industry nowadays. Major car firms are looking for neural network (NN)-based solutions for FDI and accommodation. A new on-line FDI system is proposed in this research for automotive engine air path, which uses adaptive neural network as fault classifier. Radial basis function (RBF) neural networks are adapted for better classification capability and robustness. The developed FDI system is adapted on-line and therefore is capable to cope up with system uncertainties, environmental changes, engine parameter changes, disturbances and engine to engine changes.

Two component and two sensor faults have been investigated as four typical and practical examples of internal combustion (IC) engine faults. Two component faults are exhaust gas recycle (EGR) valve clogged and air leakage in the intake manifold. Two sensor faults are intake manifold pressure and temperature sensor faults. All the faults are considered at four different levels of intensity. A well-known benchmark, the mean value engine model (MVEM), is used for fault-simulation. The adaptive FDI method is compared with a non-adaptive method. The adaptive method out performs the non-adaptive method on simulated data from MVEM. The adaptive method is able to detect and isolate different types of the faults as well as the intensities of the fault. Further, the adaptive method is evaluated for robustness under closed-loop control. The adaptive method is found robust against throttle angle change, load change, engine parameter change and all the changes happening simultaneously. The adaptive method is novel and it is believed to be a contribution to knowledge in this field of study.

A one-litre four-stroke Volkswagen petrol car engine test-bed is used for further experimentation. Real data is acquired over a wide range of engine operations from the engine test-bed. Five different sensor bias faults are superimposed on the fault-free real engine data and the adaptive FDI system is evaluated against these sensor faults including throttle angle, torque, crankshaft speed, intake manifold pressure and temperature sensors. The fault diagnosis results confirm that the sensor faults as small as 2% are clearly detected and isolated for different data sets. Unknown fault detection is also investigated for three different faults using novelty detection method. Lastly, satisfactory sensor fault accommodation is achieved for three different sensors using predictive neural networks.

ACKNOWLEDGEMENT

First of all, I would like to thank the School of Engineering for awarding bursary and 'Universities UK' for awarding ORSAS. This prestigious international award helped the research project financially.

I would like to express my sincerest gratitude to my Director of Studies, Professor Dingli Yu for his professional guidance, support and encouragement throughout the course of my research studies and the presentation of this thesis. highly appreciate his availability and quality time given to me during past three years without which I could not have finished my research up to an excellent standard. I would also like to thank my second supervisor Dr. Gomm for his guidance and Alan Carrier for helping in real data acquisition from engine test bed.

On a more personal note, I would also like to extend my heartiest thanks to my wife and my sons for their tolerance, co-operation and support during the course of my studies. Finally, I would like to thank my parents for their encouragement and the almighty for His grace.

Mahavir Singh Sangha

March 31st, 2008.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

A. SYMBOLS

c	Centre vector
g	Cost function gradient vector
H_u	Fuel lower heating value (kJ/kg)
I	Scaled moment of inertia (kg m^2)
I_{ac}	Actual moment of inertia
J	Cost function
K	Proportional gain
k	Index of the best-matching centre
L_{th}	Stoichiometeric air fuel ratio (14.67)
$\dot{m}_{ap}$	Air mass flow into intake port (kg/sec)
$\dot{m}_{at}$	Air mass flow past throttle plate (kg/sec)
$\dot{m}_{EGR}$	EGR mass flow (kg/sec)
$\dot{m}_f$	Engine port fuel mass flow (kg/sec)
$\dot{m}_{ff}$	Fuel film mass flow (kg/sec)
$\dot{m}_{fi}$	Injected fuel mass flow (kg/sec)
$\dot{m}_{fv}$	Fuel vapor mass flow (kg/sec)
N_1	Minimum prediction horizon
N_2	Maximum prediction horizon
N_u	Control horizon
n	Engine speed (krpm)
n_{cyl}	Number of cylinder
n_{ui}	Maximum lag in the inputs of polynomial model
n_y	Maximum lag in the outputs of polynomial model
p_a	Ambient pressure (bar)

P_b	Load power (kW)
p_d	Differential pressure
P_f	Friction power (kW)
p_i	Manifold pressure (bar)
P_p	Pumping power (kW)
R	Gas constant (here 287×10^{-5})
sp	Set-point
t_d	Time delay
T_a	Ambient temperature (degrees Kelvin)
T_D	Differential time
T_e	Engine temperature
T_{EGR}	EGR temperature (degrees Kelvin)
T_I	Integral time
T_i	Intake manifold temperature (degrees Kelvin)
u	Throttle angle (degrees)
u_c	Control variable/vector
u_n	Nominal control
u^l	Lower bound of control variables
u^u	Upper bound of control variables
V	Lyapunov function
V_d	Engine displacement
V_i	Manifold + port passage volume (m^3)
W_H	Hessian Matrix of the Lagrangian function
W	Matrix of weights
w	Output layer weight vector
X	Input matrix
X_f	Proportion of injected fuel deposited on intake manifold
x	System input vector
y	System output vector
$\hat{y}$	Model output vector
z_s	Weighted sum of all inputs
α_c	Centre learning rate

α_k	Step-length of the *kth* iteration
η_i	Volumetric efficiency
λ_c	Control weighting factor
λ_e	Eigenvalue of a characteristic equation
λ	Air fuel ratio
$\Delta\tau_d$	Injection torque delay time (sec)
κ	Ratio of the specific heats=1.4 for air
τ_f	Fuel film evaporation time constant (sec)
ϕ	Activation function vector
$\phi_\mu(x)$	Merit function of line search
Φ	Matrix of activation function outputs
σ	Width vector

B. ABBREVIATIONS

AFR	Air fuel ratio
BLS	Batch least squares
CO	Carbon monoxide
CO_2	Carbon dioxide
DBW	Drive-by-wire
DI	Direct injection
DPS	Differential pressure
ECU	Electronic control unit
EFI	Electronic fuel injection
EGO	Exhaust gas oxygen
EGR	Exhaust gas recirculation
EVC	Exhaust valve closing
EVO	Exhaust valve opening
GDI	Gasoline direct injection
GMI	Gasoline manifold injection
HC	Hydrocarbon
IA	Ignition angle
IC	Internal combustion
LS	Least squares
MAE	Mean absolute error
MIMO	Multiple input and multiple output
MLP	Multi-layer perception
MVEM	Mean value engine model
NO_x	Nitrogen oxide
RAS	Random amplitude signals
RBF	Radial basis function
RLS	Recursive least squares
RNN	Recursive neural network
SI	Spark ignition
SISO	Single input and single output
T	Torque
TWC	Three-way catalyst
UEGO	Universal exhaust gas oxygen
VL	Valve lift

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The greenhouse gases and air pollution are the most discussed topics among ecologist for last several decades. The vehicle exhaust contains nitrogen compounds that get into the atmosphere and form acids, although the natural sources are much more limited but the biggest non-natural source is burning fossil fuels, especially gasoline. Acid rain forms when molecules of oxidized sulphur and/or nitrogen in the atmosphere combine with water, forming acidic compounds that dissolve in the water that becomes rain. The acid rains in California in early sixties opened the eyes of ecologist and life scientists and the first wave to save environment started.

Old cars and other vehicles on the road emit a lot of harmful gases in the environment. Some times these harmful emissions many fold with faults in the vehicle engine. This significant increase in emissions causes severe air pollution and damage the ozone layer which results in global warming and climate change. Therefore fault detection; isolation and accommodation have become one of the most important aspects of automobile design. There are strict laws to be abided by all the automobile manufacturers. Continuous efforts are being made to improve the design of the Electronic Control Unit (ECU) in an automotive engine in order to ensure early detection and isolation of engine faults that can be hazardous for human life or adversely affect the fuel efficiency of the engine or lead to increase in air pollution.

A fault can be defined as an undesired deviation of one or more variables of the system from the normal/healthy behaviour. Determination of the presence of a fault in the system is called fault detection. Identification of occurring fault from a number of possible faults is called fault isolation. Some times it may be possible to auto-correct the fault and this is known as fault accommodation. Fault accommodation may not be possible for component faults but is possible for sensor faults. In this research neural networks are investigated for fault diagnosis in the air path of an automotive engine.

1.2 WHY THIS RESEARCH

Automobiles manufactured since the early 1980s are equipped with smart systems, known as on-board diagnostics. On-board diagnostics or OBD systems were developed to help repair technicians identify problems associated with the computerized engine systems of modern vehicles. These systems are made up of various sensors and a computer that communicates its findings to a technician by means of diagnostic trouble codes that are stored in the automobile's computer called on-board system. Legislation for on-board diagnostic capability in all vehicles was first enforced from 1988 in United States of America. It was known as OBD – I and it only monitored the oxygen sensor, EGR system, fuel delivery system and engine control module.

The limitation of OBD1 was that it couldn't detect certain kinds of problems such as a non-functioning or missing catalytic converter, ignition misfires, or evaporative emission problems. In addition, the MIL would only illuminate after a failure had occurred, it had no way of monitoring progressive deterioration of emissions-related components.

According to the federal Clean Air Act Amendments of 1990, the California Air Resource Board (CARB) and Environment Protection Agency (EPA) required all 1996 and newer light duty vehicles to have more advanced OBD systems, known as OBD II (Tan and Saif, 2000) in America. The OBD II system monitors vehicle conditions and components that are related to vehicle emissions, such as the catalyst in the catalytic converter, engine misfire, the engine coolant temperature, and the oxygen sensors. Federal law requires

some states that do not meet federal air quality standards to implement an OBD II inspection program to ensure that emissions control equipment is operating correctly. The Environmental Protection Agency (EPA), as well as the California Air Resource Board (CARB) mandated "On Board Diagnosis II" (OBD-II) for all light duty vehicles. All diesel cars sold in the EU must have an OBD system from year 2003 (Nyberg and Stutte, 2004).

Cars and trucks are responsible for approximately half of the air pollution that causes smog, acid rain, and climate change (Air resources environmental fact sheet, 2005). While it is true that modern cars emit less air pollution than older vehicles due to new engine management technology and emission control equipment, they are only cleaner if all emission control systems are operating properly. The OBD II system can often detect a vehicle malfunction before the driver becomes aware of the problem. Early detection and repair of malfunctions will result in less emission. In addition, early repair of minor problems may prevent more significant and more expensive engine problems that could develop if left undetected. For example, a poorly performing spark plug can cause the engine to misfire, a condition sometimes unnoticed by the driver, but one that will be detected by the OBD II system. This engine misfire can, in turn, quickly degrade the performance of the catalytic converter and permanently damage the catalyst. By responding to the check engine light (turned on by the OBD II system) in a timely manner, the driver would be faced with a relatively inexpensive spark plug repair. However, without OBD II detection, the driver could be faced with an expensive catalytic converter repair in addition to the spark plug repair. By storing the malfunction information in the computer's memory at the time it occurs, OBD II allows the service technician to more accurately identify the problem and make the proper repairs. This saves time for the repair technician, money for the consumer, and reduces air pollution.

The OBD II system monitors a variety of engine conditions and outputs while the car is being driven. When the OBD II system detects a problem with the emission control system, a dashboard light is illuminated indicating "Check Engine" or "Service Engine Soon." A corresponding diagnostic trouble code is stored in the computer's memory documenting which emissions control component is experiencing the problem, and under what conditions. The repair technician will retrieve the diagnostic trouble code information from the

computer using a computer scan tool. By using this information, a properly trained technician can more accurately find and fix the problem. If the malfunction indicator light illuminates with a steady, continuous light, the vehicle operator should contact a repair technician and schedule a service visit. This is not an emergency situation, but the vehicle should be serviced soon. However, if the malfunction indicator light blinks or flashes, this indicates certain severe engine malfunctions. When this occurs, the vehicle operator should stop the car immediately and refer to the owner's manual to determine if the car can be driven or if it should be towed to a service station. Continued operation of the vehicle could result in damage to the engine or emissions control components, specifically the catalytic converter, a very costly component. Sometimes the malfunction indicator light goes out by itself. This indicates that the problem that initially triggered the light no longer exists. This could happen if, for example, the gas cap was not on tight, but was then fixed. In this case, the light should reset itself and go out after several trips, eliminating the need for a service visit.

OBD-II has continuous monitoring and fault detection capability for all vehicle components whose failures can result in emission levels beyond 1.5 times of the Federal Test Procedure (FTP) standards. This means that the OBD-II system should trigger and detect the deterioration of any component that results in a FTP exhaust emission increase of 0.075 g/mi of HC, 1.7 g/mi of CO, and 0.2 g/mi of NOx (Tan and Saif, 2000). Once a failure has been detected, a malfunction indicator light (MIL) is to be illuminated to inform the driver of a problem, and a diagnostic trouble code (DTC) is to be stored in the on board memory. The OBD II system may tell the technician that certain components are not ready for testing and the vehicle owner may be asked to drive the vehicle under a variety of conditions for a few days to complete the drive cycle, and then return for testing. The drive cycle is different for each vehicle, but generally can be completed by operating the vehicle in a combination of in-town and highway driving. Essentially a drive cycle puts a vehicle through enough different situations to allow the OBD II system to adequately evaluate the various components. A vehicle may not be ready for a number of reasons, including a recently disconnected battery, or recent clearing of diagnostic codes using an OBD II scan tool.

Short coming of OBD II is that even in case of an increased emission problem, there is just an indicator which driver has a tendency to over ook. This can lead to environmental damage due to increased air pollution. To overcome this problem, OBD III is discussed as a program to minimize the delay between the detection of an emission malfunction by the OBD-II system and the actual repair of the vehicle. This can be achieved by transmitting wireless signal (DTC code) from vehicle ECU to the roadside reader and then further to the nearest dealer. There are legal issues of 'unsuspicious mass surveillance of private vehicle property' to be sorted out before OBD II is mandated by EPA.

All vehicles sold in the UK after December 31st 2000 are required by legislation to allow the European On-Board Diagnostic (EOBD) protocol (Official Journal of the European Communities, 1998). EOBD is the European equivalent of the American OBD II standard. These days OBD-III is being discussed as a program to minimize the delay between the detection of an emission malfunction by the OBD-II system and the actual repair of the vehicle. EOBD2 stands for 'Enhanced On-Board Diagnostics. Second Generation'. EOBD2 tends to refer to manufacturer-specific features available on some OBDII/EOBD tools to access additional parameters/information from a car, over and above the normal parameters and information available within the EOBD/OBDII standard. EOBD2 features are normally highly manufacturer-specific, and will usually only be available for a certain car manufacturer, e.g. Ford.

For safety and reliability of vehicle it is required to automate the fault diagnosis procedure. Use of artificial intelligence and sophisticated and dedicated computers has made it possible and a number of fault detection and isolation (FDI) techniques have been developed in contemporary era. For example a Mercedes S class car (1998-model) utilises computing power of 40 microprocessors and over 100 motors (Denton, 2004) for comfort, control and FDI in the car.

A lot of research is going on in finding simple, reliable and efficient techniques to fulfil mandatory OBD-II requirements for automotives. The basic reasons for this are to ensure human safety, pollution control and increase in fuel efficiency of the engine. A fault in automotive engine can affect all these

three reasons adversely and lead to hazardous situation for the driver and can cause unwanted air pollution and decrease in fuel efficiency. Millions of people travel by cars everyday and therefore the reliability of the vehicle is very important. Faults are often harmful to automobiles if early detection is not made. Prompt detection helps to minimise the maintenance and repair costs of the system and contributes towards increased system reliability.

This research investigates the possibility of finding a simple and reliable FDI technique, which is able to classify different faults along with fault sizes utilising an artificial neural network (ANN) based technique for quick and accurate fault diagnosis in the automotive engine air path. If successful, it will benefit the entire automobile industry and will also contribute to reduction in vehicle emissions due to unwanted faults. The reduction in vehicle emission is the utmost requirement to save ozone layer and the global warming. Sensor faults even have a possibility of accommodation which will save money, time and energy for the driver.

1.3 RESEARCH AIMS AND OBJECTIVES

The principle aims and objectives of this research project are as follows:

1. To carry out a comprehensive literature review on different FDI techniques and engine FDI techniques, their advantages and disadvantages, different faults considered and US, EU/UK regulations for on-board diagnostics and future directions.
2. To modify mean value engine model (MVEM) to simulate different faults and enhance MVEM by including load dynamics etc.
3. To review well established NN training algorithms. Optimise and utilise them for engine FDI.
4. To find a new, reliable and efficient method for engine FDI.
5. To investigate robustness to load changes and parameter variability of the engine under closed-loop control.
6. To implement, evaluate and redesign the technique with real data.
7. To explore possibilities for sensor fault accommodation and unknown fault detection.
8. To conclude the results and explore scope for further research.

The first aim is to explore critically all type of FDI systems in practice especially automotive engine FDI systems and to study their pros and cons. There are strict government regulations for on-board FDI in a vehicle. The first aim also covers an in-depth knowledge of the rules and regulations in practice worldwide regarding emissions from a vehicle on road. The USA is the first country to enforce such legislation which is followed by the UK and subsequently by the European Union and all of them have their own OBD regulations which are more or less the same.

The second aim is to extend current application of neural networks in engine FDI by focusing on the diagnosis of the cause and intensity of faults in an engine air path under different operating conditions. A Mean Value Engine Model (MVEM) simulation (Hendricks et al. 1993, 1996 and 2000) is used initially for development of the techniques and performance assessment. The MVEM is modified in order to simulate different faults e.g. air leakage in the intake manifold, EGR valve clogged and manifold pressure and temperature sensors malfunction with a bias fault etc. It covers all sorts of faults i.e. component, actuator and sensor. Air leakage is a component fault whereas EGR valve clogged can a fault with the valve actuator or with valve itself.

The third aim is to review and drive some of the well-known algorithms for fault detection and isolation for radial basis function (RBF) neural networks e.g. *K*-means algorithm, *P*-nearest neighbourhood, batch least squares (BLS), recursive least squares (RLS) method etc. Subsequently optimise the algorithms with necessary modifications and use them according to the requirements of the application in engine FDI.

The fourth aim is to find a new neural network algorithm for engine FDI with better classification capability and robustness. The new algorithm should be able to cope with environmental changes, uncertainties and changes due to aging of the engine and other components and sensors. This can be achieved with an adaptive FDI algorithm which has the capability of on-line learning. The new adaptive algorithm should also be compared with the other well known algorithms to demonstrate better classification results and robustness. Robustness should be examined under closed-loop operation for sinusoidal

throttle angle change, load change, engine parameter change (due to aging or some other reason) and also for all the changes happening simultaneously.

The sixth aim is to real data assessment of the developed scheme. To fulfil this objective the real data has to be acquired from an engine test bed in the workshop. The data should be acquired for a wide range of operation of the engine so that the developed FDI scheme may be checked and assessed for all possible different type of engine operational modes. Redesign the scheme if required to optimise the results.

The seventh aim is to explore the possibility of sensor fault accommodation and detection of unknown faults in the engine air path. This is important because it is a kind of immediate repair of the fault which will definitely result in emission control from the vehicle if the sensor fault had increased the emissions. Fault accommodation is also very cost effective because basically no cost is paid for repair as far as the accommodation of fault puts the sensor values in the green area (acceptable area) of its behavioural spectrum under different operating conditions. This is important to note that if the sensor accommodation fails for a particular operating condition then immediate repair will be required in case it has affected emissions adversely. Unknown fault detection capability for an FDI system is also an important aspect because it may be rare but has a possibility for an unknown fault to happen. If the unknown fault is affecting the emissions adversely then it needs to be detected and subsequently rectified as soon as possible.

The last aim of this research is to conclude the results and explore the possibilities for further improvements and the possibilities of implementation of the developed FDI scheme in the electronic control unit of the vehicle. This will require first testing of the scheme with hardware in loop simulation (HILS). If the developed scheme passes satisfactorily in the HILS test then the algorithm codes can be translated from MATLAB codes to C/C++ codes and a digital signal processor can be programmed accordingly and embedded in the vehicle ECU using controller area network (CAN) protocols.

Steady state FDI at different constant-speed engine operations was investigated during MSc project in year 2004 and all the aims of this research

are fulfilled during study of the PhD degree from February 2005 to March 2008.

1.4 OVERVIEW

This thesis is organised as follows:

In Chapter 1, an introduction to why this research is required is given. The legislative requirements for EOBD/OBD-II and other reasons for the motivation of on board diagnostics are discussed. It is not just the desire to improve engine efficiency and human safety but there are wider issues involved to protect the environment from the air pollution which has resulted in acid rain, smog and climate change e.g. the North-eastern section of the United States, the South-eastern section of Canada, Central Europe and Scandinavian countries where acid rain and smog are caused by high numbers of factories, power plants and pollution from vehicles on the road. The principle aims and objectives of this research are also listed and briefly discussed.

In Chapter 2, an extensive literature survey is done on fault detection and isolation (FDI) methods in general and also particularly for automotive engine. A number of FDI systems are thoroughly reviewed e.g. traditional limit alarm system, statistical process control method, fuzzy logic method, parameter estimation, observer based methods, model based methods, structured hypothesis, principle component analysis, wavelet networks, analytical redundancy, hybrid model-based technique, residual generation using parity equations, physical redundancy and other untraditional FDI methods like probabilistic approach etc. Modern artificial intelligence (AI) methods especially neural network based methods are also reviewed for automotive engine FDI e.g. Kohonen network, classical RBF classifier, support vector machine (SVM).

In Chapter 3, mean value engine model (MVEM) is discussed, which is a simplified set of dynamic engine models based on models of the most important engine subsystems. A brief introduction to the working of diesel and petrol engines is given before discussing the basic MVEM sub-models e.g.

fuel mass flow dynamic sub-model, Intake manifold filling dynamics and crankshaft speed dynamics.

In Chapter 4, an off-line fault diagnosis method for automotive engine is investigated. This method utilises the well known algorithms namely *K*-means, *P*-nearest neighbours and batch least-squares (BLS) method for an RBF neural network fault diagnosis system. This chapter also includes simulation of air leakage in intake manifold, EGR valve clogged and intake manifold pressure and temperature sensor faults. The data is collected from MVEM in a variety of operating conditions for all the four faults (with four different intensities for each fault) and no-fault states. The neural network is trained and tested on different sets of data and the limitations and short-comings of the FDI system are noticed.

In Chapter 5, a new adaptive fault diagnosis algorithm for on-line FDI of an automotive engine is developed and mathematically derived. The developed technique utilises the gradient descent method for the widths and recursive least-squares algorithm for choosing the weights in the output layer of the RBF classifier. The adaptive classifier is trained and tested for four different faults (with four different intensities for each fault) and the results are compared with a non-adaptive classifier developed previously in Chapter 4. This chapter highlights how the short-comings of the non-adaptive (off-line) classifier are overcome by the adaptive (on-line) classifier by comparing the results from both.

In Chapter 6, the robustness assessment of the developed adaptive FDI technique is investigated under closed-loop control. A PID controller is designed for crankshaft speed feedback control and the robustness of FDI technique is assessed under different load conditions, sinusoidal throttle angle change, speed set point change, engine parameter change and all the changes taking place simultaneously. The evaluations are performed on mean value engine model (MVEM) simulations and are also compared with simulation results from the non-adaptive classifier. The simulation results confirm the proposed adaptive FDI system is robust against various uncertainties, disturbances, and environment changes.

In Chapter 7, real engine data is collected from the Volkswagen engine test bed when the engine runs under different driving conditions in terms of different speeds and loads. Five different sensor faults are simulated on no fault data collected for a wide range of engine operation. Adaptive FDI system is evaluated against the real data and the robustness assessment is also done. The experiments show that all simulated sensor faults were detected and isolated clearly.

In Chapter 8, fault accommodation for four different sensors and unknown fault detection for three different unknown faults are investigated. The unknown faults are investigated using novelty detection (Bishop, 1994) technique. The non-adaptive RBF classifier is able to detect two out of three unknown faults correctly. Sensor fault accommodation is carried out for four different sensors on simulated engine no-fault data using predictive neural networks. The developed accommodation scheme is able to accommodate sensor faults with less than 5% average error.

In Chapter 9, conclusions are drawn for each major section of the thesis briefly. Further research work is discussed to bridge the gap between the academic research and its industrial application in real life. The developed adaptive FDI system is required to be tested with hardware in loop simulation (HILS) before considering it for implementation in the electronic control unit of a real car/vehicle. A concrete proposal for HILS testing and further investigation of adaptive FDI and further development of other more efficient techniques for unknown fault detection and accommodation of sensor faults is presented.

CHAPTER 2

LITERATURE SURVEY

2.1 BASIC FDI CONCEPTS

A fault can be defined as any kind of malfunction in a system which leads to an unacceptable behaviour or poor performance of the overall system. The malfunctions can occur any where in the healthy system such as on actuators, components or sensors and are called actuator faults, component faults and sensor faults respectively. A fault in a dynamic system is a change of the system structure or a deviation of the system parameters from the nominal situation. The set of interacting components of the plant or the interface between the plant and the controller are changed by the fault. These faults cause variance in the dynamical input/output (I/O) properties of the plant from the nominal ones and, hence, change the performance of the closed-loop system which further results in a degradation or even a loss of the system function.

In continuous-variable system described by an analytical model (e.g. differential equation), faults are usually described as additional external signal or as parameter deviation. The first form of faults is called additive faults and is represented by an unknown input that enters the model equation as addend. The second form is called multiplicative fault: in this case the system parameters are scaled depending on the fault size. Also disturbances and model uncertainties change the plant behaviour. Disturbances are usually represented by unknown input signals that have to be added up to the system

output, while model uncertainties change the model parameters in a similar way as multiplicative faults. Faults are often classified as follows:

- Plant component faults: faults that change the dynamical I/O properties of the system.
- Sensor faults: the plant properties are not affected, but the sensor readings have substantial errors.
- Actuator faults: the plant properties are not affected, but the influence of the controller on the plant is modified.

A remark is necessary concerning the distinction of the notions of fault and failure. A fault causes a change in the characteristics of a component such that the mode of operation or performance of the component is changed in an undesired way. Hence the required specifications on the system performance are no longer met. However, a fault can be worked around by fault-tolerant control so that the faulty system remains operational. The notion of a failure, describes the inability of a system or a component to accomplish its function.

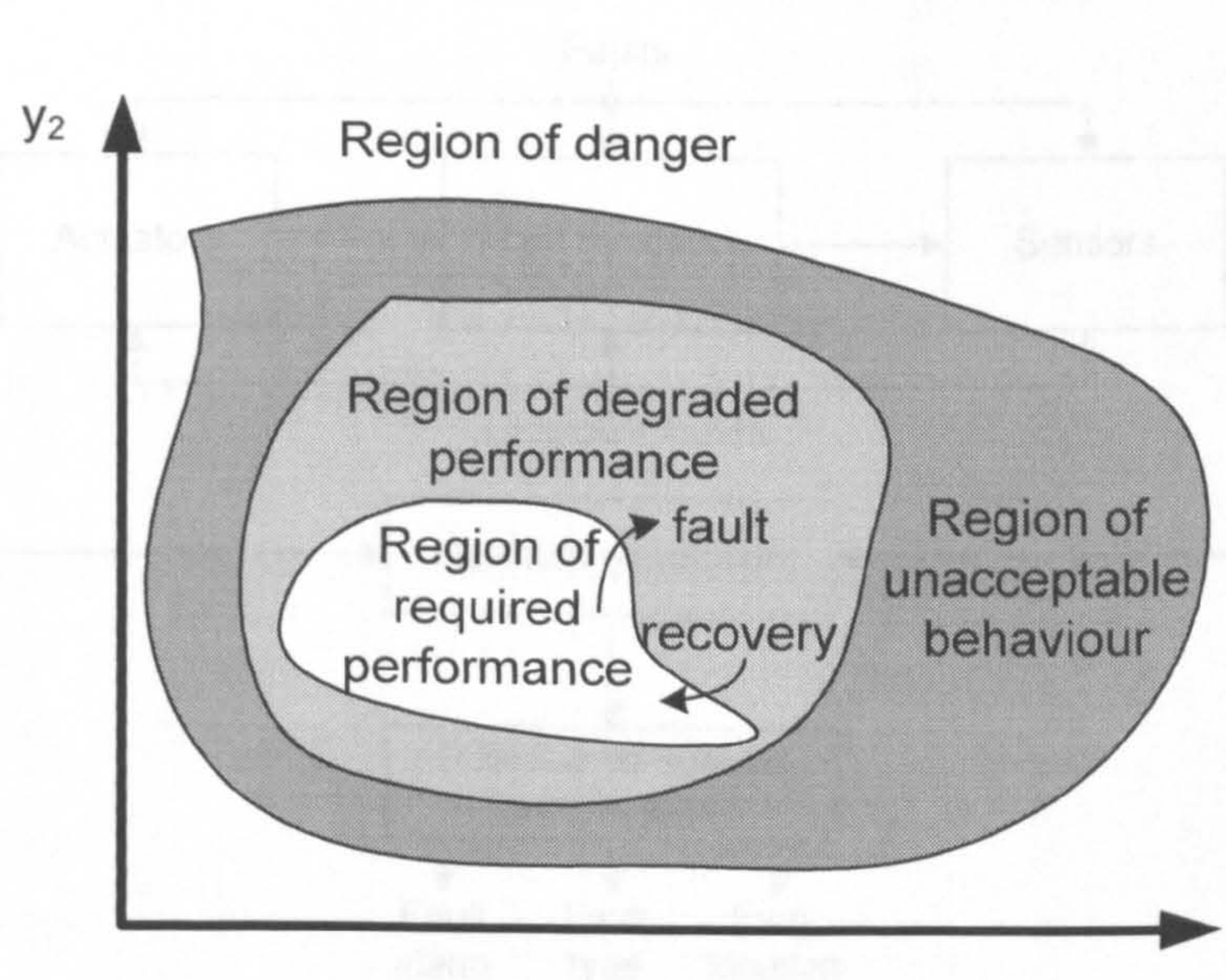


Fig. 2.1: System subject to fault

Assuming that the system performance can be described by two variables y_1 and y_2 (Fig. 2.1). In the region of required performance, the system satisfies its function. During its time of operation the system should remain in this

region. The controller makes the nominal system remain in this region despite of disturbances and uncertainties of the model. The controller may even hold the system in this region if small faults occur.

A block diagram for FDI system with unknown inputs and faults inputs is shown in Fig. 2.2 (Yu, 2005). The system measurements are used for residual generation and residual evaluation. The residual evaluation further leads to fault detection and isolation. To represent a non-linear dynamic system subjected to faults and unknown inputs, the following bilinear state equation model (Frank, 1994) can be used:

$$\dot{x}(t) = g(x(t),u(t)) + \delta_g(x(t),u(t),f(t)) + f_a(f(t),x(t),u(t)) \quad \text{(2-1a)}$$

$$y(t) = h(x(t),u(t)) + \delta_h(x(t),u(t),f(t)) + f_s(f(t),x(t),u(t)) \quad \text{(2-1b)}$$

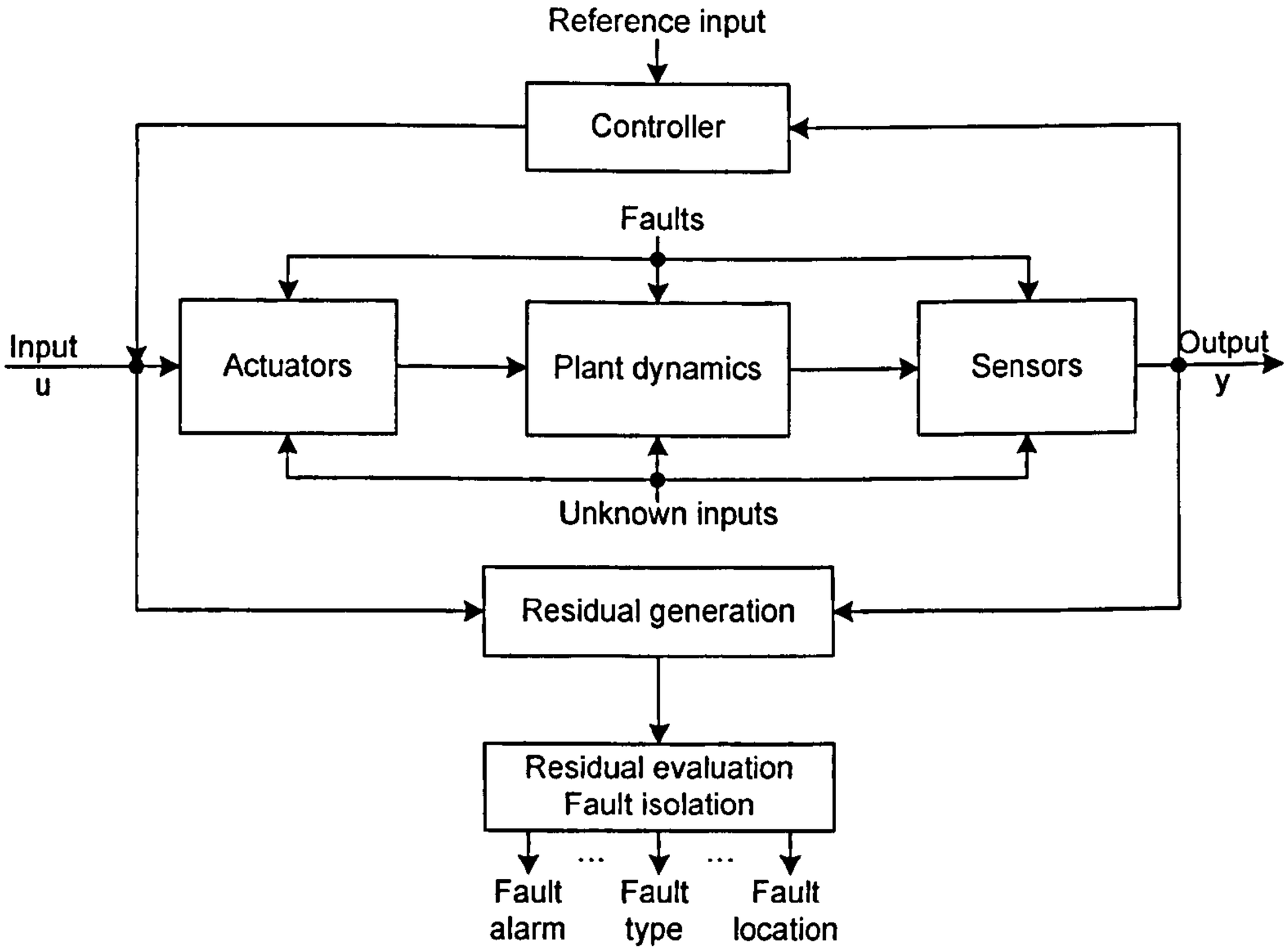


Fig. 2.2: Block diagram of an FDI system

Where $x(t) \in R^n$, $u(t) \in R^m$ *and* $y(t) \in R^p$ are state, input and measured output vectors respectively. $g(.)$ *and* $h(.)$ are known vector functions of

compatible dimensions. Further, $f(t) \in R^q$ is a fault vector and $f_a($ and $f_s(.)$ are known vector functions representing the effects of component and actuator faults and sensors faults respectively. The vector functior δ_g represents the modelling errors of the components and actuators and δ_h those of the sensors. δ_g and δ_h can also be used to represent signal disturbance and noise. They are generally referred to as unknown inputs

In the special case of linear systems, the following model is considered

$$\dot{x}(t) = Ax(t) + Bu(t) + Ed(t) + Gf_a(t) \qquad 2\text{-}2a)$$

$$y(t) = Cx(t) + Du(t) + Qf_s(t) \qquad 2\text{-}2b)$$

Where *A, B, C* and *D* are known matrices and *d(t)* is an unknown input. If *E* is known, *d(t)* is termed structured, otherwise it is termed as unstructured. The fault signals $f_a(t)$ and $f_s(t)$ are unknown time functions and their distribution matrices *G* and *Q*, may or may not be known, depending upon the structure and knowledge of the system. Without loss of generality, it can be assumed that *E, G, C* and *Q* are full rank.

Fault Detectability and Isolability

A transfer function model of the system (2-2) is given as

$$\begin{aligned} y(s) &= G_u(s)u(s) + G_d(s)d(s) + G_a(s)f_a(s) + G_s(s)f_s(s) \\ &= G_u(s)u(s) + G_d(s)d(s) + G(s)f(s) \end{aligned} \qquad (2\text{-}3)$$

Based on this form, two basic definitions are given by Patton (1994).

Definition 2.1. Fault detectability: a fault is said to be detectable if $G_i(s) \neq 0$, where $G_i(s)$ is the i^{th} sub-matrix of the transfer matrix $G(s)=[G_1(s), \ldots, G_p(s)]$ in (2-3)

Definition 2.2. Fault isolability: given two signals $f_1(t)$ and $f_2(t)$ such that the system response to the two signals is

$$y(s) = G_1(s)f_1(s) + G_2(s)f_2(s)$$

then the signal $f_1(t)$ is said to be isolable from $f_2(t)$ if

$$R\{G_1(s)\} \not\subset R\{G_2(s)\} \qquad (2\text{-}4)$$

where $R\{.\}$ represents the range space of a matrix and $G_1(s)$ and $G_2(s)$ are the transfer functions relating f_1 and f_2 to *y*.

In these terms, the observer based robust FDI problem can be stated as follows:

- For fault detection a fault effect must be isolable from the effect of the unknown inputs.
- For fault isolation the effect of a fault must be isolable from the effect of the unknown inputs and the other faults.

2.2 REVIEW OF GENERAL FDI METHODS

2.2.1 Limit Alarm Method

In the limit alarm system, the process variable ranges are defined for a healthy operation along with a reasonable tolerance for upper and lower limits. If the value of any process variable is observed to exceed the upper or the lower tolerance limit then it is considered as a fault condition. Automatic shut down of the process is possible in case of a fault condition. Such systems heavily rely on human expert interface for the fault isolation.

Fault detection using limit boundaries is shown in Fig. 2.3. The healthy operating range for the process variable is 20 to 25. The upper limit and lower limit for the variable is set to 27.5 and 17.5 respectively. This tolerance is

carefully calculated and designed by a process control engineer. The signal falling outside of the defined lower and upper limits causes alarm signal and detects a fault in the process. An auto shut down limit is also defined in case the process variable exceeds the limits too much to save the plant from damage. As shown in the graph the auto shut down limit for the process variable is set to 35. The biggest advantage of such systems is its simplicity and inexpensiveness.

Fault detection of sensor malfunction can be carried out by hardware redundancy which involves comparing the measurements of repeated hardware elements for consistency (Patton et al., 1989). This method is expensive and requires extra space for the additional sensors to install. On the other hand analytical redundancy method uses representative model instead of repeated hardware sensors etc. The signals from the model are compared with the signals from the actual process for the fault detection.

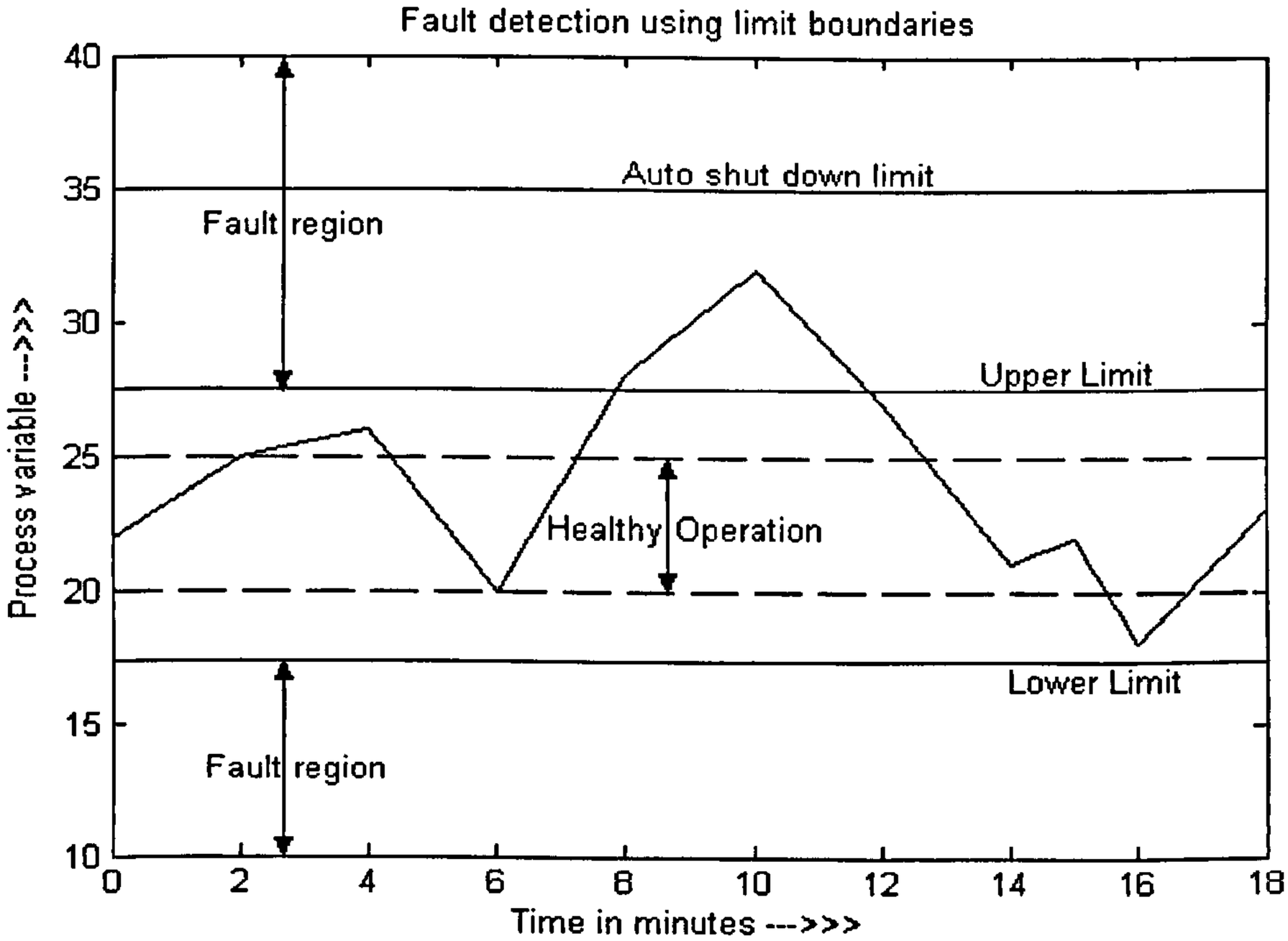


Fig. 2.3: Fault detection using limit boundaries

However, for a dynamic system the output relies on the input and therefore a constant limit is not useful, while variable limits are very difficult to choose.

Therefore this method is very limited to be used for a small range of simple systems.

2.2.2 Statistical Process Control (SPC)

The SPC methods are based on charting the final quality variables of a process. The real time quality measurements are taken by using on-line analytical instruments such as spectrometry and chromatography systems. These instruments are very expensive and are available for limited applications. Therefore many processes adopt an off line method for obtaining the quality measurements. After collecting a sample of process data representing no fault state statistical confidence bounds are calculated for each monitored variable. The confidence bounds can have confidence limits such as 97% or 99% etc. These confidence bounds are incorporated into a SPC chart and the process variables can then be monitored on line to see if they fall within expected limits. Exponentially weighted moving average charts, Shewart charts (Mcfarlane, 1996) and cumulative charts are the most commonly used charts for SPC. These charts are used to detect the occurrence of any abnormal activity.

Statistical process control techniques in particular the control charts have been widely used in manufacturing industry (Xie *et al.*, 2001). Usually control charts are implemented for the purpose of process monitoring. When a process is considered out of control, an alarm is raised so that engineers can look for assignable causes of variation and try to eliminate them. Traditional control charts aroused in such a way that corrective action are taken only after the occurrence of the out of control signal which indicates that the process performance has changed to a state significantly far from the original state.

SPC uses output measurements to control the stability of a process and to detect causes of out of control situations (Montgomery, 1996). The main drawback of SPC is that its effective implementation cannot be achieved unless we have a good understanding of the key factors that will make the application successful (Rungasamy *et al.*, 2002).

Many industrial processes have a large number of inputs and output variables and it may not be practical to draw charts for all the variables and apply SPC strategy for monitoring without first doing principal component analysis (PCA) to limit the number of variables to be monitored for faults and safe operation.

2.2.3 AI and Knowledge based methods

It is not always possible to develop model based FDI system for a process especially when quantitative model are not available due to complexity of process or due to some other reasons. In such cases quantitative knowledge based methods are used for fault diagnosis. These methods can also be designed to complement the quantitative methods by incorporating process knowledge at the residual evaluation stage. Knowledge based diagnosis methods mainly fall into two categories, symptom based and quantitative model based methods. Symptom based methods are based on knowledge about the process history and are evaluated using rule based systems such as expert systems or fuzzy logic. With qualitative models, the knowledge is derived in terms of facts and rules from the description of system structure and behaviour in first principles (*Weerasinghe, 1998*). A main component of qualitative modelling is causal graphs, such as signed directed graphs (SDG), which are based on tracing process malfunctions to their source.

In this section, the expert system, fuzzy logic and neural network methods will be discussed separately.

A. Expert Systems Method

Expert systems are used in a variety of areas, and are the most popular developmental approach in the field of artificial intelligence. The major use of artificial intelligence today is in expert systems, AI programs that act as intelligent advisors or consultants. Drawing on stored knowledge in a specific domain, an inexperienced user applies inference capability to tap the knowledge base. As a result, almost anyone can solve problems and make decisions in a subject area nearly as well as an expert. It is not easy to give a precise definition of an expert system, because the concept of expert system

itself is changing as technological advances in computer systems take place and new tasks are incorporated into the old ones. In simple words, it can be defined as a computer program that models the reasoning and action processes of a human expert in a given problem area. Expert systems, like human experts, attempt to reason within specific knowledge domains.

An expert system permits the knowledge and experience of one or more experts to be captured and stored in a computer. This knowledge can then be used by anyone requiring it. The purpose of an expert system is not to replace the experts, but simply to make their knowledge and experience more ideally available. Typically there are more problems to solve than there are experts available to handle them. The expert system permits others to increase their productivity, improve the quality of their decisions, or simply to solve problems when an expert is not available. Valuable knowledge is a major resource and it often lies with only a few experts. It is important to capture that knowledge so others can use it. Experts retire, get sick, move on to other fields, and otherwise become unavailable. Thus the knowledge is lost. Books can capture some knowledge, but they leave the problem of application up to the reader.

Expert systems provide a direct means of applying expertise. An expert system has three main components: a *knowledge base*, an *inference engine*, and a *man-machine interface*. The knowledge *base* is the set of rules describing the domain knowledge for use in problem solving. The prime element of the *man-machine interface* is a working memory, which serves to store information from the user of the system and the intermediate results of knowledge processing. The *inference engine* uses the domain knowledge together with the acquired information about the problem to reason and provide expert solution.

In nut-shell an expert system can be thought of a body that holds a large amount of expert knowledge about the subject under consideration in an advisory capacity to users of lesser knowledge of different situations. The information may take the form of system diagnosis and prognosis and advice on possible corrective actions. For the design of a diagnostic expert system for a process, the expert system should hold the knowledge of process experts, such as process operators and engineers, information about the

process behaviour under no fault and fault conditions and manuals of process equipments. The knowledge base is then compiled in the form of facts and rules and can be used as fault diagnosis tool with in the domain of expertise. High costs and long development time are two major disadvantages of such systems.

The rule-based expert systems on process diagnosis are primarily shallow reasoning-based expert systems which help users to diagnose the cause of process deviations and suggest corrective actions. These expert systems generally contain human expertise in the form of rules or decision trees which are often augmented with causal graphs or decision trees (Melton *et al.*, 1997).

Ebersbach and Peng (2008) developed an expert system to analyse vibration data with similar accuracy as an expert maintenance engineer in an automated software package allowing high analysis output, and hence suitable for commercial condition monitoring laboratories or on-site use.

Tomaszewski et al., (2007) compared the performance of the statistical fault prediction models with expert estimations and found that the statistical methods clearly outperform the expert estimations. The main reason for the superiority of the statistical models was their ability to cope with large datasets.

It is possible only in a narrow domain to programme a computer system to emulate the decision-making ability of an expert human. It is very difficult to encode enough knowledge into a system to make it multi-dimensional expert and therefore expert systems not suitable for FDI a large systems which have hundreds of variables and parameters.

B. Fuzzy Logic Method

Fuzzy logic means approximate reasoning, information granulation, computing with words and so on. Ambiguity is always present in any realistic process. This ambiguity may arise from the interpretation of the data inputs and in the rules used to describe the relationships between the informative

attributes. Fuzzy logic provides an inference structure that enables the human reasoning capabilities to be applied to artificial knowledge-based systems. Fuzzy logic provides a means for converting linguistic strategy into control actions and thus offers a high-level computation.

Fuzzy logic provides mathematical strength to the emulation of certain perceptual and linguistic attributes associated with human cognition, whereas the science of neural networks provides a new computing tool with learning and adaptation capabilities. The theory of fuzzy logic provides an inference mechanism under cognitive uncertainty; computational neural networks offer exciting advantages such as learning, adaptation, fault tolerance, parallelism, and generalization.

Fuzzy logic is based on the way the brain deals with inexact information. Fuzzy systems are structured numerical estimators. They start from highly formalised insights about the structure of categories found in real world and then articulate fuzzy IF THEN rules as a kind of expert knowledge. Fuzzy systems combine fuzzy sets with fuzzy rules to produce overall complex nonlinear behaviour. In the past decade, fuzzy systems have supplanted conventional technologies in many scientific applications and engineering systems especially in control systems and pattern recognition (classification).

One of the biggest differences between crisp and fuzzy sets is that the former always has unique membership functions whereas every fuzzy set has an infinite number of membership functions that may represent it. This enables fuzzy systems to be adjusted for maximum utility in a given situation. In a broad sense, as pointed out by Professor Lotfi Zadeh (inventor of fuzzy logic, 1973) any field can be fuzzified and hence generalised by replacing the concept of a crisp set in the target field by the concept of a fuzzy set. The incoming data from the process is fuzzified to give linguistic labels such as, large, medium, small etc rather than crisp values. The fuzzification is evaluated according to its degree of membership of a certain specified membership function. The logical operators AND, OR and NOT are applied to the membership function when developing the rule base. Structure of a simple fuzzy logic system is given in Fig. 2.4.

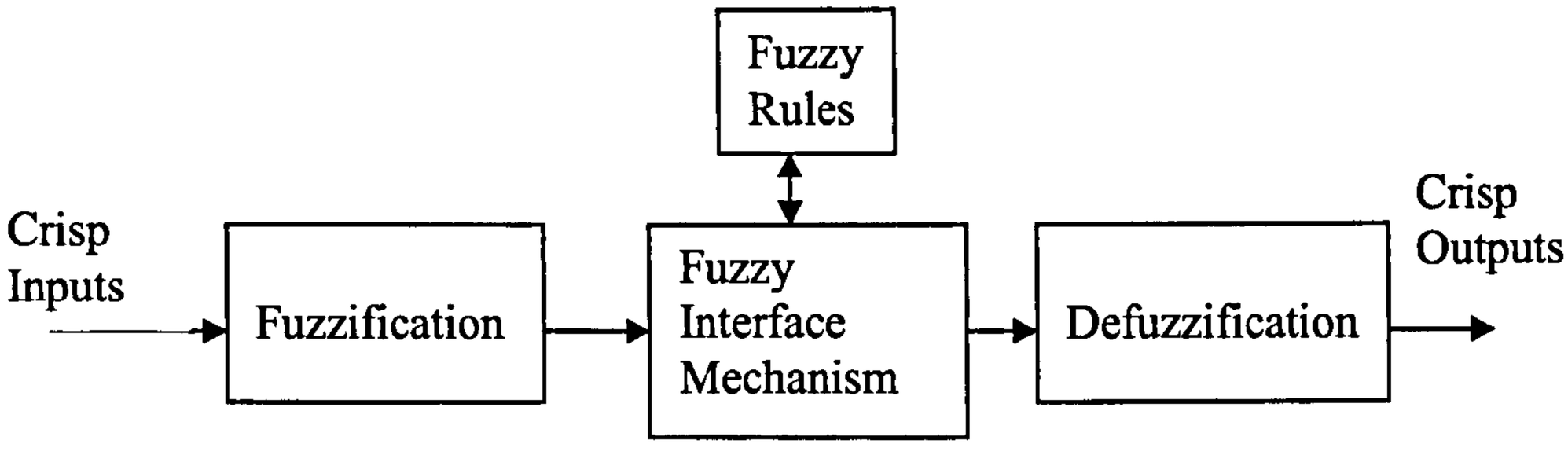


Fig. 2.4: Structure of a simple fuzzy logic system

One major feature of fuzzy logic is its ability to express the amount of ambiguity in human thinking and subjectivity in a comparatively undistorted manner.

Laukonen and Passino (1995) trained fuzzy systems to perform estimation and identification. A fuzzy approximation technique was used to solve nonlinear estimation and system identification problems. A fuzzy system was constructed to interpolate between input and output data to provide an approximation for the function defined by the input-output data pair association. Uniform training and the modified learning from examples technique were illustrated by a simple pendulum example and the use of the modified learning from examples approach was demonstrated in constructing a fuzzy system which can identify actuator failures in an F-16 aircraft. Hissel *et al.,* (2004), presented a diagnosis oriented model of a proton exchange membrane fuel cell (PEMFC). The diagnosis model was based on fuzzy logic system and was tuned with the help of genetic algorithms. The fuzzy classifier worked well for two different types of faults in the fuel cell system i.e. accumulation of water/nitrogen in the anode compartment and drying of the proton exchange membrane.

A fuzzy model was implemented (Lu *et al.*, 2000) to detect vacuum leak in the electric engine controller (EEC) as part of the end-of-line test at automotive assembly plants. The fuzzy model learnt automotive diagnostic knowledge through machine learning techniques. The fuzzy model generated fuzzy rules and optimised fuzzy membership functions automatically and this auto-tuning made the system adaptive to different vehicle models. The machine learning

system allowed the human knowledge to be incorporated in the knowledge base through the generation of prior rules and the modification of the membership functions after rule generation. The diagnostic system was tested on two different vehicle models and the test results showed that the system was effective, fast and suitable for running on PC platform.

C. Artificial Neural Network (ANN) Method

An artificial neural network (ANN) is an information-processing system that has certain performance characteristics in common with biological neural networks. Artificial neural networks have been developed as generalizations of mathematical models of human cognition or neural biology, based on the assumptions that:

- Information processing occurs at many simple elements called neurons.
- Signals are passed between neurons over connection links.
- Each connection link has an associated weight, which, in a typical neural net, multiplies the signal transmitted.
- Each neuron applies an activation function (usually nonlinear) to its net input (sum of weighted input signals) to determine its output signal.

A neural network is characterised by (1) its patter of connections between the neurons (called its architecture), (2) its method of determining the weights on the connections (called its training, or learning, algorithm), and (3) its activation function. Each neuron is connected to other neuron by means of directed communication links, each with an associated weight. The weights represent information being used by net to solve a problem. Neural nets can be applied to a wide variety of problems, such as storing and recalling data or patterns, classifying patterns, performing general mappings from input patterns to output patterns, or finding solutions to constrained optimisation problems.

Pattern recognition is one of the major applications of ANNs. Pattern recognition is basically 'classification' of known sets of patterns. Artificial

Neural Networks (ANNs) are found very useful for this purpose. The main applications of pattern classification are in the field of image and speech recognition. They are also used in the field of fault diagnosis to classify the input patterns. It requires a set of process data that covers all fault conditions to be diagnosed. The input data to ANN consists of all the fault states including no fault state. The data is normalised and scaled to appropriate level before putting it through ANN for training.

If sufficient data is provided, an ANN can learn any type of non-linearity. A properly trained ANN can further be used for classification of faults. ANN has ability to generalise to new inputs i.e. a trained network is capable of providing meaningful outputs when presented with input data that has never been used before.

Yu *et al.,* (1999), investigated radial basis function (RBF) neural networks for process fault diagnosis. The output prediction error between a neural network model and a non-linear dynamic process was used as residual for diagnosing actuator, component and sensor faults for a real multivariable chemical reactor (Yu *et al.,* 2003). Real data experiment showed that sensor faults can be detected and isolated without a process mathematical model by using a semi-independent neural model. It demonstrated a general approach which could be employed for sensor fault diagnosis of any real systems but the successful classification would depend on the model sensitivity for residual generation.

Other authors investigated fault classification for double-circuit lines based on combined supervised and unsupervised neural networks (Aggarwal et al., 1999). A neural network based on combined supervised and unsupervised training methodology provided the ability to accurately classify the fault type by identifying different patterns of the associated voltages and currents. This hybrid classifier was also compared with supervised classifier and it was found that the hybrid classifier performed better and was suitable for double-circuit-line faults classification. In the hybrid classifier, first self organising maps (SOM) were used with unlabelled data to form internal clusters. Then labels are assigned during the supervised stage and back propagation (BP) algorithm is used for supervised stage.

This two stage FDI method is effective and suitable for double-circuit line faults. The reason is that faults in double-circuit lines have less number of classes which are diagnosed in stage 1 of the method using SOMs. Once the class of fault is known then in the second stage (supervised stage) can pin point the fault more precisely and isolate it. This technique may not be suitable for systems which have larger numbers of fault classes.

Yang *et al.*, (2004), proposed a Kohonen neural network (KNN) based classifier for fault diagnosis of rotating machinery. They showed that the neural network based on adaptive resonance theory (ART) were capable of carrying out on-line training without forgetting previously trained patterns. For the purpose of testing the proposed network and to compare it with the other FDI systems, the vibration signal was selected as raw inputs due to simplicity, accuracy and efficiency. On comparison it was found that ART-Kohonen neural network performed better than self-organising feature maps (SOFM), learning vector quantisation (LVQ) and radial basis function (RBF) networks for this particular application of diagnosing rotating machinery where the vibration signals substantially change with the different types of the fault e.g. faulty bearing, misalignment and unbalance. Many authors used KNN for process monitoring and fault diagnosis. This method may not be suitable for the detection of faults which do not cause substantial change in the vibration characteristics of the machine.

Jamsa-Jounela *et al.*, (2003) investigated application of Kohonen self-organising maps in conjunction with heuristic rules to detect equipment malfunctions in a copper flash smelter industry with an objective to detect process failure at an early stage. The FDI system worked satisfactorily but expert-knowledge of the human operator about the process still plays an important role in fault detection. To eliminate the role of an expert-human operator, more information about process states is required which cannot easily be measured therefore it is necessary to rely on operator's observations and make necessary observations.

Kowalski and Orlowska-Kowalska, (2003), investigated self organising Kohonen networks and multilayer perceptron (MLP) networks for FDI of rotor, stator and roller bearing of an induction motor. Measurement data of stator current and mechanical vibration spectra were used for training and testing of

the neural networks. SOMs were used at initial stage of classification as an introductory step and then MLP networks were used for final fault detection and isolation. The results demonstrate that the proposed method can effectively used for the recognition of induction motor faults.

Jakubek and Strasser (2002) proposed an automatic fault diagnosis scheme for automotives and their approach modelled the distribution function of available fault free data using ellipsoidal basis function network. More attention was paid to automatic adaptation of pre-existing network to new data points using gradient optimisation with algebraic constraints and numerical examples with real data showed good results. The detection scheme processes hundreds of measurements but the main problem is that there is no further information available except the available data. The authors have not tried PCA to reduce the number of measurements to be processed. This is important because some measurements do not contribute any information for residual generation.

Dalmi *et al.*, (1999), investigated four different FDI algorithms e.g. radial basis function and multilayer perceptron for supervised training and Kohonen and counter-propagation networks for unsupervised training for fault diagnosis of an autonomous mobile vehicle. The authors found that counter-propagation algorithm is better for the detection of unknown faults and RBF network is more suited for the detection of known faults. The same authors also investigated residual based pattern recognition approach for FDI (Dalmi *et al.*, 2001). Unsupervised neural networks with on-line training were used for the classification of the unknown faults.

Scholkopf *et al.*, (1997), compared classical RBF classifier, support vector (SV) machine with Gaussian kernel, and a hybrid system with the centres determined by the SV method and the weights trained by back-propagation algorithm. All the three systems were tested on United States postal service database of hand written digits. The results showed that the SVM achieves the highest accuracy in correct recognition of hand written digits followed by the hybrid system. Therefore the SV approach is not only well founded theoretically but also is superior in a practical application.

Vong *et al.*, (2006), used least squares SVM and Bayesian interface for prediction of automotive engine power and torque. Approximate power and torque model of an engine was determined by training the sample data acquired form the dynamometer and this resulted in reduced number of dynamometer tests required for engine tune up. Bayesian framework was applied to infer the hyper-parameters used in LS-SVM so as to eliminate the work of cross validation which led to substantial reduction in training time. The LS-SVM methodology was also compared with multilayer feed forward neural networks and was found better for this particular application. The LS-SVM training time is also considerably small as compared to traditional neural network method. The construction of gasoline automotive engine power and torque functions using LS-SVM is a successful attempt and may also be suitable to apply to other type of engines.

Neural networks can be used even for unknown fault diagnosis. Li *et al.*, (2002) presented a novel technique for the determination of an appropriate threshold for RBF classifier. Threshold based neural network output interpretation is appropriate for FDI systems. Generally neural network based FDI systems not only classify known *normal* and *fault* input vectors but also recognise that a particular input is neither *normal* nor a member of the one of the existing *fault* categories. Bishop (1994) called it 'novelty detection'. By using a threshold based classifier outputs of the neural network may be readily interpreted as an *unknown fault* when none of the *normal* or *fault* output neuron exceeds the threshold. According to Li *et al.*, (2002), the ability to detect unknown faults may be at the expense of a decrease in the classification performance for known classes and therefore a second phase should be performed. In this second phase the classifier threshold may be modified to translate the *unknown faults* into *known faults*. When sufficient data has been collected about *unknown fault* then the classifier should be retrained using all available data. The authors have derived a novel technique for threshold determination of RBF network where there may be possible new classes or unknown faults. The results suggest a two-phase approach to RBF classifier when there is a possibility of unknown faults. The authors have successfully overcome the problem of decrease in classification performance for known classes because of the threshold adjustment made for the detection of unknown classes by performing a second phase of known fault detection with modified threshold.

2.2.4 Model Based Methods

A variety of diagnosis methods have been proposed under the umbrella of model-based techniques. The feature of all these techniques is that some form of mathematical knowledge of the process of interest along with inputs and outputs are used to generate superfluous information about that process. This redundant information is then used in a diagnostic process to arrive at decisions regarding fault or no-fault conditions. Model based diagnosis has proven a useful alternative to the traditional approaches, and has potentially the following advantages:

- • It can provide a higher diagnosis performance, in terms of detecting faults with small size and shorter detection time.
- • Different faults can be isolated.
- • It can be performed over a large operating range.
- • Disturbances can be compensated for, which enables high diagnosis performance in spite of present disturbances.
- • It is applicable to a wider range of components. Not all hardware can be duplicated.
- • No extra hardware is needed, which saves space, weight and sometimes even money.

The disadvantage of model-based diagnosis is the need of an accurate model, computing capability and perhaps also a more complex design procedure. Building the model is probably what takes the most work when designing a diagnosis system and it also is the most important part since the model normally is what limits the performance of the diagnosis system. The analytical or model based approach has become a subject of tremendous research in recent years. In a survey by Isermann and Balle (1997), it was stated that model based methods are the most frequently applied methods for fault diagnosis. A variety of diagnosis methods have been proposed under the umbrella of model-based techniques. A number of survey papers can be found on the subject such as Isermann (1997), Gertler (1988), Frank (1996) and Chen and Patton (1999). The feature of all these techniques was that some form of mathematical knowledge of the process of interest along with inputs and outputs were used to generate superfluous information about that process. This redundant information was then used in a diagnostic process to

arrive at decisions regarding fault or no-fault conditions. Model based methods used 'analytical redundancy' instead of 'physical or hardware redundancy'. Figure 2.5 shows the structure of a model based FDI system.

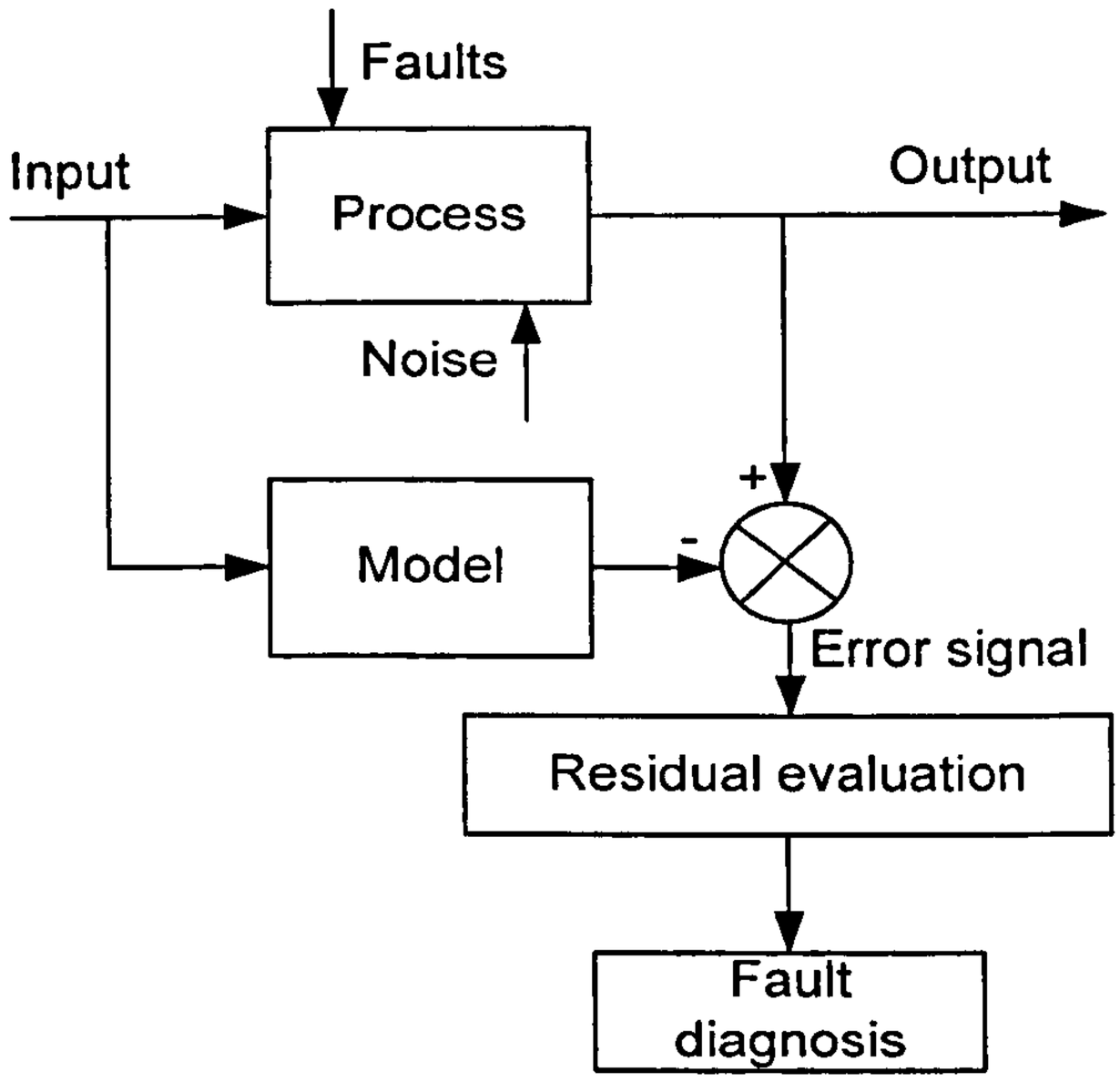


Fig. 2.5: Model based FDI system

The error between the process and the model output is called residual. A single residual is sufficient for fault detection but a set of residuals is required for fault isolation. In healthy condition the residual is zero. Residual is generated only in the fault condition when the process variables deviate from the normal values.

Pisu *et al.*, (2003) presented the design of a new model-based fault diagnosis method for monitoring the vehicle chassis performance based on a hierarchical structure of the FDI schemes that reduced the computational effort. A set of low level detection unit was realised to determine basic faults and then the information was shared with the top level to determine the different types of faults. To demonstrate the validity of the approach the hierarchical structure was applied to a brake-by-wire system without mechanical backup. The validity of this approach was then examined by extensive number of simulations performed under noisy measurements and unmodelled dynamics. The approach was successful.

Butler *et al.,* (1999), discussed a simulation and modelling package developed at Taxas A&M University, V-Elph 2.01. It was composed of detailed models of four major types of components e.g. electric motor, internal combustion engines, batteries and support components that could be integrated to model and simulate drive trains having all electric, series hybrid, and parallel hybrid configurations. Simulation results such as fuel consumption, vehicle emissions and complexity were compared for different types of vehicles. The authors have basically discussed a new simulation and modelling package to study issues related to electric vehicles (EV) and hybrid electric vehicles (HEV) design such as energy efficiency, fuel economy and vehicle emission. This modelling package can be run on any PC or UNIX based computer.

Crossley and Cook (1991) developed a nonlinear mathematical model of a four cylinder spark ignition engine to incorporate into an overall vehicle driveline model. Emphasis was placed on the formulation of engine component functional relationships and the validation of modelled system dynamics by engine dynamometer testing. This work showed that a relatively low order engine model can capture the main transient effects for use in traction control system design.

Modelling vehicle engine diagnostics as a signal fault diagnostic problem requires a good understanding of signal behaviour relating to various vehicle faults. These signals range from simple binary modes to complex spark ignition timing signals. Two important tasks in vehicle signal diagnostics are to find what signal features are related to various vehicle faults and how can these features be effectively extracted from signals. Crossman *et al.,* (2003), presented results in signal faulty behaviour analysis, automatic signal segmentation, feature extraction and selection of important features. This research is challenging to automotive fault diagnosis and the results are useful to other applications such as pattern recognition and intelligent systems. Authors have also developed agent-based system for vehicle FDI.

A hybrid feature selection algorithm was evaluated using a fuzzy diagnostic system developed for vehicle signal diagnosis which automatically generated fuzzy rules and fuzzy membership functions from training data and optimised

the fuzzy membership functions (Lu and Chen, 1998). The approach was successful.

Frisk and Nyberg (2002) presented an algorithm for designing linear residual generators. The algorithm was able to design residual generators for any model described by general linear differential-algebraic equations. The algorithm was based on polynomial matrix algebra for which efficient computational tools were commercially available. The algorithm could handle all types of models handled by earlier algorithms (e.g. models described by state space, transfer function and descriptor form) but also a more general class of models not handled by previous algorithms.

Yu et al., (2005) investigated application of empirical mode decomposition (EMD) method and Hilbert spectrum for fault diagnosis of roller bearings. The orthogonal wavelet bases were used to translate vibration signals of a roller bearing into time scale representation. The real vibration signals measured from roller bearings with faults were analysed and the results showed that the proposed method was better than the traditional envelop spectrum method in extracting the fault symptoms. This approach is only suitable for application where the faults affect the vibration signals. If the faults in the system do not affect the vibration signals then they are not detectable using this approach.

A robust FDI scheme for abrupt and incipient faults in nonlinear uncertain dynamic systems was presented by Zhang *et al.*, (2002). The proposed scheme consisted of a bank of nonlinear adaptive estimators. One was the fault detection and approximation estimator and others were for fault isolation (each of them associated with a specific type of fault). The detection and approximation detector was used for normal operating condition to monitor any fault. If a fault was detected then the fault isolation estimators were activated and the fault detection and approximation estimator adopts the mode of approximating the fault by using online approximation method. The main contribution was the design of fault isolation scheme as the key part of the diagnosis architecture. The residual of each fault isolation estimator was associated with an adaptive threshold which could be implemented online by using linear filtering method. This would reduce false isolation decision. The effectiveness of the proposed FDI scheme is illustrated by the example of three tank (MIMO) system.

Hybrid model-based techniques employing quantitative models and graph-based dependency models for intelligent diagnosis were investigated by Luo *et al.*, (2003). It was found promising to integrate intelligent model-based diagnostic processes into the initial design phase for vehicle health management. The results of the test designs from different model-based diagnostic schemes were layered onto a graph-based dependency model which was more compact for real time diagnostic implementation. Intelligent model-based diagnostic process provided a systematically solution for vehicle health management problem.

The most common approach to design diagnostic systems is to use residual generators which are usually constructed considering all the model equations. There are many advantages of considering a small subset of model equations called minimal structurally singular (MSS) sets of equations. Biteus and Nyberg (2003) presented a new method for finding residual generators for MSS sets. This approach was used on a non-trivial non-linear point-mass satellite system. MSS models from the dynamic and partially dynamic approaches were used and the result was analysed. The numerical simulation of MSS models showed good results.

2.3 REVIEW OF ENGINE FDI METHODS

Engine health monitoring process is widely discussed. Fundamentally, the monitoring process involves diagnosing engine parameters which deviate from 'normal' value. Faults can come from several sources. Traditionally, engine faults have only been identified after the engine has broken down. If there is an on board parameter monitoring device to reflect the online engine condition, then diagnosis can be affected earlier. One of the objectives of this research is to design a condition monitoring and fault diagnosis system which can be used to investigate the possible faults at component level for an engine. The system must be capable of detecting and isolating particular faults on the engine.

There are a number of books that have been published on fault detection for engineering processes (Patton, 1989; Chen, 1999; Gertler, 1998; Chiang,

2001) and various fault diagnosis techniques have been reported in the literature. A few detailed survey papers are available (Isermann, 1984; Gertler, 1988). Isermann reported that process faults could be detected when based on the estimation of immeasurable process parameters and state variables. He also described the suitable choice of using parameter estimation for continuous-time models in fault detection. He then reported the use of process model knowledge and model-based (Isermann, 1997) fault diagnosis of technical processes. Gertler pointed out the possibility of using both model-based and model-free methods for fault detection. He asserted that the sensitivity and robustness of models play a role in the selection of a desired model and filtering may be applied to improve sensitivity within a given model framework. Modelling errors affect the fault detection process and may falsify the failure signatures. The following gives a brief overview of some techniques in general terms.

2.3.1 Model Based Engine FDI Methods

The model-based technique is the most common approach to fault detection and isolation. This method utilises an explicit mathematical model of the plant under investigation. The sensor measurements are compared to analytically computed values (i.e. from the model) of the respective variable and the resulting difference is called the residual. The residuals are indicative of the presence of faults in the system. There are a few approaches to residual generation in model based fault diagnosis.

A. Parameter Estimation Method

Parameter estimation is an approach to the detection of parametric faults. A reference model is obtained by identifying the plant in a fault-free condition. The models may be constructed from first principles to relate the model parameters directly to parameters that have physical meaning in the process. Any deviation from the reference plant model serves as a basis for fault detection. This technique may be applied to a non-linear system where the structure is known. The parameter estimation method is appropriate if the process faults are associated with changes in model parameters (i.e.

multiplicative faults), and appropriate mathematical models are available. Benvenuti *et al.*, (2003), presented an estimation technique for injector characteristics based on a set of measurements that could be carried out using sensors present in the car e.g. intake manifold pressure, crankshaft speed, throttle-valve plate angle, injection timings and exhaust air/fuel ratio etc. All these measurements could also be taken by a single sensor placed at the exhaust pipe output known as Universal Exhaust Gas Oxygen (UEGO) sensor. Vemuri (1999) investigated an adaptive technique to estimate the presence of an unknown constant bias in the UEGO sensor. The simulation results showed that the proposed technique achieved robust diagnosis.

Conatser *et al.*, (2004), investigated fault diagnosis of automotive electronic throttle control (ETC) system. The ETC system was modelled and a parity diagnostic strategy applied to detect the presence of any fault. The specific fault was isolated using a parametric estimation methodology. The nonlinear parity residual generation method used forward and inverse system models to compare actual and modelled system input/output values to detect system faults. Numerical results were presented and discussed which demonstrated the performance of the health monitoring algorithms in diagnosing a suite of system failures. The fault diagnosis was successful for ETC system.

B. Observer-based Techniques

The observer-based technique can be used for both non-linear and linear plant models. The plant states may provide important information to the plant operating condition. However, some plants states are known to be non-measurable and an observer can be employed to estimate these plant states.

Once the residuals are obtained, evaluation of these residuals may be carried out. Residual evaluation is a decision making stage to classify the particular faults in the plant. Some of the faults gradually build up and may provide a certain fault pattern. This fault pattern can be recognised based on the history of the plant or some theoretical considerations. Once the residuals are obtained, they may be classified into further categories, i.e. good condition, moderate, serious or danger to give the user a clear picture of the plant condition. The users can then react to these decisions as appropriate.

Model based methods cover a wide class of fault detection and isolation (FDI) techniques and a verity of these methods had been proposed (Frank *et al.*, 2000). The feature of all these techniques was that some form of mathematical knowledge of the process of interest along with inputs and outputs were used to generate superfluous information about that process. This redundant information was then used in a diagnostic process to arrive at decisions regarding fault or no-fault conditions.

A number of model based FDI techniques (Isermann and Balle, 1997; Nyberg and Nielsen, 1997; and Nyberg and Perkovic, 1998) for automotive engines was previously investigated. Nyberg and Nielsen (1997) developed a fault diagnosis system based on non-linear semi-physical model and used a combination of different residual generation methods. It was capable of detecting and isolating faults in throttle actuator, throttle sensor, air mass flow sensor and manifold pressure sensor. A combination of different residual generation methods was used. This had an advantage of fewer model constraints which made it possible to use less restrictive model building and this led to better diagnostic performance. The scheme was experimentally validated on a real production engine. Nyberg and Perkovic (1998) investigated leaks in the air path of a turbo charged SI engine. Estimation of leakage area before and after the throttle could be found. The control algorithm could be reconfigured to suppress the effects of the leakage on emissions. Leaks of 2 mm diameters could be found and their method of diagnosis could be formulated as recursive least squares (RLS) algorithm and therefore it was suitable for on-line implementation.

Gertler *et al., (1991, 1993* and *1995)* and his group were involved in a project with General Motors during 1991-93 and a simulation study was done. A linear model was not found sufficient because of its limited operating range and then a hybrid model with linear core was developed. Five residuals in the form of parity equations were used to diagnose faults in throttle angle sensor, EGR-valve, fuel injectors, manifold pressure sensor, engine speed sensor and lambda sensor. These were also input to the diagnosis system. The residual structure was able to distinguish all faults except fuel injector and lambda sensor faults. The problem was that the characteristics of the lambda sensor were unknown. During 1994-95, they tried this scheme on a real 3.1

litre V6 engine in a production vehicle. Both off-line and on-line versions were developed and they stressed the importance of simple algorithms because of limited computing power available on-board. Instead of using five linear parity equations, they used six non-linear parity equations as residuals. The same faults as before were diagnosed and the new residual structure provided isolation between all faults. To increase robustness, the residuals are ow-pass filtered and threshold crossings were counted. It was reported that they were able to diagnose faults of 10% size. This approach cannot detect faults below the size of 10% with satisfactory performance and therefore, s not suitable for smaller faults.

Rizzoni *et al.*, (1989 and 1991) used an extended version of the detection filter derived from a fourth order linear state space engine model. The work was partially supported by Ford. The diagnosis system measured throttle angle, manifold pressure and engine speed to produce two residuals. The scheme was shown to be able to diagnose 10% faults in throttle angle sensor and manifold pressure sensor on a real Ford 3.0 litre engine. The use of a linear model restricted the operating range of the model and data was shown for 50-60 kPa and 1050-1130 rpm. It seemed like the residuals were sensitive to engine transients. Rizzoni *et al.*, (1993) used a non-linear model to generate five residuals based on parity equations. The throttle angle, engine speed, manifold pressure and injected fuel were measured to diagnose faults in a real 4 cylinder 1.3 litre engine. The diagnosed components were throttle angle sensor, engine speed sensor, and manifold pressure sensor and fuel injectors. The load was decoupled. The plots of the residuals were shown for the case of 10% faults of the diagnosed components. Later in year 1994, Krishnaswami *et al.*, (1994); and Luh and Rizzoni (1994) started using a non-linear discrete Narmax model which was a linear combination of second order polynomials, i.e. it was linear in the parameters. Inputs to the diagnosis system were demanded throttle angle, injected fuel and measured air mass flow and engine speed. Forward and reverse models were used to generate four corresponding parity equation residuals. The load is decoupled. The scheme was tested on a real Ford 3.0 litre engine over a standardised test schedule and was reported to be able to diagnose 10% faults in the air mass flow sensor, 20% faults in the engine speed sensor, 15% faults in the throttle actuator and 40% faults in the fuel injector. 40% faults in the fuel injector is a

very big fault. The fault sizes are very big for other faults also and therefore are easy to diagnose. This FDI scheme may not be suitable for smaller faults.

Observer based non-linear estimation was investigated by Yong-Wha (1998). The focus is on the use of physical models to estimate unmeasured or immeasurable variables and parameters to be used for control and diagnostic purposes. In the non-linear parity equation residual generation (NPERG) diagnostic approach, sensor faults were detected and isolated using output estimators, while input and plant parameter faults were isolated using input estimators. In the case of sensor faults, the information supplied by the faulty sensor to the control module can be decoupled by removing the faulty sensor and replacing the information with estimates obtained through reliable measurements and models. If the system is not observable after removing the faulty sensor, a model of the system can be used to estimate the faulty sensor reading and used as an observer. This is a good approach for sensor fault accommodation but it may not be successful for small bias faults in the sensors. In chapter 8 of this thesis similar fault accommodation scheme for sensor faults is investigated using predictive neural networks and has successfully demonstrated accommodation of 10% sensor bias faults with less than 5% mean accommodation error.

C. Structured Hypothesis Method

Real time supervision using production sensors and additional sensors installed was investigated by Nyberg and Stutte, (2004) and they proposed a new method for adaptive thresholds to handle modelling errors. It was realised that the proposed method for normalisation of the test quantities significantly increased the diagnosis performance. The diagnosis system was constructed within the framework of structured hypothesis tests. The hypothesis tests used were binary i.e. the task was to test one null hypothesis against one alternative hypothesis. The car used for experimentation was E-class Mercedes-Benz with 2.21 litre diesel engine. Two sensor and two component faults were considered. The sensor faults were simulated in Matlab from real no fault data while component faults were implemented in the car. The FDI scheme was successful for a small set of faults and the intensity of the fault cannot be diagnosed.

Structured hypothesis based on statistical hypothesis tests (SHT) (Nyberg, 1999a, 1999b, 2000, 2001 and Nyberg and Krysander, 2003) was also mainly investigated by Nyberg in earlier studies. Nyberg (1999a and 1999b) designed a complete diagnosis system for the air-intake system of an SI engine. The complete design procedure was discussed which included design of test quantities and selection and tuning of the hypothesis tests. The diagnosis system was then experimentally validated using a real engine. The method was applied to the diagnosis of sensor faults and leakage in the air intake system of an SI engine. Two previous methods presented in Nyberg and Nielsen (1997), and Nyberg and Perkovic (1998), solved these two diagnosis problems by two different methods. However, each of these methods was dedicated to a certain type of faults and the two methods could not easily be combined into one single system capable of diagnosing both leaks and sensor faults.

Nyberg (2000) investigated two different methods for diagnosing leakage in the air path of an automotive engine. The first method was based on comparison between the measured and estimated air flow and the other method was based on an estimation of the leakage area. Theoretically and experimentally it was found that the principle based on estimated leakage area gives better results if only leakage is considered. If other faults also needed to be diagnosed then the principle based on comparison of estimated and measured air flow performed better. According to Nyberg (2001), the fault models are powerful tools for handling all types of faults. Structured hypothesis test (SHT) framework can diagnose faults that are modelled as deviations in constant parameters, arbitrary signals, abrupt signals, a change in signal variance and also their combinations. Fault models increase possibility to isolate different faults. For instance, by knowing that two different faults are acting in a different way, it can distinguish between the faults even though they are acting on the same component. Nyberg and Krysander (2003) combined AI, FDI and statistical hypothesis testing (SHT) in a framework for diagnosis. The isolation mechanism was based on AI and thus could handle multiple faults simultaneously using symptomatic logic. It was shown that standard FDI methods such as residuals based on parity relations or observers could be used within the framework. The diagnostic tests used were statistical hypothesis testing and therefore could handle noisy data easily which was an advantage as compared to structured residuals. In nut

shell, the framework presented could efficiently handle fault models, several different fault types, more than two behavioural modes per component, general differential-algebraic models, noise, uncertainties, decoupling of disturbances, static and dynamic systems and isolation of multiple faults.

D. Principal Component Analysis Method

Principal component analysis (PCA) was investigated by Gomm *et al.*, (2000) with practical application to a real industrial nuclear fuel-processing plan. A simulated chemical process was also used to assist the development of PCA technique. Industrial plants often have many process variable measurements available, which can be used as neural network inputs for training and fault classification. Using PCA the number of neural network parameters can be reduced up to 50 percent with a very little sacrifice in the performance of the classifier. Results were presented to illustrate the performance of the developed scheme on applications to the simulated and the real industrial data.

E. Wavelet Networks Method

Wavelet networks are a class of neural networks consisting of wavelets. A new notion of wavelet network was proposed as an alternative to feed forward neural networks for approximating arbitrary nonlinear functions by Zhang and Benveniste, (1992). It gave a new method for identification of general nonlinear static systems from input/output observations by cascading an affine transform and a multidimensional wavelet. These affine transforms and the synaptic weights were identified from possibly noisy input/output data. This new type of network was called wavelet network. It was inspired by both the feed-forward neural networks and wavelet decompositions. An algorithm of back-propagation type was proposed and experimental results were reported. This approach is efficient and is suitable where other more traditional methods are not able to generate residuals for FDI. Wavelet networks have application in very complex and highly non-linear systems.

Algorithms for wavelet network construction were proposed by Zhang, (1997) for the purpose of nonparametric regression estimation. The proposed algorithms were more constructive as compared to back propagation algorithm, in the sense that they automatically determine the network size and estimate the network coefficient in a reasonable number of iterations. More attention was paid to sparseness of training data so that problem of large dimension could be better handled. Wavelet networks can also be trained by back-propagation algorithms because of its neural network like structure. Author used Radial wavelets instead of tensor product wavelets because they could generate single scaling wavelet frames and it weekly depended on its dimension. Due to the radial structure of the wave lets it can also be considered as RBF network. The difference between the two is that the former originates from a wavelet frame with multi-scale structure.

F. Other untraditional methods

Alternative to model-based FDI for automotive engines have also been researched. Barigozzi *et al.*, (2004) developed a probabilistic approach for complex systems made up of a large number of components and demonstrated the method for diagnosis of an engine throttle body and angular sensors measuring the throttle plate angle. A probabilistic approach to fault diagnosis of industrial automobile applications specifically the diagnosis of the throttle body and the angular sensors measuring the throttle plate angle was described. The sensors, actuators and diagnostic tests were described as stochastic finite-state machines (FSM). The diagnosis system was assumed to be composed by apparatuses and tests. Apparatuses were all the system components which could be subject to fault and tests were sources of information which could be used to monitor the system. Transitions between states were probabilistic and forced by events which described either the occurrence of faults or normal working condition. In the composition of FSM models it is possible force the elimination of composite states with negligible probability of occurrence and to normalise the reduced model for further compositions which would lead to a significant reduction in the size of the diagnostic system at the cost of an approximate solution.

Capriglione *et al.*, (2004), presented an analytical redundancy based procedure designed for the on-board real-time fault detection, isolation and accommodation (IFDIA) of sensors typically mounted in public transportation vehicles. The IFDIA procedure essentially had five steps: data acquisition, measurement, residual generation, fault detection and isolation and fault accommodation. Fault detection and isolation step followed some IF THEN rules. The accommodation step accommodated faults by substituting the faulty sensor output with the expected (calculated or predicted) output. Implementation of the procedure on a PC required 238 ms to detect and isolate a sensor fault and further 240 ms for accommodation. This performance was better than required by the system dynamics and therefore was found suitable for on-board operation.

According to Yang *et al.*, (2001), faults relating to gas pressure in the cylinder of a diesel engine could be diagnosed by the use of an instantaneous angular speed fluctuation ratio (IASFR) of the engine. It was realised that direct measurement of the gas pressure in the cylinder was impractical because installing pressure transducers inside the cylinder was generally difficult and uneconomical for the practical use. In contrast, instantaneous angular speed measurement of the diesel engine is very convenient, economical and reliable. The instantaneous angular speed waveforms both in the fuel leakage condition and in normal condition were measured under various engine speeds and loads in laboratory condition. The characteristic parameters for detecting the faults relating to the gas pressure in the cylinder were obtained successfully.

2.3.2 Model-Free Technique

The fault diagnosis techniques which have been reviewed above required a plant model. However, some plants may be too complex to efficiently derive a plant model. A model-free technique is then required. There are several model-free techniques reported (Gertler, 1998). Two of the techniques are physical redundancy and spectrum analysis. A brief description of these techniques is presented as follows:

A. Physical Redundancy

This approach can be used to detect sensor faults. To carry out the diagnostic process, multiple sensors are installed to measure the same physical quantity. Any serious discrepancy between the measurements indicates a fault in the plant. If only two sensors are used, fault isolation is not possible. If three sensors are used, a voting scheme can be used to isolate the faulty sensor. Nyberg and Stutte (2004) installed additional pressure and temperature sensors and a turbocharger speed sensor in a Mercedes-Benz car for the experimental setup of an FDI scheme based on framework of structured hypothesis tests. However, physical redundancy involves additional hardware cost and therefore has a very limited application. But on the contrary, analytical redundancy techniques are widely used in the industry because of low costs.

B. Spectrum Analysis

Spectrum analysis of plant measurements may be used for fault detection. Most plant variables exhibit a typical frequency spectrum under normal operating conditions. Any deviation from this normal condition is an indication of abnormality.

Shoji *et al.,* (2002) proposed fault diagnosis system based on the spectrum of vibrations or sounds obtained from an operating machine, because the time series data of vibrations or sounds are complicated and include noise. The difference between normal and abnormal data becomes clearer comparing time series data. It is suitable for the detection of the fault by utilizing changes of spectral data. Using this method, it is shown that it can even detect unknown fault patterns. The authors have used neural networks for fault pattern recognition. The authors investigated this FDI method for electromagnetic valve and a wood slicing machine. This method may not be useful for more complicated systems like automotive engines where noise levels of vibration and sound can be very high and in a complex machine there can be hundreds of possibilities of abnormal vibration and it may not be possible to train a neural network for such a large data.

Toshiyuki *et al.*, (2000) proposed two new fault diagnosis systems in which one diagnoses a fault based on behaviour of the object system, and other diagnoses a fault based on power spectrum of the object system. In the latter, neural network learns power spectrum of both the normal and fault states for the object. The authors verifier the effectiveness of the developed FDI system on simulations. This technique is quite similar to the technique developed by Shoji *et al.*, (2002) and has the same limitation of handling bulky data. Both these techniques may be suitable for off-line application but are not suitable for real time application.

2.3.3 FDI Using Neural Networks

The good classification properties of neural networks for engine FDI are well documented by Principe *et al.*, (2000). Antory (2007), investigated application of data driven monitoring technique to diagnose air leaks in an automotive diesel engine. The model was derived solely from the measurement signals because there is no requirement to make any simplifying assumptions to build the model. The interdependency of the original signals was captured and transformed into a new and smaller number of independent signals. The remaining (not captured) signals contained mainly uninformative and noisy data. It was shown that diagnostic PCA model performed better in comparison to a physical model when detecting air leaks at intake manifold chamber especially for a small diameter of air leak. Another advantage of this diagnosis model is that it can be used to detect any type of fault in a similar manner to the air leakage fault. The effectiveness of the model was validated using an experimental automotive 1.9L four-cylinder diesel engine test-bed. Small air leaks in the inlet manifold with a diameter size of 2-6mm were accurately detected.

Neural network based models have a simple structure and a good generalisation capabilities. Tan and Saif (2000) investigated external recurrent neural networks to identify the nonlinear dynamic model of the intake manifold and the throttle body process in an automotive engine. Levenberg – Marquardt algorithm was applied for weight estimation in this study. It was realised that neural network based modelling of intake manifold is comparable if not better than the first principle based models. It was

concluded that the neural network based models can capture the inherent nonlinearity and the dynamics of the manifold pressure as well as the mass flow processes in automotive engines. The experimental results presented in the paper show that the NN based models are suitable for modelling for the purpose of control and on-board diagnosis of automotive engines.

The pattern recognition and classification abilities of neural networks are applied to crankshaft speed fluctuation data for engine fault diagnosis and multidimensional mapping capabilities were investigated as an alternative to large lookup-tables and calibration functions by Shayler *et al.*, (2000). The possibility of replacing specific sections of engine control strategy software with neural network based systems was also investigated. In each investigation the neural networks used had multi-layer perceptron (MLP) architecture and had been trained with back-propagation algorithm. The pattern recognition and classification abilities of neural networks were applied to crankshaft speed fluctuation data for engine fault diagnosis.

Manzie *et al.*, (2001) proposed a radial basis function (RBF) based approach for the fuel injection control of an automotive engine. On line learning was achieved using gradient descent updates method the proposed approach requires no prior knowledge of engine system. The RBF network was implemented on a four-cylinder engine and outperforms a production engine control unit. Thus, it has a possibility to use an RBF network in place of the lookup tables currently used in fuel injection systems to estimate airflow into the cylinder. The proposed scheme was evaluated on a dynamometer test rig. The engine used was a four-stroke, 2.41 Mitubishi Magna TE engine with no EGR. The RBF controller was implemented in Labview on a standard PC equipped with data acquisition card NIDAQ PC-LPM 16.

Jakubek and Strasser (2002) presented a new training algorithm for ellipsoidal basis function networks to use as few basis functions as possible to reduce the amount of necessary model parameters. This was accomplished by adapting the spread parameters using Taylor's theorem. The initial problem was that a large number of measurements were to be monitored with as few parameters as possible with a reasonable computational effort which was sorted out with principle component analysis (PCA). The resultant network used significantly less number of basis

functions than an RBF network of the same accuracy. Application to measured data from a real automotive process shows that the proposed algorithm achieved good results.

Kimmich *et al.*, (2005) contributed and showed a systematic development of fault detection and diagnosis methods for intake system and injection system along with combustion process of a diesel engine. Black box modelling with the special local linear neural network, comprising of both automatic model structure generation and identification of its parameters was used. The five different identified reference models calculating special features were used to set up five independent parity equations which yielded to the residuals. The signal processing and the model based fault detecting algorithms were implemented on a dSPACE rapid control prototyping system (Hardware and Software) in Matlab/Simulink. Experimental investigations presented were performed on Opel four cylinder diesel engine test bench and in vehicle. Experiments on a dynamic test bench demonstrated the detection and diagnosis of different process faults under real time conditions.

Capriglione *et al.*, (2003) used multilayer perceptron artificial neural networks (ANNs) with the back-propagation training algorithm for fault diagnosis and accommodation in automotive engines. It was felt impracticable to produce real faults during engine operation and therefore a suitable set of possible faults were simulated. Many different ANN architectures were tried for residual generation. The learning set of each ANN was constituted by 3300 real samples acquired in different fault free operating conditions. The networks were trained by using Levenberg Marquardt back-propagation algorithm. On the basis of data acquired in fault-free conditions, the diagnostic performances in an on-line condition were verified. The fault accommodation showed a good performance with maximum error of 5%. The applicability of ANN FDI to on-board engine control units was also demonstrated.

Capriglione *et al.*, 2007 presented the implementation of instrument fault detection, isolation and accommodation system developed for real-time automotive applications for a Fiat 1.2L SI engine and applied to the main engine-operating control sensors. The system was able to identify and accommodate different kind of faults e.g. short circuit, open circuit, no

calibration (bias) and hold (constant reading). Analytical redundancy based systems are more useful for automotive applications because physical redundancy based systems require triplication of sensor leads which increase costs and volume. The accommodation accuracy was not very good but could be accepted as they were compatible with the specific application. There is a possibility of programming a digital signal processor (DSP) with a neural network fault diagnosis system and embedding it in the electronic control unit of a car. The ECU would then be able to classify and isolate all sorts of faults, which were considered for neural network training.

Isermann (2005) proposed three different detection modules to generate symptoms mainly based on production type sensors. The symptoms were generated with non-linear output error and input error parity equations for special model-based characteristic quantities like volumetric efficiency, oscillations of pressure, flow and angular speed and oxygen content. The generation of about twenty symptoms then allowed an in-depth fault diagnosis using fuzzy logic interface scheme.

Vinsonneau et al., (2001) investigated fault diagnosis strategy for a model of Jaguar car engine. Real engine data was used to model manifold dynamics over several engine speeds and operation conditions. A number of fault scenarios were considered. A nonlinear observer method was used to detect both additive and multiplicative sensor faults. Fault isolation logic was also designed. Isermann and Schwarte (2004) developed a fault diagnosis method for the intake system of a diesel engine. Signal models and parity equations residuals were generated by applying semi-physical dynamic process model identification with local linear dynamic neural networks. These residuals led to the symptoms which were the basis for the diagnosis of several faults. All the experiments were performed on a dynamic test bed for 2.01 diesel engine. The diagnosis scheme demonstrated the detection and diagnosis of several implemented faults in real time with reasonable calculation effort. The calculation effort was an important aspect because of limited processing capability available on-board. Tawel *et al.*, (1998) presented neural networks as a means of creating control and diagnostic strategies that help in meeting mandatory government requirements on emission control efficiently and robustly. This research described a VLSI design that permits neural networks to execute in real time and do misfire detection on-board.

Shiraishi et al., (1995) presented a new automotive fuel-injection controller using the cerebellar model articulation controller (CMAC) neural network and implemented to maintain air-to-fuel ration at its stoichiometric value. CMAC neural network for fuel-injection control was demonstrated on a research automobile. The CMAC's fast computation is a distinct advantage for control within the time span of an engine revolution. It required almost no prior knowledge of the engine dynamics and modelled the characteristics of the engine very quickly in real time. It is also adaptive and is effective in dealing with the inherent time delays in the oxygen sensor.

Leonhardt et al., (1995) presented two methods of real time supervision of injection and combustion in turbocharged diesel engines. Cylinder pressure and crankshaft speed contain hidden information about internal motor conditions. A real time data reduction method was implemented to generate characteristic fault symptoms from this information and which may further be processed by approximation or classification algorithms utilising ANNs. This approach is validated by simulation and experimental data obtained from a 4-cylinder 1.6 litre diesel engine.

According to Evans-Pughe (2006), major car firms are looking for neural networks to solve the demanding engine control and diagnosis requirements mandated by the government in order to control emissions. Ford has introduced the Econoline van which uses a neural network based misfire detection algorithm in its V10 engine. Chrysler is working on developing a neural network based method for controlling variable valve timing in next generation fuel efficient engines. General Motors recently produced a paper with UK based neural network chip firm Axeon on a sensor-replacement application.

While MLP networks have been utilised for fault diagnosis in automotive engines to some extent but the other architectures like RBF have not. It was investigated and found that RBF networks have better classification capabilities than MLP networks (Sangha *et al.*, 2004a, 2005a, 2005b, Gomm *et al.*, 2000). In previous study in MSc thesis (Sangha, 2004b), both MLP and RBF networks were investigated for steady state fault diagnosis. The steady state data for no fault state and all fault states was collected and normalised.

Some data was utilised to train the neural network and some for testing. During the static fault diagnosis, Levenberg-Marquardt algorithm is used when back propagation algorithm acts very slowly for MLP training. "The training time of an MLP using back-propagation (BP) was experimentally determined to increase exponentially with the size of the problem; that is, although the required number of patterns increases only linearly with the number of weights, the training time of larger networks seems to scale exponentially to their size. This indicates that there are problems that cannot be solved practically with MLPs trained with BP," (Principe *et al.*, 2000). The steady state fault diagnosis was successfully demonstrated with good results and it was found that RBF out performed MLP for engine air path faults classification.

In the existing researches, little attention has been given to the problem of diagnosing faults with different intensities. The field of on-board engine fault diagnosis using artificial neural networks is still not fully explored and has a lot of potential for further research. This work has been carried out in this PhD study as mentioned in the research aims and objectives.

2.4 SUMMARY

After discussing basic FDI concepts, general FDI methods are reviewed in depth which included classical limit alarm system, statistical process control approach, AI and knowledge based systems and general model based methods.

Model based FDI methods are most extensively used in different industries and the use of AI is increasing day by day. The major car firms are looking for neural networks based solutions for FDI and accommodation as car engine dynamics are severely nonlinear and multivariable. An exhaustive literature survey on different engine FDI methods is also conducted. It included parameter estimation, observer based methods, structured hypothesis, principle component analysis, wavelet networks, and other untraditional FDI methods like probabilistic approach and IASFR. A short introduction is given to model free techniques e.g. physical redundancy and

spectrum analysis. Lastly an automotive engine FDI exploiting neural networks are reviewed.

It can be seen from the review that though neural networks have been used for modelling and classification of engines for their condition monitoring and FDI, the disturbance effects are seldom considered and addressed. These effects significantly affect the correctness and accuracy of FDI of industrial systems. In this PhD project, it is proposed to on-line adapt the neural network classifier so that the effects of disturbance can be eliminated to a maximal possible amount. Thus, the reliability and applicability of the developed FDI method is greatly enhanced.

CHAPTER 3

MEAN VALUE ENGINE MODEL

3.1 INTRODUCTION

In this chapter, a brief introduction to the working of different types of internal combustion (IC) engines is given along with a brief history of engine development.

A well-known engine benchmark mean value engine model (MVEM) is used throughout this research project. This engine model is available in Matlab/Simulink format and is widely used by the researchers and industrial system developers. The major sub-models and the main system equations of MVEM are discussed in Section 3.4 ahead.

3.2 DIESEL ENGINE SYSTEM

The history of the diesel engine can be traced back over a hundred years to the late 19th century. Rudolf Diesel (1858-1913), is the pioneer who invented the diesel engine and the engine was then named after him. He obtained the first patent number 608845 for the internal combustion diesel engine (California Energy Commission, 2006) in 1892. The first fully operational diesel engine was introduced in 1897. Since then, it has been the most widely used industrial source of power. His original concept allows a maximum amount of work to be obtained from a given heat source which has yet to be

improved commercially. The diesel engine is a spark-less compression-ignition engine, which transforms the energy stored in fuel into motion and useful work. The compression-ignition principle employed by the diesel engine distinguishes it from the petrol engine which uses a spark-ignition (SI) principle. In diesel engines, the air is compressed and the fuel is then injected at an appropriate point in the cycle once the air is well above the ignition temperature. In the SI engine, a spark plug is used to ignite the fuel mixture.

In general, diesel engines can be classified into two categories, two-stroke and four-stroke engines. In the two-stroke engine, combustion occurs in the region of top dead centre (TDC) of every revolution and gas exchange at every revolution at bottom dead centre (BDC). TDC is the position where the cylinder reaches its maximum upward movement and vice versa for the BDC. In the four-stroke engine, there are four cycles namely, intake, compression, expansion, and exhaust. The combustion only occurs once every two revolutions. The main advantage of the four-stroke cycle is that it provides a longer period for the gas exchange process which results in purer trapped charge. This allows time for sufficient fresh air to be drawn into the combustion chamber to mix with the diesel and the exhaust can be drawn out of the combustion chamber. It also lowers the thermal loading associated with engine internal components like pistons, cylinder heads and liners. The application of the two-stroke engine can be seen in marines and stationary applications while four-stroke engines are used in the majority of other applications.

Today's diesels inject fuel directly into an engine's cylinders using tiny computers to deliver precisely the right amount of fuel the instant it is needed (Bosch, 2005). All functions in a modern diesel engine are controlled by an electronic control module that communicates with an elaborate array of sensors placed at strategic locations throughout the engine to monitor everything from engine speed to coolant and oil temperatures and even piston position. Tight electronic control means that fuel burns more thoroughly, delivering more power, greater fuel economy, and fewer emissions than yesterday's diesel engines could achieve. Modern direct-injection diesel engines produce low amounts of carbon dioxide, carbon monoxide, and unburned hydrocarbons etc.

3.3 PETROL/GASOLINE ENGINE SYSTEM

A petrol engine (UK) or a gasoline engine (US) can be classified as one of the following types:

- Four-stroke cycle engine
- Two-stroke cycle engine
- Wankel engine

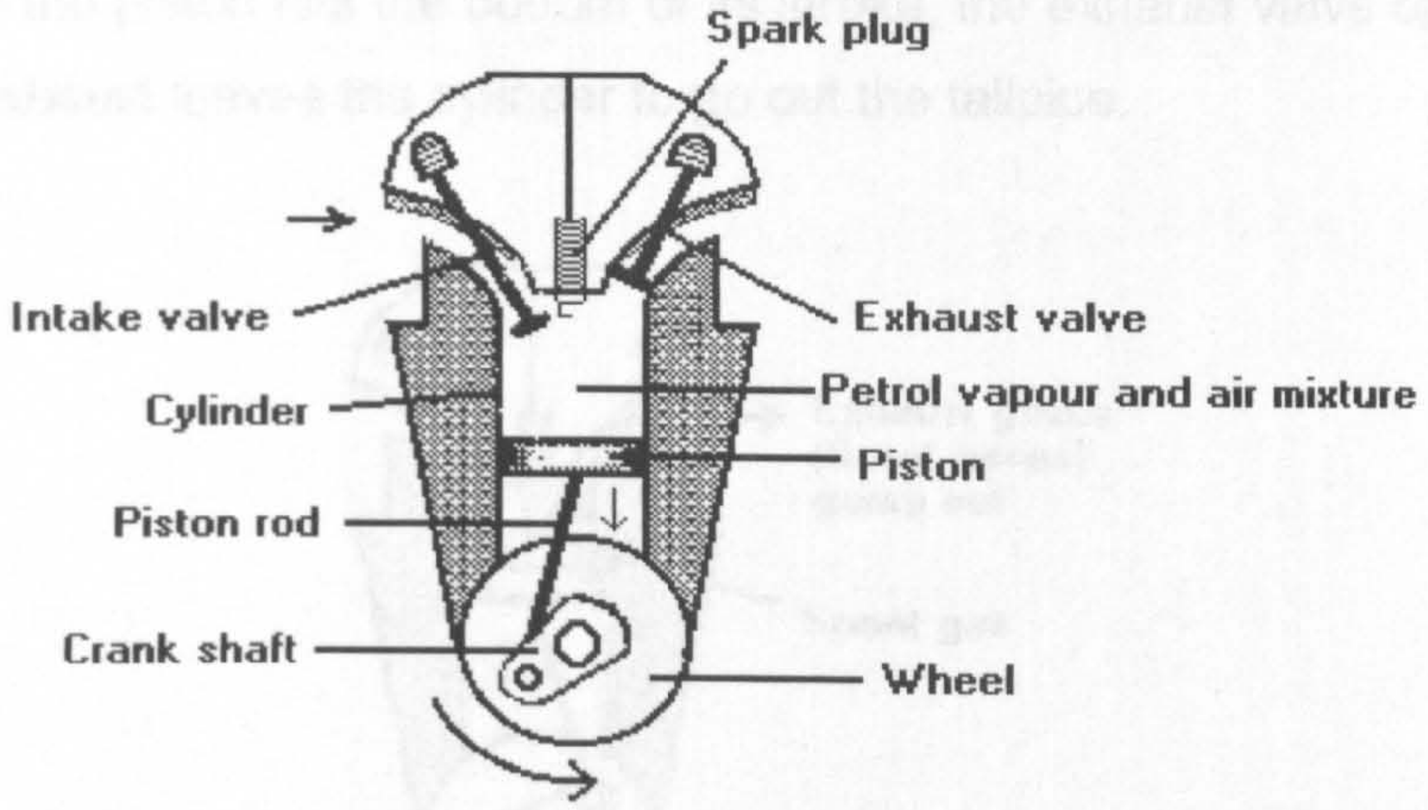
The internal combustion engine is an engine in which the combustion, or rapid oxidation, of gas and air occurs in a confined space called a combustion chamber. This exothermic reaction of a fuel with an oxidizer creates gases of high temperature and pressure, which are permitted to expand. The defining feature of an internal combustion engine is that useful work is performed by the expanding hot gases acting directly to cause pressure, further causing movement of the piston inside the cylinder. For example by acting on pistons, rotors, or even by pressing on and moving the entire engine itself. This contrasts with external combustion engines, such as steam engines and Stirling engines, which use an external combustion chamber to heat a separate working fluid, which then in turn work by moving a piston.

The term Internal Combustion (IC) engine is almost always used to refer specifically to reciprocating engines, Wankel engines and similar designs in which combustion is intermittent. However, continuous combustion engines, such as jet engines, most rockets and many gas turbines are also internal combustion engines. All internal combustion engines depend on the exothermic chemical process of combustion; the reaction of a fuel typically with air, although other oxidisers such as nitrous oxide may be employed (Bell, 2004).

The most common modern fuels are made up of hydrocarbons and are derived from mostly petroleum. These include the fuels known as diesel fuel, gasoline and petroleum gas, and the rarer use of propane gas. Most internal combustion engines designed for gasoline can run on natural gas or liquefied petroleum gases without major modifications except for the fuel delivery components. Liquid and gaseous bio-fuels, such as Ethanol and bio-diesel, a form of diesel fuel that is produced from crops that yield triglycerides such as soy bean oil, can also be used. Some can also run on Hydrogen gas.

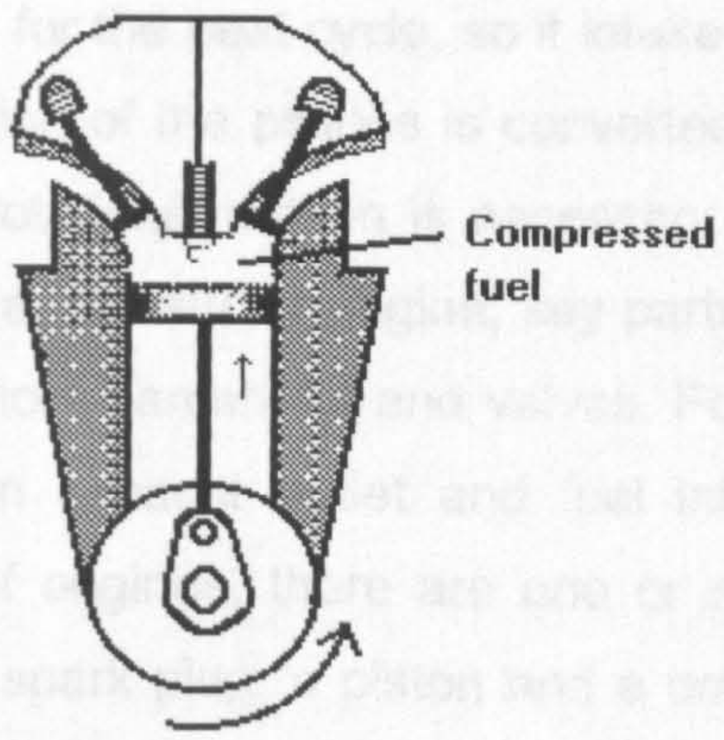
Electrical/Gasoline-type ignition systems (that can also run on other fuels as previously mentioned) generally rely on a combination of a lead-acid battery and an induction coil to provide a high voltage electrical spark to ignite the air-fuel mix in the engine's cylinders. This battery can be recharged during operation using an electricity-generating device, such as an alternator or generator driven by the engine. Gasoline engines (Bosch, 2004) take in a mixture of air and gasoline and compress to less than 170 psi and use a spark plug to ignite the mixture when it is compressed by the piston head in each cylinder.

1. The piston starts at the top, the intake valve opens, and the piston moves down to let the engine take in a cylinder-full of air and gasoline. This is the intake stroke. Only the tiniest drop of gasoline needs to be mixed into the air for this to work.



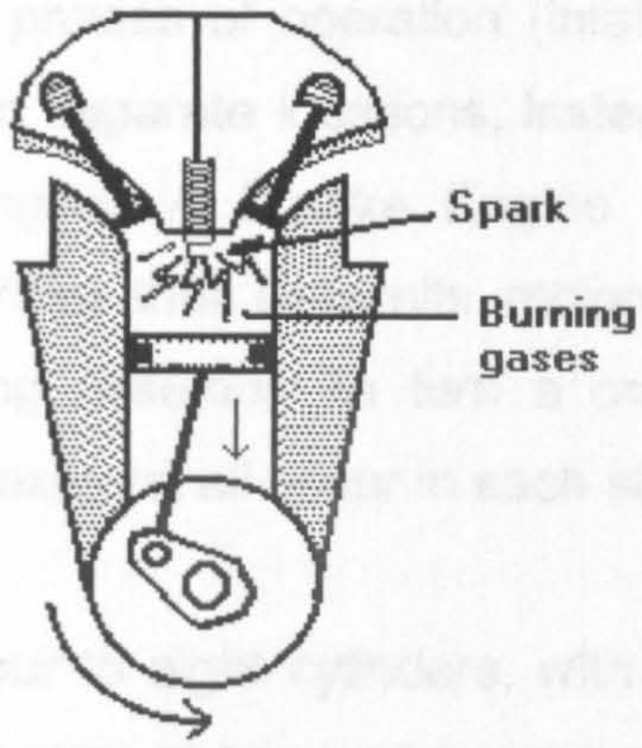
(a)

2. Then the piston moves back up to compress this fuel/air mixture. Compression makes the explosion more powerful.



(b)

3. When the piston reaches the top of its stroke, the spark plug emits a spark to ignite the gasoline. The gasoline charge in the cylinder explodes, driving the piston down.



(c)

4. Once the piston hits the bottom of its stroke, the exhaust valve opens and the exhaust leaves the cylinder to go out the tailpipe.

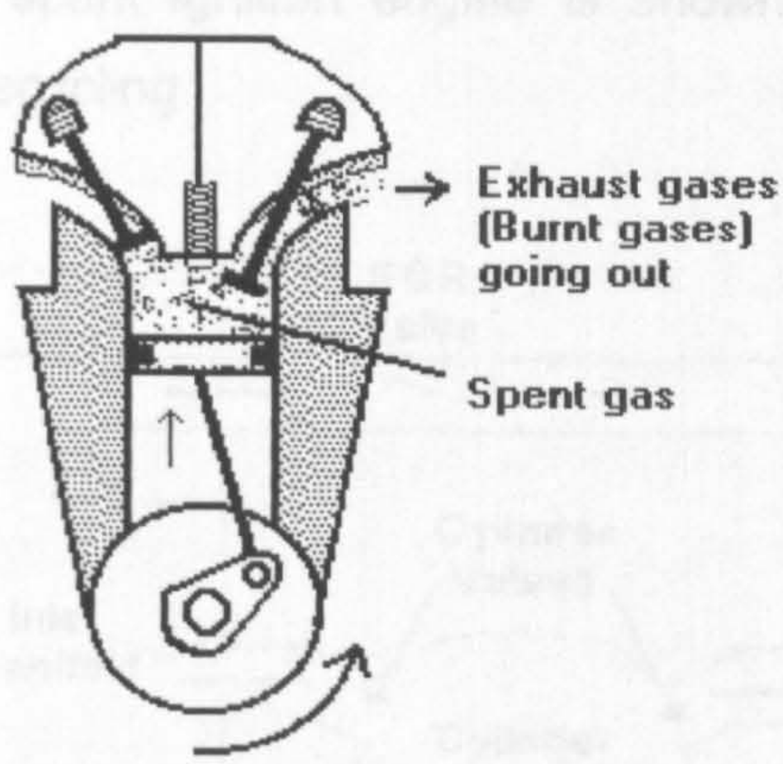


Fig. 3.0: Engine states (a) Suction (b) Compression (c) Explosion (d) Exhaust

Now the engine is ready for the next cycle, so it intakes another charge of air and gas. The linear motion of the pistons is converted into rotational motion by the crankshaft. The rotational motion is necessary because it rotates the car's wheels with it. For a four-stroke engine, key parts of the engine include the crankshaft, one or more camshafts and valves. For a two-stroke engine, there may simply be an exhaust outlet and fuel inlet instead of a valve system. In both types of engines, there are one or more cylinders and for each cylinder there is a spark plug, a piston and a crank. A single sweep of the cylinder by the piston in an upward or downward motion is known as a

stroke and the downward stroke that occurs directly after the air-fuel mix passes from the carburettor to the cylinder, where it is ignited.

A Wankel engine has a triangular rotor that orbits in a chamber around an eccentric shaft. The four phases of operation (intake, compression, power, and exhaust) take place in separate locations, instead of one single location as in a reciprocating engine. A Bourke Engine uses a pair of pistons integrated to a Scotch Yoke that transmits reciprocating force through a specially designed bearing assembly to turn a crank mechanism. Intake, compression, power, and exhaust all occur in each stroke of this yoke.

Most car engines have four to eight cylinders, with some high performance cars having ten, twelve, or even sixteen, and some very small cars and trucks having two or three. In previous years some quite large cars, such as the DKW and Saab 92, had two cylinders, two stroke engines.

Air path for a typical spark ignition engine is shown in Figure 3.1 which includes exhaust gas recycling.

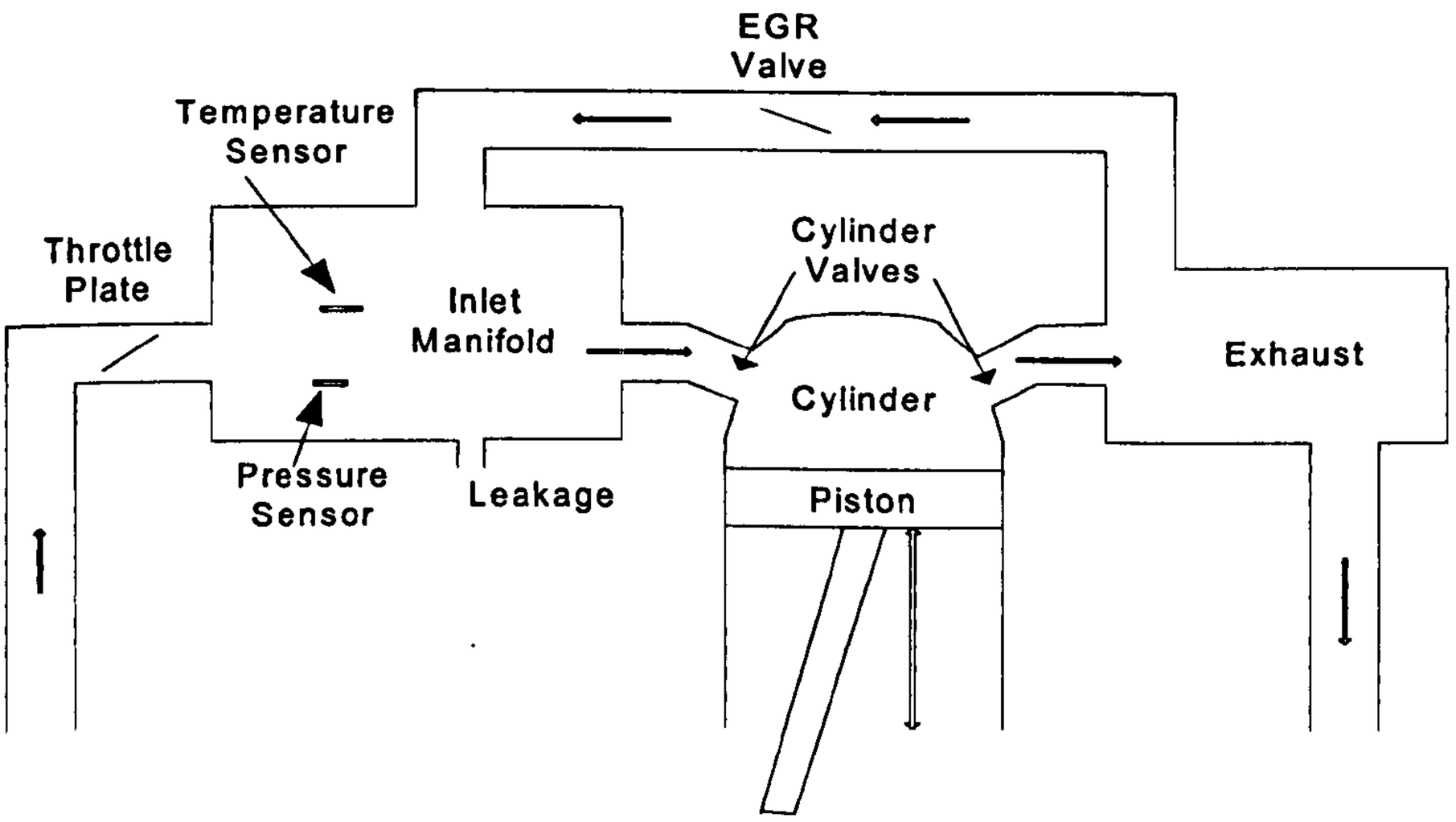


Fig. 3.1: Schematic view of the air intake and exhaust system of an automotive engine

Air flows over the throttle and enters the inlet manifold. Some part of the exhaust is also fed back in the inlet manifold and this is called exhaust gas recycle (EGR). The flow of exhaust gas in the inlet manifold is controlled by

an EGR valve. From the inlet manifold the gas flows into the combustion chamber (cylinder) where the fuel is also injected and the combustion takes place with spark ignition (in case of a gasoline engine). The burnt gases go to the exhaust manifold and then pass over the lambda sensors. Finally the burnt gases are released into the atmosphere from the tail pipe.

3.4 MEAN VALUE ENGINE MODEL (MVEM)

All IC engines contain significant nonlinearities, which dominate their dynamic behaviour. The MVEM is a fairly good approximation of medium speed IC engine dynamics. This model includes the latest results and efficiency enhancement system such as Exhaust Gas Recycle (EGR) unit. MVEM is simplified dynamic engine models, which are based on collections of physical models of the most important engine subsystems. They are intended to operate as predictors of averages rather than the cycle-to-cycle values of the most important engine states and variables. Such models are currently used for a number of purposes in engine control applications such as engine control system design and engine diagnostics (Hendricks, *et al.* 2000). The main focus of this research is on Fault Detection & Isolation (FDI) and not on the MVEM simulation. A MVEM consists of three important subsystems, which describe the behaviour of the fuel mass flow, the intake manifold filling dynamics and the acceleration of the crankshaft. In this chapter, the main system equations and simulink models are briefly explained. The block diagram of MVEM is shown in fig 3.2.

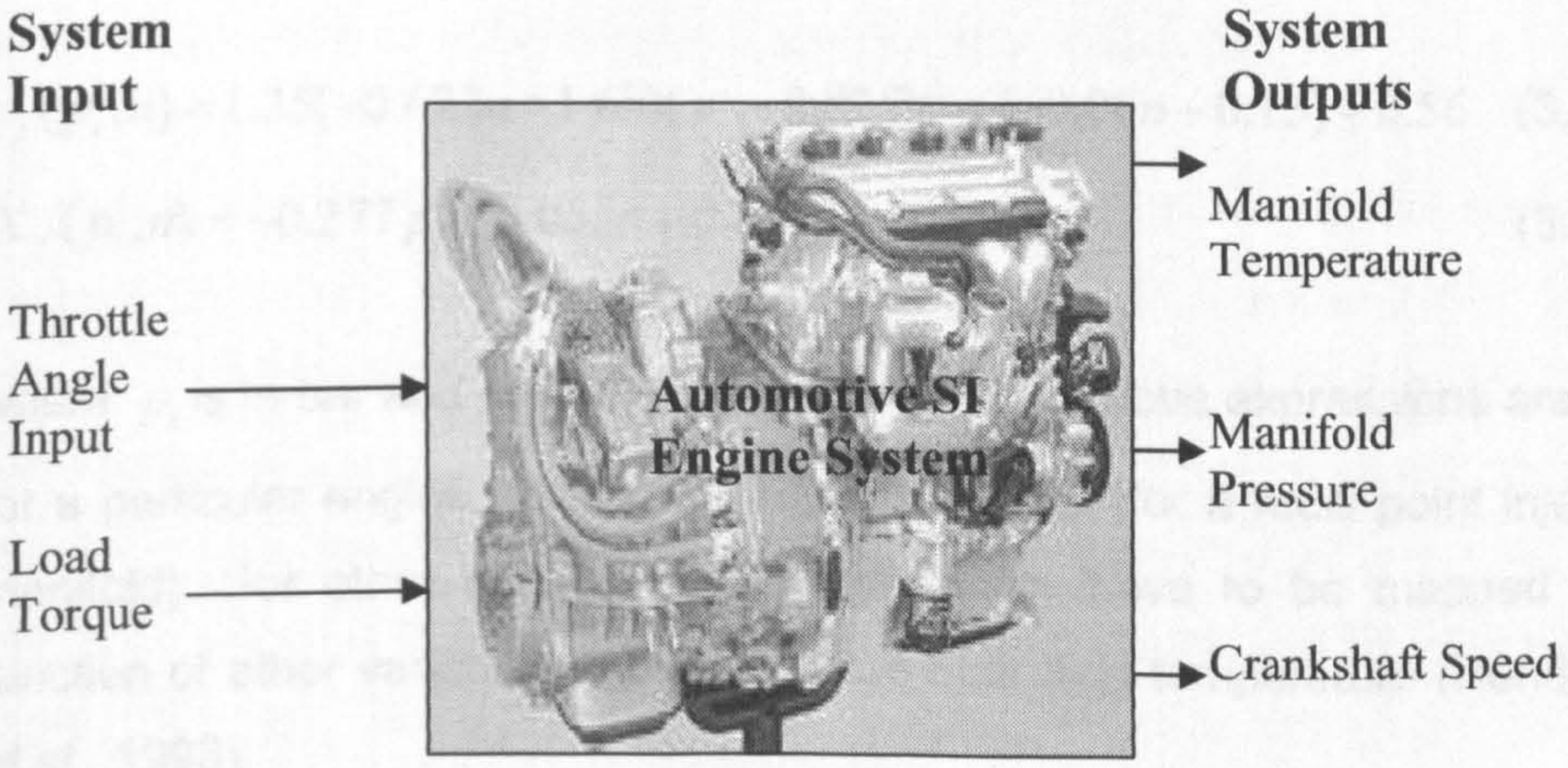


Fig 3.2: Engine block diagram

MVEM has throttle angle as the only input and inlet manifold temperature, pressure and crankshaft speed as the system outputs. MVEM consists of three important subsystems which describe the behaviour of the fuel mass flow, the intake manifold filling dynamics and the acceleration of the crankshaft. These three subsystems are described briefly here in their most common forms. A detailed derivation of all the sub-models of the MVEM can be found in Hendricks *et al.* (1996 and 1993).

3.4.1 Fuel Mass Flow Dynamic Sub-model

The fuel flow dynamics for a multi-point injection engine can be given by the following model:

$$\ddot{m}_{ff} = \frac{1}{\tau_f}(-\dot{m}_{ff} + X_f \dot{m}_{fi}) \tag{3.1}$$

$$\dot{m}_{fv} = (1 - X_f)\dot{m}_{fi} \tag{3.1a}$$

$$\dot{m}_f = \dot{m}_{fv} + \dot{m}_{ff} \tag{3.1b}$$

The model is based on keeping track of the fuel mass flow. The parameters in the model are time constant for fuel evaporation, τ_f and the proportion of the fuel which is deposited on the intake manifold or close to the intake manifold, X_f. These parameters are operating point dependent and thus the model is nonlinear in spite of its linear form (Hendricks, *et al.* 2000). The fuelling model parameters can be approximately expressed in terms of the states of the model as:

$$\tau_f(p_i, n) = 1.35(-0.672n + 1.68)(p_i - 0.825)^2 + (-0.06n + 0.15) + 0.56 \tag{3.2}$$

$$X_f(p_i, n) = -0.277 p_i - 0.055n + 0.68 \tag{3.3}$$

where p_i is in bar and *n* is in krpm (1000 x rpm). Above expressions are only for a particular engine running after fully warm-up (for a multi point injection manifold). For other engines these parameters have to be mapped as a function of other variables and at different operating temperature (Hendricks, *et al.*, 1993).

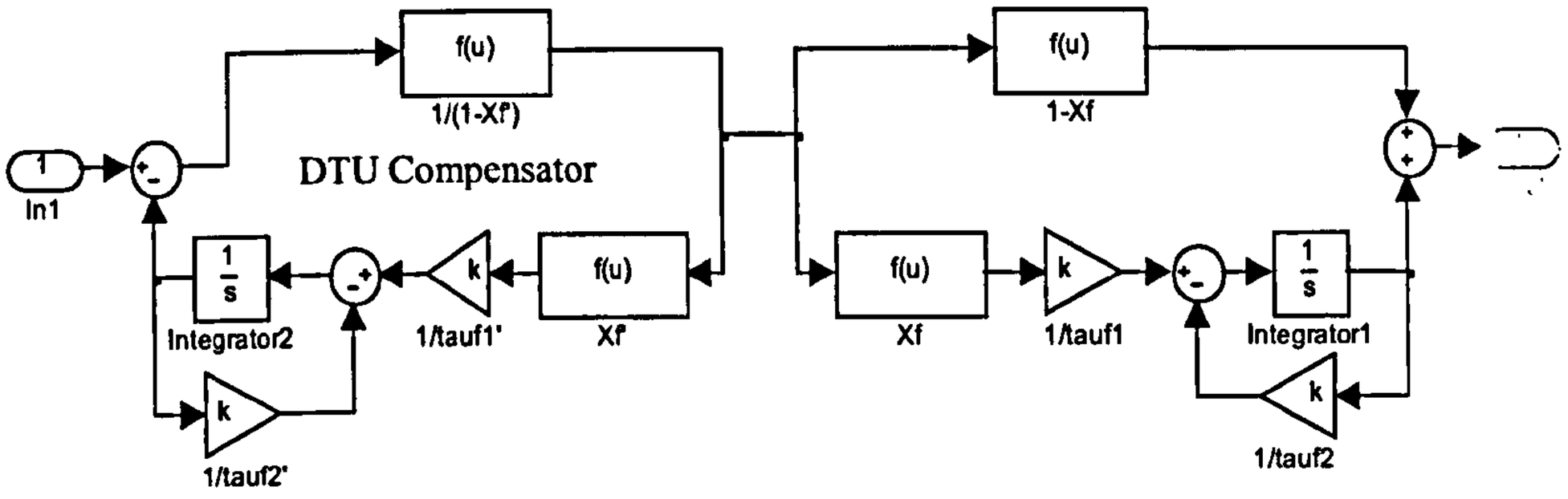


Fig. 3.3: DTU compensator and dynamic fuel flow model

Therefore the above two equations are only given as examples of what can be generally expected. A block diagram of this sub-model is given as Fig. 3.3.

The compensator for DTU fuel flow model can be described as:

$$\ddot{m}_{ff}' = \frac{1}{\tau_f}(-\dot{m}_{ff}' + X_f \dot{m}_{fi}) \tag{3.4}$$

$$\dot{m}_{fi} = \frac{1}{1-X_f}(m_{fu} - \dot{m}_{ff}') \tag{3.5}$$

The fuel flow dynamic equations are usually not used in the engine control system but what is used are compensators based on the models which effectively cancel out the fuelling dynamics. The compensators are inserted in front of the injector driving algorithm and ideally adjust the fuel flow so that the fuelling dynamics do not influence the requested fuel quantity.

The variables in above four equations represent the projected fuel mass flow and the projected fuel puddle mass respectively in the DTU and Aquino models (Aquino, 1981). In general the '*X*' and 'tau' in the models and in the corresponding compensators will not be the same but are intended. Both the compensators are differentiating networks and only have a transient effect on the fuel command signal. The compensators are each unique to the model for which they are derived because the systems are nonlinear. A transient

method of mapping the parameters of the compensators is suggested in (Hendricks, 1993).

3.4.2 Inlet Manifold Filling Dynamics

Assuming the intake manifold temperature to be constant and equal to the ambient and EGR temperature, the manifold filling state equation can be given as

$$\dot{p}_i = \frac{RT_i}{V_i}\left(-\dot{m}_{at} + \dot{m}_{ap} + \dot{m}_{EGR}\right) \tag{3.6}$$

where p_i is absolute manifold pressure (bar), $\dot{m}_{at}$ is air mass flow past throttle plate (kg/sec), $\dot{m}_{ap}$ is air mass flow into intake port (kg/sec), $\dot{m}_{EGR}$ is EGR mass flow (kg/sec), T_i is intake manifold temperature in Kelvin, V_i is (manifold + port passage) volume (m^3) and R is gas constant (287×10^{-5}).

It can be seen from equation (3.6) that change in absolute manifold pressure with respect to time is dependent on three factors i.e. air mass flow past throttle plate $\dot{m}_{at}$, air mass flow into intake port $\dot{m}_{ap}$ and EGR mass flow $\dot{m}_{EGR}$. The temperature is considered constant equal to T_i and V_i and R are also constants. This isothermal assumption is an approximation and may not lead to good results because the EGR temperature can easily approach up to 200 degree centigrade and the quantity of EGR flow can be as high as 25% of the total flow. The manifold filling dynamics in reality is based on as adiabatic operation rather than isothermal. For the best accuracy the above equation can be modified as follows:

$$\dot{p}_i = \frac{\kappa R}{V_i}\left(-\dot{m}_{ap}T_i + \dot{m}_{at}T_a + \dot{m}_{EGR}T_{EGR}\right) \tag{3.7}$$

where T_a is ambient temperature (Kelvin), T_{EGR} EGR temperature in Kelvin and κ is ratio of specific heats which is 1.4 for air. The only problem with adiabatic assumption is that the intake manifold temperature must be known

accurately and instantaneously, whereas the traditional temperature transducers have a time constant of up to three seconds.

The port and throttle air mass flow in equation (3.2) is given by expressions which were derived in Hendricks and Sorenson, (1990) and modified in Hendricks *et al.*, (1996).

$$\dot{m}_{ap}(n, p_i) = \frac{V_d}{120RT_i}(e_v.p_i)n \tag{3.8}$$

where n is engine speed (rpm/1000), e_v is volumetric efficiency based on manifold conditions and V_d is engine displacement (litres). Air mass flow into intake port $\dot{m}_{at}$ is a function of crankshaft speed and absolute manifold pressure as shown in equation (3.8). Where as air mass flow past throttle plate $\dot{m}_{at}$ is a function of throttle plate angle and absolute manifold pressure and can be given as:

$$\dot{m}_{at}(\alpha, p_i) = m_{at1}\frac{p_a}{\sqrt{T_a}}\beta_1(\alpha)\beta_2(p_r) + m_{at0} \tag{3.9}$$

where α is throttle plate angle (degrees), p_a is ambient pressure (bars), p_r is relative pressure and p_c is critical pressure which is a constant, and where

$$\beta_1(\alpha) = 1 - \cos(\alpha) - \frac{\alpha_0^2}{2!} \tag{3.10}$$

and

$$p_r = \frac{p_i}{p_a} \tag{3.11}$$

and

m_{at0}, m_{at1}, α_0 *and* p_c are also constants.

$$\beta_2(p_r) = \begin{cases} \sqrt{1 - \left(\frac{p_r - p_c}{1 - p_c} - \right)^2}, & if\ p_r \geq p_c \\ 1 & if\ p_r < p_c \end{cases} \tag{3.12}$$

In the throttle mass flow equation (3.9) the expression which serves in place of the isentropic flow equation has been derived on the basis of a two flow path model presented in Hendricks *et al.*, (1996) and further extended in work presented in Chevalier *et al.*, (2000). Equation (3.12) is a compact approximation of this theory which is useful for calculations. The correct expression, containing more details can be found in Chevalier *et al.*, (2000). In particular equation (3.12) is not correct in the presence of heavy manifold pumping fluctuations and for about $p_r > 0.98$. Equation (3.12) often has to be modified because it is common that original equipment manufacturers often use large throttle body openings to avoid even small pressure drops across the throttle body. This changes the effective critical pressure p_c in equation (3.12).

The sub-model for throttle plate opening angle is presented in Fig. 3.4. It can be seen that the port air mass flow provides the feedback around the integrator and it is thus is responsible for the effective time constant of the sub-system.

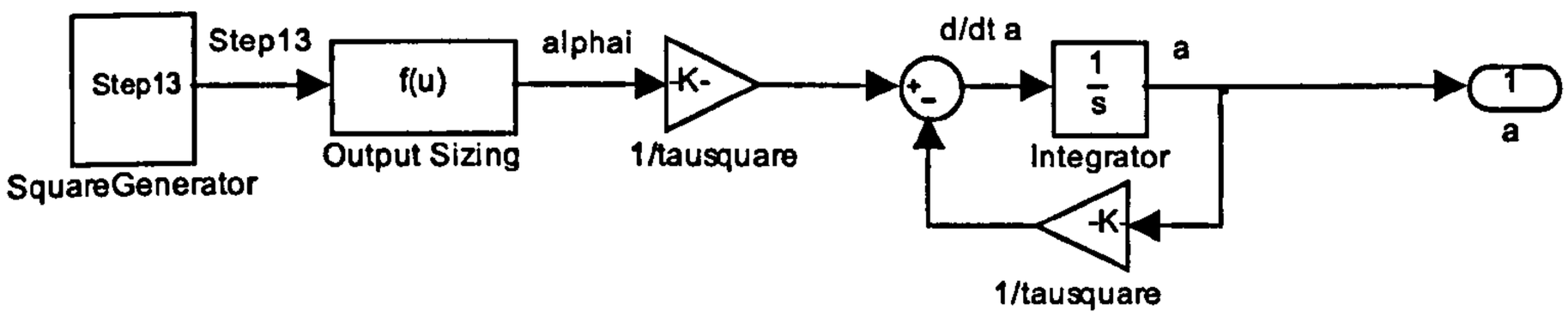


Fig. 3.4: Block diagram which simulates throttle plate angle

In equation (3.12) the intake manifold temperature has to be known accurately and instantaneously. Using the law of energy conservation a state equation which describes the time development of the intake manifold temperature can be given as:

$$\dot{T}_i = \frac{RT_i}{p_i V_i}\left[-\dot{m}_{ap}(\kappa - 1)T_i + \dot{m}_{at}(\kappa T_a - T_i) + \dot{m}_{EGR}(\kappa T_{EGR} - T_i)\right] \qquad (3.13)$$

The change in intake manifold temperature with respect to time is dependent on three factors as shown in equation (3.13) i.e. air mass flow into intake port, air mass flow past throttle plate and EGR mass flow respectively.

Equations (3.7) and (3.13) are the essential sub-models for the adiabatic MVEM shown in Fig. 3.5 and block diagrams of these sub-models are given in Fig. 3.6 and Fig. 3.7 respectively.

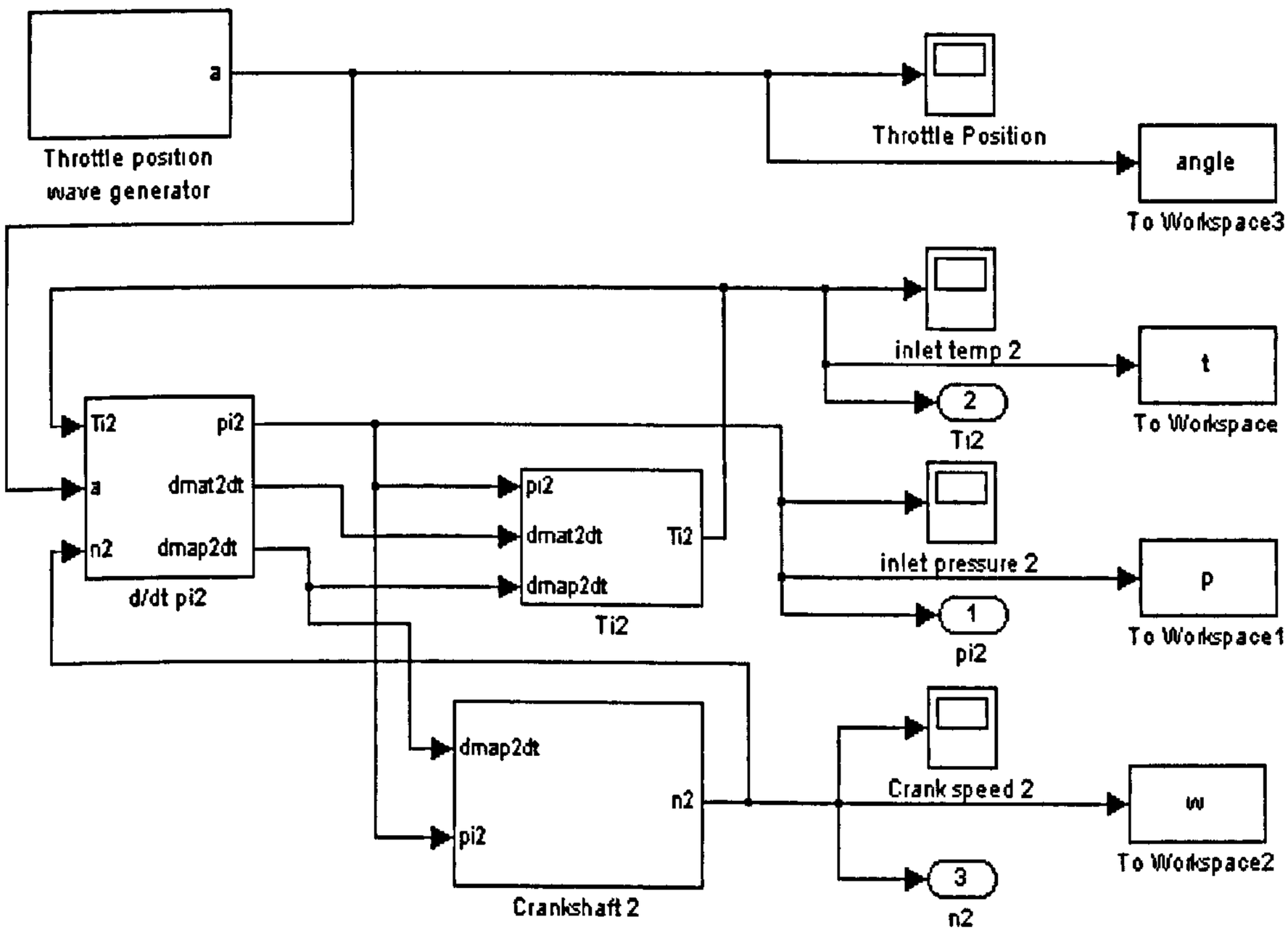


Fig. 3.5: Adiabatic MVEM Simulink Model

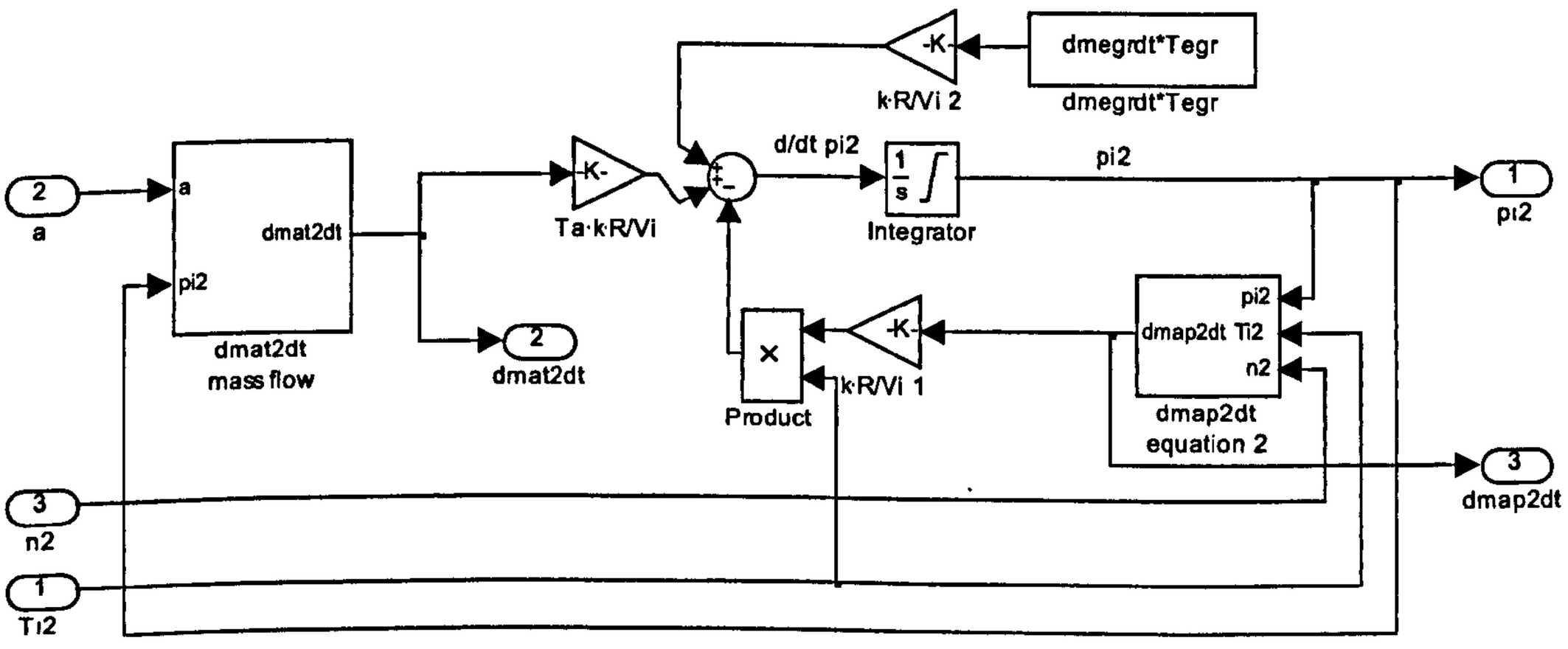


Fig. 3.6: Simulink model for intake manifold gas pressure dynamics

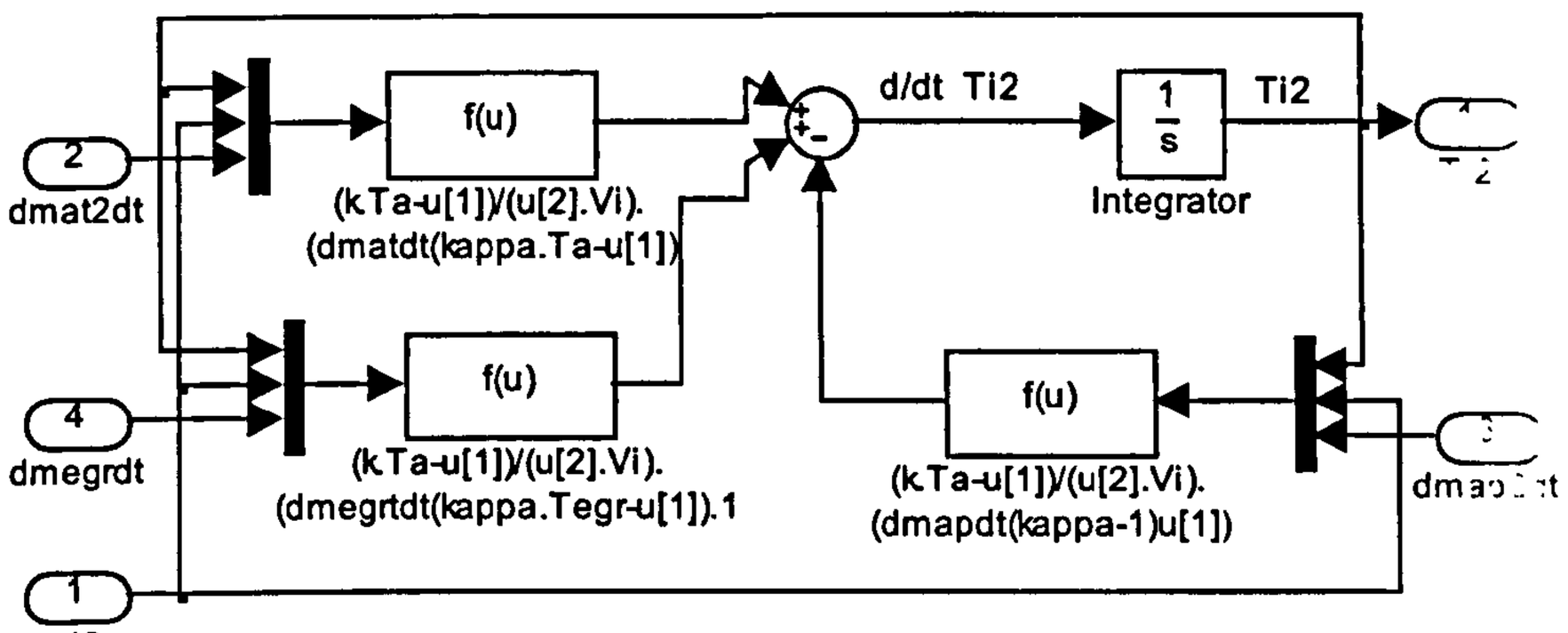


Fig. 3.7: Simulink model for intake manifold gas temperature dynamics

3.4.3 Crank Shaft Speed Dynamics

Applying the law of conservation of rotational energy, the crank-shaft dynamics of an SI engine (MVEM) is described by the following equation:

$$\dot{n} = -\frac{1}{In}\left(P_f(p_i,n) + (P_p(p_i,n) + P_b(n))\right) + \frac{1}{In} H_u \eta_i(p_i,n,\lambda)\dot{m}_f(t-\Delta\tau_d) \quad (3.14)$$

where I is the scaled moment of inertia of the engine and its load and the mean injection/torque time delay has been taken into account with variable $\Delta\tau_d$. λ is taken equal to "1" which corresponds to air/fuel ratio of 14.7 for gasoline and 14.5 for diesel. At a λ value of "1" we have stoiciometry or the point at which the most complete combustion takes place. λ gives a measure of Air Fuel Ratio which is independent of the type of fuel being used. λ > 1.0 => Excess Air (Lean) and λ < 1.0 => Excess Fuel (Rich). H_u is fuel lower heating value (kJ/kg). t is time in seconds and $\dot{m}_f$ is engine port mass flow in kg/sec. η_i is volumetric efficiency which is a function of absolute manifold pressure, crankshaft speed and air / fuel ratio.

Friction power P_f and pumping power P_p both are functions of absolute manifold pressure P_i and crankshaft speed n. Whereas load power P_b if only

the function of the crankshaft speed n. η_i is volumetric efficiency which is a function of absolute manifold pressure, crankshaft speed and air / fuel ratio.

The moment of inertia and the injection torque delay time can be given by the following equations:

$$I = I_{ac}\left(\frac{2\pi}{60}\right)^2 .1000, \qquad \Delta\tau_d = \frac{60}{n}\left(1+\frac{1}{n_{cyl}}\right) \tag{3.15}$$

and I_{ac} is the actual moment of inertia for a port injected engine. The functions which describe the frictional, pumping and load power as well as thermal efficiency can be found in Hendricks *et al.*, (1996).

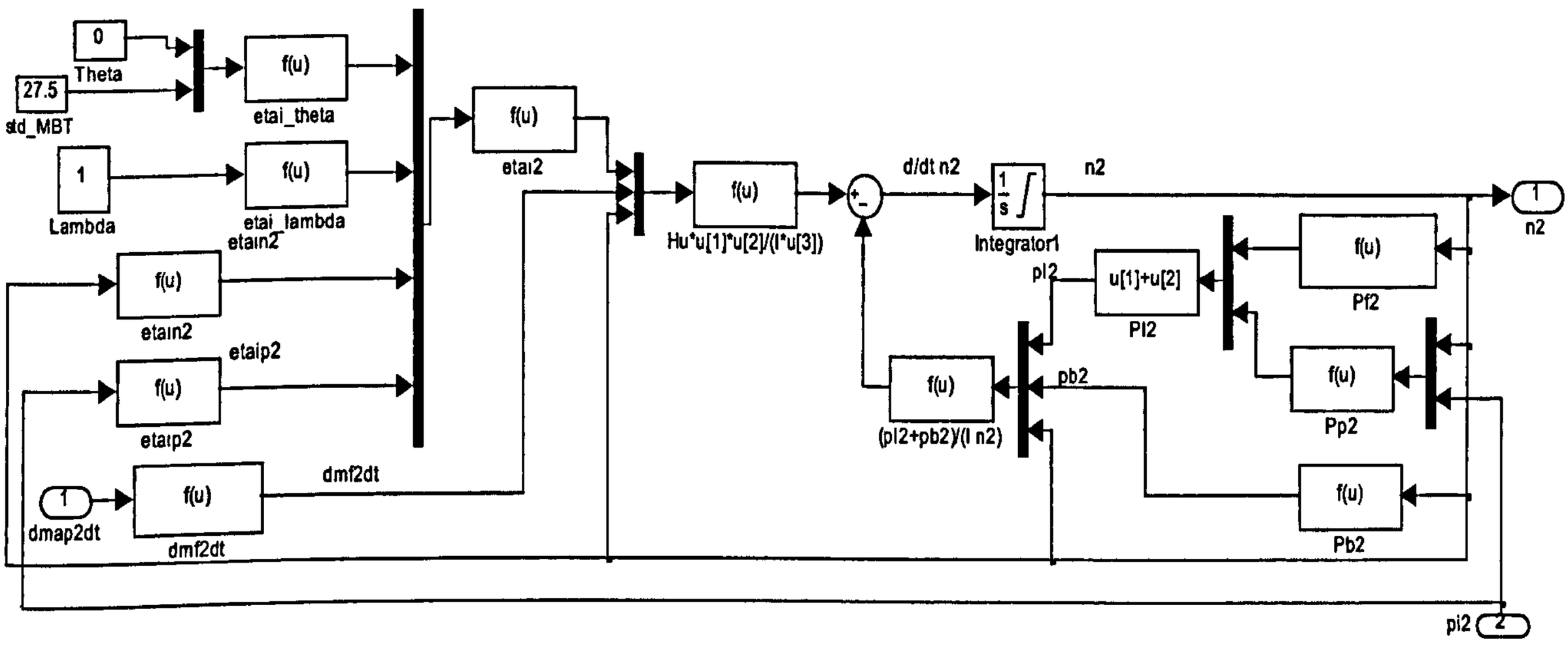


Fig. 3.8: Simulink model of the crankshaft speed state equation

The crankshaft is driven due to fuel flow to the engine and therefore the fuel flow has to be determined in such a way that the air/fuel ratio has a reasonable value such as $0.8 \leq \alpha \leq 1.2$. Fuel flow to the crankshaft speed state equation has thus to be given by an expression of the form

$$\dot{m}_f = \frac{\dot{m}_{ap}}{\lambda_{des} L_{th}}$$

The desired value of the air fuel ratio lambda is 1 for an engine which uses a three way catalyst. For maximum power during acceleration the value may be increased to 0.85 or 0.90. When going idle from cruise fuel is usually cut off completely. When cursing at a constant speed lambda should be between 1 to 1.1, depending on the engine and its operating mode.

3.5 SUMMARY

A brief introduction of the basic working of diesel engine and petrol/gasoline engine is given. This research is mainly about fault diagnosis of the engine air path and therefore the schematic view of the air path of the engine is shown and briefly explained. The mean value engine is extensively used in this research and therefore the basic MVEM equations and the important MATLAB/Simulink models and sub-models are reviewed e.g. fuel mass flow dynamics, intake manifold temperature and pressure dynamics, crankshaft speed dynamics and the throttle plate angle dynamics. The derivations of all the equations are available in the previous works of Hendricks and others and therefore instead of deriving all equations, most of them are just referred in the published work of other authors.

CHAPTER 4

FDI WITH NON-ADAPTIVE CLASSIFICATION

4.1 INTRODUCTION

Fault diagnosis of an automotive spark ignition (SI) engine using a non-adaptive classification method has been investigated in this chapter. The study is based on the mean value engine model (MVEM) simulation that is reviewed in the previous chapter. Four realistic and practical faults are considered for investigation, which are air leakage in the intake-manifold, Exhaust Air Recycle (EGR) valve clogged in different positions, intake-manifold pressure and temperature sensor bias faults. All the four faults are considered at four different levels of intensity. The experiment is designed and conducted in three different stages of engine operation to enhance generality.

The RBF neural networks with the *K*-means, *P*-nearest neighbour's and batch least squares (BLS) training algorithms are used for the investigation. The faults considered are similar to those studied in other research (e.g., Nyberg and Stutte, 2004; Capriglione *et al.*, 2003 and 2007; Gertler et al., 1995; Nyberg and Nielsen, 1997). The air leakage fault (as small as 2mm leakage-diameter) is also investigated by Nyberg and Perkovic (1998), and Antory (2007). Similar sensor bias faults as in this research are investigated by Nyberg (1999) using structured hypothesis tests.

4.2 FAULT SIMULATION

Two component and two sensor faults with four different levels of intensity have been investigated as four typical and practical examples of SI engine faults. The two component faults are exhaust air recycle (EGR) valve clogged and air leakage in the intake manifold. The two sensor faults are intake manifold pressure and temperature sensor faults.

The sensor faults can occur due to two reasons:

(i) aging and wear and tear of the mechanical parts of the deflection meter and
(ii) electrical fault e.g. short circuit or open circuit fault in the signal cable.

The electrical faults such as open circuit and short circuit faults are easy to detect because they cause a full deflection or zero deflection in the meter respectively. On the contrary, the aging and mechanical faults cause an incorrect meter reading i.e. over- or under- reading of the actual values. Both cases of under- and over- reading of the temperature or pressure measurements are considered here.

Air leak in the air path of an engine can happen due to any of the following reasons (Reineman, 2000) or a breakage due to some physical damage:

(i) missing air cap
(ii) loose air cap
(iii) leaks in air cap or vapour vent lines
(iv) disconnected purge lines

Current OBD regulations require monitoring of any leaks (for 2003 year model and after) that exceed 0.02 inch in diameter (0.5 mm approx). It is not practical to create some component faults in a running engine in real life, such as air leakage in manifold or EGR valve stuck. Therefore the faults are simulated in a Matlab/Simulink engine model in this research. The air leakage is simulated in the modified mean value engine model as a percentage of the total air mass flow in the intake manifold explained later. The EGR valve can

be stuck up in any position where there is a failure of the EGR valve positioning control. This will lead to a fixed percentage EGR flow through the valve. There can be many reasons for failure of the EGR valve positioning system, which have not been investigated here. The investigation is focused on the detection and isolation of the fault with different intensities and not on pin-pointing the actual component failure of the EGR system viz.

- EGR fault open circuit,
- EGR fault vent solenoid,
- EGR step motor 1 fault,
- EGR step motor 2 fault,
- EGR vacuum regulator fault,
- EGR boost solenoid control, etc.

Details of the simulation of the faults are described in the following sub-sections.

4.2.1 No Fault

No fault case implies that all the sensors are working well and there is no air leak in the intake manifold. And also, the EGR is assumed to be 1/6 (16.67%) of the total air mass flow in the intake manifold. In order to collect no fault data, the MVEM simulation is run for 6 seconds. The simulation is run for 6 seconds because in this period of time all the output variables reach the steady state condition. Therefore, to collect sufficient information of the transient state is fully reflected in the collected input/output data. The sampling time is chosen to be 0.3 seconds and 20 data points are collected for throttle angle input and for all the three output variables i.e. inlet manifold pressure, temperature and crankshaft speed.

4.2.2 Air Leakage Faults

To collect the engine data subjected to the air leakage fault, equation (3-12) is modified to

$$\dot{P}_i = \frac{\kappa R}{V_i}\left(-\dot{m}_{ap}T_i + \dot{m}_{at}T_a + \dot{m}_{EGR}T_{EGR} - \Delta l\right) \tag{4-1}$$

where Δl is used to simulate the leakage from the air manifold, which is subtracted to increase the air outflow from the intake manifold. $\Delta l = 0$ will represent no air leakage in the intake manifold. Fig. 3.7 which represents Simulink model for intake manifold air pressure dynamics is modified accordingly as shown Fig. 4.1.

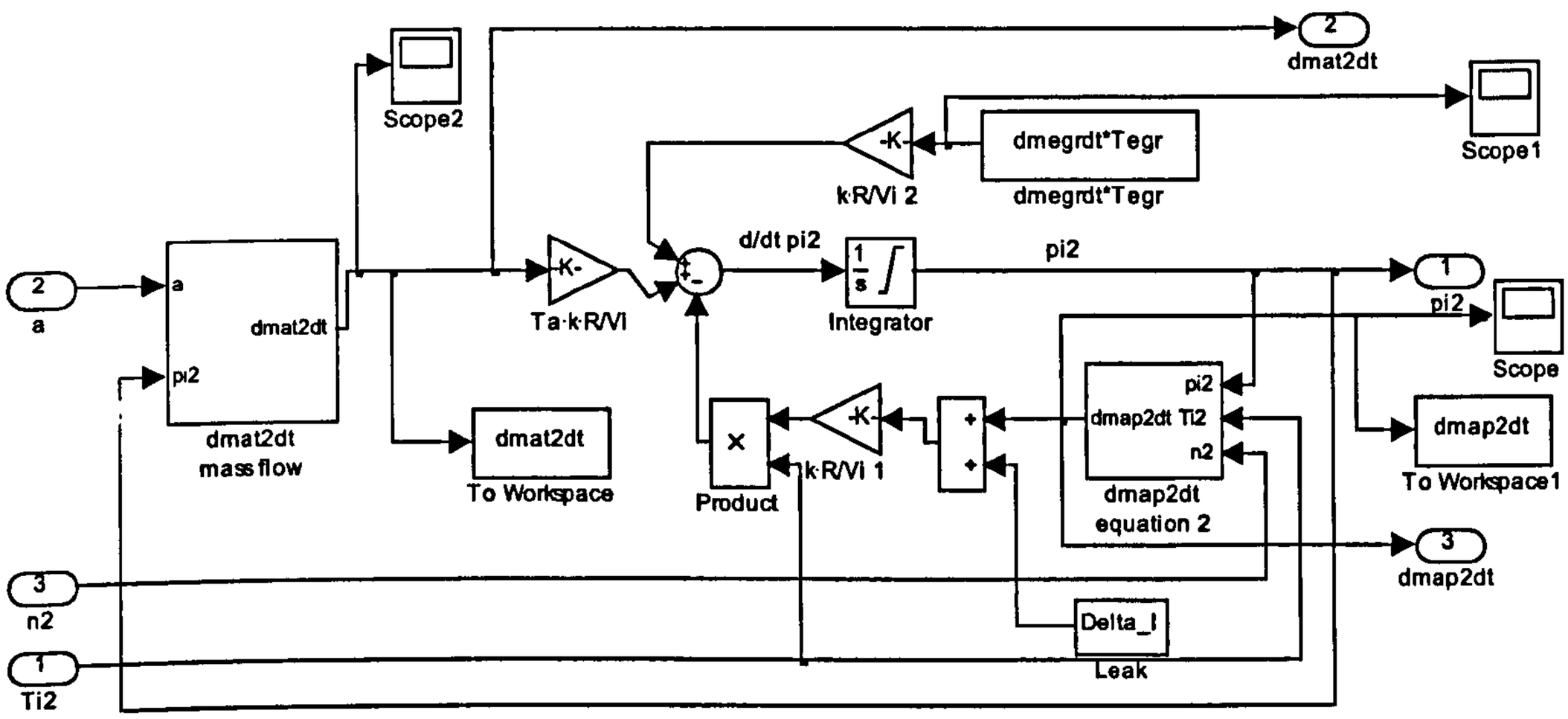


Fig. 4.1: Modified Simulink model for intake manifold air pressure dynamics

Comparing Fig. 3.7 and Fig. 4.1 the simulation is modified to increase the outflow of air from the intake manifold to represent leakage in the intake manifold. The air leakage levels are simulated as 5%, 10%, 15% and 20% of the total air intake in the intake manifold, respectively. For example, if total air mass flow in the intake manifold at steady state (for 30° throttle angle) is 0.0317 kg/sec then 5% air leakage will be 0.0317x0.05 = 0.001585 kg/sec.

4.2.3 EGR Valve Faults

The normal value of EGR is kept as 1/6 of the total air mass flow, i.e. 16.67%. The EGR can be as high as 20% of the total air mass flow in the intake

manifold. Thus, a realistic value of EGR feedback is chosen for the experiments. The value of $\dot{m}_{EGR}$ for different fault intensities are regulated as 0%, 25%, 50%, 75% and 100% of the total EGR air mass flow. Where 0% EGR air mass flow corresponds to the EGR valve stuck in hundred percent-closed positions and 100% corresponds to full EGR air mass flow, i.e. no fault condition.

Table 4.1: All 17 fault states and multiplying factors (MF)

S.No.	Fault Name	MF
1	No Fault	
2	Leakage 5%	
3	Leakage 10%	
4	Leakage 15%	
5	Leakage 20%	
6	EGR clogged 25% closed	
7	EGR clogged 50% closed	
8	EGR clogged 75% closed	
9	EGR clogged 100% closed	
10	Temp. sensor 20% over reading	MF=1.2
11	Temp. sensor 10% over reading	MF=1.1
12	Temp. sensor 10% under reading	MF=0.9
13	Temp. sensor 20% under reading	MF=0.8
14	Pressure sensor 10% over reading	MF=1.2
15	Pressure sensor 20% over reading	MF=1.1
16	Pressure sensor 20% under reading	MF=0.9
17	Pressure sensor 10% under reading	MF=0.8

4.2.4 Temperature/Pressure Sensor Faults

Temperature and pressure sensor faults are considered in four different intensities: Sensors over-reading 20% or 10% and sensors under-reading 10% or 20% of the normal value. The fault data for the sensors is generated using multiplying factors of 1.2, 1.1, 0.9 and 0.8 for over reading 20%, 10% and under reading 10%, 20% respectively. Faulty data are collected when throttle angle is changing between 20° and 40° for all the fault conditions

including no fault condition. Note that the total range of the throttle angle is $[20^0, 40^0]$ in the MVEM, which represents the whole operating space. All the 17 fault states with multiplying factors are given in Table 4.1.

4.3 NORMALISATION OF DATA

Data normalisation is necessary before inputting it through a Neural Network for learning so that variable with higher numerical value will not dominate the learning. Moreover, the graphical data analysis will be difficult due to huge difference in the numerical values of data. One way to normalise data is to find deviation from the normal steady state as given in equation (4-2).

$$\text{Deviation} = \frac{x - x_{ss30}}{x_{ss30}} \tag{4-2}$$

where, x_{ss30} is the steady state value of the variable at 30^0 throttle angle input. Considering steady state values of all the three outputs as normal at 30^0 throttle angle input for no fault condition (i.e. x_{ss30}), the deviation is calculated using above formula along with the use of a proper multiplier. The steady state values of manifold temperature, pressure and the crankshaft speed at 30^0 throttle angle input are 293.79 Kelvin, 0.8038 bar, 3.5810 rpm/1000 respectively.

All the three outputs of MVEM (i.e. manifold temperature and pressure and crankshaft speed) and the throttle angle input are normalised and are used as an input to the neural network for training because all of these contain useful transient information of the engine air path dynamics. This is further explained in section 4.4.3 ahead. The four neural network inputs are graphically shown in Fig. 4.2, 4.3, 4.4 and 4.5 for constant speed run explained in section 4.5.1.

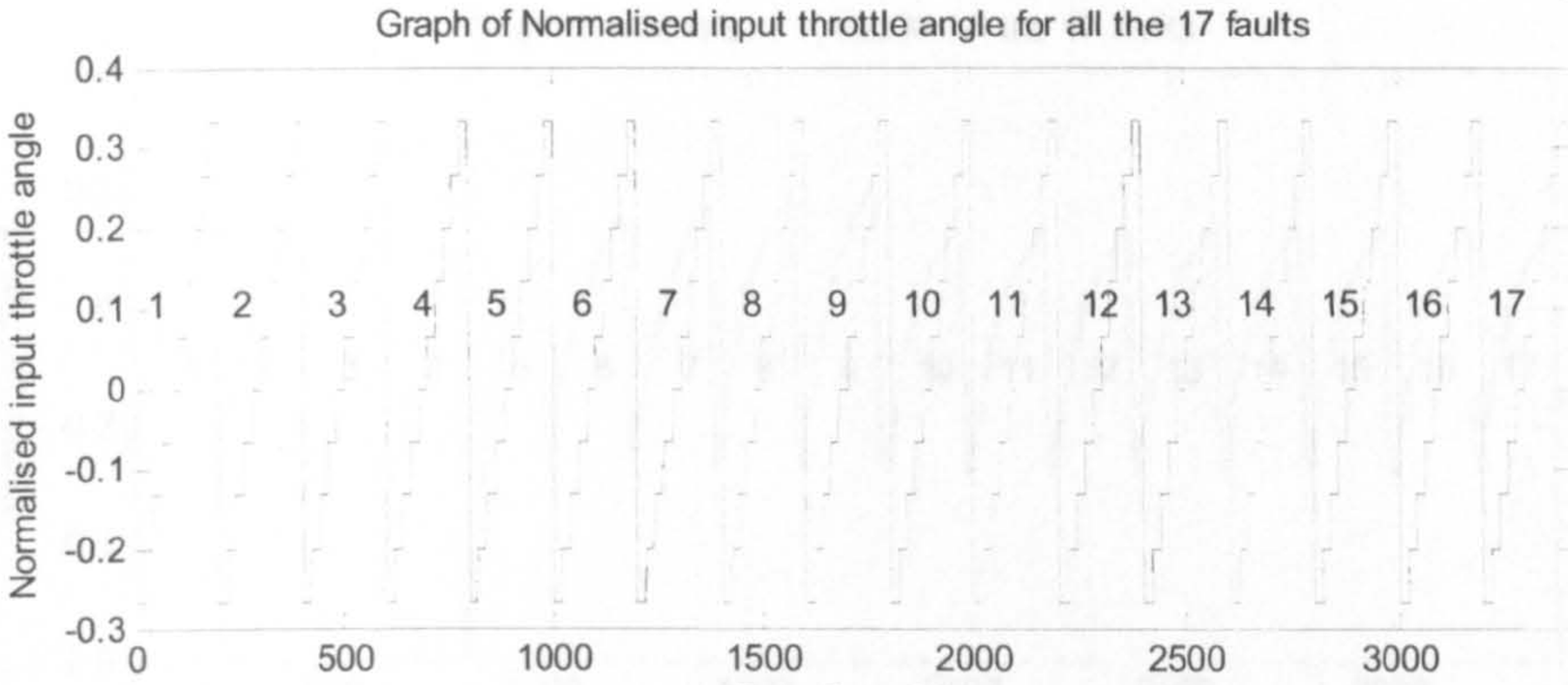


Fig. 4.2: Normalised throttle angle input for all the 17 faults

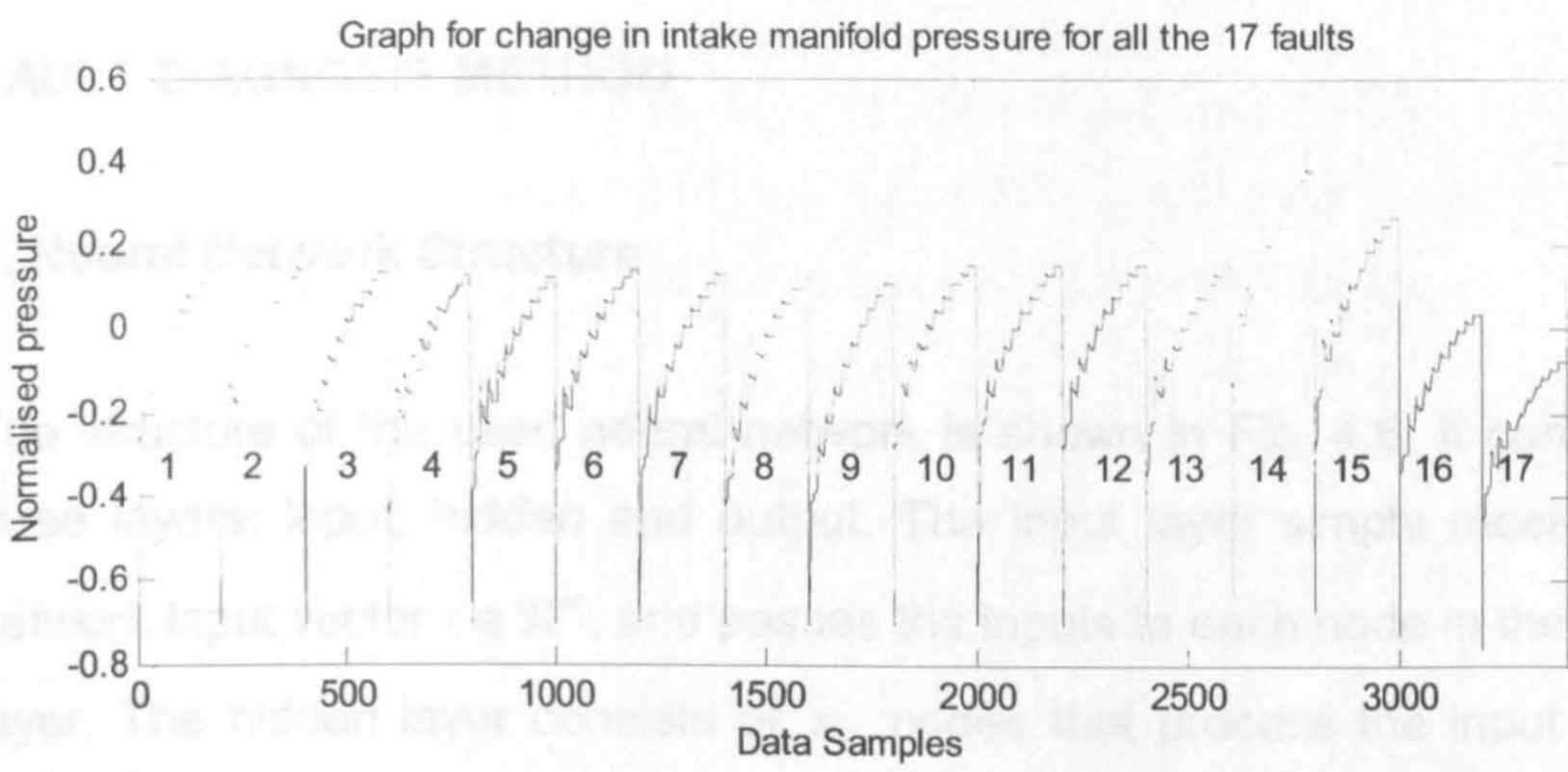


Fig. 4.3: Normalised manifold pressure input for all the 17 faults

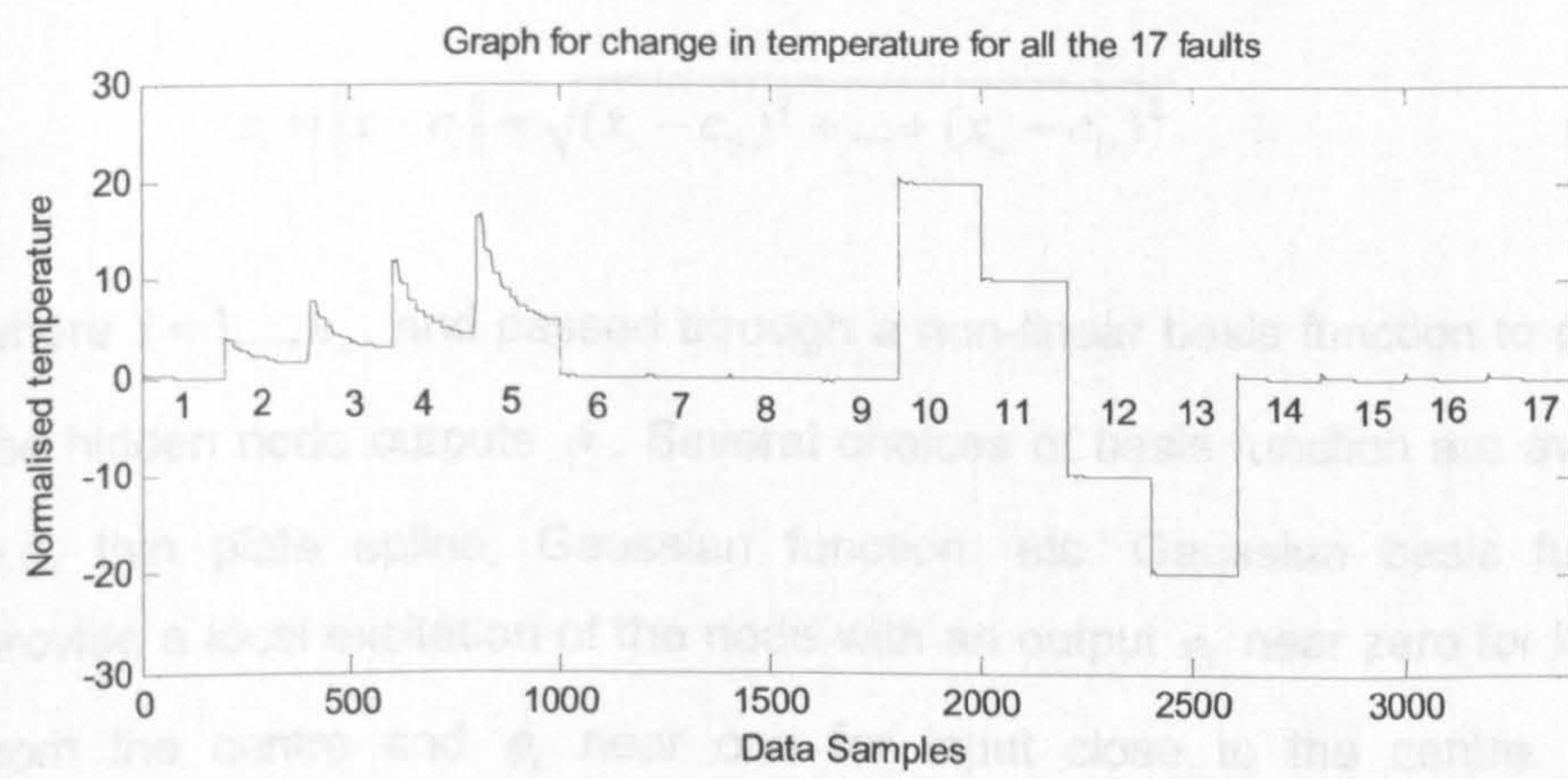


Fig. 4.4: Normalised manifold temperature input for all 17 faults

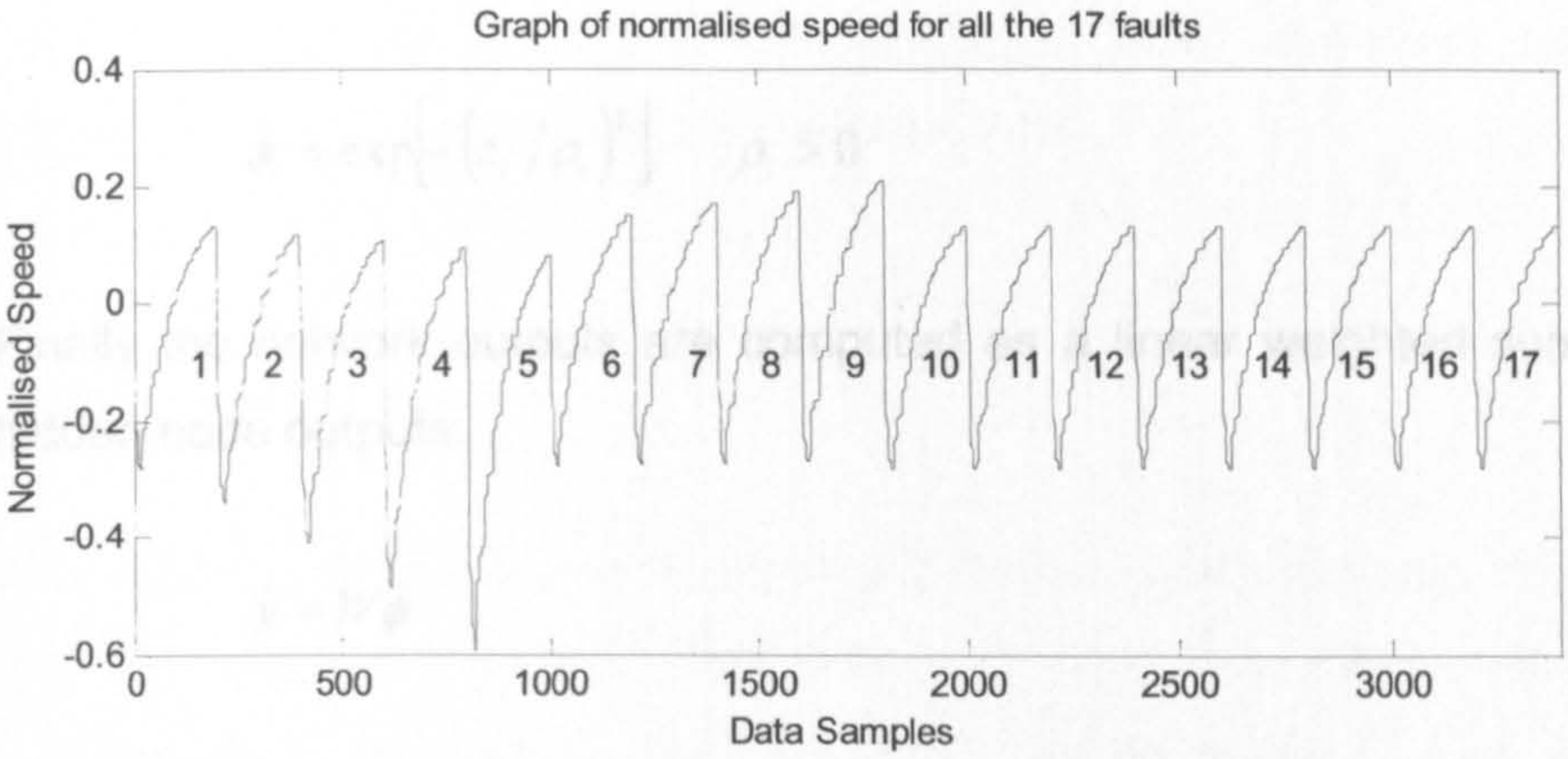


Fig. 4.5: Normalised crankshaft speed input for all the 17 faults

4.4 FAULT DIAGNOSIS METHOD

4.4.1. Neural Network Structure

The structure of the used neural network is shown in Fig. 4.6. It consists of three layers; input, hidden and output. The input layer simply receives the network input vector $x \in \Re^n$, and passes the inputs to each node in the hidden layer. The hidden layer consists of n_h nodes that process the input vector. The i^{th} node in the hidden layer contains an individual centre vector c_i of the same dimension as *x* and a scalar width ρ_i. The Euclidean distance between the input and the centre vectors is calculated,

$$z_i = \|x - c_i\| = \sqrt{(x_1 - c_{i1})^2 + \ldots + (x_n - c_{in})^2} \qquad (4\text{-}3)$$

where $i = 1,\ldots,n_h$, and passed through a non-linear basis function to produce the hidden node outputs ϕ_i. Several choices of basis function are available, e.g. thin plate spline, Gaussian function, etc. Gaussian basis functions provide a local excitation of the node with an output ϕ_i near zero for input far from the centre and ϕ_i near one for input close to the centre. This is especially suitable for classification applications and is therefore used in this work. The Gaussian basis function is defined as

$$\phi_i = \exp\left[-\left(z_i / \rho_i\right)^2\right] \qquad \rho_i > 0 \tag{4-4}$$

Finally the network outputs are computed as a linear weighted sum of the hidden node outputs:

$$\hat{y} = W\phi \tag{4-5}$$

Where $\hat{y} \in \Re^q$ is the output vector, $W \in \Re^{q \times n_h}$ is the output layer weight matrix with element w_{ij} connecting the j^{th} hidden node to the i^{th} output, and ϕ is a vector containing the hidden node outputs.

For training RBF neural network the *K*-means algorithm is used to choose the centres, *P*-nearest algorithm decides the widths and the batch least squares algorithm calculates weights for the output layer of RBF network.

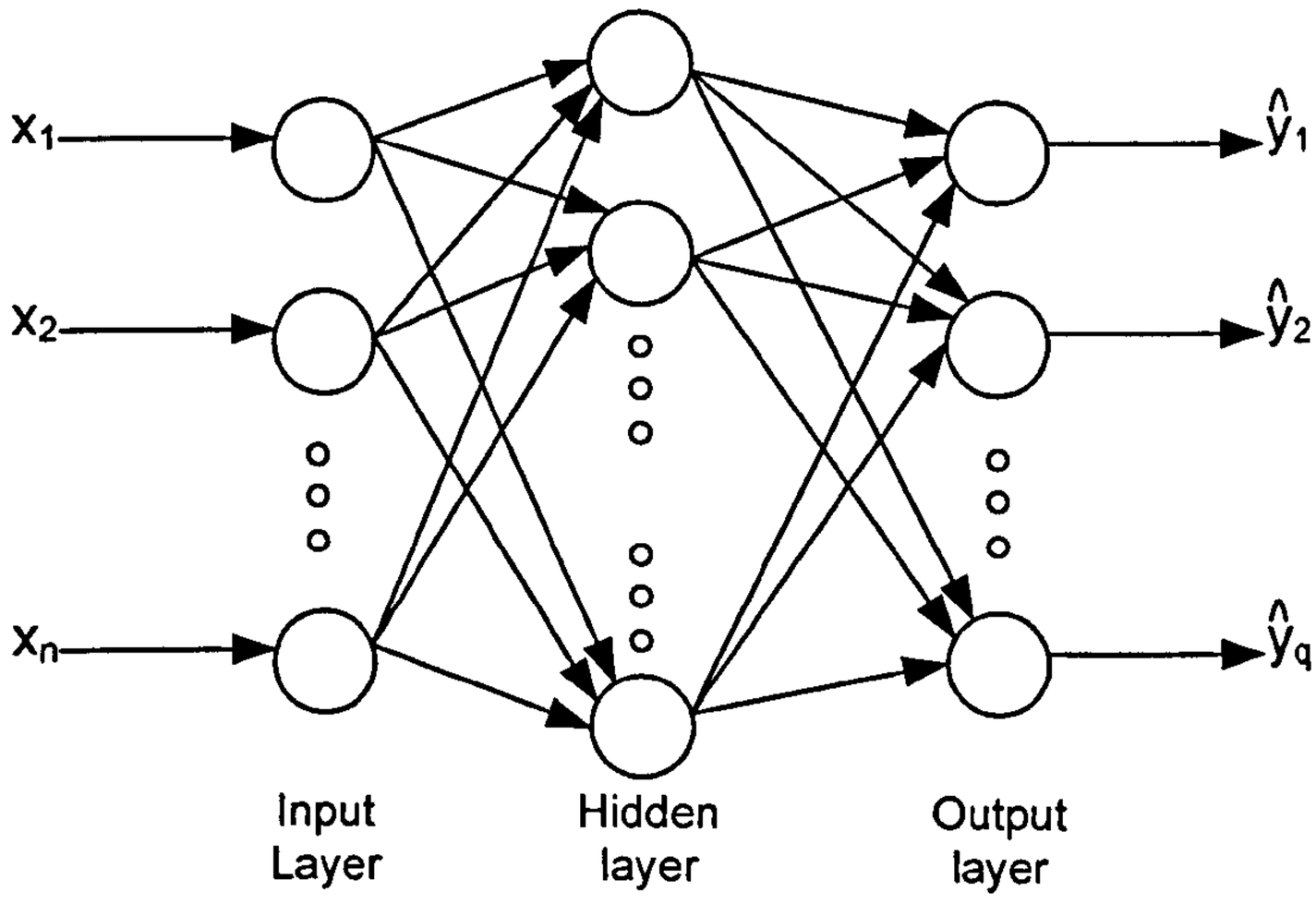


Fig. 4.6 The RBF neural network structure

4.4.2. RBF Off-line Training Algorithms

Training an RBF network means optimising the parameters of centres, widths and weights in the network. For training the RBF network the *K*-means, *P*-

nearest neighbours and batch least squares (BLS) algorithms are used. These algorithms are reviewed or derived below.

A. K-means Algorithm

The objective of the *K*-means clustering method is to minimise the sum of squared distances from each input data to its closest centre so that the data is adequately covered by the activation functions $\phi_i(t)$. The *K*-means clustering method proceeds as follows:

(i) Randomly choose some input data to be the initial centres. The number of the centres is designed according to the complexity of the problem.

(ii) Let $p(x)$ denotes the index of the best-matching centre for the input vector x. Find $p(x)$ at iteration t by minimising the sum squared distances:

$$p(x) = \arg\min \left\| x(k) - c_i(k) \right\|^2 \tag{4-6}$$

Where $c_i(k)$ is the centre of the i^{th} activation function at iteration k.

(iii) Update the centres of the activation functions by using the following rule:

$$c_i(k+1) = \begin{cases} c_i(k) + \alpha_c \left[x(k) - c_i(k) \right], & if\ k = p(x) \\ c_i(k), & otherwise \end{cases} \tag{4-7}$$

Where α_c is the centre learning rate that lies in the range (0, 1).

(iv) Increment k by 1 and go back to step (ii). Continue the algorithm until no noticeable changes are observed in the centres C_i.

B. P-nearest Neighbour's Algorithm

The P-nearest neighbours algorithm (Moody and Darken, 1989) determine width for each centre as the square root of average squares of the distances from the centre to the nearest p centres as described in the following equation:

$$\sigma_i = \sqrt{\frac{1}{p}\sum_{j=1}^{p}\left\|c_i - c_j\right\|^2}, \quad i = 1,\cdots,n_h \tag{4-8}$$

where c_j is the j^{th} nearest centre to c_i

C. Batch Least Squares (BLS) Algorithm

BLS algorithm is widely used for off-line training. The batch least squares method is responsible for training the weights by using the following equation:

$$W = (\phi^T\phi)^{-1}\phi^T Y \tag{4-9}$$

Where W is the matrix of weights, ϕ is the matrix of activation function outputs and Y is the matrix of training targets. If the RBF neural network has n inputs, n_h hidden nodes and q outputs as shown in Fig. 4.6 then the dimensions of W, ϕ *and* Y will be (q by n_h), (n_h by 1) and (N by q) respectively. (N is the number of samples in a set of data).

4.4.3 Fault Diagnosis Method

The RBF network, as the fault classifier will receive all possible and relevant signals containing fault information, and has 17 outputs with each indicating one of the investigated states in Table 1. The information flow for the fault diagnosis is illustrated in Fig. 4.7. According to the engine air path dynamics, four variables are chosen as the network inputs: the throttle angle, the

manifold pressure, the manifold temperature and the crankshaft speed as shown in Fig. 4.7.

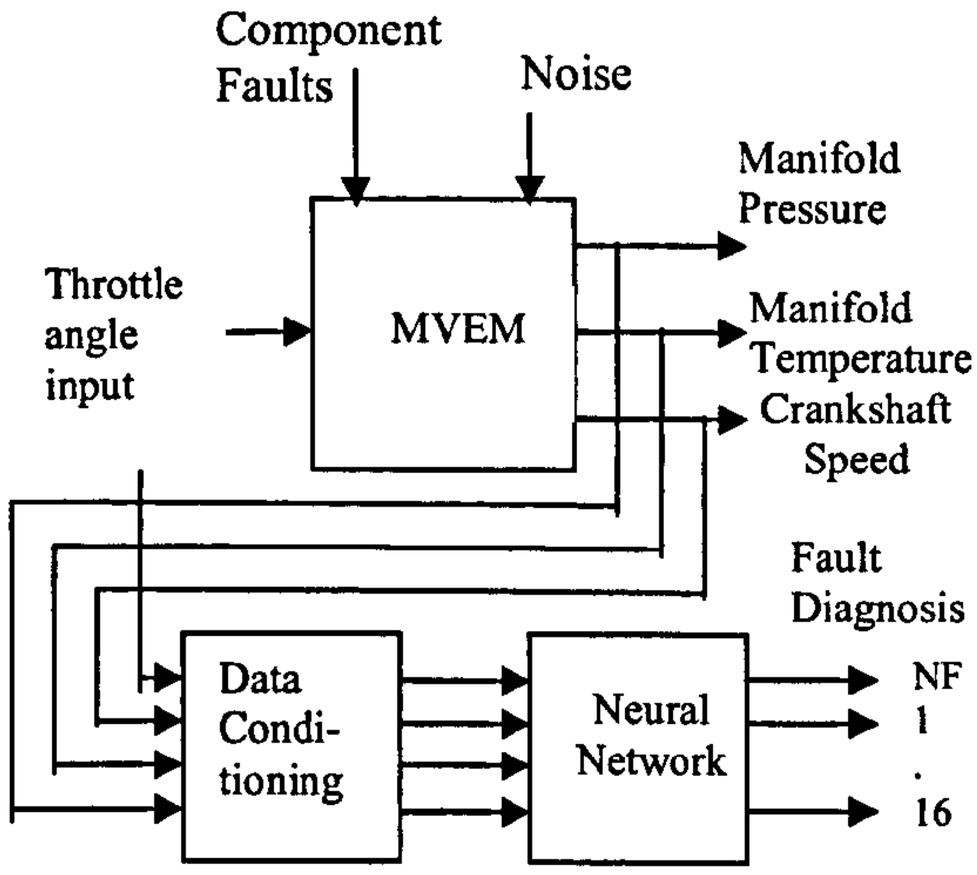


Fig. 4.7 Information flow of the fault diagnosis

Two levels, 0 and 1, are used as the output targets of the classifier. Thus, the target matrix is a unity diagonal matrix of dimension 17 (when there is one training pattern for each fault) with each column being used as the classifier-training target vector. A successfully trained network will therefore diagnose the fault intensity as well as the fault type, as shown in Table 4.1.

4.5 TRAINING AND TESTING OF NEURAL CLASSIFIER

Three cases are considered with increasing generality of engine operation. In case 1, engine runs at different steady state speeds and the faults occur one by one. For example, the engine runs at a constant speed corresponding to a fixed throttle angle input of 24°. In case 2, engine accelerates or retards from a mean speed corresponding to throttle angle input of 30° and the faults occur one by one. In case 3, engine accelerates or retards from any initial speed to any final speed and different faults occur one by one. The third case is the most general case and previous two cases are the sub-sets of the third case. The neural network is trained and tested for all the three cases and results are shown in the next three sub-sections.

4.5.1 Case 1: Constant Speed Run

For this part of the experiment, the engine is run in steady-state for a particular throttle angle input and a fault occurs. The throttle angle is also used as a neural network input along with pressure, temperature and speed as shown in Fig. 4.7. The network has 17 outputs, one to indicate each of the investigated states in Table 4.1. The training data is collected for 11 different throttle angles ranging from 20°, 22°, 24°, ..., 40° for all the 17 different states and therefore 11 data sets are collected, one for each throttle angle input. The network is trained to output a high value (1) on one output to indicate a particular operating state with other outputs low (0). A successfully trained network will therefore indicate the fault intensity as well as the fault type. In each throttle angle and for each fault 20 samples are collected. There are 10 throttle angles and 17 faults. Therefore, ten data sets (10x20x17=3400) are used for training and one data set (1x20x17=340) is used for testing at a time. For example when the neural network is tested for 26° throttle angle then the training data set excluded 26° throttle angle data. And when the RBF neural network is tested for 22° throttle angle then 22° throttle angle data is not included in the training set. The training data matrix size is 4x3400 and the target matrix size is 17x3400. The testing data matrix size is 4x340.

Training data sets for case 1 and case 2 have the same pattern for 17 faults, and therefore, one training target matrix X_o is formed and used for both case 1 and case 2. Target matrix has 3400 rows and 17 columns. Its first column has ones from the first row to 200th row and the other entries are zeros, the second column has ones from the 201st row to the 400th row and the other entries are zeros, the last column has ones from the 3210th row to the 3400th row and the other entries are zeros. This is shown as follows:

Row Numbers	X_o
$1 \sim 200$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 1 & \cdots & & & \\ 0 & \vdots & \ddots & & & \\ 0 & & & \ddots & & \\ 0 & & & & 1 & \\ \vdots & & & & & \ddots \end{bmatrix}$
$201 \sim 400$	
$\vdots$	
$3201 \sim 3400$	

Fig. 4.8: Target matrix X_o

Thus, the transpose of the i^{th} row in X_o is used as the training target vector for the i^{th} training pattern. 130 hidden nodes are chosen by k-means method. Widths and weights are trained using P-nearest neighbours and batch least square algorithms respectively. The trained network is then tested for all faults occurring at some throttle angle inputs which was not included in the training data. The RBF network structure is 4x130x17.

Fig. 4.9 shows typical test results for 26° throttle angle. The fault detection threshold is chosen as 0.5. High thresholds may lead to missed detections whilst low thresholds will cause false alarms. Threshold is chosen as 0.5 by utilising experience in minimising false alarm rate. The simulation is run for different values of threshold and threshold = 0.5 (mean value of two target values i.e. 0 and 1) is chosen for minimal false alarms. The classification result shown in Fig. 4.9 which has misclassifications which would cause false alarms.

The network has been tested for many different throttle angles but the results are more or less the same and therefore only one result is shown here.

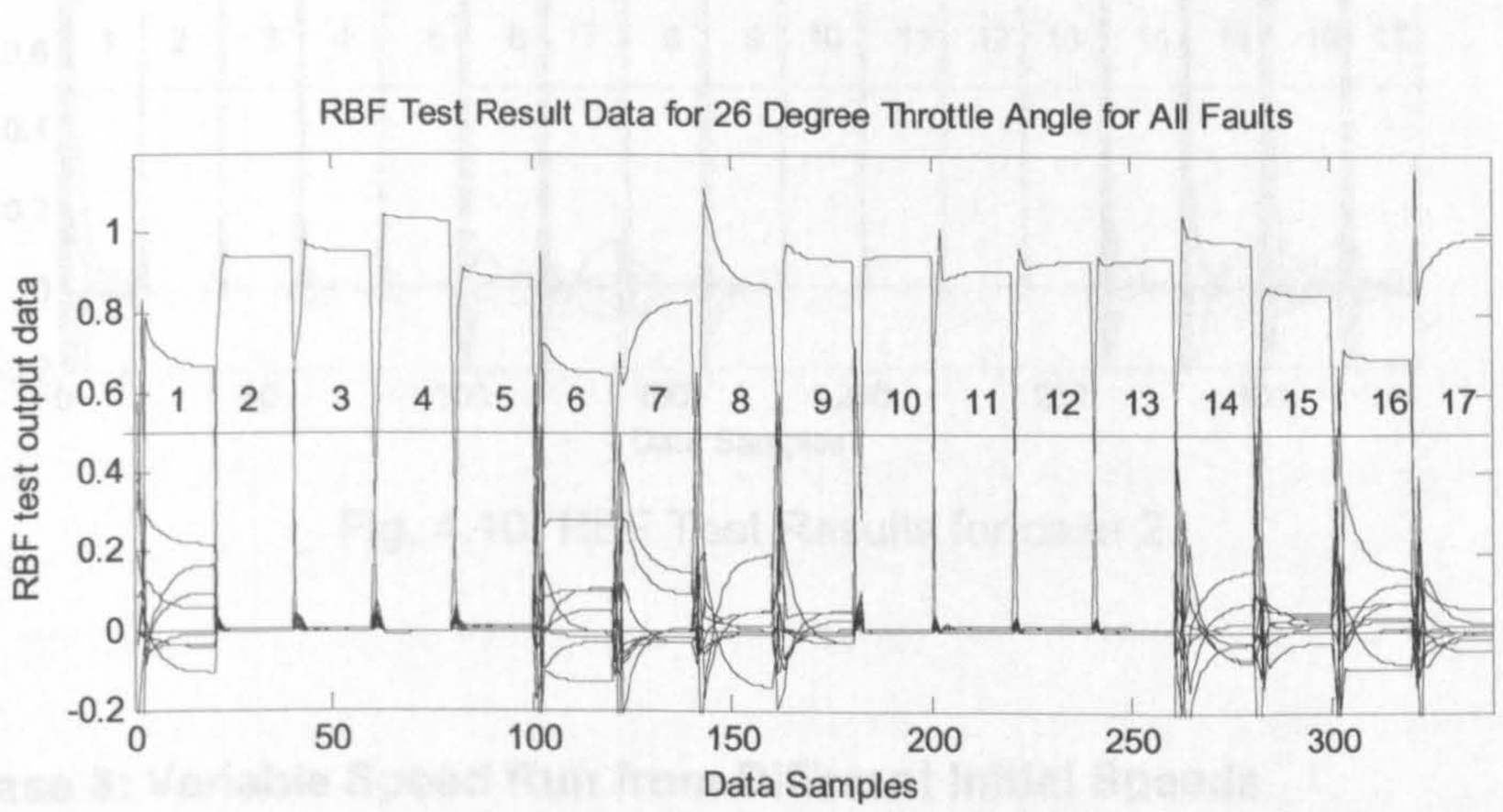


Fig. 4.9: RBF test results for case 1

4.5.2 Case 2: Variable Speed Run

In order to train the neural network (NN) dynamically, throttle angle inputs of 20°, 22°, 24°, ..., 40° are applied for each fault condition, keeping all the initial conditions set to 30° throttle angle. It is assumed that the engine is initially

running for 30° throttle angle before accelerating or retarding the speed, i.e. increasing or reducing the throttle angle.

30° is chosen as the initial condition because it is the mean value of the selected operation range of the engine. The RBF NN is trained from this data as before, for the whole range of throttle angles from 20° to 40° excluding one throttle angle which is used for testing as explained in the previous case. 190 centres are chosen by *K*-means method. Widths and weights are trained using *P*-nearest neighbours and batch least square algorithms respectively as before. The RBF network structure is 4x190x17. The trained network is then tested for all faults occurring at some throttle angle inputs which was not included in the training data. The test results for when engine is retarding from 30° to 26° are shown in Fig. 4.10. There are misclassifications in the result which would cause false alarms.

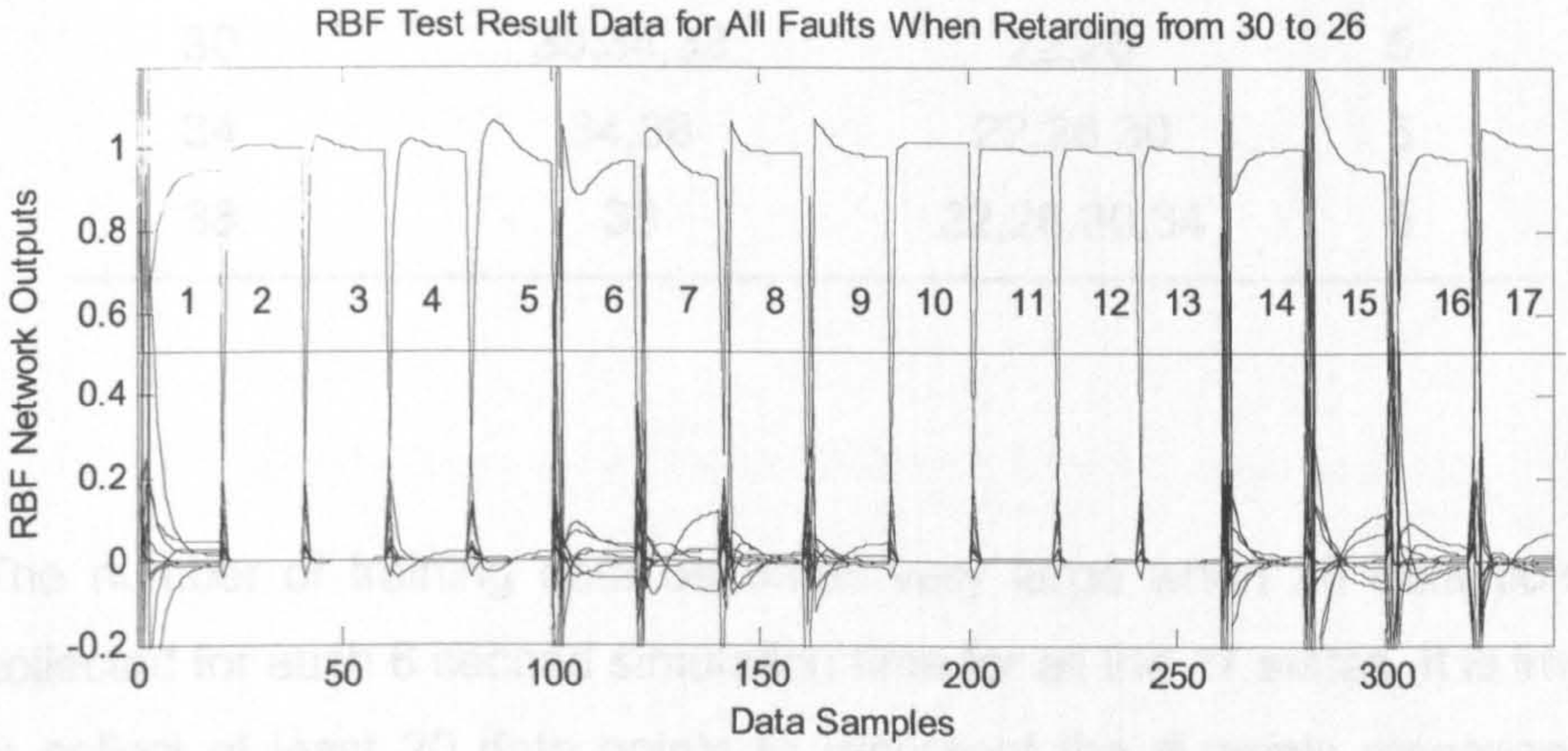


Fig. 4.10: RBF Test Results for case 2

4.5.3 Case 3: Variable Speed Run from Different Initial Speeds

In training of the fault classifier, the training data should represent the engine dynamics in the whole operating space. Considering practical implementation, we collect a number of sets of training data with each of which collected when engine runs in a different range of throttle angles and on different working modes. These ranges are distributed in the entire operating space from 20° to 40° and the modes include accelerating, constant speed and decelerating. This is the most general case when the engine initial conditions are any value and the engine accelerates or retards to any value when a fault occurs. The

previous two cases are subsets of this case. The engine is initialised to steady state for different throttle angles and then accelerated and retarded to other throttle angles as shown in Table 4.2. Table 4.2 can be read in the way that for example, the first line has 5 data sets. In the first set the throttle angle has a constant speed from 22° to 22°, in the second set the throttle angle change from 22° to 26°, in the third set the throttle angle changed from 22° to 30°, etc. 25 sets of data are collected all together.

Table 4.2: Training data for case 3

Initial Throttle Angle	**Engine Speed Accelerating to throttle angle**	**Engine Speed Retarding to throttle angle**	**Number of Sets**
22	22,26,30,34,38	Nil	5
26	26,30,34,38	22	5
30	30,34,38	22,26	5
34	34,38	22,26,30	5
38	38	22,26,30,34	5

The number of training data becomes very large when 20 data points are collected for each 6 second simulation time for all the 17 states. It is important to collect at least 20 data points to represent the dynamic response of the engine properly. For the 5 throttle angles considered there are 5x5=25 different acceleration/deceleration cases each for 17 states and each state has 20 data points. Thus, the number of training data points is 5x5x20x17=8500. The sizes of the input and target matrices for network training are therefore 4x8500 and 17x8500 respectively.

This large training data takes about 10 minutes for training with 200 hidden nodes on a standard Pentium-IV computer. The structure of the RBF is 4x200x17 and the test results for constant speed, acceleration and decelerating engine operations are shown in Fig. 4.11. It can be seen in the Fig. 4.11 that even a huge network of 200 hidden nodes is not able to classify large fault-set of 17 faults and most of the faults are not classified at all.

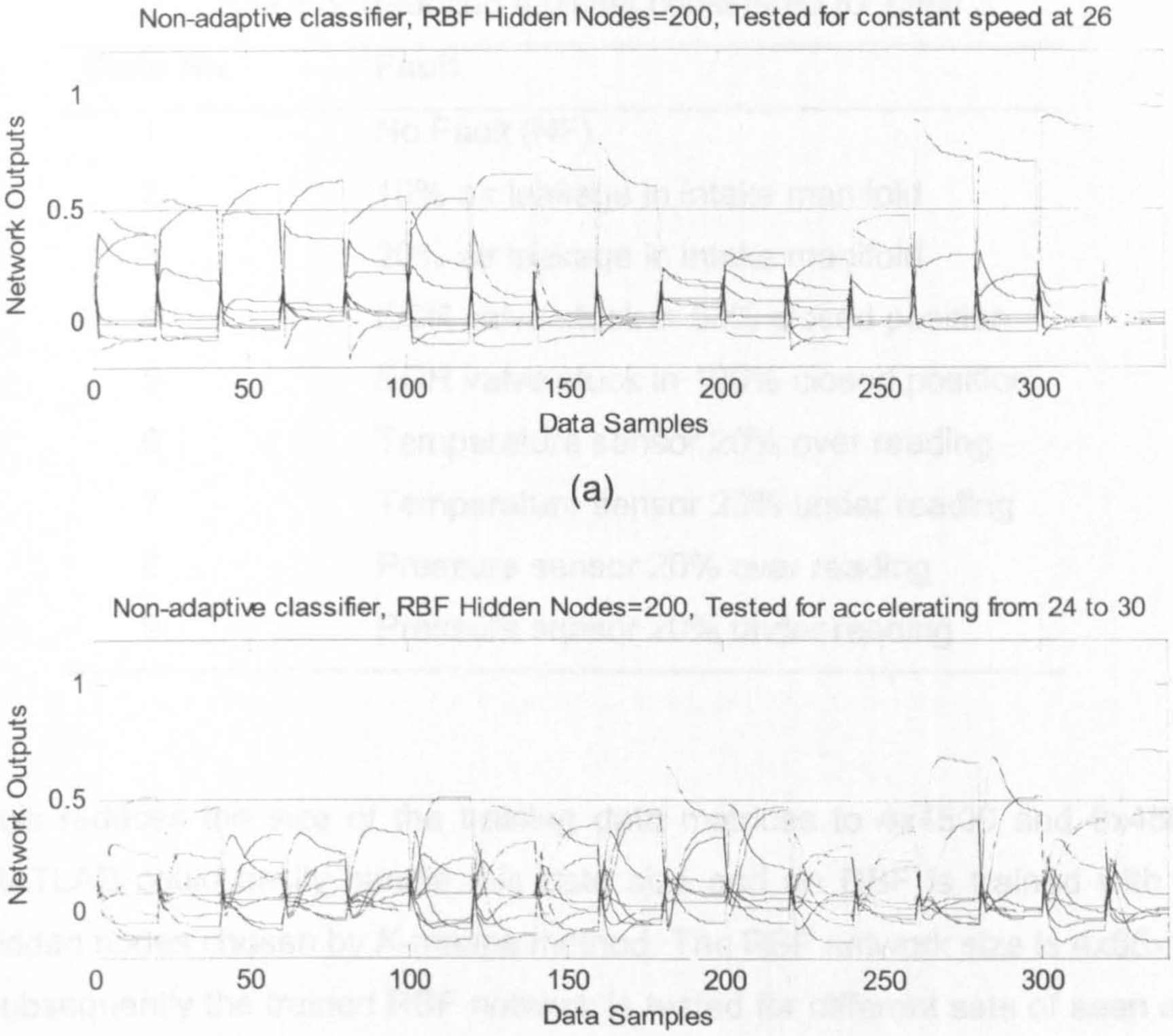


Fig. 4.11: RBF test results for case 3 (a) Constant speed at 26° throttle angle (b) Accelerating from 24° to 30° (c) Decelerating from 38° to 30° throttle angle

This off-line classifier is unable to classify a large set of faults and therefore the fault set is reduced to investigate with 9 faults. In order to reduce the fault set and training data volume, all four faults are considered with two levels of intensities instead of four as shown in Table 4.3.

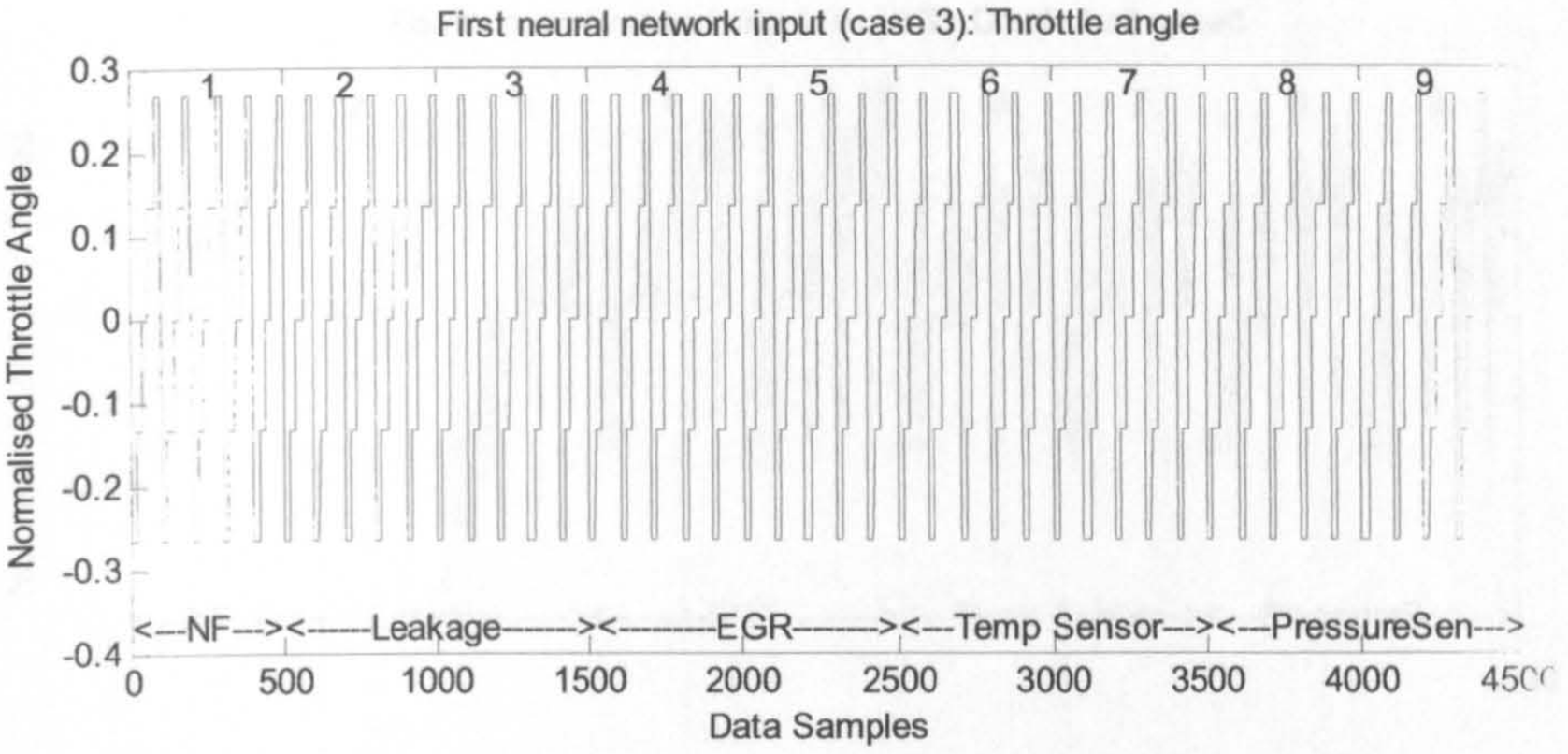
Table 4.3: Reduced fault set considered for case 3

State No.	Fault
1	No Fault (NF)
2	10% air leakage in intake manifold
3	20% air leakage in intake manifold
4	EGR valve stuck in 50% closed position
5	EGR valve stuck in 100% closed position
6	Temperature sensor 20% over reading
7	Temperature sensor 20% under reading
8	Pressure sensor 20% over reading
9	Pressure sensor 20% under reading

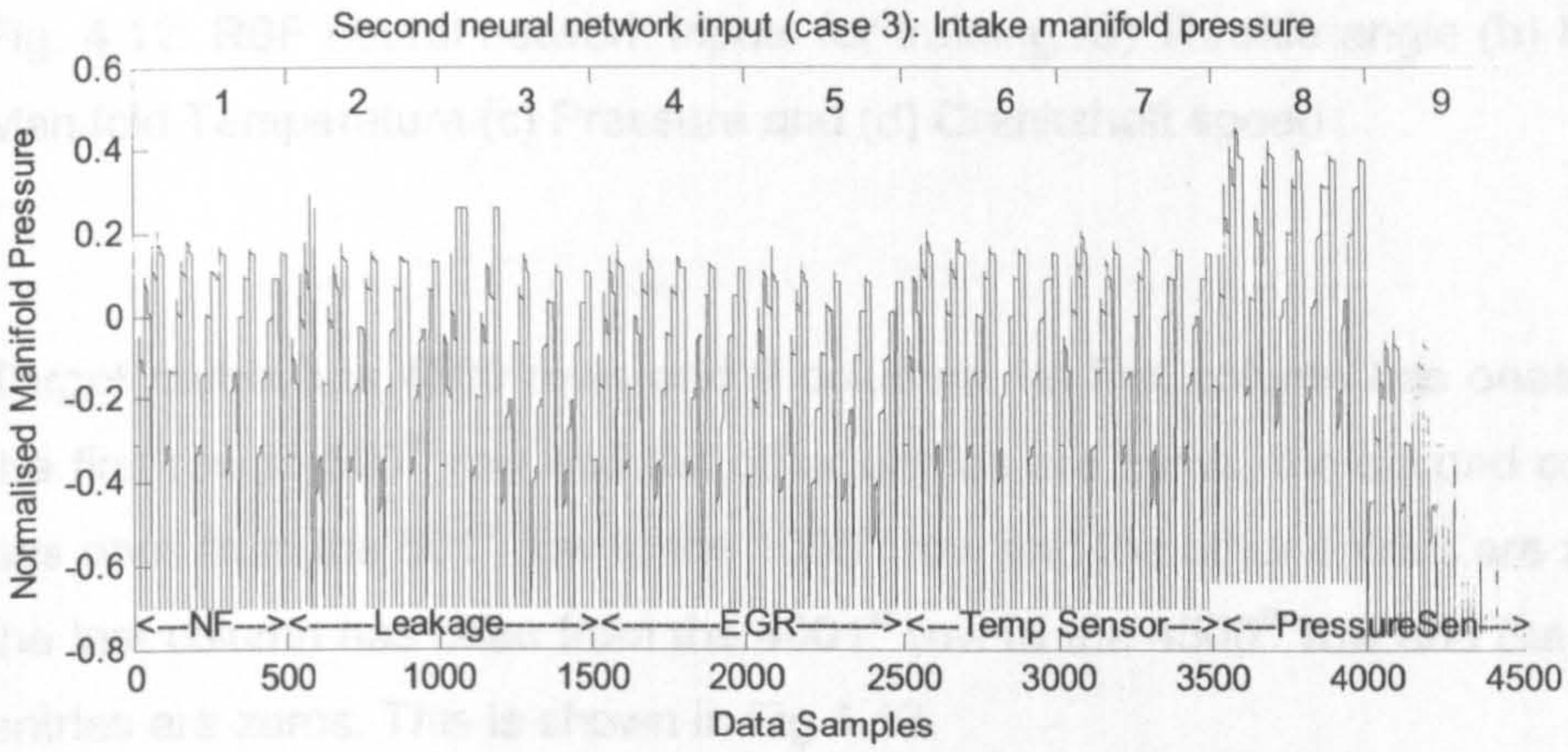
This reduces the size of the training data matrices to 4x4500 and 9x4500. MATLAB could easily handle this data size and an RBF is trained with 65 hidden nodes chosen by *K*-means method. The RBF network size is 4x65x17. Subsequently the trained RBF network is tested for different sets of seen and unseen throttle angles with different sets of noisy data as shown in Table 4.4. Up to 2% measurement noise is added to the test data to check robustness of the classifier against measurement noise. The neural network inputs for training with a reduced set of faults are graphically shown in Fig. 4.12.

Table 4.4: Testing data for case 3

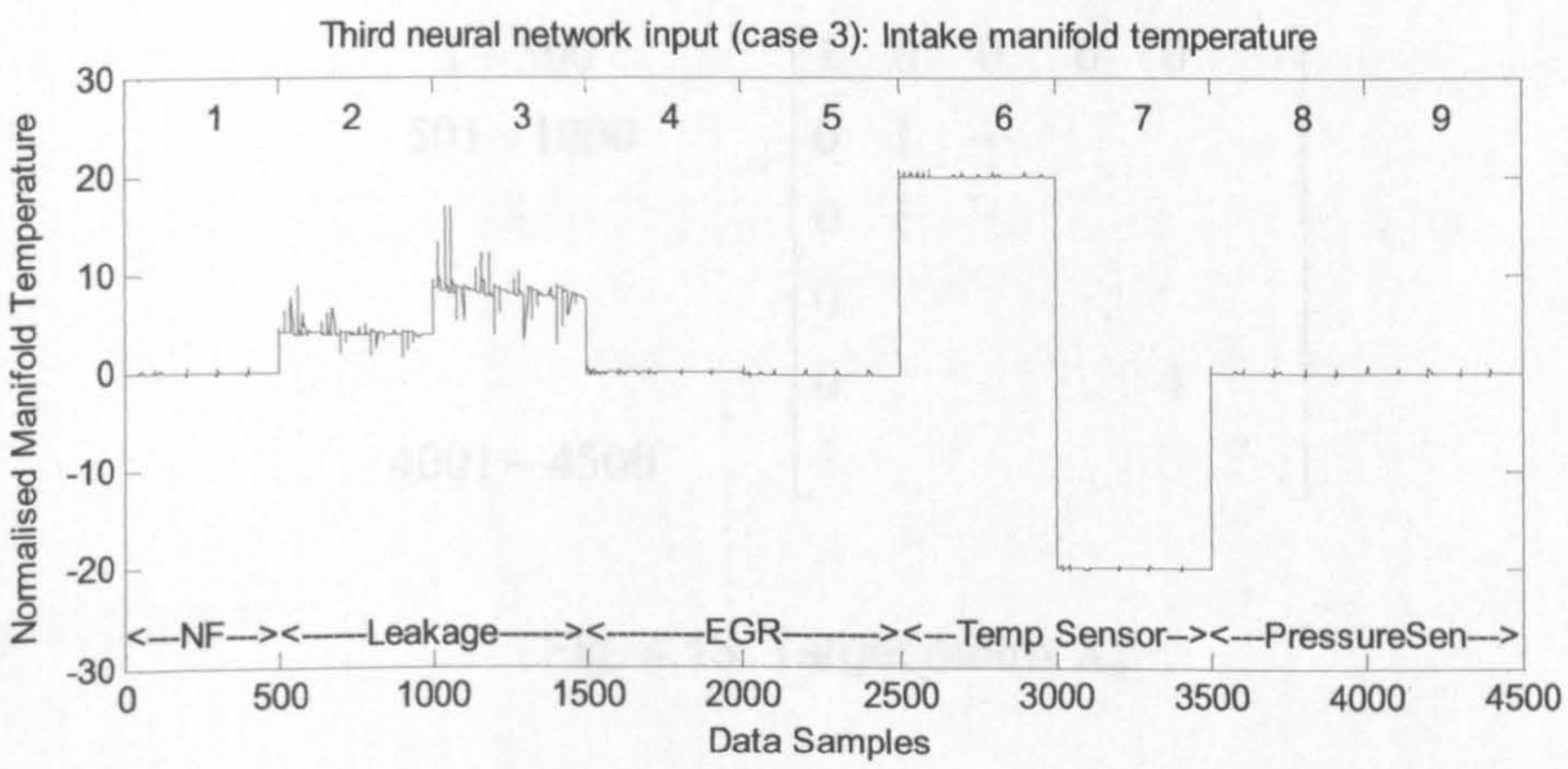
Initial throttle angle	Final throttle angle	Remarks
22	22	Seen throttle angle
26	26	Seen throttle angle
38	30	Seen throttle angles
24	30	Partially seen throttle angles
28	32	Unseen throttle angles
36	28	Unseen throttle angles


First neural network input (case 3): Throttle angle
Normalised Throttle Angle
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
1
2
3
4
5
6
7
8
9
<---NF---><-----Leakage------><--------EGR-------><---Temp Sensor---><---PressureSen--->
0
500
1000
1500
2000
2500
3000
3500
4000
4500
Data Samples

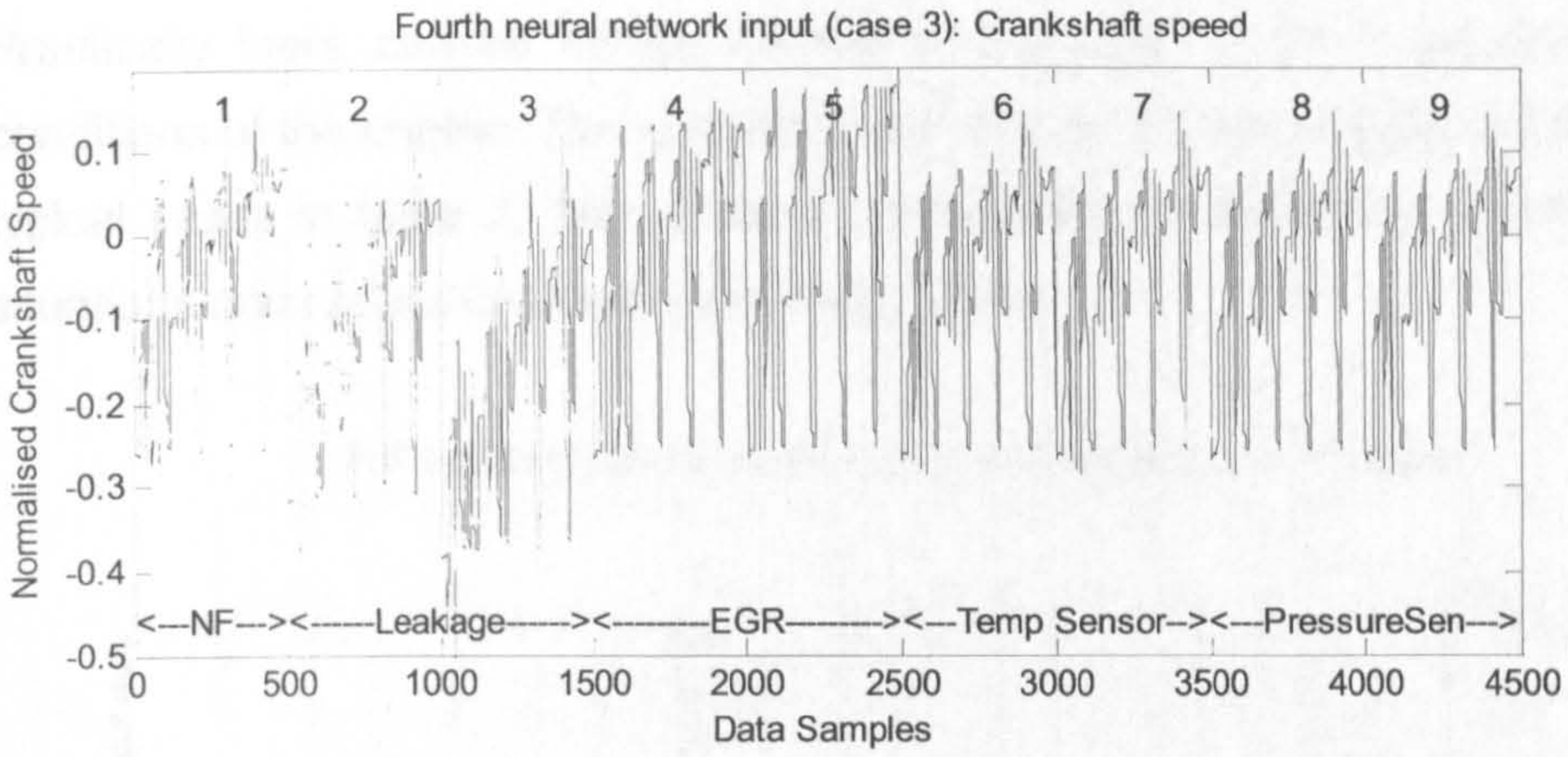
(a)


Second neural network input (case 3): Intake manifold pressure
Normalised Manifold Pressure
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
1
2
3
4
5
6
7
8
9
<---NF---><-----Leakage-----><-------EGR-------><---Temp Sensor---><---PressureSen--->
0
500
1000
1500
2000
2500
3000
3500
4000
4500
Data Samples

(b)


Third neural network input (case 3): Intake manifold temperature
Normalised Manifold Temperature
30
20
10
0
-10
-20
-30
1
2
3
4
5
6
7
8
9
<---NF---><-----Leakage-----><-------EGR-------><---Temp Sensor--><---PressureSen--->
0
500
1000
1500
2000
2500
3000
3500
4000
4500
Data Samples

(c)



(d)

Fig. 4.12: RBF neural network inputs for training (a) Throttle angle (b) Intake Manifold Temperature (c) Pressure and (d) Crankshaft speed

Target matrix has 4500 rows and 9 columns. Its first column has ones from the first row to 500th row and the other entries are zeros, the second column has ones from the 501st row to the 1000th row and the other entries are zeros, the last column has ones from the 4001st row to the 4500th row and the other entries are zeros. This is shown in Fig 4.13.

Row Numbers	X_o
$1 \sim 500$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 1 & \cdots & & & \\ 0 & \vdots & \ddots & & & \\ 0 & & & \ddots & & \\ 0 & & & & 1 & \\ \vdots & & & & & \ddots \end{bmatrix}$
$501 \sim 1000$	
$\vdots$	
$4001 \sim 4500$	

Fig. 4.13: Target matrix X_o

Test results show that the trained network could interpolate the unseen data well and classifies all the faults with few misclassifications. The result for the faults occurring when the initial condition of throttle angle is 24° and final condition is 30°, are shown in Fig. 4.14 and when the initial condition of

throttle angle is 28° and final condition is 32°, are shown in Fig. 4.15. Practically there can be infinite number of possibilities for initial and final conditions of the engine. The network is trained for the five equally distributed typical cases in table 2. The network interpolates the remaining in-between situations and results in proper fault classification.

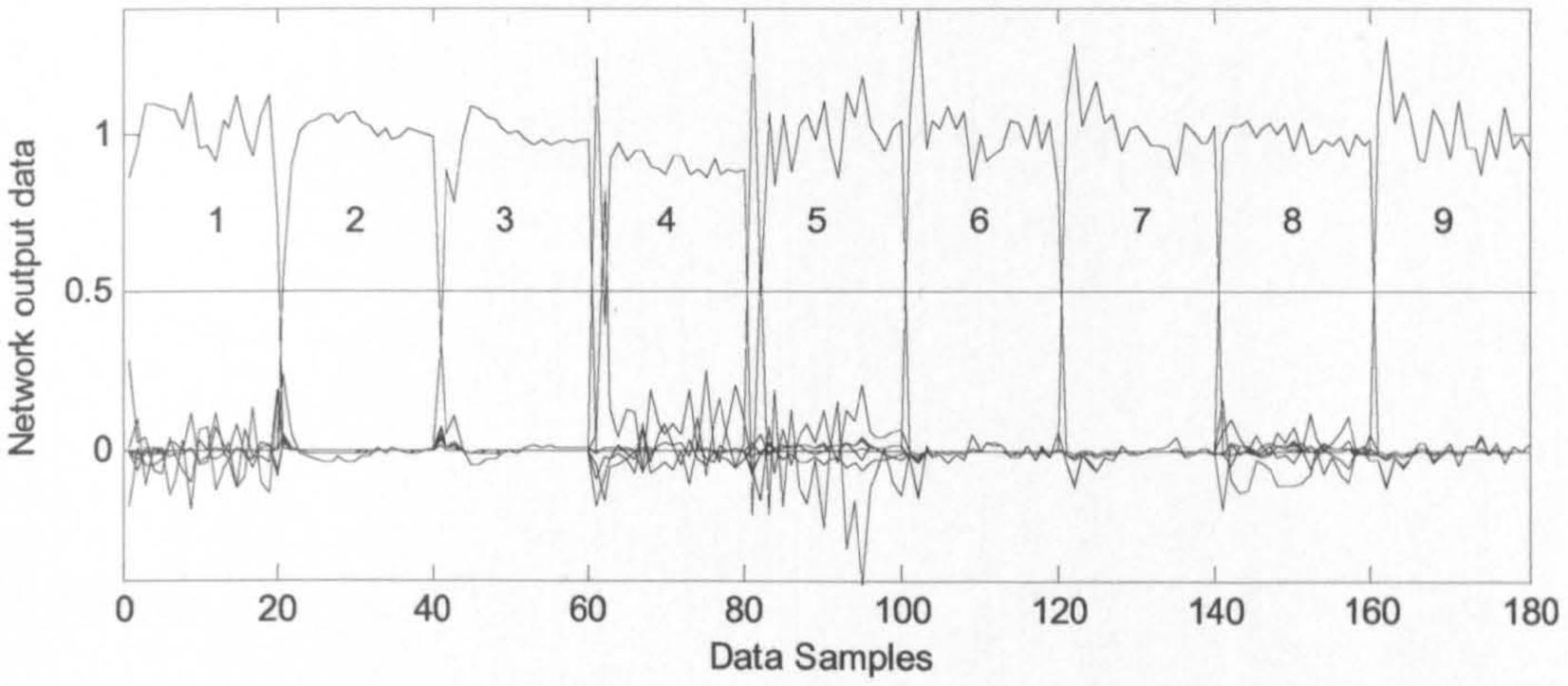


Fig. 4.14: RBF test result 1 for eight faults for case 3

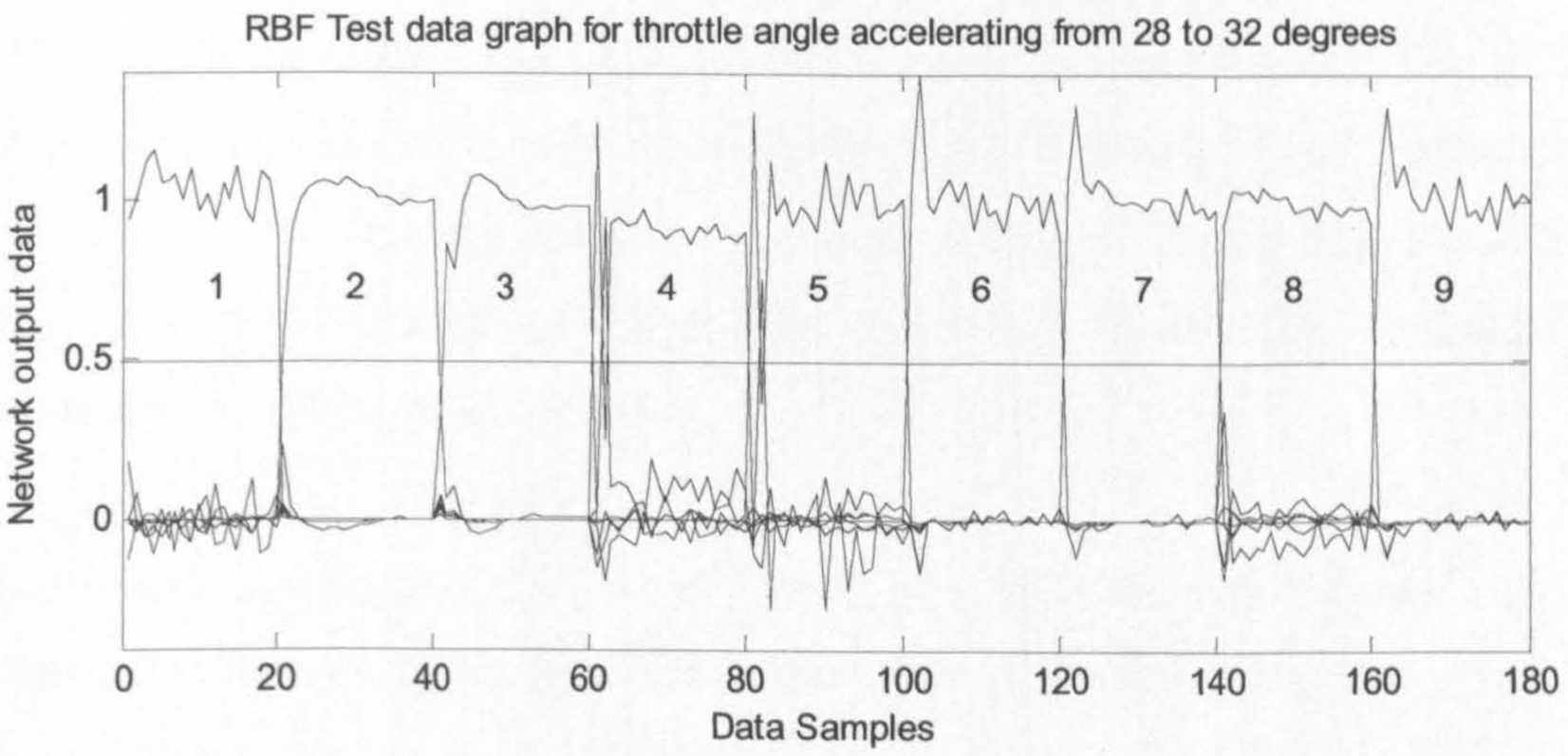


Fig. 4.15: RBF test result 2 for eight faults for case 3

4.6 DISCUSSION

A neural network classification approach with off-line training for fault diagnosis in automotive engines is investigated in this chapter. The technique is first demonstrated for classifying 16 fault states in three different engine operating conditions. The simulated faults can be detected and isolated for different types and for different intensities for case 1 and 2 but the classifier

could not classify faults for a more general case i.e. case 3. It is further investigated for a more general case with a smaller fault-set in section 4.5.3. Here, a neural network is trained for eight fault conditions in order to reduce the size of the training data. The trained RBF neural network is found to interpolate the between samples and is tested for unseen data sets. The results are acceptable.

CHAPTER 5

FDI WITH ADAPTIVE CLASSIFICATION

5.1 INTRODUCTION

In the previous chapter, a non-adaptive classification algorithm was investigated and it was found that although it worked fine for a smaller training data set but it was unable to handle larger training data sets satisfactorily. Therefore, the algorithm was found unsuitable for larger fault-set to be classified. Another problem with non-adaptive algorithm was that it trained off-line and as result it was unable to cope with environmental changes, disturbances and uncertainties. Therefore the need for an efficient adaptive classification algorithm was felt.

In this chapter, a new on-line FDI scheme is proposed for automotive engines using an adaptive neural network classifier. It has the following three salient features:

(a) Using the strong nonlinear mapping (classifying) ability of the ANN to cope with the multivariable, severe nonlinearity of engine dynamics.
(b) The classifier is made adaptive to cope with the significant parameter uncertainty, disturbance and environment change; and
(c) On-line fault diagnosis which can be directly implemented in an on-board diagnosis system.

During operation, the network classifier learns parameter changes in the engine due to aging or environment change. It can also adapt to engine-to-

engine differences within a batch of products. Gaussian radial basis function (RBF) neural nets are used for this purpose and both weights and widths are adapted on-line. Every sample of engine data is first tested for a fault and then used to update the neural network.

The proposed approach is applied to diagnose same simulated faults in an SI engine air path as discussed in the previous chapter. The faults are simulated in MVEM as explained before e.g. two component and two sensor faults with four different levels of intensities. The algorithm is compared with the previously used non-adaptive classifier and its robustness is demonstrated with simulation results.

5.2 ADAPTIVE ALGORITHMS

For training the non-adaptive RBF network and the initial off-line training of the adaptive network, the *K*-means, *P*-nearest neighbours and batch least squares (BLS) algorithms were used. When the adaptive classifier is used on-line, the centres remain fixed, as they have been chosen distributed in the whole operating space, while the widths and weights are adapted to minimise the classification error caused by any time-varying dynamics and model uncertainty. The widths are adapted using a gradient descent algorithm and the weights are adapted using the recursive least squares (RLS) algorithm. These algorithms are reviewed and derived below.

5.2.1 Gradient Descent method for the widths

When the RBF network is used to model a nonlinear mapping or a dynamic system, the width in each hidden layer node is usually chosen as a constant using the *P*-nearest rule, or all widths are just chosen equal to the same value, as it is believed that the modelling error is not sensitive to the width. However, when the RBF network is used as a classifier, the classification is strongly sensitive to the Gaussian local function, which is mainly characterised by the width. Therefore, a gradient descent algorithm is derived to on-line adapt the widths to achieve a minimal objective function.

The width parameter adaptation algorithm is explained here. The predictive value of RBF output can be given by the following equation:

$$\hat{y}_j = w_{ij}\phi_1 + w_{2j}\phi_2 + ... + w_{n_h j}\phi_{n_h}$$

where $i = 1,....., n_h$ and n_h is the number of hidden nodes in the network and $j = 1,....., q$ and q is the number of outputs of the network. w represents the weights in output layer and ϕ is Gaussian basis activation function.

$$\hat{y} = [w_{1j} \quad w_{2j} \quad \cdots \quad w_{n_h j}] \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{n_h} \end{bmatrix}$$

$$\hat{y} = W^T \phi$$

The cost function for error is defined as

$$J = e_1^2 + ... + e_q^2 = \sum_{j=1}^{q} e_j^2 \tag{5-1}$$

where $e_j = y_j - \hat{y}_j$ is the j^{th} classifier output error and y_j is the j^{th} training target. Then, according to the gradient method, the i^{th} width can be adapted as

$$\rho_i(k+1) = \rho_i(k) - \alpha \frac{\partial J(k)}{\partial \rho_i(k)}, \quad i = 1, \cdots, n_h \tag{5-2}$$

where α is a learning factor and $0 < \alpha < 1$. The gradient can be easily derived from equations (4-3)-(4-5) and (5-1),

$$\frac{\partial J}{\partial \rho_i} = \sum_{j=1}^{q} \left[\frac{\partial J}{\partial e_j} \cdot \frac{\partial e_j}{\partial \hat{y}_j} \cdot \frac{\partial \hat{y}_j}{\partial \phi_i} \cdot \frac{\partial \phi_i}{\partial \rho_i} \right] \tag{5-3}$$

Partially differentiating J with respect to e_j; from equation (5-1)

$$\frac{\partial J}{\partial e_j} = 2e_j \qquad j = 1, \ \ldots\ldots, \ q \tag{5-3a}$$

e_j is defined as $e_j = y_j - \hat{y}_j$ and therefore

$$\frac{\partial e_j}{\partial \hat{y}_j} = 0 - 1 = -1 \tag{5-3b}$$

Partial derivative of $\hat{y}_j$ with respect to ϕ_i is:

$$\frac{\partial \hat{y}_j}{\partial \phi_i} = w_{ij} \qquad i = 1, \ \ldots\ldots, \ n_h \ \ and \ \ j = 1, \ \ldots\ldots, \ q \tag{5-3c}$$

Gaussian basis activation function ϕ is given as

$$\phi_i = e^{-\left(\frac{\|x - C_i\|^2}{\rho_i^2}\right)} \qquad i = 1, \ \ldots\ldots\ldots, \ n_h$$

Therefore

$$\frac{\partial \phi_i}{\phi \rho_i} = e^{-\left(\frac{\|x - C_i\|^2}{\rho_i^2}\right)} \cdot \left(\left(-\|x - C_i\|^2\right) \ . \ (-2) \ . \ \frac{1}{\rho_i^3}\right)$$

$$\frac{\partial \phi_i}{\partial \rho_i} = \phi_i . 2 . \frac{\|x - C_i\|^2}{\rho_i^3} \tag{5-3d}$$

The four partial derivatives derived in equations (5-3a)-(5-3d) are substituted into (5-3) giving,

$$\frac{\partial J}{\partial \rho_i} = \sum_{j=1}^{q} \left[-4 e_j w_{ij} \phi_i \frac{\|x - C_i\|^2}{\rho_i^3} \right]$$

Or

$$\frac{\partial J}{\partial \rho_i} = -4\phi_i \frac{\|x - c_i\|^2}{\rho_i^3} \sum_{j=1}^{q} (e_j w_{ij})$$

Substituting this back in equation (5-2), we have

$$\rho_i(k+1) = \rho_i(k) + 4.\alpha.\phi_i(k) \frac{\|x(k) - c_i\|^2}{\rho_i^3(k)} \sum_{j=1}^{q} [e_j(k) w_{ij}(k)] \qquad (5\text{-}4)$$

This training algorithm can update the width parameters to minimise the sum of squared error defined in (5-1).

5.2.2 Recursive Least Squares Algorithm

If the RBF network has *d* inputs, *q* outputs and n_h hidden nodes, the output matrix with *N* samples ($\hat{Y}^{N\times q}$) can be written as

$$\hat{Y} = \Phi(X)W$$

where $X^{N\times d}$ is the input matrix, $\Phi(X)^{N\times nh}$ is the matrix of activation function outputs and $W^{nh\times q}$ is the matrix of weights. The RLS method is used for on-line training. The RLS algorithm given in (Ljung, 1999) is summarized as follows.

$$L(k) = \frac{P(k-1)\varphi(k)}{\lambda(k) + \varphi^T(k)P(k-1)\varphi(k)}$$

$$W(k) = W(k-1) + L(k)\left[y^T(k) - \varphi^T(k)W(k-1)\right] \qquad (5\text{-}5)$$

$$P(k) = \frac{1}{\lambda(k)}\left[P(k-1) - \frac{P(k-1)\varphi(k)\varphi^T(k)P(k-1)}{\lambda(k) + \varphi^T(k)P(k-1)\varphi(k)}\right]$$

where $y(k)$ and $\phi(k)$ represent the network target vector and hidden layer output vector at simple instant *k* respectively. Also, $y^T(k)$ and $\varphi^T(k)$ are k^{th} row vector in matrix *Y* and Φ. *P*(k) is the covariance matrix and *L*(k) is the gain matrix, $\lambda(k)$ is called the forgetting factor and lies in the range of (0,1).

The parameters $L(k)$, $W(k)$ and $P(k)$ are updated orderly with the activation function outputs $\varphi(k)$ at each sampling time after setting the value of $\lambda(k)$ and the initial values of *W* and *P*.

5.3 ON-LINE RBF CLASSIFIER ADAPTATION

The main contribution of adaptive algorithm is that while the fault classifier diagnoses faults on-board, the classifier is adapted on-line so that the model-plant mismatch, parameter uncertainty and especially the time varying dynamics caused by mechanical wear of components and environment change can be modelled. In this way the classification error and consequently the false alarm rate can be greatly reduced. In fact, these effects are main drawbacks for the fixed parameter neural network to be used practically.

The fault classification and on-line adaptation can be implemented as follows. Firstly, the measurements are read into the electronic control unit (ECU). Then, the data is fed into the classifier to diagnose faults. After this, the target will be modified according to whether a fault or several faults are detected. If a fault is detected the on-line training target vector will be changed to the target vector corresponding to the occurred fault. Then the measurements and the modified target are used to update the classifier. In the adaptation, the width in each hidden node is adapted using the gradient descent algorithm in (5-4) and the centre locations remain fixed as previously described. This is followed by adaptation of the weights using the RLS algorithm (Ljung, 1999) in (5-5).

There are two points that should be addressed. One is to identify the occurrence of a fault. To reduce the effect of peak noise on the fault detection so as to reduce the false alarm rate, the mean absolute modelling error for each classifier output is calculated for the previous *M* samples as the residual, i.e.

$$r_j = \frac{1}{M}\sum_{i=k-M+1}^{k}\left|y_j(i)-\hat{y}_j(i)\right|, \quad j=1,\cdots q \tag{5-6}$$

and a fault is believed to be fired when

$$r_j \geq r_t \tag{5-7}$$

where k is the sample instant, r_j is the residual and r_t is the threshold to be designed according to the noise level. Another point is that a multi-epoch training of the width in one sample period using the gradient descent method is employed. It was found that a single iteration updating with the gradient descent method would not reach the minimum if the learning rate is chosen small, while a large learning rate will cause unstable convergence. The recursive updating of the widths runs until the following is satisfied,

$$\left|\frac{\partial J}{\partial \rho_i}\right| \leq \sigma, \quad i = 1, \cdots, n_h \tag{5-8}$$

where σ is a pre-specified small positive constant, or a pre-specified number of iterations is reached. The fault diagnosis and classifier adaptation within one sample period is illustrated in Fig. 5.1.

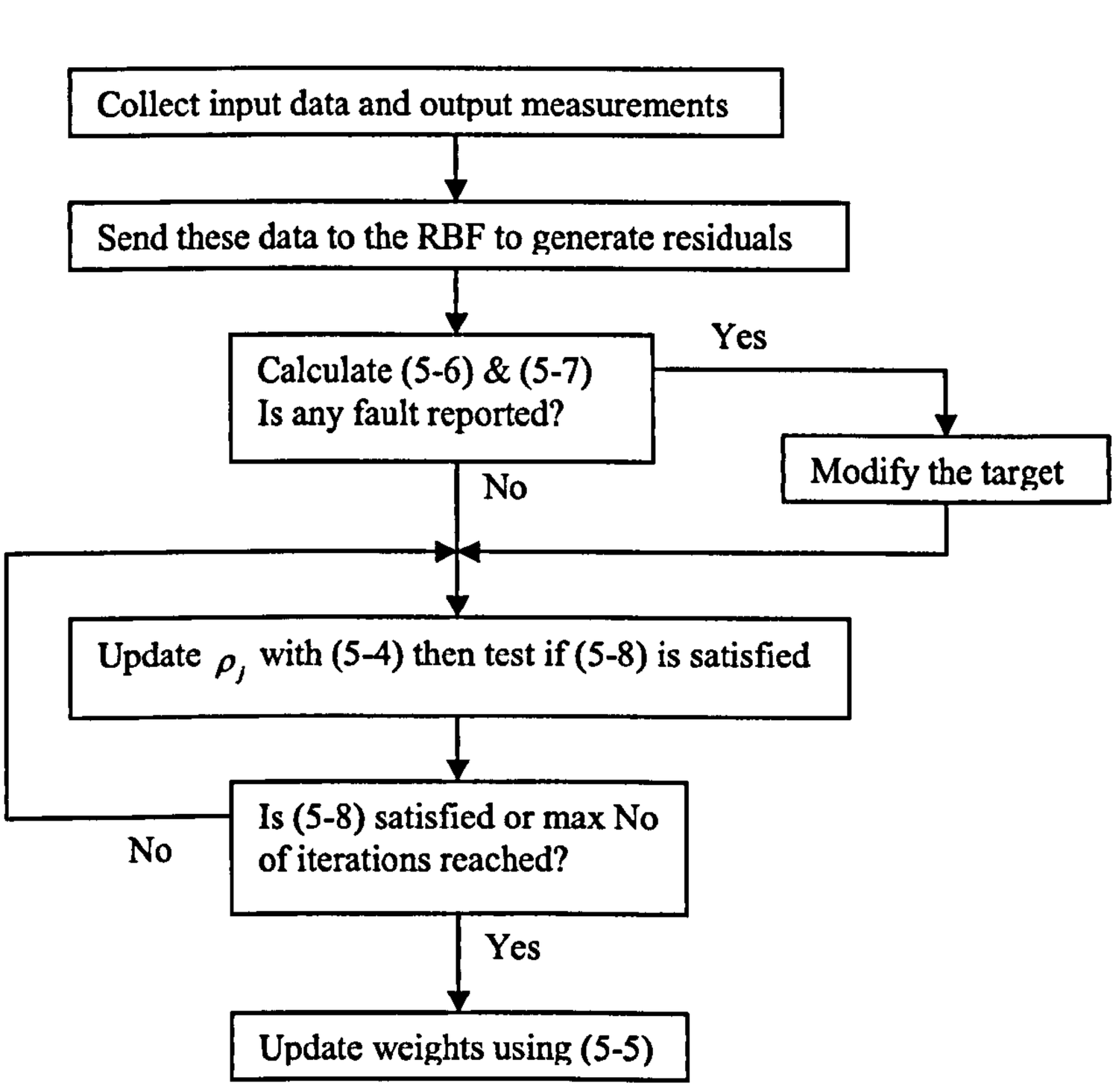


Fig. 5.1: Flow chart of fault diagnosis and classifier updating

5.4 Data Collection & Normalisation

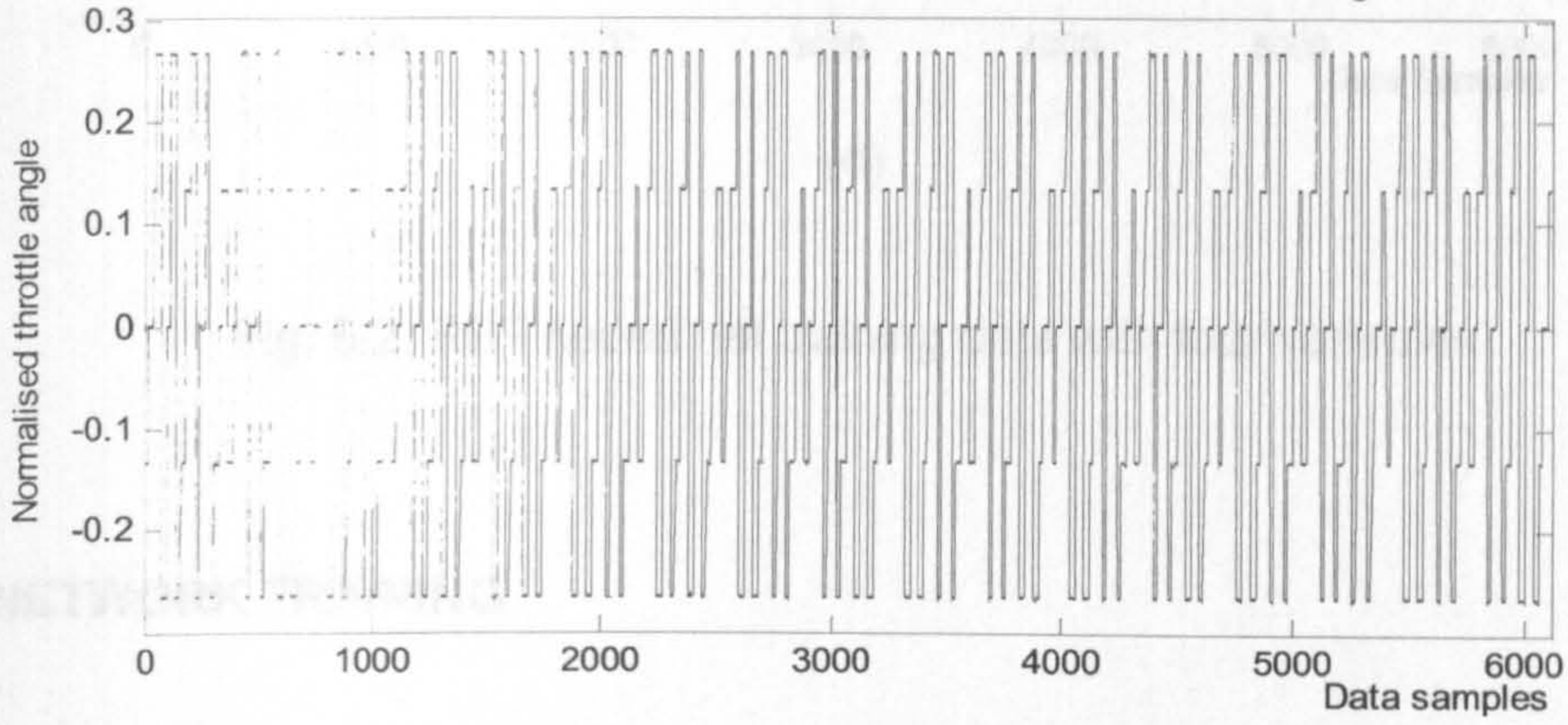
Data is collected and normalised as explained in Chapter 4 in section 4 . and 4.5.3 respectively. Twenty five data sets are presented in Table 4.2 which created a huge data set of 4x8500 for RBF training. In order to reduce the training data size, the data of constant speed was removed from the Table 4.2 as shown in Table 5.1. Therefore, Table 5.1 shows 4 data sets in a row instead of 5 data sets. The data sets of constant speed are removed because they were mere repetition of data because each data set contains first data sample as the constant speed data and therefore a separate set for constant speed was not required.

Table 5.1: Details of data sets collected for training and testing of RBF networks

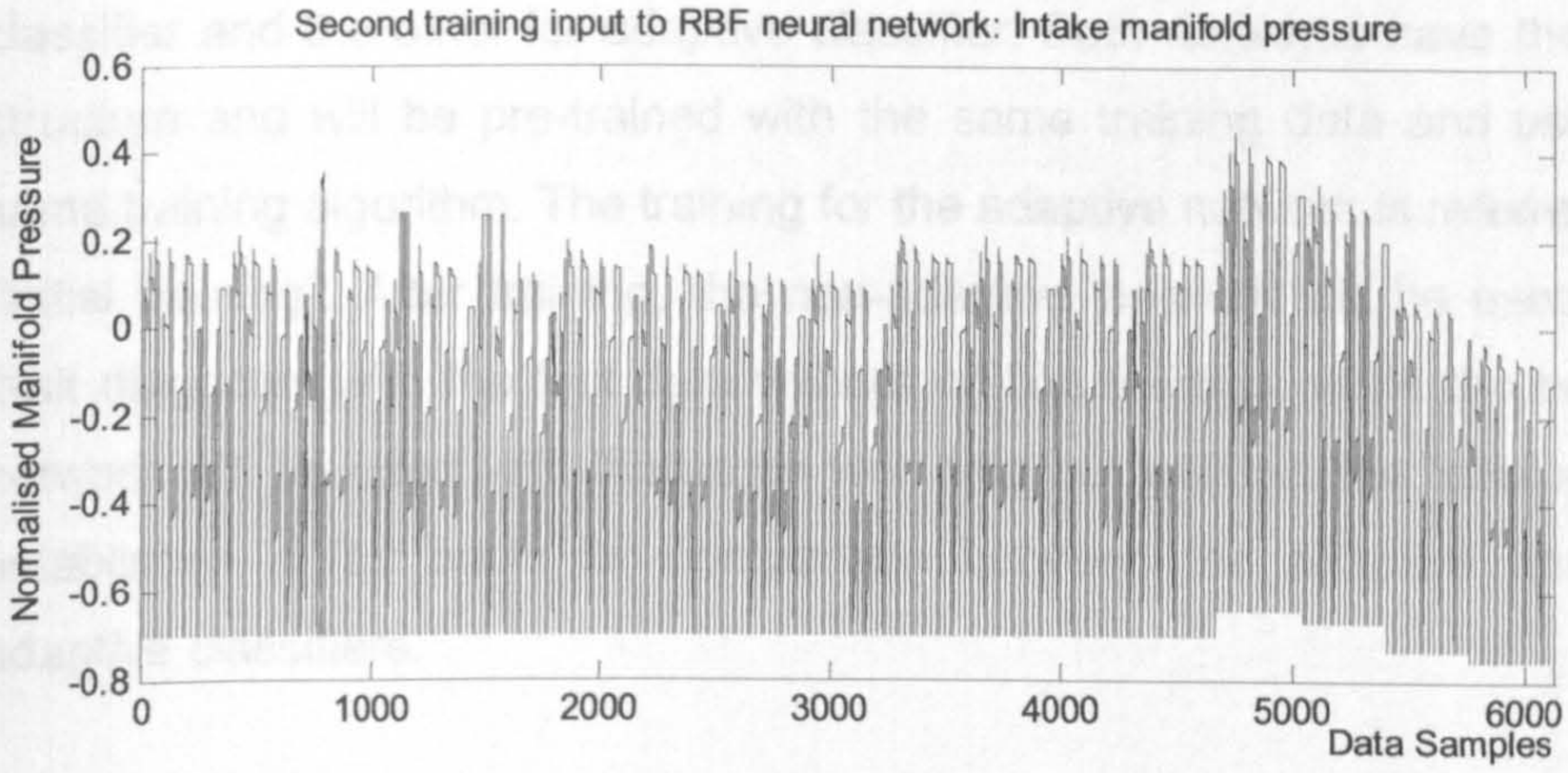
Initial Throttle Angle	Engine Speed Accelerating to throttle angle	Engine Speed Retarding to throttle angle	Number of Sets
22	26,30,34,38	Nil	4
26	30,34,38	22	4
30	34,38	22,26	4
34	38	22,26,30	4
38	Nil	22,26,30,34	4

It may be noted that when the engine accelerates with throttle angle increasing from 22° to 26°, the speed acceleration is different from that when it accelerates with throttle angle from 22° to 38° in the same period of time. Same is the case with deceleration and therefore the no fault and fault data is collected for different speed accelerations and decelerations. Data for all the four variables is collected i.e. throttle angle, manifold pressure, temperature and crankshaft speed. To evaluate the robustness of the developed method to measurement noise, a noise of normal distribution with zero mean and unity variance is added to the collected data. The amplitude of the noise is set to about 2% of the average of the signal amplitude to simulate the

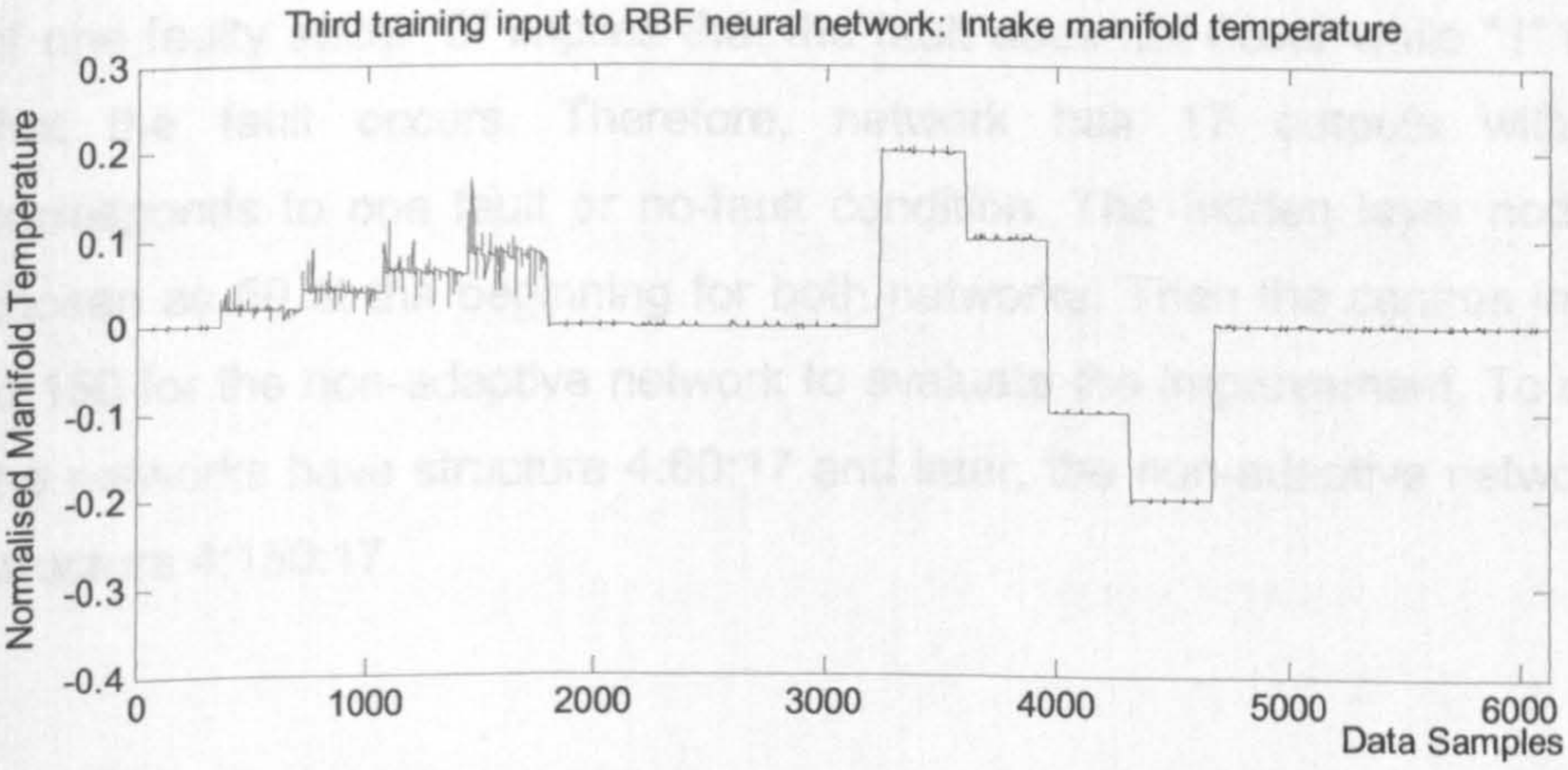
measurement noise expected in a real engine system. Out of 20 data sets, two data sets are reserved for testing and remaining 18 data sets are used for training of the RBF neural network. The four normalised variables used as training data (18*340 = 6120 samples) are graphically shown in Fig. 5.2. Clearly the size of training data is reduced by 28% as compared to 8500 samples initially used for training in section 4.5.3 in Chapter 4.



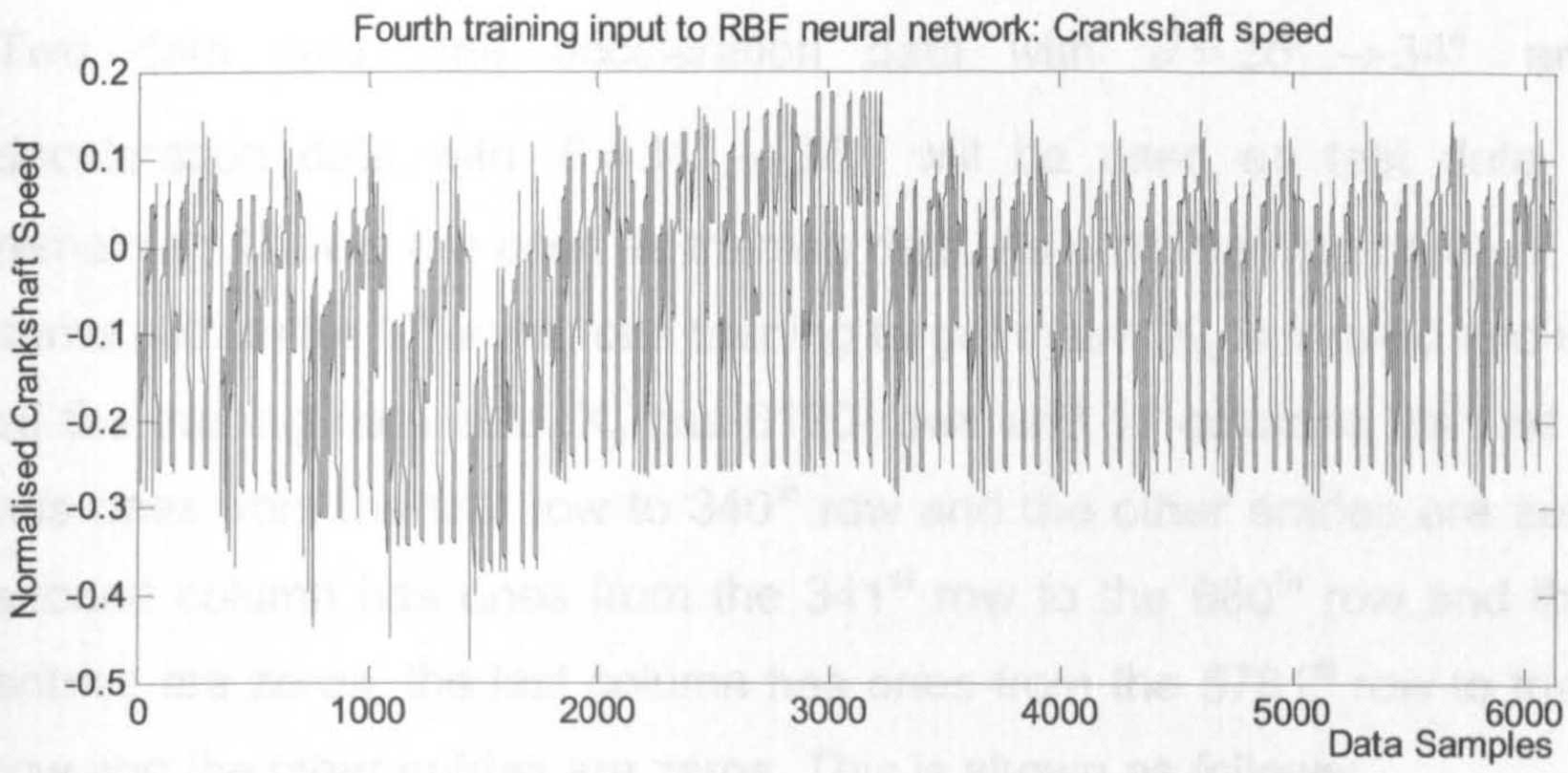
(a)



(b)



(c)



(d)

Fig. 5.2: RBF neural net training data with four variables

5.5 NETWORK TRAINING

Two RBF networks are used for fault classification, with one for non-adaptive classifier and the other for adaptive classifier. Both networks have the same structure and will be pre-trained with the same training data and using the same training algorithm. The training for the adaptive network is referred to as "initial training". After training, the non-adaptive network will be used to do fault diagnosis with the test data without on-line training, while the adaptive network will be used with the same test data but with on-line training. This establishes a fair basis for comparison between the adaptive and non-adaptive classifiers.

The network has 4 inputs. Each network output is used to indicate occurrence of one faulty state: "0" implies that the fault does not occur while "1" implies that the fault occurs. Therefore, network has 17 outputs with each corresponds to one fault or no-fault condition. The hidden layer nodes are chosen as 60 at the beginning for both networks. Then the centres increase to 150 for the non-adaptive network to evaluate the improvement. To outline, the networks have structure 4:60:17 and later, the non-adaptive network has structure 4:150:17.

Two data sets, one acceleration data with $\theta = 26^o \rightarrow 34^o$ and one deceleration data with $\theta = 34^o \rightarrow 26^o$ will be used as test data, all the remaining 18 sets are used as training data. As each training data set has the same pattern for 17 faults, one training target matrix X_o is formed and used for all the training data sets. X_o has 6120 rows and 17 columns. Its first column has ones from the first row to 340th row and the other entries are zeros, the second column has ones from the 341st row to the 680th row and the other entries are zeros, the last column has ones from the 5781st row to the 6120th row and the other entries are zeros. This is shown as follows:

$$
\begin{array}{cc}
\text{Row Numbers} & \qquad\qquad X_o \\
1 \sim 340 & \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 1 & \cdots & & & \\ 0 & \vdots & \ddots & & & \\ 0 & & & \ddots & & \\ 0 & & & & 1 & \\ \vdots & & & & & \ddots \end{bmatrix} \\
341 \sim 680 & \\
\vdots & \\
5781 \sim 6120 &
\end{array}
$$

Thus, the transpose of the i^{th} row in X_o is used as the training target vector for the i^{th} training pattern.

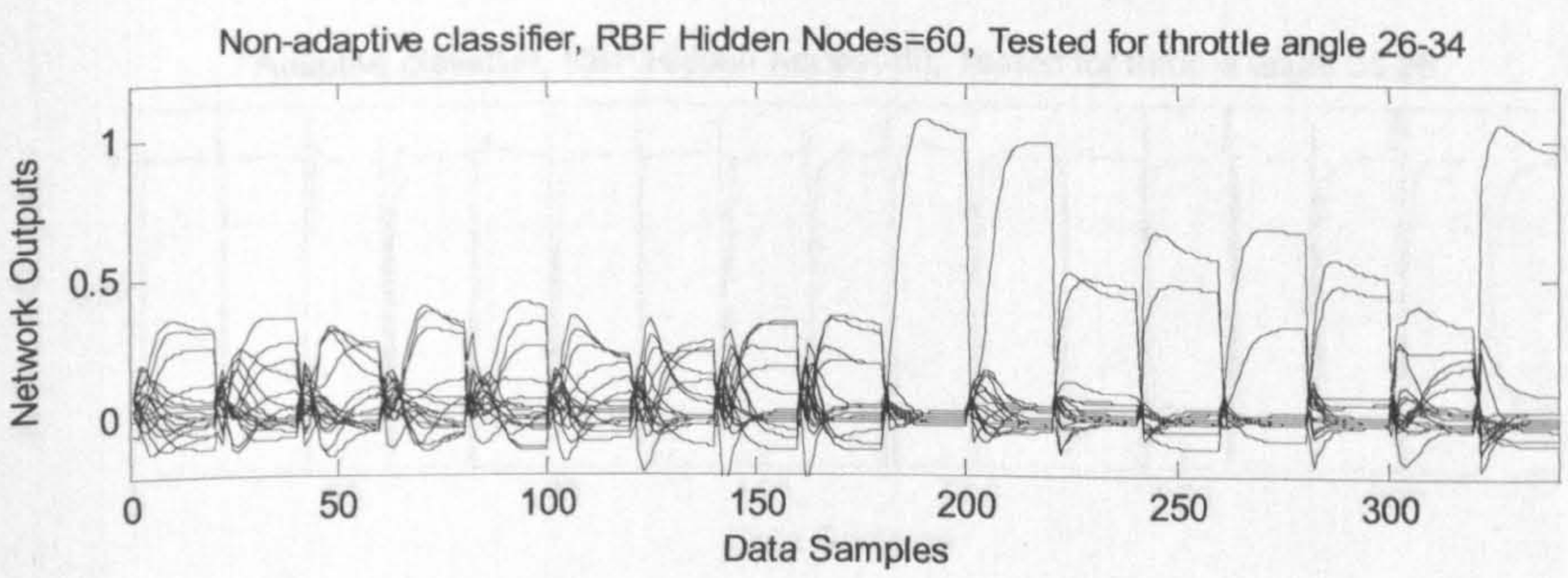
The 60 centres are chosen using the *K*-means clustering algorithm from the training data sets. The widths were chosen using the *P*-nearest neighbours' algorithm, and the weights were trained using the Recursive Least Squares algorithm.

5.6 FAULT DIAGNOSIS

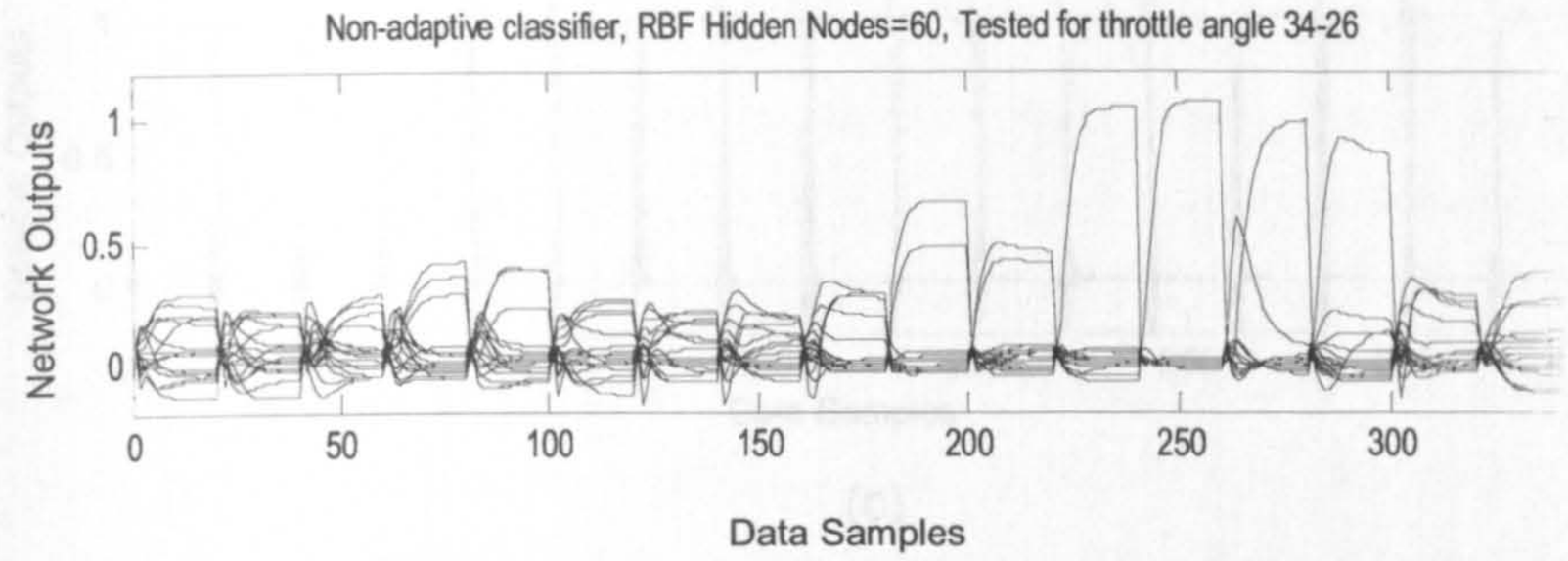
After training with the training data sets, the two networks are used to diagnose faults with the two test data sets described before and another data set when the throttle angle is constant, 30°. The classification results of the non-adaptive network are displayed in Fig. 5.3. The adaptive network is used to classify the faults in the on-line mode with network adaptation as illustrated in the flow chart in Fig. 5.1, where M in (5-6) was chosen as $M = 1$ (to see the residuals before the treatment), the fault detection threshold in (5-7) was

chosen as $r_t = 0.5$. High thresholds may lead to missed detections whilst low thresholds will cause false alarms. r_t is chosen as 0.5 by utilising experience in minimising false alarm rate. The simulation is run for different values of r_t and $r_t = 0.5$ (mean value of two target values i.e. 0 and 1) is chosen for minimal false alarm rate. Mathematically r_t should be a little bit higher than 0.5 according to the level of noise in the testing data. But $r_t = 0.5$ is found as a good compromise between reliability of detection and insensitivity to noise. The threshold for the gradient of the objective function in (5-8) was chosen as $\sigma = 0.00001$. The forgetting factor for the RLS algorithm in (5-5) was chosen a constant value of $\lambda = 0.99$. The diagnosis results are shown in Fig. 5.4. Then, the centres of the non-adaptive network are increased to 150. The fault detection results are displayed in Fig. 5.5. To see residuals more clearly, the outputs of the adaptive classifier in Fig. 5.4b are displayed separately in Fig. 5.6.

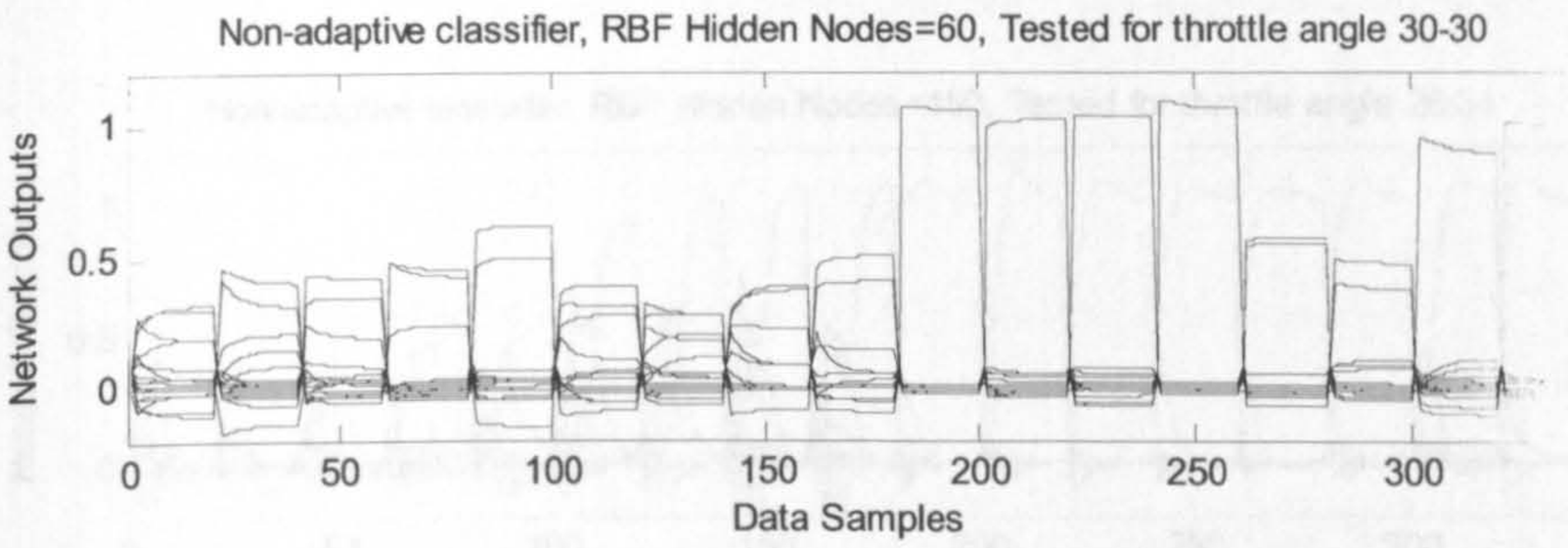
It is clearly seen in Fig. 5.3 that the non-adaptive classifier with 60 centres are not able to classify the simulated faults to an acceptable level. Whilst the adaptive network clearly classifies the faults (see Fig. 5.4 and 5.6) with just a few peak values that may cause false alarm when 0.5 is used as the fault detection threshold. Further more, if M in (5-6) is chosen as 3 or greater, the averaged residual will be greatly reduced and the false alarm rate will be zero. It can also be seen in Fig. 5.5 that even with the much more centres 150, though the performance is improved, it still has many misclassifications and is still not as good as the adaptive method. In addition, 150 centres greatly increase the computing load and are more difficult to implement in practice.



(a)

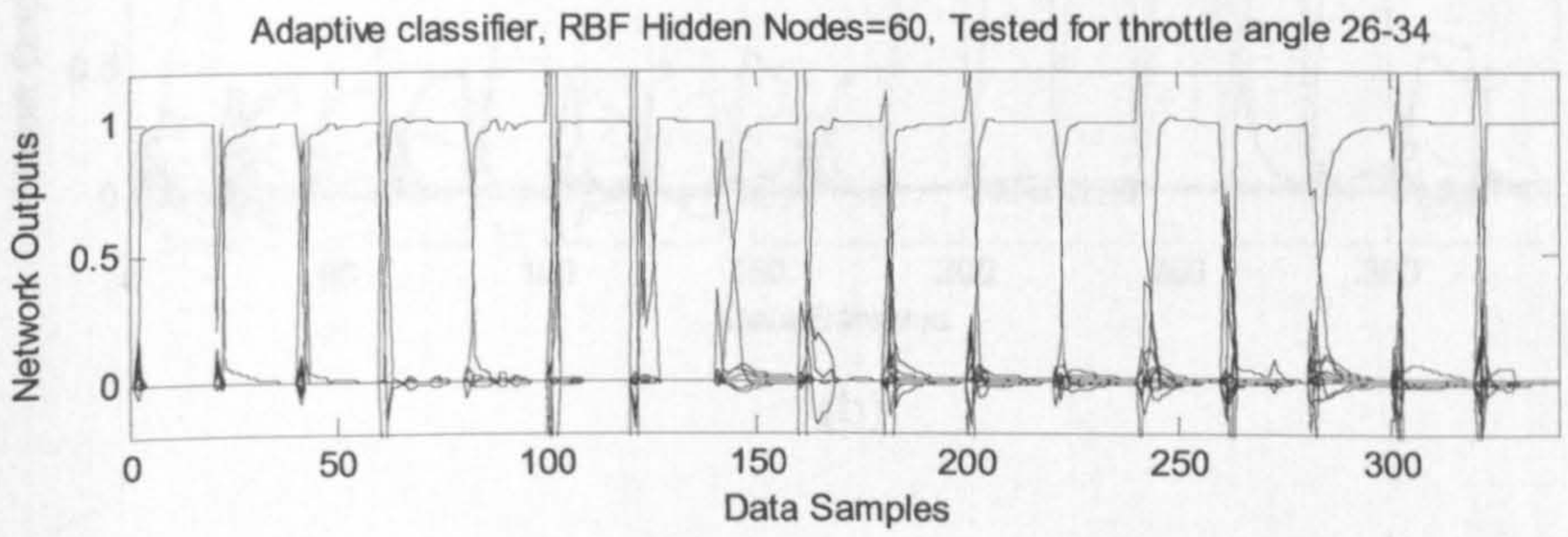


(b)

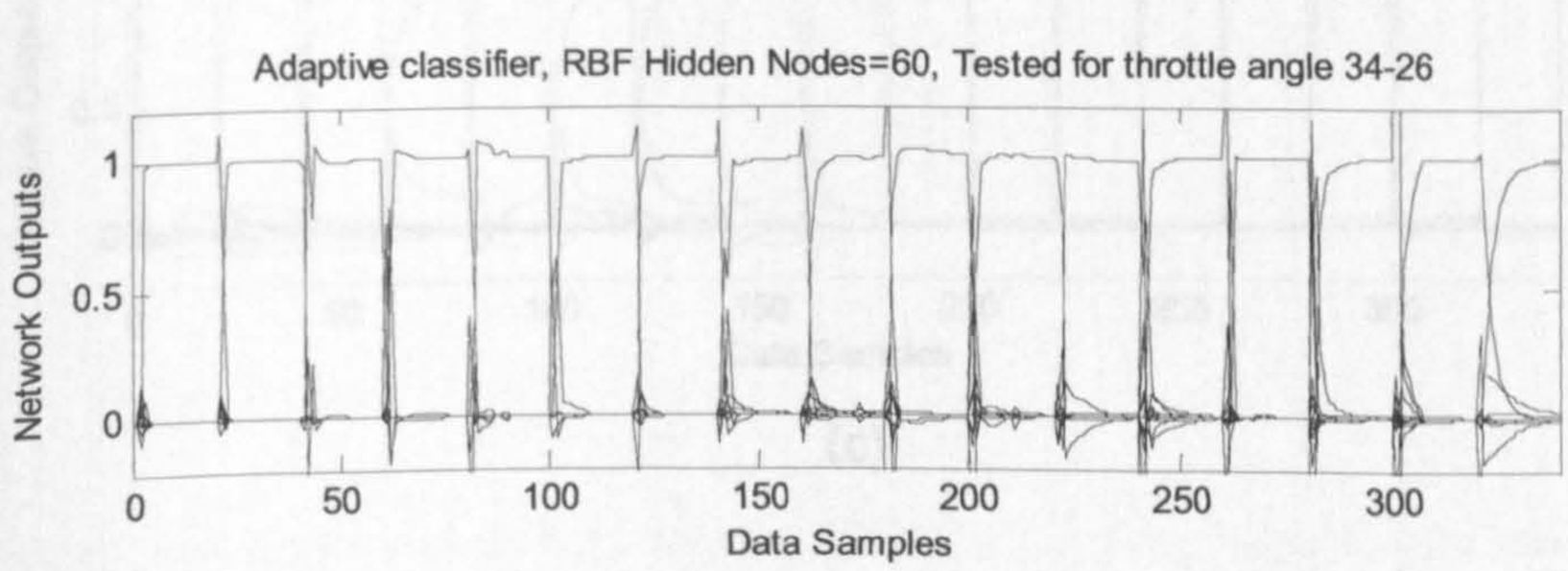


(c)

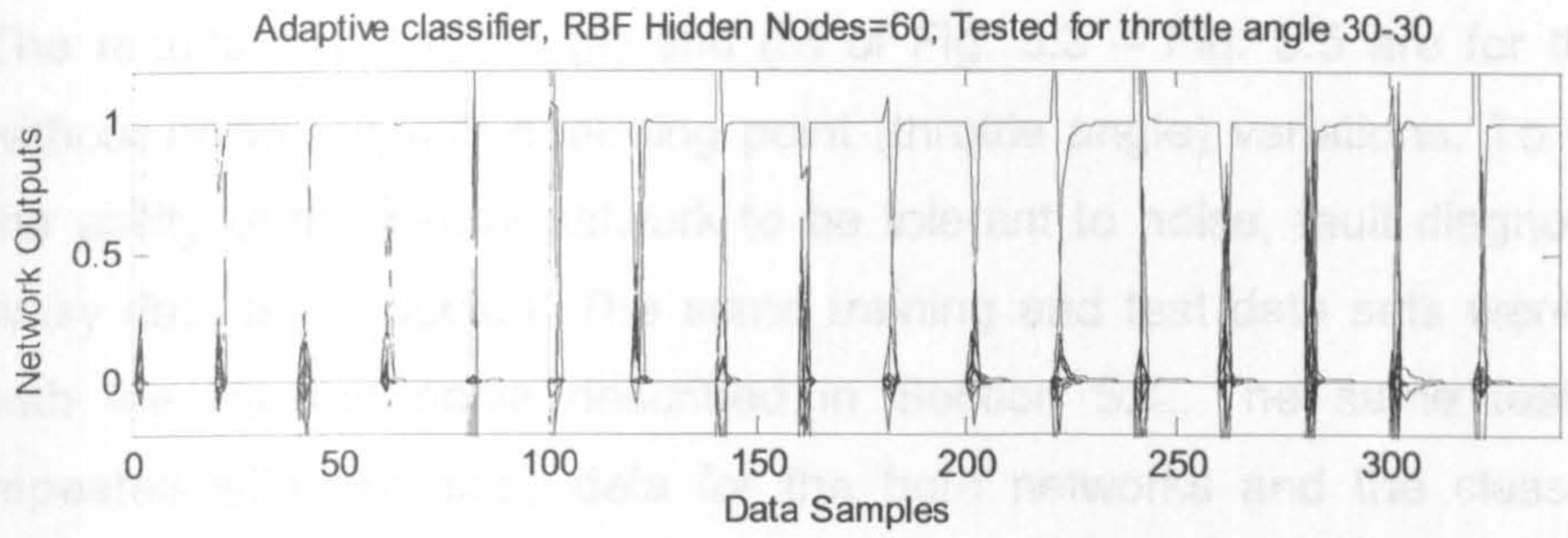
Fig. 5.3: Test results of the non-adaptive classifier with n_h = 60



(a)

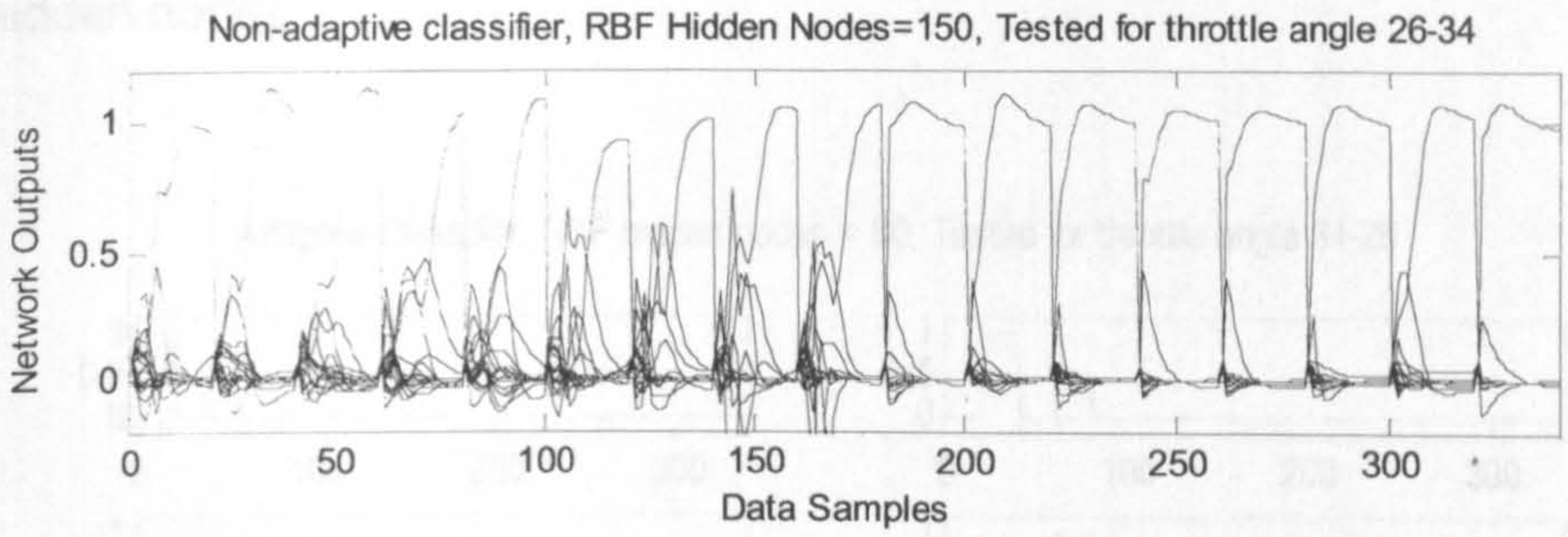


(b)

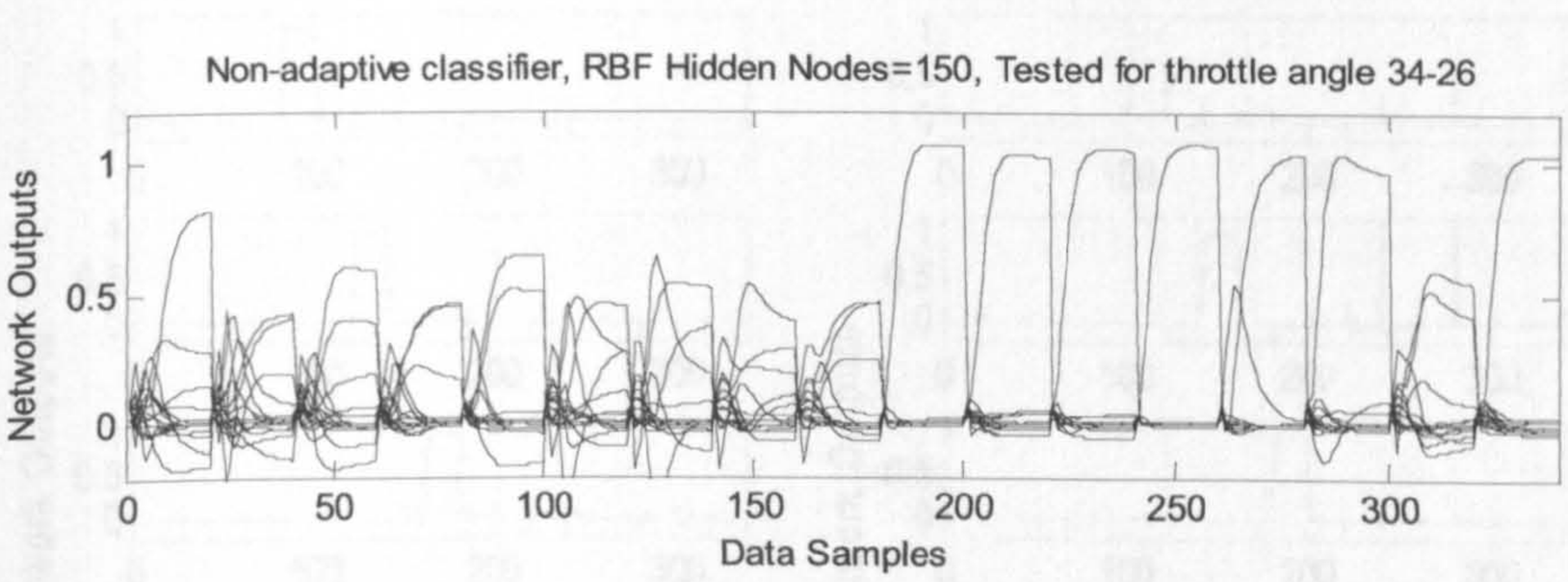


(c)

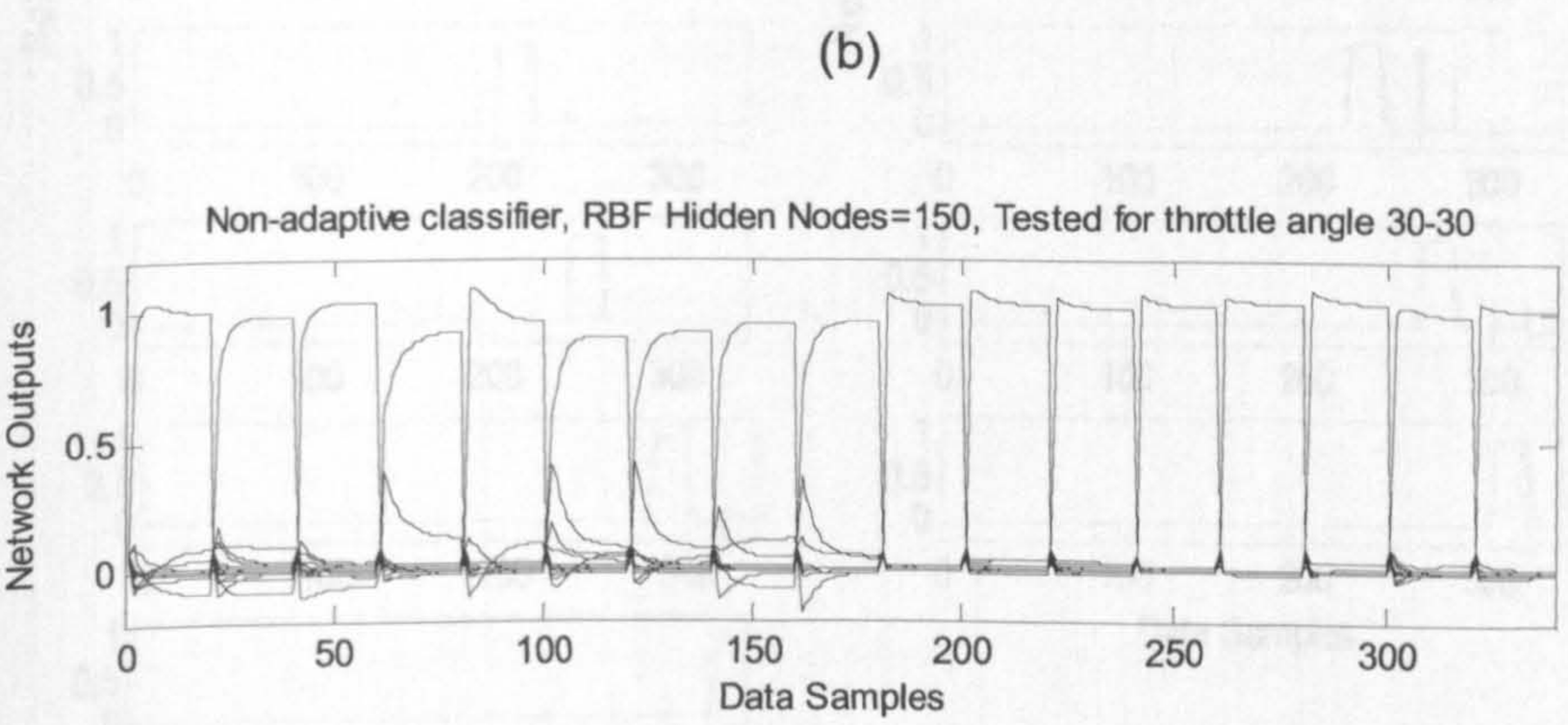
Fig. 5.4: Test results of Adaptive classifier n_h = 60



(a)



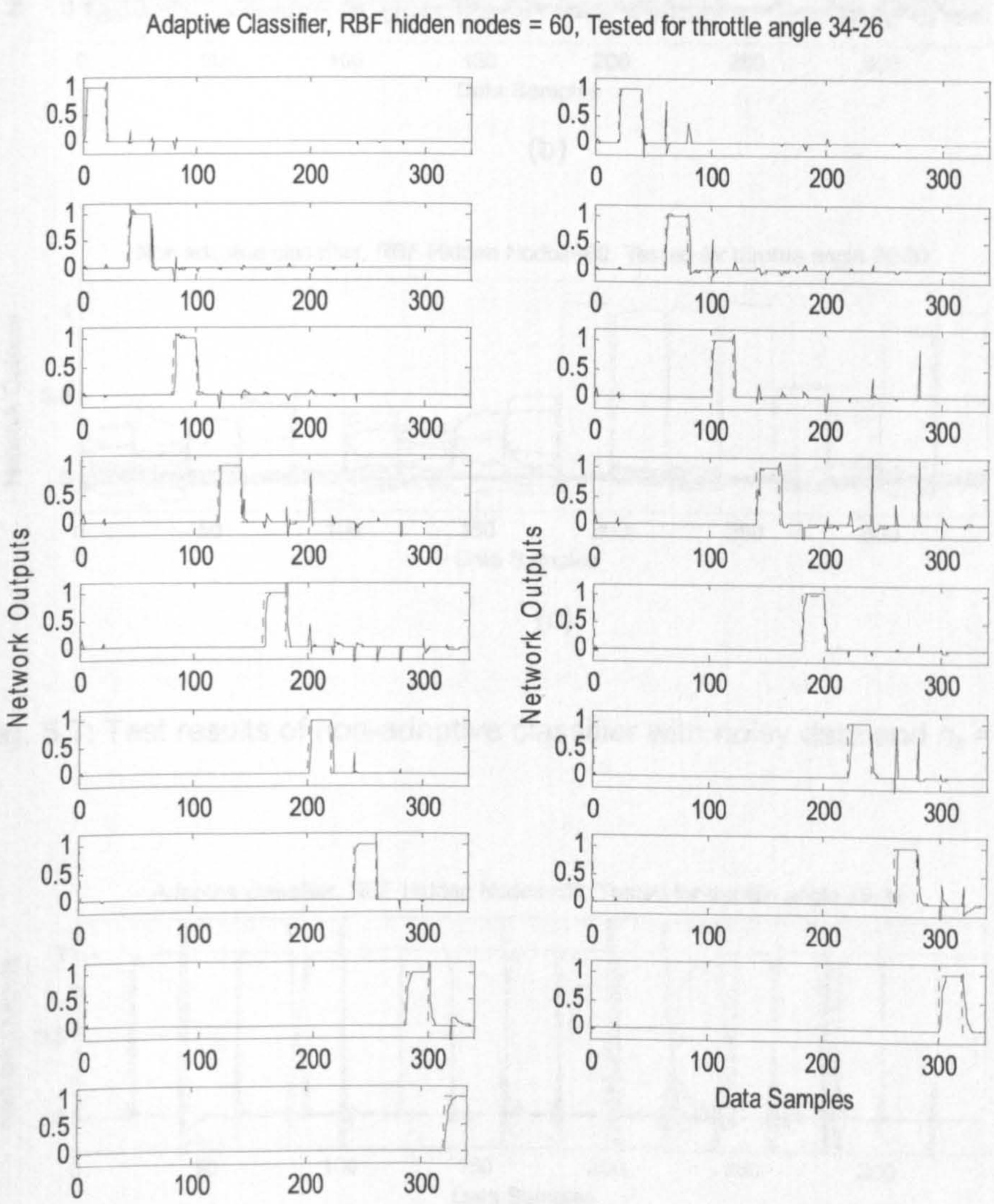
(b)



(c)

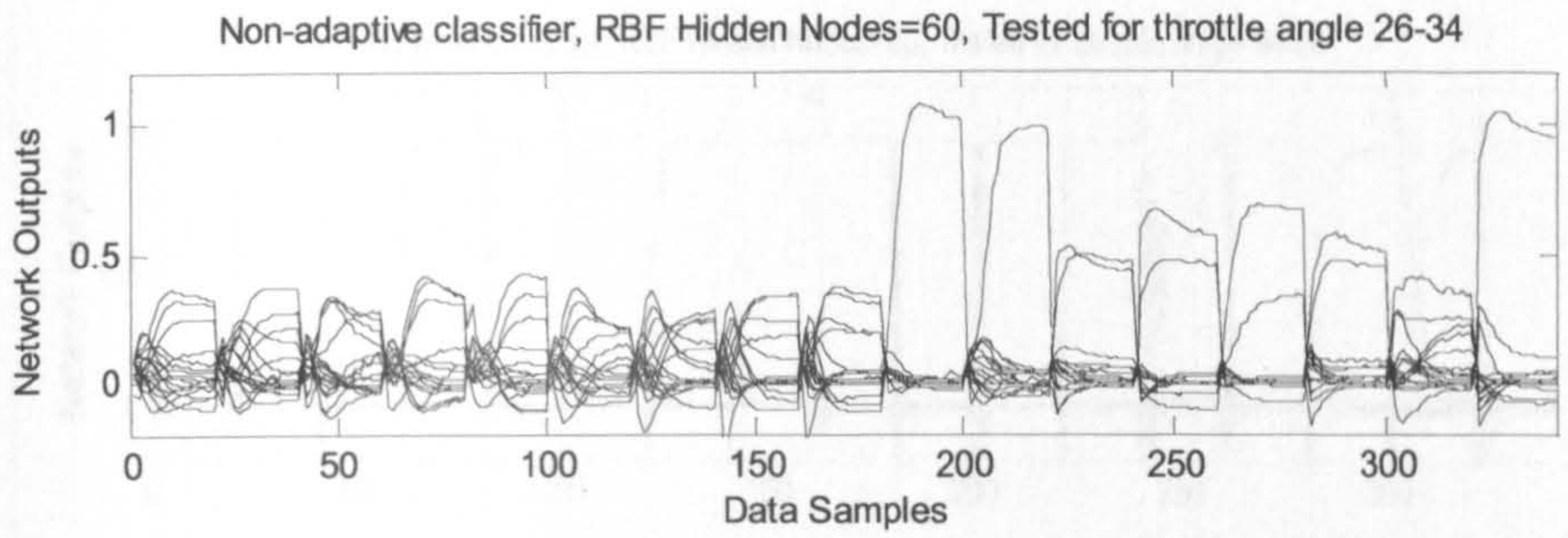
Fig. 5.5: Test results of non-adaptive classifier with n_h =150

The results displayed in (a) and (b) of Fig. 5.3 – Fig. 5.5 are for the data without noise but with operating point (throttle angle) variations. To validate the ability of the neural network to be tolerant to noise, fault diagnosis with noisy data is conducted. The same training and test data sets were added with the random noise described in Section 5.4. The same tests were repeated with the noisy data for the both networks and the classification results are displayed in Fig.5.7 – Fig. 5.10. It can be seen that the classification is not sensitive to the noise and is again observed that the adaptive classifier with 60 hidden nodes gives much better results than the non-adaptive, off-line trained network with 60 hidden nodes and even 150 hidden nodes.

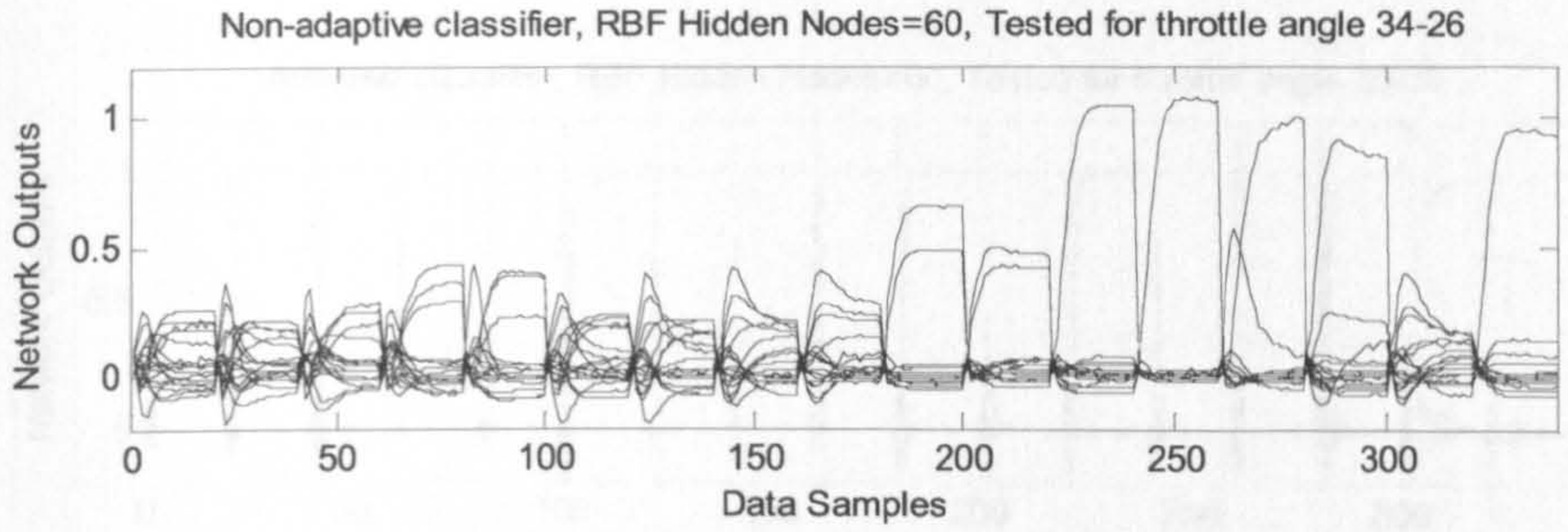


Left and right hand columns show results for state No. 1,3,5,7,9,11,13,15,17 and 2,4,6,8,10,12,14,16 respectively

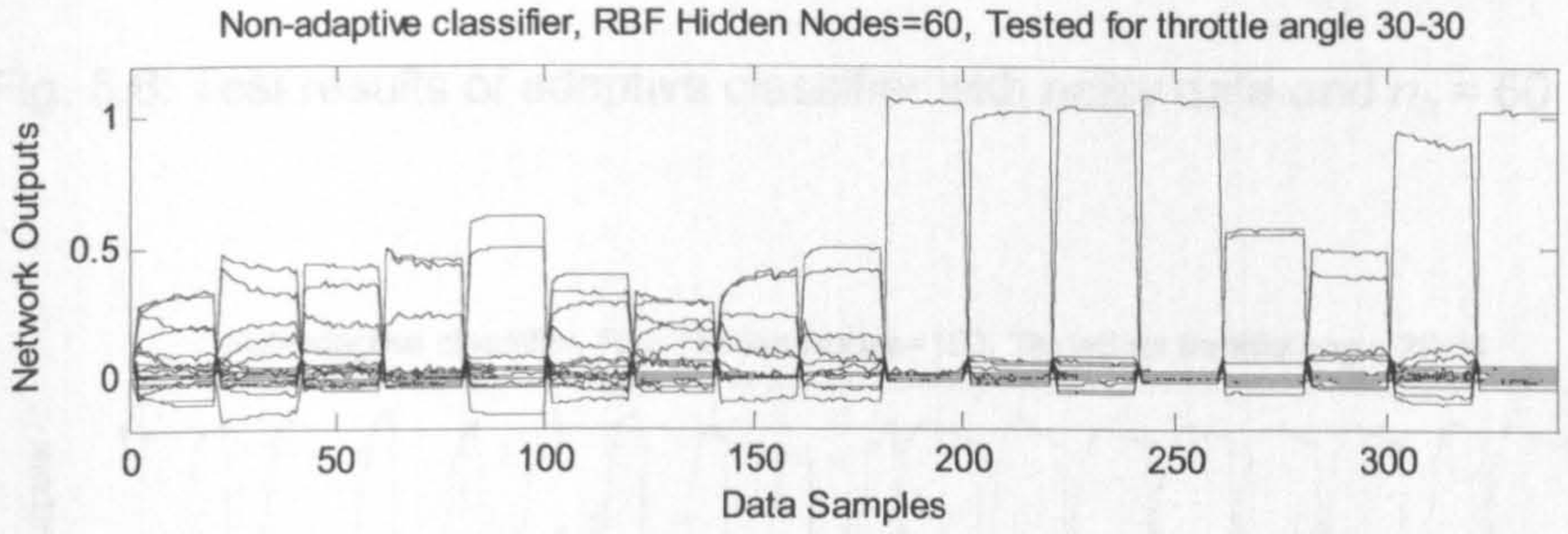
Fig. 5.6: Each output displayed separately for the result in Fig. 5.4b



(a)

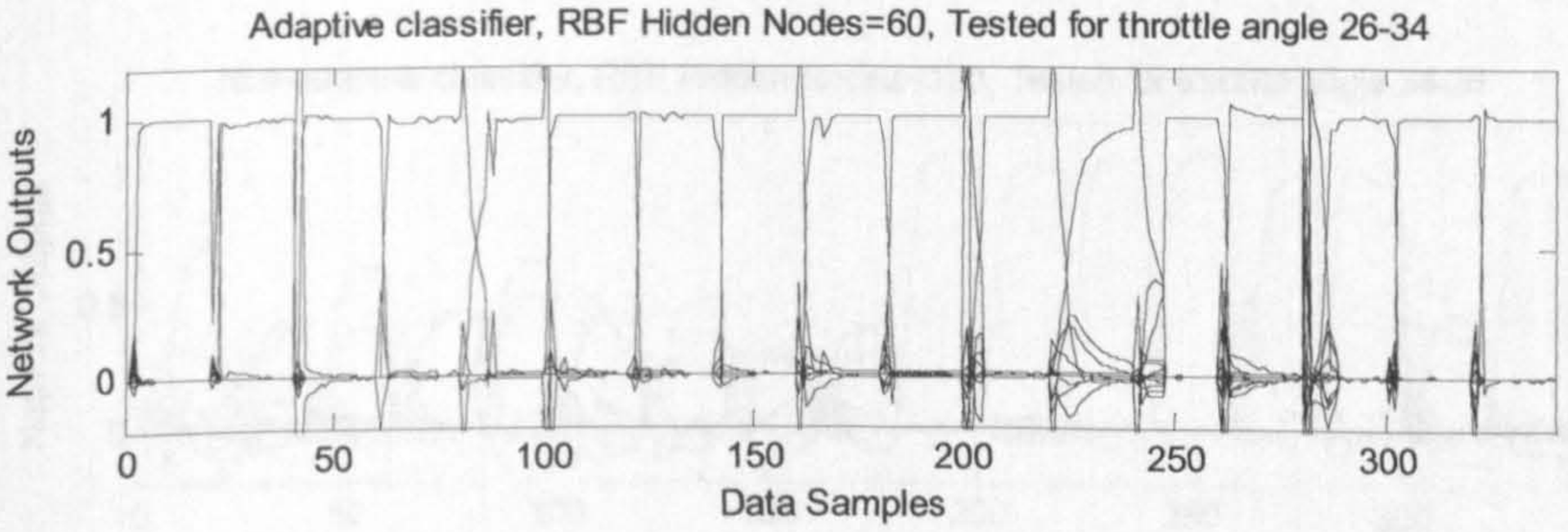


(b)



(c)

Fig. 5.7: Test results of non-adaptive classifier with noisy data and $n_h = 60$



(a)

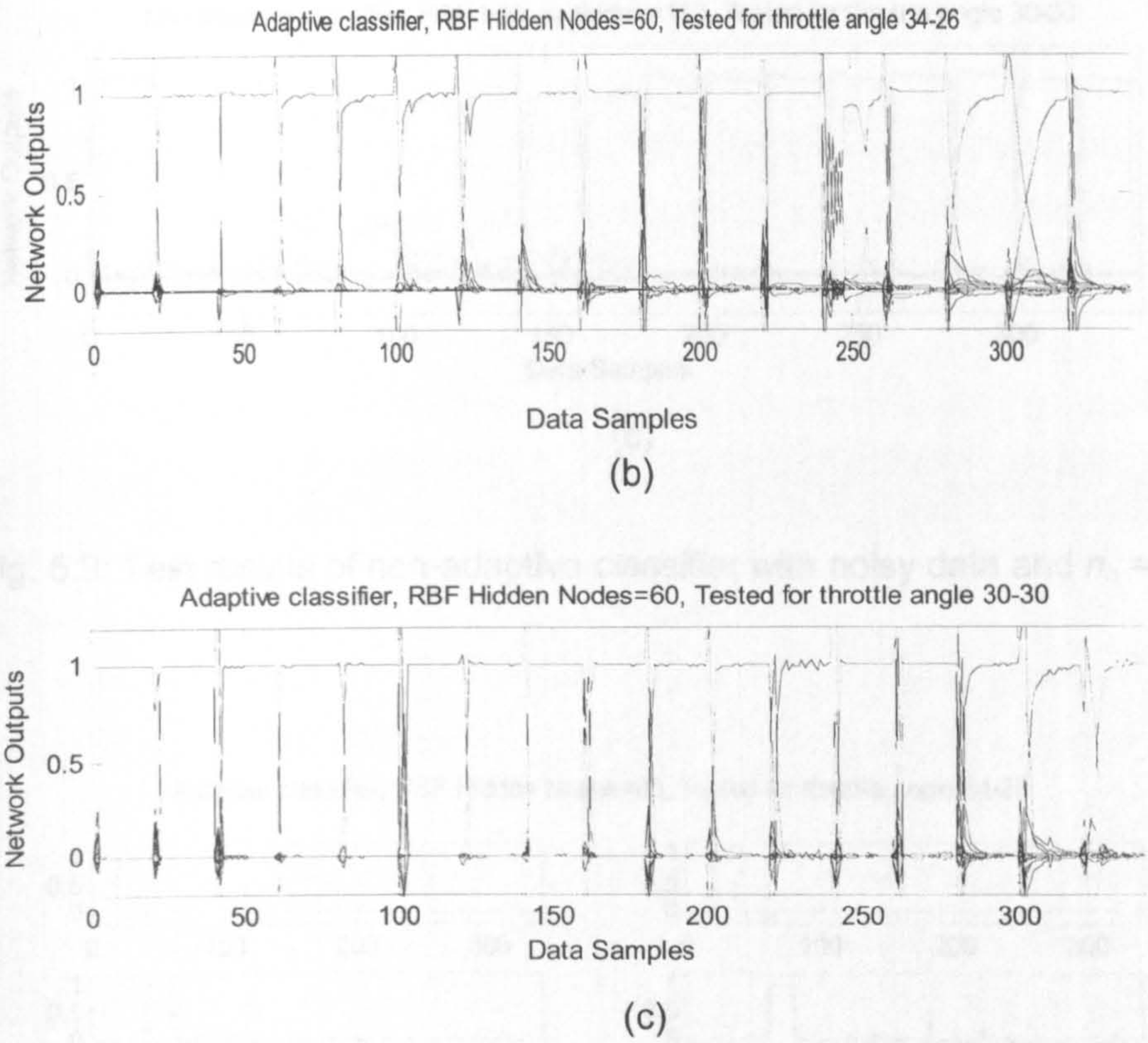
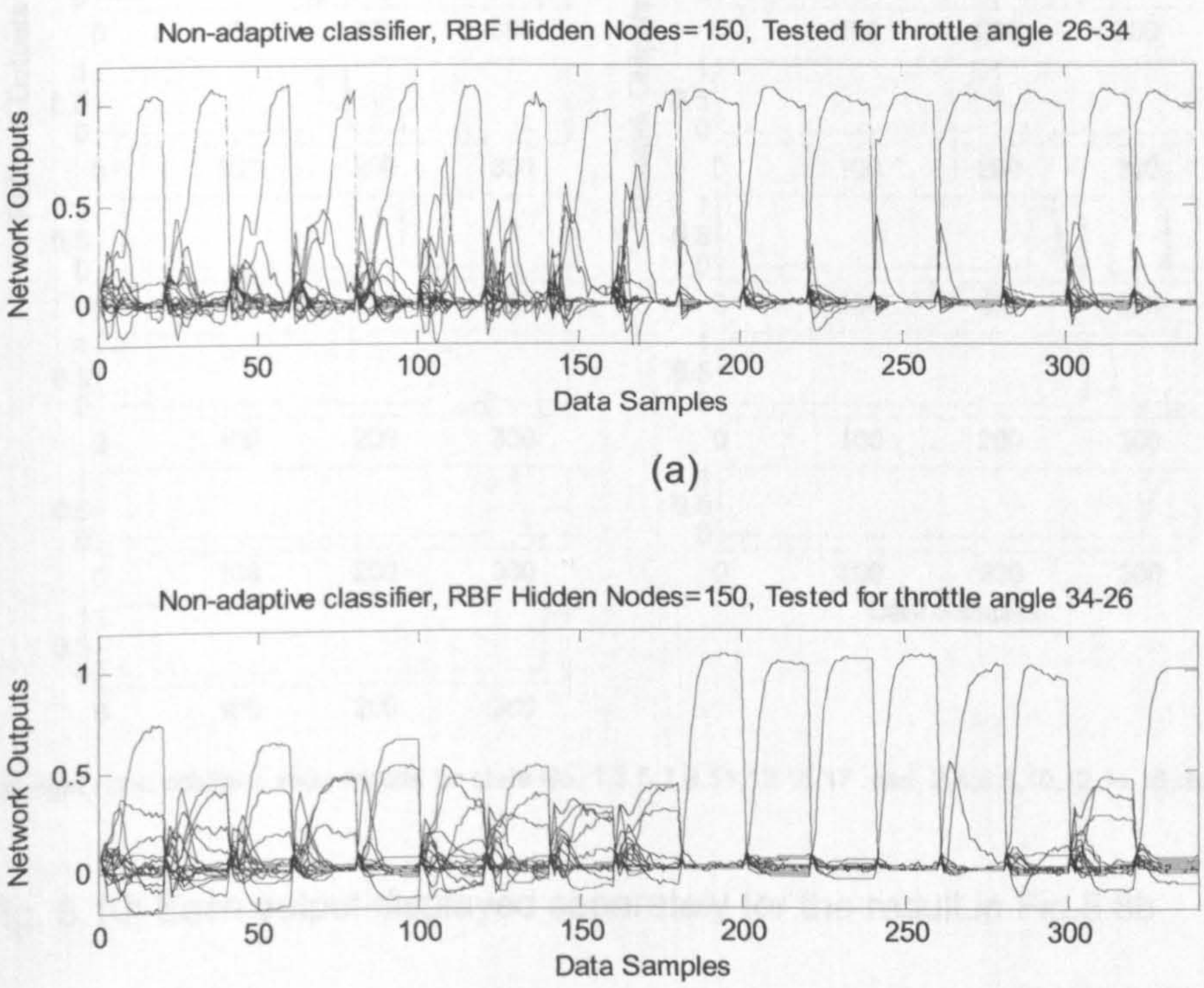
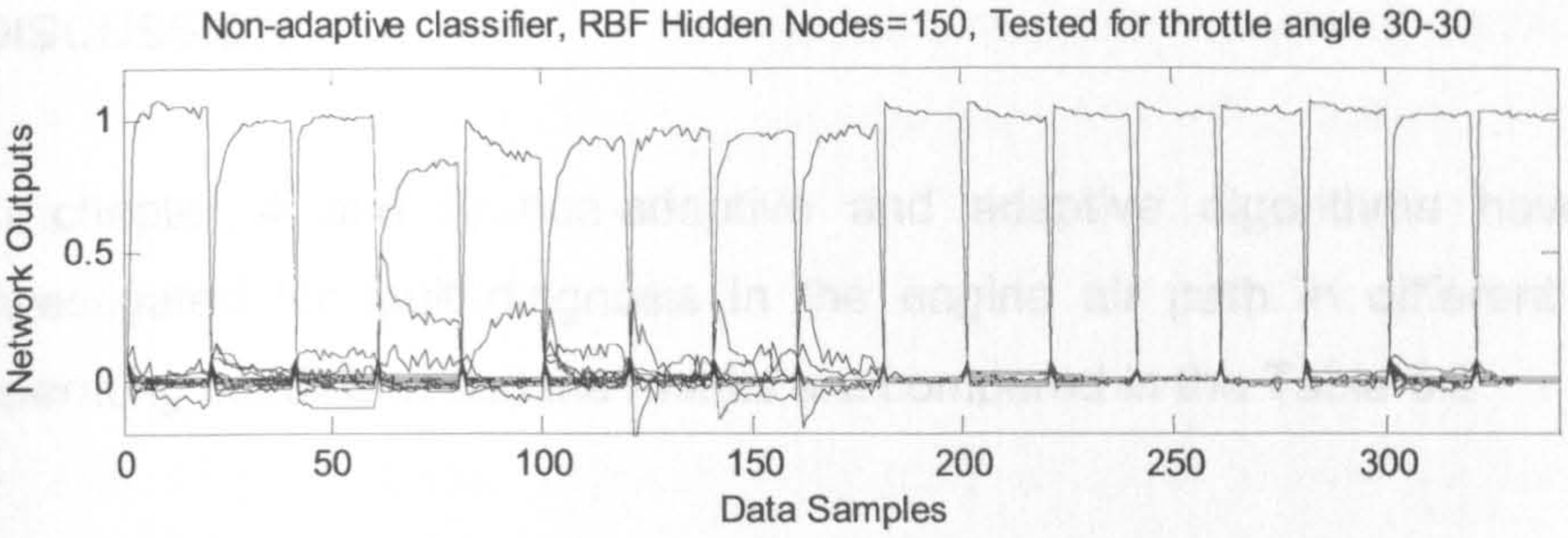


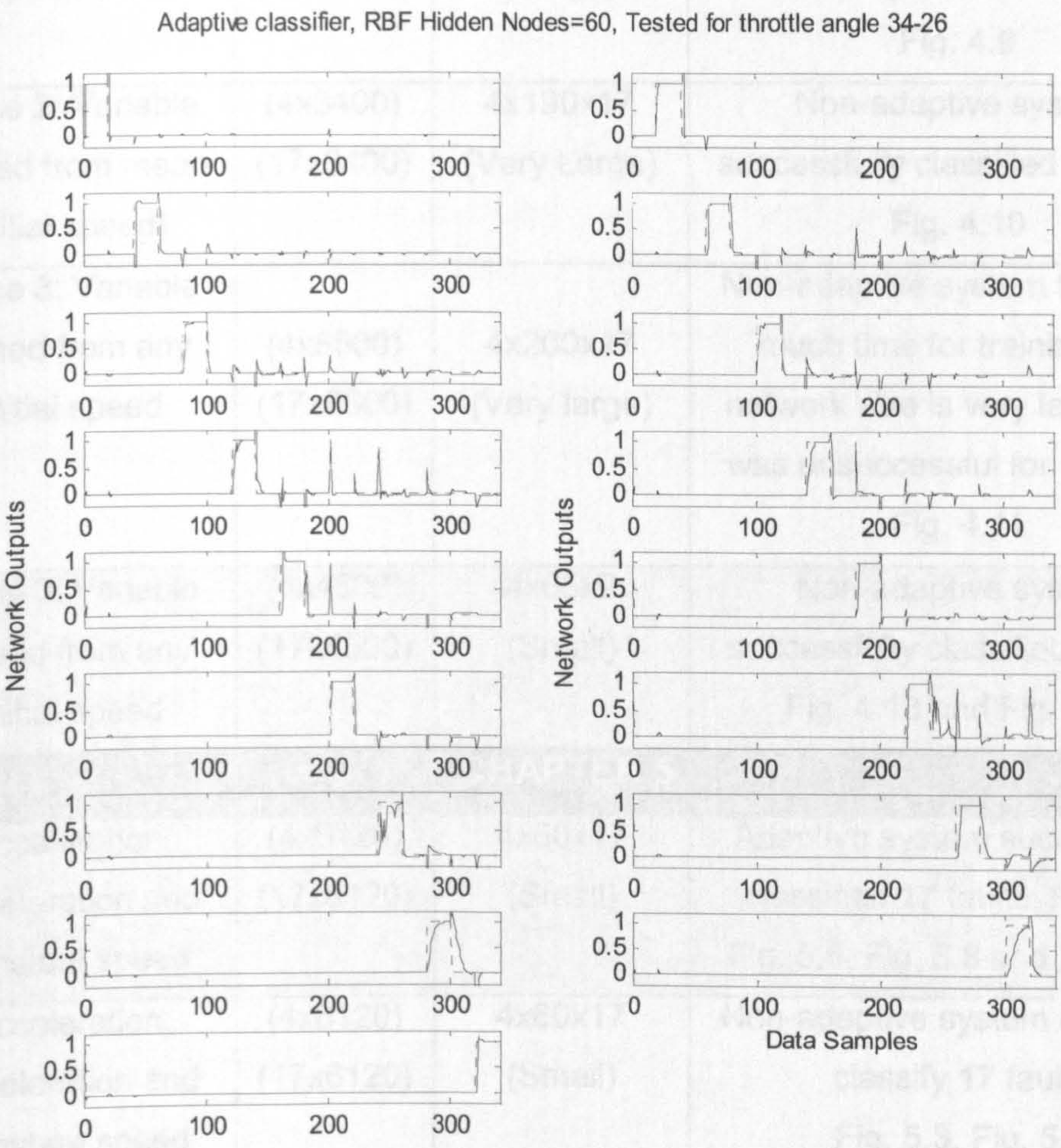
Fig. 5.8: Test results of adaptive classifier with noisy data and n_h = 60





(c)

Fig. 5.9: Test results of non-adaptive classifier with noisy data and n_h =150



Left and right hand columns show results for state No. 1,3,5,7,9,11,13,15,17 and 2,4,6,8,10,12,14,16 respectively

Fig. 5.10: Each output displayed separately for the result in Fig.5.8b

The on-line learning utilises the fault detection result so that faults are not learned as dynamics changes.

5.7 DISCUSSION

In chapter 4 and 5, non-adaptive and adaptive algorithms have been investigated for fault diagnosis in the engine air path in different engine operating conditions and the results are compared in the Table 5.2

Table 5.2: Comparison of results of adaptive and non-adaptive methods

Engine Operating Condition	Training Data size	RBF Neural Network size	Remarks
CHAPTER 4			
Case 1: Constant Speed Run	(4x3400) (17x3400)	4x130x17 (Large)	Non-adaptive system successfully classified **17** faults Fig. 4.9
Case 2: Variable Speed from mean initial speed)	(4x3400) (17x3400)	4x190x17 (Very Large)	Non-adaptive system successfully classified **17** faults Fig. 4.10
Case 3: Variable speed from any initial speed	(4x8500) (17x8500)	4x200x17 (Very large)	Non-adaptive system takes too much time for training, the network size is very large, and was unsuccessful for **17** faults Fig. 4.11
Case 3: Variable speed from any initial speed	(4x4500) (17x4500)	4x65x9 (Small)	Non-adaptive system successfully classified **9** faults Fig. 4.13 and Fig. 4.14
CHAPTER 5			
Acceleration, deceleration and constant speed	(4x6120) (17x6120)	4x60x17 (Small)	Adaptive system successfully classified **17** faults. Fig. 5.4, Fig. 5.6, Fig. 5.8 and Fig. 5.10
Acceleration, deceleration and constant speed	(4x6120) (17x6120)	4x60x17 (Small)	Non-adaptive system could not classify **17** faults Fig. 5.3, Fig. 5.7
Acceleration, deceleration and constant speed	(4x6120) (17x6120)	4x150x17 (Large)	Non-adaptive system was unsuccessful for acceleration and deceleration but successful for constant speed run for **17** faults. Fig. 5.5c and Fig. 5.9c

It was seen in chapter 4 that non-adaptive classifier could not classify large fault set of 17 faults for the most general case of engine operation but was able to classify smaller set of 9 faults. On the other hand, the adaptive classifier is able to classify large set of 17 faults and also the size of the RBF network is significantly smaller as compared to non-adaptive classifier. The smaller size of neural network (4x60x17) is much easier to realise as compared to a larger network (4x190x17).

The adaptive fault classifier on-line updates its widths and weights to learn changes in the system dynamics and environment. To avoid that the classifier also learns the faults as the changes of dynamics, the on-line learning target is modified according to the fault diagnosis result. Therefore, this scheme will still work after one fault is detected. The assessment results on the mean value engine model show that the four simulated faults with different fault sizes are clearly diagnosed. The results are much better than that by a non-adaptive classifier in terms of much lower misclassification rate and much smaller network size. The developed adaptive method is robust to engine dynamics changes including the engine-to-engine differences caused by batch production and the parameter variation caused by long-term mechanical wear of engine components, as well as noise.

One of the drawbacks of developed method is that it is not sensitive to very slowly developing incipient faults, as these faults would be treated as system uncertainty and be ignored by the network. A method is proposed by (Zhang *et al.*, 2002) to detect and isolate incipient faults in nonlinear system which also has a possibility of application in automotive engine FDI. Another drawback of this method is that it can classify the predefined faults which have been considered for the training of the neural network but cannot diagnose the faults which have not been predefined. It implies that unknown faults cannot be diagnosed by this method.

Capriglione *et al.*, (2003 and 2007) have demonstrated the applicability of neural network based FDI system to on-board control units by prototype implementation on a digital signal processor (DSP) which allowed dynamic and diagnostic performances in on-line conditions to be verified.

CHAPTER 6

ROBUSTNESS ASSESSMENT UNDER CLOSED-LOOP CONTROL

6.1 INTRODUCTION

In this chapter the robustness of adaptive neural network classifier presented in Chapter 5 and also in (Sangha *et al.*, 2006) is thoroughly investigated for the closed-loop system with crankshaft speed feedback for a wide range of operational modes, including robustness against fixed and sinusoidal throttle angle inputs, change in load, change in an engine parameter, and all these changes occurring simultaneously. The evaluations are performed on mean value engine model (MVEM). This is to confirm the proposed method is robust against various uncertainties, disturbances, and environment changes.

The adaptive algorithm is also compared with a non-adaptive algorithm in terms of the robustness against a wide range of operational modes for changes in speed set-point, load, and the engine parameter. The nobility of this work consists in the successful demonstration of robustness of the developed adaptive neural network-based FDI algorithm.

6.2 FAULT DIAGNOSIS FOR CLOSED-LOOP SYSTEM

According to the engine air path dynamics, four variables are chosen as the network inputs i.e. the throttle angle, the manifold pressure, the manifold temperature, and the crankshaft speed. This is the same as described before. The RBF network, as the fault classifier has 17 outputs each indicating one of the investigated states, one for no-fault and 16 for 16 faults that are the same as considered in the previous chapter. The information flow for the fault diagnosis is illustrated in Fig. 6.1.

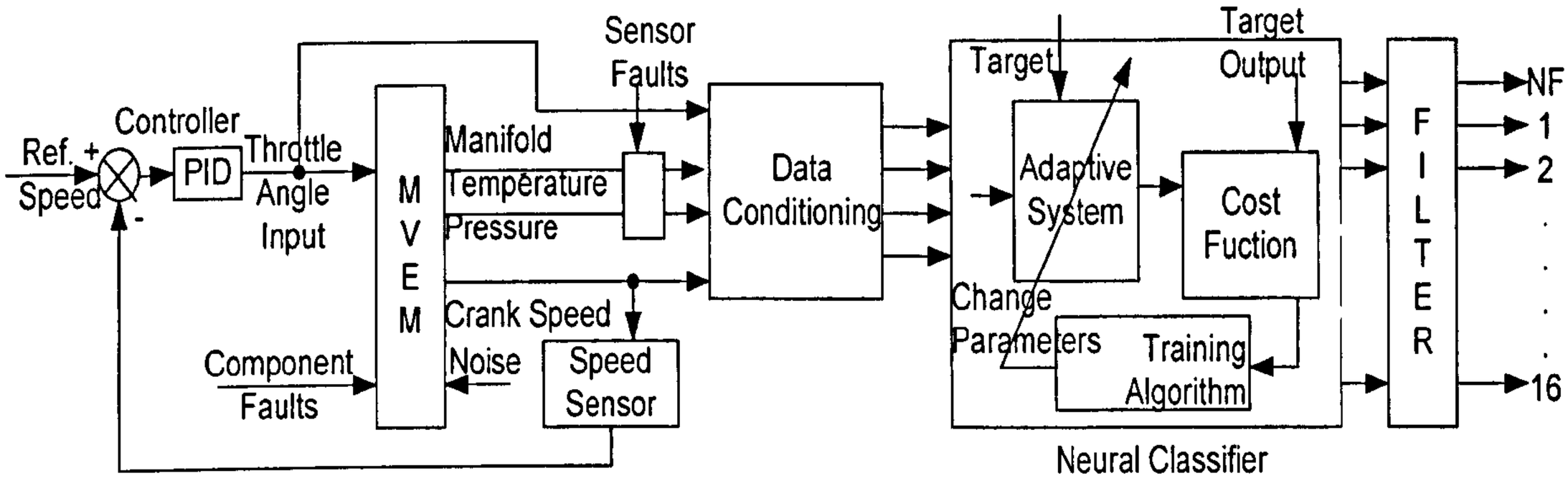


Fig. 6.1: Information flow of the fault diagnosis.

The speed feedback system in a real automotive is fulfilled by a human element (the driver). The speed is controlled to track a set-point by adjusting the throttle angle. In the model, the human controller is represented by a PID controller. The mean value engine model (MVEM) receives a controlled throttle angle input. Component faults are simulated in the model one by one and an appropriate level of measurement noise is added to all input and output measurements. All four inputs and outputs are conditioned, normalised, and fed to the adaptive classifier. Widths in the hidden nodes and the weights in the output layer of the RBF network are adapted to minimise the sum squared error between the output from the adaptive system and the pre-decided target output. The gradient descent method is used for the widths of the RBF network. The width in each hidden layer node is chosen as a constant using the *P*-nearest rule. The classification is sensitive to the Gaussian local function, which is mainly characterised by the width. Therefore, the gradient descent algorithm is useful for on-line adaptation of widths to achieve a minimal objective function as explained before in section 5.2.1.

6.3 PID CONTROLLER DESIGN

A closed-loop PID control is shown in Fig. 6.2.

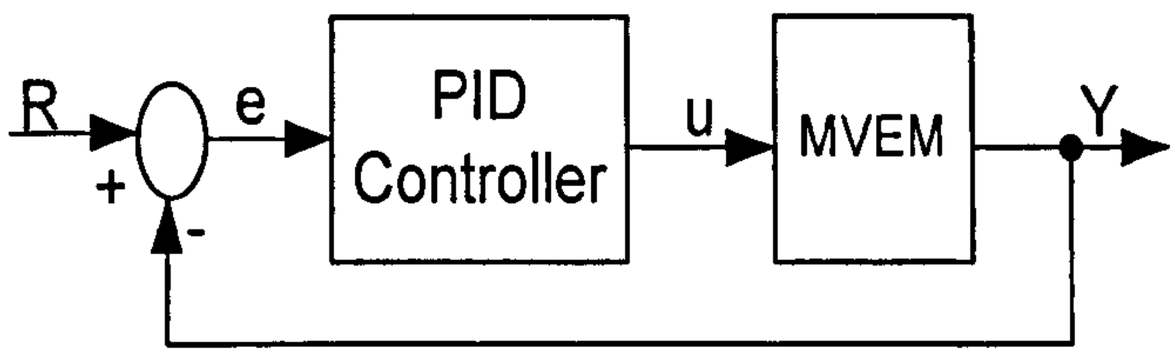


Fig. 6.2: Closed-loop PID control.

The variable '*e*' represents the tracking error, the difference between the desired reference signal '*R*', and the actual output '*Y*'. This error signal '*e*' is sent to the PID controller, and the controller computes both the derivative and the integral of this error signal. The controller output signal '*u*' is equal to the proportional gain K_p times the magnitude of the error plus the integral gain K_i times the integral of the error, and plus the derivative gain K_d times the derivative of the error, as shown in equation (6-1):

$$u(t) = K_p e(t) + K_i \int e(t)dt + K_d \frac{de(t)}{dt} \qquad (6\text{-}1)$$

This signal '*u*' is put into the MVEM, completing the feedback loop fed back to the reference. The well known Ziegler-Nichols method is used for tuning the PID controller. Initially, K_i and K_d gains are set to zero. The proportional gain is increased until it reaches the critical-gain K_c at which the output of the loop starts to oscillate. K_c and the oscillation period P_c are used to set the gains as K_p=0.45*K_c and K_i=1.2*K_p/P_c. The desired output is achieved without the use of the derivative gain. Therefore, the derivative gain is kept zero to keep the controller as simple as possible.

A set of five random values in the range of 2 to 4 kRPM are applied as reference signals. Each random speed is sustained for 6 seconds before the speed signal is changed to the next value because the outputs of the simulation reach their steady state values in six seconds. The data is sampled every 0.5 seconds. Therefore, 12 data points are collected in every

six seconds of time. The output response of the crankshaft speed for the no fault case for five different reference signals is shown in Fig. 6.3.

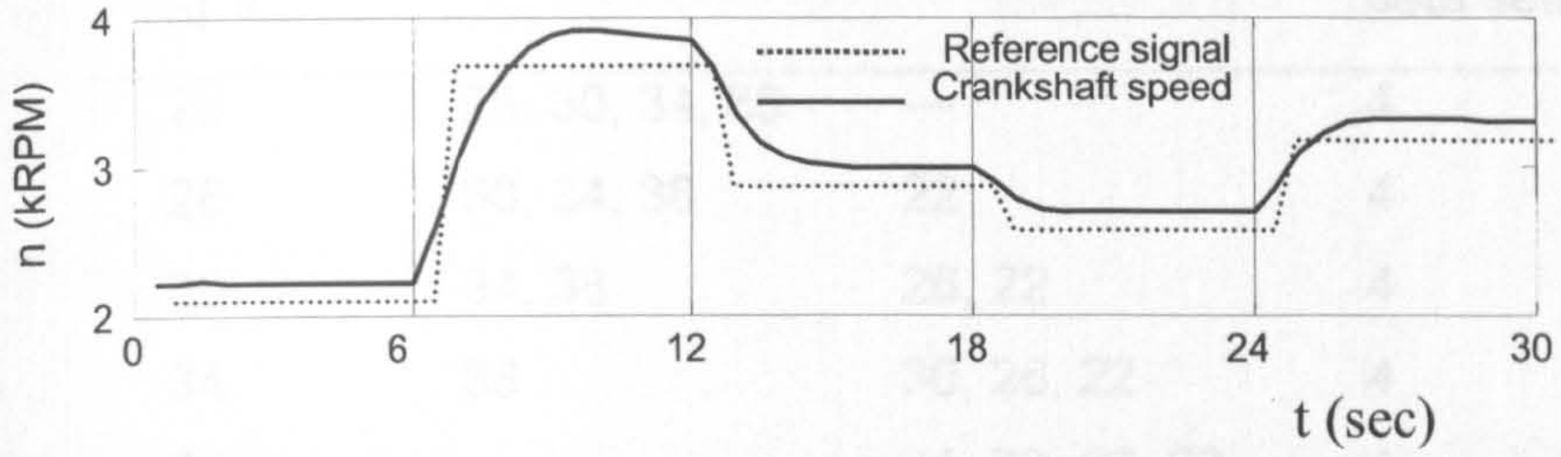


Fig. 6.3: No fault outputs for five random speed reference signals.

The output crankshaft speed follows the input reference speed without much overshoot, delay time, and with a small steady-state error. The chosen PID (K_p =10, K_i =10, K_d = 0) settings give an acceptable level of performance of the controller for further experimentation.

6.4 NETWORK TRAINING

Two RBF networks are used for fault classification, with one for the non-adaptive classifier and the other for the adaptive classifier. Both networks have the same structure and will be trained with the same training data as well as using the same training algorithm. The training for the adaptive network is referred to as initial training. After training, the non-adaptive network will be used to do fault diagnosis with the test data without on-line training, while the adaptive network will be used with the same test data but with on-line training as before in Chapter 5. The comparison between the adaptive and non-adaptive classifiers is done for their relative performances. The network input variables are chosen as before: throttle angle, manifold pressure, manifold temperature and crankshaft speed. Therefore, the network has 4 inputs. Each network output is used to indicate the occurrence of one faulty state 0 (zero), which implies that the fault does not occur while 1 (one) implies that the fault occurs. Therefore, the network has 17 outputs with each corresponding to the one fault or no-fault condition. Twenty data sets for different initial and final throttle angle positions are collected as shown in Table 6.1.

Table 6.1: Data sets collected for training and testing of RBF networks.

Start degree of θ	Accelerating	Decelerating	No. of data sets
22	26, 30, 34, 38	---	4
26	30, 34, 38	22	4
30	34, 38	26, 22	4
34	38	30, 26, 22	4
38	---	34, 30, 26, 22	4

Twenty data sets are collected as shown in the Table 6.1. Two data sets, one acceleration data with $\theta = 26^\circ \rightarrow 34^\circ$ and one deceleration data with $\theta = 34^\circ \rightarrow 26^\circ$, are used as test data. All the remaining 18 sets are used as training data. As each training data set has the same pattern for 17 faults, one training target matrix X_o (Fig. 6.4) is formed and used for all the training data sets. X_o has 204 rows and 17 columns. Its first column has ones from the first row to 12th row with all other entries as zeros and the second column has ones from the 13th row to the 24th row with all other entries as zeros, and the last column has ones from the 193rd row to the 204th row with all other entries as zeros.

Row Numbers X_o

$$\begin{array}{c} 1 \sim 12 \\ 13 \sim 24 \\ \vdots \\ \\ \\ 193 \sim 204 \end{array} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 1 & \cdots & & & \\ 0 & \vdots & \ddots & & & \\ 0 & & & \ddots & & \\ 0 & & & & 1 & \\ \vdots & & & & & \ddots \end{bmatrix}$$

Fig. 6.4: Target matrix X_o.

Thus, the transposition of the i^{th} row in X_o is used as the training target vector for the i^{th} training pattern. The centres are chosen using the *K*-means clustering algorithm from the training data sets. The widths were chosen using the *P*-nearest neighbour's algorithm. The weights were trained using

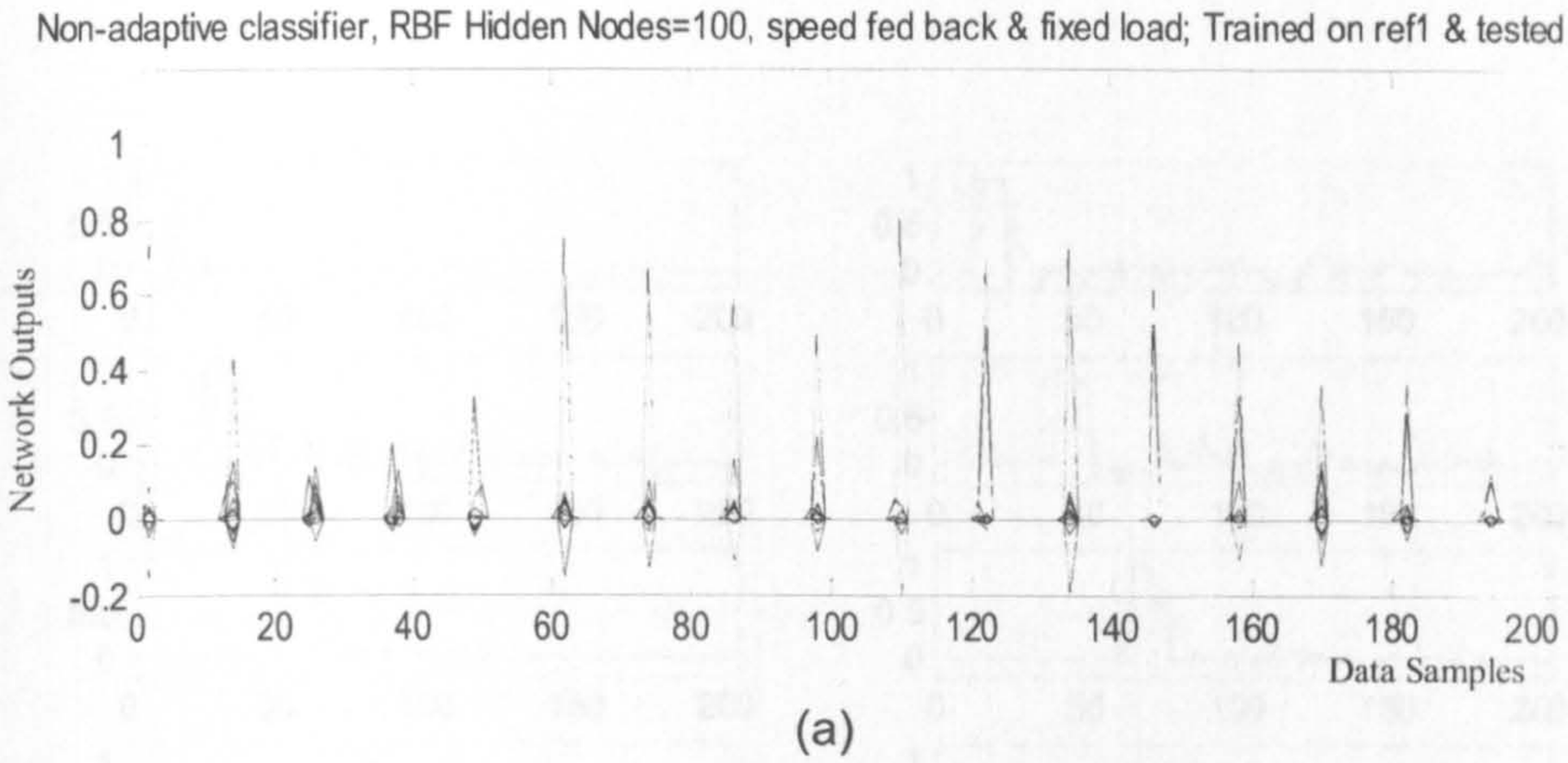
the RLS algorithm. Two levels, 0 and 1, are used as the output targets of the classifier. Thus, the target matrix is a unity diagonal matrix of dimension 17 (when there is one training pattern for each fault) with each column being used as the classifier-training target vector. A successfully trained network will therefore diagnose the fault intensity as well as the fault type.

6.5 FAULT CLASSIFICATION

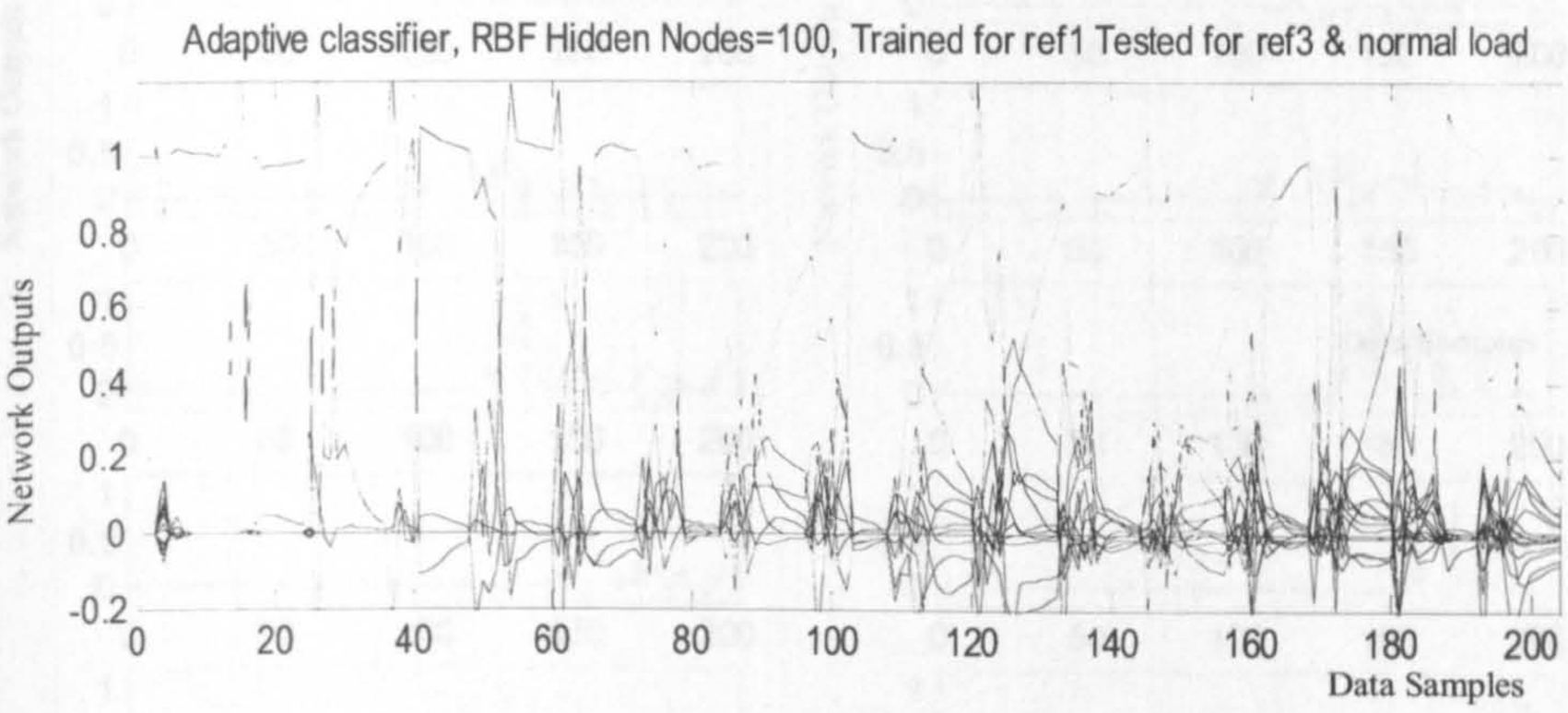
Both adaptive and non-adaptive networks are used to diagnose faults with test data sets after training with the training data sets. The fault detection threshold in (5-7) was chosen as $r_t = 0.5$. High thresholds may lead to missed detections while low thresholds will cause more false alarms. Mathematically, r_t should be a little bit higher than 0.5 according to the level of noise in the testing data. However, $r_t = 0.5$ is found as a good compromise between reliability of detection and insensitivity to noise.

M in (5-6) is chosen as 3. The averaged residual will be greatly reduced and the false alarm is consequently reduced. The threshold for the gradient of the objective function in (5-8) was chosen as $\sigma = 0.00001$. The forgetting factor for the RLS algorithm was chosen as a constant value of $\lambda = 0.99$.

The three different reference signals 2.5 kRPM, 3.0 kRPM, and 3.5 kRPM are chosen as Ref1, Ref2 and Ref3 for the speed control, respectively. No fault and faulty data is collected for all three reference signals. Both the non-adaptive and the adaptive RBF neural network classifiers are then trained and tested for six different sets of data. The results for training the networks on the Ref 1 and testing on the Ref 3 data are shown in Fig. 6.5 and Fig. 6.6. The number of centres for the adaptive and non-adaptive networks is chosen as 100. It is clear that the non-adaptive classifier is not able to classify the simulated faults while the adaptive network classifies the faults with just a few peak values that may cause false alarms when 0.5 is used as the fault detection threshold. These faults are classified when the engine is under closed-loop speed control.



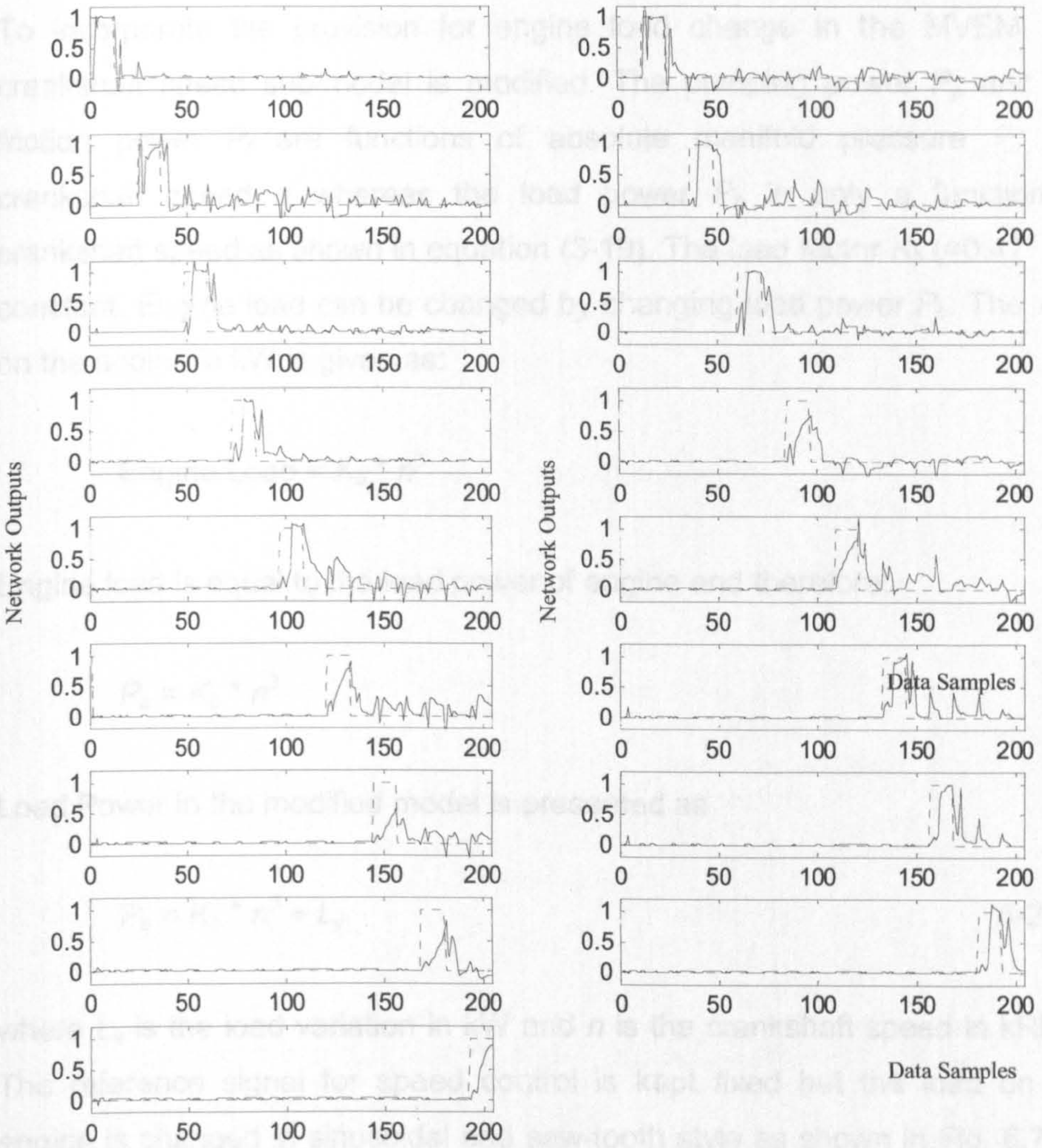
(a)



(b)

Fig. 6.5: Networks trained on Ref1 and tested on Ref3: (a) Non-adaptive; (b) Adaptive.

In comparison, it is found that the adaptive classifier has performed far better than the non-adaptive classifier. Unlike the non-adaptive classifier, the adaptive classifier is able to identify all the faults but with false alarms. For clarity, Fig. 6.5b is shown in an expanded form in Fig. 6.6 with every fault classification shown separately. It can be seen that state 1 has one false alarm, state 2 has two false alarms, state 3 has one false alarm, and so on. Here the requirement of data filtration is felt because of the false alarms. FDI with data filtration is investigated in section 6.7 ahead.



Left and right hand columns show results for state No. 1,3,5,7,9,11,13,15,17 and 2,4,6,8,10,12,14,16 respectively

Fig. 6.6: Details of each fault classification in Fig. 6.5b is shown separately for clarity.

6.6 ROBUSTNESS ASSESSMENT OF FDI SYSTEM

Further to introducing speed feedback control, robustness assessment of the FDI system is carried out in the following three different modes in increasing generality of engine operation:

(1) Load change

(2) Engine parameter change

(3) All the changes happening simultaneously

6.6.1 Load Change

To incorporate the provision for engine load change in the MVEM, the crankshaft speed sub-model is modified. The pumping power P_p and the friction power P_f are functions of absolute manifold pressure P and crankshaft speed n whereas the load power P_b is only a function of crankshaft speed as shown in equation (3-19). The load factor K_b (=0.47) is a constant. Engine load can be changed by changing load power P_b. The load on the engine in kW is given as:

$$\text{Engine Load} = K_b * n^3$$

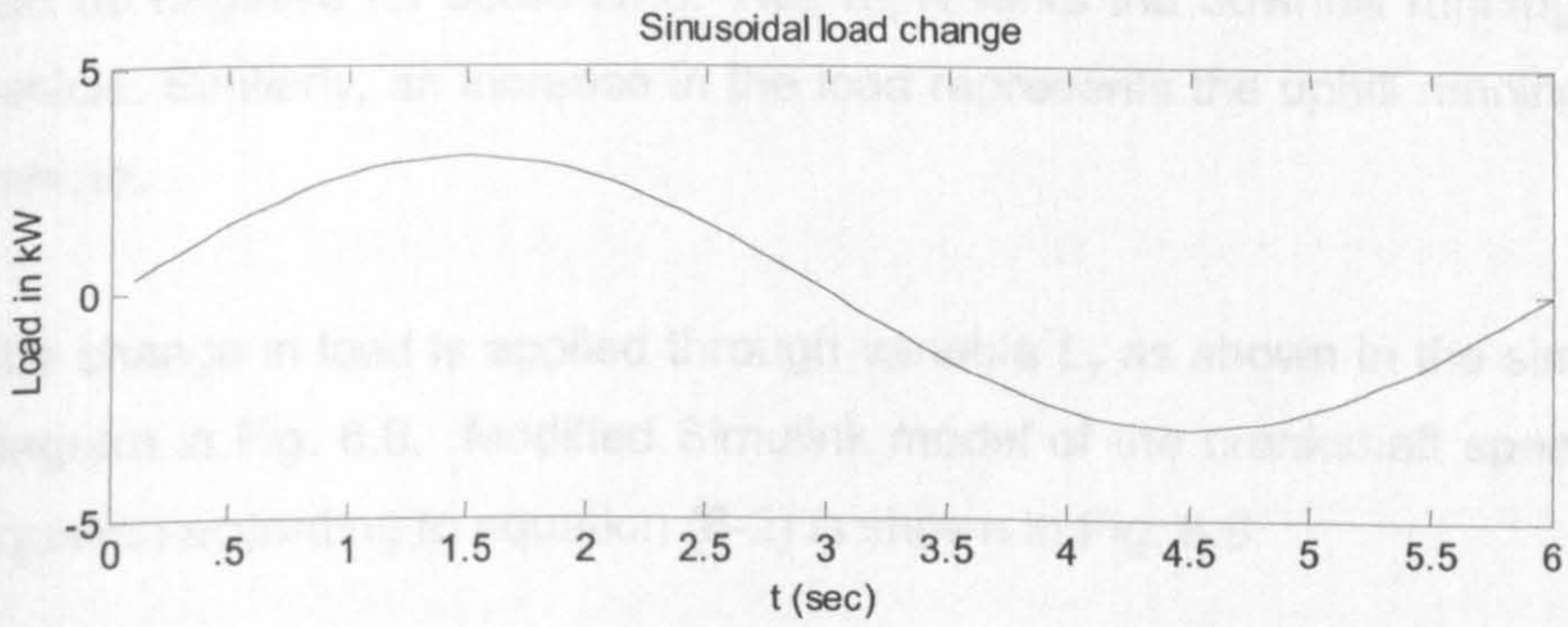
Engine load is equal to the load power of engine and therefore,

$$P_b = K_b * n^3$$

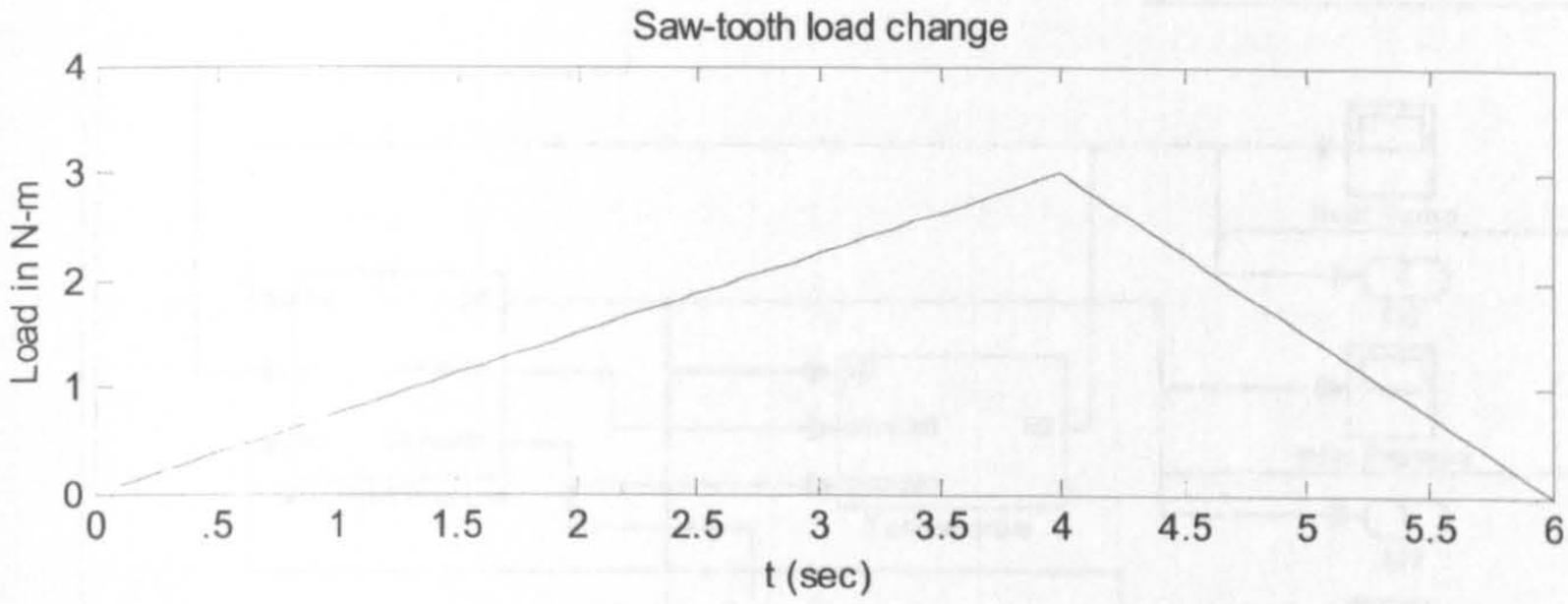
Load Power in the modified model is presented as

$$P_b = K_b * n^3 + L_v \qquad (6\text{-}2)$$

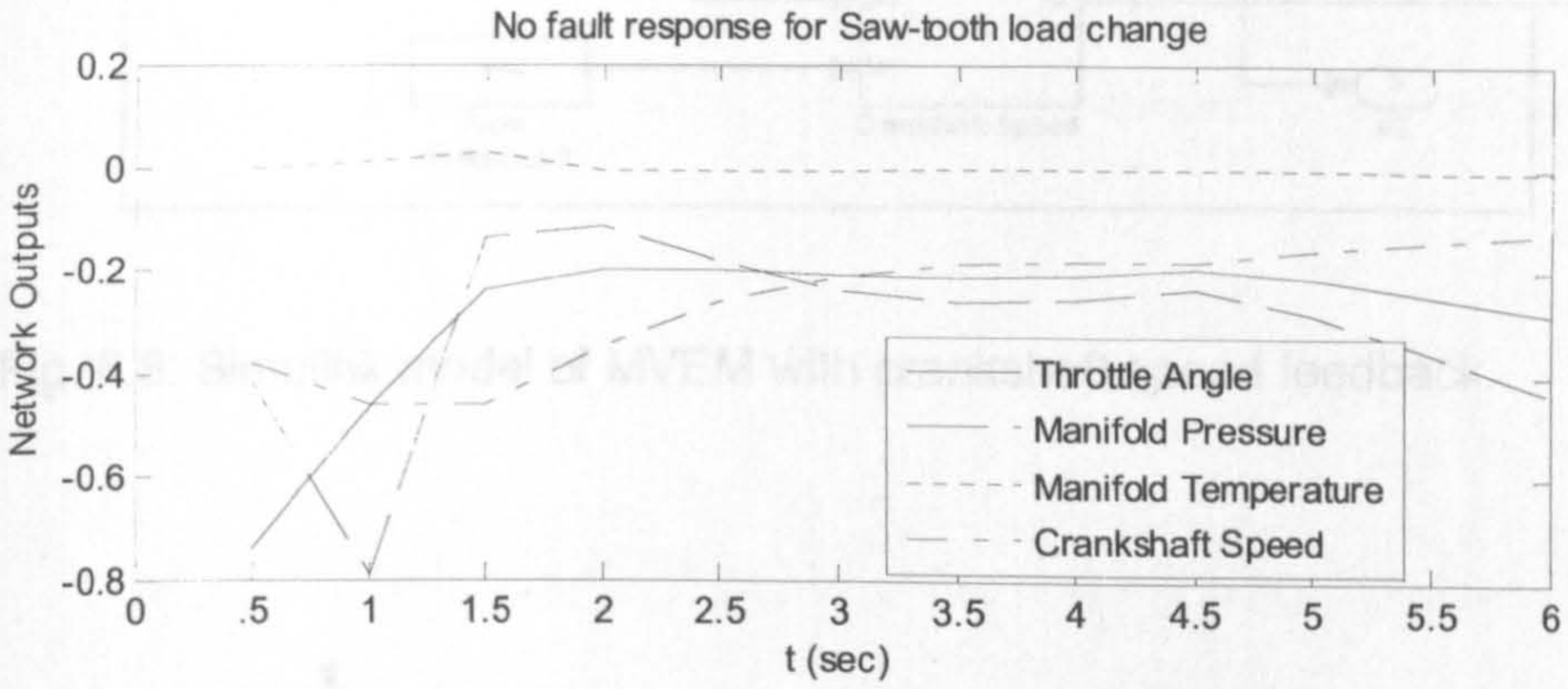
where L_v is the load variation in kW and n is the crankshaft speed in kRPM. The reference signal for speed control is kept fixed but the load on the engine is changed in sinusoidal and saw-tooth style as shown in Fig. 6.7 (a) and (b).



(a)



(b)



(c)

Fig. 6.7: (a) Sinusoidal (L_v) load change; (b) Saw-tooth load change; (c) No fault response for Saw-tooth load change.

In the case of a sinusoidal load change, the load on the automobile (engine) can be negative for some time. This represents the downhill running of the vehicle. Similarly, an increase in the load represents the uphill running of the vehicle.

The change in load is applied through variable L_v as shown in the simulation diagram in Fig. 6.8. Modified Simulink model of the crankshaft speed state equation according to equation (6-2) is shown in Fig. 6.9

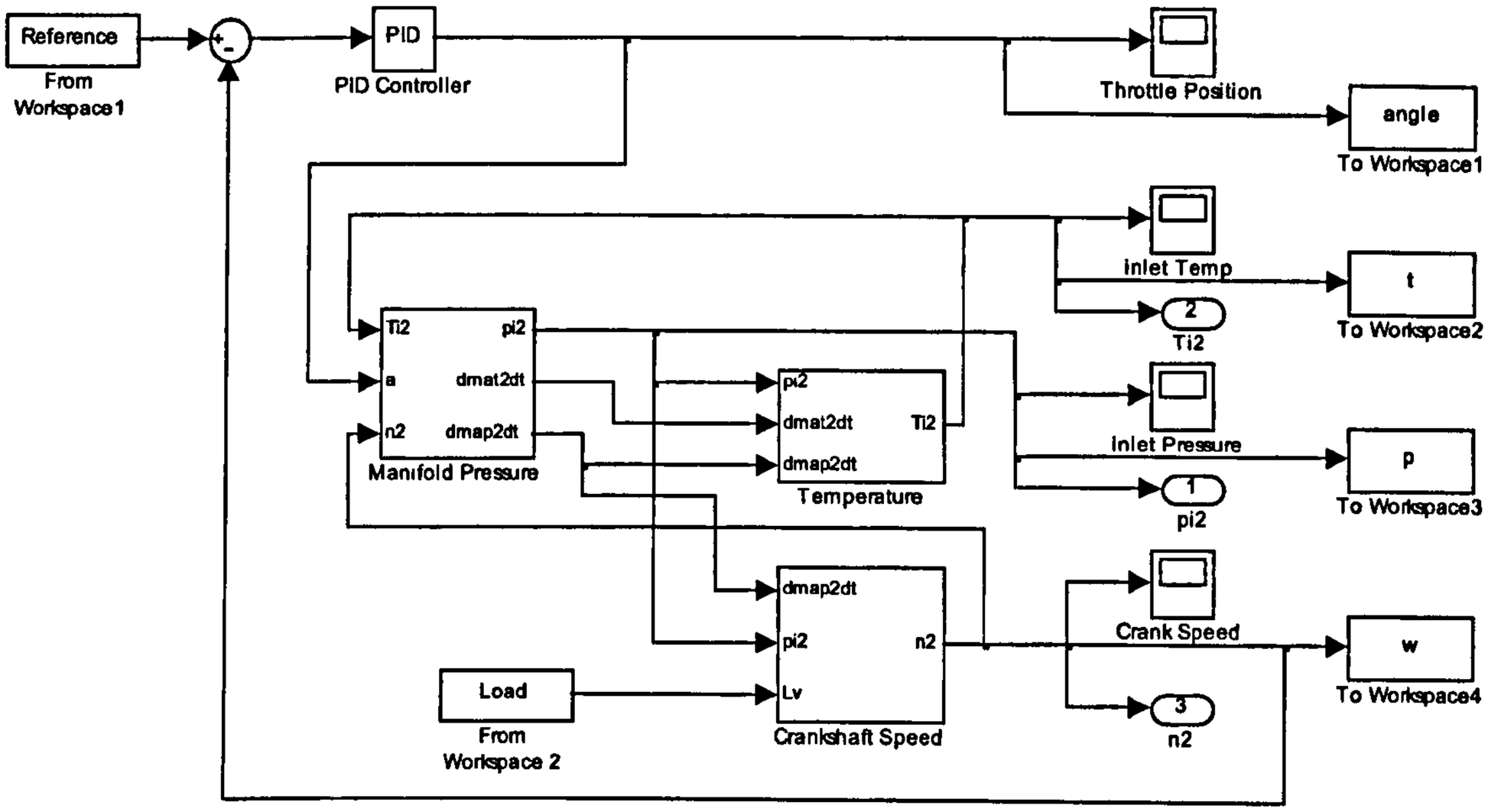


Fig. 6.8: Simulink model of MVEM with crankshaft speed feedback.

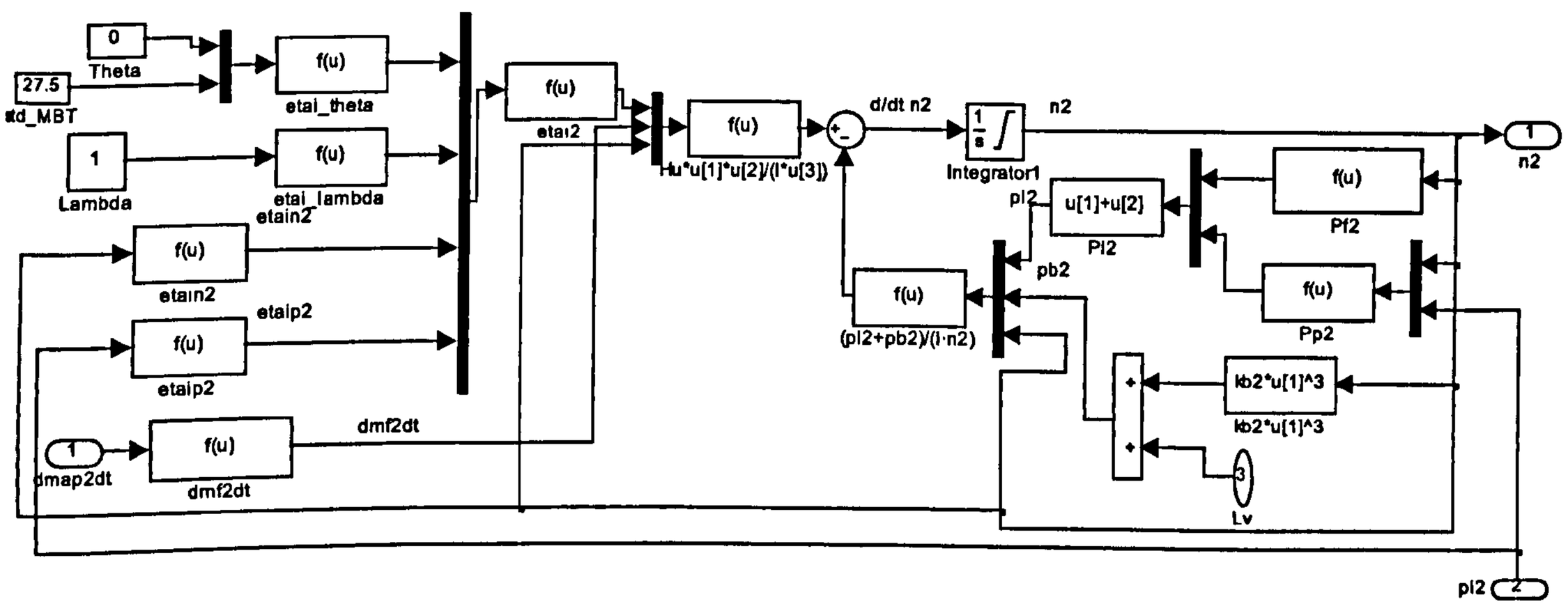
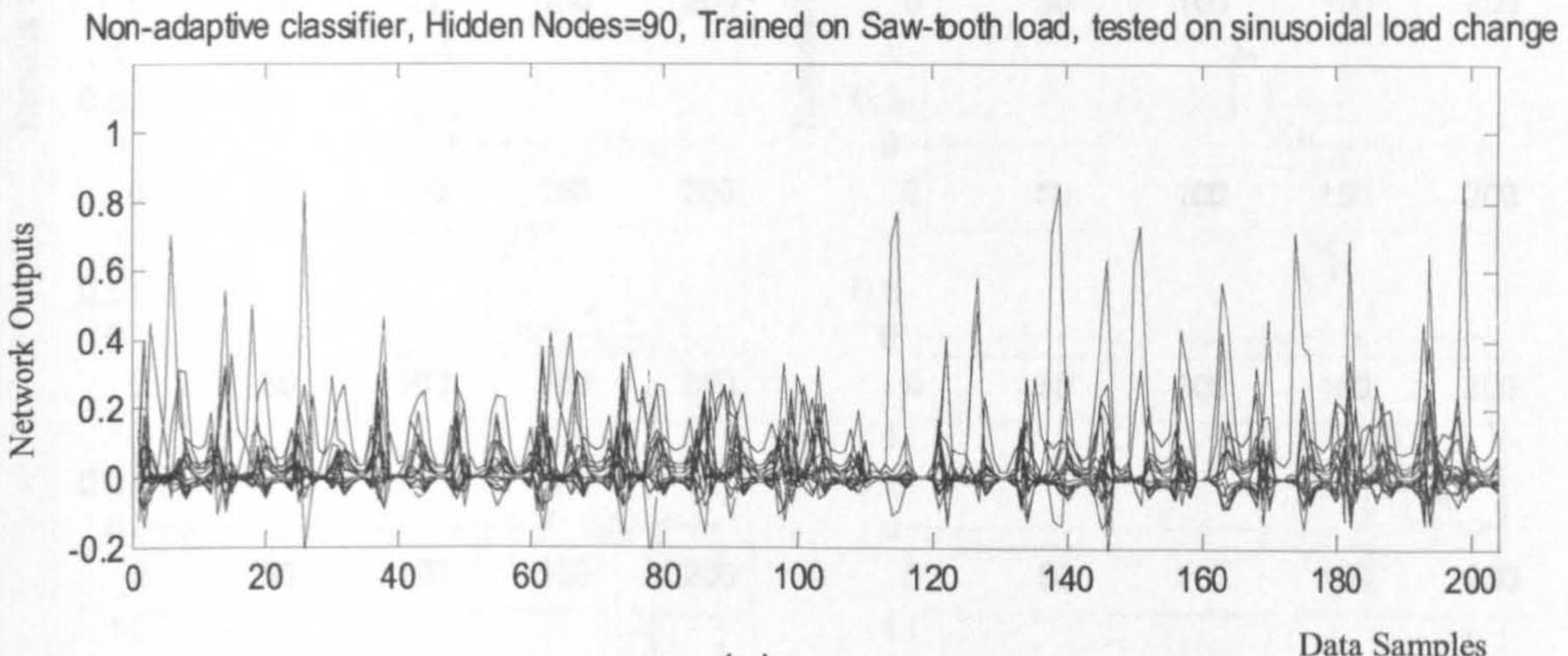


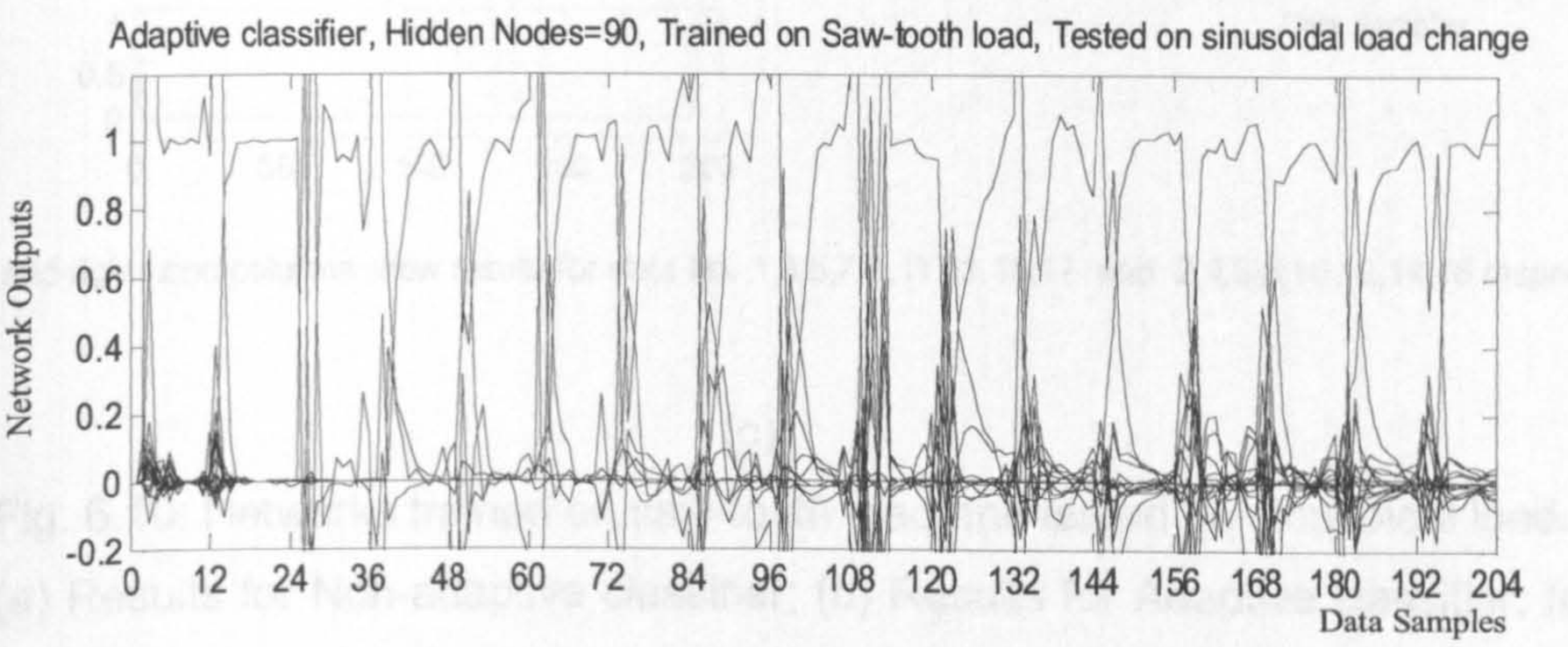
Fig. 6.9: Modified Simulink model of the crankshaft speed state equation

Two sets of data are collected for no fault and faulty conditions; the first set of data for the sinusoidal change in the load and the second set for the saw-tooth change in the load. The reference input signal is kept constant at 2.5kRPM for both data sets. First of all, both networks are trained with data for the sinusoidal load change and tested with data for the saw-tooth load

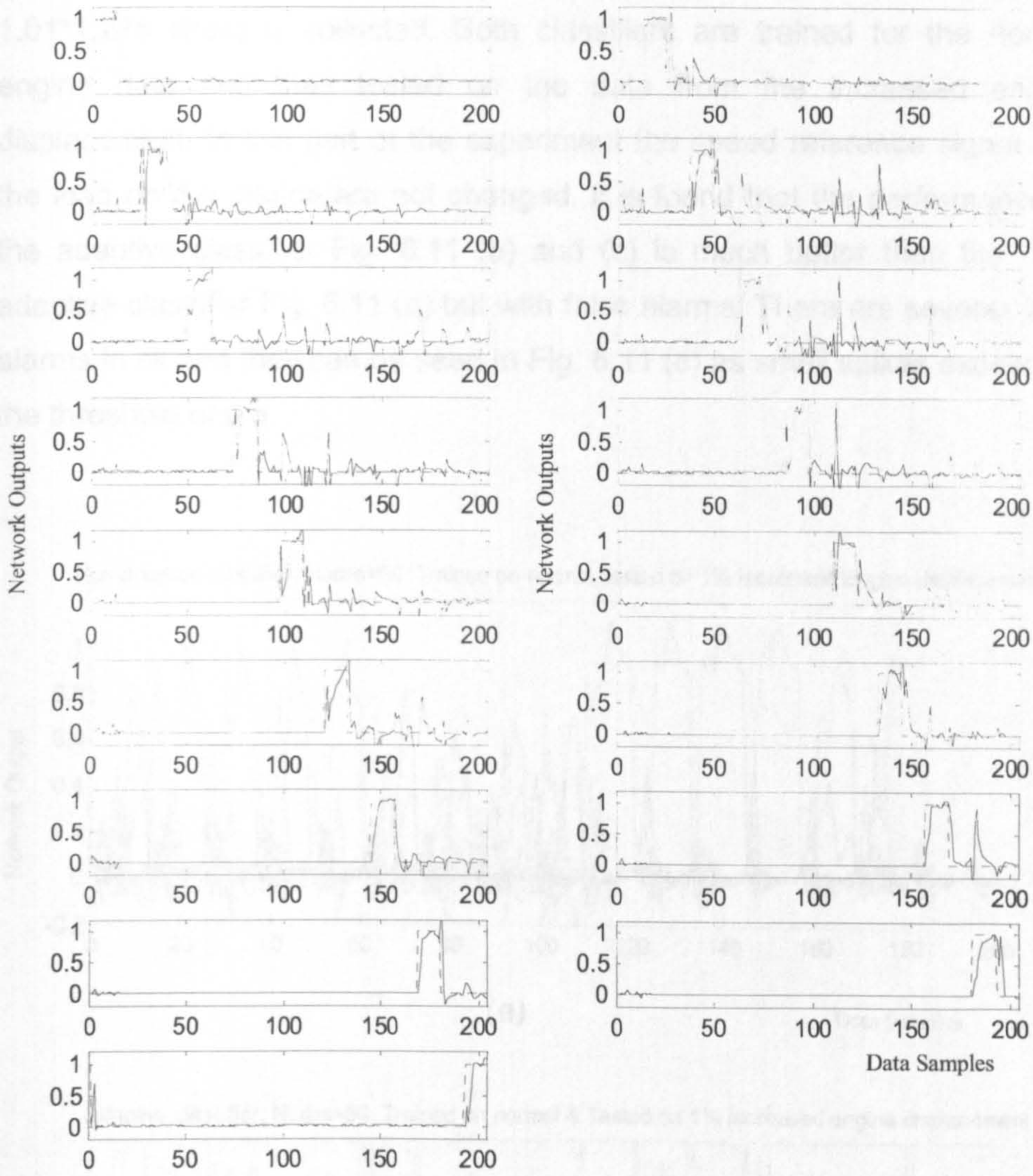
change and then vice-versa. With both training data sets the classification results were found satisfactory. The classification test results for both classifiers when tested for the sinusoidal load change are shown in Fig. 6.10. The results of the non-adaptive classifier are not good as shown in Fig. 6.10 (a) and it is not able to identify different faults. The adaptive classifier is able to identify all the faults as shown in Fig. 6.10 (b) and (c) but with false alarms. There are several false alarms in all and they can be seen in Fig. 6.10 (c), where the small spikes exceeding the threshold of 0.5 can be eliminated by a low-pass filter.



(a)



(b)



Left and right hand columns show results for state No. 1,3,5,7,9,11,13,15,17 and 2,4,6,8,10,12,14,16 respectively

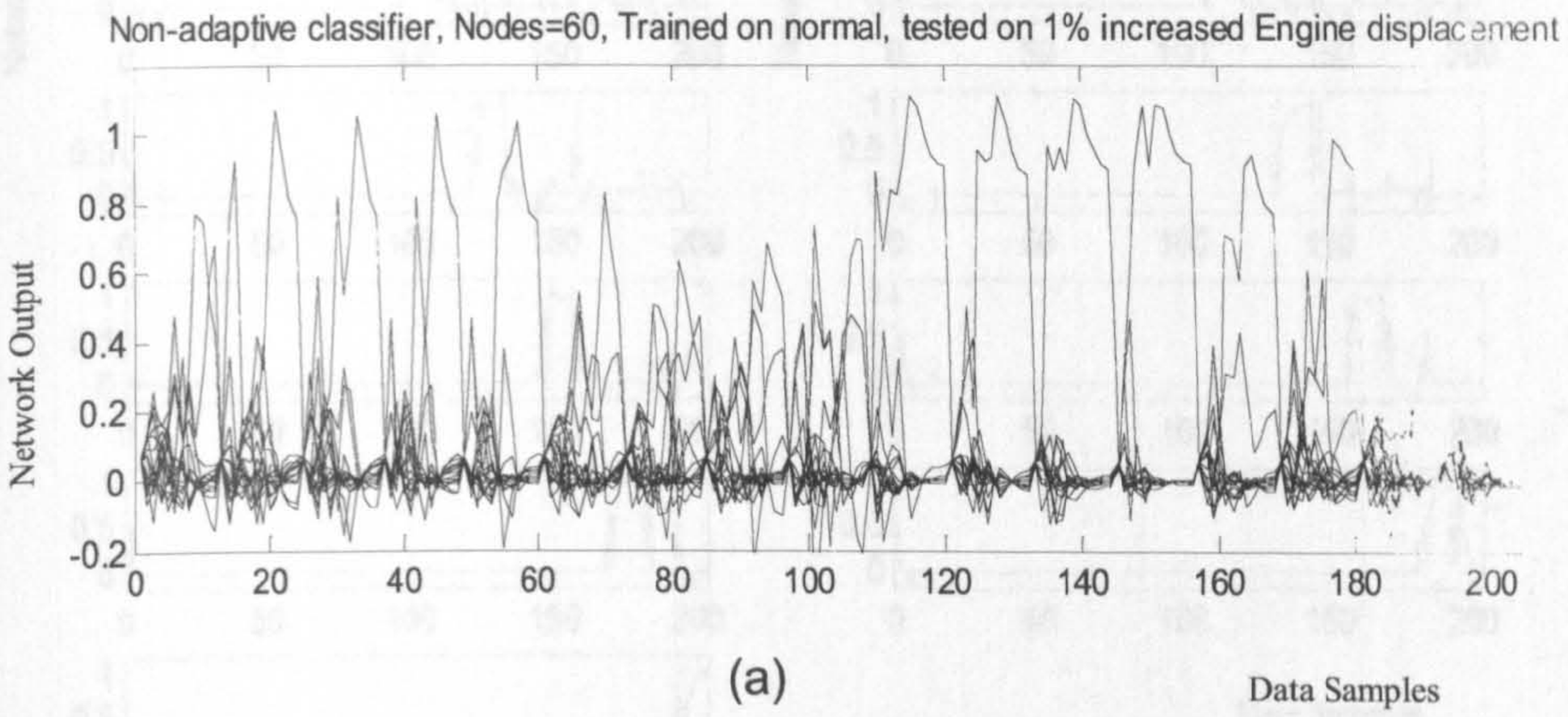
(c)

Fig. 6.10: Networks trained on saw-tooth load and tested on sinusoidal load. (a) Results for Non-adaptive classifier; (b) Results for Adaptive classifier; (c) Details of each fault classification in; (b) shown separately for clarity.

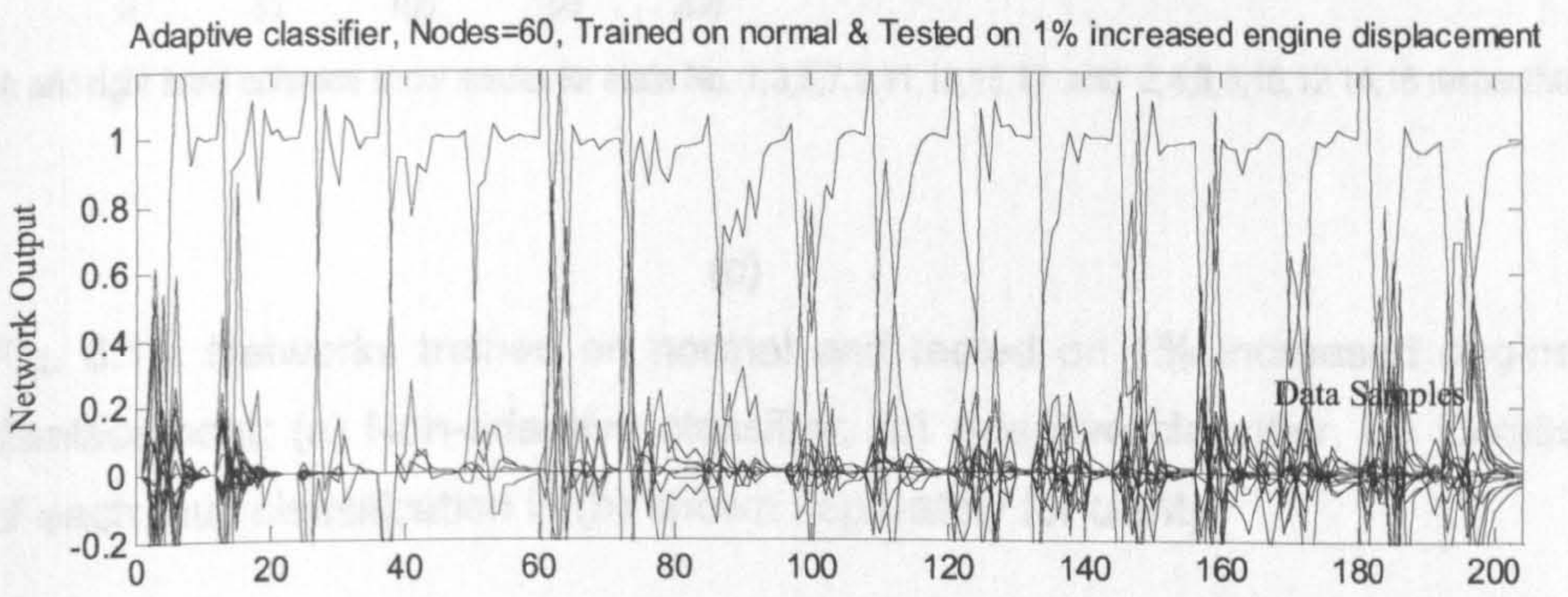
6.6.2. Engine Parameter Change

The engine displacement is a constant for an engine and is 1.275 litres for the MVEM. After a few years of operation, the engine displacement has a tendency to increase by a small amount due to abrasion. In order to check

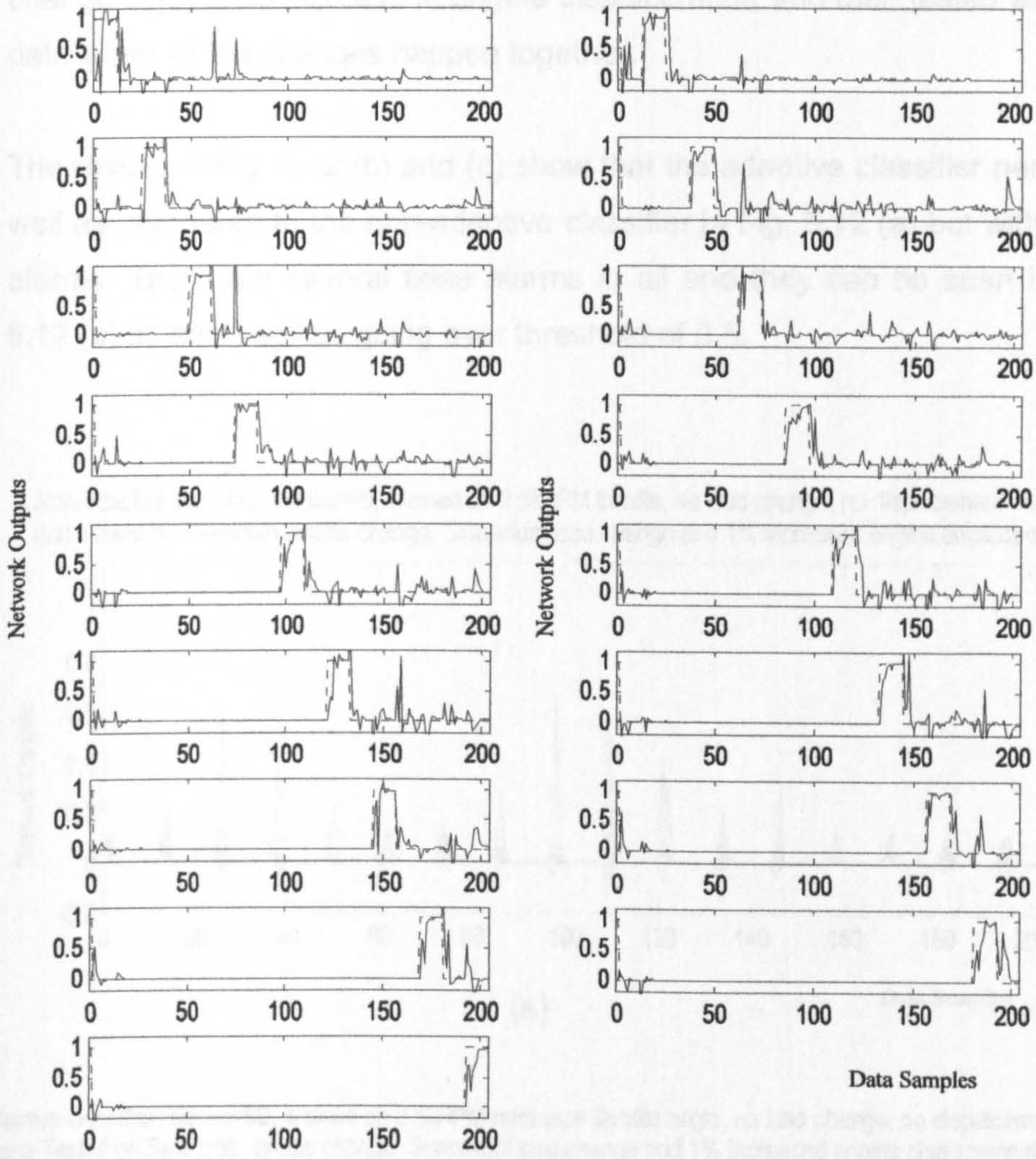
the robustness of the classifier against such an aging effect of the engine, the no fault and faulty data for 1% increased engine displacement i.e. 1.01*1.275 litres) is collected. Both classifiers are trained for the normal engine data and then tested on the data from the increased engine displacement. In this part of the experiment the speed reference signal and the load on the engine are not changed. It is found that the performance of the adaptive classifier Fig. 6.11 (b) and (c) is much better than the non-adaptive classifier Fig. 6.11 (a) but with false alarms. There are several false alarms in all and they can be seen in Fig. 6.11 (c) as small spikes exceeding the threshold of 0.5.



(a)



(b)



Left and right hand columns show results for state No. 1,3,5,7,9,11,13,15,17 and 2,4,6,8,10,12,14,16 respectively

(c)

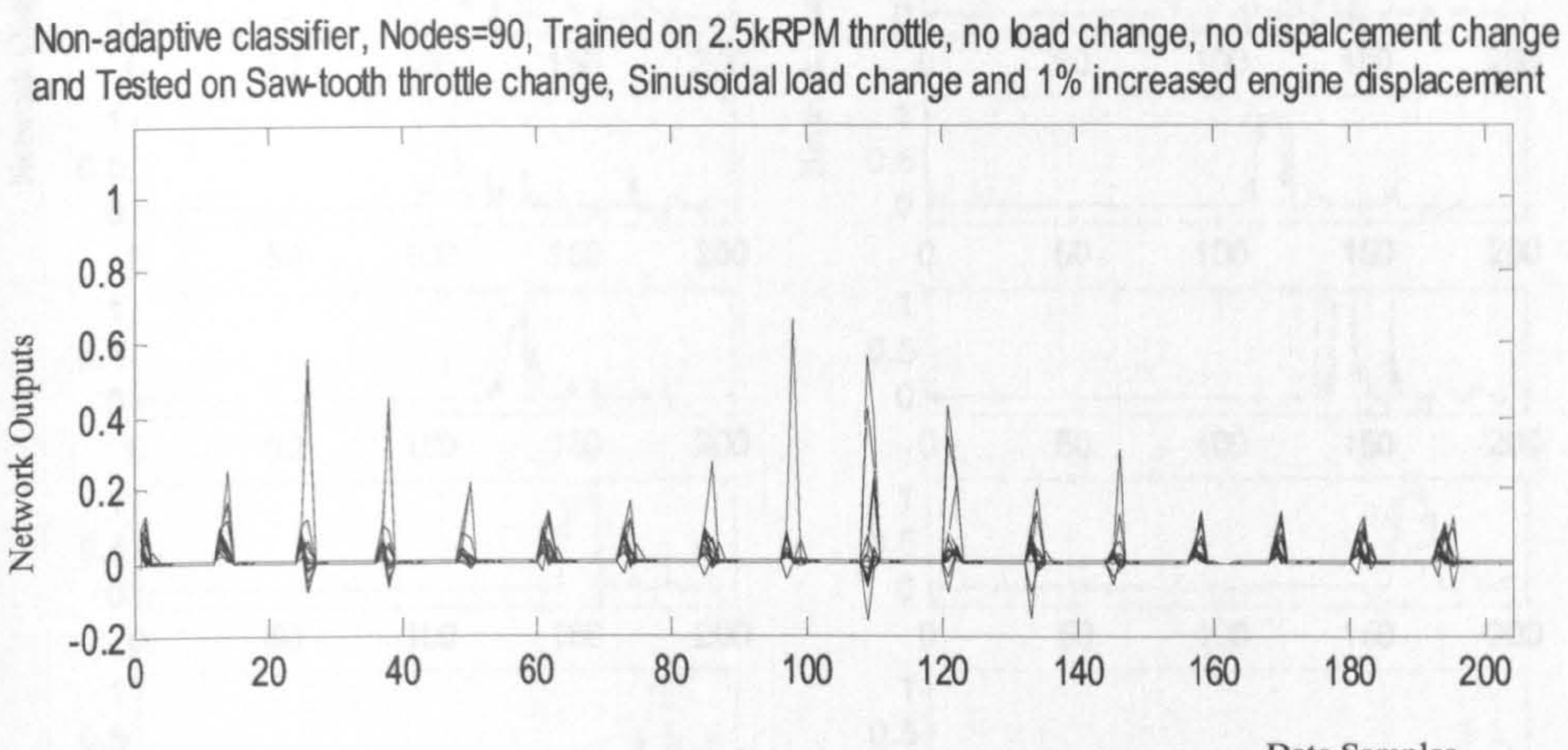
Fig. 6.11: Networks trained on normal and tested on 1% increased engine displacement: (a) Non-adaptive classifier; (b) Adaptive classifier; (c) Details of each fault classification in (b) shown separately for clarity.

6.6.3 All the Changes Happening Together

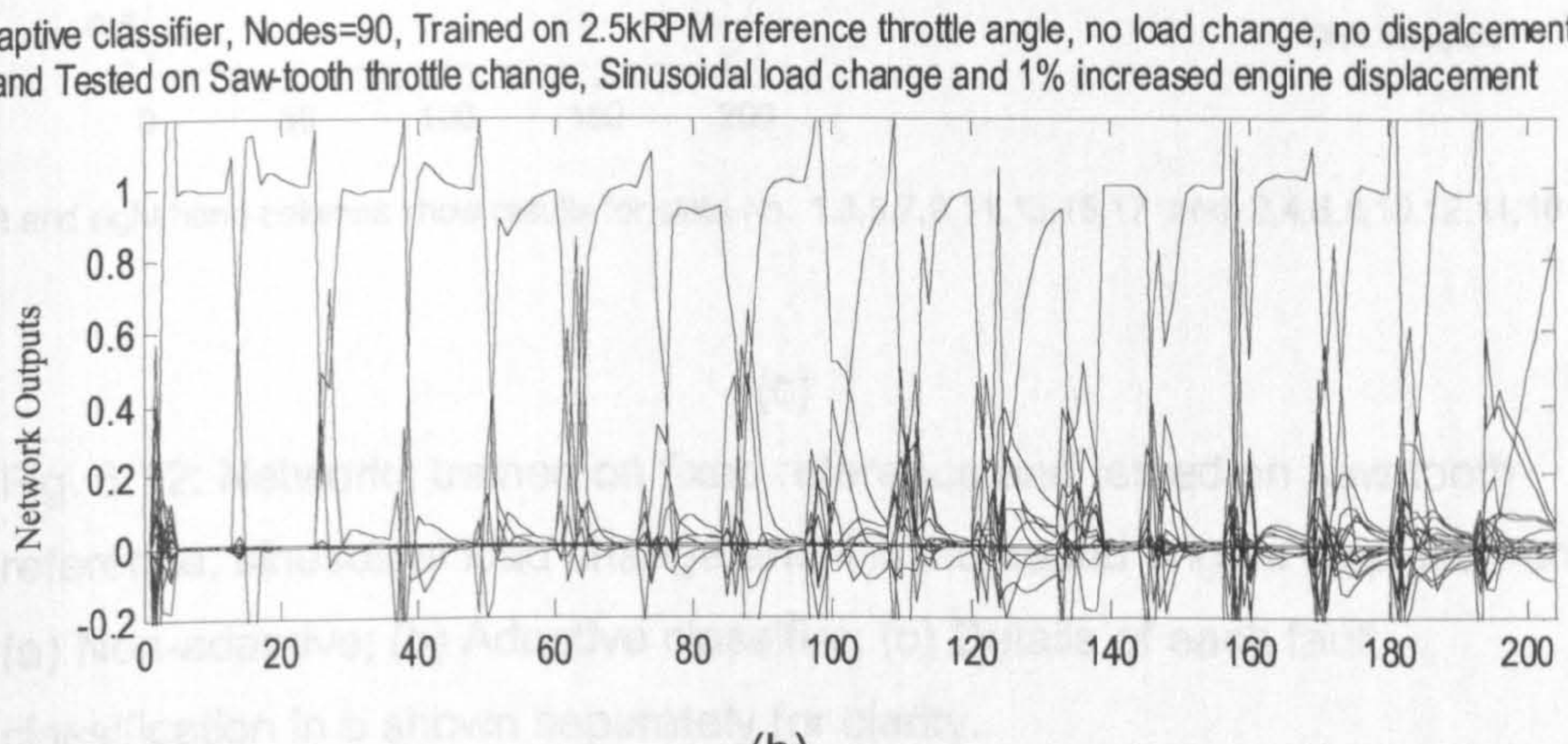
In this section data is collected for all the changes happening simultaneously, i.e. when the reference is a saw-tooth signal, the load is changing in the sinusoidal style and the engine displacement is increased by 1%. Both non-

adaptive and adaptive classifiers are trained on data with fixed reference, no change in load, no increase in engine displacement, and then tested with the data when all the changes happen together.

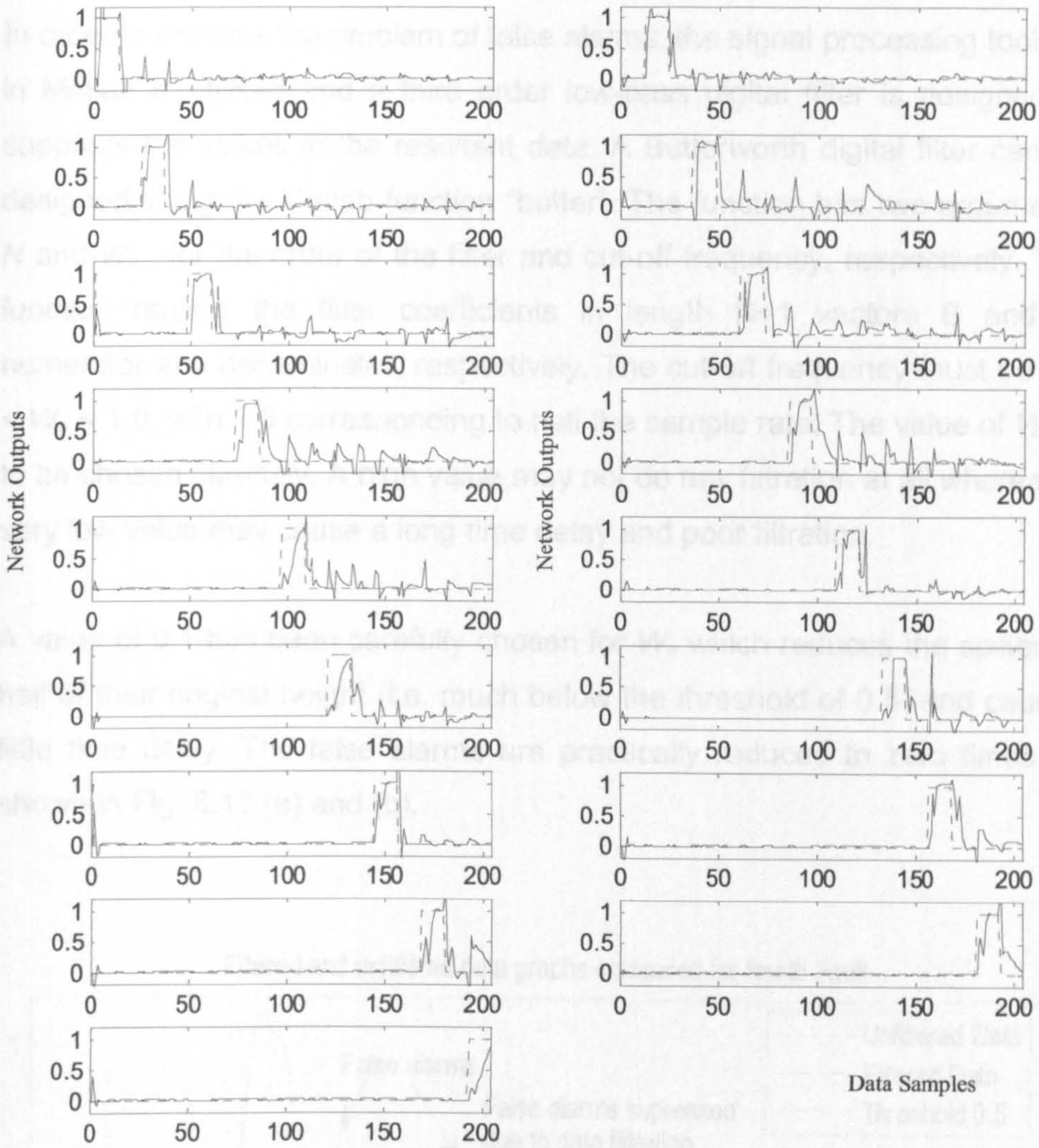
The results in Fig. 6.12 (b) and (c) show that the adaptive classifier performs well as compared to the non-adaptive classifier in Fig. 6.12 (a) but with false alarms. There are several false alarms in all and they can be seen in Fig. 6.12 (c) as small spikes going over threshold of 0.5.



(a)



(b)



Left and right hand columns show results for state No. 1,3,5,7,9,11,13,15,17 and 2,4,6,8,10,12,14,16 respectively

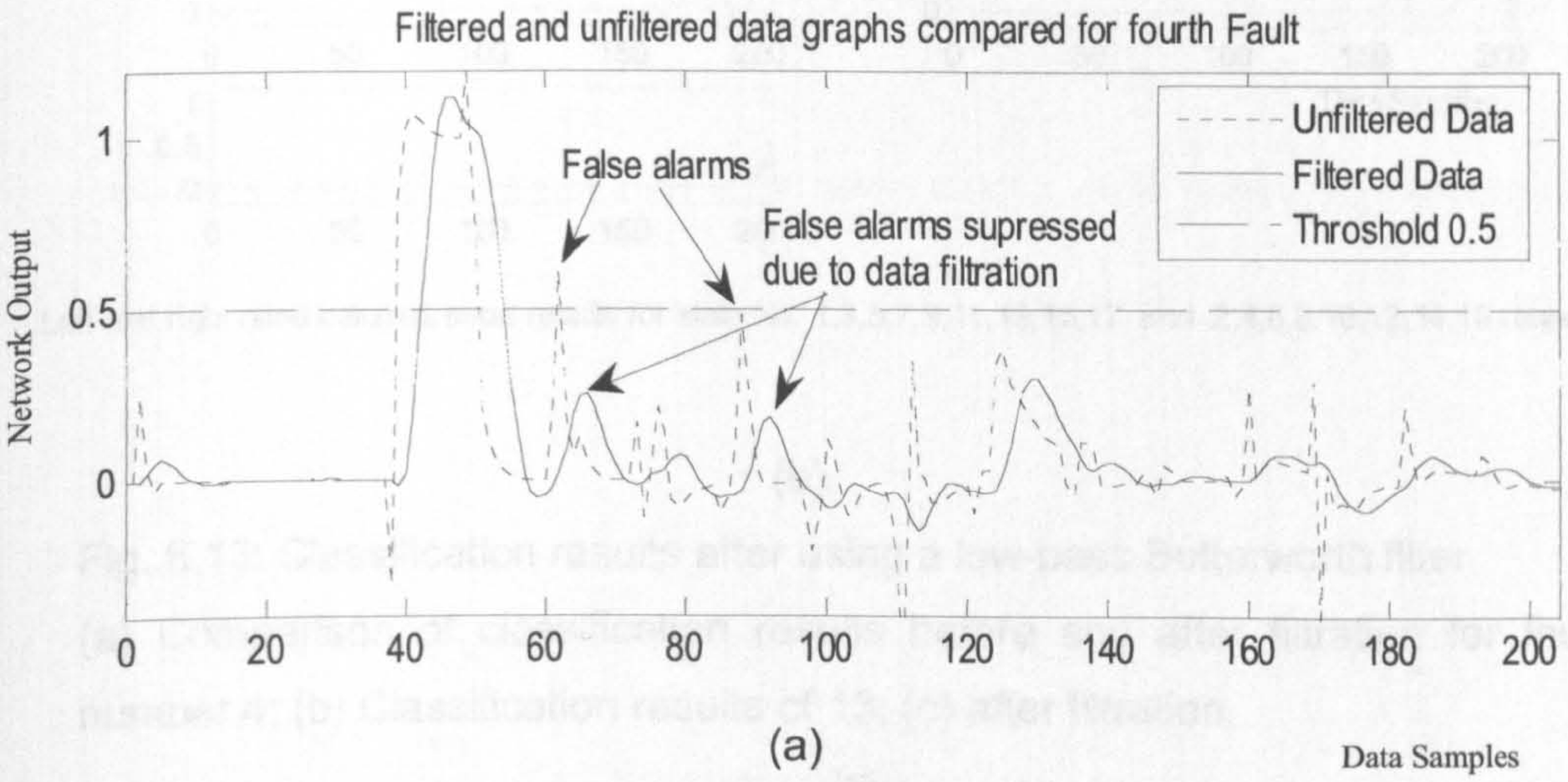
(c)

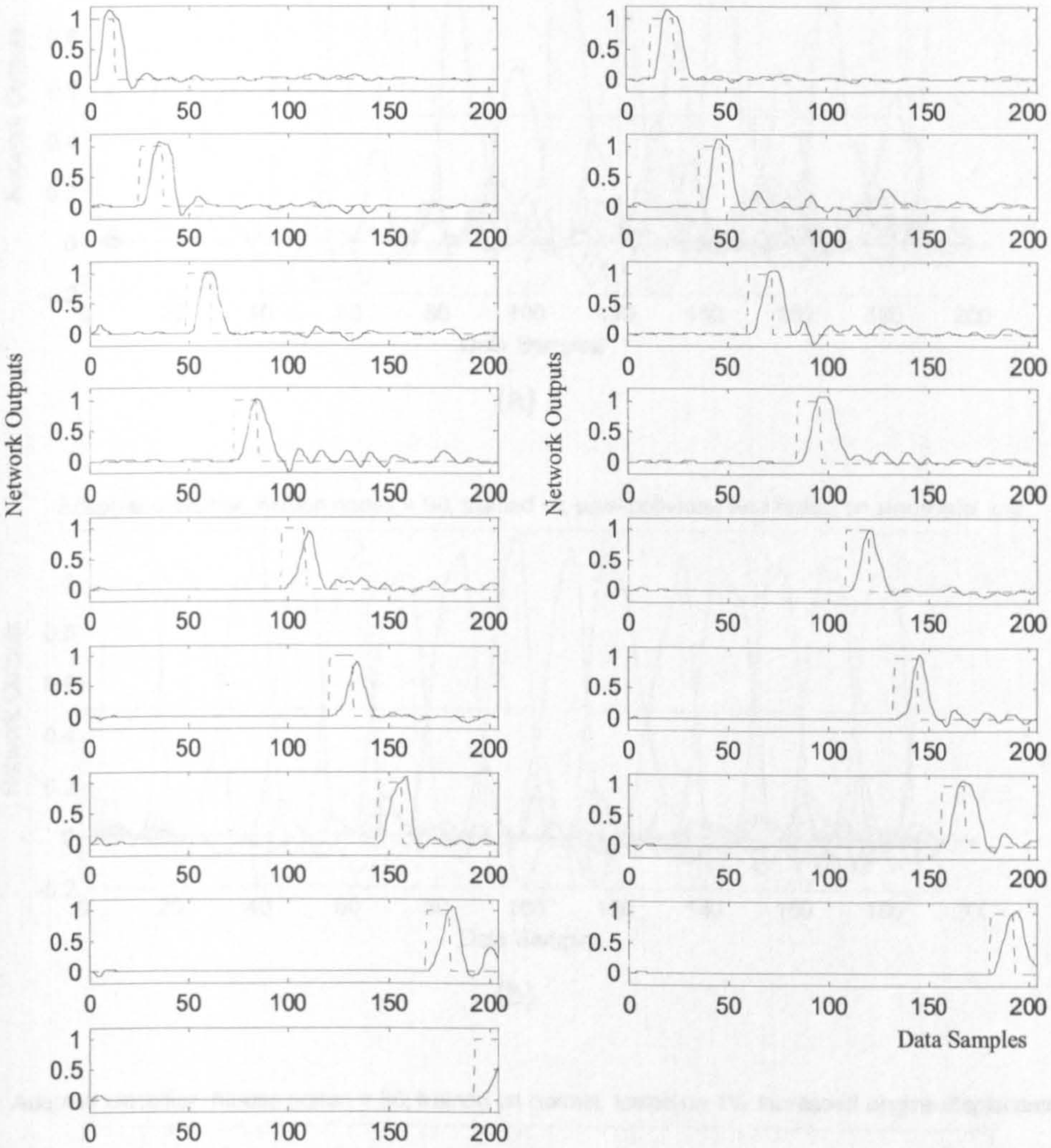
Fig. 6.12: Networks trained on fixed reference and tested on saw-tooth reference, sinusoidal load change and 1% increased engine displacement: (a) Non-adaptive; (b) Adaptive classifier; (c) Details of each fault classification in b shown separately for clarity.

6.7 FDI WITH DATA FILTERATION

In order to improve the problem of false alarms, the signal processing toolbox in Matlab is utilised and a third order low-pass digital filter is designed to suppress the spikes in the resultant data. A Butterworth digital filter can be designed using the Matlab function "butter". The function has two arguments N and W_n, for the order of the filter and cut-off frequency, respectively. The function returns the filter coefficients in length N+1 vectors B and A, numerator and denominator, respectively. The cut-off frequency must be 0.0 < W_n < 1.0, with 1.0 corresponding to half the sample rate. The value of W_n is to be chosen carefully. A high value may not do any filtration at all whereas a very low value may cause a long time delay and poor filtration.

A value of 0.1 has been carefully chosen for W_n which reduces the spikes to half of their original height (i.e. much below the threshold of 0.5) and causes little time delay. The false alarms are practically reduced to zero times as shown in Fig. 6.13 (a) and (b).



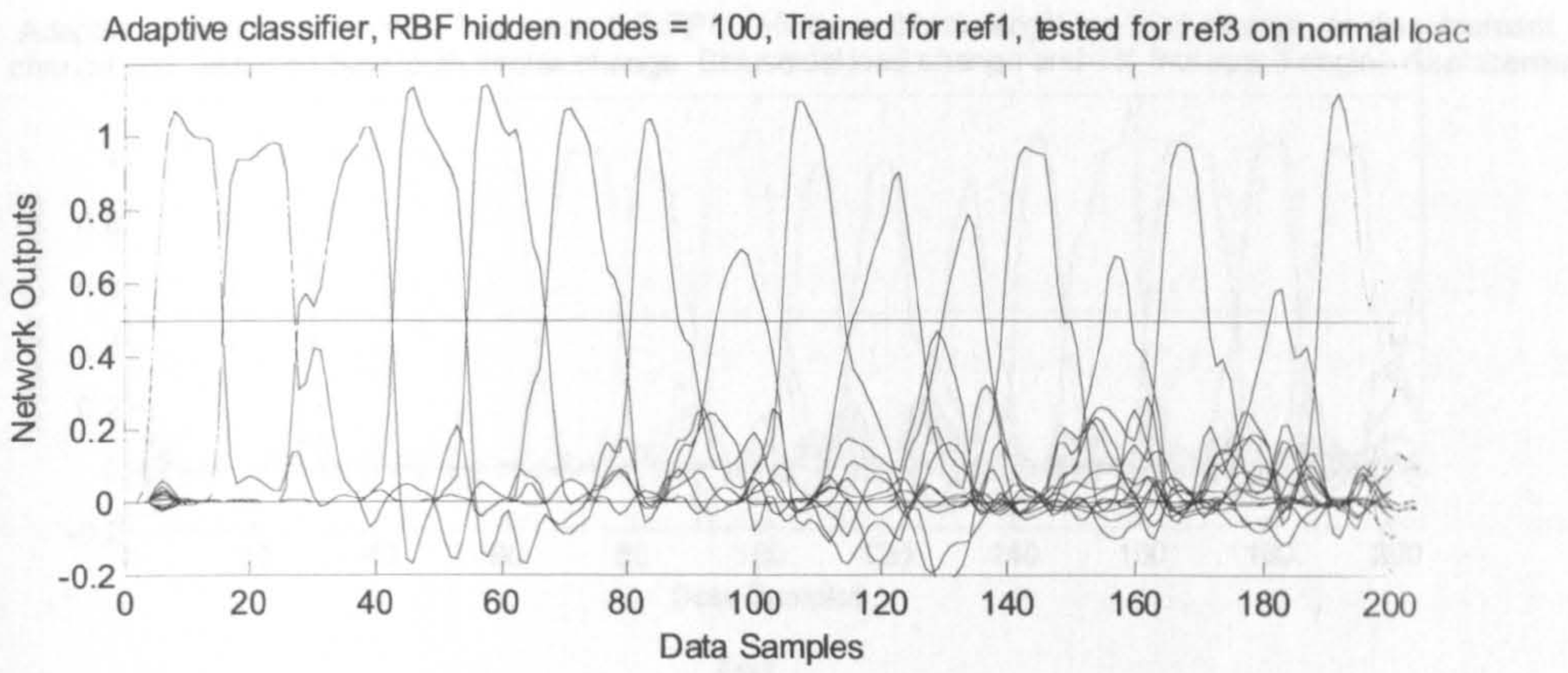


Left and right hand columns show results for state No. 1,3,5,7,9,11,13,15,17 and 2,4,6,8,10,12,14,16 respectively

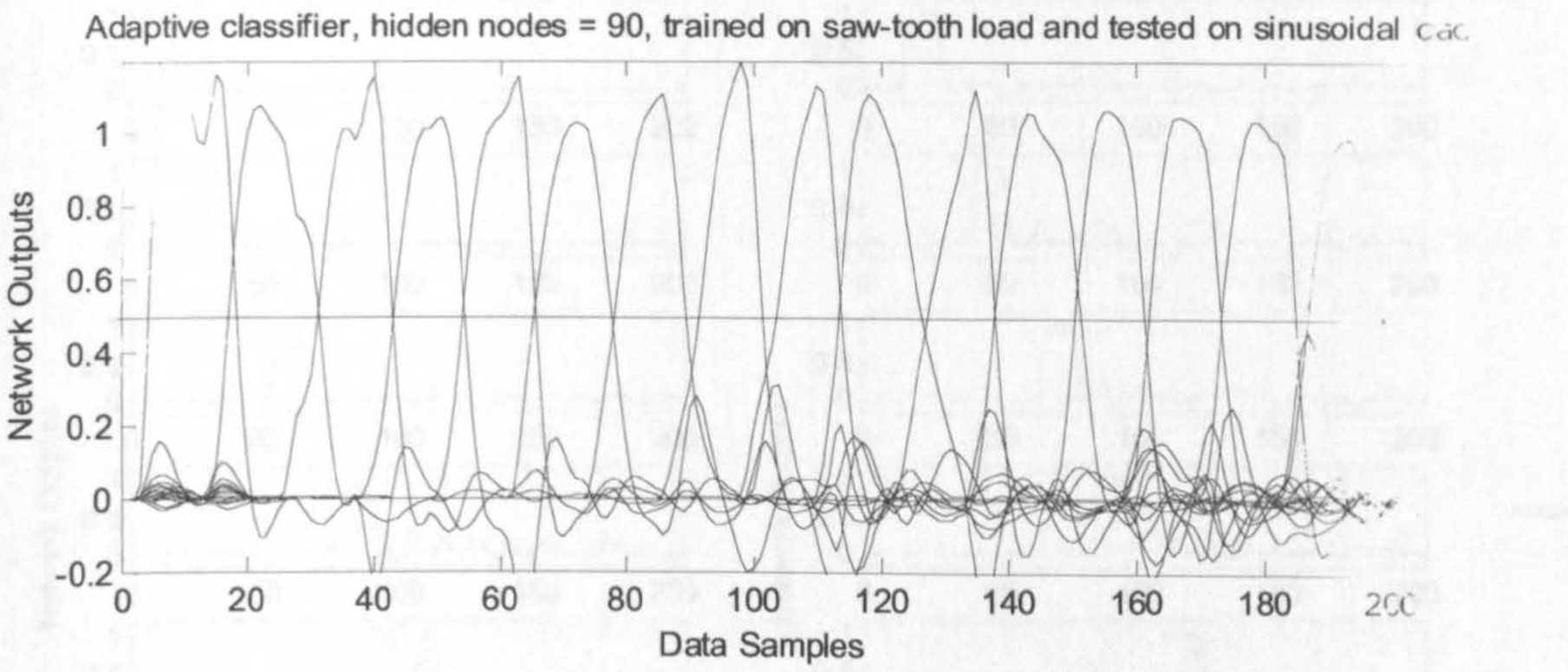
(b)

Fig. 6.13: Classification results after using a low-pass Butterworth filter.
(a) Comparison of classification results before and after filtration for fault number 4; (b) Classification results of 13; (c) after filtration.

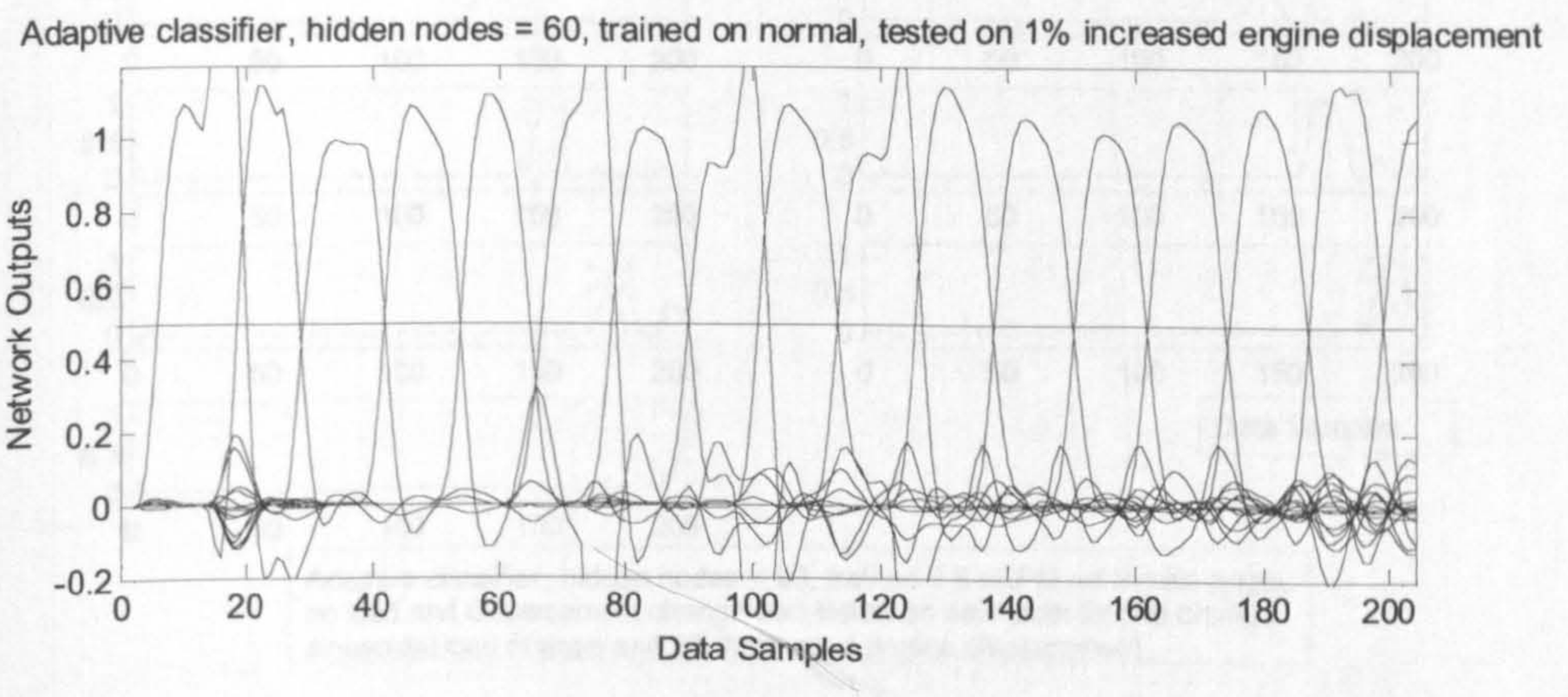
It can be seen from graphs in Fig. 6.13 that the spikes causing false alarms have been filtered out and make the classification more robust and reliable. Low-pass filtered results for Fig. 6.5, Fig. 6.10 and Fig. 6.11 are shown in Fig. 14.



(a)



(b)



(c)

Fig. 6.14: Filtered results for Fig. 6.5, Fig. 6.10 and Fig. 6.11 respectively in (a), (b) and (c)

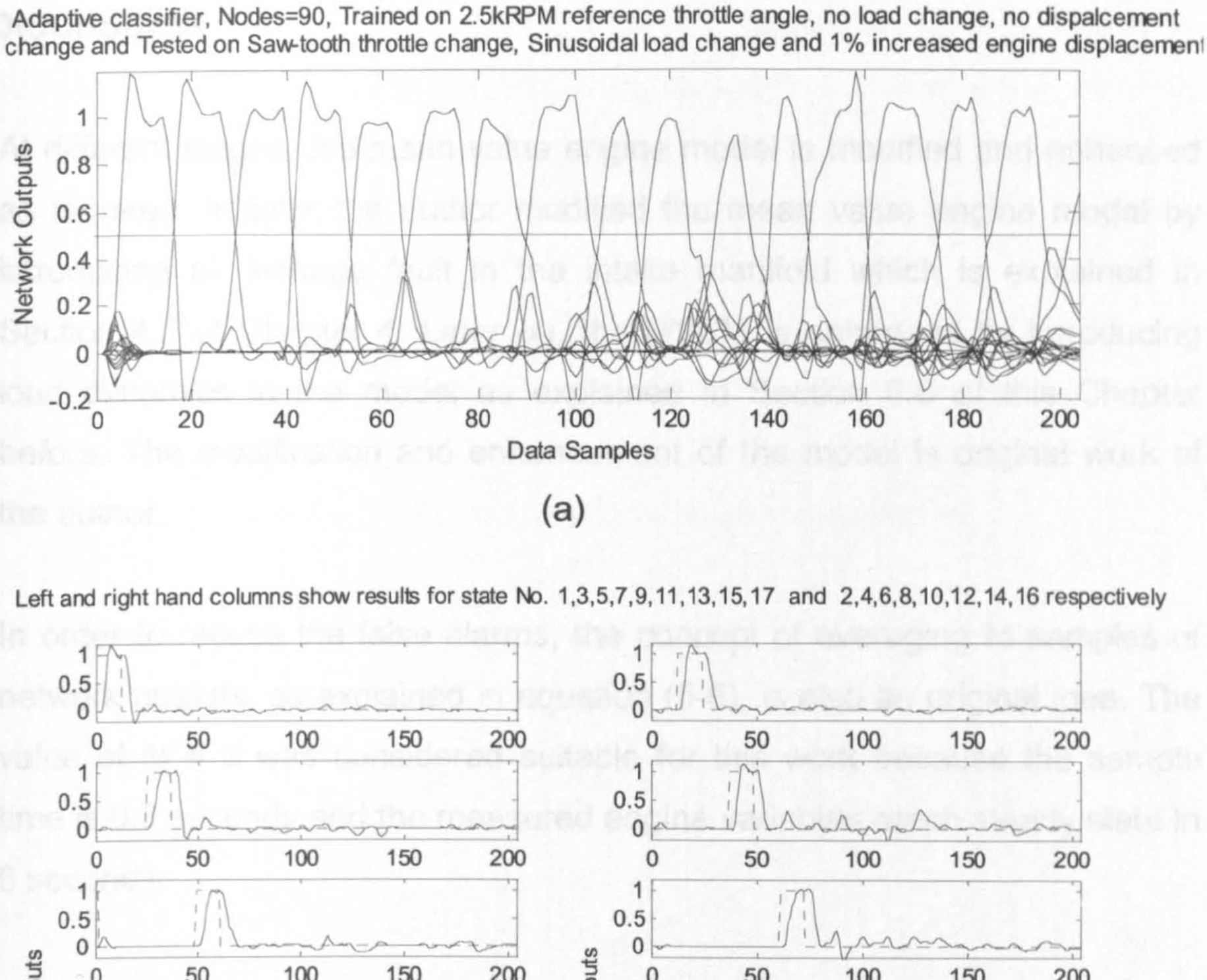


Fig. 6.15: (a) Filtered results for Fig. 6.12 (b) Details of each fault classification separately shown for clarity

6.8 DISCUSSION

At different stages, the mean value engine model is modified and enhanced as required. Initially, the author modified the mean value engine model by introducing air leakage fault in the intake manifold which is explained in Section 4.2 of Chapter 4. Later on, the MVEM is enhanced by introducing load dynamics in the model as explained in Section 6.6 of this Chapter before. The modification and enhancement of the model is original work of the author.

In order to reduce the false alarms, the concept of averaging M samples of network outputs, as explained in equation (5-6), is also an original idea. The value of M = 3 was considered suitable for this work because the sample time is 0.3 seconds and the measured engine variables reach steady state in 6 seconds.

The robustness of the adaptive RBF-based FDI system is investigated for a wide range of operational modes with increasing generality. Robustness assessment has been carried out against fixed and sinusoidal throttle angle inputs (i.e. set point change), change in load, change in engine parameter, and all these changes occurring at the same time for both adaptive and non-adaptive networks. The adaptive network performs very well and the simulation test results are satisfactory after data filtration for all the sixteen faults considered. The non-adaptive classifier fails to cope up with the load change, parameter change, etc. Thus, it is not robust whereas the adaptive network classifies all the faults correctly. The false alarm is reduced to zero by the use of a low-pass filter at the output of the neural network before making a decision whether a particular fault has occurred or not.

The adaptive fault diagnosis method outperformed the non-adaptive method in the following ways:

- It could handle bigger set of faults,
- It is more robust to throttle angle, and is robust to load and engine parameter variation, and
- The size of the required RBF neural network is small.

A neural network of smaller size is easier to realise and can work faster. Smaller network would require less computational time and therefore could work in real time. The adaptive methodology developed for engine fault diagnosis in this research is a novel piece of work and is author's contribution to knowledge.

Robustness assessment against different types of unknown faults and simultaneously occurring multi-faults are not considered and robustness evaluation for real engine data is carried out in the next chapter.

CHAPTER 7

REAL DATA EVALUATION

7.1 INTRODUCTION

In this chapter five different sensors in automotive engine have been investigated for positive and negative bias faults on real engine data. The sensors considered are throttle angle position, crankshaft speed, torque, and intake manifold pressure, and intake manifold temperature sensors. The faults considered are common and realistic, and have been considered previously by several researchers (Antory, 2005; Nyberg and Stutte, 2004; Capriglione *et al.*, 2004; Vinsonneau *et al.*, 2001). The throttle angle position, crankshaft speed and manifold pressure sensors have also been previously investigated by Capriglione *et al.*, (2003) and implemented on a digital signal processor (DSP). The intake manifold pressure and temperature sensors have also been previously investigated in Chapter 4, 5 and 6 on simulated data from MVEM (Sangha *et al.*, 2005b, 2006 and 2007). It was not practical to make a leakage in the intake manifold of a real running engine at a desired time and therefore gas leakage fault is not considered for real data evaluation. The automotive engine available in the workshop for experimentation does not have an EGR system and therefore EGR valve clogged fault is also not investigated for real data evaluation.

A small bias in a sensor can adversely affect the engine efficiency especially when the sensed value is directly or indirectly utilised in the electronic control unit (ECU) of the engine for engine control. Some sensor faults can even lead to increased pollution due to less efficient engine operating state. Timely

detection of faults can prevent the development of possible catastrophes in the near future. Some sensor faults with which the sensor output is used for feedback control will affect both dynamic and steady state performance. Therefore, sensor fault detection and isolation is important for automotives. An adaptive on-line fault diagnosis algorithm derived in Chapter 5 is evaluated on real engine data to cope with engine-to-engine variations, parameter uncertainty, disturbances, variations due to environmental changes and aging of the engine and measurement devices.

It is important to mention here that sensor failures investigated in (Capriglione et al., 2007) e.g. short circuit (zero deflection), open circuit (full deflection), hold (reading-halt) and short circuit between two sensors, are not considered in this research due to the fact that these are easier to be diagnosed compared to a sensor bias fault as small as 2%.

7.2. EXPERIMENTAL SET-UP

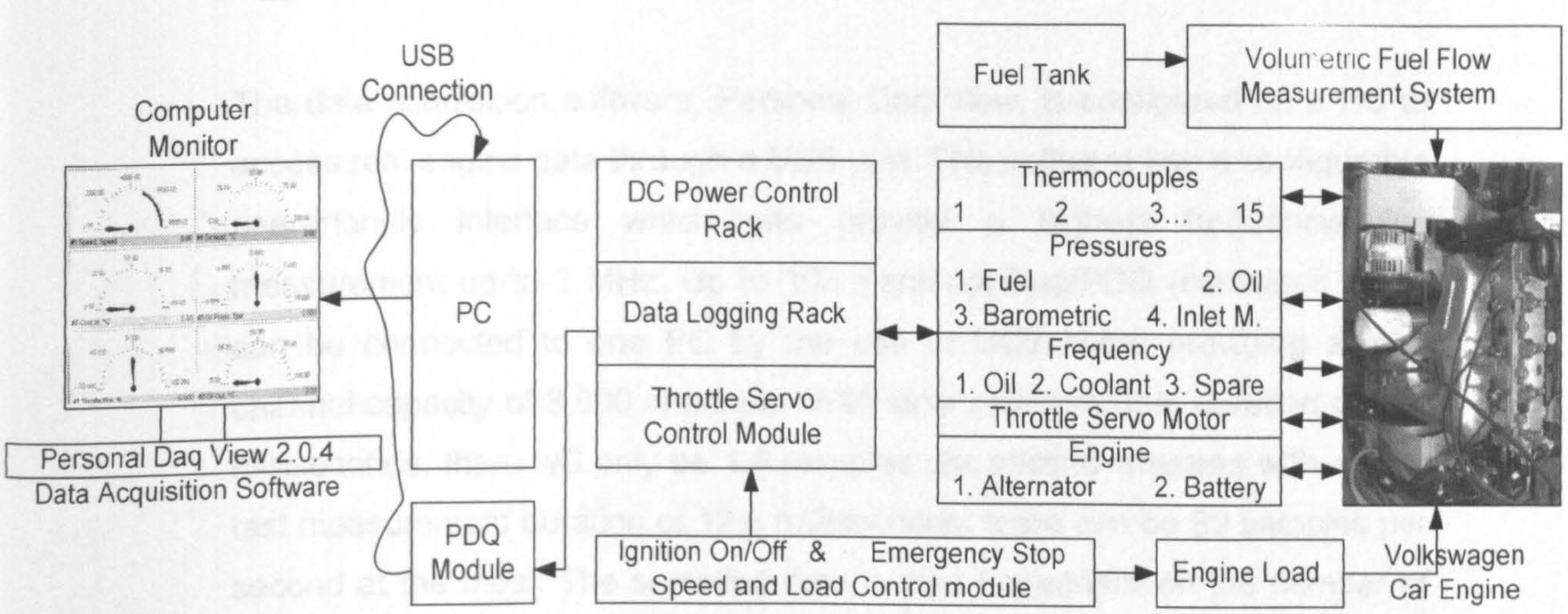


Fig. 7.1: Schematic diagram of engine test bed

A Volkswagen petrol car engine test bed is used for real engine data collection. Fig. 7.1 shows the schematic diagram of the experimental set-up. Main details of the engine specifications are given in Table 7.1. The engine test bed has a provision for 15 thermocouple inputs, 24 digital I/O and 2

frequency/pulse differential signals from transducers at different locations in the entire engine system.

Table 7.1: Volkswagen car engine specifications

Engine Code	Number of Cylinders	Nominal output in kW/rpm	Maximum torque N-m/rpm	Bore/ Stroke in mm	Capacity in cm^3	Compression
ATE	4	37/5000	86/3000... 3600	67.1/70.6	999	10.5

The interface hardware of the test bed, Personal Daq (PDQ), provides a connection between the computer USB port and the data logging rack. The data logging rack is directly connected to the transducer connection panel through cables. All the transducers installed in the engine test bed are connected to the transducer panel to provide an interface with the data logging rack.

The data acquisition software, Personal Daq View, is configured on a PC to access real engine data through a USB port. This software has a configurable user-friendly interface which can provide a highest frequency/pulse measurement up to 1 MHz. Up to 100 Personal Daq/PDQ (combined units) can be connected to one PC by the use of USB hubs; providing a total channel capacity of 8,000 channels. With slow measurement duration of 610 milliseconds, there will only be 1.6 samples per second whereas with a very fast measurement duration of 12.5 milliseconds, there can be 80 samples per second at the most. The sampling time is also dependent on the number of different variables to be acquired. A sample time of 1.4 seconds was found appropriate for this experiment to catch transient dynamics of the variables required for fault diagnosis.

7.3 REAL DATA ACQUISITION

For fault detection and isolation via neural networks, training and testing data sets are required and all of which should contain samples with no fault and all

faults considered. It is impracticable to produce real faults during engine operation and to acquire training data for every possible operating mode of an engine. Therefore a suitable set of possible faults was simulated on the basis of data acquired in fault-free condition which is explained in section 5.1.

Data acquisition is one of the most important parts of these experiments. The data is acquired in many different operating states of the engine to cover the entire range of normal engine operation. The engine is carefully operated to run in different speed and torque conditions to cover the maximum spectrum of real engine runs on road. The different sets of acquired engine data are shown in Table 7.2.

Table 7.2: Ranges of important variables in acquired real engine data

	Variation in % throttle angle position	Variation in crankshaft speed (RPM)	Variation in torque (N-m)
Data sets 1 & 2 (High speed variation run)	0.28 ~ 77	720 ~3976 (High variation)	-1.0 ~ 85
Data sets 3 & 4 (Approximately constant speed run)	20 ~ 50	2127 ~ 2192 (Almost Constant)	28 ~ 48
Data sets 5 & 6 (Low speed variation run)	4 ~ 80	1510 ~ 2870 (Low variation)	-0.25 ~ 85.8

The speed variation during experiments covers a wide range form nearly 700 rpm to 4000 rpm. The torque variation ranged from -1.0 N-m to 86 N-m. The maximum allowable torque for this engine is 86 N-m; and therefore the engine was run up to the maximum allowable torque. The engine data was acquired on different days to ensure different ambient conditions. Six different data sets were collected on different days and different sessions for training and testing of neural classifier. One raw data set is graphically shown in Fig. 7.2 for illustration.

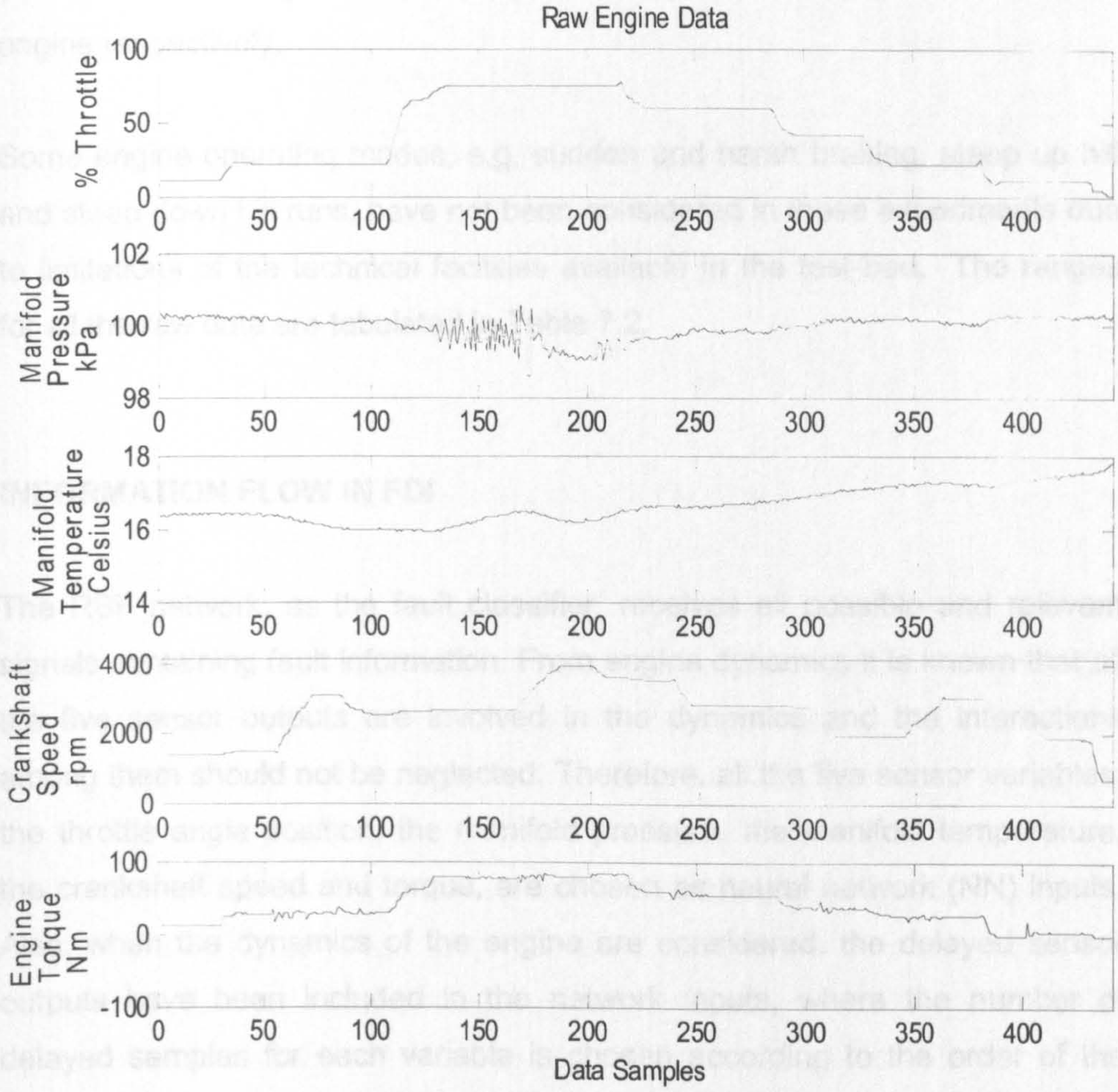


Fig. 7.2: Raw engine data in data set 2

The sampling time for the raw data in Fig. 7.2 was 1.4 seconds and a total of 441 samples were acquired and recorded on computer using the Personal Daq View software. Initially the engine was run idle for 10 minutes for proper heating up and then throttle position and load were simultaneously changed and necessary readings were recorded for 617.4 seconds as shown graphically in Fig. 7.2. In the beginning, the engine is run on low throttle (slow speed) for a minute as if run in congestion and then, gradually, the throttle is increased to increase speed up to nearly 4000 rpm to represent high-speed run. Then, the throttle is gradually decreased in steps to reduce the engine speed and this is how a high variation in engine speed is achieved. The load on the engine is also changed within allowable limits during data recording to achieve up hill and down hill running conditions of the engine. An increase

and decrease in engine load corresponds to up hill and down hill runs of the engine respectively.

Some engine operating modes, e.g. sudden and harsh braking, steep up hill and steep down hill runs, have not been considered in these experiments due to limitations of the technical facilities available in the test-bed. The ranges for all the raw data are tabulated in Table 7.2.

7.4 INFORMATION FLOW IN FDI

The RBF network, as the fault classifier, receives all possible and relevant signals containing fault information. From engine dynamics it is known that all the five sensor outputs are involved in the dynamics and the interactions among them should not be neglected. Therefore, all the five sensor variables, the throttle angle position, the manifold pressure, the manifold temperature, the crankshaft speed and torque, are chosen as neural network (NN) inputs. Also, when the dynamics of the engine are considered, the delayed sensor outputs have been included in the network inputs, where the number of delayed samples for each variable is chosen according to the order of the dynamics. Because, including the delayed outputs considerably increase the number of network inputs and lead to a much larger network size, which requires more computing time and possibly lower generalization ability and the classification performance has not been significantly improved. In this research, both ways of including and excluding the past output data in the network inputs are tested and the results are compared.

The network has 11 outputs with each indicating one of the investigated states: one for no-fault state, 5 for positive bias of the 5 sensors and another 5 for negative bias of the 5 sensors. The information flow for the fault diagnosis method is illustrated in Fig. 7.3.

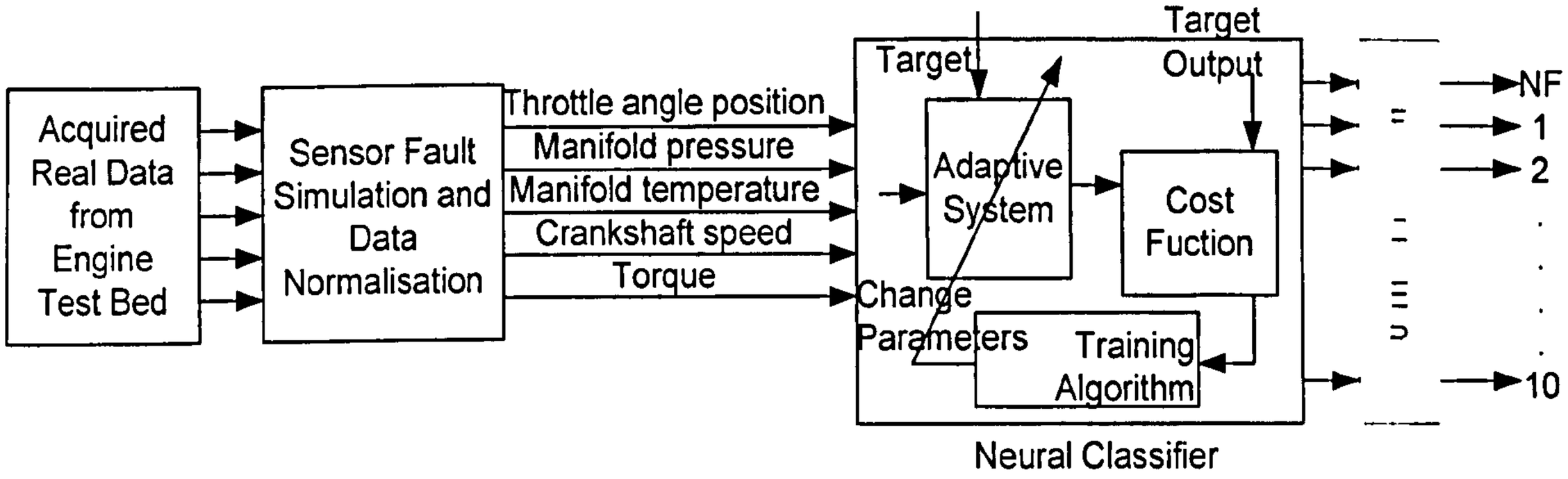


Fig. 7.3: Information flow of the fault diagnosis method

The acquired raw data is linearly normalised to the range of [0 1] and the different 10 sensor faults are simulated by superimposing on the normal sensor output with one fault at a time. Then all five inputs are fed to the adaptive classifier. Widths in the hidden nodes and the weights in the output layer of the RBF network are on-line adapted during the monitoring to minimise the sum-squared error between the output from the adaptive network and the pre-decided target output. Gradient descent method is used for adjusting the widths of the RBF network as explained in Chapter 5. The width in each hidden layer node is initially chosen as a constant using the *P*-nearest rule. The classification is sensitive to the Gaussian local function, which is mainly characterised by the width. Therefore, a gradient descent algorithm is used to on-line adapt the widths to achieve a minimal objective function as given in equation (5-1).

7.5 FAULT DIAGNOSIS

7.5.1 Fault Simulation on the Real Data

As real sensor faults with the amplitude as small as 2% and at some specific time periods are not easy to be introduced to the engine test bed in practice. Therefore, the sensor faults are simulated by superimposing a bias with a certain percentage of its normal sensor output on real data. This does not affect system dynamics and fault detection. Both +-2% and +- 5% bias are used and this is achieved using a multiplying factor (MF) of 1.02 and 0.98 and

1.05, 0.95 respectively. The 2% faults are shown in the Table 7.3 together with all the names of faults simulated.

Table 7.3: Simulated faults and their multiplying factors

State No.	Fault description	MF
1	No Fault (All the sensors reading correctly)	1.00
2	Throttle angle sensor 2% over reading	1.02
3	Throttle angle sensor 2% under reading	0.98
4	Manifold pressure sensor 2% over reading	1.02
5	Manifold pressure sensor 2% under reading	0.98
6	Manifold temperature sensor 2% over reading	1.02
7	Manifold temperature sensor 2% under reading	0.98
8	Crankshaft speed sensor 2% over reading	1.02
9	Crankshaft speed sensor 2% under reading	0.98
10	Torque sensor 2% over reading	1.02
11	Torque sensor 2% under reading	0.98

317 samples are used in each data set where one of 11 fault states is generated to all the samples of the set. This is repeated for 11 times, to form 317 x 11 = 3487 samples in the whole faulty data set. In this way the fault classification can be tested and viewed clearly.

7.5.2 Network Structure Selection and Initial Training

Before the network is used on-line, it needs to be pre-trained off-line. This is referred to as initial training. With 5 inputs and 11 outputs determined in Section 7.4, the remaining structure of the network to be determined is the number of hidden layer nodes. Different numbers of hidden nodes ranging from 5 to 35 were tried and network performance, in terms of good classification and time taken for computations, was analysed. Twenty hidden nodes were found appropriate for a satisfactory level of performance because more hidden nodes gave a slightly better performance but the computational

load on the microprocessor increases substantially in terms of real time processing. Therefore the structure of the network was taken as 5: 20: 11.

The target of network output in the training is given by a 11-dimension row vector with 0 or 1, a "0" (zero) output implies that the fault does not occur while a "1" (one) output implies that the fault occurs. As 11 fault states are simulated with each for 317 data samples, one target vector will be used repeatedly for 317 times and the vector has only one "1" corresponding to the fault and all the other entries are "zero". Thus, target matrix X_o has 317x11 = 3487 rows and 11 columns. Each column of X_o represents i^{th} target for one of the 11 states. Its first column has ones from the first row to the 317^{th} row and the other entries are zeros, the second column has ones from the 318^{th} row to the 634^{th} row and the other entries are zeros, the last column has ones from the 3171^{st} row to the 3487^{th} row and the other entries are zeros. This is shown in Fig. 7.4:

$$
\begin{array}{cc}
\text{Row Numbers} & X_o \\
1 \sim 317 & \begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 & 0 \\ 0 & 1 & \cdots & & & 0 \\ \vdots & \vdots & \ddots & & & \vdots \\ \vdots & & & \ddots & & \\ & & & & \ddots & 0 \\ 0 & 0 & \cdots & & 0 & 1 \end{bmatrix} \\
318 \sim 634 & \\
\vdots & \\
3171 \sim 3487 &
\end{array}
$$

Fig. 7.4: Target matrix X_o

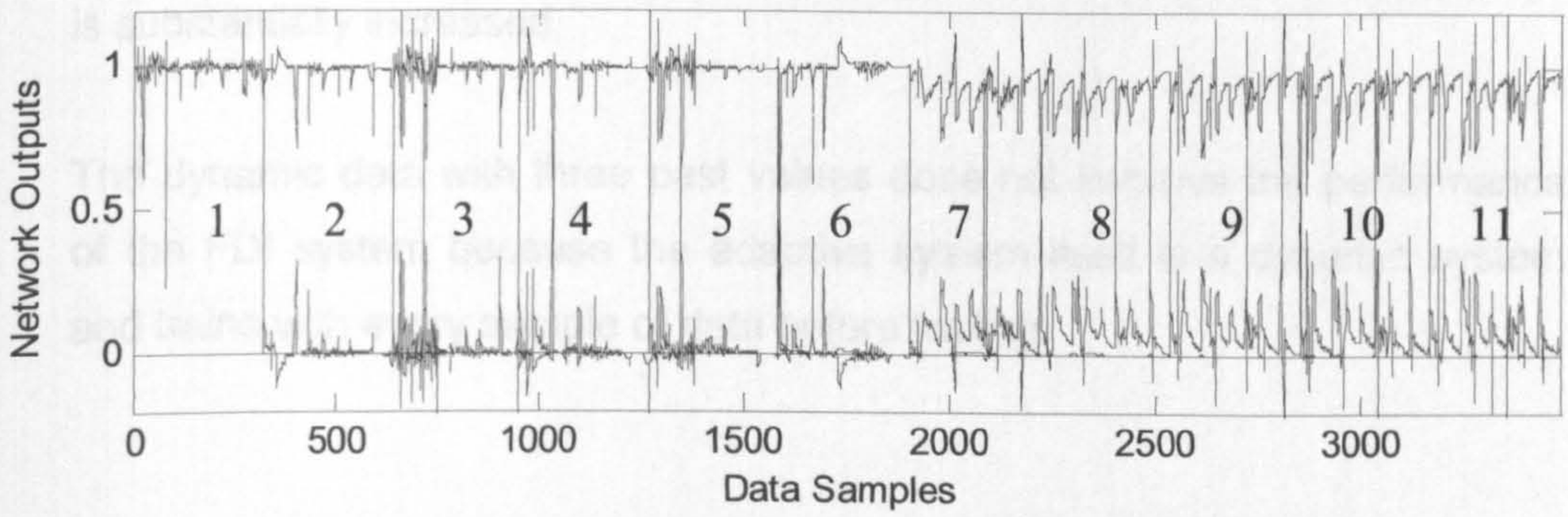
Thus, the transpose of the i^{th} row in X_o is used as the training target vector for the i^{th} training pattern. The centres are chosen using the *K*-means clustering algorithm from the training data set. The widths were initially chosen using the *P*-nearest neighbour's algorithm, and the weights were trained using the RLS algorithm.

7.5.3 Fault Classification Results

The adaptive network is used to diagnose faults with test data sets after initial training with the training data set. The fault detection threshold in equation (5-7) was chosen as $r_t = 0.5$ with M in equation (5-6) chosen as M = 3. This replies that one of the network outputs must be over 0.5 continuously for 3 samples, a fault is then believed occurred. High thresholds may lead to missed detections whilst low thresholds may cause more false alarms. A proper choice can be found in experiment and $r_t = 0.5$ is found as a good compromise between reliability of detection and insensitivity to noise in this application. The threshold for the gradient of the objective function in equation (5-8) was chosen as $\sigma = 0.00001$. The forgetting factor for the RLS algorithm was chosen as a constant value of $\lambda = 0.99$.

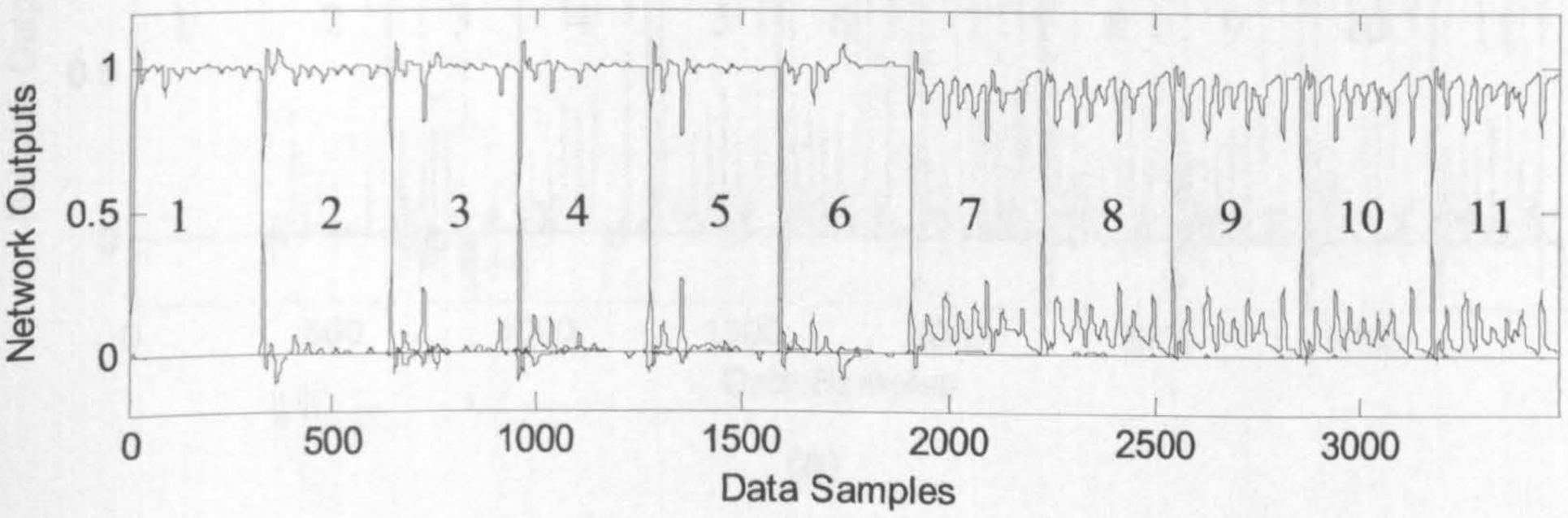
The adaptive network is initially trained on data set 6 for 5% sensor faults and tested on other data sets. The results for network testing on data set 5 with 2% fault are shown in Fig. 7.5.

Adaptive classifier, Hidden Nodes=20, Trained on set6,
tested on set5 of real engine data for 2% sensor faults

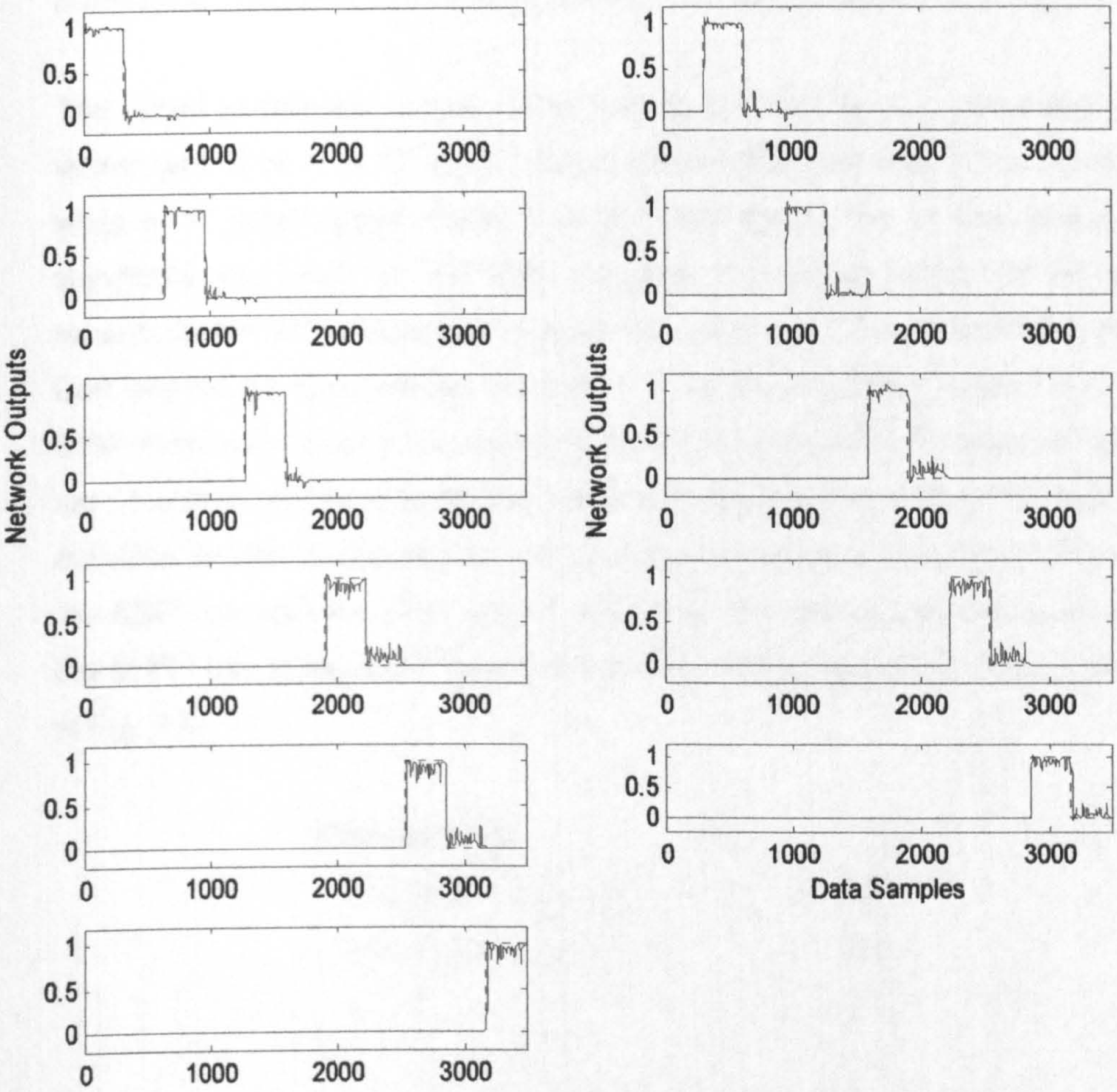


(a)

Adaptive classifier, Hidden Nodes=20, Trained on set6,
tested on set5 of real engine data for 2% sensor faults



(b)



Left and right hand columns show results for state No. 1,3,5,7,9,11 and 2,4,6,8,10 respectively

(c)

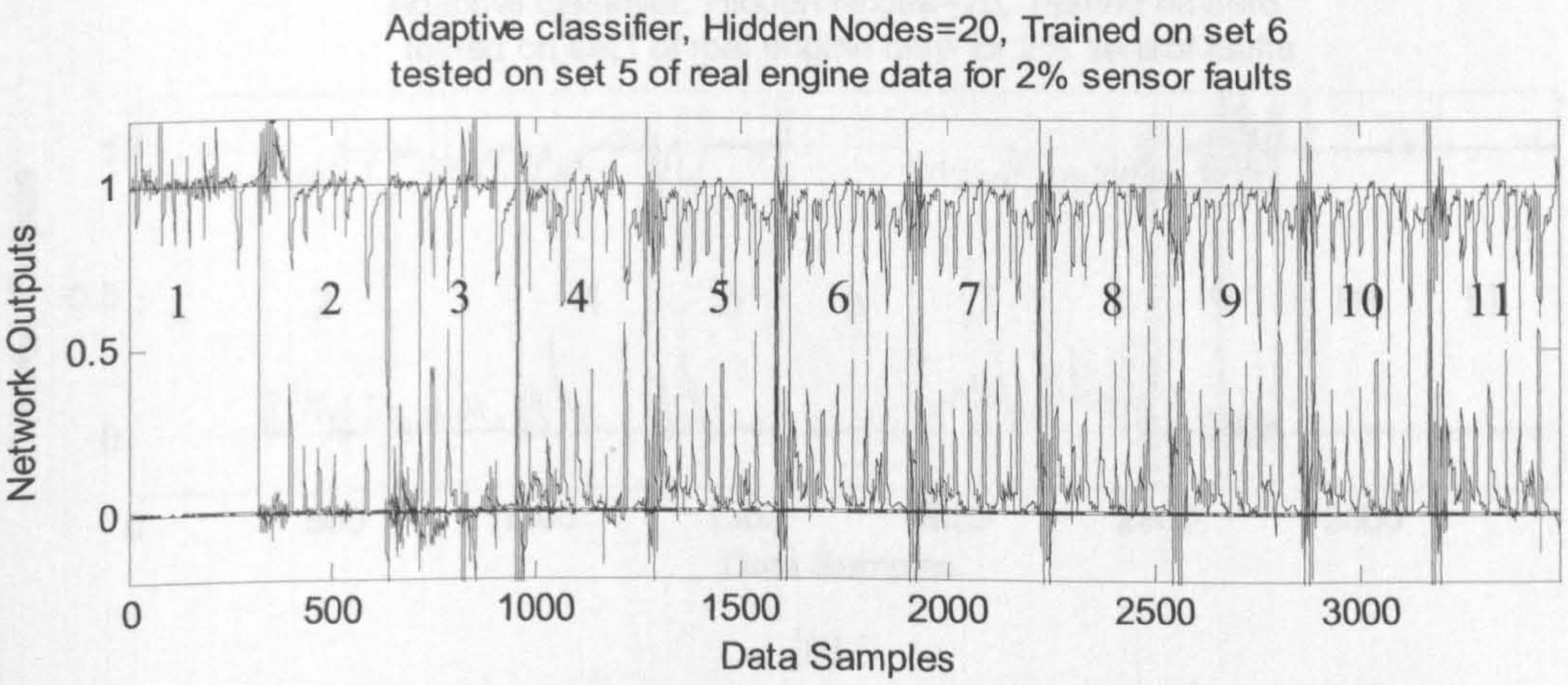
Fig. 7.5: Classification result for 2% faults when the network is trained on data set 6 and tested on data set 5. (a) Without data filtration (b) after low pass data filtration (c) Each state separately shown for clarity

There would be misclassifications for unfiltered data as shown in Fig. 7.5(a). There are a number of spikes crossing the threshold of 0.5 which may cause false alarms if their average value in three samples is over 0.5. To further reduce the false alarms the classifier outputs are filtered by a low-pass filter. The low-pass Butterworth filter attenuates the high frequency spikes and the result for the same fault is shown in Fig. 7.5b. There are still a few spikes

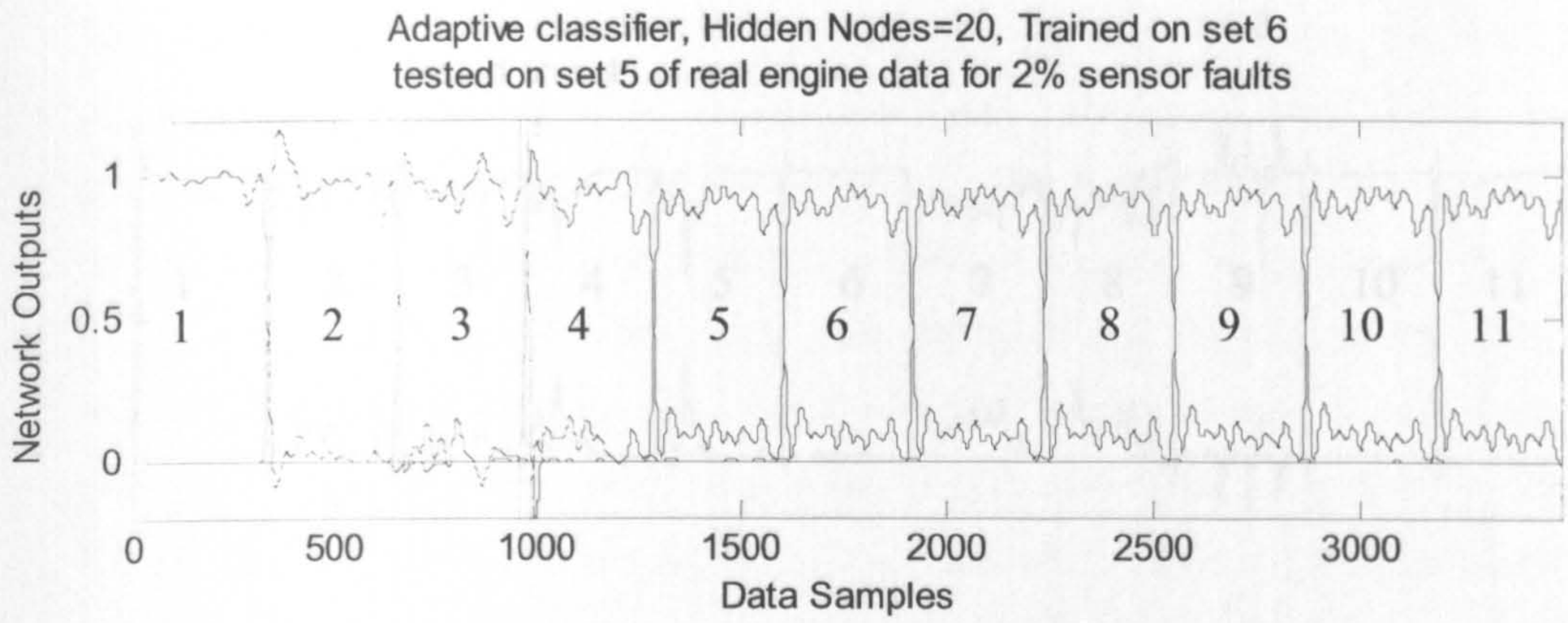
visible but these do not cross the threshold of 0.5 and therefore cannot cause a false alarm. To analyse the resultant classification, all the 11 states are shown separately in Fig. 7.5c for better visibility. It can be seen that all the ten fault states and the no fault state are clearly classified and the results are satisfactory. But in this fault diagnosis system, only the current values of the five variables were used as inputs to the adaptive neural network for training as explained in information flow of the FDI method before in Fig. 7.3.

In order to improve the results further, three past values of dynamic data are used as input to the neural network for training for all the five variables along with the current values. Therefore, there are 20 inputs to the neural network viz. manifold pressure [p(k), p(k-1), p(k-2), p(k-3)], manifold temperature [t(k), t(k-1), t(k-2), t(k-3)], crankshaft speed [n(k), n(k-1), n(k-2), n(k-3)], throttle angle position [th(k), th(k-1), th(k-2), th(k-3)] and torque [q(k), q(k-1), q(k-2), q(k-3)] . This will substantially increase the size of the network to 20x20x11 instead of 5x20x11. The test results are shown in Fig. 7.6. When these results are compared with the previous results shown in Fig. 7.5, it is observed that there is a very little improvement in the results but the size of neural network is substantially increased.

The dynamic data with three past values does not improve the performance of the FDI system because the adaptive system itself is a dynamic system and trains with every sample of data before testing.



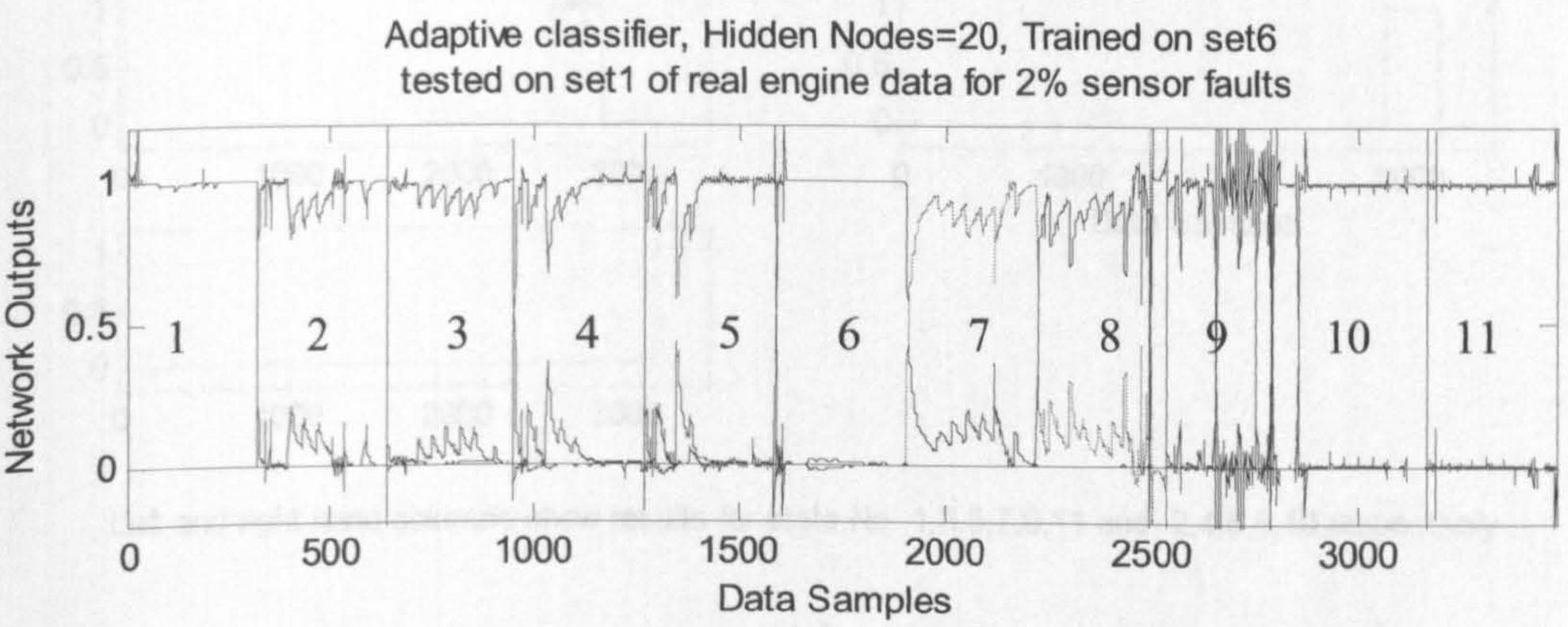
(a)



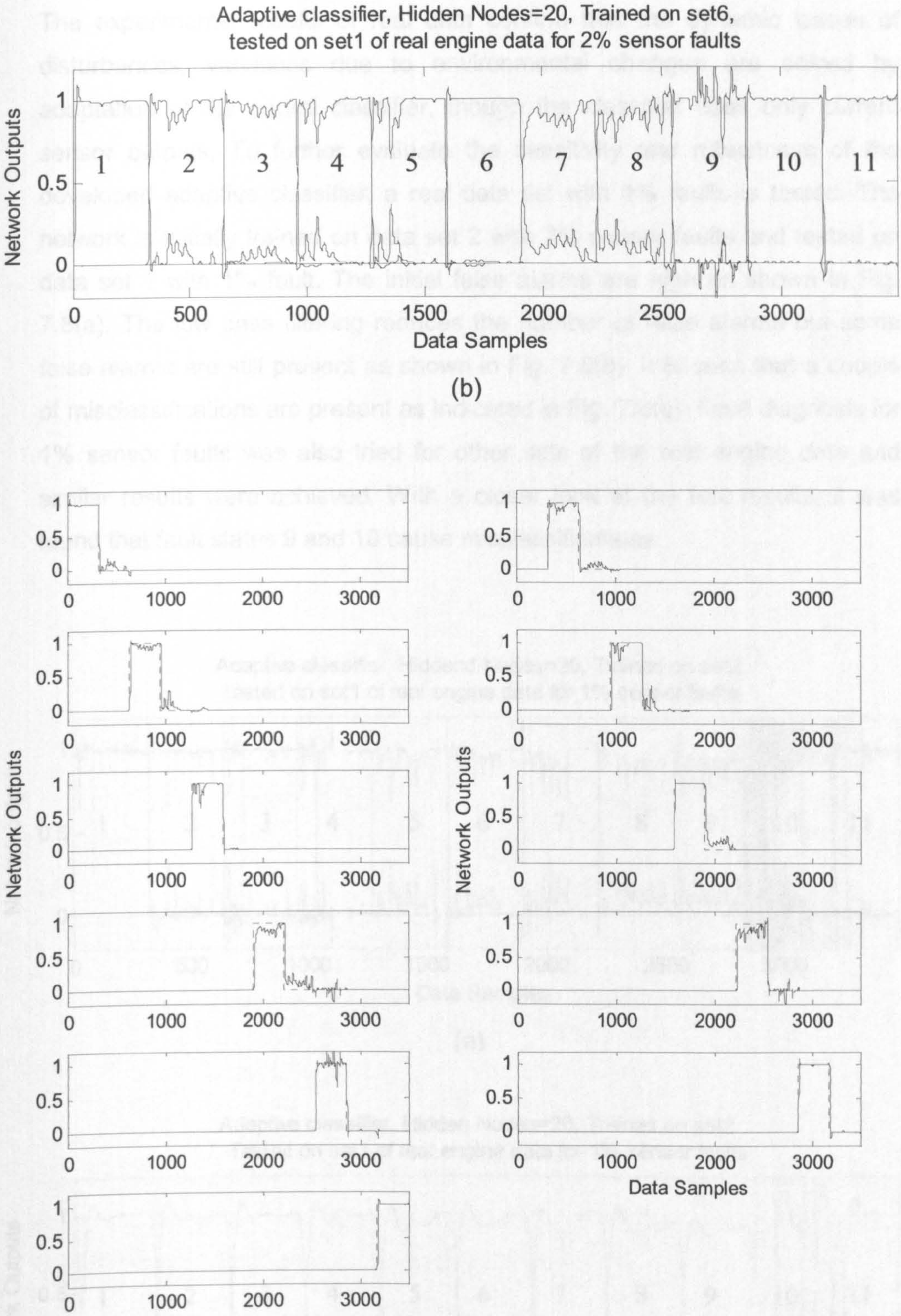
(b)

Fig. 7.6: Classification result for 2% faults when the network is trained on data set 6 and tested on data set 5 using three past and present values of all the five variables (a) Without data filtration (b) after low pass data filtration

Another test result for network initial training on data set 6 with 5% faults and testing on data set 1 with 2% faults is shown in Fig. 7.7. Fig. 7.7c shows the results are satisfactory with all the ten sensor faults and no fault state clearly classified with no false alarms or misclassification. In order to improve the results further, three past values of dynamic data are used as input to the neural network for training for all the five variables along with the current values as before. But it does not make any noticeable difference or improvement.



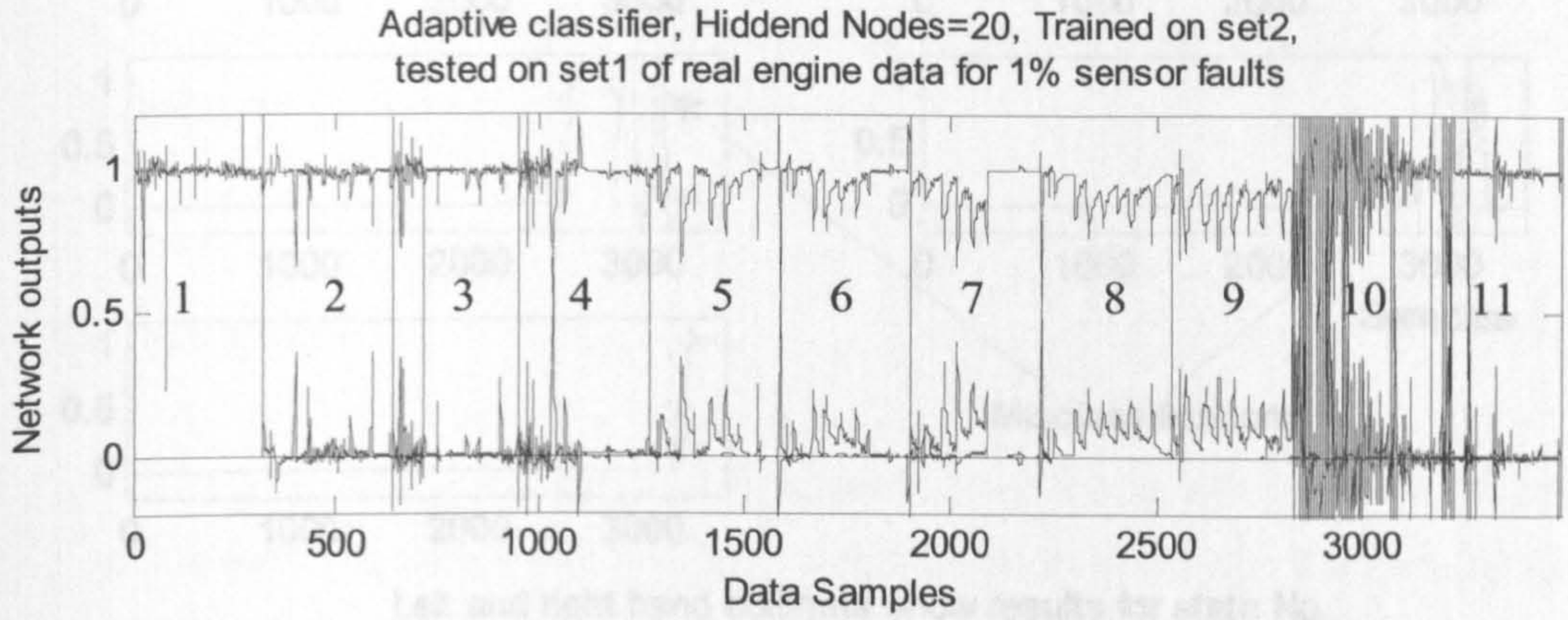
(a)



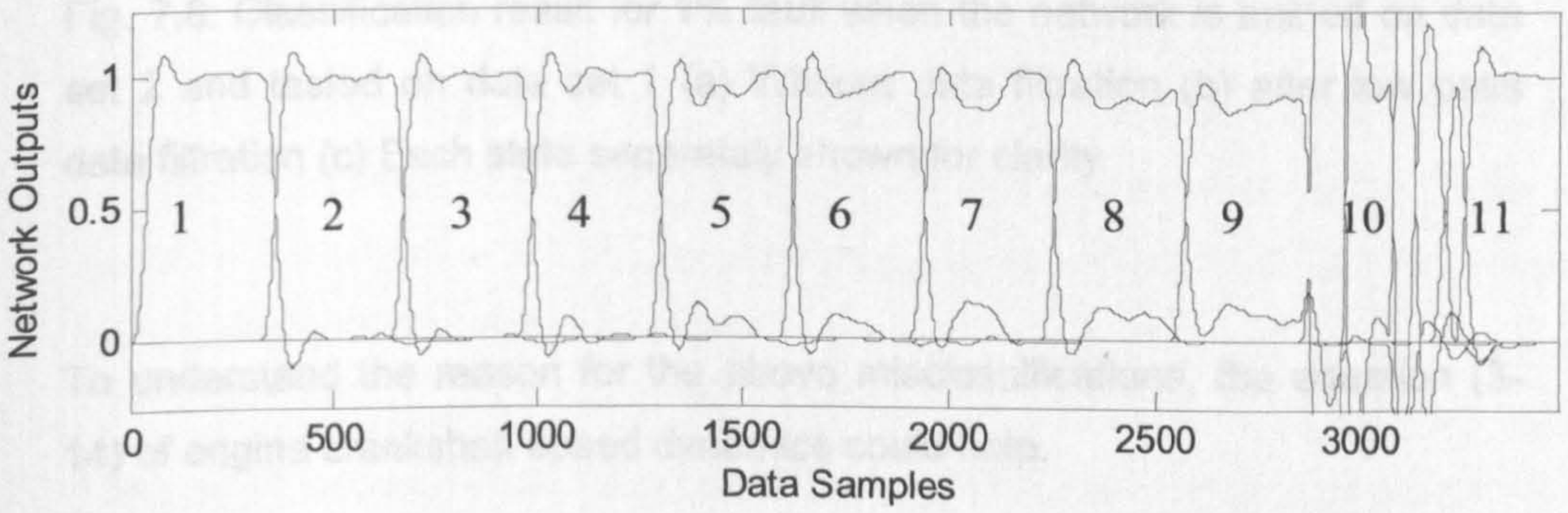
Left and right hand columns show results for state No. 1,3,5,7,9,11 and 2,4,6,8,10 respectively

Fig. 7.7: Classification result for 2% faults when the network is trained on data set 6 and tested on data set 1. (a) Without data filtration (b) after low pass data filtration (c) Each state separately shown for clarity

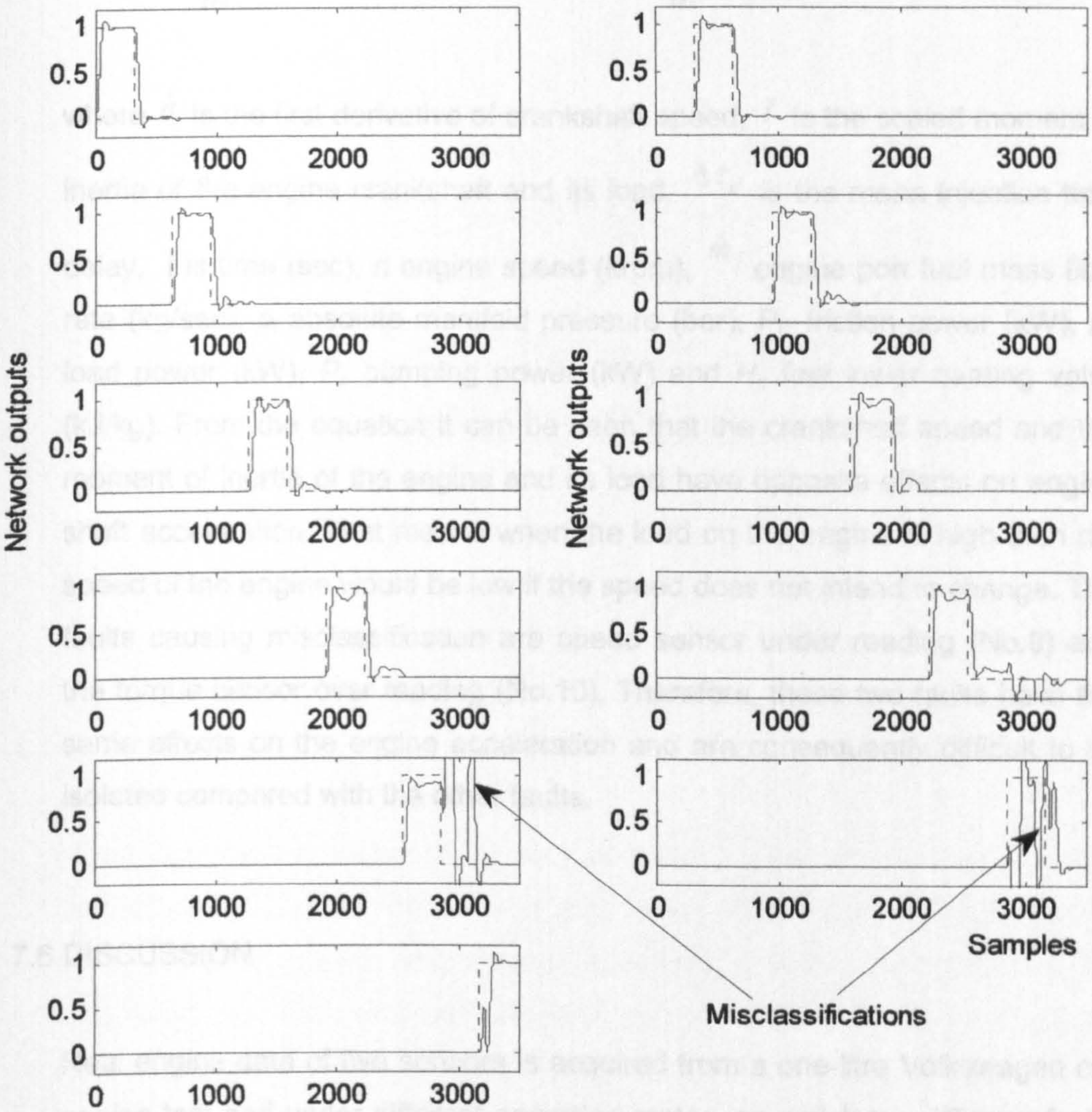
The experimental results of real data confirm that the dynamic issues of disturbances, variations due to environmental changes are solved by adaptation of the neural classifier, though the classifier uses only current sensor outputs. To further evaluate the sensitivity and robustness of the developed adaptive classifier, a real data set with 1% faults is tested. The network is initially trained on data set 2 with 2% sensor faults and tested on data set 1 with 1% fault. The initial false alarms are high as shown in Fig. 7.8(a). The low pass filtering reduces the number of false alarms but some false alarms are still present as shown in Fig. 7.8(b). It is seen that a couple of misclassifications are present as indicated in Fig. 7.8(c). Fault diagnosis for 1% sensor faults was also tried for other sets of the real engine data and similar results were achieved. With a closer look at the test results, it was found that fault states 9 and 10 cause misclassifications.



(a)



(b)



Left and right hand columns show results for state No.
1,3,5,7,9,11 and 2,4,6,8,10 respectively

(c)

Fig. 7.8: Classification result for 1% fault when the network is trained on data set 2 and tested on data set 1 (a) Without data filtration (b) after low pass data filtration (c) Each state separately shown for clarity

To understand the reason for the above misclassifications, the equation (3-14) of engine crankshaft speed dynamics could help.

$$\dot{n} = -\frac{1}{In}\left(P_f(p_i,n)+(P_p(p_i,n)+P_b(n)\right)+\frac{1}{In}H_u\eta_i(p_i,n,\lambda)\dot{m}_f(t-\Delta\tau_d)$$

where $\dot{n}$ is the first derivative of crankshaft speed, I is the scaled moment of inertia of the engine crankshaft and its load, $\Delta\tau_d$ is the mean injection time delay, t is time (sec), n engine speed (krpm), $\dot{m}_f$ engine port fuel mass flow rate (kg/sec), p_i absolute manifold pressure (bar), P_f friction power (kW), P_b load power (kW), P_p pumping power (kW) and H_u fuel lower heating valve (kJ/kg). From the equation it can be seen that the crankshaft speed and the moment of inertia of the engine and its load have opposite effects on engine shaft acceleration. That means when the load on the engine is high then the speed of the engine would be low if the speed does not intend to change. The faults causing misclassification are speed sensor under reading (No.9) and the torque sensor over reading (No.10). Therefore, these two faults have the same effects on the engine acceleration and are consequently difficult to be isolated compared with the other faults.

7.6 DISCUSSION

Real engine data of five sensors is acquired from a one-litre Volkswagen car engine test bed under different operating states, on which ten different faults are superimposed. The fault detection and isolation scheme using adaptive RBF classifier is employed to diagnose faults from real data. The experimental results confirm that the sensor faults as small as 2% are clearly detected and isolated for different data sets.

The training and testing of the adaptive neural network is investigated both with static and dynamic data but results are more or less the same. This is because the adaptive FDI system itself is dynamic and trains dynamically with each sample of data before testing.

The neural network classifier is on-line adapted for its widths and weights to cope with model uncertainty, time varying dynamics and environment change, so that the sensitivity of the scheme to the faults and the robustness to the uncertainty and disturbances are maintained.

CHAPTER 8

UNKNOWN FAULT DETECTION AND FAULT ACCOMMODATION

8.1 INTRODUCTION

Nowadays a large number of sensors are used in a car for improved reliability and increased comfort. Sensors used by motor vehicle systems are following a trend towards greater integration of processing power in the actual sensors. Integrated pre-processing unit and analogue-to-digital converter in a sensor, make the signal interference proof. Sensors with local intelligence, known as 'intelligent sensors', are also available commercially but are very expensive. This level of integration allows built-in monitoring and diagnostic ability in a sensor, which leads to much improved reliability and control of the vehicle. The correctness of operation of an electronic control unit (ECU) for a car engine depends upon a number of measurements, e.g. crankshaft speed, inlet manifold pressure, throttle position, air fuel ratio, etc. A growing demand on security and comfort has pushed an increased use of suitable sensors and actuators. In-vehicle conditions are optimised using environmental sensors for evaluating the outside conditions, e.g. road condition, visibility, presence of ice and intensity of rain, etc. Anti-crash sensors, global positioning system (GPS), anti-skid braking and anti-spin traction control, etc. are used in most of the new generation of cars for improved security, fuel efficiency and comfort. Most importantly, the driver status/condition is also monitored on-line by a more complex measurement system in high end cars, to avoid crashes due to

human errors. As operations under any faulty sensor condition will cause high cost, low security and low comfort, automotive measurement system should also be sensor fault tolerant like aircrafts and nuclear power plants.

Sensor fault correction is essentially a three fold system. The first stage is fault detection, which indicates occurrence of a known or unknown fault. The second stage involves establishment of type and location of the fault and this is called fault isolation. When a fault is detected and isolated, it is desirable to auto-correct the fault during operation if possible, before the vehicle is scheduled for repair. The auto correction of fault may not be possible for some components and actuator faults; but some known sensor faults may be corrected according to their known no-fault characteristics. This is known as sensor fault accommodation.

Sensor fault accommodation for three different sensors along with unknown fault detection has been investigated in this chapter. Once a sensor fault is detected then using pattern classification approach, the accommodation is achieved by replacing the sensor values with the predicted values. The present values of the faulty sensor are not used as an input to the neural predictor. The values for the faulty sensor are predicted on the basis of the no fault behaviour of different sensors. The predictor used healthy sensors as input and predicts a suitable value for the faulty sensor. The results demonstrated acceptable average prediction error for the faulty sensor.

Three different unknown faults have been investigated for detection while neural network is trained for four different sensor faults. Unknown fault detection is carried out using data with 'unknown faults' simulated in MVEM. When none of the 'no-fault' or 'fault' output neurons exceed the threshold, it can be interpreted as detection of 'unknown fault' (Li *et al.*, 2002). This process is also known as novelty detection (Bishop, 1994).

When none of the 'fault' output neurons exceed the threshold then it implies that none of the 'known-faults' has occurred and when 'no-fault' output neuron does not exceeds the threshold then it implies that some fault has occurred; it may be a known fault or an unknown fault. But when both the 'no-fault' output neuron and the 'fault' neurons do not exceed the threshold simultaneously then it implies that an 'unknown fault' has occurred. This method has been

used in this chapter for detection of 'unknown faults'. The results demonstrate successful detection of unknown faults and also demonstrate that some small unknown faults may not be detectable using this method.

8.2 SIMULATION OF FAULTS

Four sensor faults have been investigated as four typical and practical examples. All the four faults are considered with positive and negative bias of 10% on their correct values. The sensor faults can occur due to two basic reasons:

(i) wear and tear of the mechanical parts of the deflection meter or some changes in the value of resistance or capacitance used in the circuit due to aging (e.g. leakage of dielectric material or change in dielectric strength of the material), and

(ii) electrical fault such as short circuit or open circuit in the signal cable or on the circuit board.

The electrical faults are easy to detect because open circuit and short circuit faults will cause a full deflection or zero deflection in the meter respectively, and can be easily detected by a value check on the sensor output. On the contrary, the aging and mechanical faults cause incorrect meter readings and thus are difficult to detect. They can cause over-read or under-read the sensed variable and are known as bias faults.

The above mentioned faults can be planned and the data under faulty condition can be collected to train the neural classifier before the classifier is used on-line. These faults are therefore called 'known faults'. Any fault in the system for which the classifier is not trained is categorised as 'unknown fault'. The RBF classifier is only trained for the sensor faults in this study and thus any component fault happened in the system would be treated as unknown fault. For instance, an air leakage in the inlet manifold or the EGR valve clogged in closed position etc. There can be other unknown faults which may change some vital parameters like the air/fuel ratio.

A.) No-Fault: For no-fault situation, EGR is assumed to be 1/6 (16.67%) of the total air mass flow in the intake manifold. Practically EGR in a car can be as high as 20% of the total air mass flow. It is also assumed that all the sensors are working well and no component is malfunctioning. The no-fault data is collected for different throttle angle inputs ranging from 20 to 40 degree (the idling throttle angle for the engine is 15 degrees) for different operating points.

B.) Sensor Faults: Different multiplying factors (MFs) are used to generate fault data for eight different sensor faults by times the MF to a correct data (no-fault) as shown in Table 8.1.

Table 8.1: Faults and no-fault states and multiplying factors

State	Name of Fault	MF
0.	No-Fault (NF)	1
1.	Throttle angle position sensor 10% under reading	0.9
2.	Throttle angle position sensor 10% over reading	1.1
3.	Intake Manifold pressure sensor 10% under reading	0.9
4.	Intake Manifold pressure sensor 10% over reading	1.1
5.	Intake Manifold temperature sensor 10% under reading	0.9
6.	Intake Manifold temperature sensor 10% over reading	1.1
7.	Crankshaft Speed sensor 10% under reading	0.9
8.	Crankshaft Speed sensor 10% over reading	1.1

C.) Unknown Faults: Three different unknown faults are simulated in the model:

(i) Air leakage in the intake manifold

The air leakage in the intake manifold is simulated as explained before in section 4.2.2.

(ii) EGR valve clogged in closed position

EGR valve clogged in closed position (100% closed) would stop EGR flow and this is simulated as explained before in section 4.2.3.

(iii) Air/fuel ration fault

For a normal operation, λ is taken equal to "1" which corresponds to air/fuel ratio of 14.7 for gasoline engines and 14.5 for diesel engines. At λ equal to "1", it is stoiciometry or the point at which the most complete combustion takes place. λ gives a measure of air/fuel ratio which is independent of the type of fuel being used. λ being more than one implies excess air (Lean), while less than one implies excess fuel (Rich). We set λ=14.7+2.0=16.7 as fault of excess air while λ=14.7-2.0=12.7 as fault of excess fuel.

8.3 RBF TRAINING & TESTING PROCEDURES

Procedure for the training and testing of algorithm for unknown faults is explained step by step as follows:

a) *K*-means and *P*-nearest neighbours' algorithms are used to get the widths and the centres in the hidden layer of an RBF network utilising the training data set, the first set of the data collected from simulation.
b) Gaussian basis function is used as activation function and the activation function outputs are calculated.
c) Weights in the output layer are calculated by using batch least squares algorithm for minimum modelling error for the target values pre-defined for known faults.
d) All the three calculated matrices viz. centres, widths and weights are saved for the testing phase.
e) Another set of data with unknown faults is used for the testing purpose and is fed into the trained classifier. The network utilises the previously calculated values of centres, widths and weights. The network outputs are calculated.

f) The network outputs are compared with the ideal target values and 0.5 is considered as the decision boundary.

g) If none of the network outputs (including no-fault output) is more than 0.5 then it is considered as detection of an unknown fault.

h) Entire procedure from a) to g) is repeated with other sets of data with different unknown faults.

8.4 DATA FLOW FOR FDI

Fig. 8.1 shows the block diagram of the FDI system. First of all the MVEM is run and different data sets for all the four sensor outputs are collected as explained in Section 8.2. Then the collected data is normalised as explained in Section 4.3 of Chapter 4. All the eight faults are simulated on no-fault data as explained in section 8.2 using MFs listed in Table 8.1. The neural network is trained with the training data set. Then the different testing data sets are passed through the pre-trained neural classifier. The outputs of the classifier are filtered by a low-pass filter before decision stage. In the decision stage, one of the nine states (from NF to 8) will be high while others are low to show the occurrence of no-fault or any one of the faults. After the detection and isolation of fault, the fault accommodation would take place according to the fault detected. This is explained in Section 8.7 ahead.

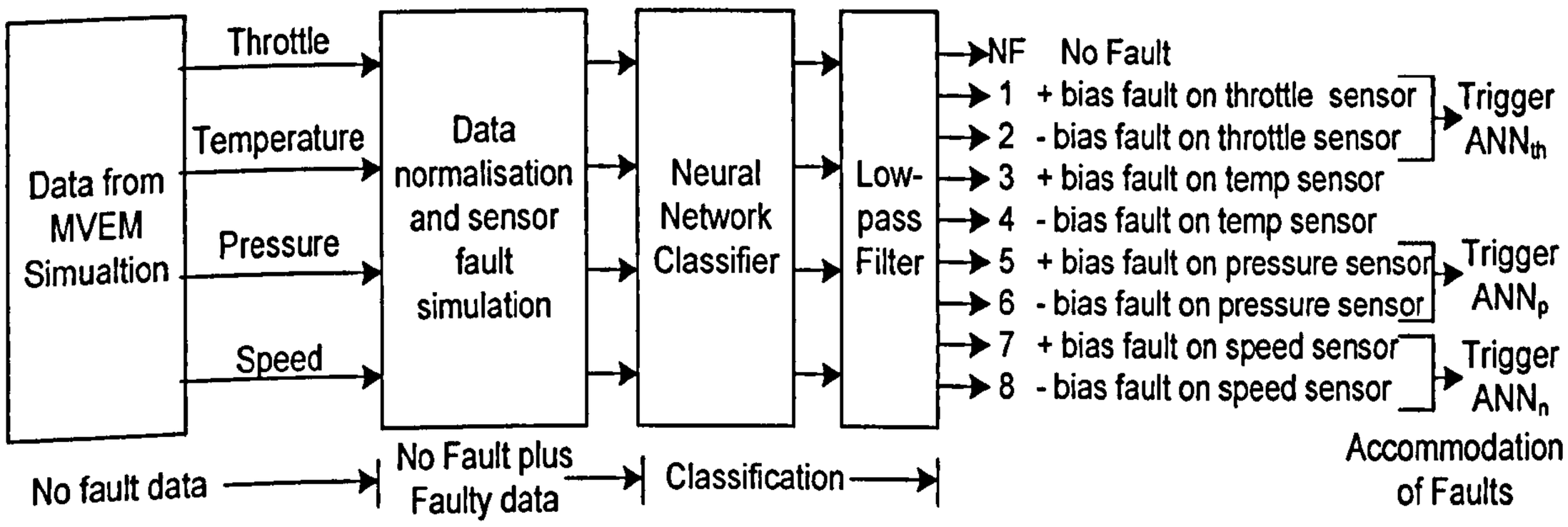


Fig. 8.1: Block diagram of FDI system

8.5 SIMULATION & FDI RESULTS

8.5.1 Data Collection

(A) RBF Training data collection

First of all the MVEM is run for different throttle angle inputs i.e. 20°, 22°, 24°, ..., 40° for no-fault condition and data for throttle angle position, inlet manifold pressure, temperature and crankshaft speed is collected. In the same way the data for each fault condition listed in Table 8.1 is collected for all the different throttle angle inputs. 12 data points are collected for each throttle input in 6 seconds at a sample time of 0.5 second. 120 data points are collected for each state for 10 different throttle angle inputs. There are 9 states in all and therefore the size of the training data set will be 1080 x 4 (12 x 10 x 9 = 1080). The target matrix size would be 1080 x 9 as explained in Section 8.6.2 ahead.

(*B*) Testing Data collection for known faults

Several testing data sets are separately collected for five different random throttle angle inputs to the MVEM. Same as before, 12 data points are collected for each throttle input in 6 seconds at a sample time of 0.5 seconds. Therefore, 60 data points are collected for each state for 5 different throttle angle inputs. The size of the testing data set is 540 x 4 (12 x 5 x 9 = 540).

(*C*) Testing Data collection for unknown faults

All the unknown faults are simulated one by one for 6 seconds. Each time when the simulation is run, 60 data points are collected at 0.5 seconds sample time for 5 different throttle angle inputs. Four different data sets are collected; one for each unknown fault i.e. 20% air leakage in the inlet manifold, EGR valve clogged in closed position, air/fuel ration thin (16.7) and air/fuel ratio thick (12.7). The size of each testing data set is 60 x 4.

8.5.2 Data Pre-processing

All the data sets collected are pre-processed. One data set is used as training data and the other as test data. The data is normalised by subtracting the steady state values and then scaled to the range of [0 1] as explained before in Section 4.3 of Chapter 4.

The target matrix x_o has ones in the first column up to the 120th row and all the other entries are zeros, the second column has ones from the 121st row to the 240th row and so on. The last column has ones from the 961st row to the 1080th row. This is shown in Fig. 8.2 as follows:

Row Numbers X_o

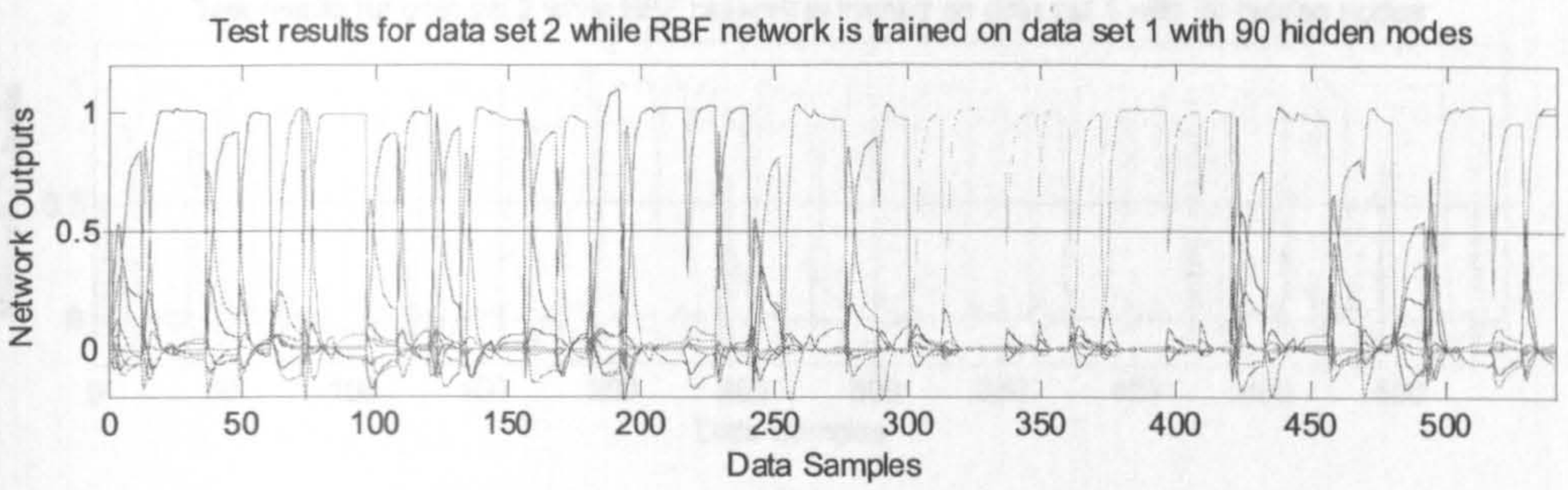
$$\begin{array}{c} 1 \sim 120 \\ 121 \sim 240 \\ \vdots \\ \\ \\ 961 \sim 1080 \end{array} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 1 & \cdots & & & \\ 0 & \vdots & \ddots & & & \\ 0 & & & \ddots & & \\ 0 & & & & 1 & \\ \vdots & & & & & \ddots \end{bmatrix}$$

Fig. 8.2: Target matrix X_o

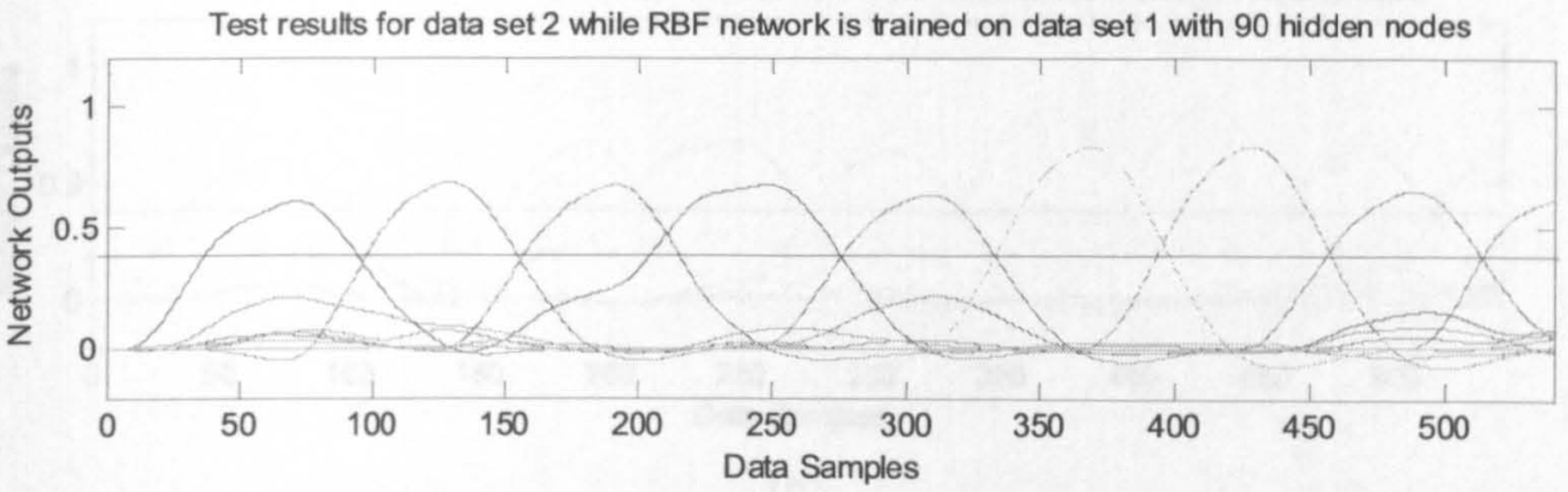
Thus, one of the 9 columns in the target matrix is associated to a fault condition. With the chosen input variables and the target, RBF network was trained with the training data set, where 90 centres were chosen using the *K*-means clustering algorithm. The widths were chosen using the *P*-nearest algorithm, and the weights were trained using the batch least squares (BLS) algorithm.

8.5.3 Detection of known faults

Training and testing procedure given in Section 8.4 is followed and the testing results for the second and third sets of data are shown in Fig. 8.3 and Fig. 8.4.

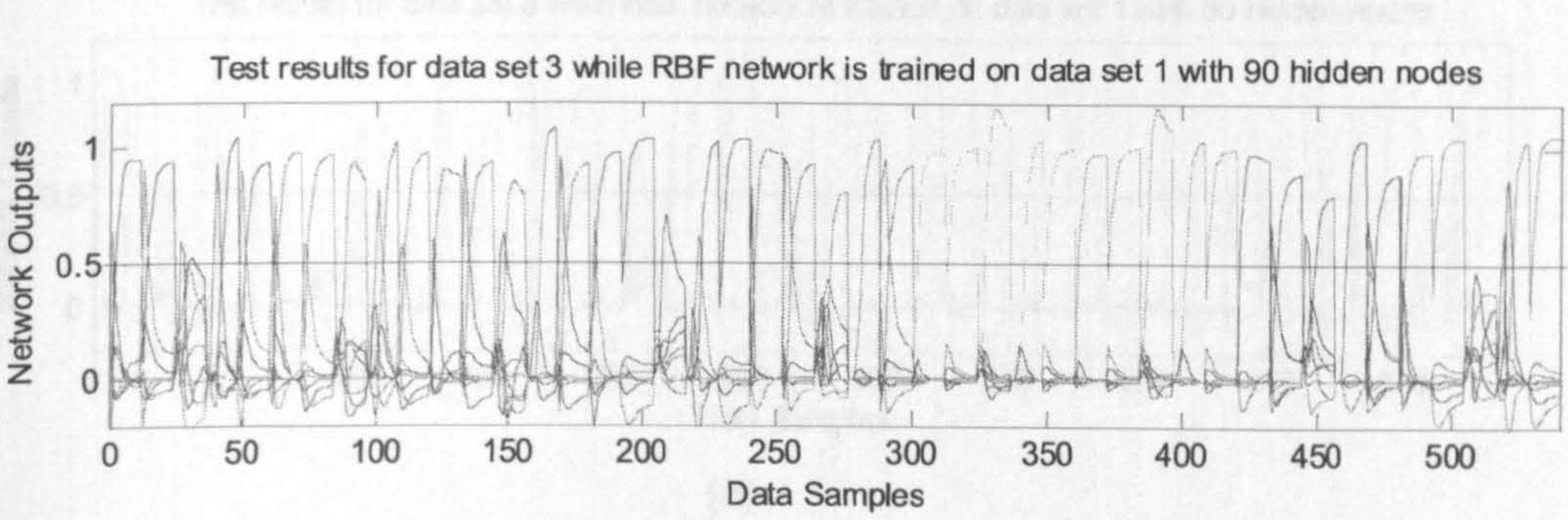


(a)

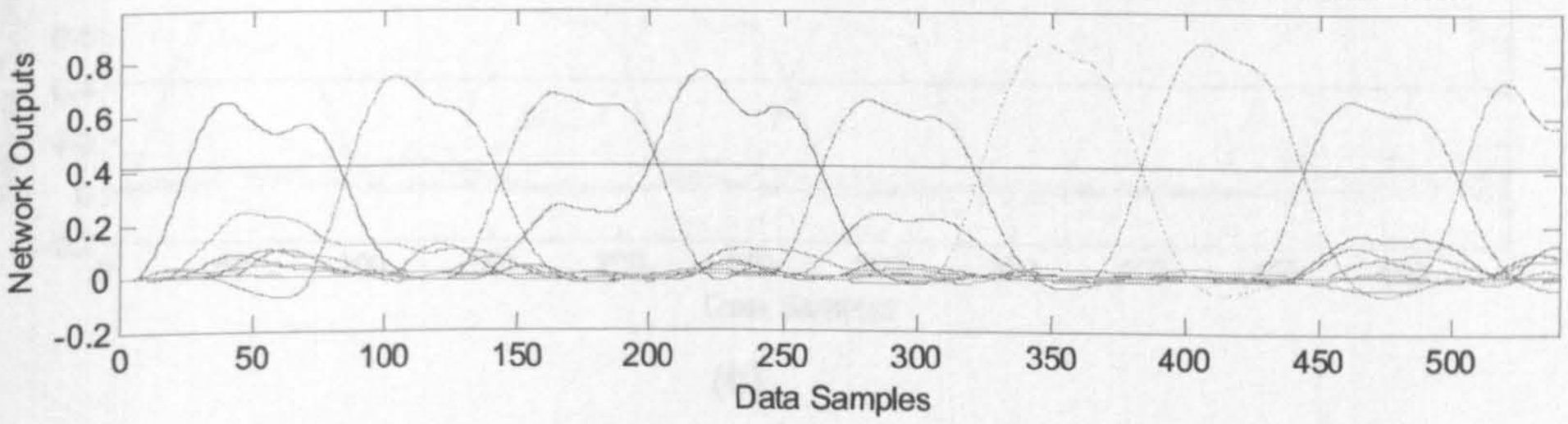


(b)

Fig. 8.3: Test results for data set 2 (a) before data filtration (b) after data filtration

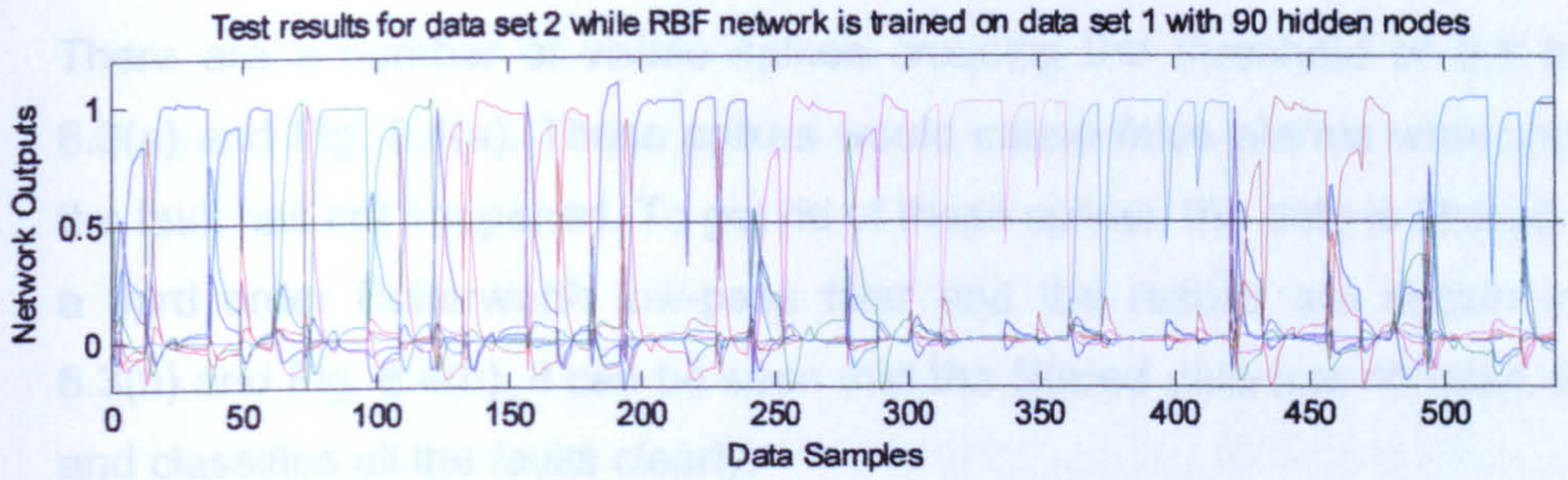


(a)

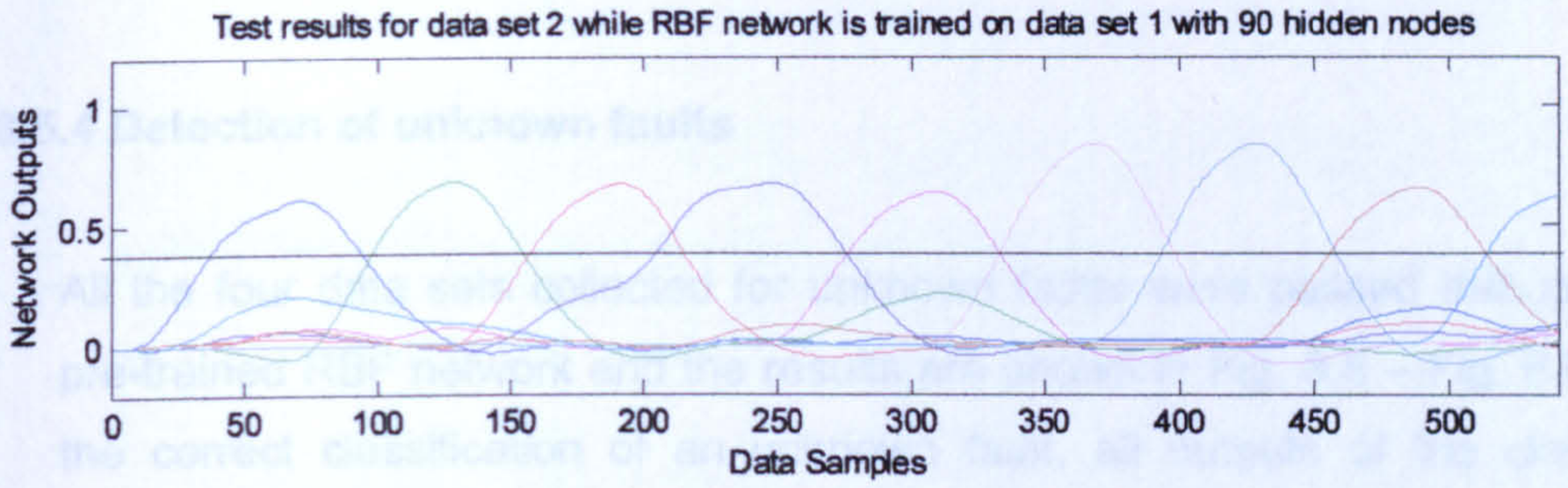


(b)

Fig. 8.4: Test results for data set 3 (a) before data filtration (b) after data filtration

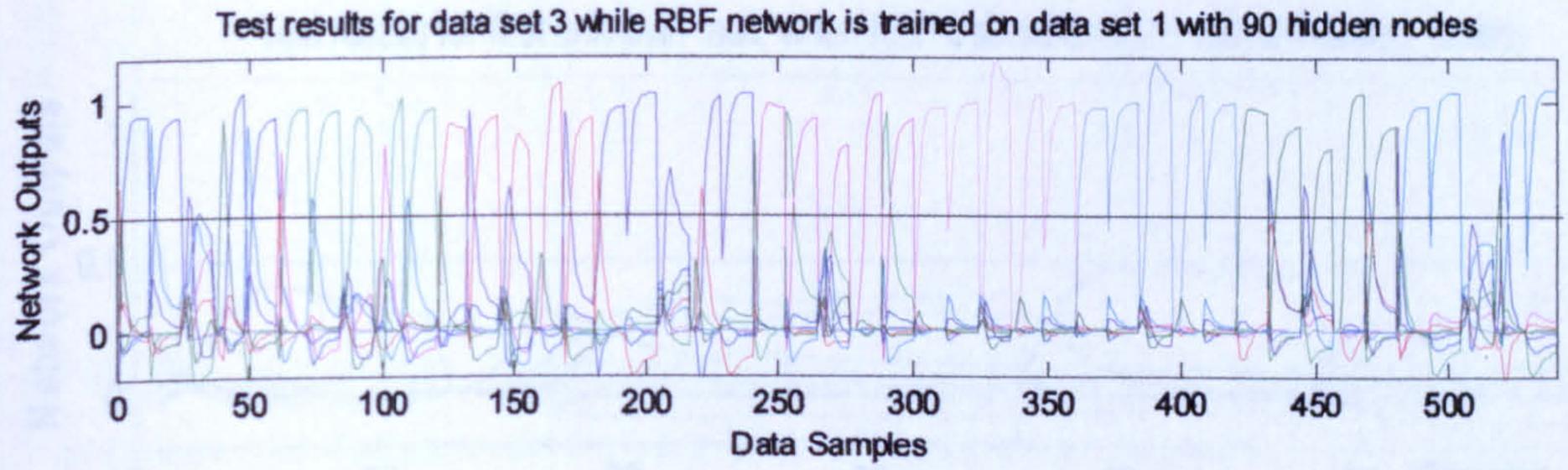


(a)

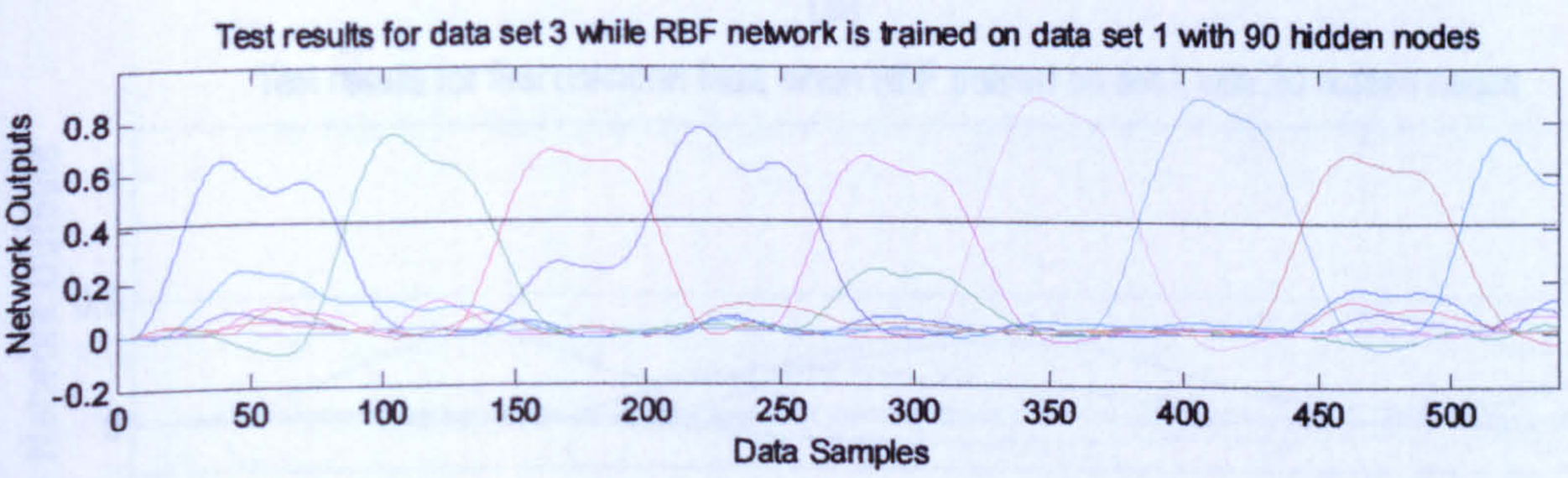


(b)

Fig. 8.3: Test results for data set 2 (a) before data filtration (b) after data filtration



(a)



(b)

Fig. 8.4: Test results for data set 3 (a) before data filtration (b) after data filtration

There are a number of visible spikes crossing the threshold of 0.5 in Fig. 8.3(a) and Fig. 8.4(a). These spikes would cause false alarms when actually the fault has not happened. To get rid of these spikes, the data is filtered using a third order Butterworth low-pass filter and the results are shown in Fig. 8.3(b) and Fig. 8.4(b). It can be seen that the filtered data has no false alarms and classifies all the faults clearly.

8.5.4 Detection of unknown faults

All the four data sets collected for unknown faults were passed through the pre-trained RBF network and the results are shown in Fig. 8.5 – Fig. 8.8. For the correct classification of an unknown fault, all outputs of the classifier should remain lower than the threshold of 0.5. The first, third and fourth faults are clearly classified as shown in Fig. 8.5, Fig. 8.7 and Fig. 8.8. But for the second fault (EGR valve clogged in closed position), some test result values exceed the threshold as shown in Fig. 8.6.

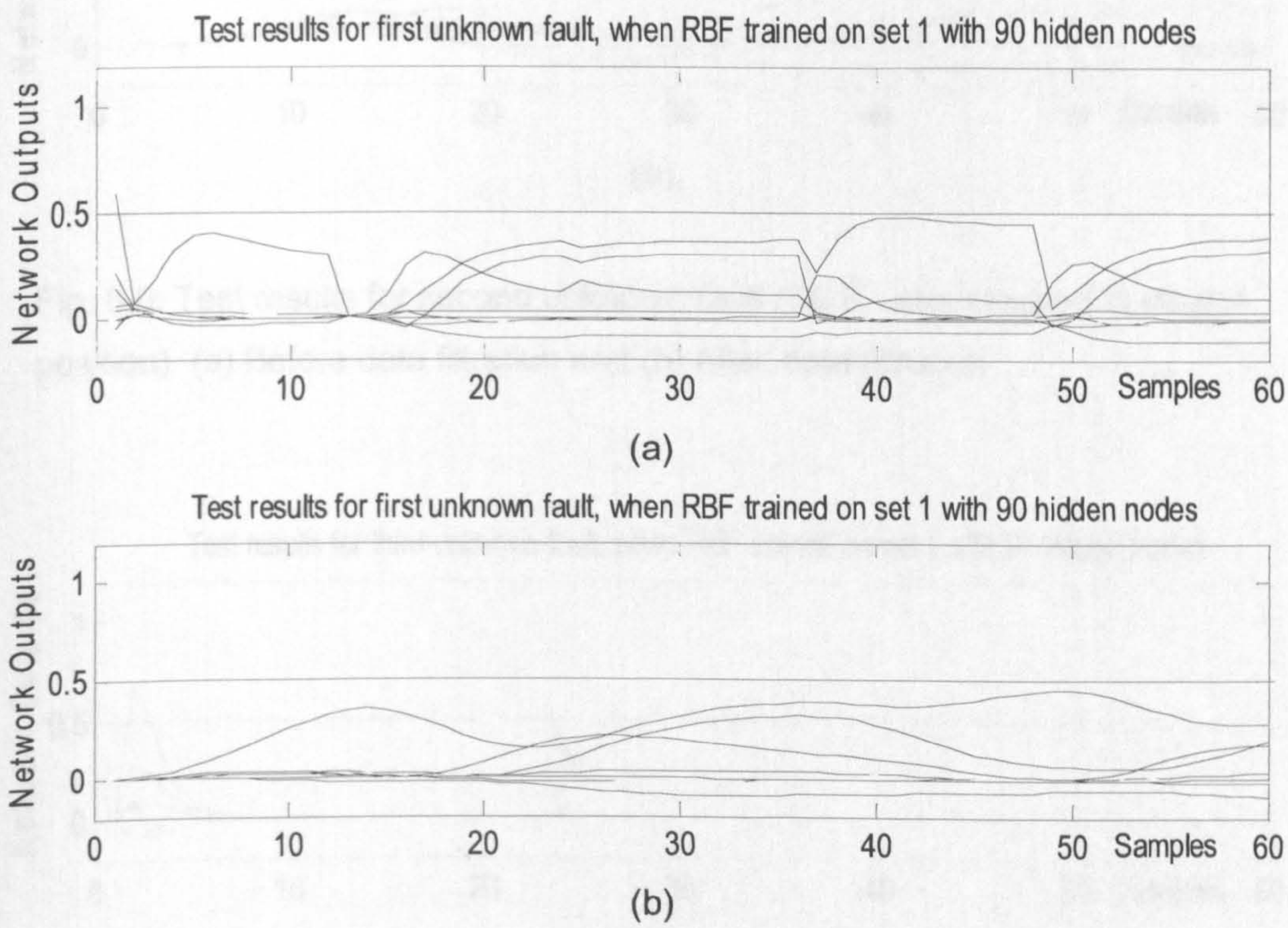


Fig. 8.5: Test results for first unknown fault (20% air leakage in inlet manifold)
(a) Before data filtration and (b) After data filtration

None of the known fault states and no- fault state should exceed the threshold, which did not happen in case of second fault and therefore it was not detected.

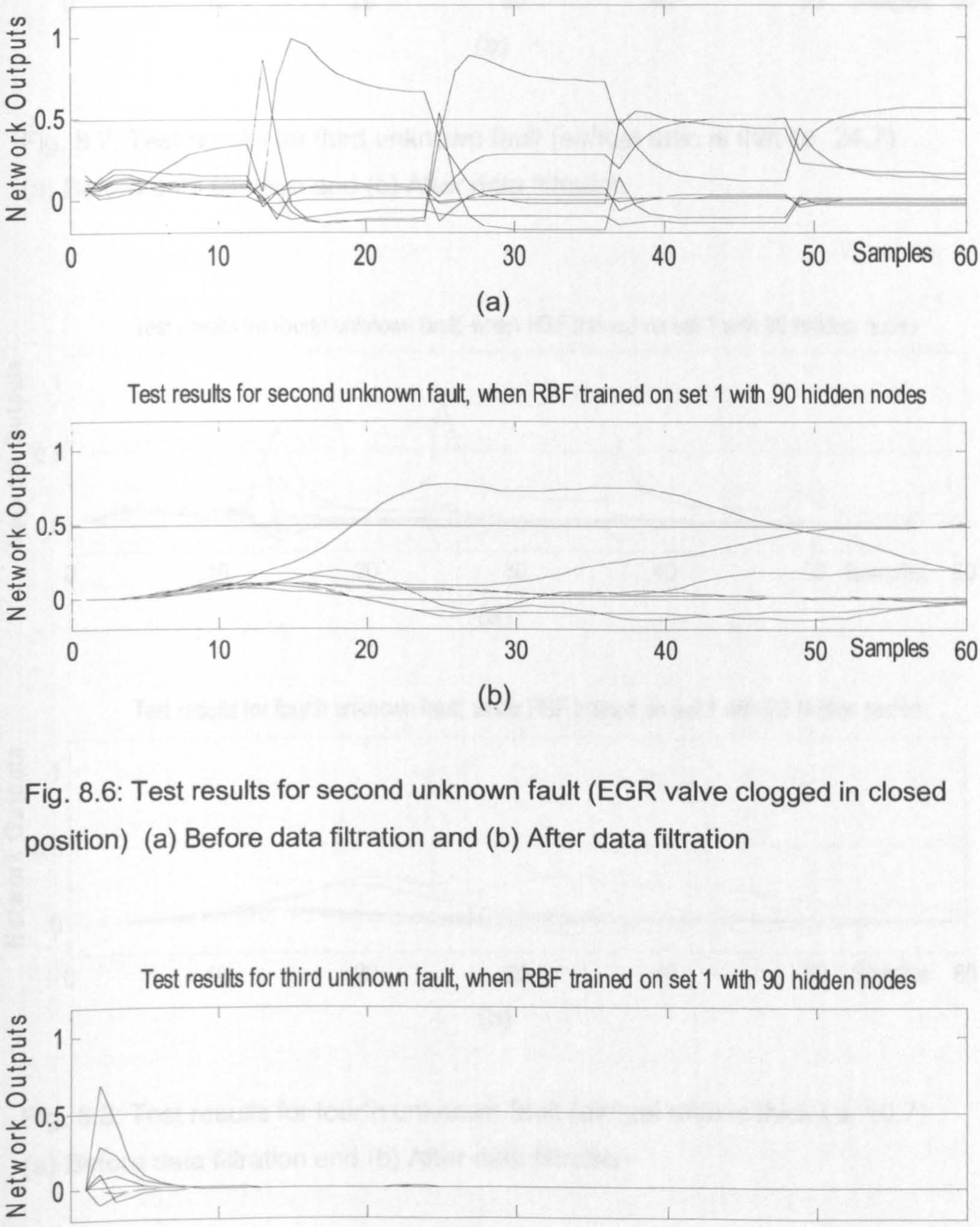
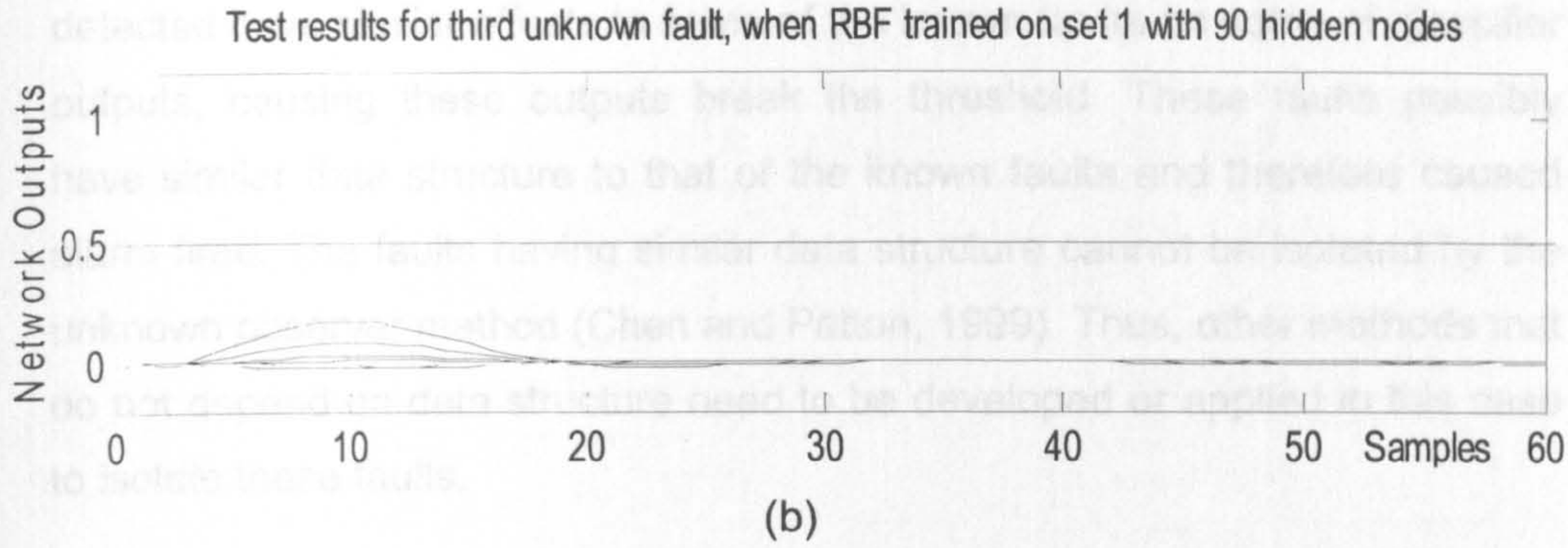
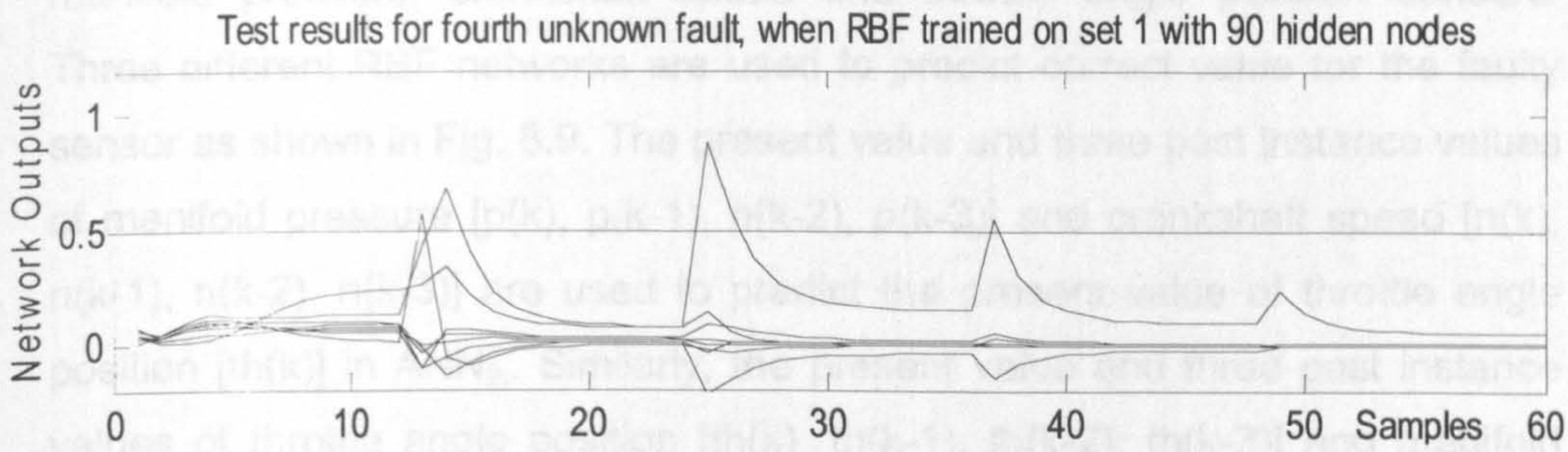


Fig. 8.6: Test results for second unknown fault (EGR valve clogged in closed position) (a) Before data filtration and (b) After data filtration

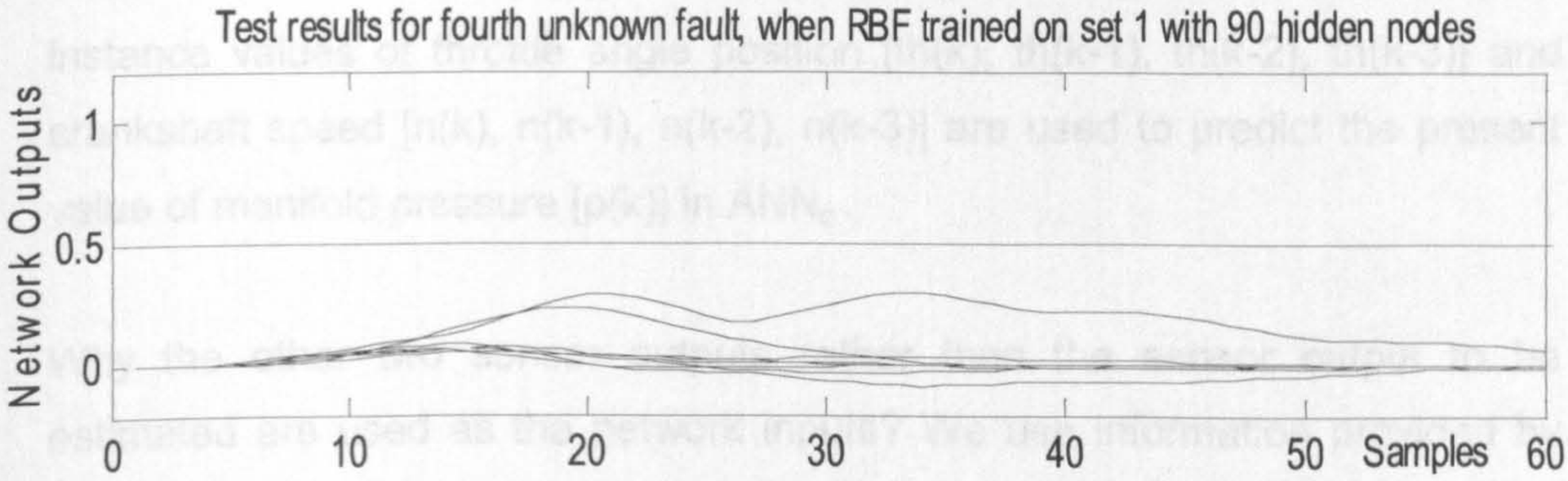


(b)

Fig. 8.7: Test results for third unknown fault (air/fuel ratio is thin i.e. 24.7)
(a) Before data filtration and (b) After data filtration



(a)



(b)

Fig. 8.8: Test results for fourth unknown fault (air/fuel ratio is thick i.e. 10.7)
(a) Before data filtration and (b) After data filtration

The test results demonstrated that some unknown faults can be detected while some others cannot be detected using the developed method. The reason for that can be analysed. The unknown faults that have not been

detected have similar effects to some of the known faults on some of classifier outputs, causing these outputs break the threshold. These faults possibly have similar data structure to that of the known faults and therefore caused alarm fired. The faults having similar data structure cannot be isolated by the unknown observer method (Chen and Patton, 1999). Thus, other methods that do not depend on data structure need to be developed or applied in this case to isolate these faults.

8.6 FAULT ACCOMMODATION

Three sensors are considered for fault accommodation in this section: manifold pressure, crankshaft speed and throttle angle position sensors. Three different RBF networks are used to predict correct value for the faulty sensor as shown in Fig. 8.9. The present value and three past instance values of manifold pressure [p(k), p(k-1), p(k-2), p(k-3)] and crankshaft speed [n(k), n(k-1), n(k-2), n(k-3)] are used to predict the present value of throttle angle position [th(k)] in ANN_{th}. Similarly, the present value and three past instance values of throttle angle position [th(k), th(k-1), th(k-2), th(k-3)] and manifold pressure [p(k), p(k-1), p(k-2), p(k-3)] are used to predict the present value of the crankshaft speed [n(k)] in ANN_n. The present value and three past instance values of throttle angle position [th(k), th(k-1), th(k-2), th(k-3)] and crankshaft speed [n(k), n(k-1), n(k-2), n(k-3)] are used to predict the present value of manifold pressure [p(k)] in ANN_p.

Why the other two sensor outputs rather than the sensor output to be estimated are used as the network inputs? We use information provided by other sensors rather than the information provided by it to reconstruct the concerned sensor output. This is because when the sensor has a fault the faulty output of the sensor will not affect the estimated value of the network because its output is simply not used as the network input. Different numbers of the hidden nodes have been tested for the RBF network predictor and 20 hidden nodes are found giving minimal prediction error for a given set of training data. Therefore the structure of each RBF network predictor is 8 x 20 x 1. The same RBF training algorithms are used as for the classifier before.

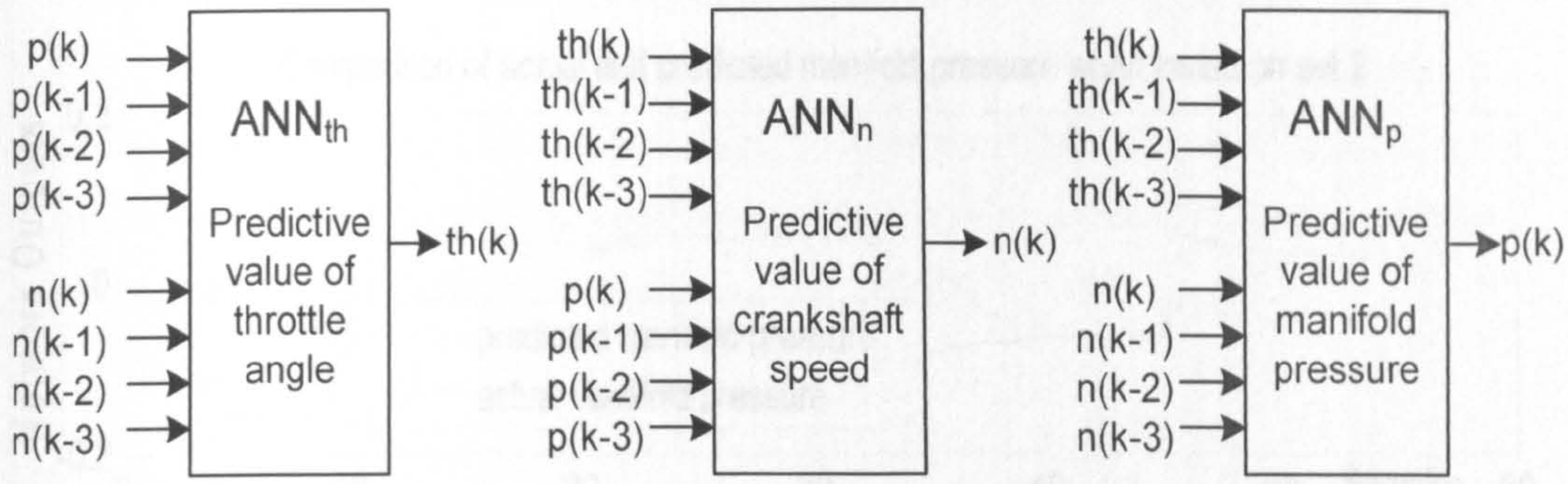
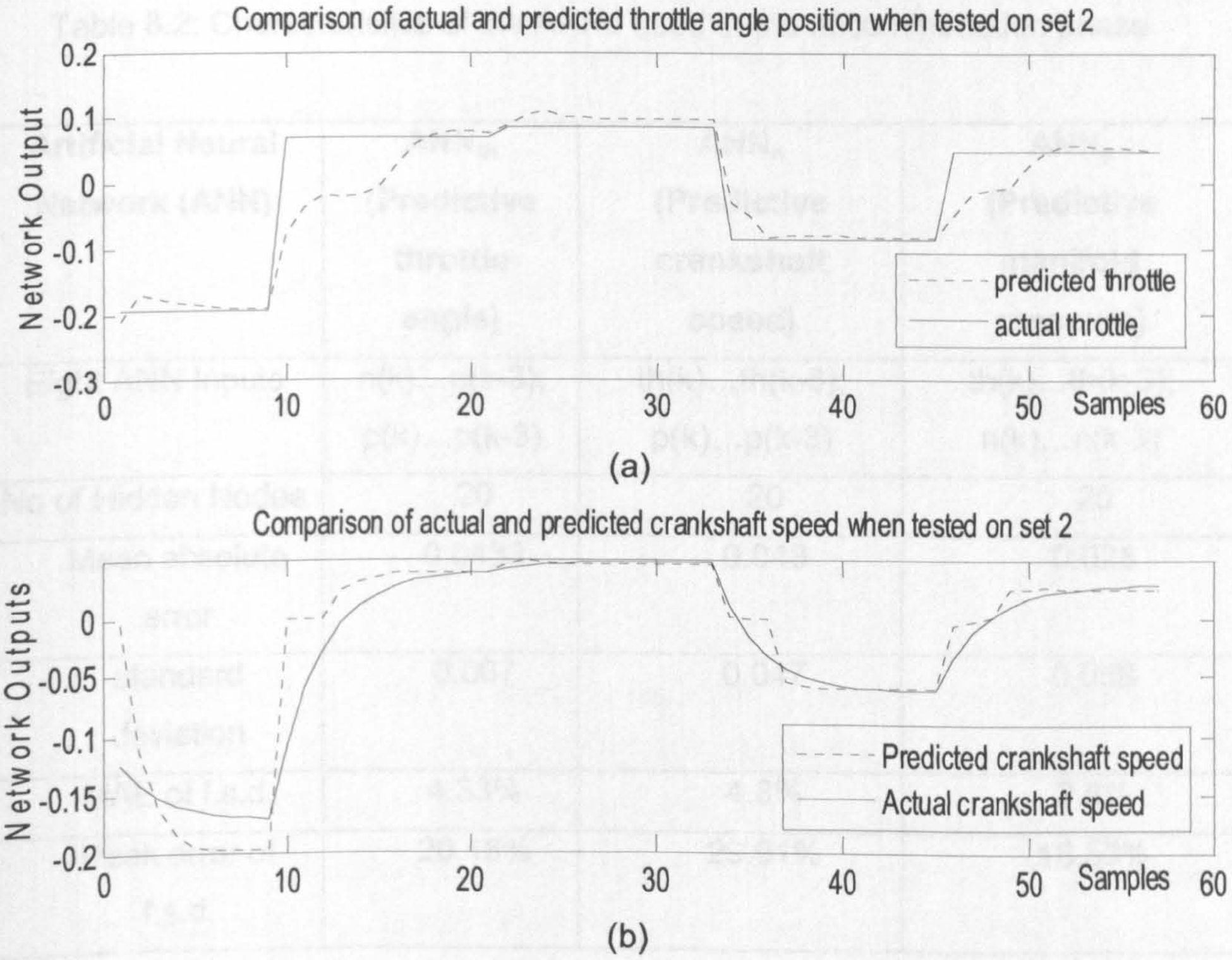
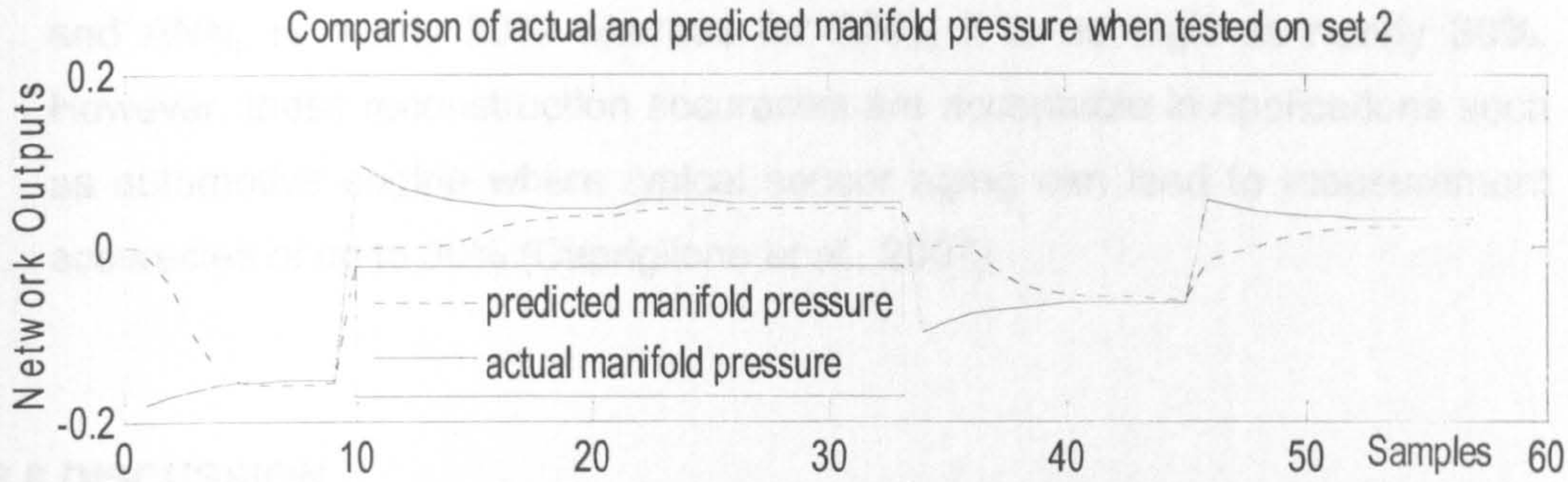


Fig. 8.9: Structures of the three RBF predictors for throttle angle, crankshaft speed and inlet manifold pressure

8.7 ACCOMMODATION PERFORMANCE EVALUATION

The reconstruction capabilities of the three ANNs are investigated for the purpose of accommodation system performance evaluation. With a set of test data that is different from the training data set, the RBF predictor performance is tested and the results are displayed in Fig. 8.10.





(c)

Fig. 8.10: Comparison of predicted and actual values of throttle angle, crankshaft speed and manifold pressure in (a), (b) and (c) respectively

From the data in Fig. 8.10 the average reconstruction error for the three predictors are calculated. It is found that for ANN_{th} and ANN_n are below 5% whereas for ANN_p it is below 3% as shown in Table 2.

Table 8.2: Characteristics of the ANNs used in the accommodation phase

Artificial Neural Network (ANN)	**ANN_{th} (Predictive throttle angle)**	**ANN_n (Predictive crankshaft speed)**	**ANN_p (Predictive manifold pressure)**
Eight ANN Inputs	n(k)…n(k-3); p(k)…p(k-3)	th(k)…th(k-3); p(k)…p(k-3)	th(k)…th(k-3); n(k)…n(k-3)
No of Hidden Nodes	20	20	20
Mean absolute error	0.0433	0.048	0.028
standard deviation	0.067	0.047	0.058
MAE of f.s.d.	4.33%	4.8%	2.8%
Peak error of f.s.d.	20.15%	29.91%	18.53%

Where f.s.d. is the full scale deflection. In all the tests, a peak error for ANN_{th}, and ANN_p is nearly 20% whereas for ANN_n it is as high as nearly 30%. However, these reconstruction accuracies are acceptable in applications such as automotive engine where typical sensor aging can lead to measurement accuracies of up to 30% (Capriglione *et al.*, 2007).

8.8 DISCUSSION

Sensor fault detection, isolation (FDI) and accommodation for automotive engines are investigated in this chapter. Different faults cause different structures in measurement data and therefore FDI is achieved by pattern classification approach while fault accommodation is achieved with prediction, where radial basis function (RBF) neural networks are used as classifiers and predictors. Three sensor outputs are on-line estimated from information excluding their own past outputs, and are used to replace the faulty sensor outputs temporarily till the sensor is repaired, to achieve fault accommodation. The key technique here is not using the measurement of the sensor to be predicted as the input of the predictor. In this way, the faulty sensor output will not affect the prediction, so that the predicted value can be used to replace the faulty sensor output. The sensor faults of $\pm 10\%$ amplitude were successfully detected, isolated and accommodated.

Moreover, several unknown faults are evaluated with the developed method and found that they are detected with this method. Ten percent bias fault is simulated from an engine simulation benchmark, the mean value engine model (MVEM), and the developed method is evaluated. Three unknown faults were tested and two of them can be detected with the developed method. Satisfactory results are achieved in simulations but improvement for better efficiency of the method is required.

The sensor fault accommodation with an average prediction error of less than 5% achieved in this study is an excellent result because typical sensor fault due to aging can lead to measurement error of up to 30% (Capriglione *et al.*, 2007). Current production cars in Europe are equipped with a preliminary EOBD that uses the default values in the look-up tables to replace the faulty sensor output. This is not as accurate as fault accommodation error of less

than 5% achieved in this research by predictive neural networks. The traditional method of using look-up tables also has a disadvantage of running the engine in uneconomical mode because the default values are approximate values and this might also increase tail pipe emissions. For the peak weather conditions like heavy snow, the default values from the look-up table may lead to low fuel economy.

The work of sensor fault accommodation for automotive engine using predictive neural networks in such a way is an original work of the author.

CHAPTER 9

CONCLUSIONS AND FURTHER WORK

9.1 CONCLUSIONS

All the aims and objectives listed in the first chapter of the thesis have been achieved and are briefly concluded as below:

A comprehensive literature survey on different FDI methods and engine FDI methods was conducted which included traditional limit alarm system, statistical process control method, fuzzy logic method, parameter estimation, observer based methods, model based methods, structured hypothesis, principle component analysis, wavelet networks, analytical redundancy, hybrid model-based technique, residual generation using parity equations, physical redundancy and other untraditional FDI methods like probabilistic approach and IASFR. Neural network based methods were also reviewed for automotive engine FDI e.g. Kohonen network, classical RBF classifier, support vector machine (SVM). It was concluded from the review that though neural networks have been used for modelling and classification of engine data for their condition monitoring and FDI, the disturbance effects are seldom considered and addressed. It was also concluded that little attention has been paid to the problem of diagnosing faults with different intensities. Moreover, the field of on-board engine fault diagnosis using artificial neural networks is still not fully explored. Application of ANN for automotive FDI has a lot of potential for present and future research.

A general introduction of diesel and petrol (gasoline) engine's working principle is briefly given in the beginning of chapter 3. The mean value engine model developed by Hendricks is used for simulations during the entire research period after minor modification. The basic MVEM equations and important Simulink models and sub-models are briefly reviewed, e.g. fuel mass flow dynamics, intake manifold filling dynamics, throttle plate angle dynamics, crankshaft speed dynamics, etc.

A non-adaptive RBF classifier is investigated for engine fault classification and the training algorithms are reviewed and derived. The *K*-means algorithm is used to choose the centres, *P*-nearest algorithm for widths and batch least squares algorithm for calculating weights in the output layer for RBF network. Sixteen different fault states were simulated along with no fault state. The non-adaptive network was able to classify all the faults for constant speed run and variable speed run from fixed initial speed, but failed to classify the faults for the most general case of variable engine speed from different initial speeds. But on the other hand, the non-adaptive classifier was able to classify all the faults in the reduced set of eight faults for the most general case. At this stage, the developed classifier is able to classify faults as small as 10% air leakage in manifold, EGR valve clogged in 50% closed position, 20% under reading and over reading bias in manifold pressure and temperature sensors. This classifier is not able to handle larger fault-set e.g. fault set with 16 faults. Moreover this method uses an off-line algorithm and therefore is not able to do FDI against disturbances, parameter change and model uncertainty.

A new adaptive RBF classification method is developed which can handle larger sets of faults and is also suitable for implementation in an on-board FDI system of an automobile. The developed algorithm can train and test classifier in an on-line condition. The widths of the RBF classifier are adapted using gradient descent method and the weights are adapted using recursive least squares algorithm whereas the centres remain fixed, as they are chosen to be distributed in the whole operating range. The classification results of adaptive classifier are compared with the results of non-adaptive classifier for a large set of 16 faults. It is found that adaptive classifier performs satisfactorily and is able to classify all the faults correctly unlike non-adaptive classifier. The performance of the non-adaptive classifier improves when the

number of hidden nodes for the RBF is increased from 60 to 150 but is still not able to classify all the faults correctly. The adaptive method is novel and it is believed to be contribution to knowledge in this field.

The robustness of adaptive neural network classifier is thoroughly assessed for the closed-loop system with crankshaft speed feedback control for a wide range of operational modes, including robustness against fixed and sinusoidal throttle angle inputs, change in load, change in an engine parameter, and all changes occurring simultaneously. The evaluations are performed on mean value engine model. The simulation results confirm that the proposed method is robust against uncertainty, disturbances, and environment change. The robustness assessment is also carried out for non-adaptive system and is compared with robustness assessment results of adaptive system. The adaptive system out performs the non-adaptive system. The nobility of the work consists in the successful demonstration of robustness of the developed adaptive FDI algorithm.

The real data assessment of adaptive neural network classifier is also carried out for five different sensor bias faults. A Volkswagen petrol car engine test bed is used for real data collection. The real data is collected for a wide range of operation of the engine. The load on the engine is changed from zero to maximum allowable load (86 N-m) and the crankshaft speed is also changed from 700 to 4000 rpm with percentage throttle angle changed up to 80%. The developed adaptive classifier is able to classify sensor bias faults as small as 2% without false alarms or misclassifications. Dynamic data with all inputs having three past values are used as input to the neural network for training for all the five sensor variables along with the current values. Such trained network makes a little improvement. This is because the adaptive FDI system itself can cope with fault classification in dynamic system due to adaptation.

Unknown fault diagnosis is carried out using novelty detection (Bishop, 1994) technique. Three different unknown faults are simulated in MVEM and tested on the non-adaptive RBF network for four sensor bias faults of 10%. The classifier was able to clearly classify two unknown faults but failed for the third fault. It is concluded that all unknown faults cannot be diagnosed by this method. The fault accommodation is also achieved for three sensor bias faults of 10% using predictive neural networks. The predictions are made for

the faulty sensor on the basis of healthy sensor behaviour. The accommodation is achieved for three sensors with an average prediction error below 5%.

9.2 FURTHER WORK

9.2.1 Introduction

A new adaptive FDI system has been developed and thoroughly checked for robustness and also assessed on real engine data in this research. The adaptive system has shown good classification capabilities and sensor faults as small as 2% can be detected and isolated without any misclassifications or false alarms. Further research work is required to bridge the gap between the academic research and its industrial application in real life. The developed FDI scheme has not been investigated for on-board application in an automobile ECU. In order to accomplish this, further research aims can be listed as follows:

i. To run adaptive FDI algorithm in real time with hardware-in-loop simulation (HILS) and check for performance and reliability.
ii. To develop algorithm to handle multiple faults happening simultaneously.
iii. To develop unknown fault detection and isolation system.
iv. To develop sensor fault accommodation system.
v. To implement the complete adaptive FDI and accommodation system in to the vehicle ECU.

For the fulfilment of above listed aims, additional hardware and software are requited to be installed in the engine test bed. The fifth objective cannot be achieved in the workshop but can only be achieved by the vehicle manufacturer. All the aims listed above have only two objectives.

I. To increase engine efficiency and fuel economy and
II. To decrease emission up to or below the level required by EOBD regulations

The following additional hardware and software from dSPACE would be required to complete the experimental setup. The additional hardware and software required are shown in dashed lines in Fig. 9.1.

a) List of hardware required

1. High performance PC with dedicated graphics card and 2 GB system RAM
2. DS1104 R&D Controller board
3. Combined Connector/LED Panel [CLP1104]

b) List of software required

1. Control Desk Standard – Developer Version [CS_D]
2. Real Time Interface [RTI]
3. Microtec C Compiler [CCPPPC]

Due to fault in sensor if the deviations of sensor's output occur and cause the vehicles emissions to exceed maximum values set by the EOBD/OBD-II regulations, then the EOBD/OBD-II system takes a predetermined course of action. The diagnostic functions enable sensor faults to be recorded and fault codes stored in ECU memory. This will assist the technician for the repair work. The on-board diagnostic system replaces the faulty sensor readings with default values which may protect sensitive components. For example, if an over-fuelling or misfire condition is detected, corrective actions can be implemented to protect the catalytic converter from overheating and permanent damage.

Here arises the requirement of auto correction for the sensor faults which is known as fault accommodation. The driver should immediately take the vehicle to the dealer for repair when the malfunction indicator lamp (MIL) lights, but this does not happen in real life. There is always a delay period between the fault indication and the actual repair. The delay in repair would unnecessarily cause environmental pollution. Fault accommodation would result in reduced emissions and increased vehicle reliability. It will also reduce the costs of repair and in some cases, where fault accommodation system

takes the emissions back to the permissible limits; the repair may not be required at all.

9.2.2 Methodology Overview

There are a number of fault diagnosis systems in practice but major car firms are now looking at neural networks to solve the demanding engine control and diagnostic requirements (Evans-Pughe, 2006). For instance Ford has introduced the Econoline van, which uses a neural net to detect misfire in its V10 engine. Applications of artificial neural networks (ANNs) to engine modelling and control have previously been presented by many authors (Tan and Saif, 2000; Kimmich *et al.*, 2005; Manzie *et al.*, 2001; Jakubek and Strasser, 2002). The application of a data-driven monitoring technique to accurately diagnose air leakage in the inlet manifold plenum chamber of an automotive engine with a diameter size as small as 2 mm can be found in (Antory, 2007). A hardware-in-the-loop simulation (HILS) system was developed and performance of a commercial Electronic Stability Program (ESP) Electronic Control Unit (ECU) was evaluated for a virtual vehicle under various driving conditions (Lee *et al.*, 2007). This HILS system can be used in various applications such as benchmarking, comparison of commercial ECUs, and detection of fault and malfunction of ESP ECUs.

Sensor fault accommodation for an automotive engine using artificial neural networks is quite a new and developing field of research. Fault accommodation of inlet manifold pressure, crankshaft speed and throttle angle position sensors using neural networks was investigated by Capriglione *et al.*, (2007) but the mean absolute fault accommodation errors were high and the peak errors were very high. The same sensors were also investigated in chapter 8 using different algorithms of neural networks and nearly similar results were achieved. Further, sensor FDI for automotive engine with real data evaluation was investigated by Sangha *et al.*, (2007) using adaptive neural networks and classification results for as small as 2% sensor faults were achieved.

Using dSPACE hardware and software, the fault classification and accommodation algorithms in Matlab/Simulink code can be run online with the ECU of automobile in real-time. A dSPACE prototyping system is programmed automatically from a block diagram. It is based on off-the-shelf

hardware and software components from which own personal dSPACE system can be configured for immediate use in conjunction with a PC.

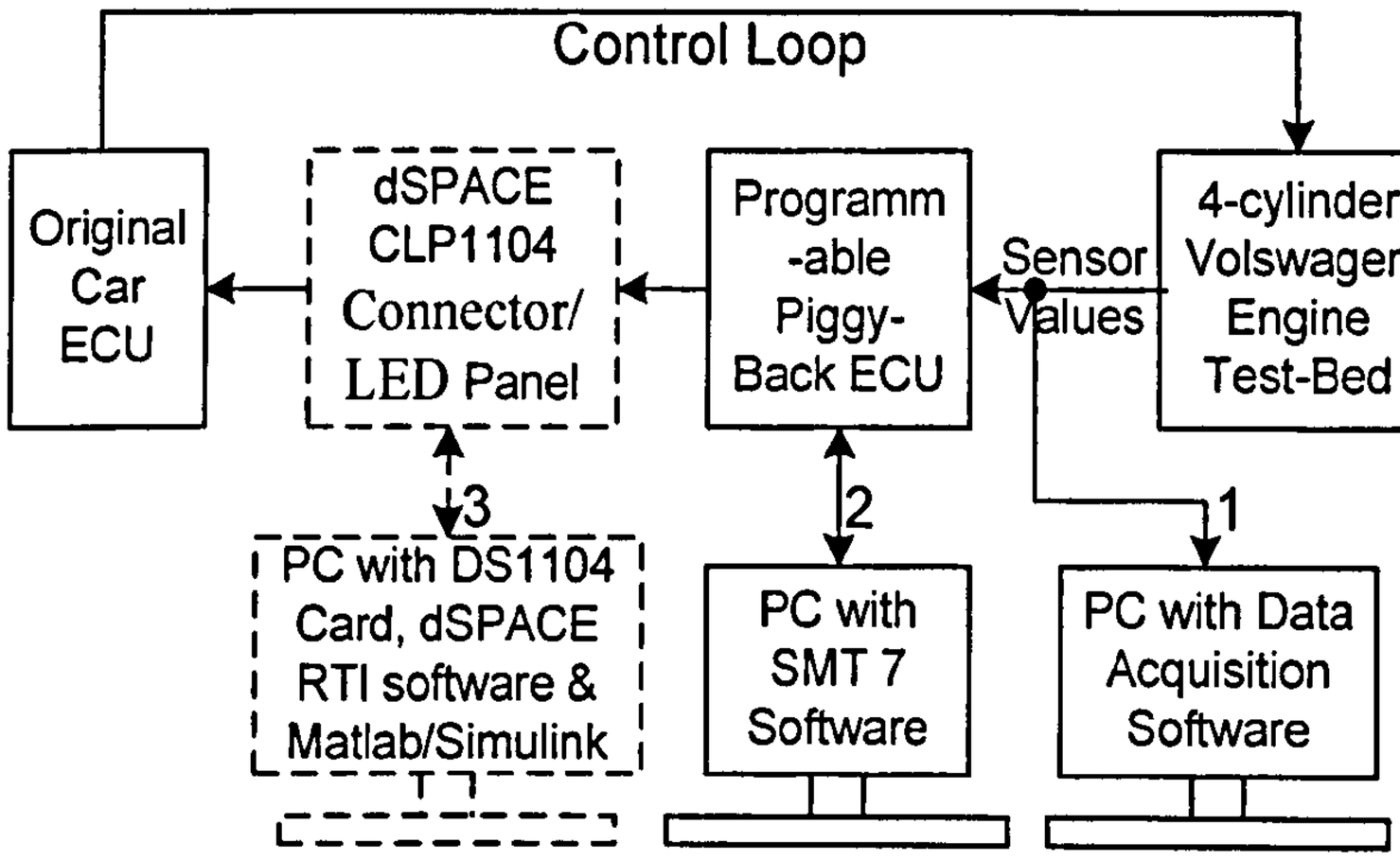


Fig. 9.1: Block diagram of experimental setup

9.2.3 Basic approach

Fig. 9.1 shows the block diagram of the experimental setup. The firm lines in the block diagram show the hardware and software already available and the dash lines show the additional hardware and software required to complete the experimental setup.

The right most blocks show the engine test bed and the PC with powerful data acquisition software (Daq View/ Lab View) and hardware. All the sensor values are read into piggy-back ECU and can be modified by the ECU tuning software by Digital Data Systems (SMT 7) run on the second PC. Further the sensor values are read into the original car engine ECU through dSPACE connector/LED panel (CLP1104). At this stage, different FDI algorithms can be run in Matlab/Simulink in real-time through the third PC and the control signals can be sent to the engine through the original car ECU.

The DS1104 R&D Controller board (hardware) from dSPACE upgrades the third PC into a powerful development system for rapid control prototyping (RCP). The DS1104 is specifically designed for the development of high

speed multivariable digital controllers and real-time simulations. It is complete real-time control system based on a 603 PowerPC floating point processor running at 250 MHz. For advanced I/O purposes, the board includes a slave-DSP subsystem based on the TMS320F240 DSP microcontroller. The real-time interface (RTI) software provides Simulink blocks for graphical configuration of A/D, D/A, digital I/O lines and incremental encoder interface etc. Real –time workshop generates the model code while RTI provides blocks that implement the I/O capabilities of dSPACE systems in Simulink models. The real-time model is compiled, downloaded, and started automatically on the real-time hardware. Different channels of the same I/O board can be used with different sample rates, and even in different subsystems. DS1104 R&D board has comprehensive functionalities and can turn into a hardware-in-loop (HIL) control system with RTI software.

Usually the piggy back ECU is connected between existing ECU and the engine sensor inputs and outputs with an objective to adjust the sensor readings such as air temperature, engine speed or crank position and effectively lie to the standard ECU forcing more aggressive ignition timing. But here the piggy back ECU would be used to manipulate the sensor values and for introducing sensor bias faults. Initially all the piggy back ECU maps would be set to defaults so that it does not affect/change any sensor values. Later on, the maps would be modified to achieve phantom sensor bias faults as required. These phantom sensor faults will be diagnosed by fault diagnosis system in Matlab in third PC and further the faults would be accommodated by the fault accommodation system. In order to verify the accommodation of sensor faults, the primary sensor values would be recorded and analysed before and after the fault accommodation phase.

REFERENCES

Aggarwal R. K., Xuan Q.Y., Dunn R. W., Johns A. T. and Bennett A. (1999), A novel fault classification technique for double-circuit lines based on a combined unsupervised/supervised neural network, *IEEE Transactions on Power Delivery*, vol. **14**(4), pp. 1250-1256.

Air resources environmental fact sheet (2005), On-Board Diagnostics a New Generation of Motor Vehicles, New Hampshire department of environmental services, Air Resources Division, ARD – 30.

Antory D. (2007), Application of a data-driven monitoring technique to diagnose air leaks in an automotive diesel engine: A case study, *Mechanical Systems and Signal Processing*, vol. **21**, pp. 795-808.

Antory D. (2005), Fault diagnosis application in an automotive diesel engine using auto-associative neural networks, *Proc. of Int. Conf. Computational Intelligence, Modelling and Control Automation*, Vienna, Austria, Nov 28-30, pp. 109-116.

Aquino C. F. (1981), Transient A/F control characteristics of the 5 litre central fuel injection engine, *Transactions of SAE*, SAE Technical Paper No. 810494.

Barigozzi A., Magni L., and Scattolini R. (2004), A probabilistic approach to diagnosis of industrial systems, *IEEE Transactions on Control Systems Technology*, vol. **12**(6), November, pp 950-955.

Bell A. G. (2004), Four-Stroke Performance Tuning, 2nd ed., Haynes Publishing, reprint 2004, London, ISBN 1-85960-435-8.

Benvenuti L., Benedetto M.D.D., Gennaro S. D. and Sangiovanni-Vincentelli A. (2003), Individual cylinder characteristic estimation for a spark injection engine, *Automatica*, vol. **39**, pp. 1157-1169.

Bishop, C.M., (1994). Novelty detection and neural-network validation, *IEE Proc. Vision Image and Signal Processing*, vol.**141**, pp. 217-222.

Biteus J. and Nyberg M., (2003), Residual generators for DAE systems utilizing minimal subsets of model equations, *Proceedings of IFAC Safe process '03*, Washington, USA.

Bosch R. (2004), Gasoline Engine Management: Systems and Components, 2nd ed., Robert Bosch GmbH, England, ISBN 1-86058-434-9.

Bosch R. (2005), Diesel Engine Management: Systems and Components. 4th ed., Robert Bosch GmbH, England, ISBN 0-470-02689-8.

Butler K. L., Ehsani M., and Kamath P. (1999), A Matlab-based modelling and simulation package for electric and hybrid electric vehicle design. *EEE Transactions on Vehicular Technology*, vol. **48**(6), pp. 1770-1778.

California Energy Commission, (2006), Super scientists, Rudolf Diesel On-line available at http://www.energyquest.ca.gov/scientists/diesel.html.

Capriglione D., Liguori C. and Pietrosanto A. (2007), Real – Time Implementation of IFDIA Scheme in Automotive Systems, *IEEE Trans. on Instrumentation and Measurement*, vol. **56**(3), pp. 824-830.

Capriglione D., Liguori C., Pianese C. and Pietrosanto A. (2004), Analytical redundancy for sensor fault isolation and accommodation in public transportation vehicles, *IEEE Trans. on Instrumentation and Measurement*, vol. **53**(4), pp. 993-999.

Capriglione D., Liguori C., Pianese C. and Pietrosanto A. (2003), On line sensor fault detection, isolation and accommodation in automotive engines, *IEEE Transactions on Instrumentation and Measurement*, vol. **52**(4), pp. 1182-1189.

Chen, J. and Patton, R.J. (1999). Robust model-based fault diagnosis for dynamic systems. Dordrecht: Kluwer Academic Press.

Chevalier, A., Muller, M. and Hendricks, E. (2000), On the validity of mean value engine models during transient operation, *Transactions of SAE*, SAE Technical Paper No. 2000-01-1261.

Chiang L. H., Russell, E. L. and Braatz R. D. (2001), Fault detection and diagnosis in industrial systems, Springer, ISBN:1852333278.

Conatser R., Wagner J., Ganta S. and Walker I. (2004), Diagnosis of automotive electronic throttle control systems, *Control Engineering Practice*, vol. **12**, pp. 23-30.

Crossley P. R. and Cook J. A. (1991), A nonlinear engine model for drive-train system development, International Conference on Control, 25-28 March, Edinburgh, UK, vol. **2**(332), pp. 921-925.

Crossman J. A., Guo H., Murphey Y. L. and Cardillo J. (2003), Automotive signal fault diagnostics – Part I: Signal fault analysis, signal segmentation, feature extraction and quasi-optimal feature selection, *IEEE Trans. on Vehicular Technology*, vol. **52** (4), pp. 1063-75.

Dalmi, I., Kovacs, L., Lorant, I., Terstyanszky, G., (1999), Application of supervised and unsupervised learning methods to fault diagnosis, Proc. of the 14th World Congress of IFAC, pt. 17, vol.**17**, pp. 91-96, ISBN 0 08 042754 5.

Dalmi, I., Kovacs, L., Lorant, I., Terstyanszky, G., (2001), Diagnosing priori unknown faults by radial basis function neural network, Proc. 4th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes 2000 (Safeprocess 2000), pt.2, vol. **2**, pp. 399-403, ISBN 0 08 043250 6.

Denton, T. (2004). Automobile electrical and electronic systems, *Elsevier Butterworth Heinemann Publication*, ISBN 0750662190, 8.

Ebersbach S. and Peng Z. (2008), Expert system development for vibration analysis in machine condition monitoring, *Expert Systems with Applications*, vol. 34(1), pp. 291 – 299.

Evans-Pughe C. (2006), Learning to drive [tightening emissions regulations], Engineering & Technology, Institution of Engineering and Technology, UK, vol. **1**(2), pp. 42-45.

Frank P. M., Ding S. X. and Marcu T. (2000), Model-based fault diagnosis in technical processes, *Transactions of Institute of Measurement and Control*, vol. **22(1)**, pp 57-101.

Frank, P. M. (1996), Analytical and Quantitative Model-Based Fault Diagnosis – A Survey and some new results. *European journal of Control*, vol. **2**, pp. 6-28.

Frank, P. M. (1994), Enhancement of robustness in observer-based fault detection, *International Journal of Control*, vol. **59**(4), pp. 955-981.

Frisk E. and Nyberg M., (2002), Residual generation for fault diagnosis of systems described by general linear differential-algebraic equations, 15th *IFAC World Congress*, July 21-26, Barcelona, Spain.

Gertler J. J. (1998), Survey of model-based failure detection and isolation in complex plants. *IEEE Control System Magazine*, vol. **8**(6), pp. 3-11.

Gertler J., Costin M., Fang X., Kowalczuk Z., Kunwer M., and Monajemy R., (1995), Model based diagnosis for automotive engines – algorithm development and testing on a production vehicle, *IEEE Transactions on Control Systems Technology*, vol. **3**(1), March, pp 61-69.

Gertler J., Costin M., Fang X., Hira R., Kowalczuk Z. and Luo Q., (1993), Model based on board fault detection and diagnosis for automotive engines, *Control Engineering Practice*, vol. **1**(1), pp 3-17.

Gertler J. J., Costin M., Fang X., Hira R., Kowalczuk Z. and Luo Q., (1991), Model-based on-board fault detection and diagnosis for automotive engines. IFAC Fault Detection, Supervision and Safety for Technical Processes, Baden-Baden, Germany, pp. 503-508.

Gertler, J. (1988), Survey of Model-Based Failure Detection and Isolation in Complex Plants. *IEEE Control Systems Magazine*, vol. **8**(6), pp. 3-11.

Gomm J. B., Weerasinghe M. and Williams D., (2000), Diagnosis of process faults with neural networks and principal component analysis. *Proc I MechE Part E,* vol. **214**, pp. 131-142.

Hendricks E., Engler D. and Fam M., (2000), A Generic Mean Value Engine Model for Spark Ignition Engines. Institute of Automation, Denmark Technical University. www.iau.dtu.dk/~eh/.

Hendricks E., Chevalier A., Jensen M., Sorenson S. C., Asik J. and Trumpy D., (1996), Modelling of the Intake Manifold Filling Dynamics, *SAE* Technical Paper No. 960037.

Hendricks E., Vesterholm T., Kaidantzis P., Rasmussen P. B. and Jensen M., (1993), Nonlinear Transient Fuel Film Compensation, *SAE* Technical Paper No. 930767.

Hendricks E., Sorenson S. C. (1990), Mean value modelling for spark ignition engines, *Transactions of the SAE*, SAE Technical Paper No. 900616.

Hissel D., Pera C. M., and Kauffmann M. J. (2004), Diagnosis of automotive fuel cell power generator, Journal of power sources, vol. **128**, pp. 239-246.

Isermann R. (2005), Model-based fault detection and diagnosis – status and applications, *Annual Reviews in Control*, vol. **29**, pp. 71-85.

Isermann R. and Schwarte A. (2004), Fault diagnosis of a turbocharged diesel engine with dynamic neural networks and parity methods, Proceedings of the 6th IASTED International Conference on Intelligent Systems and Control, Aug 23-25, Honolulu, HI, USA, pp. 283-288.

Isermann R. and P. Balle (1997), Trends in the application of Model-based Fault Detection and Diagnosis of Technical Processes. *Control Engineering Practice*, vol. **5**(5), pp. 709-719.

Isermann R. (1997), Supervision, Fault-Detection and Fault-Diagnosis Methods – An Introduction. *Control Engineering Practice*, vol. **5**(5), pp. 639-652.

Isermann R. (1984), Process fault detection based on modelling and estimation methods: A survey, *Automatica*, vol. **20**, pp. 387-404.

Jakubek S. and Strasser T. (2002), Fault diagnosis using neural networks with ellipsoidal basis functions, Proc. Of American Control Conference, May 8-10, Anchorage, AK, pp. 3846-3851.

Jamsa-Jounela S. L., Vermasvuori M., Enden P. and Haavisto S. (2003), A process monitoring system based on the Kohonen self-organizing maps, Control Engineering Practice, vol. **11**, pp. 83-92.

Kimmich F., Schwarte A. and Isermann R (2005). Fault detection for modern diesel engines using signal and process model-based methods, *Control Engineering Practice*, vol. **13**, pp. 189-203.

Kowalski C. T. and Orlowska-Kowalska T. (2003), Neural networks application for induction motor faults diagnosis, *Mathematics and Computers in Simulation*, vol. **63**, pp. 435–448.

Krishnaswami V., Chun-Luh G. and Rizzoni G. (1994), Diagnosis of exhaust emission control systems during the E.P.A. tailpipe inspection program, IFAC Fault Detection, Supervision and Safety for Technical Processes, Espoo, Finland, pp. 381-386.

Laukonen E. G. and Passino, K.M. (1995), Training fuzzy system to perform estimation and identification, *Engineering applications of artificial intelligence*, vol. 8(5), pp. 499-514.

Lee, S. J., Park, K., Hwang, T. H., Hwang, J. H., Jung, Y. C., Kim, Y. J. (2007), Development of hardware-in-the-loop simulation system as a test

bench for ESP unit, *International Journal of Automotive Technology*, vol. **8**(2), pp. 203-209.

Leonhardt S., Ludwig C., and Schwarz R. (1995), Real-time supervision for diesel engine injection, *Control Engineering Practice*, vol. **3**(7), pp. 1003-1010.

Luh G. C. and Rizzoni G. (1994), Identification od non-linear MIMO IC engine model during I/M240 driving cycle for on-board diagnosis, Proceedings of the American Control Conference, Baltimore, Maryland, pp. 1581-1584

Li, Y., Pont, M. J. and Jones, N. B. (2002). Improving the performance of radial basis function classifiers in condition monitoring and fault diagnosis applications where 'unknown' faults may occur, *Pattern Recognition Letter*, vol. **23**, pp. 569-577.

Ljung, L. (1999), System Identification---Theory for the User, 2nd edition, Prentice-Hall, pp. 361-369.

Lu Y. and Chen T.Q. (1998), A fuzzy diagnostic model and its application in automotive engineering diagnosis, *Applied Intelligence, the International Journal of Artificial Intelligence,* vol. **9**(3), pp. 231-243.

Lu Y., Chen T. Q. and Brennan H. (2000), A fuzzy system for automotive fault diagnosis: Fast rule generation and self-tuning, *IEEE transactions on vehicular technology*, vol. **49**(2), pp. 651-660.

Luo Jianhui, Tu F., Azam M., Pattipati K., Willett P., Qiao L. and Kawamoto M. (2003), Intelligent model-based diagnostics for vehicle health management, *Proceedings of SPIE*, vol. **5107**, Security and condition monitoring issue III, pp. 13-26.

Manzie C., Palaniswami M. and Watson H. (2001), Gaussian networks for fuel injection control, *ImechE, Proc Instn Mech Engrs,* vol. **215**, Part D, pp. 1053-1068.

Mcfarlane D. C. (1996), A robust approach to dynamic statistical process control, Proceedings of 35th IEEE Conference on Decision and Control, 11-13 Dec., Kobe, Japan, vol. **3**, pp. 3113-14 .

Melton K. D., English J. R. and Taylor G. D. (1997), A statistical process control approach to process diagnosis in discrete manufacturing

environments, International Journal of Quality Science, vol. 2(2), pp. 87-105.

Montgomery, D. C. (1996), Introduction to Statistical Quality Control, 3rd edition, John Wiley, New York, NY.

Moody, J. and Darken, C. (1989), Fast learning in networks of locally-tuned processing units, *Neural Computation*, vol. **1**(2), pp. 281-294.

Nyberg M. and Stutte T., (2004), Model based diagnosis of the air path of an automotive diesel engine, *Control Engineering Practice,* vol. **12**, pp. 513-525.

Nyberg M. and Krysander M., (2003), Combining AI, FDI, and statistical hypothesis-testing in a framework for diagnosis, *Proceedings of IFAC safe process' 03*, Washington, USA.

Nyberg M., (2001), A general framework for fault diagnosis based on statistical hypothesis testing, American association for artificial intelligence (www.aaai.org), pp. 1-8.

Nyberg M., (2000), Evaluation of test quantities for leakage diagnosis in the air path of an automotive engine, IFAC fault detection, supervision and safety for technical processes, Budapest, Hungary, pp. 143-148.

Nyberg M., (1999a), Automatic design of diagnosis systems with application to an automotive engine, *Control Engineering Practice*, vol. **7,** pp. 993-1005.

Nyberg M., (1999b), Model based diagnosis of both sensor-faults and leakage in the air-intake system of an SI engine, *SAE paper* 1999-01-0860.

Nyberg M. and Perkovic A., (1998), Model based diagnosis of leaks in the air intake system of an SI engine, *SAE Paper No. 980514*.

Nyberg M. and Nielsen L., (1997), Model based diagnosis for the air intake system of SI engine, journal of commercial vehicles, *SAE Transaction*, vol. **106**, pp. 9-20.

Official Journal of the European Communities (1998). L 350/1, Directive 98/69/EC of the European Parliament and of the Council of 13 October 1998 relating to measures to be taken against air pollution by emissions from motor vehicles and amending Council Directive 70/220/EEC.

Patton, R. J. (1994), Robust model-based fault diagnosis: the state of the art, Proceedings of IFAC Symposium on Fault Detection, Supervision and

Safety for Industrial Processes, 13-16 June, Espoo, Finland, vol. 1, pp. 1-24.

Patton R., Frank P., and Clark R. (1989), Fault *diagnosis in dynamic systems*, Englewood Cliffs, NJ, Prentice Hall.

Pisu P., Soliman A. and Rizzoni G. (2003), Vehicle chassis monitoring system, Control Engineering Practice, vol. **11**, pp. 345-354.

Principe J. C., Euliano N. R. and Lefebvre W. C., (2000), *Neural and Adaptive Systems: Fundamentals Through Simulations*, John Wiley & Sons, Inc.

Reineman, M. (2000), Effectiveness of OBD II Evaporative Emission Monitors – 30 Vehicle Study, U.S. Environmental Protection Agency Report, EPA420-R-00-018.

Rizzoni G. and Hampo R. (1989), Real time detection filters for on-board diagnosis of incipient failures. *SAE Paper*, No. 890763.

Rizzoni G. and Min P. S. (1991), Detection of sensors failures in automotive engines. *IEEE Transactions on Vehicular Technology*, vol. **40**(2), pp 487-500.

Rizzoni G., Azzoni P. M. and Minelli G. (1993), On-board diagnosis of emission control system malfuctions in electronically controlled spark ignition engines, Proceedings of the American Control Conference, pp. 1790-1795.

Rungasamy S., Antony F. and Ghosh S. (2002), Case Study: Critical success factors for SPC implementation in the UK small and medium enterprises: some key findings from a survey, The TQM Magazine, vol. 14(4), pp. 217 – 224.

Sangha M. S., Gomm J. B., Yu D. L., Page G. F. (2004a), An investigation of Neural Networks in fault diagnosis of an auto motive engine air path, *2nd ASIM Workshop*, Wismar, Germany – Modelling, control and diagnostics of combustion engine processes, September 16-17, pp. 23-32, ISBN 3-901608-27.

Sangha M. S. (2004b), An Investigation of Neural Networks in Fault Diagnosis of an Automotive Engine Air Path, October, MSc Thesis, Liverpool John Moores University, UK.

Sangha M. S. and Gomm J. B. (2005a), Steady-state fault diagnosis of an automotive engine, GERI Annual Research Symposium, June 22, Liverpool John Moores University, UK.

Sangha M S, Gomm J B, Yu D L (2005b). Dynamic FDI of an Automotive Engine Using Artificial Neural Networks, *16th IFAC World Congress*, July 4-8, Prague.

Sangha M. S., Yu D. L. and Gomm J. B. (2006), On-board monitoring and diagnosis for spark ignition engine air path via adaptive neural networks, *Proc. of IMechE Part D, Journal of Automobile Engineering*, vol. **220**, pp. 1641-1655.

Sangha M S, Yu D L, Gomm J B (2007). Adaptive FDI for Automotive Engine Air Path and Robustness Assessment, *International Journal of Automotive Technology*, v. **8** (5), pp. 637-650.

Schoelkopf B., Sung K. K., Burges C. J. C., Girosi F., Niyogi P., Poggio T. and Vapnik V. (1997), Comparing support vector machines with Gaussian kernels to radial basis function classifiers, *IEEE Trans. Neural Networks*, vol. **45**(11), pp. 2758-2765.

Shoji H., Toshiyuki A., and Shenq Z. (2002), Study of machine fault diagnosis system using neural networks, Proceedings of the International Joint Conference on Neural Networks (IJCNN '02), vol. 1, May 12 – 17, Honolulu, pp. 956 – 961.

Shayler J. P., Goodman M. and Ma T. (2000), The exploitation of neural networks in automotive engine management systems, *Engineering Applications of Artificial Intelligence*, vol. **13**, pp. 147-157.

Shiraishi H., Ipri S. L., and Cho D.-I. (1995), CMAC neural network controller for fuel injection systems, *IEEE Transactions on Control Systems Technology*, vol. **3**(1), pp. 32-38.

Tan Y., Saif M., (2000), Neural-networks-based nonlinear dynamic modelling for automotive engines, *Neurocomputing*, vol. **30**, pp. 129-142.

Tawel R., Aranki N., Puskorius G. V., Marko K. A., Feldkamp L. A., James J. V., Jesion, G., Feldkamp T. M. (1998), Custo VLSI ASIC for automotive applications with recurrent networks, Proceedings of the 1998 IEEE Internal Joint Conference on Neural Networks, May 4-9, Anchorage, AK, USA, pp. 598-602.

Tomaszewski P., Hakansson J., Grahn H. and Lundberg L. (2007), Statistical models vs. expert estimation for fault prediction in modified code – an industrial case study, *The Journal of Systems and Software*, vol. 80(8), pp. 1227 – 1238.

Toshiyuki A., Takashi K., Baojie X. and Shaji H. (2000), Fault diagnosis system for machines using neural networks, *JSME International Journal, Series C: Mechanical Systems, Machine Elements and Manufacturing*, vol. **43**(2), pp. 364 – 371.

Vemuri A. T. (1999), Diagnosis of Sensor Bias Faults, *Proceedings of American Control Conference*, San Diego, California, June, pp. 460-464.

Vinsonneau J.A.F., Shields D.N., King P.J. and Burnham K.J. (2001), Fault detection and modelling for an automotive engine, *Proceedings of the 14th International Conference on Systems Science*, vol. **3**, Wroclaw, Poland, 11-14 Sept., pp. 129-136.

Vong Chi-Man, Wong Pak-Kin and Li Yi-Ping (2006), Prediction of automotive engine power and torque using least squares support vector machines and Bayesian interface, *Engineering Applications of Artificial Intelligence*, vol. **19**, pp. 277-287.

Weerasinghe M., (1998), Fault detection and diagnosis for complex multivariable processes using neural networks, PhD thesis, Liverpool John Moores University, British thesis service no. DX212596.

Xie, M., Goh, T. N. and Cai D. Q. (2001), An integrated SPC approach for manufacturing processes, Fifth International Conference of the European Operations Management Association, June 1998, Dublin, Ireland, vol. 12(2), pp. 134-138.

Yang B. S., Han T. and An J. L., (2004), ART-KOHONEN neural network for fault diagnosis of rotating machinery, *Mechanical Systems and Signal Processing*, vol. **18**, pp. 645-657.

Yang J., Pu L., Wang Z., Zhou Y. and Yan X. (2001), Fault detection in a diesel engine by analysing the instantaneous angular speed, *Mechanical systems and signal processing*, vol. **15**(3), pp. 549-564.

Yong-Wha Kim, (1998), Automotive Engine Diagnosis and Control via Nonlinear Estimation, *IEEE Control Systems*, pp. 84-99.

Yu, D. L. (1995), Fault diagnosis for industrial systems with emphasis on bilinear systems, PhD thesis, Coventry University, UK.

Yu, D. L., Chang T. K. and Yu, D. W. (2005), Fault tolerant control of multivariable processes using auto-tuning PID controller, *IEEE Trans on Systems, Man and Cybernetics Part B*, vol. **35**(1), pp. 32-43.

Yu, D. L. and Gomm, J. B. (2003), Implementation of neural network predictive control to multivariable chemical reactor, *Control Engineering Practice*, vol. **11**(11), pp. 1315-1323.

Yu, D. L., Gomm, J. B. and Williams, D. (1999). Sensor fault diagnosis in a chemical process via RBF neural networks, *Control Engineering Practice*, vol. **7**(1), pp. 49-55.

Zhang Q. (1997), Using wavelet network in nonparametric estimation, *IEEE Transactions on Neural Networks*, vol. **8**(2), March, pp. 227-236.

Zhang Q. and Benveniste A. (1992), Wavelet Networks, *IEEE Transactions on Neural Networks*, vol. **3**(6), November, pp. 889-898.

Zhang, Xiaodong, Polycarpou, M.M. and Parisini, T. (2002), A robust Detection and Isolation Scheme for Abrupt and Incipient Faults in Nonlinear Systems. *IEEE Trans. Automatic Control*, vol. **47**(4), pp. 576-593.

APPENDEX 1

Author's publications

1. Sangha M S, Yu D L, Gomm J B (2006). On-Board Monitoring and Diagnosis for Spark Ignition Engine Air Path via Adaptive Neural Networks, *Proceedings of IMechE*, v. **220**, Part D, Journal of Automobile Engineering, pp. 1641-1655.

2. Sangha M S, Yu D L, Gomm J B (2007). Adaptive FDI for Automotive Engine Air Path and Robustness Assessment, *International Journal of Automotive Technology*, v. **8** (5), pp. 637-650.

3. Sangha M S, Yu D L, Gomm J B (2007). Robustness Assessment of Adaptive FDI System for Engine Air Path, *International Journal of Information & Systems Sciences*, v. **3** (1), pp. 180-205.

4. Sangha M S, Yu D L, Gomm J B (2008). Robustness Assessment and Adaptive FDI for Car Engine, *International Journal of Automation and Computing*. v. **5**(2), pp. 109-118.

5. Sangha M S, Gomm J B, Yu D L (2007). Neural Network Fault Classification of Transient Data in an Automotive Engine Air Path, *International Journal of Modelling Identification and Control*. (to appear in 2008).

6. Sangha M S, Yu D L, Gomm J B (2007). Sensor Fault Diagnosis for Automotive Engines with Real Data Evaluation, submitted to *IEEE Transections of Industrial Electronics.*

7. Sangha M S, Gomm J B, Yu D L (2005). Dynamic FDI of an Automotive Engine Using Artificial Neural Networks, *16th IFAC World Congress*. July 4-8, Prague.

8. Sangha M S, Yu D L, Gomm J B (2006). Engine Air Path Fault Diagnosis Using Adaptive Neural Classifier, *IEE International Control Conference*, 30 Aug to 01 Sept., Glasgow, UK.

9. Sangha M S, Yu D L, Gomm J B (2007). Adaptive Fault Diagnosis and Robustness Assessment under Closed-Loop Control, *13th Chinese Automation & Computing Society Conference*, 15 September, Staffordshire, England.

10. Sangha M S, Yu D L, Gomm J B (2007). Real Data Evaluation of FDI system for Sensor Faults in SI Engines, *13th Chinese Automation & Computing Society Conference*, 15 September, Staffordshire, England.

11. Sangha M S, Gomm J B, Yu D L and Page G. F. (2004). An Investigation of Neural Networks in Fault Diagnosis of an Automotive Engine Air Path, *2nd ASIM Workshop Wismar*, Germany – Modelling, control and diagnostics of combustion engine processes, September 16-17, pp. 23-32, ISBN 3-901608-27.

12. Sangha M S, and Gomm J B (2005). Steady State Fault Diagnosis of an Automotive Engine Air Path, *GERI Annual Research Symposium*, June 22, Liverpool John Moores University, UK.

13. Sangha M S, Yu D L, Gomm J B (2006). Automotive Engine Fault Diagnosis via Adaptive Neural Networks, *GERI Annual Research Symposium*, June 15, Liverpool John Moores University, UK.

14. Sangha M S, Yu D L, Gomm J B (2007). Real data evaluation of an adaptive fault diagnosis system for sensor faults in IC engine, *GERI Annual Research Symposium*, June 27, 2007, Liverpool John Moores University, UK.

15. Sangha M S, Jones K O, Yu D L, Gomm J B (2007). Adaptive FDI System for Engine Air Path and Its Robustness Assessment, *Proceedings of the International Conference on Information Technologies (InfoTech-2007)*, Bulgaria, September 21-23, v. **1**, pp.

PAGE/PAGES EXCLUDED UNDER INSTRUCTION FROM UNIVERSITY