

Fault Detection and Isolation In Gas Turbine Engines

Zakieh Sadough

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science at

Concordia University

Montréal, Québec, Canada

August 2012

© Zakieh Sadough, 2012

CONCORDIA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: Zakieh Sadough

Entitled: Fault Detection and Isolation In Gas Turbine Engines

and submitted in partial fulfilment of the requirements for the degree of

Master of Applied Science

Complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Dr. Soleymani
_____ Dr. dolatabadi, External
_____ Dr. Hashtrudi Zad, Examiner
_____ Dr. Khorasani, Supervisor
_____ Dr. Meskin, Supervisor

Approved by: _____

Dr. W. E. Lynch, Chair
Department of Electrical and Computer Engineering

_____ 2012 _____
Dr. Robin A. L. Drew
Dean, Faculty of Engineering and Computer Science

ABSTRACT

Fault Detection and Isolation In Gas Turbine Engines

Zakieh Sadough

Aircraft engines are complex systems that require high reliability and adequate monitoring to ensure flight safety and performance. Moreover, timely maintenance has necessitated the need for intelligent capabilities and functionalities for detection and diagnosis of anomalies and faults. In this thesis, fault diagnosis in aircraft jet engines is investigated by using intelligent-based methodologies. Two different artificial neural network schemes are introduced for this purpose. The first fault detection and isolation (FDI) scheme for an aircraft jet engine is based on the multiple model approach and utilizes dynamic neural networks (DNN). Towards this end, multiple DNNs are constructed to learn the nonlinear dynamics of the aircraft jet engine. Each DNN represents a specific operating mode of the healthy or the faulty conditions of the jet engine.

The inherent challenges in fault diagnosis systems is that their performance could be excessively reduced under sensor fault and sensor degradation conditions (such as drift and noise). This thesis proposes the use of data validation and sensor fault detection to improve the performance of the overall fault diagnosis system. In this regard the concept of nonlinear principle components analysis (NPCA) is exploited by using autoassociative neural networks.

The second FDI scheme is developed by using autoassociative neural networks (ANN).

A parallel bank of ANNs are proposed to diagnose sensor faults as well as component faults in the aircraft jet engine. Unlike most FDI techniques, the proposed solution simultaneously accomplishes sensor faults and component faults detection and isolation (FDI) within a unified diagnostic framework.

In both proposed FDI approaches, by using the residuals that are generated from the difference between each network output and the measured jet engine output as well as selection of a proper threshold for each network, criteria are established for performing the fault diagnosis of the jet engines. The fault diagnosis tasks consists of determining the time as well as the location of a fault occurrence subject to the presence of disturbances and measurement noise. Simulation results presented, demonstrate and illustrate the effective performance of our proposed neural network-based FDI strategies.

*To my mother Dr. Nabian and my father Dr. Sadough,
for their tireless support throughout my life
and to my husband Hamid for his care, love and patience*

ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisor Professor Khashayar Khorasani for his sincere encouragement, guidance, and support during my Masters studies. Completion of my masters would not have been possible without his ideas and suggestions. He was always accessible with his vast knowledge. I appreciate all his contributions of time and ideas.

I would also like to acknowledge the advice and guidance of Dr. Nader Meskin, who shared his experience and offered valuable suggestions. Many thanks for his assistance in the preparation and completion of this study.

My special thanks are to my committee members, Dr. Soleymani, Dr. Hashtrudi Zad and Dr. Dolatabadi for devoting their valuable time in guiding and evaluating my work.

I wish to thank all of my friends and colleagues who provided me a warm and friendly environment and memorable times in Montreal.

My deepest gratitude goes to my family for their unflagging love and support throughout my life. For my lovely mother whose love, faith and support are always giving me confidence and power to explore, grow and develop. For my father who motivates me to work hard by his endeavour and ambition. For my brother who always encourages me. And most of all for my loving, supportive, encouraging, and patient husband Hamid whose faithful support during the stages of this thesis is so much appreciated.

TABLE OF CONTENTS

LIST OF FIGURES	xi
LIST OF TABLES	xvi
LIST OF ABBREVIATIONS AND SYMBOLS	xviii
1 Introduction	1
1.1 Motivation of the Work	5
1.2 Literature Review on Fault Diagnosis	6
1.2.1 Fault Diagnosis	8
Artificial Neural Networks	9
Dynamic Neural Networks	10
1.2.2 Data Validation	11
1.2.3 Sensor Fault Detection	13
1.3 Thesis Contributions	16
1.4 Thesis Outline	18
2 Background Information	19
2.1 Dynamic Neural Networks (DNNs)	20
2.1.1 Dynamic Neuron Model	21
2.1.2 Dynamic Neural Network Architecture	23

2.1.3	Extended Dynamic Backpropagation Algorithm	25
2.2	Autoassociative Neural Networks	27
2.3	Autoassociative Neural Network (ANN) Structure	28
2.4	Data Preprocessing	32
2.5	Aircraft Jet Engine Mathematical Model	33
	Intake Duct	34
	Compressor	36
	Combustion Chamber	37
	Turbine	37
	Nozzle	38
	Rotor Dynamics	38
	Volume Dynamics	39
	Control Inputs	41
2.6	Conclusion	42
3	A Multiple-Model FDI Scheme Using Dynamic Neural Networks	44
3.1	Dynamic Neural Network FDI Approach	46
3.1.1	Threshold Selection Criterion	49
3.2	Simulation Results	51
3.2.1	System Identification Phase	52
3.2.2	Fault Detection and Isolation	55

3.2.3	Fault Detection Analysis Case Studies	58
3.2.4	Fault Isolation Case Studies Analysis	60
	Single Fault Scenario	60
3.2.5	Concurrent Component Fault Scenario	66
3.2.6	Conclusions	70
4	Data Validation Using Autoassociative Neural Network	71
4.1	Introduction	71
4.2	Autoassociative Neural Networks for Data Validation and Faulty Sensor Cor- rection	73
4.2.1	Network Training	74
4.3	Simulation Results	77
4.3.1	Training of the Autoassociative Neural Networks	77
	Noise Reduction	80
	Inaccurate Sensor Correction	82
	Removing Outlier using the ANN	82
	Error Correction	84
4.3.2	Effects of the None-Validated Data on Engine Fault Detection	92
4.3.3	Effects of Data Validation on Component Fault Detection	95
4.4	Conclusion	97

5	Multiple-Model Sensor and Components FDI Using Autoassociative Neural Networks	98
5.1	Sensor Fault Detection Scheme Using ANN	99
5.2	Engine Component Fault Diagnosis	100
5.3	Simulation Results	103
5.3.1	Sensor Fault Detection and Isolation	105
5.3.2	Component Fault Detection and isolation	113
	Single Fault Scenario	113
	Concurrent Components Faults and Sensor Fault Scenarios	119
	Concurrent Component Fault Scenario	122
5.4	Comparison of Autoassociative Neural Networks With Dynamic Neural Networks	123
5.5	Conclusion	125
6	Conclusions and Future Work	127
6.1	Thesis Summary	127
6.2	Suggestions for Future Work	129
	Bibliography	131

LIST OF FIGURES

1.1	Analytical versus hardware redundancy based FDI [1].	3
1.2	Schematic of a fault diagnosis approach [2].	4
1.3	GPA principle [3].	7
1.4	The impact of various sensor faults on system measurements [1]	14
2.1	A dynamic neuron having an internal IIR filter [4].	22
2.2	Dynamic neural network architecture.	24
2.3	Architecture of autoassociative neural network where σ denotes sigmoidal nodes and l indicates linear nodes.	29
2.4	A dual-spool jet engine [5].	34
2.5	The aircraft jet engine modules and information flowchart and interdependencies ([6, 7] and [8]).	35
3.1	The dynamic neural network architecture proposed for performing fault detection and isolation simultaneously.	48
3.2	The multiple model architecture corresponding to the four specific or more bank of dynamic neural networks.	57
3.3	The DNN0 generated residuals corresponding to THC, THT and N_1 for the case of a 5% $\Delta\Gamma_{LC}$ fault.	59

3.4	The DNN0 generated residuals corresponding to PLC, THT and N_1 for the case of a 5% $\Delta\eta_{LC}$ fault.	59
3.5	The DNN0 generated residuals corresponding to THC, THT and N_1 for the case of a 5% $\Delta\Gamma_{HC}$ fault.	60
3.6	The DNN1 to DNN8 generated residuals corresponding to THC, THT and N_1 for the case of a 5% $\Delta\Gamma_{LC}$ fault.	62
3.7	The DNN1 to DNN8 generated residuals corresponding to PLC, THT and N_1 for the case of a 5% $\Delta\eta_{LC}$ fault.	63
3.8	The DNN1 to DNN8 generated residuals corresponding to THC, THT and N_1 for the case of a 5% $\Delta\Gamma_{HC}$ fault.	64
3.9	Concurrent faults (5% $\Delta\Gamma_{LT}$ fault injected at t=16 seconds and the 5% $\Delta\eta_{LT}$ fault injected at t=25 seconds.)	69
4.1	Architecture of autoassociative neural network where σ denotes sigmoidal nodes and l denotes linear nodes	73
4.2	ANN structure for the gas turbine engine sensor validation.	79
4.3	Input and the output of the autoassociative neural network for noise filtering.	81
4.4	Input and the output of ANN corresponding to the recovery of sensor inaccuracies.	83
4.5	Input and the output of the ANN corresponding to the THC measurement signal outlier noise.	85

4.6	Input and the output of the ANN corresponding to the N1 measurement signal outlier noise.	85
4.7	Input and the output of the ANN corresponding to the PLC measurement signal outlier noise.	86
4.8	The sensor recovery rate and the deviation rate corresponding to TLC.	88
4.9	The sensor recovery rate and the deviation rate corresponding to PLC.	88
4.10	The sensor recovery rate and the deviation rate corresponding to THC.	89
4.11	The sensor recovery rate and the deviation rate corresponding to PHC.	89
4.12	The sensor recovery rate and the deviation rate corresponding to THT.	90
4.13	The sensor recovery rate and the deviation rate corresponding to PHT.	90
4.14	The sensor recovery rate and the deviation rate corresponding to N1.	91
4.15	The sensor recovery rate and the deviation rate corresponding to N2.	91
4.16	Effect of the non-validated data that produce false alarms (when input fuel flow rate is at 85% of its maximum).	93
4.17	Effect of the non-validated data that produce false alarms, (when input fuel mass flow rate is at 75% of its maximum).	94
5.1	Sensor fault detection scheme.	100
5.2	The multiple model architecture for bank of ANNs for fault diagnosis.	102
5.3	The ANN1 sensor reconstruction and the residual error for the sensor drift fault with the rate of 0.06% per second on THT and N2.	107

5.4	The ANN2 sensor reconstruction and the residual error for the sensor drift fault with the rate of 0.06% per second on THT and N2.	108
5.5	The ANN3 sensor reconstruction and the residual error for the sensor drift fault with the rate of 0.06% per second on THT and N2.	108
5.6	The ANN4 sensor reconstruction and the residual error for the sensor drift fault with the rate of 0.06% per second on THT and N2.	109
5.7	The ANN5 sensor reconstruction and the residual errors for the sensor drift fault with the rate of 0.06% per second on THT and N2.	109
5.8	The ANN6 sensor reconstruction and the residual errors for the sensor drift fault with the rate of 0.06% per second on THT and N2.	110
5.9	The ANN7 sensor reconstruction and the residual errors for the sensor drift fault with the rate of 0.06% per second on THT and N2.	110
5.10	The ANN8 sensor reconstruction and the residual errors for the sensor drift fault with the rate of 0.06% per second on THT and N2.	111
5.11	The ANN4 sensor reconstruction and residual errors for the sensor drift fault of 0.15% per second on PHC.	111
5.12	The ANN8 sensor reconstruction and the residual errors for the sensor drift fault of 0.15% per second on PHC.	112
5.13	The ANN1 to ANN8 generated residuals corresponding to the case of a 4% $\Delta\Gamma_{HT}$ fault.	114

5.14	The ANN1 to ANN8 generated residuals corresponding to the case of a 1% $\Delta\eta_{HT}$ fault.	115
5.15	The ANN1 to ANN8 generated residuals corresponding to the case of a 2% $\Delta\eta_{LT}$ fault.	117
5.16	The ANN6 and ANN8 generated residuals corresponding to the case of a 3% $\Delta\eta_{LT}$ fault.	118
5.17	The ANN5 generated residuals corresponding to the case of a 8% and 9% $\Delta\Gamma_{HT}$ fault.	118
5.18	The ANN1 to ANN8 generated residuals for the case of a 4% $\Delta\eta_{HT}$ fault injected at t=15 seconds and 20 K bias injected on the THC sensor at t=20 seconds.	121
5.19	concurrent fault	123

LIST OF TABLES

3.1	The definitions and descriptions of the considered components faults.	49
3.2	Selected neural networks parameters used for representing the healthy engine model.	55
3.3	The noise standard deviations (as percentage of the nominal noise at the cruise condition).	57
3.4	Threshold values that are used for achieving fault detection and isolation. . .	58
3.5	The fault detection results.	66
3.6	The fault isolation results (- or \surd denotes that one cannot isolate or can isolate a fault).	67
4.1	Trained autoassociative neural networks.	80
4.2	The percentage of the noise filtered by the ANN.	80
4.3	The percentage of the noise filtered by the ANN.	83
4.4	Outlier removing by the ANN.	84
4.5	The fault detection results.	96
5.1	The definitions and descriptions of the considered components faults.	101
5.2	The ANN structures and percentage of the noise filtering.	104
5.3	The selected thresholds for each ANNs output.	105

5.4	The minimum and the maximum biases (C.F. denotes complete failure, when the sensor reaches its full scale value).	112
5.5	The minimum and the maximum biases.	112
5.6	The minimum faults that are detectable by using each autoassociative neural network ANN1-ANN8 (the minimum fault that causes the residuals exceed their thresholds).	119
5.7	The maximum faults by which the corresponding ANN residuals would not exceed their thresholds.)	120
5.8	The maximum and the minimum faults that can be detected and isolated by using the bank of ANNs.	120
5.9	Characteristics and parameters of the DNN7 and ANN7.	124

LIST OF ABBREVIATIONS AND SYMBOLS

IIR	Infinite impulse response
FDI	Fault detection and isolation
ANN	Autoassociative neural network
RANN	Robust autoassociative neural network
DNN	Dynamic neural network
EKF	Extended Kalman filter
TLT	Temperature of low pressure turbine
THT	Temperature of high pressure turbine
TLC	Temperature of low pressure compressor
THC	Temperature of high pressure compressor
PLC	Pressure of low pressure compressor
PHC	Pressure of high pressure compressor
PHT	Pressure of high pressure turbine
PLT	Pressure of low pressure turbine
GPA	Gas path analysis
TCC	Temperature of combustion chamber
T_C	Temperature of compressor
T_T	Temperature of turbine

T_d	Intake temperature
P_d	Intake pressure
η_d	Intake efficiency
η_C	Compressor efficiency
η_T	Turbine efficiency
T_{amb}	Ambient temperature
P_{amb}	Ambient pressure
T_o	Output stage temperature
T_i	Input stage temperature
E	Rotor energy, J
β	Bypass ratio
W_T	Power generated by turbine, W
W_C	Power consumed by compressor
J	Rotor momentum of inertia
N	Rotational speed, RPM
N1	High pressure spool rotational speed, RPM
N2	Low pressure spool rotational speed, RPM
P	Pressure, bar
R	Gas constant, $\frac{J}{kg \cdot K}$
T	Temperature, K

V	Volume, m^3
\dot{m}	Mass flow, $\frac{kg}{s}$
\dot{m}_f	Fuel mass flow, $\frac{kg}{s}$
π	Pressure ratio
γ	Heat capacity ratio
c_p	Specific heat and constant pressure, $\frac{J}{kg.K}$
c_v	Specific heat at constant volume, $\frac{J}{kg.K}$
H_u	Fuel specific heat, $\frac{J}{kg}$
u	Speed, $\frac{m}{s}$
A	Area, m^2
M	Mach number
T_0	Temperature at sea level at standard day
P_0	Pressure at sea level at standard day
mech	Mechanical
W_f	Fuel mass flow rate

Chapter 1

Introduction

Modern complex systems require high precision and reliable performance due to the criticality and complexity. Fault diagnosis is essential for attaining such a high reliability in safety critical systems. In general, fault diagnosis is a process (or a technique) to detect the presence of faults and to determine their locations and to estimate their significance or severities in a system that is being monitored. The goal of fault detection and isolation system is to improve the reliability, availability and safety. Traditionally, redundancy and therefore fault diagnosis is achieved by using extra hardware, which is known as hardware redundancy approach to fault diagnosis. In this approach, multiple critical components such as an actuator or a sensor is used to control or measure a particular variable in the system. Typically, a voting technique is applied to the hardware redundant system to decide if a fault has occurred and the location of the component among all the redundant system parts. The major problems

encountered with hardware redundancy is that the increase in sensors leads to an increase in cost, weight, and complexity [2]. Consequently, another approach for generating redundancy known as the analytical redundancy was introduced in early 1970's by Beard [9]. Analytical redundancy, by contrast, eliminates the need for additional instrumentation hardware. In Figure 1.1, the hardware and analytical redundancy concepts are illustrated.

The analytical FDI approach can take a variety of forms including ordinary differential equations, intelligent data-driven models, and expert system models. Therefore, analytical redundancy-based fault diagnosis can generally be applied in three distinct frameworks based on the way the knowledge about the system is utilized. The first one is the mathematical model-based framework [2], [10–13], where *a priori* knowledge of the system is represented by the system's mathematical model derived by using physical principles. The second one is the learning-based or computational intelligence-based framework [14, 15] which utilizes system's historical data and data-driven models of the system. References [16, 17] provide comparison of various methods within the above two frameworks. Finally, the third framework includes the expert system-based or fuzzy rule-based approaches to fault detection and isolation (FDI) [18, 19], which use an expert's knowledge of the system operation and its failure modes to obtain a qualitative model of the system.

The analytical redundancy-based FDI, in general, consists of two main stages namely, residual generation and residual evaluation (or decision making), as shown in Figure 1.2. The residual generation process is based on comparison between the measured and the estimated

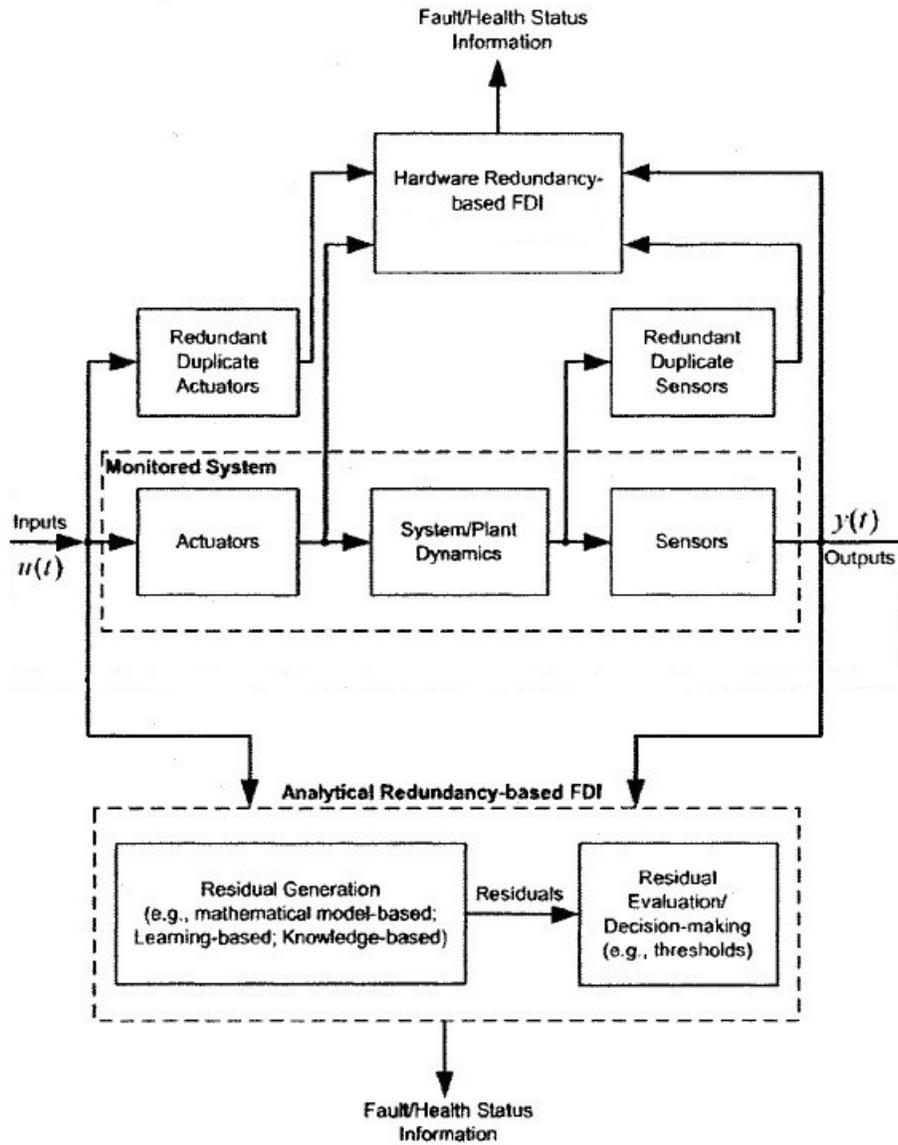


Figure 1.1: Analytical versus hardware redundancy based FDI [1].

system outputs, the resulting differences generated between the measured signals and the corresponding estimated signals obtained from the model is called a residual signal. The residual signal is expected to stay close to zero when no fault is present (normal operation) in the system, but should distinguishably diverge from the zero neighbourhood when a fault occurs. This property of the residual is used to determine whether or not faults are presented and occurred.

In the residual evaluation stage, on the other hand, the generated residuals are inspected for the likelihood of faults by analyzing the residual signal. The fault presence is determined by applying a decision rule. The decision rule may simply be a threshold test on the instantaneous values or the moving window averages of the residuals, or it may consist of more complex statistical approaches such as, likelihood ratio testing or sequential probability ratio testing [20, 21].

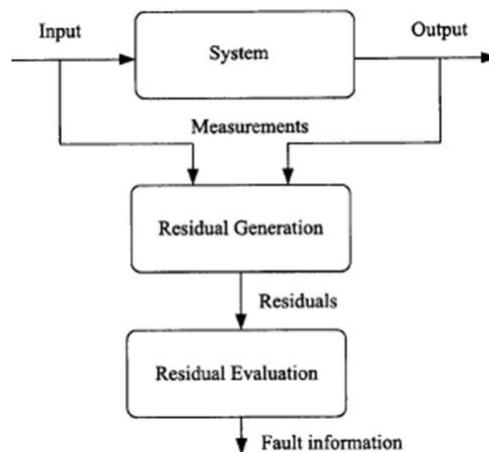


Figure 1.2: Schematic of a fault diagnosis approach [2].

1.1 Motivation of the Work

In this thesis the problem of fault diagnosis in aircraft jet engines is addressed by using intelligent-based methods. Fault diagnosis, that is the problem of fault detection and isolation (FDI), of aircraft jet engine has been a matter of wide interest in recent years due to the increasing demand and requirements on reliable operations and maintainability of these safety critical systems. Engine related costs comprise a large portion of the direct operating cost of an aircraft, particularly due to the overall maintenance costs of the propulsion system in aircraft. Fault diagnosis technologies allow one to avoid heavy economic losses due to stopped or aborted flights as well as the cost associated with untimely and unnecessary replacement of the components and parts. On the other hand, an early diagnosis of faults and anomalies in an engine makes it possible to perform important condition-based maintenance decisions and actions as opposed to conventional time-based maintenance actions.

The removal of noise and outliers from measurement signals is a major problem in jet engine fault diagnosis systems. The effectiveness and reliability of the FDI system is strongly limited by measurement uncertainties. Sensor measurements are the first essential factors needed for monitoring operating conditions of a jet engine to establish the fault diagnosis and performance analysis. The above mentioned facts, calls for the necessity of validating the quality of the measurement data prior to be used for health monitoring.

Sensor fault diagnosis is another point of investigation on engine performance evaluation. Indeed, before the measurements can be used for engine condition assessment, it must

be ensured that they correctly represent the measured physical quantities, namely that the corresponding sensor readings do not contain a fault such as bias or drift. Diagnostic tools allowing sensor fault detection are thus necessary and must be efficient specially in cases when component faults occur.

1.2 Literature Review on Fault Diagnosis

In the problem of jet engine fault diagnosis one utilizes knowledge on the measured variables taken along the engine's gas path to determine how an engine system performance differs from its desired state. Changes in the engine speeds, temperatures, pressures, fuel flows, etc., derive the required information for identifying the engine system malfunctions. Using such characteristics the most popular diagnosis procedure has appeared in the literature as the so-called Gas Path Analysis (GPA).

As expressed above the goal of GPA is to detect physical faults that consist of variety of problems or combinations of anomalies and factors such as foreign object damage (FOD), blade erosion and corrosion, worn seals, excess clearance or plugged nozzles, etc. Such physical faults cause changes in the thermodynamic performance of the engine and the components. The condition of the components can be mathematically represented by a set of independent performance parameters. The performance parameters that are mostly investigated in the literature are component efficiencies and flow capacities.

Figure 1.3 shows the GPA's main concepts [3]. In general, the fundamental idea underlying this approach is that physical faults occurring in the engine cause a change in the component performance as introduced by efficiencies and flow capacities which in turn produce observable changes in measurable parameters such as the temperature, pressure, speed, etc. If changes are then observed in the gas path measurements, the problem would be in detecting the fault and evaluating which module or thermodynamic parameter or components parameters are responsible for that change. This itself may assist for the prospective isolation of the physical faults.

A large number of methodologies have been proposed in the literature for GPA. Some of these are Kalman filter approaches [6, 7, 22, 23], neural networks [24, 25], fuzzy logic [26, 27], probabilistic networks [28], genetic algorithms [29], and hybrid diagnosis [30].

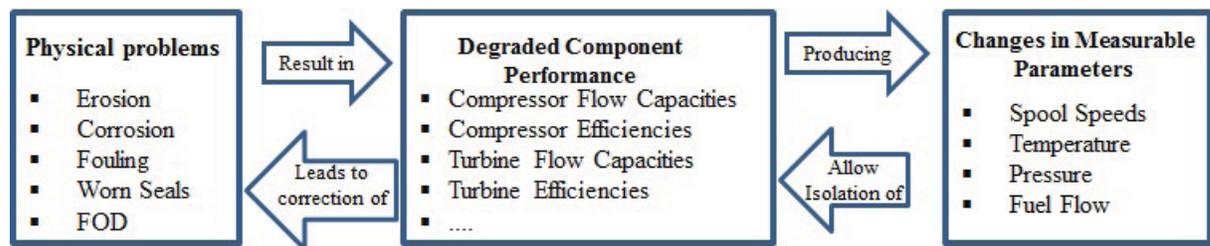


Figure 1.3: GPA principle [3].

1.2.1 Fault Diagnosis

A fault refers to unpredicted or unexpected deviation or change in a system's behaviour from that desired, for a bounded or unbounded period of time. The overall goals of the jet engine diagnostic system are to correctly detect, isolate and identify the changes in the engine modules. In other words, a fault diagnosis system is capable of performing the three tasks of detection, isolation, and identification of faults in a system, which are defined as follows [2]:

Fault detection: To make a binary decision whether something has gone wrong or that everything is fine.

Fault isolation: To determine the location of the fault, i.e., to identify which component, sensor, or actuator has become faulty.

Fault identification To estimate the severity, type or nature of the fault.

Fault diagnosis algorithms are mainly divided into two categories, namely model-based and data-driven (computational intelligent-based) techniques. Both model-based and data-driven techniques have been extensively studied in the literature for health monitoring of aircraft jet engines. Some of the important survey papers in the field of model-based FDI include Betta and Pietrosanto [31], Frank [32], Venkatasubramanian [33], [34], Isermann [10] and Marinai [35]. In recent model-based approaches, the observer based methods and Kalman filters are quite popular [6, 7, 23]. Although, such model-based techniques have their advantages in terms of on-board implementations, their reliability for health monitoring often decreases

as the system nonlinear complexities and modelling uncertainties increase. The inherent non-linearity of gas-turbine performance and diagnosis relationships as well as the limitations of the analytical model-based technique, makes the need for the application of an alternative computational technique, such as employing neural networks even more essential.

Artificial Neural Networks

Data-driven approaches such as those based on neural networks mostly rely on real-time or collected historical data from sensors and do not require a detailed mathematical model of the system [32,36–40]. Neural networks are promising tools for fault diagnosis due to their proven success in system identification and strong capability in learning nonlinear transformations that map a set of inputs to a set of outputs. Examples of works published in the field of NN-based FDI schemes include [41–45].

Applications of neural networks to engine fault diagnosis have been widely developed and discussed in the literature. Zedda and Singh [46] have proposed the use of a modular-based diagnosis system for a dual spool turbofan gas turbine. Multiple neural networks are proposed in [47] for fault diagnosis of a single shaft gas turbine. The authors in [48] have further extended multiple neural networks method to generate a cascaded network to isolate component and sensor faults. Green [39] have discussed the need to incorporate a neural network with other AI techniques to perform the estimates of the active life, diagnostics and prognostics capabilities for the engine. Romessis *et al.* [49] have applied a probabilistic neural

network (PNN) to diagnose faults and investigate the diagnostic capability of the PNN on turbofan engines. Volponi *et al.* [25] introduced a hybrid neural network where part of the model was replaced by influence coefficients. They reported that the accuracy of such a network was favourably compared with a backpropagation network and Kalman filter approach.

Dynamic Neural Networks

It has been shown in the literature that multi-layer perceptron networks (MLP) can be used as universal approximator for static nonlinearities and are capable of identifying any nonlinear unique static function [4]. In recent years several approaches have been suggested in the literature that incorporate dynamics to artificial neural networks due to the need for identification of dynamical systems. Patton *et al.* [50] have outlined artificial intelligence approaches to fault diagnosis of dynamic systems. Some of the most applied neural network structures for fault diagnosis in dynamical systems are, recurrent neural networks [51], time-delay neural networks (TDNN) [52] and dynamic neural networks (DNN) [4, 53].

In recurrent or time delayed networks, the network is fed with current or delayed values of the process inputs and outputs. They count as quasi dynamical models, since the neural network used in the structure remains a static approximator. On the other hand, the dynamic networks have dynamic elements within their structure. Therefore, they provides a viable tool for dealing with nonlinear problems and modeling complex and nonlinear dynamical systems

with great flexibility and capability. Dynamic multilayer perceptrons or dynamic neural networks have recently been applied and utilized for system identification problems due to their capabilities in modelling nonlinear dynamical systems. Such networks have a feedforward multilayer architecture and their dynamic properties are achieved by using dynamic neurons. Each neuron by itself possesses dynamic characteristics that is constructed through a locally recurrent globally feedforward (LRGF) scheme [4, 53].

Recently, dynamic neural networks have been utilized for fault diagnosis of nonlinear systems. The authors in [40] have used a multilayer perceptron network embedded with dynamic neurons for fault detection and isolation (FDI) of thrusters in the formation flight of satellites. A dynamic neural network is constructed in [54] for accomplishing the fault detection task, and a static neural classifier is then used based on the learning vector quantization (LVQ) for the fault isolation task. The authors in [8] have applied dynamic neural networks that was developed in [53] for fault detection of aircraft jet engines.

1.2.2 Data Validation

Gas turbine performance analysis make use of measurements (such as gas path temperatures, gas path pressures and rotational speeds) to recognize the poorly performing engine and to isolate and identify the cause of the deficiencies or faults. As all fault diagnosis systems require correct sensor measurements, unreliable sensors can cause the system to move the diagnosis in an erroneous direction. Therefore, data validation of sensor measurements and correcting data

from sensors is a prerequisite in applying fault diagnosis techniques. Traditional approaches to sensor validation involve periodic instrument calibration. These calibration processes are expensive. Many periodic sensor calibration techniques require the process shut down, the instrument taken out of service, and the instrument loaded and calibrated. This method can lead to damaged equipment, incorrect calibrations due to adjustments made under non-service conditions, and loss of product due to unnecessarily shutting down a process [55].

Several approaches have been used for sensor data validation. For example, analytical redundancy using an on-line nonlinear model of a turbo fan engine is proposed in [56] to provide estimates for failed sensors. The reference [57] has proposed unknown input observer as a robust sensor and actuator fault detection, isolation, and accommodation techniques. The usual approach to deal with measurements uncertainty is to use techniques based on Kalman filter (KF), which should be able to estimate engine performance parameters and measurement biases in the presence of noise [58, 59]. However, Kalman filter based estimation techniques are affected by several drawbacks, resulting in inaccuracy and lack of reliability [56]. In general, using model-based methods one always encounters uncertainties as the system complexity increases. In this thesis, autoassociative neural networks is used by incorporating nonlinear principal component analysis (NPCA) concept to detect, identify and reconstruct faulty sensors in gas turbine engine.

1.2.3 Sensor Fault Detection

Sensors are basically the output interface of a system to the external world, and convey information about a system's behaviour and internal states. Therefore, sensor faults may cause substantial performance degradation of all decision-making systems or processes that depend on data integrity for making decisions. Such systems include, but not limited to, feedback control systems, safety control systems, quality control systems, navigation systems, surveillance and reconnaissance systems, state estimation systems, optimization systems, and interestingly health monitoring and fault diagnosis systems [1].

Common sensor faults/failures include: (a) bias; (b) drift; (c) performance degradation (or loss of accuracy); (d) sensor freezing; and (e) calibration error [60]. Figure 1.4 depicts the effect of the above faults on system measurements.

The design of sensor fault diagnosis schemes using the hardware redundancy and analytical redundancy approaches have been addressed in the literature [61]. In the hardware redundancy approach, redundant sensor systems are incorporated into the control system to improve the reliability of sensor measurements and enable sensor fault detection. However, cost and space make this approach unattractive. In contrast, the analytical redundancy-based fault-diagnosis architectures use system physics based models and information processing methods to achieve the necessary redundancy.

In the literature, both data-driven and model-based approaches have been proposed to diagnose different sensor faults. The majority of model-based sensor fault diagnosis schemes

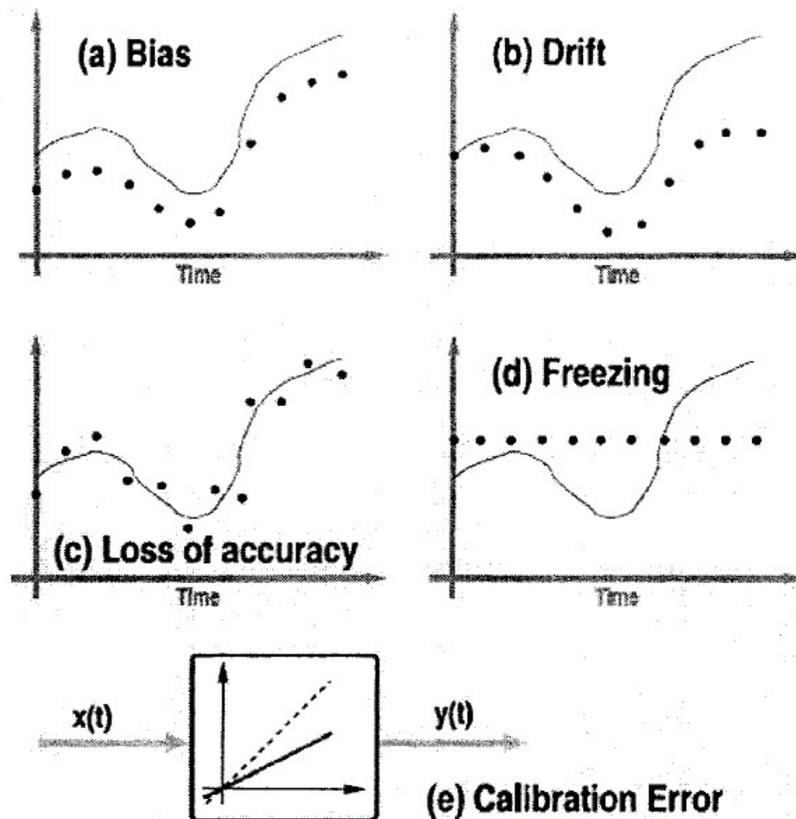


Figure 1.4: The impact of various sensor faults on system measurements [1]

rely on linear time-invariant (LTI) models [62] which can be considered as the major challenge. Unfortunately, in nonlinear, time-varying systems, LTI models can sometimes fail to give satisfactory results. As an alternative there has been a growing interest in the use of adaptive neural networks (NNs) as nonlinear system approximators [36].

There are also several sensor fault diagnosis schemes that are proposed in the literature, which are specifically designed to detect multiple sensor faults. These include the dedicated observer scheme (DOS) (Clark [63]), generalised observer scheme (GOS) (Frank [32]), multiple model Kalman filter (MMKF) (Willsky et al. [20]) and the multi-layer perceptron

(MLP)-NN based SFDIA scheme proposed in [64].

In spite of the popularity of model-based approaches in fault detection, this is challenging and sometimes inappropriate method for sensor fault detection. This is due to the fact that most model-based approaches are designed at a nominal health condition and rely on the correct input data in general [65]. It assumes that the input to the real system and the input to the model are correct (fault free). Any observed deviation (due to any faults related to the actuator, sensor or components) in the engine outputs from their reference condition values indicates the presence of a fault. This makes the sensor fault detection unreliable and challenging.

Sensor fault detection isolation and accommodation (SFDIA) via neural networks have been proposed over the years due to their nonlinear structures, online learning capabilities and no needs for explicit mathematical model [62,66–71]. Comparison between Kalman filter and neural network approaches for sensor validation or fault detection is provided in [72].

Autoassociative neural networks (ANNs) have been used extensively in the recent past few years as a solution in sensor fault detection and identification. Hines and Uhrig [55] applied the ANN method to detect faulty sensors. The author in [73] proposes a sensor fault detection and repair method based on Autoassociative neural network to detect multi-faulty transducers of an IEEE 1451 based intelligent sensor, synchronously, through a well-trained ANN. The reference [74] identified a single fault sensor using an enhanced ANN and the exact value of the fault sensor was reconstructed. The rationale for the use of ANN in sensor fault detection is their capacity to provide a robust identity mapping between the input and the

output of the network, which could be exploited in sensor fault detection. In Chapter 5 we have also shown how the ANN can be exploited for engine fault diagnosis as well.

1.3 Thesis Contributions

In this thesis, our goal is to develop novel solutions for the problem of aircraft jet engine components and sensor fault detection and isolation based on artificial neural network approaches. Towards this end, two different neural network-based schemes are proposed which each possess noble features. The contributions of this thesis in solving the above problem is listed as follows:

First, a multiple model dynamic neural network-based scheme for component fault detection and isolation is proposed which makes use of both the benefits of multiple model characteristics and the advantages of artificial neural networks. On the other hand, the dynamic neural networks used in this scheme are an ideal tool in identifying nonlinear dynamic systems under different operating modes. Indeed, unlike the approaches in the literature which use static neural network, dynamic neural network renders it possible to develop a single nonlinear neural network for a range of operating conditions. The developed neural network-based multiple model is applied to the aircraft jet engine.

Second, knowing the fact that using the validated and qualified data enhances the reliability of the FDI system, an intelligent-based approach using autoassociative neural networks is developed for validation and qualification of the jet engine data.

Third most of the standard approaches in the literature are proposed for either sensor fault detection or system fault detection or they incorporate either two or three separate subsystems to accomplish the tasks of fault diagnosis in both sensors and components. Consequently, both FDI systems need to be active in order to isolate the sensor faults from the system faults. As a novel approach, we developed an integrated diagnostic approach that simultaneously diagnose and isolate both the sensor faults and component faults. In our framework we also propose a criterion to investigate the residuals that are obtained from the difference between each neural network-based model outputs and the measured jet engine outputs in order to isolate the sensor faults and component faults. Using our approach fast and accurate detection and isolation of both sensor and component faults can be obtained. In addition, in the event of a failed sensor the scheme is capable of replacing the faulty sensor value with the virtual true sensor value. The scheme is extremely robust to measurement noise and it can very reliably and accurately perform FDI, even in presence of large measurement noise.

The capabilities of our proposed dynamic neural network-based FDI approach and autoassociative neural network-based sensor and component fault diagnosis approach are demonstrated under different fault scenarios. Finally, the two proposed fault diagnosis approaches are compared in terms of performance and their capabilities in accomplishing fault diagnosis of aircraft engines.

1.4 Thesis Outline

The remainder of this thesis is organized as follows. In Chapter 2, we briefly review the background material which will be used in the following chapters. We review the structure of the two proposed artificial neural networks namely dynamic neural networks and autoassociative neural networks. We also review the nonlinear mathematical modelling of the aircraft jet engine that is used in this work. In Chapter 3, our fault detection and isolation (FDI) methodology using dynamic neural networks is proposed along with the simulation results showing the effectiveness of our approach. The results of applying the autoassociative neural networks for data validation in the jet engine output variables are presented in Chapter 4. In Chapter 5, our proposed integrated sensor and component fault diagnosis scheme is described in detail. Finally, concluding remarks and future work are included in Chapter 6.

Some of the results of this research have already been published in the following conference and journal:

[1] S. Sina Tayarani Bathaie, **Z. Sadough** and K. Khorasani, “Fault Detection of Gas Turbine Engines using Dynamic Neural Networks”, *IEEE Canadian Conference on Electrical and Computer Engineering*, Montreal, Quebec, April, 2012.

[2] S. Sina Tayarani Bathaie, **Z. Sadough** and K. Khorasani, “Dynamic Neural Network-based Fault Diagnosis of Gas Turbine Engines”, *Neurocomputing*, NEUCOM-D-12-00466, 2012.

Chapter 2

Background Information

In this chapter, we present an overview of the background material related to our work. In this thesis we have studied fault diagnosis of aircraft jet engines using neural network methodologies. In this chapter, we introduce two different neural networks and provide the aircraft jet engine model to which our fault diagnosis methodologies are applied to. We first describe and introduce dynamic neural networks (DNN) as an efficient tool for nonlinear dynamic systems identification. The DNN is later used in Chapters 3 and 4 for engine components fault diagnosis. Next, we introduce autoassociative neural networks which will later be used for data validation and fault diagnosis in Chapters 4 and 5, respectively. Finally, we briefly describe the nonlinear mathematical model of a dual spool jet engine that is used to develop a SIMULINK model of the system for data generation and neural network training and validation.

2.1 Dynamic Neural Networks (DNNs)

Dynamic multilayer perceptrons or dynamic neural networks (DNNs) have recently been applied and utilized for dynamic system identification problems due to their capabilities in modelling nonlinear dynamical systems. Such networks have the feedforward multilayer architecture and their dynamic properties are obtained by using dynamic neurons. Each neuron by itself possesses dynamic characteristics which can lead to constructing a locally recurrent globally feedforward (LRGF) network. This kind of structure allows one to design an effective feedforward multilayer network that both has dynamic characteristics and has less complexity than time-delay and recurrent networks which use a global feedback in their structures [4, 53].

Dynamic neural networks presented in [4] and [53] have a great capability in learning the dynamics of complicated nonlinear systems where conventional static neural networks cannot yield an acceptable modeling performance. Dynamic neural networks or dynamic multilayer perceptron networks (MLP) represent an extension of static neural networks by including discrete or continuous time dynamics to the neuron model. Such an extension enhances the capability of the resulting neural network to approximate not only the static nonlinearities of the system but also its dynamic nonlinearities. The dynamic neuron model and the dynamic neural network architecture are presented in the following subsections.

2.1.1 Dynamic Neuron Model

A dynamic neuron model [4] is constructed by adding internal dynamics and by making the neuron's activity dependent on its internal states. This can be achieved by integrating an Infinite Impulse Response (IIR) filter within the standard static perceptron structure. Figure 2.1, represents the structure of such a dynamic neuron model. Three main modules are used in this structure. The first module is an adder, namely

$$x(k) = \mathbf{W}^T \mathbf{u}(k) = \sum_{p=1}^P w_p u_p(k) \quad (2.1)$$

where $\mathbf{W} = [w_1 w_2 \dots w_P]^T$ denotes the input-weight vector, P denotes the number of inputs, and $\mathbf{u}(k) = [u_1(k) u_2(k) \dots u_P(k)]^T$ is the input vector (T denotes the transpose operator). The output of the adder is passed through the IIR filter ($H(q^{-1})$) through which a dynamic mapping is then generated between the input and the output of the neuron. Applying an n^{th} order filter, the output of the filter and the filter transfer function are given by

$$\tilde{y}(k) = -a_1 \tilde{y}(k-1) - a_2 \tilde{y}(k-2) - \dots - a_n \tilde{y}(k-n) + b_0 x(k) + b_1 x(k-1) + \dots + b_n x(k-n) \quad (2.2)$$

or

$$\tilde{y}(k) = \sum_{i=0}^n b_i x(k-i) - \sum_{i=1}^n a_i \tilde{y}(k-i) \quad (2.3)$$

and

$$H(q^{-1}) = \frac{b_0 + b_1q^{-1} + b_2q^{-2} + \dots + b_nq^{-n}}{1 + a_1q^{-1} + a_2q^{-2} + \dots + a_nq^{-n}} \quad (2.4)$$

where $x(k)$ denotes the filter input, $\tilde{y}(k)$ denotes the filter output, $\mathbf{a} = [a_1, a_2, \dots, a_n]^T$ and $\mathbf{b} = [b_0, b_1, b_2, \dots, b_n]^T$ are the numerator and the denominator coefficients of the filter transfer function (feedback and feedforward filter parameters) and q is the time shift operator. Therefore, the neuron output can be expressed as:

$$y(k) = F(g \cdot \tilde{y}(k)) \quad (2.5)$$

where $F(\cdot)$ is a nonlinear activation function that produces the neuron output and g is the slope of the activation function.

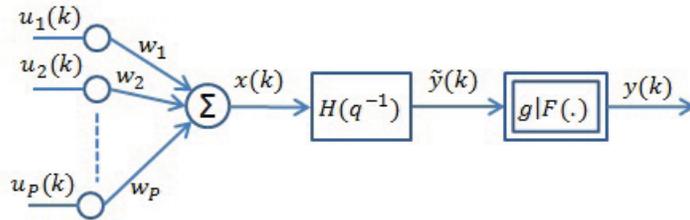


Figure 2.1: A dynamic neuron having an internal IIR filter [4].

In many cases time delays exist between inputs and outputs. For the systems that contain time delays and when these are considerable for the system performance and cannot be ignored, $\tilde{y}(k)$ can be modified as follows:

$$\tilde{y}(k) = \sum_{i=0}^n b_i x(k-i-n_k) - \sum_{i=1}^n a_i \tilde{y}(k-i) \quad (2.6)$$

In order to use such a modification, one needs the delayed inputs $x(k-i-n_k)$ and the parameter n_k which can be determined arbitrarily or by using optimization procedure [75].

2.1.2 Dynamic Neural Network Architecture

Let us consider an L-layered network as shown in Figure 2.2 using dynamic neurons that are described by a differentiable activation function $F(\cdot)$. Let N_l denote the number of neurons in the l -th layer, $O_n^l(k)$ denote the output of the n th neuron of the l th layer, and $u_p^l(k)$ denote the input of the l th layer, generated from the p -th neuron of the previous layer at discrete times k ($l = 1, \dots, L; n = 1, \dots, N_l$). It can be shown [76] that the output of the n th neuron in the l th layer is given by

$$O_n^l(k) = F[g_n^l(\sum_{d=0}^D b_{dn}^l \sum_{p=1}^{N_{l-1}} w_{np}^l u_p^l(k-d) - \sum_{d=1}^D a_{dn}^l \tilde{y}_n^l(k-d))] \quad (2.7)$$

It can be seen from equation (2.7) that the network outputs depend on the past outputs $\tilde{y}(k-1), \tilde{y}(k-2), \dots, \tilde{y}(k-n)$. Since it is assumed that the activation function $F(\cdot)$ is an invertible function (e.g. tangent hyperbolic), then network outputs will also depend on the past outputs $y(k-1), y(k-2), \dots, y(k-n)$. Consequently, the expression for the last layer outputs is given by equation (2.8), where $\Gamma(\cdot)$ is a nonlinear function. This illustrates that the

network outputs are nonlinear functions of the inputs and their delays as well as the previous output samples, that is

$$O_n^L(k) = \Gamma[y(k-1), \dots, y(k-m_s), u(k), u(k-1), \dots, u(k-n_s)] \quad (2.8)$$

The main objective of the neural network learning process is to adjust all the unknown network parameters so that the nonlinear jet engine system can be identified by the proposed dynamic neural network by using a given training set of input-output data pairs. The unknown network parameters are denoted by \mathbf{w} , \mathbf{a} , \mathbf{b} , \mathbf{g} , where $\mathbf{w} = [w_{np}^l]_{l=1, \dots, L; n=1, \dots, N_l; p=1, \dots, N_{l-1}}$ is the weight matrix, $\mathbf{a} = [a_{dn}^l]_{l=1, \dots, L; n=1, \dots, N_l; d=1, \dots, D}$ and $\mathbf{b} = [b_{dn}^l]_{l=1, \dots, L; n=1, \dots, N_l; d=1, \dots, D}$ are the filter parameters matrices, where D denotes the order of the filter, and $\mathbf{g} = [g_n^l]_{l=1, \dots, L; n=1, \dots, N_l}$ denotes the slope parameter matrix.

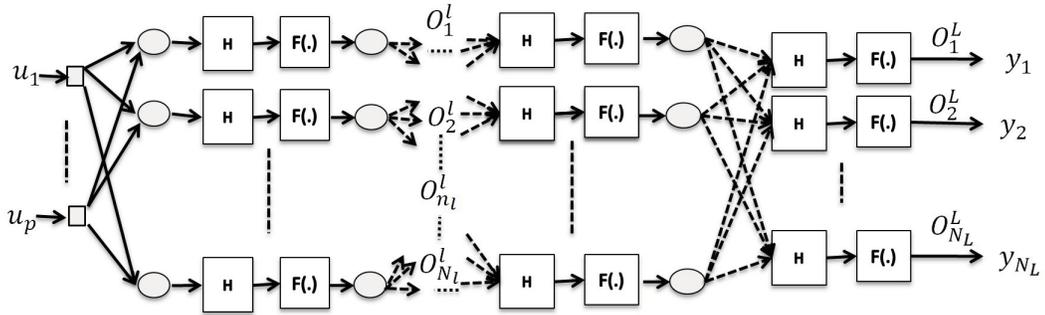


Figure 2.2: Dynamic neural network architecture.

To adjust the network parameters, pairs of healthy input and output data sets are used. The backpropagation error is widely applied for the purpose of training static networks. Its

extension to dynamic applications is known as the extended dynamic backpropagation algorithm [76].

2.1.3 Extended Dynamic Backpropagation Algorithm

In both static and dynamic neural networks, the objective is to determine an adaptive algorithm or a rule which adjusts the parameters of the network based on a given set of input-output pairs. The idea of the error backpropagation is widely applied for this purpose in static contexts and has extension to dynamic systems. To define an extended dynamic backpropagation (EDBP) algorithm, the standard approach can be applied. Assuming that the unknown parameter vectors \mathbf{w} , \mathbf{a} , \mathbf{b} , \mathbf{g} are considered as elements of a parameter vector \mathbf{v} , the learning process involves the determination of the vector \mathbf{v}^* which minimizes the performance index $J_v(k)$ according to the error function $\mathbf{e}(k)$:

$$J_v = \|e(k)\|^2 = \|y^d(k) - y(k)\|^2 \quad (2.9)$$

where $y^d(k)$ denotes the desired output of the network and $y(k)$ denotes the actual response of the network on the given input pattern $\mathbf{u}(k)$ [76].

The adjustment of the parameters of the s^{th} neuron in the m^{th} layer according to the EDBP algorithm in a M-layered network has the form,

$$v_s^m(k+1) = v_s^m(k) + \eta \delta_s^m(k) S_{vs}^m(k) \quad (2.10)$$

where $v = [w, a, b, g]$ represents the unknown generalized parameter vector, η is the learning rate, δ_s^m is the generalized output error which is described below for both hidden and output layers (equation (2.11)), and S_{vs}^m denotes the sensitivity function for the elements of the unknown generalized parameter v (equations 2.12-2.15).

$$\delta_s^m = \begin{cases} e_s(k)F'(g_s^M \tilde{y}_s^M(k)) & \text{for } m = M \\ \sum_{l=1}^{S_{m+1}} (\delta_s^{(m+1)}(k) g_l^{m+1} b_{0l}^{m+1} w_{ls}^{m+1}) F'(g_s^m \tilde{y}_s^m(k)) & \text{for } m = 1, \dots, M - 1 \end{cases} \quad (2.11)$$

where $F(\cdot)$ is a nonlinear activation function that produces the neuron output as described in equation 2.5.

The sensitivity function $S_{vs}^m(k)$ for the elements of the unknown generalized parameter v is defined as follows:

1. Sensitivity with respect to the feedback parameter a_{is}^m :

$$S_{a_{is}}^m(k) = -g_s^m \tilde{y}_s^m(k - i) \quad (2.12)$$

2. Sensitivity with respect to the feedforward parameter b_i :

$$S_{b_{is}}^m(k) = g_s^m x_s^m(k - i) \quad (2.13)$$

3. Sensitivity with respect to the weight parameter w_p :

$$S_{w_{ps}}^m(k) = g_s^m \left(\sum_{i=0}^D b_{is}^m u_p^m(k-i) - \sum_{i=1}^D a_{is}^m S_{w_{ps}}^m(k-i) \right) \quad (2.14)$$

4. Sensitivity with respect to the slope parameter g_s^m :

$$S_{g_s}^m(k) = \tilde{y}_s^m(k) \quad (2.15)$$

2.2 Autoassociative Neural Networks

Autoassociative neural networks [77] are feedforward neural networks that are used to acquire input-output models by using backpropagation training or similar learning procedures. In autoassociative neural networks the function to be learned is the identity mapping between network inputs and outputs, implying that the outputs are an approximation of the inputs. By selecting a proper internal architecture and training the network to learn the identity mapping, the autoassociative neural networks can carry out several useful data screening tasks such as reducing the measurements noise. In addition, by constructing the residuals from the difference of the inputs and outputs of the network, they can be used to detect sensor faults as well as the missing and faulty sensor data can be estimated.

Autoassociative networks are different from networks that implement associative memory even though both are used to treat noisy and corrupted data. Autoassociative neural

networks perform functional mappings while associative memories are basically classifiers, recalling a stored typical examples that most closely resembles a partial or corrupted input patterns. In bidirectional associative memory networks [78] the output is a stored pattern associated with the classification of the input. Therefore, the number of possible responses from the associative memory is finite, and for any input one of the pre-stored patterns is recalled. In contrast, the autoassociative network has no discrete classes and its outputs can be continuous variables.

2.3 Autoassociative Neural Network (ANN) Structure

The general structure of an autoassociative neural network (ANN) as shown in Figure 2.3, contains three hidden layers. The first hidden layer is called the mapping layer. The activation function of the mapping layer can be sigmoidal, tangent hyperbolic or any other similar non-linearity. The second hidden layer is called the bottleneck layer and can have linear transfer functions. The dimension of the bottleneck layer should be smaller than the dimension of the other hidden layers. The third hidden layer is called the demapping layer and has the same activation function as the mapping layer. The mapping and the demapping layers have the same dimension.

The bottleneck layer output is the compressed representation of the data given in the input layer. If the inputs are a set of observations of correlated variables, the mapping layer

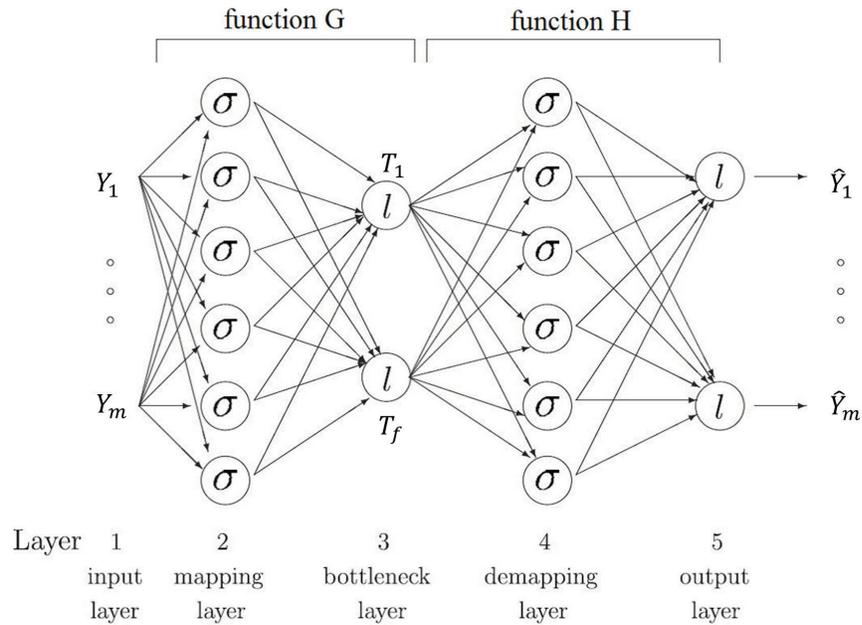


Figure 2.3: Architecture of autoassociative neural network where σ denotes sigmoidal nodes and l indicates linear nodes.

converts these sets of correlated observations into a set of uncorrelated variables. The autoassociative neural network is derived from the concept of principle components analysis which is applicable to both linear and nonlinear correlations among variables. The output of the nodes in the bottleneck layer can be viewed as principle components as compact representation of the inputs. Same as the principle component analysis, the goal of the ANN in the bottleneck layer is to compress the data into a set of new variables in new space with lower dimensionality so that the data can be described as concisely as possible. An important issue regarding ANN is that it can deal with linear and nonlinear correlations among the variables and produce a compact and concise data representation.

The use of a structure with three hidden layers as opposed to one hidden layer is due

to the need for data compression inside the network in order to filter out both noise and biases. According to the Figure 2.3, the autoassociative network should be viewed as a cascade combination of two single-hidden layer networks. The input, the mapping and the bottleneck layers together represent a nonlinear function $\mathbf{G} : R^m \rightarrow R^f$ which projects the inputs to a lower dimensional space designated as the feature space. This mapping has the following form

$$\mathbf{T} = \mathbf{G}(\mathbf{Y}) \quad (2.16)$$

where \mathbf{G} is a nonlinear vector function, composed of f individual nonlinear functions ($\mathbf{G} = [G_1, G_2, \dots, G_f]^T$). Let T_i denotes the output of the i^{th} bottleneck node or the i^{th} element of $\mathbf{T} = [T_1, T_2, \dots, T_f]$, $i = 1, \dots, f$, and $Y = [Y_1, Y_2, \dots, Y_m]^T$ denotes the network input. Therefore, the map is described according to $G_i : R^m \rightarrow R$ which has the form

$$T_i = G_i(Y), \quad i = 1, \dots, f \quad (2.17)$$

For the inverse transformation (restoring the original dimensionality of the data), the bottleneck layer output, the demapping layer and the output layer represent a second network that is modelled as a nonlinear function $\mathbf{H} : R^f \rightarrow R^m$, which reproduces an approximation to the input from the factors at the output of the bottleneck layer. This mapping has the following form

$$\hat{\mathbf{Y}} = \mathbf{H}(\mathbf{T}) \quad (2.18)$$

where \mathbf{H} is a nonlinear vector function composed of m individual nonlinear functions ($\mathbf{H} = [H_1, H_2, \dots, H_m]^T$). Each output can be described according to $H_j : R^f \rightarrow R$, such that

$$\hat{Y}_j = H_j(T), \quad j = 1, \dots, m \quad (2.19)$$

For sake of generality, the subnets representing \mathbf{G} and \mathbf{H} functions must each be capable of representing nonlinear functions of arbitrary nature. This can be achieved by providing each subnetwork with a single layer containing a sufficiently large number of nodes. The mapping layer is the hidden layer of the subnet representing \mathbf{G} , and the demapping layer is the hidden layer of the subnet representing \mathbf{H} .

Autoassociative networks require “supervised” training, where a desired output is specified for each training example. One cannot train the network representing \mathbf{G} by itself, since the output \mathbf{T} is unknown. Similarly, the network \mathbf{H} cannot be trained separately even though the desired output is known (the target output is \hat{Y}), because the corresponding input \mathbf{T} is unknown. Therefore, direct supervised training of each of these networks individually is infeasible. To circumvent this problem, the two networks are combined in series so that \mathbf{G} feeds directly into \mathbf{H} , resulting in a network whose inputs and desired outputs are known. Specifically, \hat{Y} uses both the input to \mathbf{G} and the desired output from \mathbf{H} . The combined network with \mathbf{G} and \mathbf{H} in series contains three hidden layers, since the bottleneck layer is shared, being the output of \mathbf{G} and the input layer \mathbf{H} , as shown in Figure 2.3. Finally, for classifying the data into valid and invalid sets, the residual signals defined as $R = Y - Y'$ are generated, and the

threshold values are then selected properly for each residual.

2.4 Data Preprocessing

Normalization is a “scaling down” transformation of the data. Within a data set there is often a large difference between the maximum and the minimum values. When normalization is performed the value of the signals are scaled to appreciably lower values. The two most common methods for normalization are as follows.

- min- max normalization

$$x' = 2 \frac{x - x_{min}}{x_{max} - x_{min}} - 1 \quad (2.20)$$

where x_{min} and x_{max} denote the minimum and the maximum of x , respectively. This re-scales the variable x to lie within the range $[-1,1]$.

- z-score normalization

$$x' = \frac{x - \mu(x)}{\sigma(x)} \quad (2.21)$$

where $\mu(x)$ and $\sigma(x)$ denote the mean and the standard deviation of x , respectively. This normalization produces a data set where each point has a mean close to zero and a variance close to one.

Normalization should be applied to all the data sets before commencement of the training process. One should note that the means and standard deviations that are computed

from the data must be retained and used later in the testing process for de-normalization. Otherwise, the performance of the neural network will vary significantly as it was trained on a different data representation than the un-normalized data. The advantage of the statistical normalization is that it reduces the effects of the outliers in the data.

2.5 Aircraft Jet Engine Mathematical Model

Gas turbine engines are used in many land, sea and air vehicles. The jet engine (shown in Figure 2.4) belongs to one type of gas turbine engines and is used to generate a high-speed jet for propulsion. A mathematical representation of a gas turbine is fairly common and has been investigated by several authors in the literature [8,30,79]. Based on the available literature [8] on modeling of aircraft jet engines, a Matlab Simulink model of the nonlinear dynamics of the a dual spool jet engine is used in this thesis. The simulation model was developed by using thermodynamic, aerodynamic and mechanical relationships of each of the major components. The model represents the functional relations that exist among the engine variables, such as pressures, temperatures and gas flow rates. The details of the thermodynamic relations reviewed in this section can be found in [79]. Rotor and volume dynamics are considered in order to obtain a nonlinear dynamics for the system. The engine components (compressors and turbines) are modelled by corresponding performance maps which are adopted from commercial software GSP [7]. In the following, brief explanation of each specific component and detailed mathematical expressions corresponding to the engine dynamics are presented. A schematic

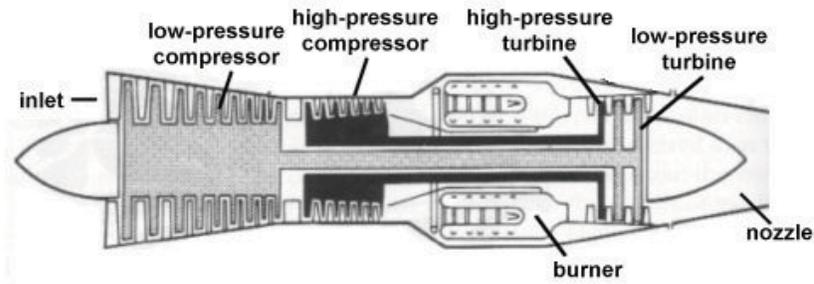


Figure 2.4: A dual-spool jet engine [5].

diagram depicting the main modules and the overall information flows and interdependencies are shown in Figure 2.5.

Intake Duct

Intake duct is placed before the compressor and supplies the engine with the required air flow at highest possible pressure. The air velocity in the intake duct decreases when air reaches the compressor. At the same time, the temperature and the pressure increase. In the engine intakes, by assuming adiabatic process, the pressure and temperature are computed as follows, with the inlet pressure ratio equation can be written as

$$\frac{P_d}{P_{amb}} = \left[1 + \eta_d \frac{\gamma - 1}{2} M^2 \right]^{\frac{\gamma}{\gamma - 1}} \quad (2.22)$$

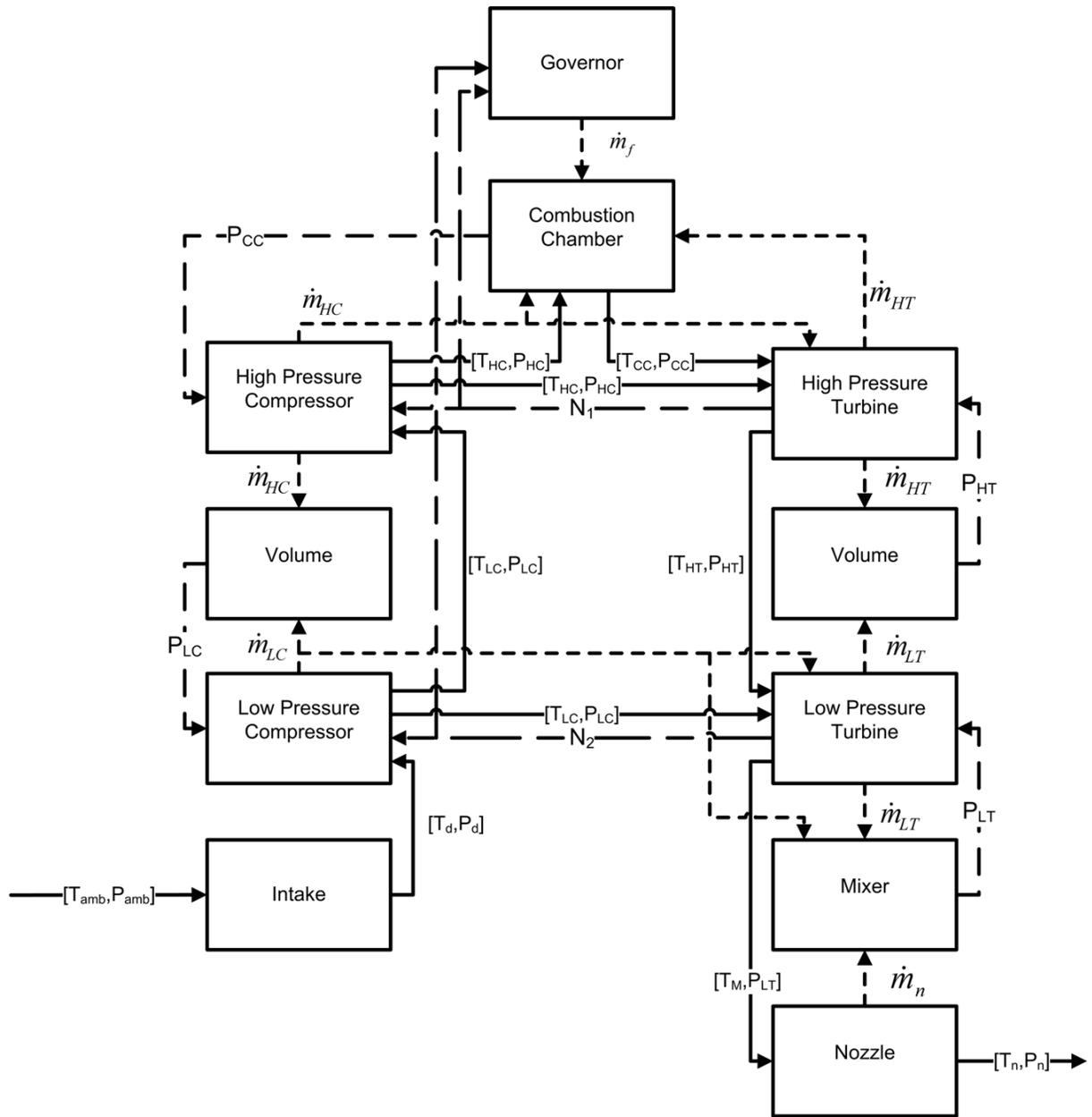


Figure 2.5: The aircraft jet engine modules and information flowchart and interdependencies ([6, 7] and [8]).

where M is the mach number in the air temperature and pressure. The inlet temperature ratio can be expressed in terms of M as

$$\frac{T_d}{T_{amb}} = 1 + \frac{\gamma - 1}{2} M^2 \quad (2.23)$$

Compressor

A compressor in a gas turbine engine is in charge of providing high-pressure air to the combustion chamber. The compressor behaviour, as a quasi-steady component, is determined by using the compressor performance map (this map is obtained from the commercial software package GSP [80]). Given the pressure ratio (π_C) and the corrected rotational speed ($N/\sqrt{\theta}$), one can obtain the corrected mass flow rate ($\dot{m}_C\sqrt{\theta}/\delta$) and efficiency (η_C) from the performance map by using a proper interpolation technique, where $\theta = \frac{T_i}{T_0}$ and $\delta = \frac{P_i}{P_0}$, i.e. $\dot{m}_C\sqrt{\theta}/\delta = f_{\dot{m}_C}(N/\sqrt{\theta}, \pi_C)$ and $\eta_C = f_{\eta_C}(N/\sqrt{\theta}, \pi_C)$. Once these parameters are obtained, the compressor temperature rise and the mechanical power are obtained as follows:

$$T_0 = T_i \left[1 + \frac{1}{\eta_C} (\pi_C^{\frac{\gamma-1}{\gamma}} - 1) \right] \quad (2.24)$$

$$W_C = \dot{m}_C c_p (T_0 - T_i) \quad (2.25)$$

It should be noted that the power consumed by the compressor is related to the speed of the shaft $W_C = \frac{J(\frac{N \cdot 2\pi}{60})^2}{2}$ where J is the momentum of inertia of the shaft and N is the speed of

the shaft.

Combustion Chamber

Combustion chamber is the place in the engine in which the fuel is burned in the high pressure air supplied by the compressor to raise the temperature. The rise in the temperature is due to the energy released by the burning fuel. The combustion chamber represents both the energy accumulation and the volume dynamics between the high pressure compressor and the high pressure turbine at the same time. The dynamics inside the combustion chamber is governed by the following equations,

$$\dot{P}_{CC} = \frac{P_{CC}}{T_{CC}} \dot{T}_{CC} + \frac{\gamma R T_{CC}}{V_{CC}} (\dot{m}_C + \dot{m}_f - \dot{m}_T) \quad (2.26)$$

$$\dot{T}_{CC} = \frac{1}{c_v m_{CC}} [(c_p T_C \dot{m}_C + \eta_{CC} H_u \dot{m}_f - c_p T_{CC} \dot{m}_T) - c_v T_{CC} (\dot{m}_C + \dot{m}_f - \dot{m}_T)] \quad (2.27)$$

Turbine

The function of the turbine in a jet engine is to extract a portion of the pressure and kinetic energy from the high-temperature combustion gases for driving the compressor and accessories. In a typical engine about 75 percent of the power produced is used internally to drive the compressor. The remaining power is used to generate the required thrust [79]. Like compressors, the behaviour of a turbine is represented by characteristic maps (from the software package GSP [80]). Given the pressure ratio (π_T) and the corrected rotational speed ($N/\sqrt{\theta}$),

the corrected mass flow rate ($\dot{m}_T \sqrt{\theta}/\delta$) and the efficiency (η_T) are obtained from the performance map, i.e. $\dot{m}_T \sqrt{\theta}/\delta = f_{\dot{m}_T}(N/\sqrt{\theta}, \pi_T)$ and $\eta_T = f_{\eta_T}(N/\sqrt{\theta}, \pi_T)$. The temperature drop and the turbine mechanical power (which is proportional to the temperature decrease in the turbine) are obtained as follows:

$$T_0 = T_i \left[1 - \eta_T (1 - \pi_T^{\frac{\gamma-1}{\gamma}}) \right] \quad (2.28)$$

$$W_T = \dot{m}_T c_p (T_i - T_0) \quad (2.29)$$

Nozzle

Nozzle is the final component of a jet engine in which the working fluid is expanded to produce a high-velocity jet. The high pressure exhaust gas is accelerated in a jet pipe located between the turbine outlet and the nozzle throat to come close to the ambient pressure and consequently, to produce thrust. The nozzle exit temperature T_{n_o} is given by

$$T_{n_i} - T_{n_o} = \eta_n T_{n_o} \left[1 - \left(\frac{1}{P_{n_i}/P_{amb}} \right)^{(\gamma-1)/\gamma} \right] \quad (2.30)$$

Rotor Dynamics

Energy balance between the shaft and the compressor results in the following differential equation:

$$\frac{dE}{dt} = \eta_{mech} W_T - W_C \quad (2.31)$$

where $E = \frac{J(\frac{N.2\pi}{60})^2}{2}$ (J is the moment of inertia of the shaft and N is the speed of the shaft).

Volume Dynamics

The volume dynamics is considered to take into account the unbalance mass flow rates among various components. Assuming that the gas has zero speed and has homogeneous properties over volumes, this dynamics can be described by the following equation:

$$\dot{P} = \frac{RT}{V} (\sum \dot{m}_{in} - \sum \dot{m}_{out}) \quad (2.32)$$

Now that all the components of the engine are described, we explain how the temperatures, the pressures or the pressure ratios and the rotational speeds can be obtained from the above nonlinear equations for each component.

For a low pressure compressor, the pressure ratio π_{LC} is calculated from the volume dynamics between the high pressure compressor and the low pressure compressor as described by equation (2.32). The rotational speed (N_2) is obtained from the solution to equation (2.5) for the spool that is connecting the low pressure compressor to the low pressure turbine. According to the pressure ratio and the rotational speed, the corrected mass flow and the efficiency are obtained from the performance maps, therefore the temperature rise can be obtained from the equation (2.24). The same procedure is followed for the high pressure compressor. The

pressure is obtained from the volume dynamics that is described by equation (2.26). The rotational speed ($N1$) is obtained from equation (2.5) for a spool that is connecting the high pressure compressor to the high pressure turbine.

Finally, the pressure ratio of high pressure turbine is obtained from the volume dynamics between the high and the low pressure turbines, and the pressure ratio for the low pressure turbine is obtained by using the volume dynamics after the low pressure turbine. The mass flow rate of the nozzle is computed as follows.

If condition (2.33) exists. the mass flow rate can be obtained from equation (2.34), otherwise, it is calculated from equation (2.34), that is

$$\frac{P_{amb}}{P_{n_i}} < \left[1 + \frac{1 - \gamma}{\eta_n(1 + \gamma)} \right]^{\frac{\gamma}{\gamma-1}} \quad (2.33)$$

$$\frac{\dot{m}_n \sqrt{T_{n_i}}}{P_{n_i}} = \frac{u}{\sqrt{T_{n_i}}} \frac{A_n P_{amb}}{R P_{n_i}} \frac{T_{n_i}}{T_{n_o}} \quad (2.34)$$

$$\frac{\dot{m}_n \sqrt{T_{n_i}}}{P_{n_i}} = \frac{u}{\sqrt{T_{n_i}}} \frac{A_n P_{crit}}{R P_{n_i}} \frac{T_{n_i}}{T_{crit}} \quad (2.35)$$

where $\frac{u}{\sqrt{T_{n_i}}} = \sqrt{2c_p \eta_n (1 - (\frac{P_{amb}}{P_{n_i}})^{\frac{\gamma-1}{\gamma}})}$, $\frac{T_{N_o}}{T_{n_i}} = 1 - \eta_n (1 - (\frac{P_{amb}}{P_{n_i}})^{\frac{\gamma-1}{\gamma}})$, and where $\frac{P_{crit}}{P_{n_i}} = (1 - \frac{1}{\eta_n} (\frac{\gamma-1}{\gamma+1}))^{\frac{\gamma}{\gamma-1}}$, $\frac{u}{\sqrt{T_{n_i}}} = \frac{2\gamma R}{\gamma+1}$, and $\frac{T_{crit}}{T_{n_i}} = \frac{2}{\gamma+1}$.

Here, it is assumed that $P_{n_i} = P_{LT}$ and $T_{n_i} = T_M$, T_M is calculated from the energy balance in

the mixer as follows:

$$T_M = \frac{\dot{m}_{LT}T_{LT} + \beta\dot{m}_{LC}T_{LC}}{\dot{m}_{LT} + \beta\dot{m}_{LC}} \quad (2.36)$$

Control Inputs

The input or the control signal of the dual spool engine is the power level angle (PLA) that is set by the pilot which is related to the fuel mass flow rate (W_f) through a variable gain. The relation between the PLA and the control inputs are considered as follows in this work [30]

$$\begin{cases} \frac{PLA \times W_f^{max}}{70} & \text{if } PLA \leq 70^\circ \\ W_f^{max} & \text{if } PLA > 70^\circ \end{cases} \quad (2.37)$$

Corresponding to different phases or stages of the flight (such as Take off, climb, cruise and descent) the engine experiences different operating regimes, namely shutdown, starting, idle thrust, acceleration, deceleration, cruise thrust, etc. In this thesis the fault diagnosis problem is addressed in the cruise mode when the engine is in its steady state.

The resulting engine model that is considered in this thesis has twelve (12) measurable variables (refer to Figure 2.5) that are defined as TLC, PLC, PHC, THC, N1, N2, PLT, TLT, PHT, THT, PCC and TCC, where they represent the low pressure compressor temperature, low pressure compressor pressure, high pressure compressor pressure, high pressure compressor temperature, high pressure spool speed, low pressure spool speed, low pressure turbine pressure, low pressure turbine temperature, high pressure turbine pressure, high pressure turbine

temperature, combustion chamber pressure, and combustion chamber temperature, respectively.

As explained in Section 1.2, common component faults are modelled as changes in the components efficiency and flow capacity. In this thesis eight (8) component faults are investigated which are namely,

- Low pressure compressor efficiency decrease
- Low pressure compressor flow capacity decrease
- High pressure compressor efficiency decrease
- High pressure turbine flow capacity decrease
- High pressure turbine flow capacity decrease
- High pressure turbine efficiency decrease
- Low pressure turbine flow capacity decrease
- Low pressure turbine efficiency decrease

2.6 Conclusion

In this chapter, an overview of the background material related to our work were presented.

Two different neural networks were introduced and the aircraft jet engine mathematical model

and the equations are provided on which our fault diagnosis methodologies are developed and applied to.

Chapter 3

A Multiple-Model FDI Scheme Using Dynamic Neural Networks

In this chapter, the problem of fault detection and isolation (FDI) of gas turbine engines is presented. A neural network-based fault detection and isolation scheme is proposed to detect and isolate faults in a highly nonlinear dynamic system corresponding to an aircraft jet engine. Towards this end, dynamic neural networks (DNN) are developed to learn the dynamics of the jet engine. The DNN is constructed based on a dynamic multilayer perceptron network which uses infinite impulse response (IIR) filters to generate dynamics between the input and output of the system. The dynamic neural networks that is described in this chapter is developed to detect and isolate component faults that may occur in a dual spool turbo fan engine. The fault detection and isolation scheme consists of multiple DNNs, each representing various operating

modes of the healthy and faulty conditions. Using the residuals produced by the difference of each network output and the measured system output a criterion has been established for fault diagnosis of the system components. Various simulations are carried out to demonstrate the performance of our proposed fault diagnosis scheme. In this chapter, it is assumed that there are no sensor faults occurring during the component fault detection and isolation process.

Recently dynamic neural networks have been utilized and employed in achieving fault detection and isolation due to their capability in learning the dynamics of nonlinear systems. The authors in [40] have used a multilayer perceptron network embedded with dynamic neurons for fault detection and isolation (FDI) of thrusters in the formation flight of satellites. A dynamic neural network is constructed in [54] for accomplishing the fault detection task, and a static neural classifier is then used based on the learning vector quantization (LVQ) for the fault isolation task. The authors in [8] have applied dynamic neural networks that was developed in [53] for fault detection of aircraft jet engines. Our proposed scheme provides an integrated solution for both fault detection and isolation of jet engines in a single framework using the dynamic neural network-based multiple model strategy. It will be shown that by using a bank of dynamic neural networks the problem of fault detection and isolation of a dual spool jet engine can be addressed quite effectively.

3.1 Dynamic Neural Network FDI Approach

In this section, a fault diagnosis methodology for a dual spool gas turbine engine is developed. Towards this end, a dynamic neural network-based multiple model scheme is proposed in which the bank of dynamic neural networks acts as an estimator or identifier of different engine operating conditions corresponding to the various but limited faulty *modes* that are of most interest or possible in the jet engine.

The dynamic neural network-based multiple model idea is derived and motivated from that of multiple model-based FDI schemes in the literature [6, 7], where the mathematical models corresponding to multiple operating conditions are replaced by a parallel bank of dynamic neural network identifiers. The basic structure of the FDI scheme that uses dynamic neural networks is illustrated in Figure 3.1. The proposed neural network-based multiple model scheme requires training data on the healthy and faulty situations in order to learn all the classes of the system behavior.

Note that for the purpose of only fault detection, the dynamic neural networks is trained with only the data corresponding to the healthy condition of the jet engine. In this case learning data can be collected directly from the healthy engine, if possible, or from a simulation model that is as realistic and high fidelity as possible. However, for the purpose of fault isolation since the dynamic neural networks need to be trained for different faulty situations, such data has to be obtained through high fidelity simulation studies or over the life of the real engine (the engine that has been deteriorated and the percentage of the deterioration is known).

According to Figure 3.1, a bank of dynamic neural network models is needed for fault detection and isolation purposes. The first model (Model 0) represents the healthy system and the others represent the corresponding assumed L faulty conditions of interest to be isolated. The residuals are generated by comparing the jet engine outputs and the dynamic neural networks outputs. Consequently, when the residual r_0 is smaller than an appropriately selected threshold, the system is considered healthy, otherwise a system is considered faulty. For fault isolation, unlike the detection phase, the residuals (r_1, r_2, \dots, r_L) are close to zero (or below their thresholds) associated with the faulty condition. Indeed, a fault is isolated by evaluating the residuals so that before the occurrence of a fault all the residuals would be above their thresholds or significantly different from zero given the healthy condition of the system. Once a fault occurs, the residual for the corresponding fault model should be close to zero or below a threshold and the residuals generated for the other fault models should then be above their thresholds or significantly different from zero.

It is worth to noting that in order to reduce the computational cost of the proposed FDI system during the normal operation of the engine, it is also possible to just have the healthy model active and once the fault is detected then fault models will be activated to isolate the faults.

The jet engine component faults considered correspond to changes in eight (8) health parameters which are the efficiencies and the flow capacities of the low pressure compressor, the high pressure compressor, the low pressure turbine, and the high pressure turbine.

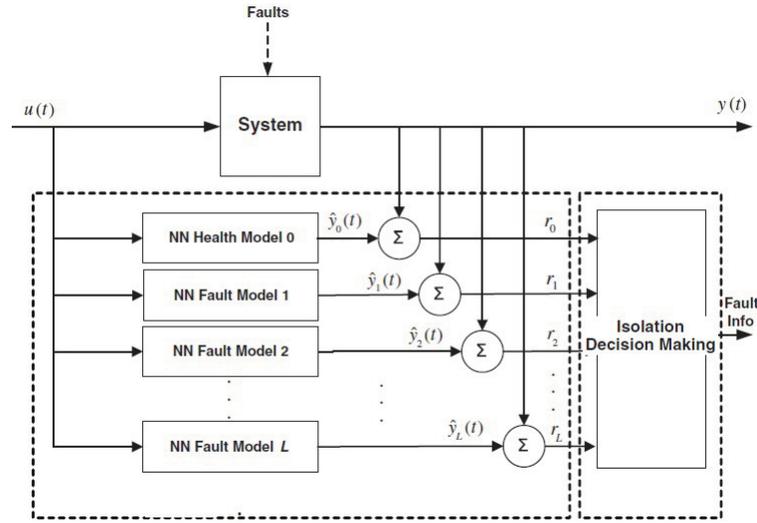


Figure 3.1: The dynamic neural network architecture proposed for performing fault detection and isolation simultaneously.

Therefore, eight (8) component faults, as shown in Table 3.1, are investigated in this thesis to be detected and isolated. Hence, a total of nine (9) models or dynamic neural networks (DNN) are needed ($L = 8$), where each model represents and is associated with one class of the jet engine behavior. The input to DNNs is fuel mass flow rate (W_f) and the outputs are engine variables such as pressures, temperatures and rotational speeds. The networks DNN1 to DNN8 correspond to the component faults and the ninth dynamic neural network DNN0 corresponds to the healthy mode of the jet engine. Table 3.1 presents the associated network labels for each fault scenario.

Component Fault	Description	Dynamic Neural Networks Label
$\Delta\Gamma_{LC}$	Low pressure compressor flow capacity decrease	DNN1
$\Delta\eta_{LC}$	Low pressure compressor efficiency decrease	DNN2
$\Delta\Gamma_{HC}$	High pressure compressor flow capacity decrease	DNN3
$\Delta\eta_{HC}$	High pressure compressor efficiency decrease	DNN4
$\Delta\Gamma_{HT}$	High pressure turbine flow capacity decrease	DNN5
$\Delta\eta_{HT}$	High pressure turbine efficiency decrease	DNN6
$\Delta\Gamma_{LT}$	Low pressure turbine flow capacity decrease	DNN7
$\Delta\eta_{LT}$	Low pressure turbine efficiency decrease	DNN8

Table 3.1: The definitions and descriptions of the considered components faults.

3.1.1 Threshold Selection Criterion

In order to evaluate the residuals and to obtain information about the faults, a simple threshold selection technique which employs the statistical parameters of the residuals is applied. As mentioned earlier, for the purpose of fault detection the residuals should ideally be very close to zero when the system is healthy and should deviate noticeably from zero when a fault occurs in the system. However, in practice due to modelling uncertainties and measurement noise, it is necessary to assign appropriate thresholds larger than zero in order to avoid false alarms. Indeed, in presence of measurement noise and system disturbances, the residual signals shall remain in the vicinity of zero under healthy condition and diverge from the neighbourhood of zero (i.e., exceed a certain threshold band around zero) when faults occur in the system. On the other hand, by selecting the thresholds too high may lead to the FDI scheme missing low severity faults. This imposes a tradeoff between reducing the number of false alarms and the number of missed alarms (i.e., missing to detect the presence of an actually occurred fault).

Below a probabilistic threshold selection method is proposed as a reliable solution to

this trade-off. Assume that the residuals are normal random variables expressed as

$$r_i(k) = \varepsilon(k), \quad k = 1, \dots, K; \quad i = 0, \dots, L \quad (3.1)$$

where $\varepsilon(k)$ is a normal random variable $N(m, v)$, with the mean m and the standard deviation v , K and L denote the size of the data used for testing and the number of operating models, respectively. A significance level β is first defined corresponding to the probability that a residual exceeds the value of the threshold denoted by T_i [75], that is

$$\beta = \text{prob}(|r_i(k)| > T_i) \quad \text{for } i = 0, 1, \dots, L \quad (3.2)$$

Standardizing ¹ the normal random variable $r(k)$ to have zero mean and a standard deviation of 1, β is now written as

$$\beta = \text{prob}(|z(k)| > t_{i\beta}) \quad \text{for } i = 0, 1, \dots, L \quad (3.3)$$

where $z(k)$ is standard normal random variable and $t_{i\beta}$ is obtained from

$$t_{i\beta} = \frac{(T_i - m)}{v} \quad (3.4)$$

¹In mathematical statistics, a random variable X is standardized by subtracting its expected value $E[X]$ (the mean of the random variable) and dividing the difference by its standard deviation $\sigma(X) = \sqrt{\text{Var}(X)}$,
 $Z = \frac{X - E(X)}{\sigma(X)}$

By specifying a certain value for an acceptable probability of false alarms (β), $t_{i\beta}$ can then be found by using the cumulative normal probability tables. In this way, by assuming a significance level β , one can obtain $t_{i\beta}$ and then the threshold T_i is selected according to

$$T_i = t_{i\beta}v + m \quad (3.5)$$

The parameters m and v for each residual ($r_0, r_1, r_2, \dots, r_L$) are empirically obtained through conducting multiple Monte Carlo simulations corresponding to random noise of 100 runs of each input setting. The mean and standard deviations of the residuals in the steady state during the healthy or faulty operation of the jet engine are computed for each run and then the average corresponding to all the runs are considered as the mean and standard deviations (m and v) for the residual signals of the bank of dynamic neural networks.

3.2 Simulation Results

The implementation of our proposed dynamic neural network-based fault detection and isolation scheme consists of three main tasks, namely (i) System Identification, (ii) Fault Detection, and (iii) Fault Isolation. The implementation of these steps and the case studies are explained and illustrated in detail below.

3.2.1 System Identification Phase

The performance of the system identification phase plays an important role in the capability of the overall fault diagnosis scheme. Three subtasks have to be performed for this phase, which include (a) the data preprocessing, (b) architecture and training, and (c) testing of the proposed dynamic neural networks.

(a) Data Preprocessing: The engine variables have various amplitude and ranges, and there is often a large difference between their maximum and minimum values. Neural network training could be made more efficient by performing certain preprocessing steps on the network inputs and targets. It has been observed that the training procedure is sensitive to the data normalization method and after several investigations we have determined that among the normalization methods discussed in Section 2.4, the max-min normalization process yields the best results. Specifically, we employ the following operation on the raw data that is generated by using our simulator, that is

$$X_n = 2 * \frac{(X - a)}{b - a} - 1 \quad (3.6)$$

where a and b denote the maximum and the minimum of the range of the signal X , respectively. This normalization is applied to both the input and the output signals of the neural networks.

(b) Dynamic Neural Networks Architecture and Training: The training data are generated from the simulation model of the jet engine, such that the engine input fuel mass flow rate is in the form of periodic triangular signal having a period of 2000 samples. Due to the high

complexity of the engine dynamics, a large amount of data is required for network to learn the dynamics of the engine. To adjust the weights and filter coefficients of each dynamic neural network, the so-called extended back propagation algorithm was used. The learning algorithm is initialized with small random values for the network parameters (namely, weights, feedback filters, activation function slope) while the IIR filter's denominator coefficients are initially set to zero to ensure stable learning. The activation functions in the hidden layers are taken as hyperbolic tangent functions, and linear activation functions for the output layer neurons.

The order of all the IIR filters is set to 2 since choosing higher order filters does not necessarily lead to better performance while it increases the computational cost. Starting with a relatively small structure, we developed an optimal architecture of our proposed nine (9) dynamic neural networks in each bank of dynamic neural networks by incrementally increasing the number of neurons in the hidden layers and also changing the learning rates for each parameter until a desired performance specification is satisfied.

In order to achieve the desired performance, first the number of training data must be large enough so that the dynamic neural network is able to learn the dynamics of the system quite well. Second, the network update parameters (weights \mathbf{w} , filter feed back parameters \mathbf{b} , filter feedforward parameters \mathbf{a} , and slope coefficient \mathbf{g}) must converge to a certain value after several iterations. Selecting an appropriate learning rate helps for faster convergence. Finally, the performance (cost) index needs to be as minimum as possible for training and testing data. Indeed the best structure has been selected by changing the number of neurons and selecting

the structure with the lowest performance (cost) index.

Each dynamic neural network model was trained by using suitable data corresponding to healthy or faulty modes. Subsequently, the performance of the constructed models are examined by using both the nominal and the faulty data. We observed that having one hidden layer in the network requires a relatively large number of neurons in that hidden layer to learn the engine dynamics. Hence, we have used two hidden layers for all the networks.

The dynamic neural networks specifications for four trained variables THC, THT, PLC and N1 that are representative of the system in the healthy condition (health model 0 in Figure 3.1) is presented in Table 3.2. This table shows the dynamic neural networks structure (number of neurons in the first hidden layer n_1 and the number of neurons in the second hidden layer n_2), the updating learning rates parameters (η_w, η_a, η_b and η_g), the number of iterations for each network and two performance indices for the training and the testing (J_{Train} and J_{Test} , respectively). The associated performance index is defined according to

$$J = \frac{\sum_{i=1}^n (y_d(n) - y_{Net}(n))^2}{\sum_{i=1}^n (y_d(n))^2} \quad (3.7)$$

where $y_d(n)$ and $y_{Net}(n)$ denote the desired and the network outputs, respectively, and n is the size of the training or the testing samples. Other dynamic neural networks corresponding to the faulty modes of the engines have almost the same specifications as those given in Table 3.2.

(c) Dynamic Neural Network Testing: The representation capabilities of the trained networks are evaluated through generalizing them with another data set of 6,000 samples that

Engine Variables	n_1	n_2	η_a	η_b	η_g	η_w	# of Iterations	J_{Train}	J_{Test}
PLC	1	3	0.3	0.5	0.5	0.5	40,000	0.17%	0.35%
THC	4	7	0.2	0.5	0.5	0.5	60,000	0.1%	0.9%
THT	4	5	0.9	0.5	0.5	0.5	60,000	0.09%	0.2%
N1	7	6	0.5	0.5	0.5	0.5	100,000	0.15%	0.7%

Table 3.2: Selected neural networks parameters used for representing the healthy engine model.

were not seen previously by the dynamic neural networks. Table 3.2 shows the value of the performance index J_{Test} that is obtained during the testing phase. It shows that the selected networks carry out the estimation of the output variables quite well.

3.2.2 Fault Detection and Isolation

In order to evaluate the performance of our proposed fault detection and isolation (FDI) scheme, nine (9) sets of training data are generated, one set of data corresponding to normal operating condition as well as eight sets of data corresponding to eight (8) engine component faults. The component faults are simulated by decreasing the value of the component efficiency or the flow capacity by specific percentages. We have selected a 5% severity as a typical level to illustrate the capabilities of our proposed FDI scheme.

It should be noted that by using only one or two output measurements it might *not be possible* to isolate all the eight (8) faults and *at least* three (3) measurements, and consequently three banks of neural networks are necessary. The structure of the neural network banks is depicted in Figure 3.2. Any output measurement can be chosen to construct a bank of dynamic neural networks. However, we have observed that different fault types and scenarios manifest

different effects on the measurement variables. In Figure 3.2 four output measurements PLC, THC, THT, and N1 that show better detection and isolation performance are depicted.

The faults are injected in the steady state (cruise) mode of the aircraft operation where the engine transients have all settled down. All the analysis was concluded in presence of noise. The measurement noise level was chosen to be twice the nominal noise at cruising condition. The nominal value of the noise levels are shown in Table 3.3 where the standard deviations are given as percentage of the nominal values at typical cruise conditions [28]. The analysis has been conducted for input profiles in the range of 70% to 95% of the maximum fuel value, as this is the fuel mass flow rate input range for the cruise mode. The ambient conditions are set to standard condition ² and the Mach number is 0.7 as a typical Mach number in cruise mode. Both fault detection and isolation are performed by using threshold selection criterion that were explained in Section 3.1.1 assuming a significant level of $\beta = 0.02$ (2%).

As an example the threshold for DNN0 network corresponding to the variable THC is obtained as follows:

Since β is set equal to 0.02, from $\beta = Prob(|z(k)| > t_\beta) = 2Prob(z(k) < -t_\beta)$ and using the cumulative normal probability tables, t_β is found to be 2.33. The mean and standard deviation of the residuals using the Monte Carlo simulation is now $m = 2.61$ and $v = 0.14$. Finally, the threshold value is found to be 2.9 by using equation (3.5).

²Standard temperature and pressure (informally abbreviated as STP) are temperature of 273.15 K (0°C, 32°F) and absolute pressure of 100 kPa (14.504 psi, 0.986 atm, 1 bar)

The thresholds of each dynamic neural network corresponding to the four engine variables are provided in Table 3.4. In the following subsections, three fault cases are considered in detail to demonstrate the performance of our developed detection and isolation scheme.

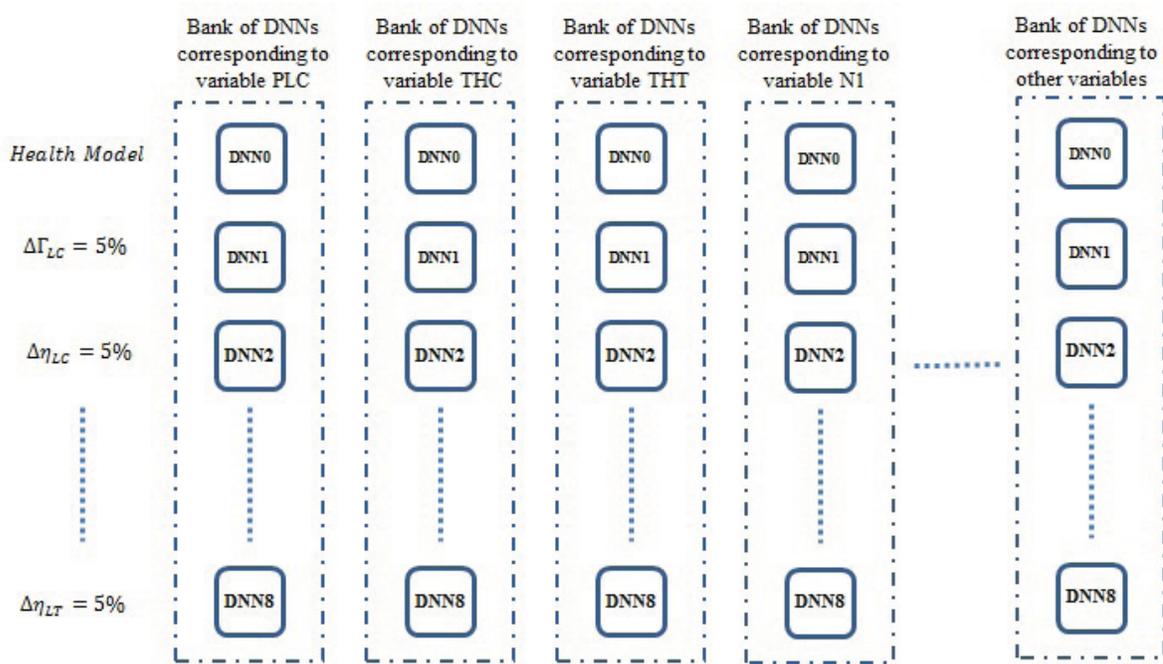


Figure 3.2: The multiple model architecture corresponding to the four specific or more bank of dynamic neural networks.

N1	THT	THC	PLC
0.051	0.097	0.094	0.164

Table 3.3: The noise standard deviations (as percentage of the nominal noise at the cruise condition).

Engine Variables	DNN0	DNN1	DNN2	DNN3	DNN4	DNN5	DNN6	DNN7	DNN8
PLC	0.014	0.03	0.03	0.01	0.015	0.009	0.015	0.09	0.015
THC	2.9	3.3	0.9	3.0	0.9	3.4	3.5	2.5	0.7
THT	8.0	7.2	3.5	8.6	2.7	6.7	5.7	0.1	6.5
N1	26.3	60.0	60.0	14.0	60.0	4.5	60.0	5.5	60.0

Table 3.4: Threshold values that are used for achieving fault detection and isolation.

3.2.3 Fault Detection Analysis Case Studies

Fault detection is conducted by using the model representing the nominal healthy operating condition namely, DNN0. The residuals for this model should not be greater than the threshold that is specified in Table 3.4 when the system components are healthy and should exceed it under faulty situations.

(a) **Fault Scenario $\Delta\Gamma_{LC}$.** A 5% $\Delta\Gamma_{LC}$ fault is injected at $t = 16$ seconds when the input fuel mass flow rate is at 70% of its maximum. Figure 3.3 shows the residuals that are generated by the $DNN0_{THC}$ (corresponding to the output measurement THC), $DNN0_{THT}$ (corresponding to output measurement THT) and $DNN0_{N1}$ (corresponding to the output measurement N1). It follows that the residual generated by $DNN0_{N1}$ is above the selected threshold, however, the residuals generated by $DNN0_{THC}$ and $DNN0_{THT}$ are not above their thresholds. This implies that among these three dynamic neural networks, the network corresponding to N1 is the only network that can detect the $\Delta\Gamma_{LC}$ fault with the selected severity level. The detection delay time for $DNN0_{N1}$ is 0.12 sec.

(b) **Fault Scenario $\Delta\eta_{LC}$.** The next scenario is a 5% $\Delta\eta_{LC}$ fault that is injected at $t = 16$ seconds when the input fuel mass flow rate is at 80% of its maximum. Figure 3.4

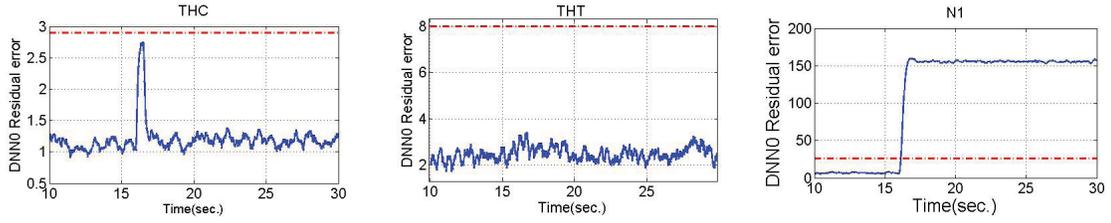


Figure 3.3: The DNN0 generated residuals corresponding to THC, THT and N_1 for the case of a $5\% \Delta\Gamma_{LC}$ fault.

shows the residuals that are generated by the dynamic neural networks. It follows that all the three generated residuals exceeded their thresholds so that this fault scenario can be detected with either of the three $DNN0_{PLC}$, $DNN0_{THT}$ and $DNN0_{N1}$ dynamic neural networks. The detection delay time for the networks are 0.05 sec, 0.05 sec and 0.02 sec, respectively.

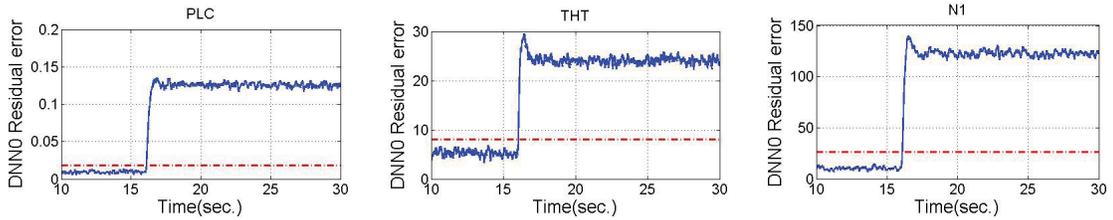


Figure 3.4: The DNN0 generated residuals corresponding to PLC, THT and N_1 for the case of a $5\% \Delta\eta_{LC}$ fault.

(c) **Fault Scenario $\Delta\Gamma_{HC}$.** The third faulty scenario corresponds to a $5\% \Delta\Gamma_{HC}$ fault that is injected at $t = 16$ seconds when the input fuel mass flow rate is at 85% of its maximum. Figure 3.8 shows the residuals that are generated by the dynamic neural networks $DNN0_{THC}$, $DNN0_{THT}$ and $DNN0_{N1}$. In this case the only residual that exceeds its threshold and remains above the threshold is the one corresponding to N_1 .

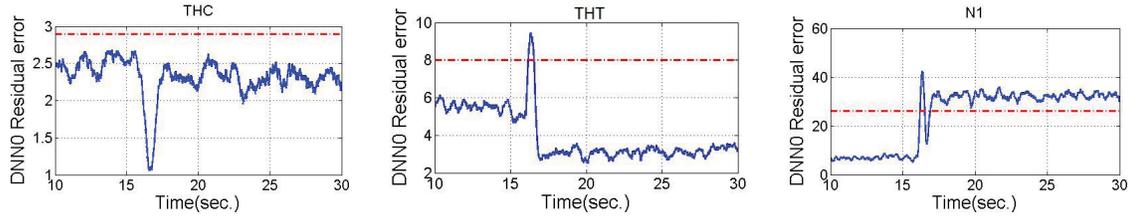


Figure 3.5: The DNN0 generated residuals corresponding to THC, THT and N_1 for the case of a 5% $\Delta\Gamma_{HC}$ fault.

3.2.4 Fault Isolation Case Studies Analysis

Single Fault Scenario

(a) **Fault Scenario $\Delta\Gamma_{LC}$.** A 5% fault is injected at $t = 16$ seconds when the input fuel mass flow rate is at 70% of its maximum. The DNN1 network is representative of this fault (refer to Table 3.1). It is expected that before the occurrence of the fault, all the residuals should be above their thresholds and following the occurrence of the fault the residuals generated by the DNN1 are below their thresholds. Other bank of dynamic neural networks should then generate residuals that are above their thresholds. Figure 3.6 shows the residuals of the eight (8) fault models DNN1 to DNN8. One can observe that this fault is isolated by the bank of dynamic neural networks corresponding to THT and N_1 networks. Note that only these DNN1 residuals are below their thresholds. Furthermore, the bank of dynamic neural networks belonging to THC is not capable of isolating this fault since the residual in this model is not below its threshold.

(b) **Fault Scenario $\Delta\eta_{LC}$.** The next fault scenario is a 5% $\Delta\eta_{LC}$ fault that is injected at

$t = 16$ seconds when the input fuel mass flow rate is at 80% of its maximum. The DNN2 network is representative of this fault model (refer to Table 3.1). Figure 3.7 shows the residuals corresponding to the fault models for this case. Considering the residuals that are obtained from the THT bank of dynamic neural networks one can conclude that this fault cannot be isolated by that bank of networks, since not only the DNN2 residual but also the DNN7 residual are below their thresholds. The same property also exists for the bank of dynamic neural networks corresponding to N_1 , since other than DNN2 in this bank of networks, DNN4 and DNN8 residuals are also below their thresholds. However, note that it is possible to isolate this fault by using the bank of dynamic neural networks corresponding to PLC, since only the DNN2 generates the residual that is below its threshold in this bank of dynamic neural networks.

(c) Fault Scenario $\Delta\Gamma_{HC}$. The third fault scenario corresponds to a 5% $\Delta\Gamma_{HC}$ fault that is injected at $t = 16$ seconds when the input fuel mass flow rate is at 85% of its maximum. The DNN3 dynamic neural network is representative of this fault. Figure 3.8 shows the fault model residuals for this case. In this case the fault can only be isolated by the N1 bank of dynamic neural networks. Similar to the previous case study, the THT bank of dynamic neural networks is not capable of isolating this fault since two of the residuals are below their thresholds after the occurrence of the fault. Also, the *THC* bank of dynamic neural networks cannot detect this fault since the DNN3 residuals corresponding to this bank of dynamic neural networks never exceed their thresholds.

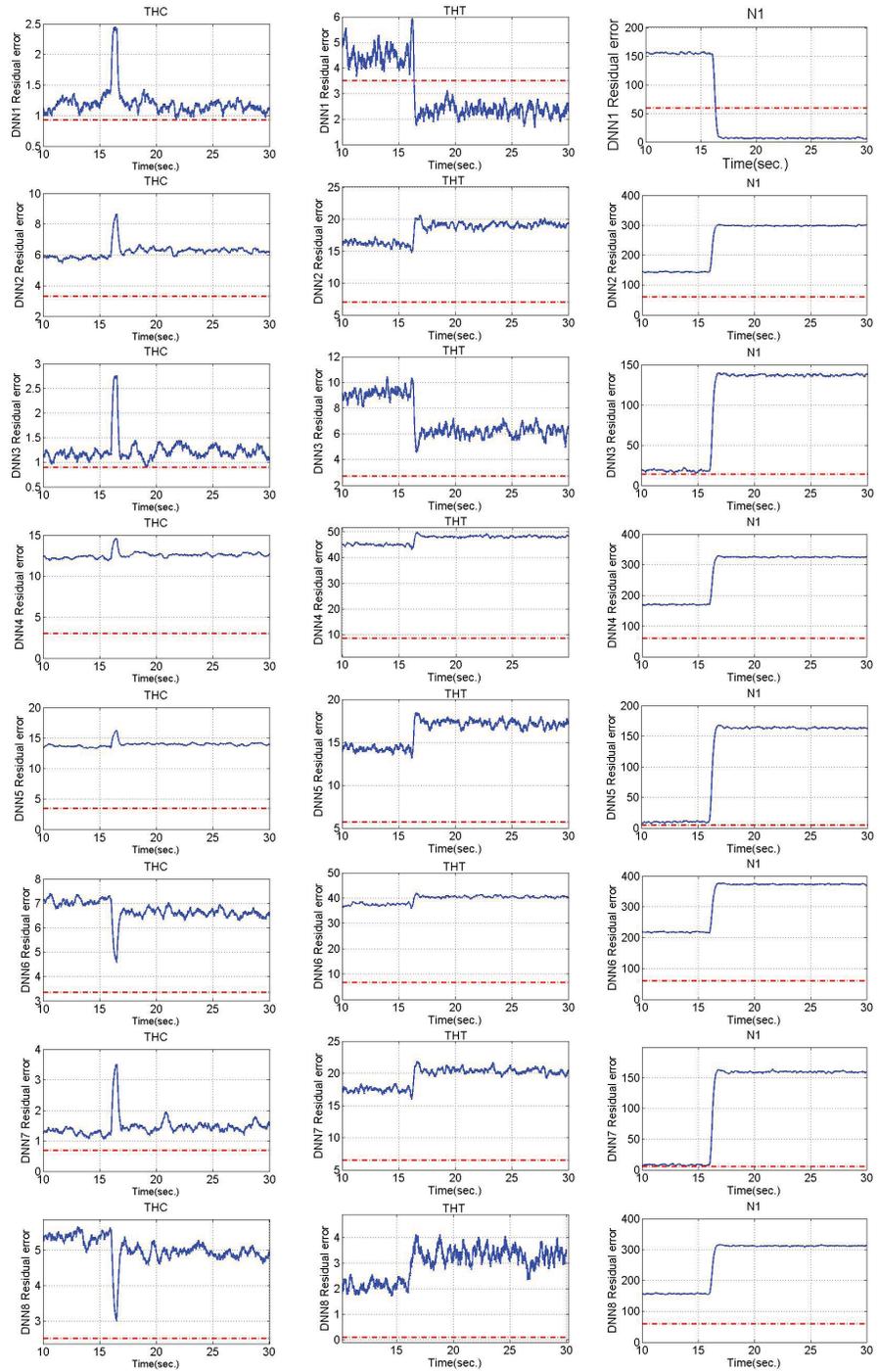


Figure 3.6: The DNN1 to DNN8 generated residuals corresponding to THC, THT and N_1 for the case of a $5\% \Delta\Gamma_{LC}$ fault.

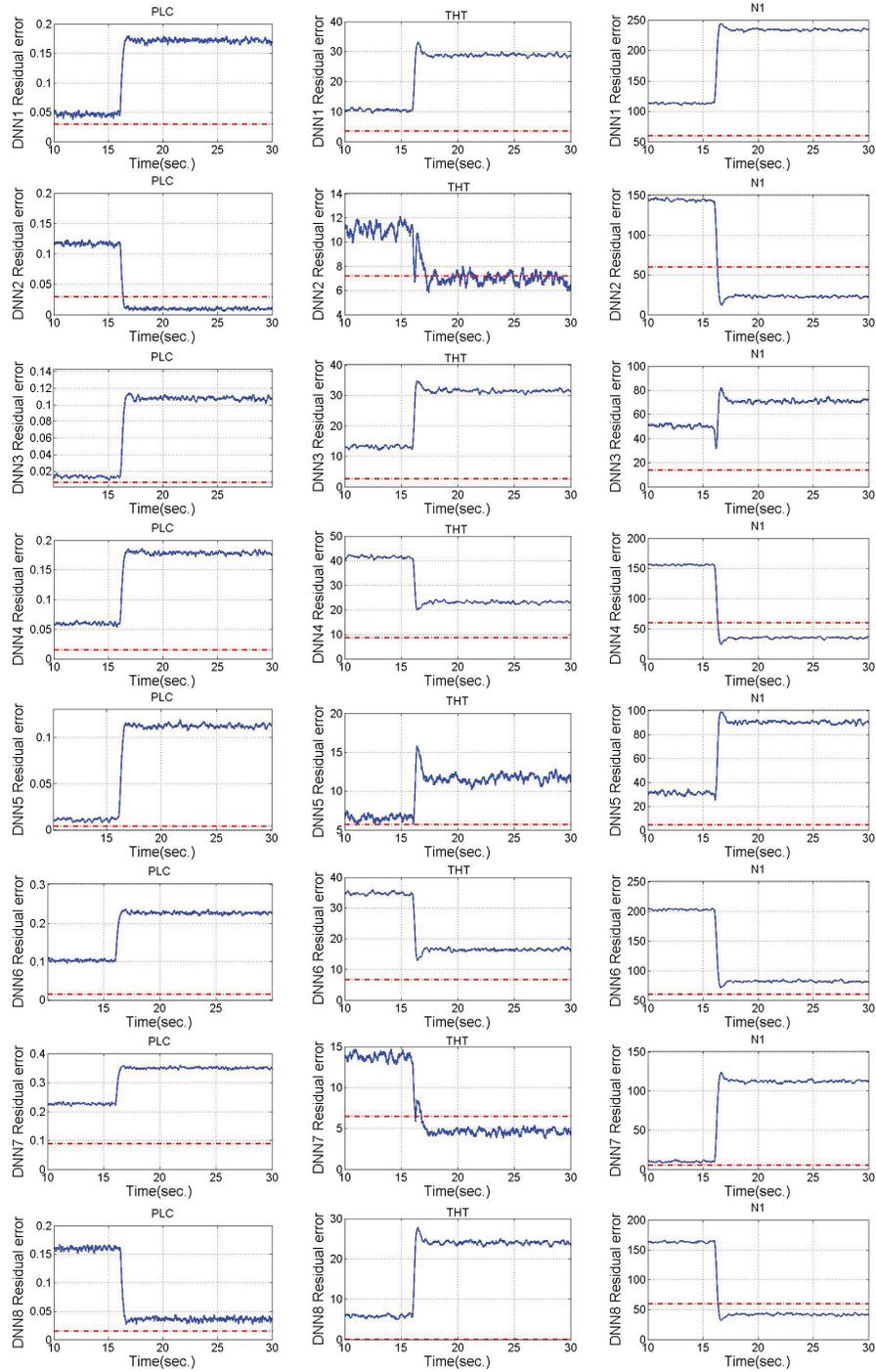


Figure 3.7: The DNN1 to DNN8 generated residuals corresponding to PLC, THT and N_1 for the case of a $5\% \Delta\eta_{LC}$ fault.

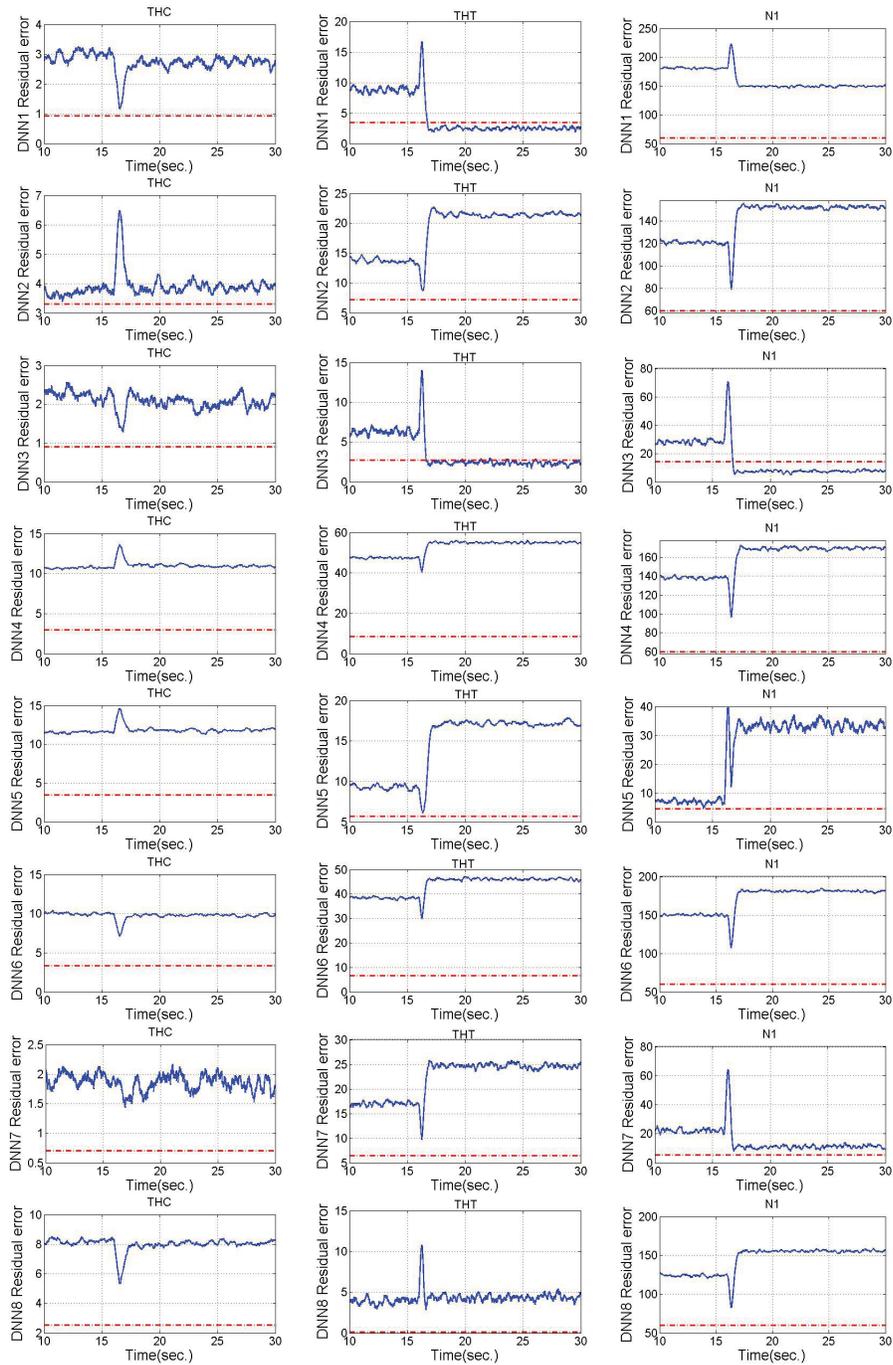


Figure 3.8: The DNN1 to DNN8 generated residuals corresponding to THC, THT and N_1 for the case of a $5\% \Delta\Gamma_{HC}$ fault.

A summary of the fault detection and fault isolation results are shown in Table 3.5 and Table 3.6, respectively. Table 3.5 shows the fault detection results using DNN0 for three different fault severities of 5%, 2% and 1%. Investigations have been conducted for six different input fuel flow rates (70%, 75%, 80%, 85%, 90% and 95% of its maximum). In this table if the fault can be detected in all the 6 input profiles with DNN0 bank of dynamic neural networks it is indicated by a \checkmark . As the percentage of the fault severity decreases, the ability of the network DNN0 in detecting the faults for all of the input profiles decreases. Indeed, for some fault scenarios DNN0 is only able to detect the fault in some specific input fuel mass flow rates as specified in Table 3.5.

Table 3.6 is prepared for two different fault severities. The results are obtained with the bank of dynamic neural networks that are *trained* with the 5% fault. One can conclude for the 5% faults that the fault $\Delta\Gamma_{HC}$ can only be isolated by the bank of dynamic neural networks corresponding to N_1 . The THC bank of dynamic neural networks can isolate the $\Delta\eta_{HT}$ and $\Delta\eta_{LC}$ faults and the THT bank of dynamic neural networks can isolate the $\Delta\eta_{HC}$, $\Delta\Gamma_{HT}$, $\Delta\eta_{HT}$ and $\Delta\Gamma_{LC}$ faults. Also, the PLC bank of dynamic neural networks can isolate the $\Delta\eta_{HT}$, $\Delta\Gamma_{LT}$, $\Delta\eta_{LT}$ and $\Delta\eta_{LC}$ faults. Therefore, considering all the banks of dynamic neural networks, all the eight (8) components faults can indeed be isolated.

However, concerning the 4% fault that the network *was not* trained for before, only few faults can be isolated by the banks of dynamic neural networks. This shows that the bank of trained networks with the 5% faults does not necessarily yield good generalization

Faults	5% Fault Severity				2% Fault Severity				1% Fault Severity			
	PLC	THC	THT	N1	PLC	THC	THT	N1	PLC	THC	THT	N1
$\Delta\Gamma_{LC}$	✓	-	-	✓	✓	-	-	✓	-	-	-	✓
$\Delta\eta_{LC}$	✓	✓	✓	✓	✓	○ ¹	✓	○ ³	✓	-	○ ¹	-
$\Delta\Gamma_{HC}$	-	-	-	✓	-	-	-	○ ⁴	-	-	-	-
$\Delta\eta_{HC}$	✓	✓	✓	✓	✓	✓	✓	○ ³	-	○ ²	✓	○ ³
$\Delta\Gamma_{HT}$	-	✓	✓	-	-	✓	✓	-	-	○ ²	-	-
$\Delta\eta_{HT}$	✓	✓	✓	✓	✓	-	✓	✓	✓	-	✓	-
$\Delta\Gamma_{LT}$	✓	-	✓	✓	✓	-	✓	-	✓	-	○ ¹	-
$\Delta\eta_{LT}$	✓	✓	-	✓	✓	-	-	✓	✓	-	-	-

Table 3.5: The fault detection results.

(- or ✓ denotes that one cannot detect or can detect a fault

- ¹ denotes that one cannot detect the fault if the input fuel mass flow rate is at 70, 75, and 95% of its maximum.
- ² denotes that one cannot detect the fault if the input fuel mass flow rate is at 70 or 95% of its maximum.
- ³ denotes that one cannot detect the fault if the input fuel mass flow rate is at 70, 75, and 80% of its maximum.
- ⁴ denotes that one cannot detect the fault if the input fuel mass flow rate is at 85, 90, and 95% of its maximum.)

performance for faults that are corresponding to the 4% severity, and hence for faults with different severities one would require other specially trained bank of dynamic neural networks.

3.2.5 Concurrent Component Fault Scenario

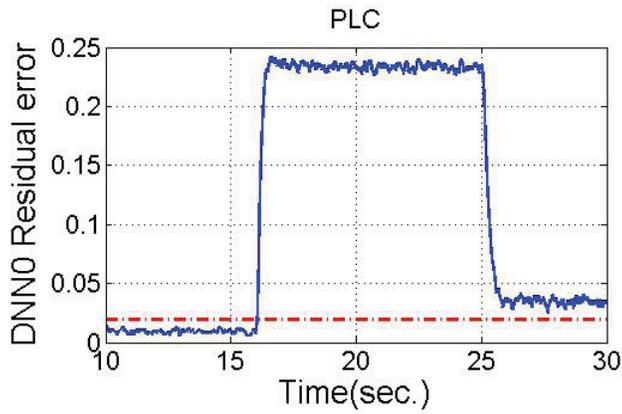
All the above results are obtained by assuming that faults do not occur exactly at the same time or concurrently. It is highly unlikely that two faults take place at the same time i.e. at each instant of time only one fault may occur in the system. However, the possibility of two concurrent faults is not unlikely. To evaluate the performance of the proposed FDI system for concurrent faults, we assume that two component faults namely $\Delta\Gamma_{LT}$ and $\Delta\eta_{LT}$ occurring

Faults	5% Fault Severity (networks trained for this fault severity)				4% Fault Severity (networks not trained for this fault severity)			
	PLC	THC	THT	N1	PLC	THC	THT	N1
$\Delta\Gamma_{LC}$	-	-	√	√	-	-	-	√
$\Delta\eta_{LC}$	√	√	-	-	√	-	-	-
$\Delta\Gamma_{HC}$	-	-	-	√	-	-	-	-
$\Delta\eta_{HC}$	-	-	√	-	-	-	-	-
$\Delta\Gamma_{HT}$	-	-	√	-	-	-	√	-
$\Delta\eta_{HT}$	√	√	√	-	-	√	-	-
$\Delta\Gamma_{LT}$	√	-	-	-	√	-	-	-
$\Delta\eta_{LT}$	√	-	-	-	-	-	-	-

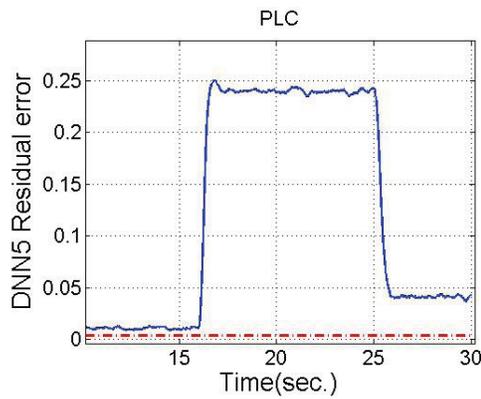
Table 3.6: The fault isolation results (- or √ denotes that one cannot isolate or can isolate a fault).

concurrently. DNN7 and DNN8 are representative of these faults, respectively. According to Table 3.6, $\Delta\Gamma_{LT}$ and $\Delta\eta_{LT}$ faults with 5% severities can only be isolated by the bank of DNNs corresponding to the variable PLC. Therefore, only the residuals corresponding to that bank of DNNs are shown in Figure 3.9. The DNN0 (healthy model) residual is under its threshold and the other DNNs residuals are above their thresholds before fault occurrence, which indicates the healthy operation of the engine. By occurrence of the first fault at t=16 seconds the DNN0 residual increases and exceeds its threshold and the fault is detected. The DNN7 residual decreases and stays below its threshold whereas other residuals corresponding to other faulty models still remain above their thresholds. Therefore, the fault is detected and isolated. However, once the second fault occurs at t=25 seconds the DNN7 residual exceeds its threshold whereas the DNN8 residual remains above its threshold since the first fault still

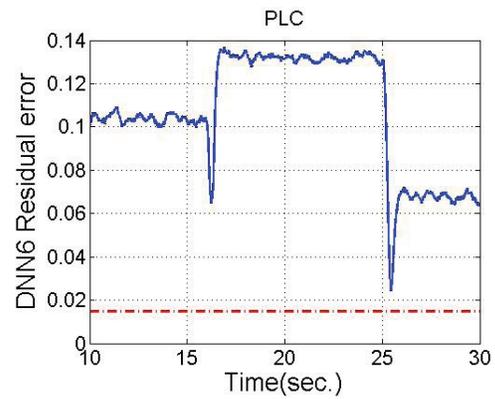
exists. The other DNNs (DNN1-DNN6) residuals are above their thresholds for those concurrent fault occurrence (DNN1-DNN4 are not shown in Figure 3.9 due to the similarity with DNN5 and DNN6). Hence, the only information that can be concluded is that the second fault has occurred but the fault cannot be isolated. For the detection and isolation of two concurrent faults, a hierarchical approach can be employed [7] which is not considered in this thesis and can be treated as future work.



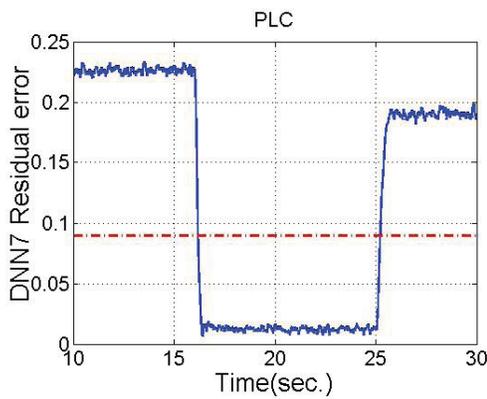
(a) The DNN0 residuals.



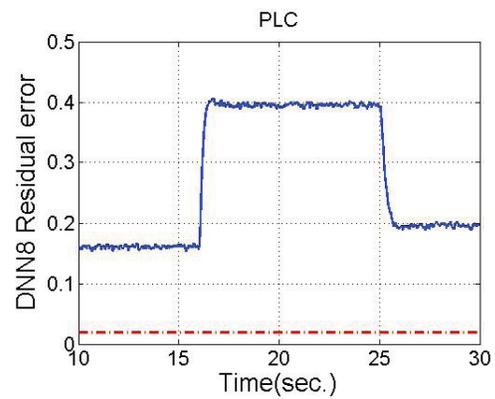
(b) The DNN5 residuals.



(c) The DNN6 residuals.



(d) The DNN7 residuals.



(e) The DNN8 residuals.

Figure 3.9: Concurrent faults ($5\% \Delta\Gamma_{LT}$ fault injected at $t=16$ seconds and the $5\% \Delta\eta_{LT}$ fault injected at $t=25$ seconds.)

3.2.6 Conclusions

In this chapter, a dynamic neural network-based multiple model scheme is proposed for fault diagnosis of aircraft jet engines. Several banks of dynamic neural networks are trained where each network corresponds to a specific faulty and/or healthy mode of the aircraft jet engine. The presented simulations demonstrate the effectiveness of the proposed strategy. Our fault detection and isolation results are summarized in Tables 3.5 and 3.6, respectively. In the next chapters we develop fault diagnosis schemes for sensor faults and for the cases when the system is subjected to higher levels of noise and presence of outliers.

Chapter 4

Data Validation Using Autoassociative Neural Network

4.1 Introduction

Performance and success of fault detection and isolation systems to detect and isolate component faults are mainly dependent on the validity and quality of the measurement data. All the measurements which are obtained from sensors in the gas turbine are subject to sensor noise, biases, drifts and other sensor faults. Such sensor faults and anomalies cause deviations from real values and can result in poor fault diagnosis. An autoassociative neural network, introduced first time by Kramer [77] is a useful neural network to perform data validation. It is practical for both filtering or signal smoothing as well as sensor error correction. It is therefore

a helpful technique to improve fault detection and isolation task and also to enhance reliability and robustness of the diagnostic system. As described in Section 2.2 in Chapter 2 such a neural network is feed forward with a symmetrical topology structure which is constructed to make the outputs the same as the inputs, and has a unique capability of characterizing the data dependency of the input data.

It will also be shown in the next chapter that autoassociative neural networks (ANN) has great capabilities for both sensor fault and component fault detection and isolation. As described in Section 2.2 on Chapter 2, the ANN architecture (Figure 4.1) consists of two parts, the mapping layer and the de-mapping layer. These layers are interconnected through the bottleneck layer which is the most important layer in the ANN. The bottleneck layer compresses the data into low dimensional representation, eliminates redundancies and extracts principal components in the output data.

The faults in the gas turbine engine can occur during the operation of the gas turbine. Due to the fact that faults affect performance and life of the gas turbine, it is necessary to diagnose and correct them. However, it is important to note that in addition to component faults, measurement noise and sensor biases are other sources of parameter changes in the gas path of a gas turbine. In order to avoid false alarms it is practically important to validate the data that are received from the sensors. Otherwise, noise, bias, or other sensor faults contained in the measurement data would easily be mistaken for engine components faults, resulting in misjudgement in the diagnosis of faults. Such a misjudgement can cause lots of financial loss

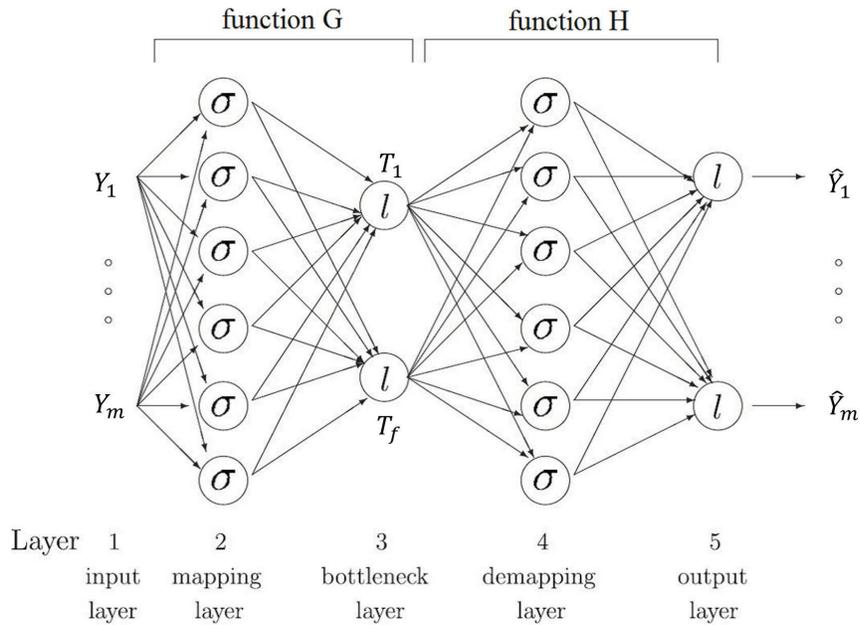


Figure 4.1: Architecture of autoassociative neural network where σ denotes sigmoidal nodes and l denotes linear nodes

for mistakenly changing a part in a good health instead of changing a simple sensor that might have less cost.

4.2 Autoassociative Neural Networks for Data Validation and Faulty Sensor Correction

Autoassociative neural network is an efficient method used to process the measurements or sensor data before performing fault detection and isolation. Sensors in a gas turbine are practically utilized in harsh operating environments and situations such as high pressures and high temperatures which generate high levels of noise or other undesirable effects that make the

fault diagnosis problem a challenging issue. Autoassociative neural networks show a significant robustness in presence of noise and sensor faults. In this chapter the autoassociative neural network is utilized for the following tasks in aircraft jet engine, namely *noise reduction*, *inaccurate sensor correction*, *filtering outliers* and *sensor error correction*.

4.2.1 Network Training

During the training process the weights would be updated by using backpropagation algorithm so that the outputs are the same as inputs. The training data and also the number of neurons in the bottleneck layer must be selected properly, so that the internal representation that is developed by the network by assigning the weights, retains the maximum possible amount of information.

The weight updating process should be accomplished by starting from a small number of neurons in the mapping, the bottleneck and the de-mapping layers and gradually the neurons are increased in these layers, however one should note that the number of neurons in the bottleneck layer is less than the mapping and the de-mapping layers and the mapping and the de-mapping layers have the same number of neurons. The important key to have an efficient ANN for data validation is to use the appropriate input data to provide a proper amount of information to the network for learning and also the most important factor is the number of the bottleneck neurons. A network with a better noise filtering and lower input-output error should be selected as the suitable network for use in the data validation problem.

The weights and biases are updated iteratively until the overall network mean square error is minimized and the output approximates the input as closely as possible. This assures that the internal representation that is constructed by the network retains the maximum possible amount of information from the original data set for a given degree of dimensional compression that is represented by the bottleneck layer.

In this thesis, a network has been trained with the data that are generated from a Matlab Simulation Model, however it is possible to retrain the network with the data from a real engine with noise. Using the data with some level of noise as the training data causes a slow learning and higher training error and less noise reduction. Knowing the fact that there are usually accurate simulators, it might be more efficient to first use the data that is generated from a simulator to find the suitable structure of the network and then retrain the network with the real engine data to adjust the weights accordingly. This procedure helps the network to match accordingly with that specific engine properties. However, the data from the real engine may contain noise. Therefore, it should be noted that the retraining phase must be terminated before the error function is actually minimized to avoid poor network generalization. In other words, in case of using the noisy data, the retraining time must be sufficiently long to provide an adequate fit and not to be too long to fit and memorize the noise, therefore a trade off is required to be considered.

Noise present in the data that is imposed to the input nodes is filtered in the mapping layer before the bottleneck layer. The reconstructed data are generated in the subsequent

demapping layers after the bottleneck layer. The percentage of noise that is removed depends on the redundancy of the input variables. For removing bias the same procedure as in the noise filtering can be applied, however in some cases the network may not be able to achieve the task properly. Under these cases a robust autoassociative neural network (RANN) is introduced to improve the ANN performance [77]. Using RANN one may need to train the network in two steps in order to be able to perform both noise filtering and sensor fault correction. Therefore, the training procedures contain the following steps. First, in order to improve the training capability of the network, the generated data from sensors must be normalized to have values between -1 and 1. Second, by using the clean and uncorrupted data (no sensor faults), variety of networks with different architectures are trained among which the best network is selected according to the best input estimation and noise reduction capabilities.

After selecting the network structure if the ANN performance is poor in bias correction then as part of the third step, retraining is required to force the network to produce uncorrupted output values for inputs containing bias or drift errors. To accomplish this, the input data must be from faulty sensor data but the output data correspond to the corrected ones. Considering the role of the two sub-networks of the ANN, the noise is filtered in the first sub-network containing the mapping layer and the bottleneck layer. This sub-network has the role of compressing the dimension of the inputs where the redundancies and random variations due to measurement noise are removed during the space dimensional compression of the inputs. On the other hand, the role of the second sub-network is to re-transform the compressed and

filtered data to its original dimension.

Based on this fact, during the retraining process it is helpful to fix the weights of the post-bottleneck layers (second sub-network) and let the training process to only update the weights in the first sub-network [77] where the noise filtering is accomplished. This implies that during the retraining process only the weights of two layers need to be updated.

4.3 Simulation Results

4.3.1 Training of the Autoassociative Neural Networks

The sensor measurements that are used in the following study are TLC, PLC, THC, PHC, TLT, PLT, N1, N2 and TCC. Another input to the network is the engine fuel flow rate (W_f) which is not normally directly measured and is the input of the gas path system. It can be obtained by using the power level angle (PLA) which is set by the pilot (details are provided in Chapter 2).

The fuel flow rate in this work can change from 70% to 98% as it is almost the range for the cruise mode. The ANN structure that is used for the data validation of a dual spool engine is shown in Figure 4.2. The training data is collected from the simulation model of the jet engine. Both input and output data are normalized by using the min-max normalization to be scaled in the range of -1 to 1. The number of neurons in each layer should be found properly for the network training. Starting with a relatively small number of neurons for the three hidden layers, an optimal architecture is developed for our proposed ANN. During

the optimization processes (that is finding the optimized number of neurons for each layer) the number of the bottleneck layer neurons are changed from 3 to 7 and the number of the mapping and the de-mapping layer neurons are changed from 10 to 60 neurons in order to obtain the best structure of the ANN. The number of the mapping and the demapping neurons for all the architectures considered are the same.

Using the batch training method a total of 7000 data are utilized for each ANN input in the training and the number of epochs vary from 30 to 60 iterations. The data are generated for different engine input fuel mass flow rates changing from 70% to 98% of the maximum fuel mass flow rates (approximate range of the fuel flow in the cruise condition). The ambient conditions are set to standard condition.

Among all the evaluated architectures, the results of nine (9) architectures are shown in Table 4.1. The table summarizes the architecture, the training error J_{Train} , test error J_{Test} and the percentage of the reduced noise variance which is the average of all the output measurements for different engine's input profiles. The noise level was chosen to be the nominal noise at the cruise conditions. The nominal value of the noise levels are shown in Table 3.3 in Chapter 3. According to Table 4.1 the maximum noise reduction belongs to the structure #4, however such a structure has a high training and testing error (J_{Train} and J_{Test}) implying that the network has not learned the input-output relationship as required. The next higher percentage of the noise reduction is related to the structure #6 with a training and testing errors of 2.15 and 2.33, respectively. The structure #2 has both the lower training and the testing errors

but such low errors imply that the network has captured somehow the random aspect or the noisy behaviour of the data. This is due to the fact that the percentage of the noise that such a network can reduce is low.

In general the structures with 6 bottleneck neurons has shown better noise reduction capabilities. Among the ANNs with 6 bottleneck layer we select #6 with the structure of 10-32-6-32-10 which has the average of 68.49% noise filtering capability.

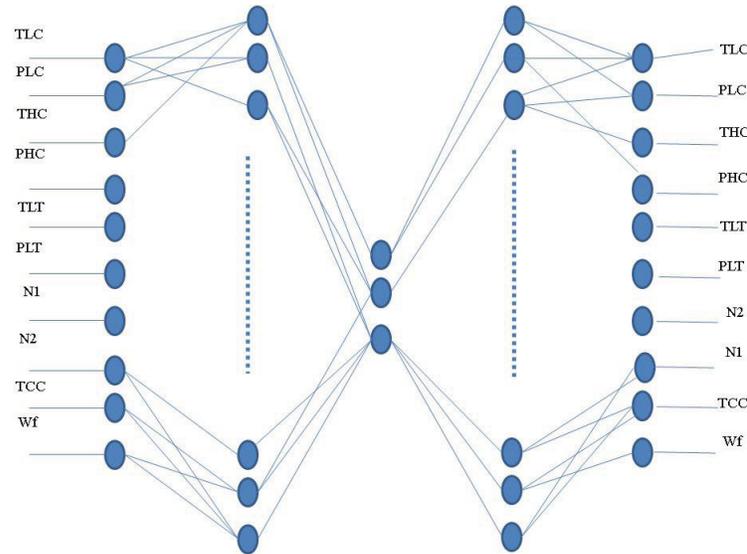


Figure 4.2: ANN structure for the gas turbine engine sensor validation.

The simulation results are presented in the following to evaluate the performance of ANN for data validation including noise reduction, inaccurate sensor correction, removing outliers and sensor error correction.

#	Structure	J_{Train}	J_{Test}	Reduced noise level (%)
1	10-29-4-29-10	2.46	2.59	8.5%
2	10-35-4-35-10	1.78	1.98	45.44%
3	10-47-5-47-10	2.13	3.47	33.06%
4	10-35-5-35-10	10.21	13.82	84.78%
5	10-30-6-30-10	2.098	2.04	59.51%
6	10-32-6-32-10	2.15	2.33	68.49%
7	10-41-6-41-10	1.91	1.97	38.14%
8	10-34-7-34-10	2.12	2.14	32%
9	10-49-7-49-10	3.55	3.82	16%

Table 4.1: Trained autoassociative neural networks.

Noise Reduction

Noise filtering properties of the autoassociative network depend on how much the network learns the interrelation among variables in its mapping and bottleneck layers. Therefore, it excludes random variations due to measurement noise in the bottleneck output, and after the de-mapping layer the network will yield "clean" corrected data. However, the level of noise that is removed from data depends on the level of the redundancy among the measurement data that is used in the ANN inputs.

	TLC	THC	THT	N1	N2	TCC	PLC	PHC	PHT
ANN Inputs Noise Level	0.49	0.17	0.28	0.18	0.14	0.19	0.31	0.33	0.21
ANN Output Noise Level	0.08	0.06	0.03	0.08	0.04	0.027	0.15	0.16	0.17
Percentage of the Noise Filtered	84%	65%	89.5%	53%	70%	86%	60%	50.26%	20%

Table 4.2: The percentage of the noise filtered by the ANN.

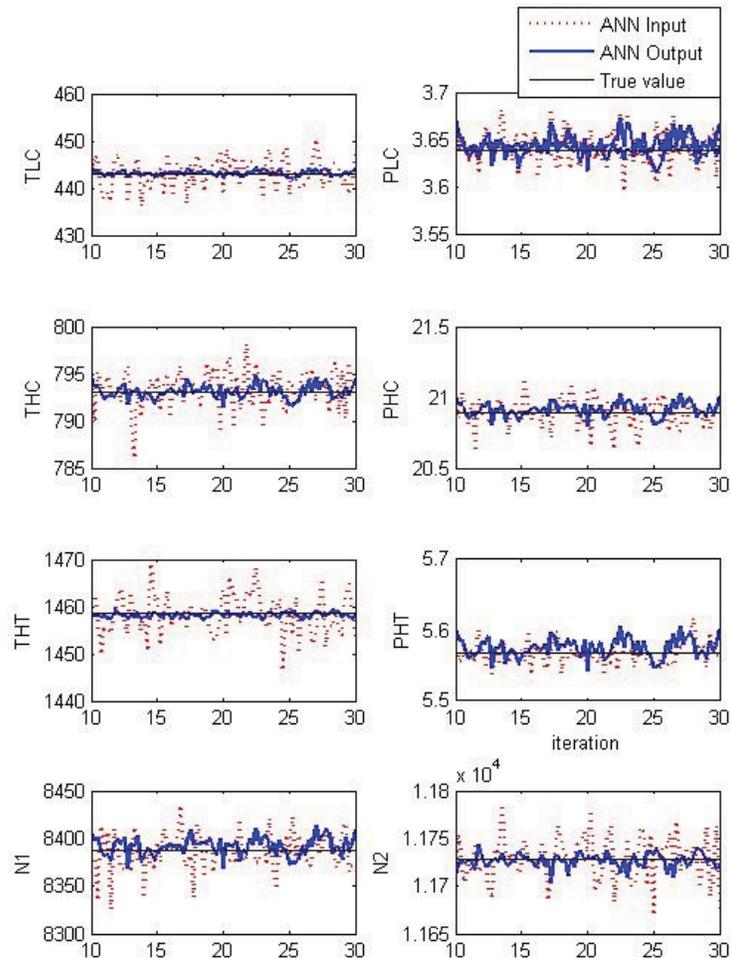


Figure 4.3: Input and the output of the autoassociative neural network for noise filtering.

The capability of the autoassociative neural network with the structure of 10-32-6-32-10 in filtering noise for each input variable is presented in Table 4.2. The noise levels are defined in the form of percentage of the standard deviations at typical cruise condition. The ANN has shown a significant capability in removing noise for most of the variables.

Figure 4.3 shows the noise filtering capability of the above ANN for the case when the fuel flow rate is at 85% of the maximum fuel flow. The ANN output is the filtered version of

the input data with a significant percentage of the noise reduction. In this case the nominal values for the above cruise condition for TLC, THC, THT, N1, N2, TCC, PLC, PHC, PHT are respectively at 443 K, 793 K, 1457 K, 8390 rpm, 11725 rpm, 1458 K, 3.64 bar, 20.9 bar and 5.57 bar.

Inaccurate Sensor Correction

The ANN is also capable of correcting the data from inaccurate sensors. Accuracy in sensors is the ability of a sensor measurement to match the actual value of the quantity being measured. Clearly, sensor accuracy is essential for success and reliability of FDI systems, However, over the life time of sensors, they may encounter loss of accuracy.

In order to evaluate the performance of ANN for this type of sensor fault, sensor inaccuracy is modelled by increasing the level of the noise variance. Figure 4.4 shows the results when the sensor noise variances are increasing linearly from the nominal sensor variance to the variance which is ten (10) times the nominal sensor noise variances. In this case, Table 4.3 shows the percentage of the noise filtered using ANN. The percentages are obtained through conducting multiple Monte Carlo simulations corresponding to random noise and by considering the deducted value of the noise root mean square (rms) in the ANN output.

Removing Outlier using the ANN

One of the major problems in gas turbine engines is the presence of outliers that may lead to false alarms in the engine fault detection system. Such noise outliers need to be removed

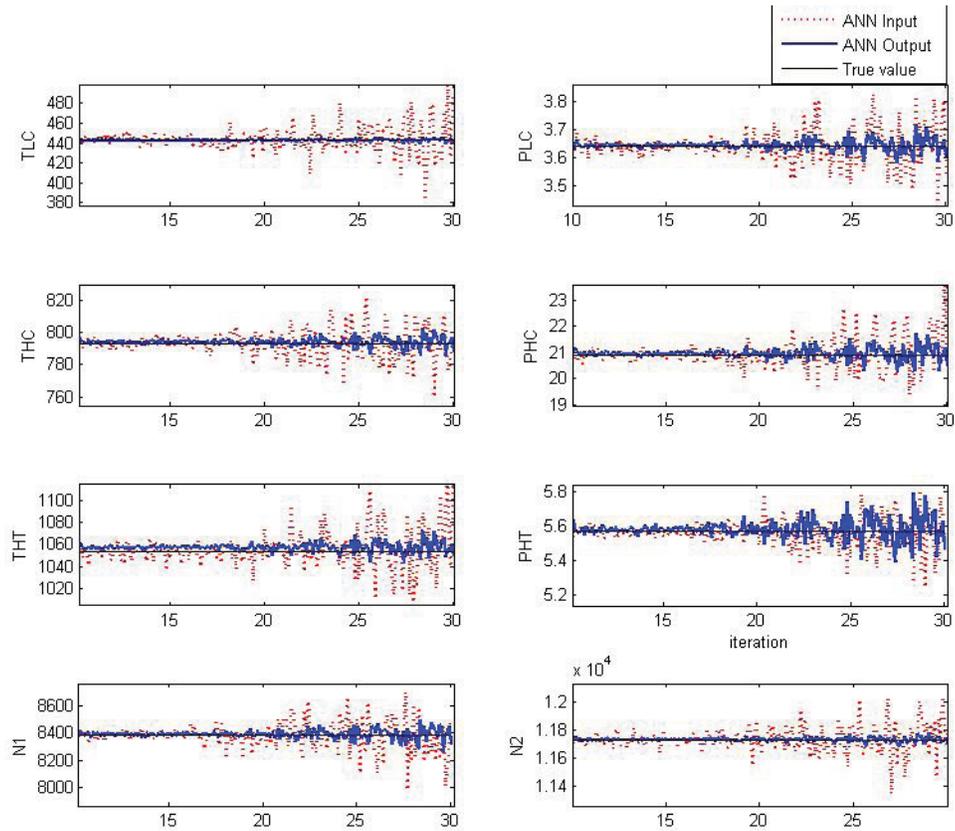


Figure 4.4: Input and the output of ANN corresponding to the recovery of sensor inaccuracies.

	TLC	THC	THT	N1	N2	TCC	PLC	PHC	PHT
ANN Inputs percentage of noise rms at nominal value	2.6	0.95	1.07	0.95	0.7	0.69	1.65	1.9	1.43
ANN Output percentage of noise rms at nominal value	0.2	0.3	0.26	0.37	0.11	0.25	0.49	0.8	0.96
Percentage of the noise filtered	92%	68%	76%	65%	84%	63%	71%	56%	33%

Table 4.3: The percentage of the noise filtered by the ANN.

before conducting engine fault detection. As typical examples, random outliers are injected to the temperature of the high pressure turbine, high pressure rotational speed and pressure of the low pressure compressor with the level of 1% for rotational speed and 10% for other measurements. The results are shown in Figures 4.5, 4.6 and 4.7 which indicate that the proposed data validation approach using the ANN is capable of removing the outliers that may have occurred on the temperature and rotational speed sensors completely. However, the ANN can only remove the significant amount of outlier for the PLC. Table 4.4 shows the results of applying ANN with presence of outliers for different sensor measurements. The outliers are selected at 1% level of the nominal values of N1 and N2 measurements and at 10% level of the nominal values for other measurement signals. In this table \checkmark denotes the complete removal of the outlier. Corresponding to the PLC and PHT measurements, the ANN is capable of removing 86% and 47% of the outliers, respectively.

Outlier removal capability of ANN								
TLC	THC	THT	N1	N2	TCC	PLC	PHC	PHT
\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	86%	\checkmark	47%

Table 4.4: Outlier removing by the ANN.

Error Correction

Besides filtering the noise, the ANN has the capability to replace the faulty data due to the sensor faults such as biases and drifts with estimated true data. The ANN is also useful in the sense that if one measurement is lost, the lost measurement can be replaced with an estimate

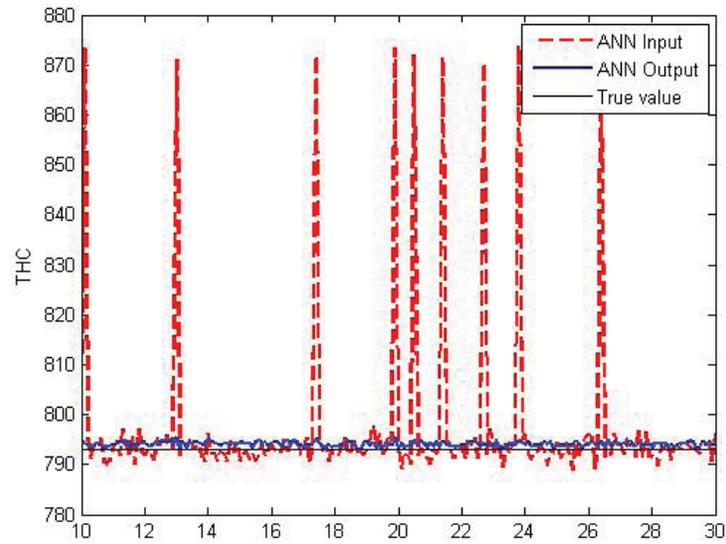


Figure 4.5: Input and the output of the ANN corresponding to the THC measurement signal outlier noise.

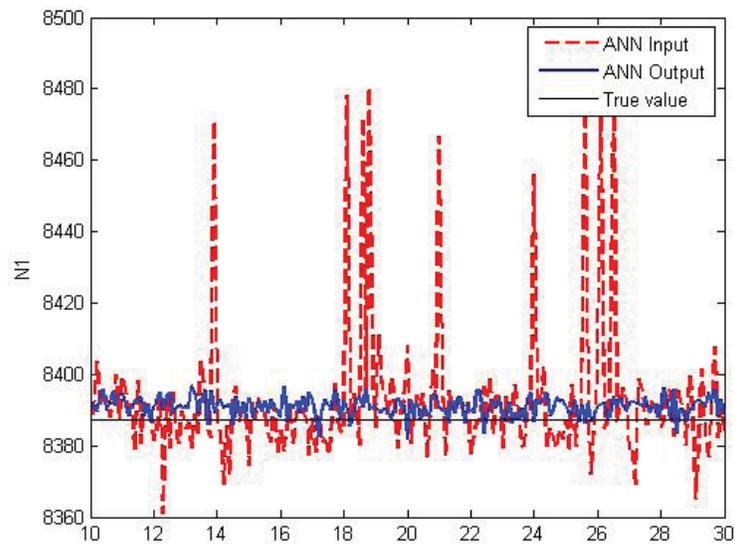


Figure 4.6: Input and the output of the ANN corresponding to the N1 measurement signal outlier noise.

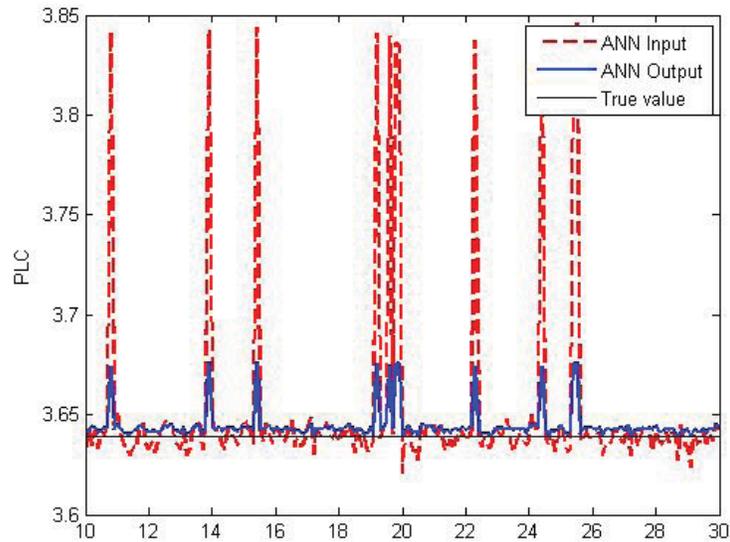


Figure 4.7: Input and the output of the ANN corresponding to the PLC measurement signal outlier noise.

from the remaining valid sensors. However, sometimes the ANN which was trained using valid data acts poorly in correcting the sensor faults for certain input variables. In such a case as proposed in [77] by using robust autoassociative neural networks the training set should be modified by including false data. The network is then retrained with the data sets that include sensor fault data in order to learn how to filter the false information. It is also recommended to adjust the weights after the bottleneck layer during the retraining and only update the weights before the bottleneck layer.

In this section, the objective is to evaluate how efficiently the ANN corrects the corrupted data and also how much other variables that are from the healthy sensors are robust to the changes in the faulty variable. Several simulations have been performed in this section.

Figures 4.8 to 4.15 show the results of our evaluations. These figures illustrate the recovery rate of the faulty sensor data as well as the percentage of the deviations for other healthy variables from their real values when there are different biases ranging from 1% to 100%. The recovery rates and the deviation rates is calculated according to [81]

$$Recovery\ Rate = \left(1 - \frac{|Y_{ANN} - Y_{Target}|}{|Y_{Target}|}\right) \times 100 \quad (4.1)$$

$$Deviation\ Rate = \left(\frac{|Y_{ANN} - Y_{Target}|}{|Y_{Target}|}\right) \times 100 \quad (4.2)$$

For instance, in Figure 4.8 the top plot is the recovery rate corresponding to the sensor TLC while the bias was increased from 1% to 100%. The bottom plot shows the deviation rate of the other healthy sensors in case the TLC sensor bias is changed from 1% to 100%. The same plots are presented for the other sensors. The results show a significant recovery for the temperature and speed measurements, and acceptable percentage of recovery for the pressure measurement. From the shown figures it might be concluded that for the low biases the performance is smaller than the large biases, however one should note that as the percentage of the biases increase, the deviation rate of the other healthy measurements increase undesirably. The biases on the variables THC, PHC, THT, N1 and N2 have less effect and produce smaller deviation on the other healthy output measurements which are from the healthy sensors.

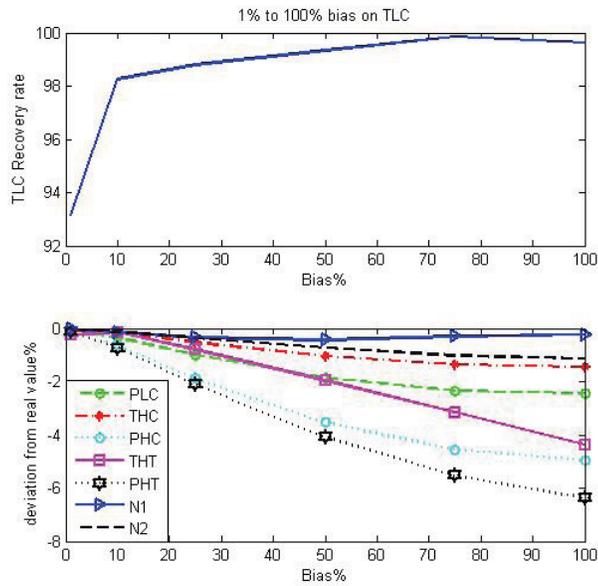


Figure 4.8: The sensor recovery rate and the deviation rate corresponding to TLC.

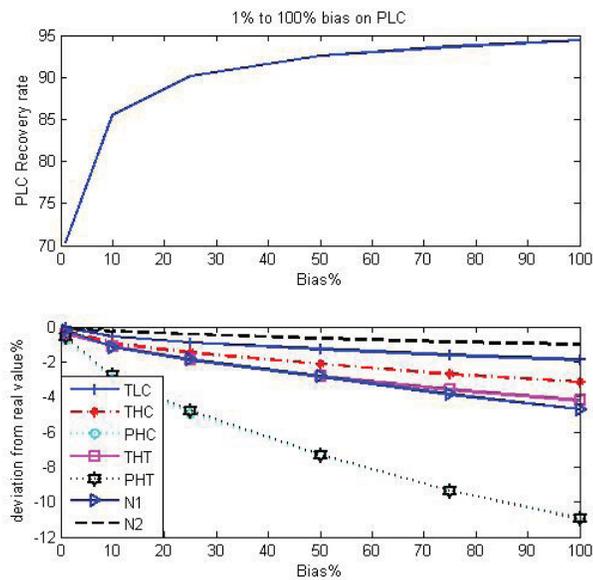


Figure 4.9: The sensor recovery rate and the deviation rate corresponding to PLC.

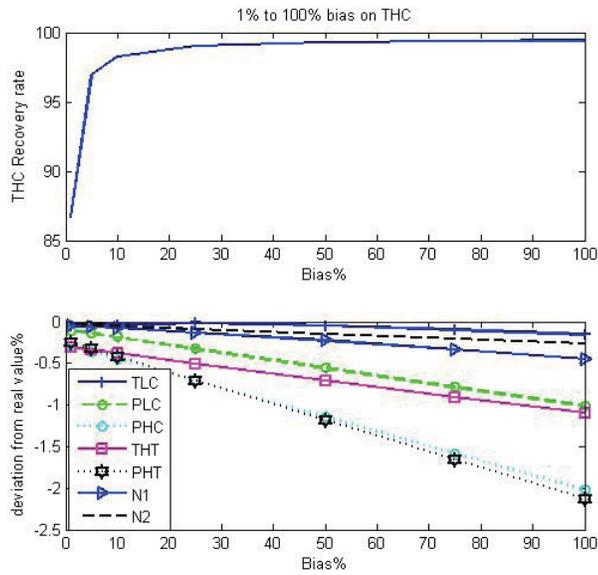


Figure 4.10: The sensor recovery rate and the deviation rate corresponding to THC.

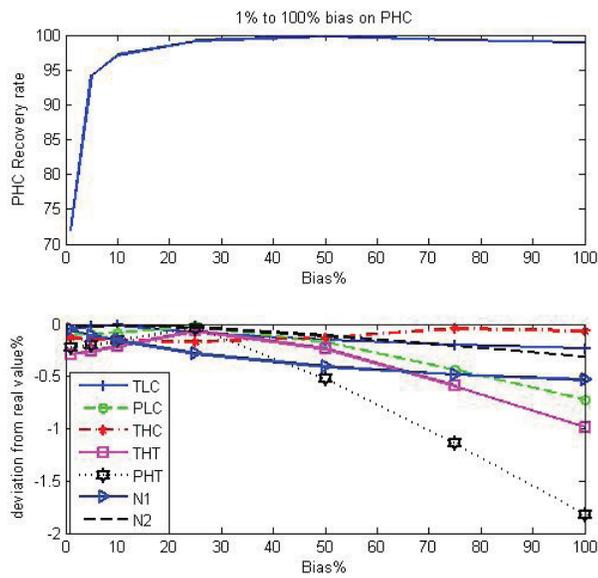


Figure 4.11: The sensor recovery rate and the deviation rate corresponding to PHC.

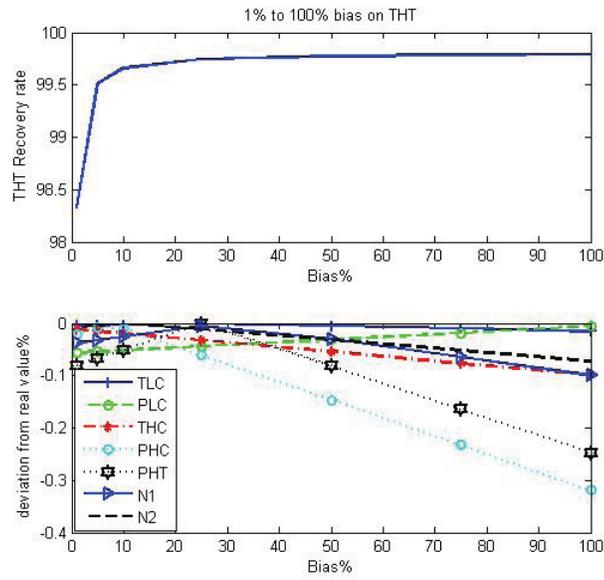


Figure 4.12: The sensor recovery rate and the deviation rate corresponding to THT.

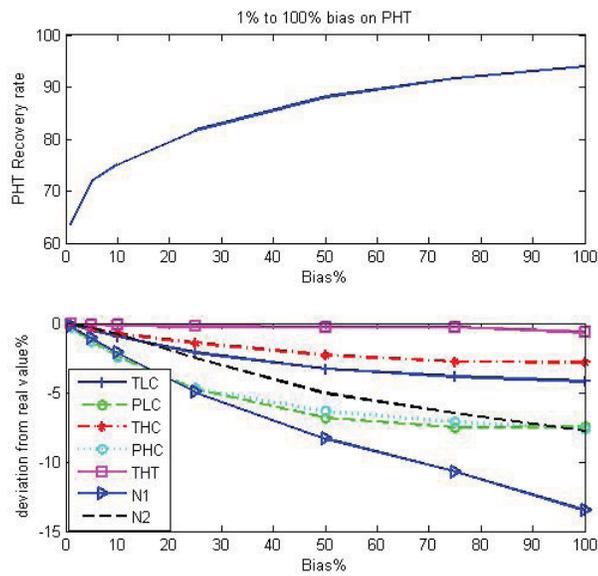


Figure 4.13: The sensor recovery rate and the deviation rate corresponding to PHT.

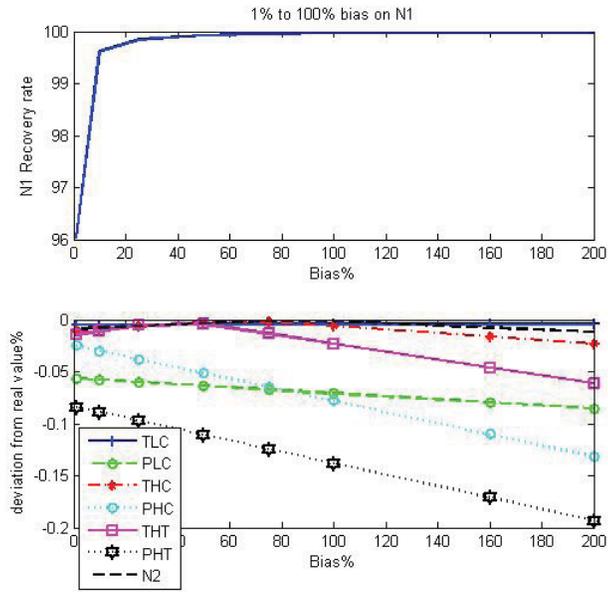


Figure 4.14: The sensor recovery rate and the deviation rate corresponding to N1.

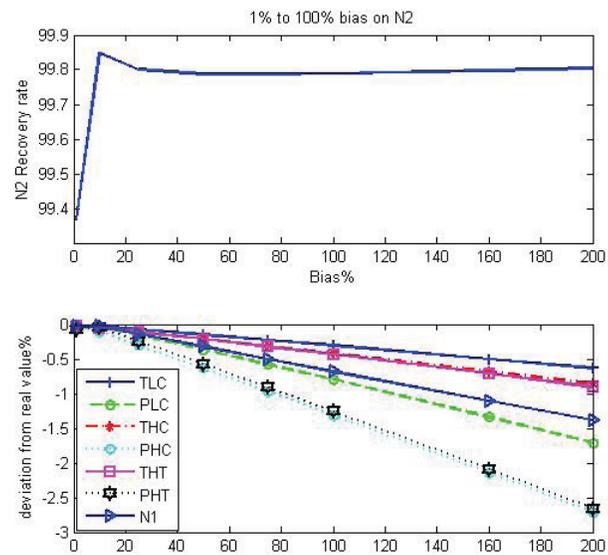
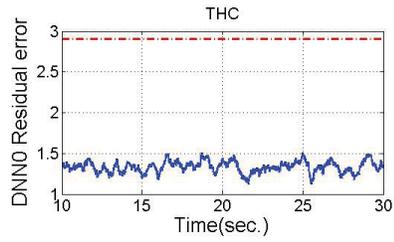


Figure 4.15: The sensor recovery rate and the deviation rate corresponding to N2.

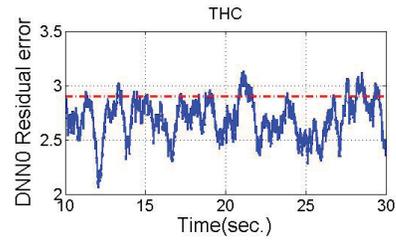
4.3.2 Effects of the None-Validated Data on Engine Fault Detection

In this section, we will examine the effects of the proposed data validation system on the engine fault diagnosis system. Sensor measurements will first be validated by the noise filtering and error correction using the proposed ANN. Then the obtained validated data from the ANN outputs will be used for fault diagnosis by using dynamic neural networks as discussed in Chapter 3. Figures 4.16 and 4.17 show the effects of the noise filtering on the performance of the dynamic neural networks when the system is in the healthy mode but the measurements have noise with twice the nominal noise level in the cruise condition.

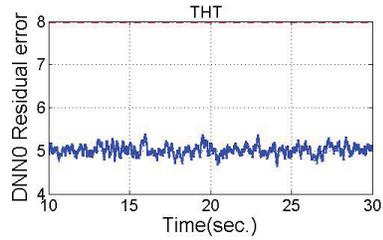
Figure 4.16 is related to the case when the input fuel flow rate is at 85% of its maximum and Figure 4.17 is related to the scenario when the input fuel flow rate is at 75% of its maximum. The DNN0 is a neural network which is the representative of the healthy operation of the gas turbine engine and is used for fault detection. Figures 4.16(a), 4.16(c) and 4.16(e) show the residuals corresponding to the variables THC, TCC and N1 that are obtained from DNN0 subject to the validated data that are generated from the autoassociative neural network. On the other hand, Figures 4.16(b), 4.16(d) and 4.16(f) show the residuals of the same DNN0 which use the non-validated data. It is obvious that false alarms would be produced in this case since the residuals exceed their thresholds at some points. Similar case exist for another scenario that is shown in Figure 4.17.



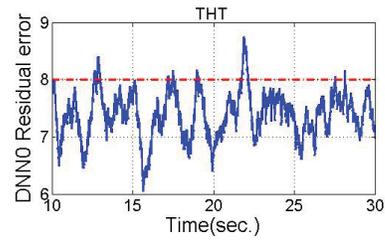
(a) DNN0 residuals corresponding to THC with validated data for the healthy operation of the engine.



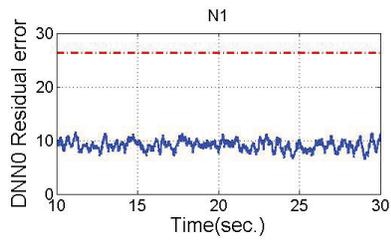
(b) DNN0 residuals corresponding to THC with non-validated data for the healthy operation of the engine



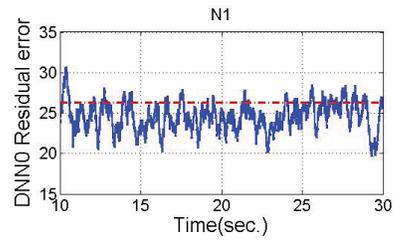
(c) DNN0 residuals corresponding to TCC with validated data for the healthy operation of the engine.



(d) DNN0 residuals corresponding to TCC with non-validated data for the healthy operation of the engine.

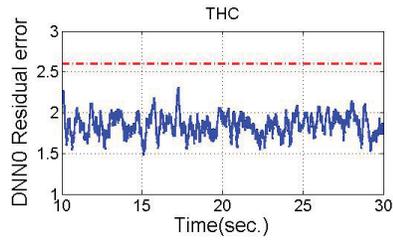


(e) DNN0 residuals corresponding to N1 with validated data for the healthy operation of the engine.

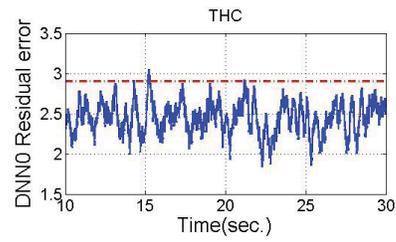


(f) DNN0 residuals corresponding to N1 with non-validated data for the healthy operation of the engine.

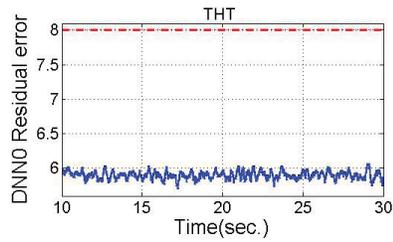
Figure 4.16: Effect of the non-validated data that produce false alarms (when input fuel flow rate is at 85% of its maximum).



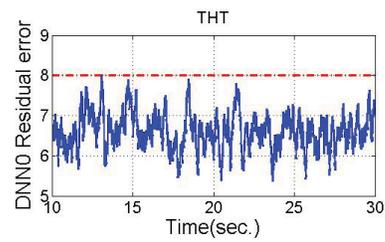
(a) DNN0 residuals corresponding to THC with validated data for the healthy operation of the engine.



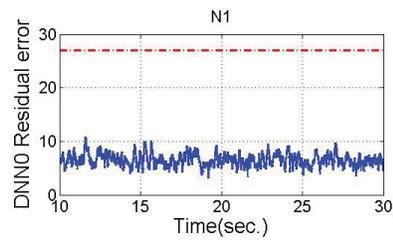
(b) DNN0 residuals corresponding to THC with non-validated data for the healthy operation of the engine.



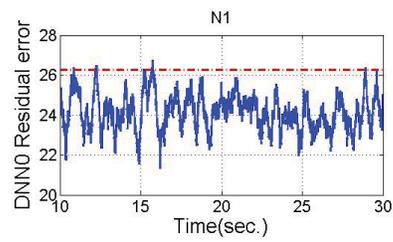
(c) DNN0 residuals corresponding to TCC with validated data for the healthy operation of the engine.



(d) DNN0 residuals corresponding to TCC with non-validated data for the healthy operation of the engine.



(e) DNN0 residuals corresponding to N1 with validated data for the healthy operation of the engine.



(f) DNN0 residuals corresponding to N1 with non-validated data for the healthy operation of the engine.

Figure 4.17: Effect of the non-validated data that produce false alarms, (when input fuel mass flow rate is at 75% of its maximum).

4.3.3 Effects of Data Validation on Component Fault Detection

When component faults occur the interrelationship among the variables may change and because the ANN network has not seen all the faulty conditions during its training phase, sometimes the network outputs cannot exactly follow the changes due to the component faults but it is still possible to detect faults using the validated data and invoke the benefits of not misdiagnosing the sensor faults instead of components faults. Table 4.5 presents the fault detection results of a bank of dynamic neural networks corresponding to the healthy mode of the engine (DNN0) for three different fault severities (5%, 2% and 1%) by using the validated data.

Regarding the 5% severity, the networks corresponding to the variables PLC and N1 can detect all different fault types that are indicated in the table. The only fault type that cannot be detected by the THC bank of networks is $\Delta\Gamma_{HT}$. Furthermore, the THT bank of networks can only detect the four fault types as presented in the table. As the percentage of the fault severity decreases, the ability of the network DNN0 in detecting faults for all the input profiles decreases. Indeed, for some fault scenarios DNN0 is only able to detect the fault in some specific input fuel mass flow rates as specified in Table 4.5.

Comparing Table 4.5 with the one presented in Chapter 3 (Table 3.5) shows that the bank of networks corresponding to PLC, THC and N1 can detect more faults with the validated data. Besides, one should note that the most important benefits of using the validated data would be to prevent false alarms due to the sensor faults.

For fault isolation using validated data and the bank of DNNs it is required that the

Faults	5% Fault Severity				2% Fault Severity				1% Fault Severity			
	PLC	THC	THT	N1	PLC	THC	THT	N1	PLC	THC	THT	N1
$\Delta\Gamma_{LC}$	✓	✓	✓	✓	✓	o ¹	o ¹	✓	✓	o ²	o ³	-
$\Delta\eta_{LC}$	✓	✓	✓	✓	✓	-	o ¹	o ⁴	✓	-	o ¹	-
$\Delta\Gamma_{HC}$	✓	✓	-	✓	✓	-	-	-	-	-	-	-
$\Delta\eta_{HC}$	✓	✓	-	✓	✓	o ¹	-	o ⁴	o ²	o ³	-	o ³
$\Delta\Gamma_{HT}$	✓	-	-	✓	✓	-	-	-	-	-	-	-
$\Delta\eta_{HT}$	✓	✓	-	✓	✓	o ²	-	✓	✓	o ¹	-	o ¹
$\Delta\Gamma_{LT}$	✓	✓	✓	✓	✓	✓	✓	✓	✓	o ²	o ¹	o ¹
$\Delta\eta_{LT}$	✓	✓	✓	✓	✓	o ¹	o ¹	o ⁴	✓	-	o ³	-

Table 4.5: The fault detection results.

(- or ✓ denotes that one cannot detect or can detect a fault

o¹ denotes that one cannot detect the fault if the input fuel mass flow rate is at 70, 75 and 95 % of its maximum.

o² denotes that one cannot detect the fault if the input fuel mass flow rate is at 70 and 95 % of its maximum.

o³ denotes that one cannot detect the fault if the input fuel mass flow rate is at 70 ,85, 90 and 95 % of its maximum.

o⁴ denotes that one cannot detect the fault if the input fuel mass flow rate is at 75 and 80 % of its maximum.)

ANN follows the changes in the data because of the component faults and only correct the sensor faults. However, as mentioned earlier the ANN trained with data of healthy engine is not able to identify the changes due to the component faults. Therefore, fault isolation with dynamic neural networks cannot be performed using the validated data. In the next chapter, we will show how such problem is solved by using a multiple ANN for improved fault diagnosis capability and performance.

4.4 Conclusion

This chapter presents the results of applying autoassociative neural networks (ANN) to the sensor validation problem in aircraft jet engine. The capability of ANN in validating sensor data including noise and outlier reduction, sensor inaccuracy correction and sensor error correction has been investigated in detail and large number of simulation results were presented. Finally, the results of utilizing the validated data for fault diagnosis using dynamic neural networks have been presented. It was indicated, however that by using such validated data dynamic neural networks does not allow fault isolation. The solution for the problem of fault isolation will be described in the next chapter.

Chapter 5

Multiple-Model Sensor and Components

FDI Using Autoassociative Neural

Networks

In the previous chapter a methodology was proposed for data validation in order to improve the performance of the aircraft engine fault detection system using validated data. However, as mentioned the problem of fault isolation of engine faults under the presence of faulty sensors is more challenging. Indeed, one autoassociative neural network which is trained with data generated in the healthy condition is not adequate to validate the data for all the engine faulty conditions.

This chapter presents the results of applying autoassociative neural networks (ANN) for

both sensor and components fault detection and isolation. We propose a bank of ANNs to diagnose sensor faults as well as the component faults while isolating them in a dual spool aircraft jet engine. The proposed fault diagnosis methodology is an integrated solution to the problem of both sensor faults and component faults even if both the engine faults and sensor faults occur concurrently. The parallel bank of autoassociative neural networks proposed for fault diagnosis can be viewed as multiple-model methodology.

5.1 Sensor Fault Detection Scheme Using ANN

After training the ANN using a backpropagation (BP) technique with a certain number of training samples, the ANN captures the interrelationship among the gas path system variables that have some degree of interdependence with each other. This makes the inputs match the outputs as closely as possible. Therefore, when non-faulty data is fed to the trained ANN, the difference between the input and output of the ANN would be zero. When the data is contaminated (a sensor is faulty), the difference between the input and the output of the ANN will be non-zero. In this manner, the ANN approach can be used to determine the occurrence of sensor faults.

Figure 5.1 shows a sensor monitoring system for a group of n measurements. In principle the ANN maps inputs m_i ($i = 1, 2, \dots, n$) to outputs, m'_i ($i = 1, 2, \dots, n$) in such a manner that $m_i = m'_i$.

As proposed in [55] and [82] when the i^{th} sensor that is input to the autoassociative

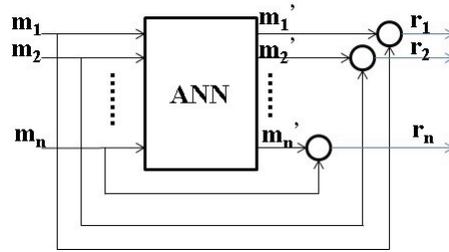


Figure 5.1: Sensor fault detection scheme.

network with the value m_i is faulty as a result of sensor drifts or biases, the network will produce m_i' as a close approximation and estimation of the true value of the measurement m_i . The difference between m_i' and m_i which is the residual can be used as an indicator of that specific faulty sensor.

5.2 Engine Component Fault Diagnosis

In this section, it will be shown that autoassociative neural networks can also be used to perform fault diagnosis in presence of sensor faults for a dual spool gas turbine engine. Occurrence of faults cause the changes in the health or performance parameters which will generate changes in the engine measuring variables and the interrelationship among them. Therefore, having only one ANN that is trained with the healthy data will not be sufficient to estimate the engine variables for all the healthy and faulty operations of the engine. Consequently, a bank of ANNs needs to be utilized. Each bank of neural networks is trained with healthy data as well as the corresponding faulty data. Hence, each network acts as an estimator of both healthy and one of the engine operating conditions corresponding to the various but limited

faulty modes that are of most interest or possible in the jet engine. Note that the fault detection and isolation tasks through this approach will be accomplished simultaneously.

The same input-output variables as those used in Chapter 4 are utilized to construct the ANNs based on TLC, PLC, THC, PHC, TLT, PLT, N1, N2 and TCC and Wf. As mentioned in Chapter 3, the jet engine component faults considered correspond to changes in the eight (8) health parameters which are the efficiencies and the flow capacities of the low pressure compressor, the high pressure compressor, the low pressure turbine, and the high pressure turbine. Therefore, eight (8) component faults, as shown in Table 5.1, are investigated for diagnosis.

Hence, a total of eight (8) models or autoassociative neural networks (ANN) are needed, where each model represents and is associated with two class of the jet engine behaviour, one faulty mode and the other the healthy mode. Table 5.1 presents the associated network labels for each fault scenario. Figure 5.2 shows the structure of the fault diagnosis system that uses a bank of ANNs.

Component Fault	Description	Autoassociative Networks Label
$\Delta\Gamma_{LC}$	Low pressure compressor flow capacity decrease	ANN1
$\Delta\eta_{LC}$	Low pressure compressor efficiency decrease	ANN2
$\Delta\Gamma_{HC}$	High pressure compressor flow capacity decrease	ANN3
$\Delta\eta_{HC}$	High pressure compressor efficiency decrease	ANN4
$\Delta\Gamma_{HT}$	High pressure turbine flow capacity decrease	ANN5
$\Delta\eta_{HT}$	High pressure turbine efficiency decrease	ANN6
$\Delta\Gamma_{LT}$	Low pressure turbine flow capacity decrease	ANN7
$\Delta\eta_{LT}$	Low pressure turbine efficiency decrease	ANN8

Table 5.1: The definitions and descriptions of the considered components faults.

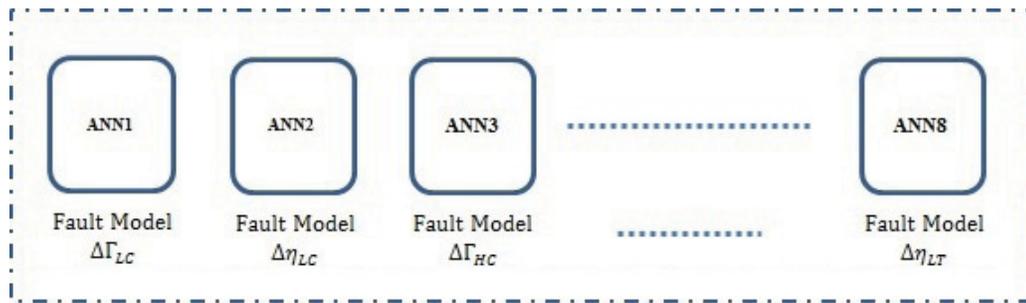


Figure 5.2: The multiple model architecture for bank of ANNs for fault diagnosis.

After constructing the bank of ANNs, the residuals are generated for each output of the ANNs by comparing the jet engine output and each ANN output. The following itemizes the three (3) scenarios that may occur during the operation of the engine and for each case the form of the residuals are described:

- * **Occurrence of no component or sensor fault:** When all the residuals of all ANNs are smaller than an appropriately selected threshold the system is considered healthy and neither component faults nor sensor faults are detected.
- * **Occurrence of component fault:** Once a component fault occurs the residual for the corresponding fault model (the residual of the ANN trained with that specific faulty case) remains under its threshold and other ANNs residuals exceed their thresholds.
- * **Occurrence of sensor fault:** The occurrence of only a sensor fault causes the residuals corresponding to the faulty sensor exceed their thresholds and this should happen in all bank of ANNs.

Considering the generated residuals of ANNs and the above described residual characteristics one can isolate the component faults from the sensor faults.

5.3 Simulation Results

As in the previous two chapters the simulations are performed in the cruise mode of the engine. The ambient pressure and the temperature are set to standard conditions and the Mach number is set to 0.7 as a typical value in the cruise mode. The analysis is conducted for input profiles in the range of 70% to 95% of the maximum fuel rate. Similar to the previous chapters, to select the eight (8) network structures, both the values of the performance index ($J = \frac{\sum_{i=1}^n (y_d(n) - y_{Net}(n))^2}{\sum_{i=1}^n (y_d(n))^2}$, where $y_d(n)$ and $y_{Net}(n)$ denote the desired and the network outputs, respectively, and n is the size of the training or the testing samples) and the generalization capability for validating the data were considered. The data that is used to train the networks are derived from the nonlinear model that is introduced in Chapter 2 and normalized to the range of [-1,1].

The ANNs are used to carry out component fault detection and isolation as well as sensor faults in the gas turbine engine. Common component faults are modelled as changes in the component efficiency and flow capacity. Eight (8) component faults are investigated as shown in Table 5.1. Corresponding to each component fault one autoassociative neural network is trained. Each network is trained with the healthy data (the gas turbine engine variables) as well as the associated faulty data. The variables used for the ANNs are made

up of nine (9) measurements (temperature and pressure of the low pressure compressor, temperature and pressure of the high pressure compressor, temperature and pressure of the high pressure turbine, the high pressure and low pressure rotational speeds and the temperature of the combustion chamber) and also the engine fuel flow which is considered as an engine input.

Considering eight (8) autoassociative neural networks corresponding to the component faults for the fuel flow rate of 70-95%, Table 5.2 specifies each network structure and the percentage of the noise that can be removed by each network for each variable that is obtained from the sensors. The networks are trained with healthy as well as corresponding faulty data with 2% and also 5% fault severities. Table 5.2 shows the significant capability of the ANNs in filtering noise in the temperature and the rotational speed variables.

Neural Networks	Network Structure	Percentage of the noise removed from measurement data							
		TLC	THC	THT	N1	N2	PLC	PHC	PHT
ANN1	10-33-7-33-10	86	57	75	47	54	18	50	19
ANN2	10-43-5-43-10	64	43	50	35	57	6	40	5
ANN3	10-47-6-47-10	73	74	79	48	36	1	58	51
ANN4	10-28-5-28-10	86.5	68	45	68	54	-	60	-
ANN5	10-33-6-33-10	88	65	81	59	79	8	11	17
ANN6	10-44-7-44-10	75	77	40	61	50	-	74.5	63
ANN7	10-42-5-33-10	76	73	63.5	66	61	10	61	15
ANN8	10-33-7-33-10	75	65	70	58.5	62	3	52	30

Table 5.2: The ANN structures and percentage of the noise filtering.

For each ANN1 to ANN8, the residuals which are the differences between the network outputs and the network inputs are calculated, and thresholds are selected for each of them. Thresholds are selected using the probabilistic threshold selection method as described in Section 3.1.1. Selected thresholds are shown in Table 5.3.

Neural Networks	TLC	THC	THT	N1	N2	PLC	PHC	PHT
ANN1	1.65	2.78	4	55	13	0.0072	0.11	0.02
ANN2	2	3.5	4.17	26.8	16.3	0.022	0.07	0.022
ANN3	1.3	2.9	22.5	36.2	28.5	0.015	0.18	0.28
ANN4	1.3	3.45	4.17	19.83	10.67	0.008	0.05	0.009
ANN5	1.3	1.43	2.27	19.33	14.04	0.007	0.068	0.02
ANN6	0.28	1.019	8.11	24.8	10.5	0.0028	0.056	0.019
ANN7	2.06	1.56	5.34	31.61	28.1	0.01	0.13	0.034
ANN8	0.4	0.6	3.55	21.81	18.9	0.0035	0.063	0.01

Table 5.3: The selected thresholds for each ANNs output.

5.3.1 Sensor Fault Detection and Isolation

For sensor fault detection and isolation the ANN outputs are compared with its inputs and from the generated residuals one can investigate if there is a sensor fault and which sensor is faulty. The outputs from the ANN provide the reconstructed and estimated values of the non-faulty inputs. If the input data is fault free, then the ANN output will be the same as the input and the difference between the output and the input will be close to zero. When one of the inputs drift or vary from the nominal value, the corresponding output will not track the input and their difference will be non-zero and this will cause the residual to exceed its threshold.

A number of simulations are performed to investigate the performance of the fault diagnosis system in evaluating sensor faults and determining how the residuals change in the ANNs outputs. Figure 5.3 illustrates a typical sensor drift case on the ANN1 where the drift has a rate of 0.06% per second from the nominal value that is injected at $t=15$ seconds on the inputs corresponding to the variables THT and N2. Figure 5.3(a) shows the actual biased and the neural network estimate for the measurements THT and N2. Figure 5.3(b) shows the

residuals between the estimated outputs and the inputs of the network for eight (8) ANN1 input-output variables. Both the residuals corresponding to the variables THT and N2 have exceeded their thresholds after several seconds, while the residuals corresponding to the other output variables remain under their thresholds. Figures 5.4 to 5.10 show the same fault case for the ANN2 to ANN8 networks, and the same results are obtained for those network as well.

As another sensor drift case, it is assumed that there is a drift of 0.15% per second from the nominal value on the PHC output, when the fuel mass flow rate is at 80% of its maximum value. The sensor reconstruction and the residuals corresponding to ANN4 and ANN8 are depicted in Figures 5.11 and 5.12 (due to the similarity in results the other ANNs are not illustrated). In this scenario only the residual corresponding to the PHC has passed the threshold, therefore the sensor fault can be isolated. However, it is clear from the residuals that there are also small deviations for the other ANN outputs. Indeed, when one of the ANN inputs deviates from its true value, it may affect other ANN outputs. Depending on the “severity” level of the sensor fault, other sensor estimates may or may not be affected by the information of the faulty sensor. Simulation results have shown that a severe fault on a sensor output (which is an input to the ANN) can create false alarms for other outputs due to the network dependencies on a selected few parameters. However, the residual levels are lower than that of the failed sensors.

Considering the above facts one can specify a minimum and a maximum value of the bias for which the specific sensor faults can be detected and isolated. Choosing the thresholds

according to Table 5.3 and assuming that the nominal values of the variables are corresponding to the case when the input fuel mass flow rate is at 85% of its maximum, the minimum and maximum biases for each variable that can be isolated are obtained according to Table 5.4. Out of this range the sensor fault is detectable but the residual is not sufficient to localize the faulty sensor and isolate it as explained above. Finally, Table 5.5 specifies the lowest and the highest value of the bias that can be isolated for each measurement considering all the ANNs. Those obtained by finding the lowest value of the minimum and the highest value of the maximum biases are shown in Table 5.4 for each variable.

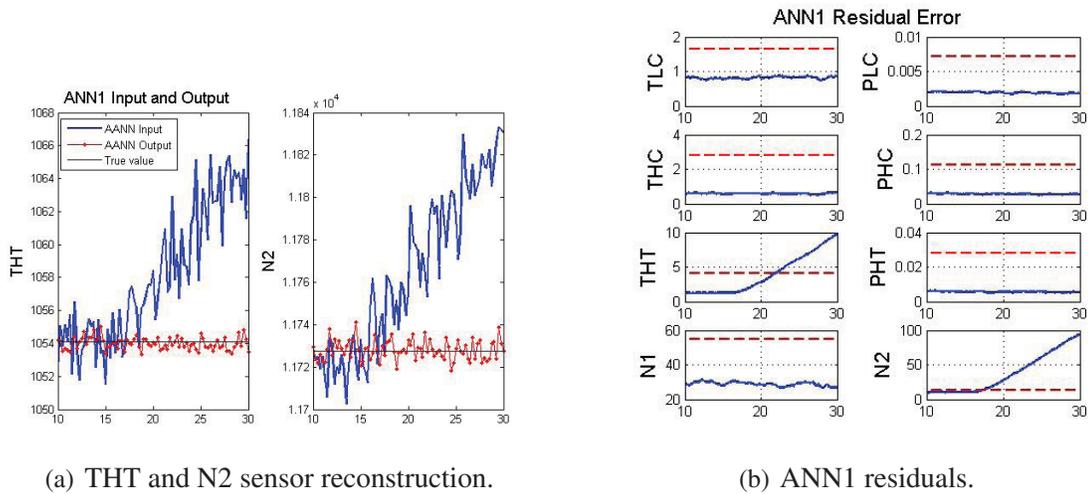
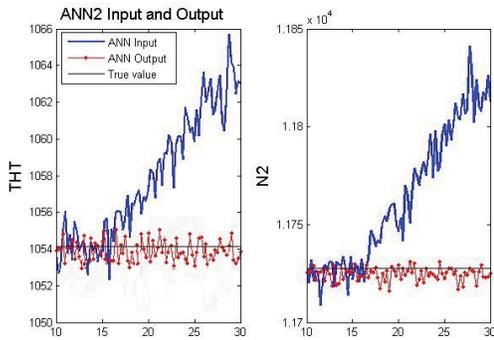
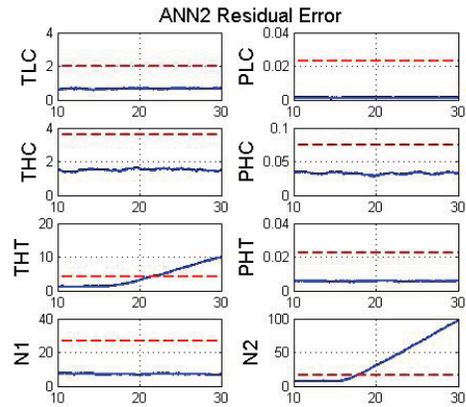


Figure 5.3: The ANN1 sensor reconstruction and the residual error for the sensor drift fault with the rate of 0.06% per second on THT and N2.

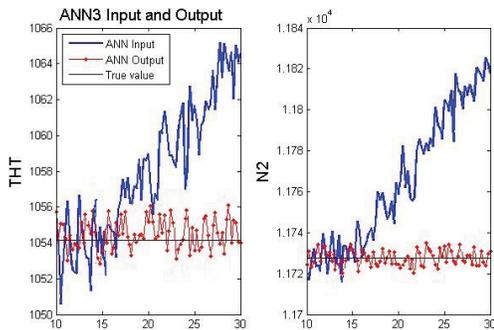


(a) THT and N2 sensor reconstruction.

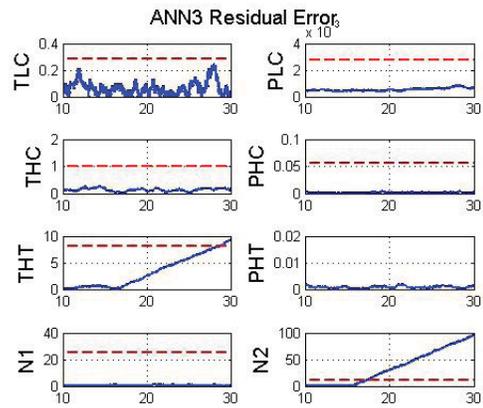


(b) ANN2 residuals.

Figure 5.4: The ANN2 sensor reconstruction and the residual error for the sensor drift fault with the rate of 0.06% per second on THT and N2.

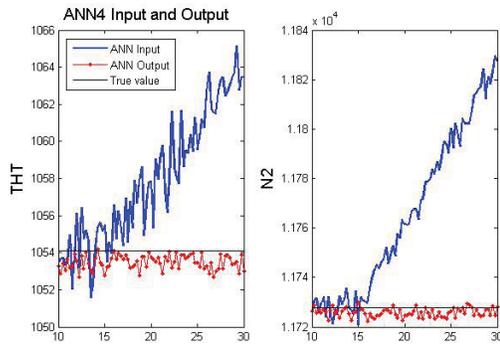


(a) THT and N2 sensor reconstruction.

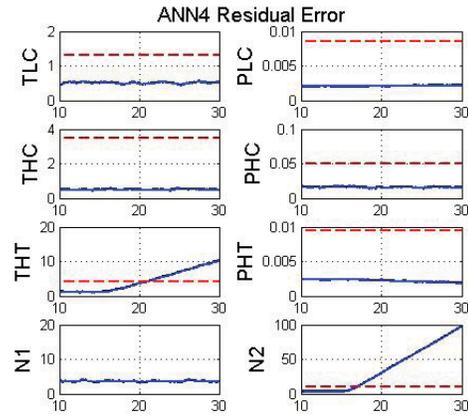


(b) ANN3 residuals.

Figure 5.5: The ANN3 sensor reconstruction and the residual error for the sensor drift fault with the rate of 0.06% per second on THT and N2.

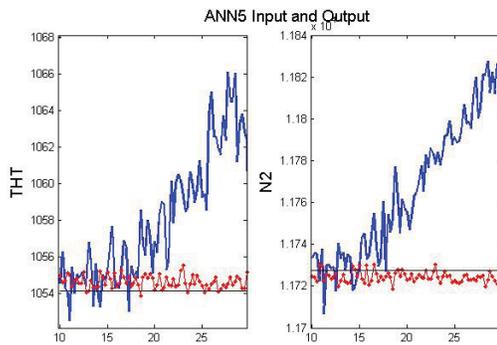


(a) THT and N2 sensor reconstruction.

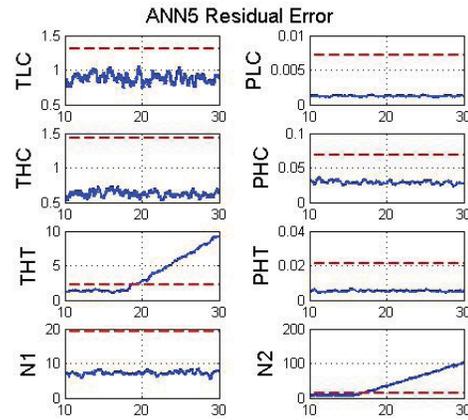


(b) ANN2 residuals.

Figure 5.6: The ANN4 sensor reconstruction and the residual error for the sensor drift fault with the rate of 0.06% per second on THT and N2.



(a) THT and N2 sensor reconstruction.



(b) ANN5 residuals.

Figure 5.7: The ANN5 sensor reconstruction and the residual errors for the sensor drift fault with the rate of 0.06% per second on THT and N2.

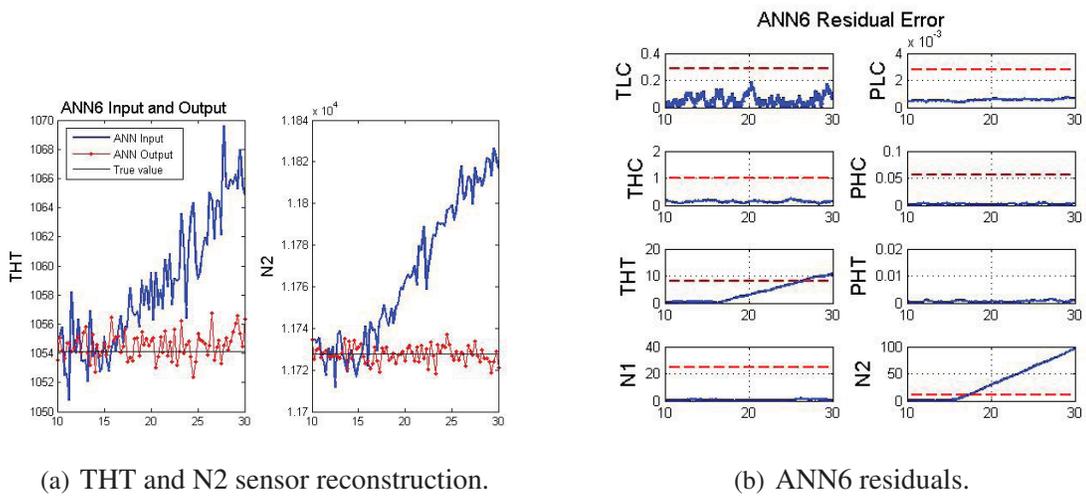


Figure 5.8: The ANN6 sensor reconstruction and the residual errors for the sensor drift fault with the rate of 0.06% per second on THT and N2.

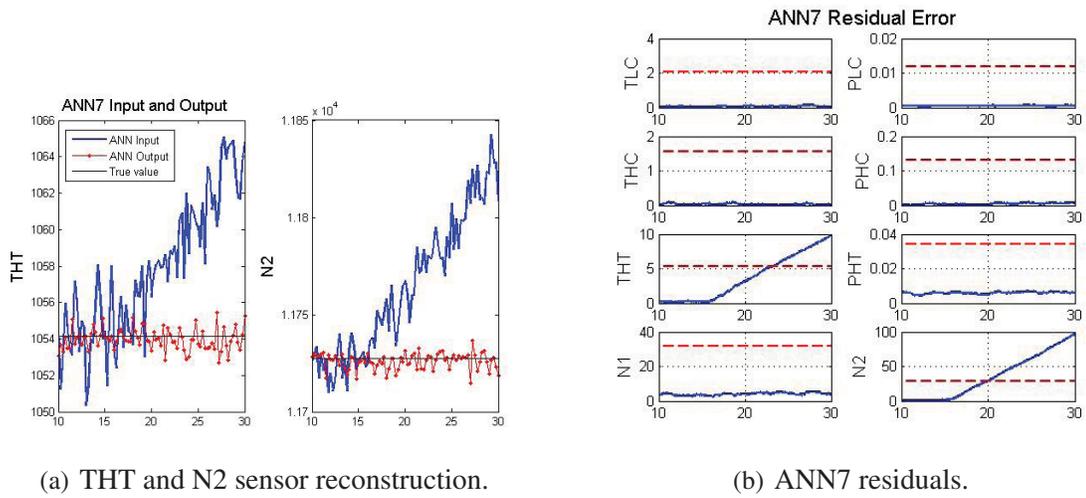


Figure 5.9: The ANN7 sensor reconstruction and the residual errors for the sensor drift fault with the rate of 0.06% per second on THT and N2.

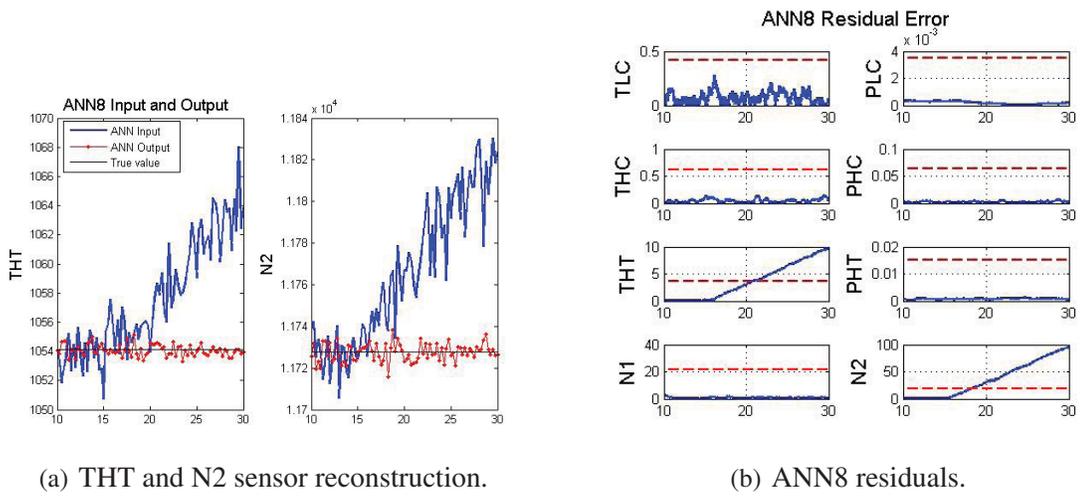


Figure 5.10: The ANN8 sensor reconstruction and the residual errors for the sensor drift fault with the rate of 0.06% per second on THT and N2.

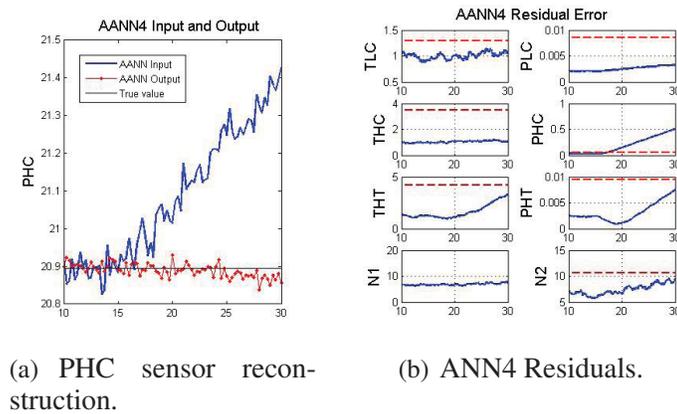
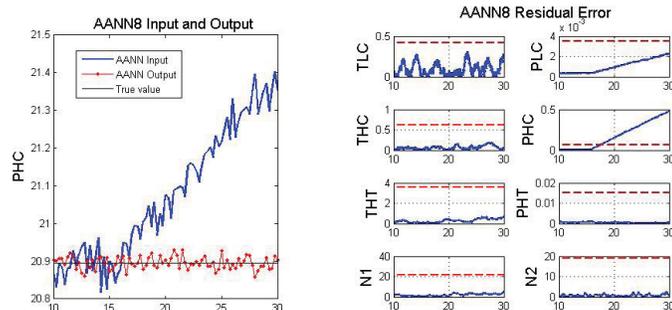


Figure 5.11: The ANN4 sensor reconstruction and residual errors for the sensor drift fault of 0.15% per second on PHC.



(a) PHC sensor reconstruction.

(b) ANN8 Residuals.

Figure 5.12: The ANN8 sensor reconstruction and the residual errors for the sensor drift fault of 0.15% per second on PHC.

Neural Network	TLC(%)	THC(%)	THT(%)	N1(%)	N2(%)	PLC(%)	PHC(%)	PHT(%)
	Min - Max	Min - Max	Min - Max	Min - Max	Min - Max	Min - Max	Min - Max	Min - Max
ANN1	0.3 - 1.27	0.3 - 8.83	0.3 - 29	0.59 - C.F.	0.1 - 16	-	0.47 - 2.4	0.44 - 0.81
ANN2	0.45 - 0.67	0.45 - 5.55	0.39 - 35	0.3 - 98	0.1 - 52.5	-	0.33 - 9.6	0.35 - 0.9
ANN3	0.29 - 0.79	0.38 - 10.72	2 - 51	0.46 - 80	0.2 - C.F.	-	-	-
ANN4	0.29 - 3.4	0.43 - 12.7	0.3 - 37	0.2 - C.F.	0.08 - 15	-	0.2 - 2.9	-
ANN5	0.29 - 0.31	0.18 - 3.6	0.2 - C.F.	0.22 - C.F.	0.1 - 7	0.19 - 0.42	0.28 - 0.87	0.35 - 0.9
ANN6	0.27 - 1.13	0.13 - 2.5	0.76 - 7.11	0.28 - 60	0.04 - 7.5	-	0.2 - 0.8	0.2 - 1.08
ANN7	0.46 - 2.82	0.19 - 3.8	0.5 - 53	0.3 - 24	0.23 - 10.7	-	0.47 - 8.6	0.33 - 4.5
ANN8	0.3 - 4.97	0.1 - 10	0.33 - 7.21	0.2 - 89.4	0.16 - 49	0.42 - 1.3	0.3 - 3.73	-

Table 5.4: The minimum and the maximum biases (C.F. denotes complete failure, when the sensor reaches its full scale value).

	TLC	THC	THT	N1	N2	PLC	PHC	PHT
Maximum Bias	4.97%	12.7%	Complete Failure	Complete Failure	Complete Failure	1.3%	8.6%	4.5%
Minimum Bias	0.27%	0.1% or	0.2%	0.2%	0.1%	0.19%	0.2%	0.2%

Table 5.5: The minimum and the maximum biases.

5.3.2 Component Fault Detection and isolation

Single Fault Scenario

To investigate the performance of the bank of ANNs in detecting component faults several fault scenarios are shown in the following case studies.

(a) **Fault Scenario 4% $\Delta\Gamma_{HT}$** . The fault is injected at $t = 16$ seconds. The ANN5 network is representative of this fault (refer to Table 5.1). Figure 5.13 shows the residuals of the eight (8) autoassociative neural networks. Before the occurrence of the fault all the residuals are below their thresholds, and once the fault occurs the residuals corresponding to the model ANN5 still remain under their thresholds whereas for the other ANNs models almost all the residuals have exceeded their thresholds except for a few of them (for instance, all the ANN1 residuals have exceeded their thresholds except the one corresponding to variable PLC or among ANN2 residuals only the one corresponding to TLC has not exceeded its threshold). Indeed what is required is that at least one of the residuals exceeds from its threshold in any of the other ANNs for fault isolation.

(b) **Fault Scenario 1% $\Delta\eta_{HT}$** . The next faulty scenario is a 1% $\Delta\eta_{HT}$ that is injected at $t=16$ seconds. The ANN6 network is representative of this fault mode (refer to Table 5.1). Figure 5.14 shows the residuals corresponding to the fault for this case. All the ANN6 residuals remain below their thresholds, while at least one residual has exceeded its threshold in the other ANNs which implies that this fault is isolated quite successfully.

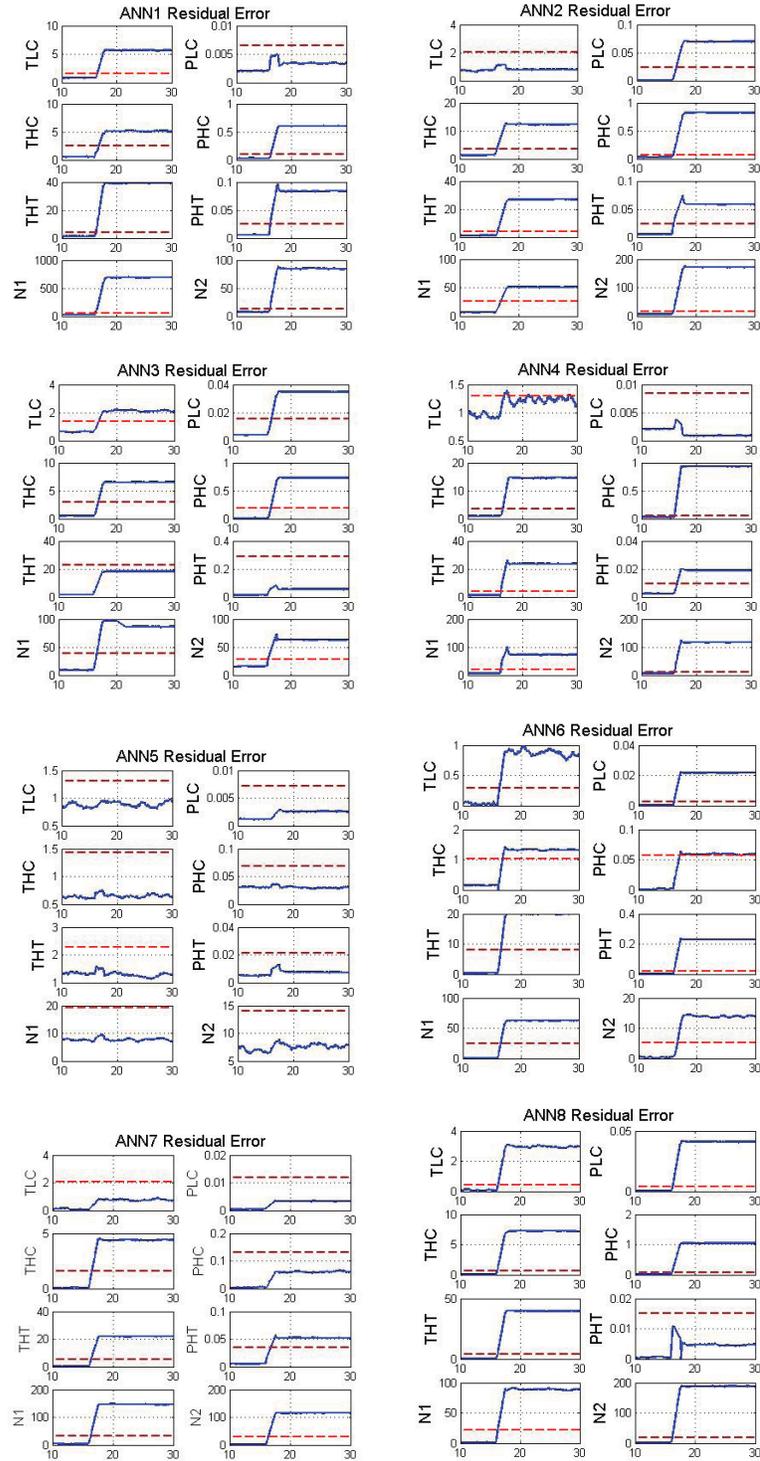


Figure 5.13: The ANN1 to ANN8 generated residuals corresponding to the case of a 4% $\Delta\Gamma_{HT}$ fault.

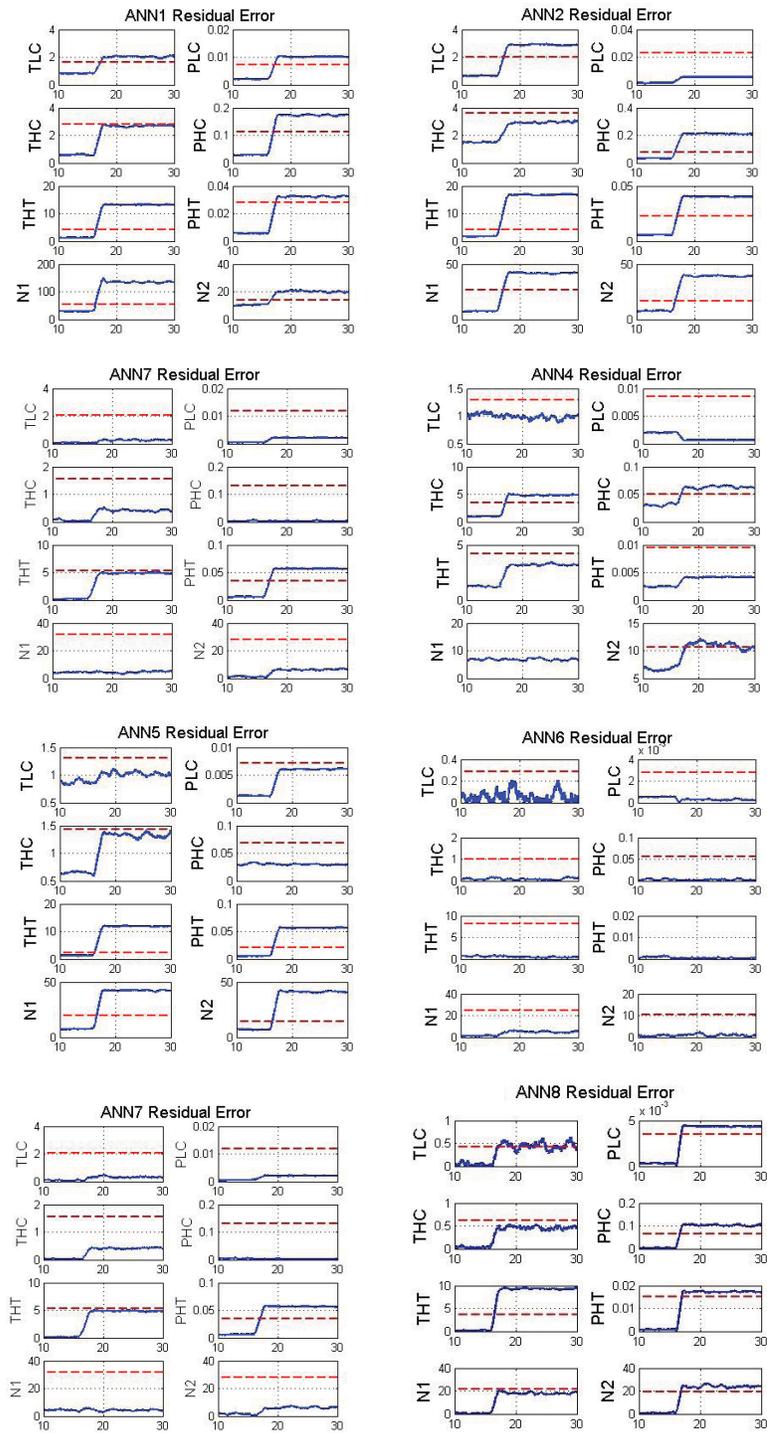


Figure 5.14: The ANN1 to ANN8 generated residuals corresponding to the case of a 1% $\Delta\eta_{HT}$ fault.

(c) **Fault Scenario $\Delta\eta_{LT}$.** For this faulty scenario for which the ANN8 is the representative as depicted in Figure 5.15, for 2% fault severity not only the ANN8 network residuals remain under their thresholds but also none of the ANN6 residuals have the chance to exceed their thresholds. This yields an inability to isolate the fault with such a severity. However, increasing the fault up to 3% as depicted in Figure 5.16 ensures the isolation as the residuals corresponding to the ANN6 have exceeded their thresholds for this fault severity.

(d) **Fault Scenario $\Delta\Gamma_{LT}$.** The ANN5 is representative of this fault case. Figure 5.17 shows the residuals of the ANN5 for the case of 8% and 9% fault severities. It can be seen that for the 8% fault severity the residuals are still under the thresholds so that this fault can be isolated. However, in case of a 9% fault severity the residuals would no longer remain under their thresholds and the isolation cannot be achieved. As mentioned previously, the networks are trained with the data that contain the 2% and 5% fault severities. Therefore, by increasing the fault severities the networks are not capable of correctly estimating the changes on the variables that are caused by the fault. In order to isolate higher fault severities one should add other fault severities data to the training data. This would clearly increase the learning process and requires larger network structures with more neurons. Alternatively, one can construct other banks of ANNs with these higher level fault severities.

A summary of the fault detection and isolation results stating the minimum and the maximum values of the faults that can be diagnosed by using the bank of ANNs are shown in Tables 5.6 - 5.8. We have injected the faults with different severities at different input

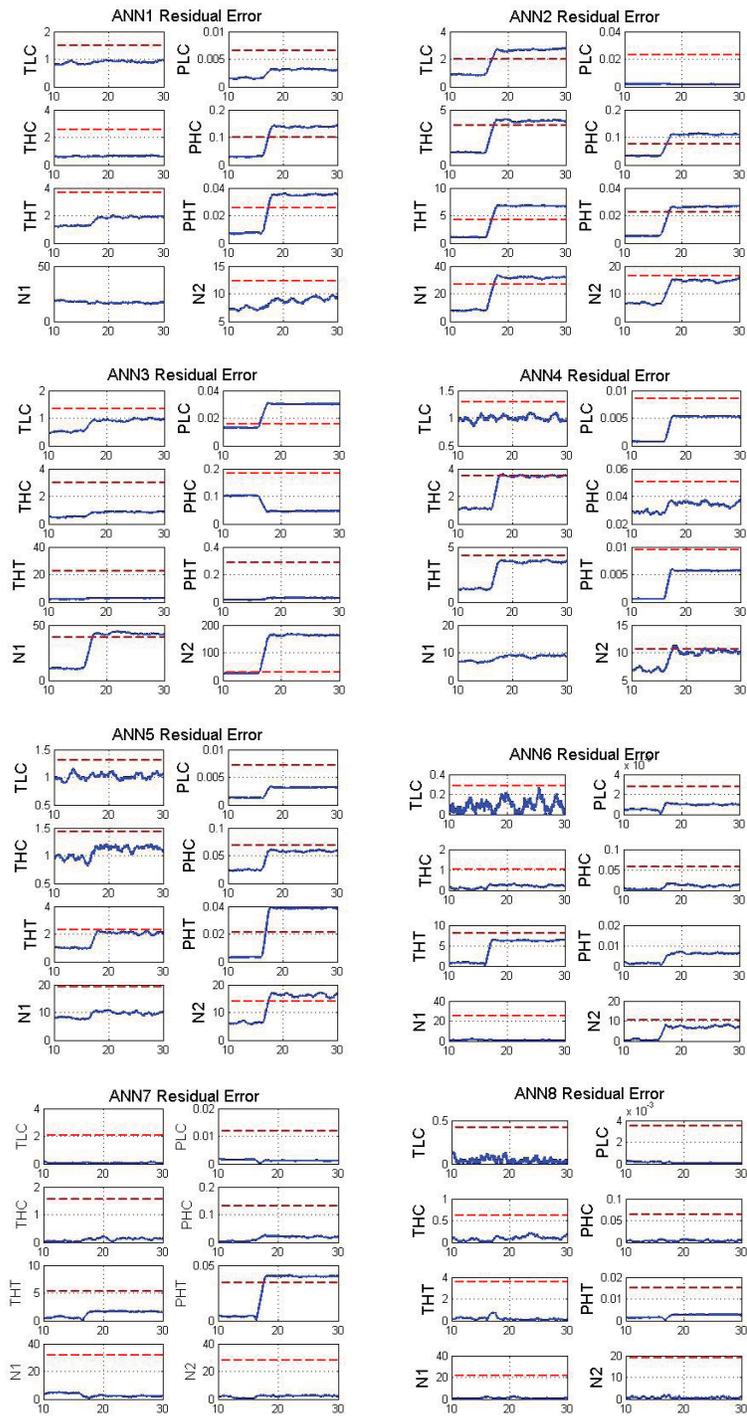


Figure 5.15: The ANN1 to ANN8 generated residuals corresponding to the case of a 2% $\Delta\eta_{LT}$ fault.

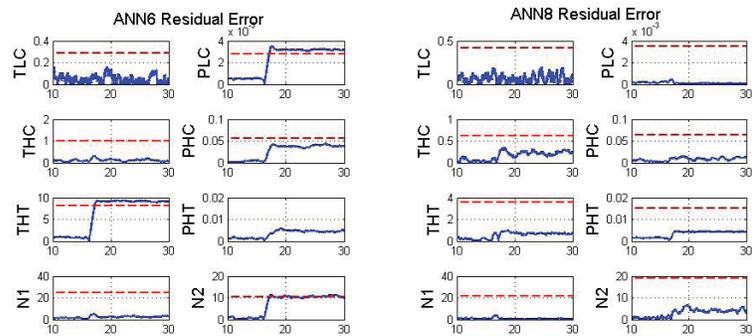


Figure 5.16: The ANN6 and ANN8 generated residuals corresponding to the case of a 3% $\Delta\eta_{LT}$ fault.

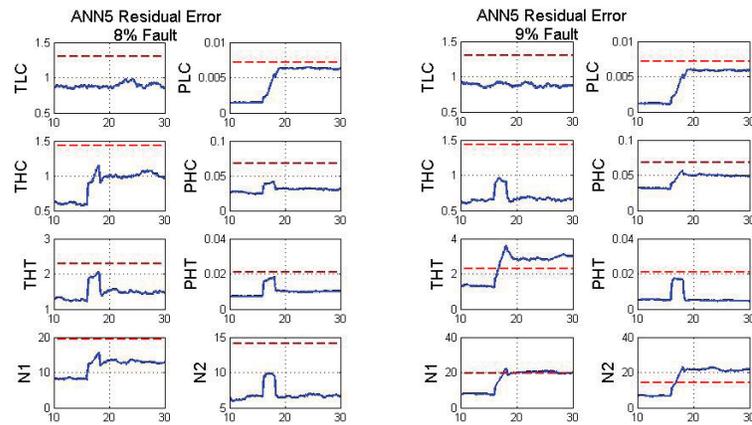


Figure 5.17: The ANN5 generated residuals corresponding to the case of a 8% and 9% $\Delta\Gamma_{HT}$ fault.

fuel mass flow rates. Table 5.6 shows minimum faults that can be detected by each ANN network which are not representative or were not trained with the specific faults. Indeed the specified percentages are the minimum faults for which at least one of the residuals on that ANN exceeds its threshold. Clearly there is no minimum value of fault for the ANN that is representative of that specific fault, therefore a dashed line (-) is used in the table for these cases. Moreover, Table 5.7 shows the maximum severity of faults of what each corresponding ANNs can be representative of. In other words, this table presents the maximum fault by which the corresponding ANN residuals would not exceed their thresholds.

Faults	ANN1	ANN2	ANN3	ANN4	ANN5	ANN6	ANN7	ANN8
$\Delta\Gamma_{LC}$	-	2%	2%	1%	2%	1%	2%	1%
$\Delta\eta_{LC}$	2%	-	1%	1%	1%	1%	2%	1%
$\Delta\Gamma_{HC}$	1%	1%	-	1%	1%	1%	1%	1%
$\Delta\eta_{HC}$	1%	1%	1%	-	1%	1%	1%	1%
$\Delta\Gamma_{HT}$	1%	1%	1%	1%	-	1%	1%	1%
$\Delta\eta_{HT}$	1%	1%	1%	1%	1%	-	1%	1%
$\Delta\Gamma_{LT}$	1%	1%	1%	1%	1%	1%	-	1%
$\Delta\eta_{LT}$	2%	2%	1%	2%	2%	3%	2%	-

Table 5.6: The minimum faults that are detectable by using each autoassociative neural network ANN1-ANN8 (the minimum fault that causes the residuals exceed their thresholds).

Summarizing the two Tables 5.6 and 5.7, the minimum and the maximum faults that can be diagnosed in general by the bank of ANNs are now shown in Table 5.8.

Concurrent Components Faults and Sensor Fault Scenarios

Given the fact that component and sensor faults do not occur exactly at the same time, it is assumed that the $\Delta\Gamma_{HT}$ component fault with a 4% severity has occurred in t=15 seconds and

Faults	ANN1	ANN2	ANN3	ANN4	ANN5	ANN6	ANN7	ANN8
$\Delta\Gamma_{LC}$	6%	-	-	-	-	-	-	-
$\Delta\eta_{LC}$	-	6%	-	-	-	-	-	-
$\Delta\Gamma_{HC}$	-	-	5%	-	-	-	-	-
$\Delta\eta_{HC}$	-	-	-	7%	-	-	-	-
$\Delta\Gamma_{HT}$	-	-	-	-	8%	-	-	-
$\Delta\eta_{HT}$	-	-	-	-	-	5%	-	-
$\Delta\Gamma_{LT}$	-	-	-	-	-	-	5%	-
$\Delta\eta_{LT}$	-	-	-	-	-	-	-	5%

Table 5.7: The maximum faults by which the corresponding ANN residuals would not exceed their thresholds.)

$\Delta\Gamma_{LC}$	$\Delta\eta_{LC}$	$\Delta\Gamma_{HC}$	$\Delta\eta_{HC}$	$\Delta\Gamma_{HT}$	$\Delta\eta_{HT}$	$\Delta\Gamma_{LT}$	$\Delta\eta_{LT}$
2% - 6%	2% - 6%	2% - 5%	1% - 6%	1% - 8%	1% - 5%	1% - 5%	3% - 5%

Table 5.8: The maximum and the minimum faults that can be detected and isolated by using the bank of ANNs.

after 5 seconds (at t=20 seconds) the THC has encountered with a 20K bias (5%). Figure 5.18 shows the residuals corresponding to different networks for this fault scenario. Due to the occurrence of a component fault at t=15 seconds all the residuals corresponding to the network ANN5 remain below their thresholds whereas the other ANNs have their residuals exceed their thresholds which result in the isolation of the component fault. After 5 seconds, due to the occurrence of the THC sensor fault, the only residual in the ANN5 that would exceed its threshold is the one corresponding to the THC output, which makes the sensor isolation task possible.

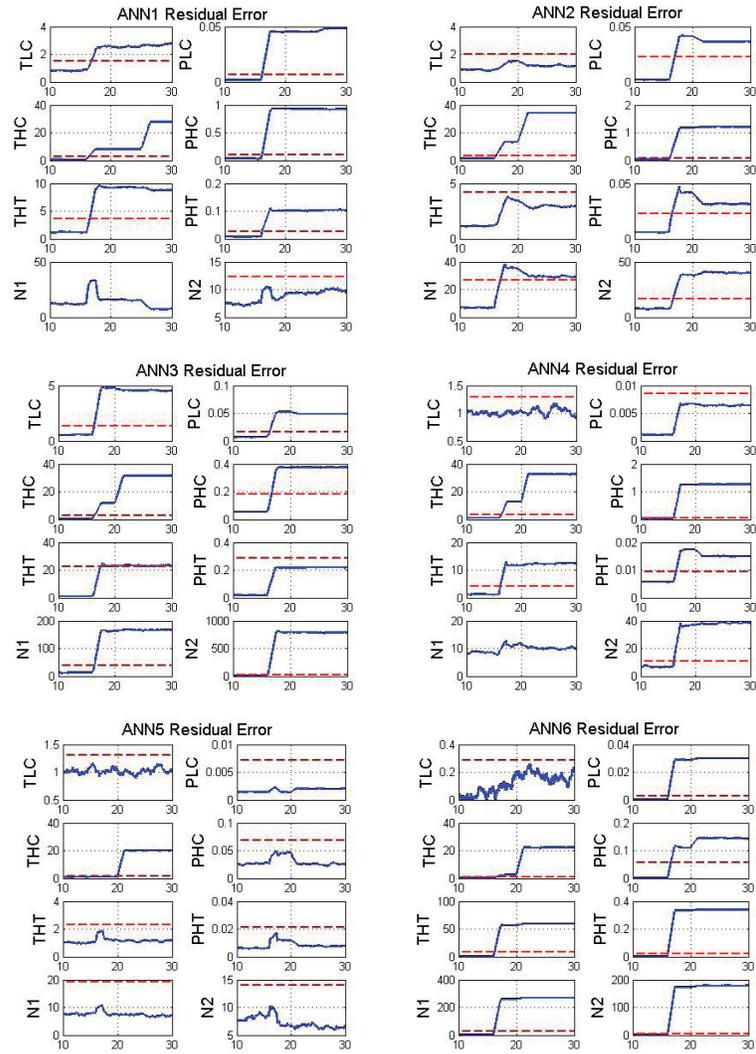
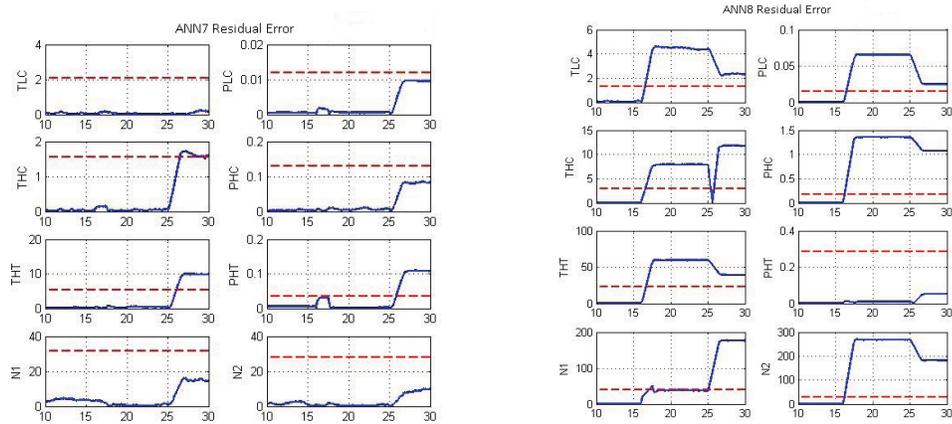


Figure 5.18: The ANN1 to ANN8 generated residuals for the case of a 4% $\Delta\eta_{HT}$ fault injected at t=15 seconds and 20 K bias injected on the THC sensor at t=20 seconds.

Concurrent Component Fault Scenario

Figure 5.19 shows the results for two typical component faults occurring concurrently, namely $\Delta\Gamma_{LT}$ and $\Delta\eta_{LT}$. ANN7 and ANN8 are representative of these faults, respectively, therefore only the residuals corresponding to ANN7 and ANN8 are shown. Figure 5.19(a) illustrates the ANN7 residuals when a 5% $\Delta\Gamma_{LT}$ fault has occurred at t=16 seconds and a 5% $\Delta\eta_{LT}$ fault has occurred concurrently at t=25 seconds. Figure 5.19(b) shows the ANN8 ($\Delta\eta_{LT}$ fault model) residuals for the same scenario. The ANN7 ($\Delta\Gamma_{LT}$ fault model) residuals remain under their threshold when the 5% $\Delta\Gamma_{LT}$ fault occurred at t=16 seconds and the ANN8 residuals exceeded their thresholds, therefore this fault is diagnosed. However, once the second fault has occurred the ANN7 residuals exceed their thresholds whereas the ANN8 residuals still remain above their thresholds due to the fact that the first fault still exists. The other ANNs (ANN1-ANN6) residuals are above their thresholds for those fault occurrences. Hence, the only information that can be concluded is that the second fault has occurred but the fault cannot be isolated. For detection and isolation of two concurrent faults, a hierarchical approach can be employed [7] which is not considered in this thesis and can be treated as future work.



(a) The ANN7 residuals for the 5% $\Delta\Gamma_{LT}$ fault injected at t=16 seconds and the 5% $\Delta\eta_{LT}$ fault injected at t=25 seconds.

(b) The ANN8 residuals for the 5% $\Delta\Gamma_{LT}$ fault injected at t=16 seconds and the 5% $\Delta\eta_{LT}$ fault injected at t=25 seconds.

Figure 5.19: concurrent fault

5.4 Comparison of Autoassociative Neural Networks With Dynamic Neural Networks

This section presents a comparison between the two proposed approaches, namely dynamic neural networks and the autoassociative neural networks for fault diagnosis of the jet engine. As explained in Chapter 3 by using dynamic neural networks approach each network is to learn the input-output dynamic characteristics of the jet engine corresponding to one variable. This is to decrease and reduce the computational cost. For instance, for the four variables PLC, THC, THT and N1, four dynamic neural networks are trained.

On the other hand, the autoassociative neural networks are multi-input and multi-output

networks with a redundancy between their inputs and output variables. For comparison Table 5.9 shows the characteristics and parameters of the ANN7 and DNN7 (both are representative of the $\Delta\Gamma_{LT}$ fault) for three engine output variables (THC, THT and N1). The size of the three DNNs corresponding to the variables THC, THT and N1 are 1-4-4-1, 1-4-5-1 and 1-7-6-1, respectively and the number of updating parameters are 78, 89 and 139, respectively, which gives a total of 306 updating parameters. On the other hand, the ANN7 has the structure of 10-42-5-42-10 and a large number of updating parameters which is 1359. The training and the testing performance indices are also shown in this table. This shows that dynamic neural networks produce a better performance, in other words, the dynamic neural networks quality of learning the engine dynamics is superior to that of the autoassociative neural networks.

Network Characteristics	Dynamic Neural Networks			Autoassociative Neural Networks		
	DNN_{THC}	DNN_{THT}	DNN_{N1}	THC	THT	N1
Size of network	1-4-4-1	1-4-5-1	1-7-6-1	10-42-5-42-10		
Number of parameters	78	89	139	1359		
Total number of parameters	306			1359		
Performance index J_{Train}	0.001	0.0008	0.0043	0.006	0.007	0.0016
Performance index J_{Test}	0.0062	0.0032	0.06	0.02	0.05	0.1

Table 5.9: Characteristics and parameters of the DNN7 and ANN7.

In dynamic neural networks, there are more learning parameters for a single neuron (weights, IIR filter nominator and denominator coefficients, and the activation function slope) and the updating algorithm, namely the “extended backpropagation” is a more computationally intensive than the general backpropagation algorithm, however the networks have smaller architectures as compared to autoassociative neural networks. Indeed, even though extended

backpropagation algorithm is a slightly more complicated than a general backpropagation algorithm, due to the size of the network the training process in the ANN is slightly more time consuming than the DNNs.

By using dynamic neural networks approach, fault detection is decoupled from the fault isolation task. Therefore, if one is only interested in the fault detection problem there is no need to activate the dynamic neural networks corresponding to the isolation task, which can reduce the computational load. In contrast, in the ANN approach the detection and isolation tasks are integrated.

Comparing Table 3.6 (the DNN isolation results) and Table 5.8 (the ANN isolation results) it can be concluded that the DNN performs satisfactorily for the faulty cases with the same fault severity as those that the networks were trained with. On the other hand, the ANN approach is capable of isolating a wider range of fault severities.

Finally, the most important difference is that by using the ANN approach the sensor faults can be isolated as well, and the ANN approach is capable of distinguishing between sensor faults and components faults.

5.5 Conclusion

In this chapter, a multiple-model sensor and components fault detection and isolation approach was introduced for aircraft jet engine using parallel bank of autoassociative neural networks. This approach provides an integrated scheme for simultaneously detecting and isolating the

sensor faults and the component faults. Various scenarios including sensor faults, component faults, concurrent sensor and components faults and concurrent component faults have been presented to investigate the performance of the proposed scheme. A discussion on the simulation results for each scenario is also provided under each section. Finally, comparison of the two proposed neural network-based fault detection and isolation approaches are presented.

Chapter 6

Conclusions and Future Work

6.1 Thesis Summary

In this thesis the problem of fault detection and isolation (FDI) in nonlinear dynamical system of an aircraft jet engine is addressed using two artificial intelligent approaches. Artificial neural networks are employed in this thesis due to their great capability in identifying any nonlinear static and dynamic function and their competence to cope with system complexity, uncertainty as well as noisy and corrupted data and information.

The main challenge in FDI systems is to diagnose incipient and abrupt faults in complex dynamic systems under the assumption that input and output measurements are affected by noise or faulty sensors. In other words, the FDI analysis can be strongly affected by the measurement uncertainty and unreliability. This thesis addresses three main problems in the

jet engine health monitoring and fault diagnosis system, namely component faults diagnosis, data validation and sensor fault diagnosis. Towards this end, two multiple model schemes using two different types of artificial neural networks were introduced.

The first multiple model FDI scheme was composed of dynamic neural networks (DNN). This FDI architecture consists of parallel bank of dynamic neural network estimators which are capable of learning the intrinsic dynamical nonlinear behaviour of the system. Each network corresponds to a specific faulty or healthy mode of the aircraft jet engine. The DNNs in this scheme were constructed based on a dynamic multilayer perceptron network which uses IIR filters to generate dynamics between the input and output of the system. Dynamic neural networks has a great capability in learning the dynamics of complicated nonlinear systems where conventional static neural networks cannot yield an acceptable modelling performance. In other words, such FDI scheme has a specific advantage in terms of making use of both the benefits of multiple model characteristics and the dynamic neural networks.

The second FDI scheme was achieved by using autoassociative neural networks (ANN). A parallel bank of ANNs were proposed to diagnose sensor faults as well as the component faults in the aircraft jet engine. Unlike most FDI techniques, the proposed solution simultaneously accomplishes sensor fault and component fault detection and isolation (FDI) within a unified diagnostic framework. Autoassociative neural networks are feedforward neural networks that use the concept of nonlinear principle components analysis (NPCA) and are practical for filtering or signal smoothing, data validation as well as sensor error correction.

In both proposed FDI schemes, while each network corresponds to a specific operating mode of the engine, generated residuals have been evaluated to determine location and time of the fault occurrence, and all these have been accomplished in presence of measurement noise. Finally, the two proposed fault diagnosis approaches were compared in terms of performance and quality of results.

6.2 Suggestions for Future Work

A large number of potential future works to extend the current research can be envisaged. Some of our plans for future research are explained in the following:

First, the extended dynamic backpropagation algorithm used for training of dynamic neural network, may sometimes get stuck in unsatisfactory local minima of the error function. Such issues can be addressed by using the methods that belong to the other class of global optimization by using stochastic methods such as adaptive random search (ARS) or simultaneous perturbation stochastic approximation (SPSA). This helps to enhance the quality of fault diagnosis by improving the quality of learning and identification capability of the dynamic neural networks.

Second, in this thesis the problem of fault detection and isolation has been addressed. Identification of faults is another essential problem in aircraft engines fault diagnosis. Indeed, accurate identification of fault severities is an invaluable asset for system maintenance as well as development of reliable autonomous recovery procedures. In our approach using

multiple-model dynamic neural networks, the identification can be pursued by adding more DNN trained for different fault severity cases.

Third, as another recommendation for further studies, fault diagnosis in other flight modes such as take off can be addressed considering the situation when the environmental conditions are varying. Furthermore, employing adaptive or dynamic thresholds techniques helps also for better quality of FDI. Adaptive thresholds can intrinsically capture the nonlinear behaviour of the engine, thereby addressing the limitations of fixed thresholds. Residuals may change with the varying control inputs and dynamic operating conditions of the engine. Therefore, using a small fixed threshold may result in significant false alarms, while using a large fixed threshold may increase the number of missed detections and isolations. The adaptive threshold automatically adapts to the changes in the engine operating conditions and engine dynamics to enhance the robustness and fault sensitivity of the FDI scheme.

Fourth another future development for advanced FDI techniques can be focused on investigation of fault diagnosis system in the case when more than one fault or combination of faults occur. For our proposed approach for detection and isolation of two concurrent faults, a hierarchical approach can be employed.

Bibliography

- [1] E. Sobhani-Tehrani, “Fault detection, isolation and identification for nonlinear systems using a hybrid approach,” Master of Applied Science thesis, Concordia University, 2005.
- [2] S. Simani, R. Patton, and C. Fantuzzi, *Model-Based Fault Diagnosis in Dynamic Systems Using Identification Techniques*. Springer-Verlag New York, Inc., 2003.
- [3] A. Volponi, “Foundation of gas path analysis,” in *von Karman Institute Lecture Series: Gas Turbine Condition Monitoring and Fault Diagnosis*, pp. 1–16, 2003.
- [4] M. Ayoubi, “Fault diagnosis with dynamic neural structure and application to a turbocharger,” in *Proc. Int. Symp. Fault Detection, Supervision and Safety for technical Process SAFEPROCESS’94*, (Espoo, Finland), pp. 618–623, 1994.
- [5] R. Flack, *Fundamentals of Jet Propulsion With Applications*, vol. 17. Cambridge Univ Pr, 2005.

- [6] E. Naderi, N. Meskin, and K. Khorasani, "Nonlinear fault diagnosis of jet engines by using a multiple model-based approach," *Journal of Engineering for Gas Turbines and Power by ASME*, vol. 134, January 2012.
- [7] N. Meskin, E. Naderi, and K. Khorasani, "Fault diagnosis of jet engines by using a multiple model-based approach," in *ASME Turbo Expo 2010*, no. GT2010-23442.
- [8] R. Mohammadi, E. Naderi, K. Khorasani, and S. Hashtrudi-Zad, "Fault diagnosis of gas turbine engines by using dynamic neural networks," in *Proc. ASME Turbo Expo*, (Glasgow, UK), pp. 25–30, June 2010.
- [9] R. V. Beard, *Failure accommodation in linear systems through self reorganization*. Phd thesis, Massachusetts Institute of Technology, Mass., USA, 1971.
- [10] R. Isermann, "Model based fault detection and diagnosis- status and applications," in *Proceedings of the 16th Symposium on Automatic Control in Aerospace*, (St. Petersburg), pp. 71–85, 2004.
- [11] R. Isermann, "Process fault detection and diagnosis methods," in *Proceedings of IFAC symposium SAFERPROCESS'94*, (Helsinki, Finland), pp. 597–612, 1994.
- [12] J. Chen and R. J. Patton, *Robust Model-based Fault Diagnosis for Dynamic Systems*. Norwell, Massachusetts: Kluwer Academic, 1999.

- [13] J. Gertler, *Fault Detection and Diagnosis in Engineering Systems*. Switzerland: Marcel Dekker, 1998.
- [14] R. J. Patton, C. Lopeztoribio, and F. J. Uppal, "Artificial intelligence approaches to fault diagnosis," in *IEE Colloquium on Condition Monitoring: Machinery, External Structures and Health*, pp. 1–18, 1999.
- [15] G. Vachtsevanos, F. L. Lewis, M. Roemer, A. Hess, and B. Wu, *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*. New Jersey, John Wiley and Sons Inc.: Hoboken, 2006.
- [16] R. Rengaswamy, D. Mylaraswamy, V. Venkatasubramanian, and K. E. Arzen, "A comparison of model-based and neural network-based diagnostic methods," *Engineering Applications of Artificial Intelligence*, vol. 14, pp. 808–818, 2001.
- [17] J. Korbicz, J. M. Koscielny, Z. Kowalczyk, and W. Cholewa, *Fault Diagnosis: Models, Artificial Intelligence, Applications*. Heidelberg, Germany: Springer-Verlag, 2004.
- [18] N. Kuipel and P. M. Frank, "A fuzzy FDI decision-making system for the support of the human operator," in *Proceedings of the IFAC Symposium SAFEPROCESS'97*, (Hull, UK), pp. 721–726, 1998.
- [19] C. Nan, F. Khan, and M. Iqbal, "Real-time fault diagnosis using knowledge-based expert system,"

- [20] A. S. Willsky and H. L. Jones, "Generalized likelihood ratio approach to detection and estimation of jumps in linear systems," *IEEE Trans. Automatic Control*, vol. 21, no. 1, pp. 108–112, 1976.
- [21] M. Basseville and I. V. Nikiforov, *Detection of Abrupt Changes: Theory and Application*. New York: Prentice-Hall Inc., 1993.
- [22] A. Volponi, "Extending gas-path analysis coverage for other fault conditions," *Von Karman Institute Lecture Series*, vol. 1, pp. 1–7, 2003.
- [23] T. Kobayashi and D. Simon, "Application of a bank of kalman filters for aircraft engine fault diagnostics," in *ASME Conference Proceeding, Turbo Expo*, (Atlanta, GA, USA), pp. 461–470, June 16-19 2003.
- [24] N. Aretakis, "Nonlinear engine component fault diagnosis from a limited number of measurements using a combinatorial approach," in *ASME GT-2002-30031*, vol. 125, pp. 642–650, 2002.
- [25] A. Volponi, H. DePold, R. Ganguli, and C. Daguang, "The use of Kalman-filter and neural-network methodologies in gas-turbine performance diagnostics: a comparative study," in *ASME 2000-GT-547*, pp. 917–924.
- [26] R. Ganguli, R. Verma, and N. Roy, "Soft computing application for gas path fault isolation," in *ASME Turbo Expo.*, vol. 2004, pp. 499–508, 2004.

- [27] R. Ganguli, "Application of fuzzy logic for fault isolation of jet engines," *Journal of Engineering for Gas Turbines and Power*, vol. 125, no. 3, pp. 617–623, 2003.
- [28] K. Mathioudakis, C. Romessis, and A. Stamatis, "Probabilistic methods for gas turbine fault diagnostics," in *von Karman Institute Lecture Series: Gas Turbine Condition Monitoring and Fault Diagnosis*, pp. 1–28, 2003.
- [29] S. Sampath and R. Singh, "An integrated fault diagnostics model using genetic algorithm and neural networks," *Journal of Engineering for Gas Turbines and Power*, vol. 128, no. 1, pp. 49–56, 2006.
- [30] R. Mohammadi, *Fault diagnosis of hybrid systems with application to gas turbine engine*. PhD thesis, Concordia University, 2009.
- [31] G. Betta and A. Pietrosanto, "Instrument fault detection and isolation: State of the art and new research trends," in *IEEE Transactions on Instrumentation and Measurement*, vol. 49, pp. 100–107, 2000.
- [32] P. Frank, "Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy- a survey and some new results," *Automatica*, vol. 26, no. 3, pp. 709–719, 1990.
- [33] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. Kavuri, "A review of process fault detection and diagnosis- part i: Quantitative model-based methods," in *Computer and Chemical Engineering*, vol. 27, pp. 293–311, March 2003.

- [34] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. Kavuri, "A review of process fault detection and diagnosis- part ii: Qualitative models and research strategies," in *Computer and Chemical Engineering*, vol. 27, pp. 313–326, March 2003.
- [35] L. Marinai, D. Probert, and R. Singh, "Prospects for aero gas-turbine diagnostics:a review," *Applied Energy*, vol. 79, pp. 109–126, 2004.
- [36] A. R. S. Sarkar, X. Jin, "Data-driven fault detection in aircraft engines with noisy sensor measurements," *Engineering for Gas Turbines and Power*, vol. 133, no. 081602, 2011.
- [37] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," in *IEEE Trans. on Neural Networks*, pp. 4–27, 1990.
- [38] R. Eustace and G. Merrington, "Fault diagnosis of fleet engines using neural networks," in *ISABE- 12th International Symposium on Air Breathing Engines*, (Melbourne, Australia), pp. 926–936, 1995.
- [39] A. Green, "Artificial-intelligence for real-time diagnostics and prognostics of gas-turbine engines," in *AIAA Conference*, (Seattle, WA, USA), July 1997.
- [40] A. Valdes, K. Khorasani, and L. Ma, "Dynamic neural network-based fault detection and isolation for thrusters in formation flying of satellites," *Advances in Neural Networks- ISNN 2009*, pp. 780–793, 2009.

- [41] D. Capriglione, C. Liguori, C. Pianese, and A. Pietrosanto, "On-line sensor fault detection, isolation, and accommodation in automotive engines," *IEEE Transactions on Instrumentation and Measurement*, vol. 52, no. 4, pp. 1182–1189, 2003.
- [42] G. Campa, M. Krishnamurty, M. Gautam, M. Napolitano, and M. M. Perhinschi, "A neural network based sensor validation scheme for heavy-duty diesel engines," in *Proceedings of the 14th Mediterranean Conference on Control and Automation*, (Como, Italy), pp. 1–6, 2006.
- [43] L. Li, L. Ma, and K. Khorasani, "A dynamic recurrent neural network fault diagnosis and isolation architecture for satellite's actuator/thruster failures," *Advances in Neural Networks–ISNN 2005*, pp. 981–981, 2005.
- [44] S. Zhao and K. Khorasani, "A recurrent neural network based fault diagnosis scheme for a satellite," in *33rd Annual Conference of the IEEE Industrial Electronics Society. IECON 2007*, (Taipei City, Taiwan), pp. 2660–2665, 2007.
- [45] E. Sobhani-Tehrani, K. Khorasani, and S. Tafazoli, "Dynamic neural network-based estimator for fault diagnosis in reaction wheel actuator of satellite attitude control system," in *IEEE International Joint Conference on Neural Networks, IJCNN'05*, pp. 2347–2352, 2005.

- [46] M. Zedda and R. Singh, "Fault diagnosis of a turbofan engine using neural-networks: a quantitative approach," in *34th AIAA, ASME, SAE, ASEE Joint Propulsion Conference*, (Cleveland, OH), 1998.
- [47] K. Kanelopoulos, A. Stamatis, and K. Mathioudakis, "Incorporating neural-networks into gas-turbine performance diagnostics," in *ASME 97-GT-35, International Gas-turbine & Aero-engine Congress & Exhibition*, (Orlando, Florida), 1997.
- [48] S. Ogaji, *Advanced gas-path fault diagnostics for stationary gas-turbines*. PhD thesis, Cranfield University, 2003.
- [49] C. Romessis, A. Stamatis, and K. Mathioudakis, "A parametric investigation of the diagnostic ability of probabilistic neural networks on turbo fan engines," in *ASME*, no. 2001-GT-11.
- [50] R. Patton, C. Lopez-Toribio, and F. Uppal, "Artificial intelligence approaches to fault diagnosis for dynamic systems," *International Journal of Applied Mathematics and Computer Science*, vol. 9, no. 3, pp. 471–518, 1999.
- [51] N. Sinha, M. Gupta, and D. Rao, "Dynamic neural networks: an overview," in *Proceedings of IEEE International Conference on Industrial Technology 2000*, pp. 491–496, Jan 2000.

- [52] M. Gupta and D. Rao, "Performance comparison of dynamic neural networks as applied to robot inverse kinematic computations," in *American Control Conference*, (Baltimore, MD, USA), pp. 2153–2157, July 1994.
- [53] A. Yazdizadeh and K. Khorasani, "Adaptive time delay neural network structures for nonlinear system identification," *Neurocomputing*, vol. 47, pp. 207–240, 2002.
- [54] I. Alzyoud and K. Khorasani, "Detection of actuator faults using a dynamic neural network for the attitude control subsystem of a satellite," in *Proceeding of International Joint Conference on Neural Networks*, (Montreal, Canada), Jul. 31 - Aug. 4 2005.
- [55] J. Hines and R. E. Uhrig, "Use of autoassociative neural networks for signal validation," *Journal of Intelligent and Robotic Systems*, pp. 143–154, 1998.
- [56] M. Zedda and R. Singh, "Neural-network-based sensor validation for gas turbine test bed analysis," in *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 2001.
- [57] E. Kiyak, A. Kahvecioglu, and F. Caliskan, "Aircraft sensor and actuator fault detection, isolation, and accommodation," *Journal of Aerospace Engineering*, vol. 24, p. 46, 2011.
- [58] C. Hajiyev and F. Caliskan, "Fault detection in flight control systems via innovation sequence of kalman filter," *Journal of Progress in System and Robot Analysis and Control Design*, pp. 63–74, 1999.

- [59] C. Hajiyeve and F. Caliskan, "Sensor/actuator fault diagnosis based on statistical analysis of innovation sequence and robust Kalman filtering," in *Aerosp. Sci. Technology*, vol. 4, pp. 415–422, 2000.
- [60] V. Palade and C. Bocaniala, *Computational intelligence in fault diagnosis*. Springer Publishing Company, Incorporated, 2010.
- [61] P. Frank, "Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy- A survey and some new results," *Automatica*, vol. 26, pp. 459–474, 1990.
- [62] I. Samy, I. Postlethwaite, and D. Gu, "Neural network based sensor validation scheme demonstrated on an unmanned air (uav) model," in *Proceedings of the 47th IEEE Conference on Decision and Control*, (Cancun, Mexico), Dec,9-11 2008.
- [63] R. Clark, "Instrument fault detection," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 14, no. 3, pp. 456–465, 1978.
- [64] Y. An, *A design of fault tolerant flight control systems for sensor and actuator failures using on-line learning neural networks*. PhD thesis, West Virginia University, 1998.
- [65] T. Kobayashi and D. Simon, "Hybrid kalman filter approach for aircraft engine in-flight diagnostics: Sensor fault detection case," in *ASME*, no. GT2006-90870, pp. 746–754, 2006.

- [66] C. G. W. Sheng, M. Napolitano, and M. Fravolini, "Online learning RBF neural networks for sensor validation," in *AIAA Guidance Navigation and Control Conference*, (Monterey, CA, USA), August 2002.
- [67] M. Napolitano, C. Neppach, V. Casdorff, and M. Innocenti, "Sensor failure detection, identification and accommodation using online learning neural architectures," in *Journal of Guidance, Control and Dynamics*, 1994.
- [68] M. Napolitano, Y. An, and B. Seanor, "A fault tolerant flight control system for sensor and actuator failures using neural networks," in *Aircraft Design*, vol. 3, pp. 103–128, 2000.
- [69] M. Napolitano, C. Neppach, V. Casdorff, and M. Innocenti, "Neural network based scheme for sensor failure detection, identification and accommodation," *AIAA Journal of Guidance Control and Dynamics*, vol. 18(6), pp. 1280–1286, 1995.
- [70] I. Samy, I. Postlethwaite, and D. Gu, "A comparative study of NN and EKF based SFDA schemes with application to a nonlinear UAV model," *International Journal of Control*, no. 83, pp. 1025–1043, 2010.
- [71] C. Neppach and V. Casdorff, "Sensor failure detection, identification and accommodation in a system without sensor redundancy," in *AIAA Journal of Guidance Control and Dynamics -1995-11*, (Reno, NV, USA), 1995.

- [72] M. Napolitano and D. Windon, "A comparison between kalman filter and neural network approaches for sensor validation," in *AIAA, Guidance, Navigation and Control Conference, San Diego, CA*, 1996.
- [73] H. Guo-jian, L. Gui-xiong, C. Geng-xin, and C. Tie-qun, "Self-recovery method based on auto-associative neural network for intelligent sensors," in *Proceedings of the 8th World Congress on Intelligent Control and Automation*, (Jinan, China), July 2010.
- [74] M. Najafi, C. Culp, and R. Langari, "Performance study of enhanced auto-associative neural networks for sensor fault detection," *Energy Systems Laboratory*, 2004.
- [75] K. Patan and T. Parisini, "Identification of neural dynamic models for fault detection and isolation: the case of a real sugar evaporation process," *Journal of Process Control*, vol. 15, no. 1, pp. 67–79, 2004.
- [76] J. Korbicz, K. Patan, and A. Obuchowicz, "Dynamic neural networks for process modelling in fault detection and isolation systems," *Int. Journal of Applied Mathematics and Computer Science*, vol. 9, no. 3, pp. 519–546, 1999.
- [77] M. A. Kramer, "Autoassociative neural networks," *Computers and Chemical Engineering*, vol. 16, no. 4, pp. 143–154, 1998.
- [78] B. Kosko, "Bidirectional associative memories," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 18, no. 1, pp. 49–60, 1988.

- [79] H. Saravanamuttoo, G. Rogers, and H. Cohen, *Gas Turbine Theory*. ISBN-13: 978-0130158475, Pearson Education, 2001.
- [80] W. Visser, O. Kogenhop, and M. Oostveen, “A generic approach for gas turbine adaptive modeling,” *Journal of Engineering for Gas Turbines and Power*, vol. 126, pp. 334–341, 2004.
- [81] P. Lu and T. Hsu, “Application of autoassociative neural network on gas-path sensor data validation,” *Journal of Propulsion and Power*, vol. 18, no. 4, pp. 879–888, 2002.
- [82] J. Hines, D. Wrest, and R. Uhrig, “Plant wide sensor calibration monitoring,” in *Proceedings of the 1996 IEEE International Symposium on Intelligent Control*, pp. 378–383, 1996.