FAULT DETECTION AND ISOLATION OF JET ENGINES USING NEURAL NETWORKS

Seyed Sina Tayarani Bathaie

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
Concordia University
Montréal, Québec, Canada

August 2012

Concordia University
School of Graduate Studies

This is to certify that the thesis prepared

By: **Seyed Sina Tayarani Bathaie**

Entitled: **Fault Detection and Isolation of Jet Engines using Neural Networks**

and submitted in partial fulfillment 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 commitee:

_____ Chair
Dr. R. Raut

_____ Examiner, External
Dr. W.F. Xie (M.I.E.) To the Program

_____ Examiner
Dr. L. A. Lopes

_____ Supervisor
Dr. K. Khorasani

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

_____ 20 _____ _____
Dr. Robin A. L. Drew
Dean, Faculty of Engineering
and Computer Science

Abstract

Fault Detection and Isolation of Jet Engines using Neural Networks

Seyed Sina Tayarani Bathaie

The main objective of this thesis is to design a fault detection and isolation (FDI) scheme for the aircraft jet engine using dynamic neural networks. Toward this end two different types of dynamic neural networks are used to learn the engine dynamics. Specially, dynamic neural model (DNM) and time delay neural network (TDNN) are utilized. The DNM is constructed by using dynamic neurons which utilize infinite impulse response (IIR) filters to generate dynamical behaviour between the input and output of the network. On the other hand, TDNN uses several delays associated with the inputs of the neurons to achieve a dynamic input-output map. We have investigated the fault detection performance of each structure. A bank of neural networks consisting of a set of 12 networks that are trained separately to capture the dynamic relations of all the 12 engine parameters are considered in this study. The results show that certain engine parameters have better detection capabilities as compared to the others. Moreover, the fault detection performance was improved by introduction of the concept of "enhanced fault diagnosis scheme" which employs several networks and monitors several engine parameters simultaneously to enhance and improve the accuracy and performance of the diagnostic system.

The fault isolation task is accomplished by using a multilayer perception (MLP) network as a classifier. The concept behind the isolation is motivated by the fact that there is a specific map between the residuals of different networks and a particular fault scenario. We show that the MLP has good capability in learning this map and isolates the faults that are occurring in the jet engine. To demonstrate our diagnostic scheme capabilities, 8 different fault scenarios are simulated and according to the simulation results, our proposed FDI scheme represents a promising tool for fault detection as well as fault isolation requirements.

Acknowledgments

I would like to dedicate this thesis to my dear mother for all the supports and love she gave to me that I know my achievement could not be done without her support and love. I would like to also express my sincere gratitude to my supervisor, Dr K. Khorasani, for his constant support, guidance and encouragement through the course of this thesis.

Contents

List of Figures

List of Tables

List of Abbreviations and Symbols

amb	Ambient
C	Compressor
CC	Combustion chamber
crit	Critical
d	Intake
f	Fuel
HC	High pressure compressor
HT	High pressure turbine
i	Stage input
LC	Low pressure compressor
LT	Low pressure turbine
mech	Mechanical
n	Nozzle
n_i	Nozzle input
n_o	Nozzle output
o	Stage output
T	Turbine
β	Bypass ratio
$\dot{m}$	Mass flow rate, $\frac{Kg}{s}$
γ	Heat capacity ratio
η	Efficiency
π	Pressure ratio
A	Area, m^2
c_p	Specific heat at constant pressure, $\frac{J}{Kg.K}$
c_v	Specific heat at constant volume, $\frac{J}{Kg.K}$
E	Rotor energy, J
H_u	Fuel specific heat,$\frac{J}{Kg}$
J	Rotor moment of inertia, $\frac{Kg}{m^2}$
M	Mach number

N	Rotational speed, rpm
N_1	Rotational speed of spool connecting the low pressure compressor to the low pressure turbine, rpm
N_2	Rotational speed of spool connecting the high pressure compressor to the low pressure turbine, rpm
P	Pressure, $pascal$
P_0	Pressure at sea level at standard day
R	Gas constant, $\frac{J}{Kg.K}$
T	Temperature, K
T_0	Temperature at sea level at standard day
u	Speed
V	Volume, m^3
W_C	Power consumed by compressor, W
W_T	Power generated by turbine, W

Chapter 1

Introduction

1.1 General Area of Research

Fault diagnosis has been a matter of wide interest during the past few years due to importance of reliable operation in industry and in everyday life. The ability to detect the fault in an operation in an early stage is not only important for the safety reasons but also it can avoid the high cost of a failure and overhaul of a system. Hence, fault diagnosis is becoming more significant in industry. In aerospace industry jet engine related costs involve a large portion of the operating cost of an aircraft. Fault diagnosis of engines allows one to avoid the high costs of a stopped flight or even unnecessary replacement of parts and elements of the jet engine. Hence, an early diagnosis of the jet engine can lead to important consequences in the maintenance cost of the aircraft.

Fault diagnosis algorithms are primarily divided into two different categories, namely model-based and data-driven (intelligent-based) approaches. Model-based approaches mostly rely on analytical redundancy and employ analytical mathematical model of the system [1] [2]. However, it is usually quite challenging to find an exact mathematical model of the system due to existence of uncertainties, unmodeled dynamics, noise and disturbances.

In contrast, data driven approaches mostly rely on historical or real-time data from the system measurements, and do not require detailed mathematical model of the system. Hence, intelligent-based methods have received lots of attention in the literature. Among the intelligent-based fault diagnosis approaches, artificial neural networks (ANN) are among the popular methods due to their promising capabilities in learning the dynamics and input-output relations of a system. They provide a viable tool for dealing with nonlinear problems and modelling complex and nonlinear systems with great flexibility and capability. However, in some cases static neural networks suffer from some drawbacks. The information flow in a static neural network is always from the input to the output in feed forward networks and there is no feedback in the network. Moreover, the static neuron does not model time delays associated with the dynamics of the system. Due to the fact that practical systems have a dynamic input-output behaviour, static neural networks cannot be an appropriate tool for modelling their dynamics. Hence, recently a great deal of attention has been paid to dynamic neural networks (DNN). Dynamic neural networks employ internal feedback within the neurons of the network. This implies that the network has a local memory characteristic and can generate dynamic input-output behaviour.

Due to the fact that the aircraft jet engine is a highly nonlinear dynamic system, a static neural network is not a proper approach to learn the dynamics of the engine. Hence, dynamic neural networks are used in the literature for modelling dynamic systems such as the aircraft engine.

1.2 Statement of the Problem

The main objective of this thesis is to develop a fault diagnosis scheme for a dual spool turbo fan engine using dynamic neural networks. The engine in a highly nonlinear dynamical system, hence, a dynamic neural network is required to learn the

dynamics of the aircraft engine. Moreover, since there is generally no accurate access to the mathematical equations of an engine, therefore developing model-based approaches would be a challenging task leading to high costs. Therefore, intelligent-based fault diagnosis schemes are of great interest. Towards this end, two different types of dynamic neural networks are used in this thesis to learn the dynamics of the engine namely, dynamic neuron unit and time-delayed neural networks. The fault diagnosis capabilities and comparisons between these two approaches are made through extensive set of simulation scenarios and case studies. An enhanced fault diagnostic approach is also presented to improve the fault detection performance by employing a bank of neural networks and residual generators.

The fault isolation task is also accomplished by using a neural network classifier and the residuals generated by the fault detection module. Simulation results show that a multi-layer perceptron can be useful tool for performing fault isolation in the aircraft jet engine.

1.3 Literature Review

1.3.1 Fault Diagnosis and Isolation (FDI)

The term fault is defined in the literature as an unexpected change in the system function. Faults are generally classified as occurring in the actuators, or sensors or components of the process.

Fault diagnosis task is carried on after a healthy operation of the system is available using either model-based approaches or process history-based approaches using either quantitative or qualitative methods [3] [4]. On the other hand, the difference between healthy model output and actual output generates a signal that is called a residual. This signal would be used as a measure to detect the occurrence of a fault in

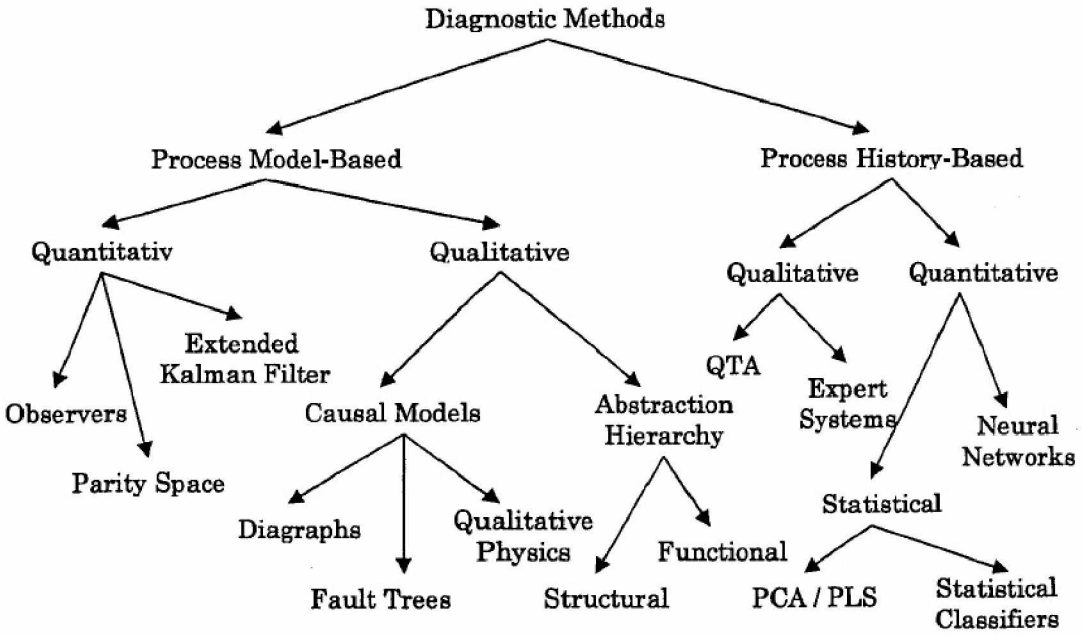


Figure 1.1: Classification of the diagnostic methods [10].

the system. This residual can also be post processed using signal processing methods or frequency analysis for isolation and fault identification tasks.

Quantitative model-based approaches are based on state estimation, parameter estimation and parity space concepts. The idea behind these methods is that a fault will cause changes to some physical parameters and can lead to a change in certain state or model parameters. Therefore, by monitoring the estimated parameters one can detect and isolate a fault [5].

In contrast, in qualitative model-based approaches the relationships between system parameters are used to describe the system behaviour by some qualitative terms such as if-then and causalities rules. Qualitative model-based methods can be used either as qualitative causal methods or abstraction hierarchies. The information flow in the causal models can be developed in various forms such as digraphs, fault trees or qualitative physics. The details regarding the qualitative methods are presented in [4].

Another fault diagnosis approach which is shown in Fig. 1.1 is based on the process history. In these approaches it is assumed that a large amount of historical data are available. These data can be transformed to useful information for fault diagnosis through various methods known as feature extraction. Feature extraction can be either qualitative or quantitative. Qualitative history-based methods can be mainly by expert systems or qualitative trend analysis [5]. Generally the rule-based expert systems need an extensive data about the history of the process and the rules and the diagnosis reasoning corresponding to these rules. Hence, the approach might be time consuming to develop due to the necessity of processing huge amount of data [5].

Neural networks and statistical classifiers are the main methods for extracting quantitative historical information. Main statistical feature extraction methods are based on partial least square (PLS), principal component analyze (PCA) and statistical pattern classifiers [6]. Generally quantitative feature extraction methods formulate the fault diagnosis problem as pattern recognition problem. The details on these approaches are presented in [6].

The most important classifier that is used also in this thesis is the neural network classifiers. These networks have been employed extensively in pattern recognition problems and system identification [7] [8]. The interest towards neural networks in fault diagnosis is due to their capabilities to cope with uncertainties, nonlinearities and complexities. Hence, neural networks are considered as powerful modelling tools for representing highly nonlinear processes.

As presented earlier, fault diagnosis task will carried on after a healthy behaviour of the system is available using either model-based approaches of process history-based approaches. It is generally desirable to have knowledge about the system beyond the

presence of a fault, that is fault detection and to actually isolate the location or characteristics of the fault. In the past, fault isolation has been limited to simple cases in which a single measure was interpreted as a specific fault. However, nowadays more sophisticated methods of fault isolation have been developed. Usually some form of a fault evaluator or reasoner is used to provide the fault isolation capability. The examples of this form are the case-based reasoning which is based on past experiences to reason ones way from the observed changes and conditions from the normal situation.

Various methods have been used to perform the isolation task in the literature. Patan *et al.* [9] presented a multiple model approach for the fault isolation of the sugar evaporation process. They used three dynamic neural networks which were trained in the faulty scenario, comparing the residuals of the faulty-trained network with the residual of the network training in the healthy mode. Therefore, when a fault has occurred the corresponding faulty network residual will be zero and the fault can be isolated. In [10] an isolation approach was presented by using a self organizing map (SOM) network followed by a linear vector quantization (LVQ) network to isolate the faults in the voltage, current and torque of a satellite. In [11] a recurrent neural network was used to detect and isolate the faults in the satellite actuators where the diagnostic scheme was designed to isolate three thruster faults occurring in the first, the second and the third pair of the satellite actuators.

1.3.2 Engine Fault Diagnosis

Aircraft engine fault diagnosis has been a matter of wide interest in recent years for performing tasks such as preventing catastrophic failures and costly component damages, increasing the flight safety by early detection of engine malfunctions, reducing time for manual fault isolation and reducing delays and cancellations by performing timely on-wing maintenance. Engine fault diagnosis algorithms can be classified into

model-based or intelligent-based (process history-based) approaches or a hybrid approach of both model-based and intelligent-based methods. Both model-based and history-based techniques have been extensively studied in the literature for health monitoring of aircraft jet engines. Kalman filters are used as one of the most popular tools in the model-based fault diagnosis that has been extensively studied in the literature [12] -[13]. A multiple model fault detection and isolation scheme for the jet engines is proposed in [13], a bank of linear kalman filters is designed where each filter corresponds to a specific operating mode of the engine and a hierarchical fault diagnosis architecture is proposed for both single and concurrent faults.

Generally engine performance is represented by commonly called health parameters [14]. These parameters are not directly measured and can only be estimated by the measured data. This procedure is often called the gas path analysis (GPA) [14]. Gas path analysis is used to monitor the health indicators of the engine. In this method the health parameters of the engine would be analyzed to monitor the health status of the engine. Common engine faults consist of various anomalies and issues such as blade erosion and corrosion, excess clearances or plugged nozzles and foreign object damage (FOD). These faults result in changes in the thermodynamic performance of the engine as measured by compressor flow capacities and adiabatic efficiencies [15].

Various approaches are used to estimate the health parameters in the gas path analysis, where the most popular methods have been Kalman filter [16] [17], least squares methods [18], fuzzy logic [19], genetic algorithms [20], and bayesian belief networks [21]. Yan *et al.* [22] applied a fusion approach of multiple classifiers for a gas path fault diagnosis. Urbain [23] employed a gas path analysis to isolate the single and multiple faults in the turbine engine.

An observer-based fault detection and isolation scheme was presented by Patton *et*

al. [24]. In their study the observers are used to generate diagnostic residual signals. They also outlined ideas for improving the robustness properties of the diagnostics system.

A condition-based monitoring system was constructed in [25] by using a qualitative model-based approach. In their approach, intelligent computerized systems monitor gas turbines to satisfy maintenance needs based on the turbine's condition rather than on a fixed number of operating hours. They showed that the developed system cuts costs and improves performance significantly. Recently artificial intelligence approaches such as neural networks [26], and Bayesian networks [21], fuzzy logic [27], have also been applied to the GPA.

Although, model-based techniques have their advantages in terms of on-board real-time applications, their reliability for health monitoring often decreases as the system complexity and modelling uncertainties increase. In contrast, data-driven approaches mostly rely on real-time or historical data from the engine measurements, and do not require a detailed mathematical model of the engine. Neural networks are popular methods that are widely used in the aircraft engine fault diagnosis [28, 20, 29, 30, 31]. Support vector machine and fuzzy logic networks have also been introduced for fault diagnosis of jet engine in the literature [32].

1.3.3 Neural Network-Based Fault Diagnosis

A wide range of various neural networks (NNs) has been implemented for the gas turbine engine fault diagnosis, such as the feed forward back propagation neural networks [28] and modular neural network system [20]. In [29] a neural network was used to detect input-output control sensor fault in the single shaft engine. The diagnosis involves dynamic observers along with a neural network to classify observer residuals into different fault classes. Joly *et al.* [30] exploited a NN as a pro-active

engine diagnosis tool which used a large amount of engine data recorded for Rolls Royce engines. A comparative study was conducted in [31] for an effective feature extraction using neural networks for novelty detection in highly dynamic systems such as the gas turbines. In their work several multilayer perceptron, linear networks, radial basis function network as well as Kohonen and probabilistic networks were constructed and trained. Multiple neural networks were utilized in [33] along with GPA method to isolate component-and-sensor faults in the engine. Several hybrid approaches were also used for fault diagnosis purposes such as a hybrid neural-network where part of the model was replaced by influence coefficients [34]. They have reported that the accuracy of such a network was favourable compared with the back-propagation neural network and the Kalman filter approach. Mohammadi *et al.* [35] also applied a hybrid fault diagnosis method to the gas turbine engine using a hybrid automata. The complete survey of neural network-based methods has been reviewed in [36, 37].

Most of the static neural networks are used for off-line and steady state engine fault diagnosis. In [38] a back propagation NN was used for fault diagnosis of the engine where noise-contained training and testing data were generated using influence coefficient matrix. In their work the inputs to the network were selected based on the number of sensors that are available in the engine. The pressure and the temperature of both the turbine and the compressor as well as the shaft speed are considered as inputs of the network. They showed that under high level noise condition NN fault diagnosis can only achieve a 50-60% of success rate. The performance of NN and Kalman filter in the engine fault diagnosis has also been a matter of interest. Volponi *et al.* [39] made a comparative study between the neural networks and Kalman filters in the gas turbine performance diagnosis.

Several other intelligent-based methods have also been used for fault diagnosis of aircraft engine in the literature. Bayesian networks are presented in [40] for the

off-line fault diagnosis of industrial gas turbine in the steady state. A hybrid method of neural networks and a support vector machine (SVM) was used for the health monitoring of the gas turbine engine [32].

1.3.4 Dynamic Neural Networks

Dynamic neural networks have received a lot of attention recently due to their capabilities in modelling nonlinear dynamical systems. The dynamic behaviour is introduced to the neural network by either internal feedback inside the neurons or through the external feedback. The dynamic network with internal feedback normally exploit FIR (finite impulse response) or IIR (infinite impulse response) filters along with the activation function to generate a dynamic characteristic in the network. These filters are built inside the neurons, therefore the neurons of such a network are generally known as dynamic neurons [41, 42, 43, 44, 45, 46, 47, 9]. Another form of dynamic neurons was presented in the time-delay neural networks where the static neuron in modified by augmenting a delay associated with the weights of the network. This modified neuron has now the capability to generate dynamical behaviour in the network [48, 49, 50]. The other category of dynamic neural networks is constructed based on the static neural network concept [7, 51]. In these networks the structure of the network is changed by using some external feedbacks to generate dynamical behaviour. Various structures are presented in the literature for dynamic neural network with external feedbacks where tapped delay lines are used [7] along with a static network to generate dynamics in the structure. Recurrent neural networks are the other form of dynamic networks which employ extensive feedback between the neurons of different layers [51]. Li *et al.* [52] used a recurrent Elman network to construct a fault diagnosis scheme for the reaction wheel on the satellites. They also applied a recurrent neural network for failure detection and isolation in the actuator

and thruster of the satellite [53].

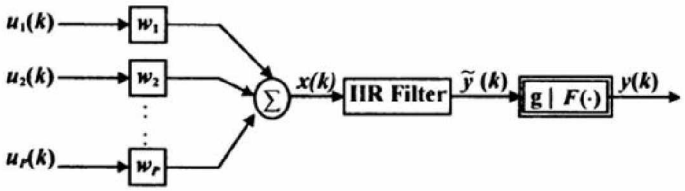


Figure 1.2: A dynamic neuron with an internal IIR filter [46].

Ayoubi [41] presented a class of neuro-dynamic structures that employ an internal feedback between the input and output of the neurons using an IIR filter to generate a local memory characteristic in the overall network based on the concept of dynamic neuron. The structure of this dynamic neural model (DNM) is shown in Figure 1.2. In this structure the filter is located prior to the activation function, however, in [50] another structure was used where the filter is placed after the activation function of the neuron. This will result in a simpler input-output relationship in the network. However, Yazdizadeh [49] showed that in order to model a nonlinear dynamical system using this structure a delayed sample of the output is required as an input to the network.

Dynamic neural networks have been recently introduced to control and perform fault diagnosis due to their capabilities in learning the dynamics of nonlinear systems. In [54] a dynamic neural unit was presented to control an unknown nonlinear system. In [45] a dynamic neural model was used to detect actuator faults in the attitude control subsystem of a satellite. Patan *et al.* [9] employed the DNM for fault detection of real sugar evaporation processes. They used a simulation perturbation stochastic approximation for updating the network parameters. Moreover, a multiple model for healthy and faulty modes was developed to perform the isolation task in the system. Valdes [44] employed the DNM for fault detection and isolation of thrusters in the formation flying of satellites in a series-parallel structure where the network is using

a delayed feedback of the actual output for the training phase and a feedback of the network activity for the recall phase. Korbicz *et al.* [55] used the DNM that is used in this thesis for process modelling and fault diagnosis. They applied their method for identification of the nonlinear system and for a two tank process [55]. Yazdizadeh *et al.* [50] applied their modified form of dynamic neural network for identification of nonlinear dynamical systems. Later on Mohammadi *et al.* [43] applied the dynamic neural network that was developed by [50] for fault detection of the aircraft jet engines. They considered three engine parameters and three separate networks are trained for fault diagnosis of the engine. The series-parallel approach was utilized in the recall phase where the residuals are generated.

Time delayed neural networks (TDNN) is another neuro dynamic structure where a static neuron is modified by introducing a delay along with each weight to generate dynamical behaviour in the network. This network was originally introduced by Wiable [48] and was used for phoneme recognition. The difference between the TDNN and taped delay network is that in the TDNN the delays exist in all layers whereas in the taped delay network delays are only presented to the input layer. These networks are powerful tools for recognition of spatio-temporal patterns and can be trained by using back propagation algorithm.

In the conventional time-delay neural networks the delays are fixed throughout the training, hence, to achieve a better performance adaptive neural networks are presented in the literature [49]. In the adaptive time delayed neural networks (ATDNN) the delays are also updated in the training procedure as well as the weights of the network. Yazdizadeh [49] applied the ATDNN for identification of four different classes of nonlinear dynamic systems. Different training structures were presented to guarantee a stable learning of all the nonlinear classes. The ATDNN was applied for identification of a two-link flexible robot in [56]. TDNN was also used in several applications in

modelling industrial systems. In addition to the original application of the TDNN in speech recognition, TDNN was also applied to several other applications, such as image sequence analysis [57] and trajectory generation [58]. TDNN was also introduced for fault diagnosis in some applications such as automobile transmission gears along with a radial basis function [59] and for damage detection of the railway bridges [60].

1.4 Thesis Contribution

The contributions of the work developed in this thesis are detailed as follows:

- An influence matrix is developed and presented to evaluate the effects of each fault on all the jet engine parameters. Faults are then classified into three different classes of High/Normal and Low severity levels using the influence matrix.

- Twelve different neural networks are trained based on the dynamic neural model approach. The fault detection performance of each network is presented on all the eight faults scenarios that are considered.

- Twelve different neural networks are trained based on the time delay neural networks approach. A series-parallel structure for the TDNN is presented to detect the faults in the jet engine.

- A single-input single-output TDNN was also developed as a fault detection tool and its performance was compared with the other neural networks methods.

- A fault isolation scheme was presented by using the MLP to isolate faults that are occurring in the jet engine.

- An enhanced fault diagnosis scheme was developed and applied to the jet engine by using a multiple residual generator strategy.

- It has been observed through extensive simulations that the SISO TDNN has better isolation capability in comparison with the DNM. Moreover, for fault detection, one can conclude that the series-parallel TDNN has the best performance as compared to the two other networks in detecting faults as small as 2 percent on average.

1.5 Thesis Outline

The organisation of this thesis is as follow: Chapter 2 includes the background information on the input-output behaviour of both the DNM and the TDNN approaches as well as their updating rules. Moreover, the engine mathematical model and the equations of a dual spool turbo fan engine are presented. The types of the faults that are considered in this thesis and the influence coefficient matrix are also presented in Chapter 2. The proposed fault detection strategies by using the dynamic neural networks as well as the simulation results and discussions are presented in Chapter 3. Chapter 4 presents our proposed fault isolation scheme and the corresponding simulation results. The conclusions followed by the future work are presented in Chapter 5.

Chapter 2

Background Information

2.1 Dynamic Neural Model Approach

A neuro-dynamic structure is presented in this section based on the Ayoubi model [41], which in thesis is called dynamic neuron model (DNM). This network is constructed based on the concept of a dynamic neuron and is constructed by adding internal dynamics which makes the neuron's activity depend on the internal neuron states. This can be obtained 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.

The input-output relations of the network can be obtained by three main modules. The first module is the adder, which is similar to the static neuron, that is

$$x(k) = \mathbf{W^T u(k)} = \sum_{\mathbf{p=1}}^{\mathbf{r}} \mathbf{w_p u_p(k)} \tag{2.1.1}$$

where $\mathbf{W} = [w_1, w_2, ...w_r]^T$ denotes the input-weight vector, r denotes the number of inputs, and $\mathbf{u}(k) = [u_1(k), u_2(k), ...u_r(k)]^T$ denotes the input vector (T denotes the transpose operator). The output of the adder is passed through an IIR filter by which

a dynamic mapping is created between the input and the output of the neuron. For example, by applying a second order filter, the output of the filter and filter transfer function would be respectively:

$$\widetilde{y}(k) = -a_1\widetilde{y}(k-1) - a_2\widetilde{y}(k-2) + b_0x(k) + b_1x(k-1) + b_2x(k-2) \tag{2.1.2}$$

$$H(q^{-1}) = \frac{b_0 + b_1q^{-1} + b_2q^{-2}}{1 + a_1q^{-1} + a_2q^{-2}} \tag{2.1.3}$$

where $x(k)$ is the filter input, $\widetilde{y}(k)$ is the filter output, $\mathbf{a} = [a_1, a_2]^T$ and $\mathbf{b} = [b_0, b_1, b_2]^T$ are the feedback and feed-forward coefficients of the filter, respectively and q is the shift operator.

The neuron output can now be expressed as:

$$y(k) = F(g.\widetilde{y}(k)) \tag{2.1.4}$$

where $F(.)$ is a nonlinear activation function that produces the neuron output and g is the parameter of the activation function defining its slope. Introducing g to the activation function can be very helpful particularly in case of nonlinear activation functions such as hyperbolic tangent or sigmoidal [61]. Due to the adaptive nature of this parameter the neuron can better model a biological neuron.

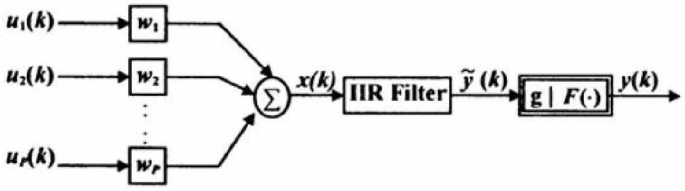


Figure 2.1: A dynamic neuron having an internal IIR filter.

Let us consider an L-layered network as shown in Figure 2.2 using the dynamic neurons that are described by a differentiable activation function $F(.)$. Let N_l denote the number of neurons in the l-th layer, $O_n^l(k)$ denote the output of the nth neuron

of the lth layer, and $u_p^l(k)$ denote the input of the lth layer, generated from the p-th neuron of the previous layer at discrete times k ($l = 1, ..., L$; $n = 1, ..., N_l$). The activity of the nth neuron in the lth layer is defined by:

$$O_n^l(k) = F[g_n^l \widetilde{y}(k)] = F[g_n^l(\sum_{d=0}^{D} b_{dn}^l \sum_{p=1}^{N_{l-1}} w_{np}^l u_p^l(k-d) - \sum_{d=1}^{n} a_{dn}^l \widetilde{y}_n^l(k-d))] \quad (2.1.5)$$

It can be seen from equation 2.1.5 that the network outputs depend on the past outputs $\widetilde{y}(k-1)$, $\widetilde{y}(k-2)$, ..., $\widetilde{y}(k-n)$. Since an activation function $F(.)$ is an invertible function (e.g. tangent hyperbolic), then network outputs will also depend on past outputs $y(k-1)$, $y(k-2)$, ... , $y(k-n)$. Consequently, the expression for the last layer outputs is given by equation (2.1.6), where $\Gamma(.)$ is a nonlinear function representing the overall network map. One should note that normally the activation function of the output layer is linear in this case $\widetilde{y}(k) = y(k)$. This shows that the network outputs are nonlinear functions of the inputs and their delays as well as the previous outputs samples, that is

$$O_n^L(k) = \Gamma[y(k-1), ..., y(k-m_s), u(k), u(k-1), .., u(k-n_s)] \quad (2.1.6)$$

2.1.1 Extended Dynamic Back-Propagation Algorithm

The proposed dynamic neural model parameters will be updated using an extended dynamic back-propagation which is similar to the static back-propagation algorithm with some modifications that makes it applicable to the dynamic neuron. The main objective of the learning process is to adjust all the unknown network parameters so that the nonlinear system can be identified by the proposed dynamic neural network by using a given training set of input-output pairs. The unknown network parameters are $\mathbf{w}$, $\mathbf{a}$, $\mathbf{b}$, $\mathbf{g}$, where $\mathbf{w} = [w_{np}^l]_{l=1,...,L;n=1,...,N_l;p=1,...,N_{l-1}}$ is the weight matrix, $\mathbf{a} =$

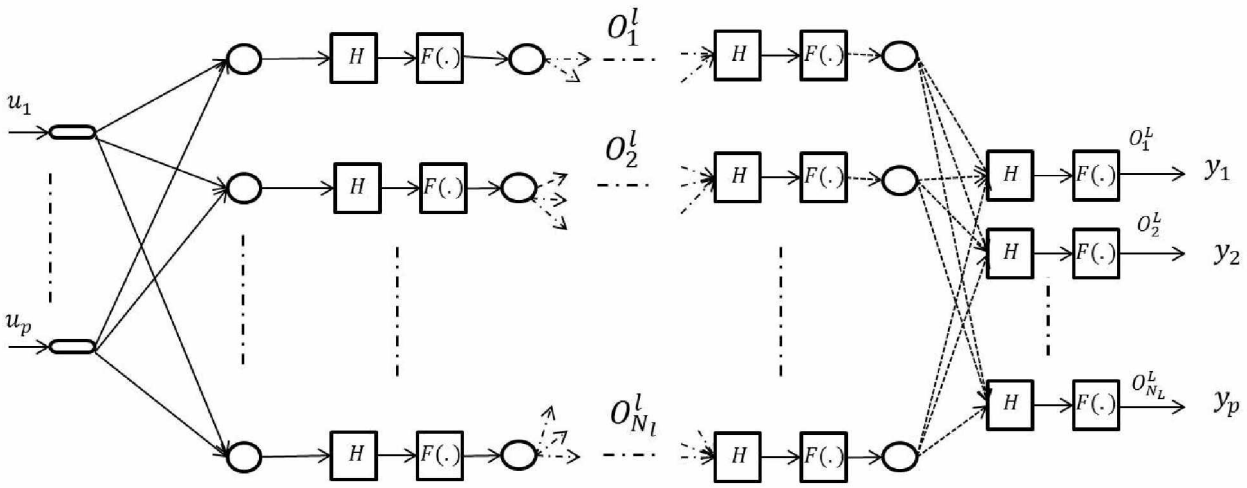


Figure 2.2: Dynamic neural network architecture.

$[a_{dn}^l]_{l=1,...,L;n=1,...,N_l;d=1,...,D}$ and $\mathbf{b} = [b_{dn}^l]_{l=1,...,L;n=1,...,N_l;d=1,...,D}$ are the filter parameters matrices, where D denotes the order of the IIR filter, and $\mathbf{g} = [g_{nd}^l]_{l=1,...,L;n=1,...,N_l}$ denotes the slope parameter matrix.

As stated earlier to adjust the network parameters, pairs of the healthy input and output data sets are used. The back-propagation error is widely applied for the purpose of training static networks. Its extension to dynamic applications are known as the extended dynamic back-propagation algorithm (EDBF) [55]. The objective of the EDBF is to adjust all the parameters of the network vector $\gamma = [w, A, B, g]$ to minimize the performance index J based upon the error function $e(k)$ which is defined by:

$$J = \frac{1}{2}\sum_{i=1}^{N}(e_i(k))^2 = \frac{1}{2}\sum_{i=1}^{N}(y_i^d(k) - y_i(k))^2 \tag{2.1.7}$$

where $e_i(k)$ denotes the error of the i^{th} output defined as difference between the desired response $y_i^d(k)$ and the actual response $y_i(k)$ and N denotes the number of outputs.

The adjustment rule of the EDBA for the network parameters has following form:

$$\gamma_n^l(k+1) = \gamma_n^l(k) + \eta\delta_n^l(k)S_{\lambda n}^l \tag{2.1.8}$$

where γ represents the unknown generalized parameter vector, η is the learning rate, S denotes the sensitivity function for the elements of the unknown generalized parameter vector and δ is the generalized output error which is described separately for the output layer and the hidden layer as given below:

- Hidden layers generalized output error

$$\delta_n^l = \sum_{z=1}^{N_{l+1}} (\delta_n^{l+1}(k) g_z^{l+1} b_{0z}^{l+1} w_{zn}^{l+1}) F'(\tilde{y}_{1n}^l) \tag{2.1.9}$$

- Output layer generalized output error

$$\delta_n^L = e_n(k) F'(\tilde{y}_{1n}^L) \tag{2.1.10}$$

The sensitivity function S is also defined as follows:

- Sensitivity with respect to the weight parameters

$$S_{w_{pn}}^{\boldsymbol{l}}(k) = g_n^l [\sum_{i=0}^{m} b_{in}^l u_p^l(k-i) - \sum_{i=1}^{m} a_{in}^l S_{w_{pn}}^{\boldsymbol{l}}(k-i)] \tag{2.1.11}$$

- Sensitivity with respect to the feedback parameters

$$S_{a_{in}}^l(k) = -g_n^l \tilde{y}_n^l(k-i) \tag{2.1.12}$$

- Sensitivity with respect to the feed-forward parameters

$$S_{b_{in}}^l = g_n^l X_n^l(k-i) \tag{2.1.13}$$

- Sensitivity with respect to the slope parameters

$$S^l_{g_n}(k) = \tilde{y^l_n}(k) \tag{2.1.14}$$

Based on equations (2.1.8)-(2.1.14) the updating law for each network parameters can be rewritten as follows:

- Hidden layers parameters

  - Weight parameters:

$$w^l_{np}(k+1) = w^l_{np}(k) + \eta[\sum_{z=1}^{N_{l+1}}(\delta^l_n(k) g^{l+1}_z b^{l+1}_{0z} w^{l+1}_{zn}) F'(\tilde{y}^l_{1n}(k))] \\ g^l_n[\sum_{i=0}^{m} b^l_{in} u^l_p(k-i) - \sum_{i=1}^{m} a^l_{in} S^l_{w_{pn}}(k-i)] \tag{2.1.15}$$

  - Filter feedback parameters:

$$a^l_n(k+1) = a^l_n(k) - \eta[\sum_{z=1}^{N_{l+1}}(\delta^l_n(k) g^{l+1}_z b^{l+1}_{0z} w^{l+1}_{zn}) F'(\tilde{y}^l_{1n}(k))] g^l_n \tilde{y}^l_n(k-i) \tag{2.1.16}$$

  - Filter feed-forward parameters:

$$b^l_n(k+1) = b^l_n(k) - \eta[\sum_{z=1}^{N_{l+1}}(\delta^l_n(k) g^{l+1}_z b^{l+1}_{0z} w^{l+1}_{zn}) F'(\tilde{y}^l_{1n}(k))] g^l_n X^l_n(k-i) \tag{2.1.17}$$

  - Slope parameters:

$$g^l_n(k+1) = g^l_n(k) - \eta[\sum_{z=1}^{N_{l+1}}(\delta^l_n(k) g^{l+1}_z b^{l+1}_{0z} w^{l+1}_{zn}) F'(\tilde{y}^l_{1n}(k))] \tilde{y}(k) \tag{2.1.18}$$

- Output layer parameters

- Weight parameters:

$$w^l_{np}(k+1) = w^l_{np}(k) + \eta[e_n(k)F'(\tilde{y}^l_{1n}(k))]g^l_n[\sum_{i=0}^{m} b^l_{in}u^l_p(k-i) - \sum_{i=1}^{m} a^l_{in}S^l_{w_{pn}}(k-i)] \tag{2.1.19}$$

- Filter feedback parameters:

$$a^l_n(k+1) = a^l_n(k) - \eta[e_n(k)F'(\tilde{y}^l_{1n}(k))]g^l_n\tilde{y}^l_n(k-i) \tag{2.1.20}$$

- Filter feed-forward parameters:

$$b^l_n(k+1) = b^l_n(k) - \eta[e_n(k)F'(\tilde{y}^l_{1n}(k))]g^l_nX^l_n(k-i) \tag{2.1.21}$$

- Slope parameters:

$$g^l_n(k+1) = g^l_n(k) - \eta[e_n(k)F'(\tilde{y}^l_{1n}(k))]\tilde{y}(k) \tag{2.1.22}$$

2.2 Time Delayed Neural Network Approach

Time delayed neural networks was primarily introduced by [48] for phenomena recognition. The structure of this network is similar to a static network with some slight modifications. In conventional static neuron the activity value of the neuron is the weighted sum of the inputs whereas in the TDNN certain delays are introduced to all the weights. Figure 2.3 depicts the structure of such a dynamic neuron. In the TDNN the number of branches that connect each two neurons may be more than one depending on the structure of the network while in a static neuron there is only one connection between each two neurons. In our proposed TDNN the number of connections between each two neurons is equal to the number of delays associated to all the

weights of a layer. The number of delays in our proposed TDNN is assumed to be fixed for the process. The capability of the TDNN for representing an input-output map of a nonlinear system in shown analyticity below.

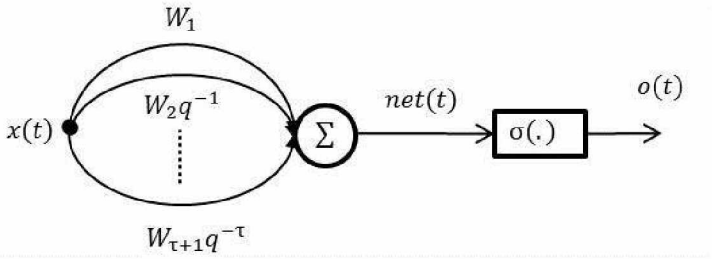


Figure 2.3: Time delay neuron.

The TDNN is constructed based on the concept of the dynamic neuron that is presented in Figure 2.3. The network consists of L layers with N_l neurons in the l^{th} layer. Since we are dealing with identification of a nonlinear dynamical system of the aircraft jet engine and the structure of the engine we consider for our investigation is in the form of a single-input single-output (SISO) system our network will also have one input and one output. However, as shown subsequently for a series-parallel structure a multi-input single-output (MISO) network will be used. Every bounded, monotonically increasing and differentiable nonlinear function may be used as an activation function. Specially we use a tangent hyperbolic activation function for the hidden layer neurons and a linear activation function for the output layer .

We use the following notation to present the input-output equations of the network. The output of the j^{th} neuron in the l^{th} layer at time t is denoted by $o_j^l(t)$. The delay associated to the weights of l^{th} layer is fixed and is denoted by τ_{max}^l where τ^l is the delay between the neurons of layer l and neurons of $l-1$ layer and is varying from $0, 1, ..., \tau_{max}^l$ The weights connecting the j^{th} neuron in the l^{th} layer to the i^{th} neuron in the $l-1^{th}$ layer are denoted by $w_{ji\tau^l}^l$. Note that j varies from 1 to N_l and i varies from 1 to N_{l1}. The equations for a typical neuron are now written as:

$$net_j^l = \sum_{i=1}^{N_{l-1}} \sum_{\tau^l=0}^{\tau_{max}^l} w_{ji\tau^l}^l o_i^{l-1}(t-\tau^l) \tag{2.2.1}$$

$$o_j^l(t) = \sigma(net_j^l(t)) \tag{2.2.2}$$

where net_j^l is the weighted input of the j^{th} neuron in the l^{th} layer at time t. The output of the i^{th} neuron in the first layer is given by

$$o_j^1(t) = \sigma(\sum_{\tau^1=0}^{\tau_{max}^1} w_{j1\tau^1}^1 x(t-1-\tau^1)) \tag{2.2.3}$$

where x_{t-1} is the external input to the network. If o_j^1 is substituted in o_j^2 it yields:

$$o_j^2(t) = \sigma[\sum_{i=1}^{N_1} \sum_{\tau^2=0}^{\tau_{Max}^2} w_{ji\tau^2}^2(\sigma(\sum_{\tau^1=0}^{\tau_{Max}^1} w_{i1\tau^1}^1 x(t-1-\tau^1-\tau^2)))] \tag{2.2.4}$$

Since i varies from 1 to N_1 and τ^1 and τ^2 vary from 0 to τ_{max}^1 and 0 to τ_{max}^2, respectively, then $o_j^2(t)$ can be written as

$$o_j^2(t) = \sigma[x(t-1), x(t-2)..., x(t-1-\tau_{max}^1-\tau_{max}^2)] \tag{2.2.5}$$

It can be shown by extension that the output of the last layer is given by:

$$o_j^L(t) = \Gamma[x(t-1), x(t-2), ..., x(t-1-\tau_{max}^1-\tau_{max}^2...-\tau_{max}^l)] \tag{2.2.6}$$

where $\Gamma[.]$ is a nonlinear mapping that the network realizes.

The proposed time-delay neural network parameters are updated by using the Levenberg-Marquardt algorithm. Levenberg-Marquardt was developed in [62] to obtain a second-order training speed while not requiring the computation of the Hessian matrix. Provided that the performance index is in the form of a sum of squares, the

Hessian matrix can be represented as:

$$H = J^T J \tag{2.2.7}$$

and the gradient can be computed as

$$g = J^T e \tag{2.2.8}$$

where J denotes the Jacobian matrix and e denotes the neural network error. The Jacobian matrix contains the first derivatives of the network errors with respect to the biases and weights and is computed by using the ordinary backpropagation algorithm which is generally less complex than computing the Hessian matrix. In the Levenberg-Marquardt algorithm a Newton-like update approximation is introduced to the Hessian matrix as follows:

$$x_{k+1} = x_k - [J^T J + \mu I]^{-1} J^T e \tag{2.2.9}$$

where this approximation would be equivalent to the Newton's method [63] when the scalar μ is zero. When μ is large, this approximation would be similar to the gradient descent algorithm having a small step size. Generally, the Newton's method is more accurate near the error function minimum and also has a faster performance rate. Hence, it is desirable to shift towards the Newton's method as quickly as possible. Thus, μ is decreased after each step and is increased only when a step increase in the performance function is observed. Using this approximation, one can be assured that the performance function is always reduced at each iteration of the algorithm.

Based on the equation (2.2.6) one can conclude that this general TDNN has the capability of learning the dynamical system depending on the past inputs. Hence, it has been used extensively in the literature for applications such as trajectory planning,

speech recognition and time series prediction [57, 59, 58]. However, as shown in equation (2.2.6) the final input-output map does not include past samples of the output, hence, theoretically the general structure of TDNN cannot be used for identification of nonlinear class of dynamic system depending on both the delayed samples of the inputs and outputs. Yazdizade [50], presented four different structures of TDNN for identifying different forms of nonlinear dynamic systems. In the following section we will study the series-pararell structure which was used for representing a more general class of nonlinear systems in the form of:

$$y(t) = f[y(t-1), y(t-2), ..., y(t-Ns), u(t-1), u(t-2), ..., u(t-Ms)] \quad (2.2.10)$$

We will see that with a small modification in the structure of the TDNN it is possible to identify a dynamical system in the most general form.

2.2.1 Series-Parallel Structure

In the series-parallel structure a delayed sample of the output is used as an additional input which yields the structure of the network as shown in Figure 2.4. This figure depicts the structure of our proposed neuro-dynamic network, where a feedforward TDNN is utilized to approximate the nonlinear function $f[.]$ of the engine dynamics. In this structure a delayed sample of the output is utilized as an input to the network. Assuming that this input (that is the delayed of the engine output) is denoted as x_2, and the original input (of the engine) as x_1, the output of 1^{st} layer neuron is given by:

$$o_j^1(t) = \sigma[\sum_{\tau^1=0}^{\tau_{max}^1} w_{j1\tau^1}^1 x_1(t-1-\tau^1) + \sum_{\tau^1=0}^{\tau_{max}^1} w_{j2\tau^1}^1 x_2(t-1-\tau^1)] \quad (2.2.11)$$

If o_i^1 is substituted in o_i^2 following along the same steps for the general TDNN it can be shown that the output of the second layer neuron can be written by:

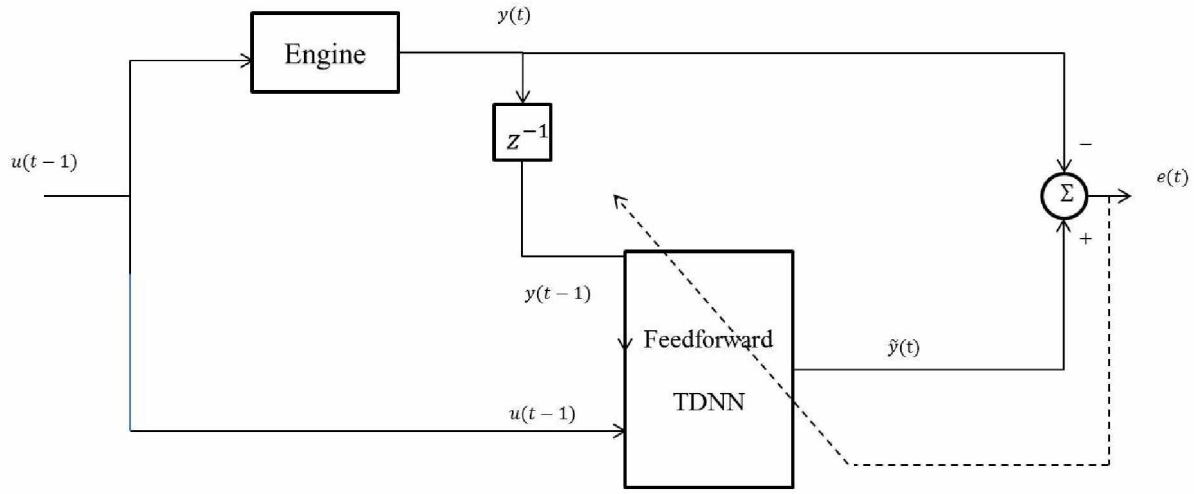


Figure 2.4: Training of the TDNN in the series-parallel structure.

$$o_j^2(t) = \sigma[x_1(t-1), ...x_1(t-1-\tau_{max}^1-\tau_{max}^2), x_2(t-1), ...x_2(t-1-\tau_{max}^1-\tau_{max}^2) \quad (2.2.12)$$

and by substituting $x_2(t)$ with y(t) and x_1 with $u(t)$ in the above equation, the output of the last layer can be written as:

$$o_j^L(t) = \Gamma[u(t-1), u(t-2), ..., u(t-1-\tau_{max}^1-\tau_{max}^2...-\tau_{max}^l), \quad (2.2.13)$$
$$y(t-1), y(t-2), ..., y(t-1-\tau_{max}^1-\tau_{max}^2...-\tau_{max}^l)]$$

The TDNN Recall Architecture

As presented in the previous subsection a series-parallel architecture has the ability to identify a nonlinear dynamic system and represents its dynamics. However, for the fault diagnosis problem, some modifications should be made to perform fault detection of the system. In the fault diagnosis the output of the system is the signal

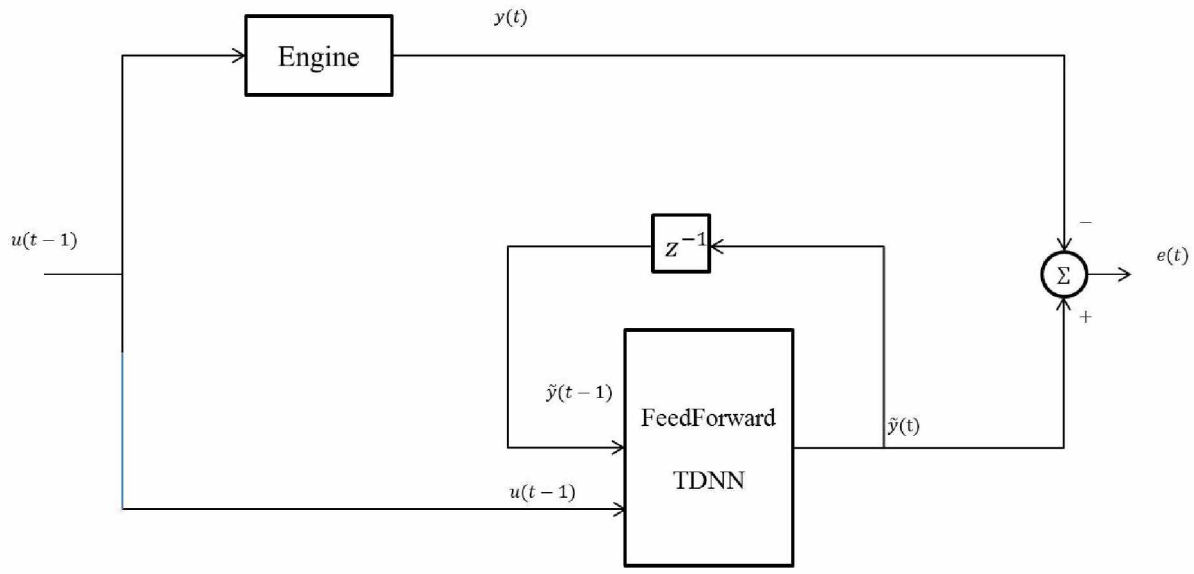


Figure 2.5: The TDNN structure in the recall phase.

which is being monitored to detect the occurrence of change or fault and the neural network is modelling the healthy system. Therefore, the residual signal which is the difference between the network output and the actual system output is needed.

When a fault occurs in the system, since the delayed sample of the output is being fed to the neural network the output of the network will not then represent the healthy engine. For this reason we need to use a different structure in the recall phase. In this phase the network has already been trained and the weights and the other network parameters have already been set to fixed values and are frozen. Hence, one can assume that the difference between the actual output of the healthy system and its estimated value by the neural network is sufficiently small. Therefore, one can use the delay of the network produced output instead of the actual system output as an input to the neural network.

Figure 2.5 depicts the structure of the resulting network in the recall phase. Note that in this structure the neural network is assumed to have been well trained since differences between the network output and the actual output could then cause a

large steady state error in the estimation error.

Moreover, unlike the DNM approach presented in the previous section, this structure is more useful for an off-line fault diagnosis. This is due to the fact that the structure of the network should be changed after its training phase and it cannot be directly used in an on-line diagnosis and should be trained first and then the modified structure can be used in the diagnosis system.

2.3 Jet Engine Mathematical Model

In this section a nonlinear mathematical model for a dual spool jet engine (Figure 2.6) is presented based on the work of Naderi *et al.* [64]. For transient response model of the jet engine, rotor and volume dynamics are considered in the equations of motions. Considering the volume dynamics, the engine components are assumed to be volume-less and a volume among the components is considered to model an imbalance mass flow rate. This modelling consideration allows the elimination of large algebraic loops and provides a reasonable ground for development of a generic and a modular model of the jet engine dynamics. The modules and the information flow among the various components for a dual spool engine are shown in Figure 2.7. In the following, detailed mathematical expressions corresponding to the engine dynamics as well as each specific component are presented.

2.4 General Overview of the Components of the Engine

A turbofan engine is a modern variation of the common gas turbine engine. In the turbofan engine, the core engine which is the combination of high pressure compressor,

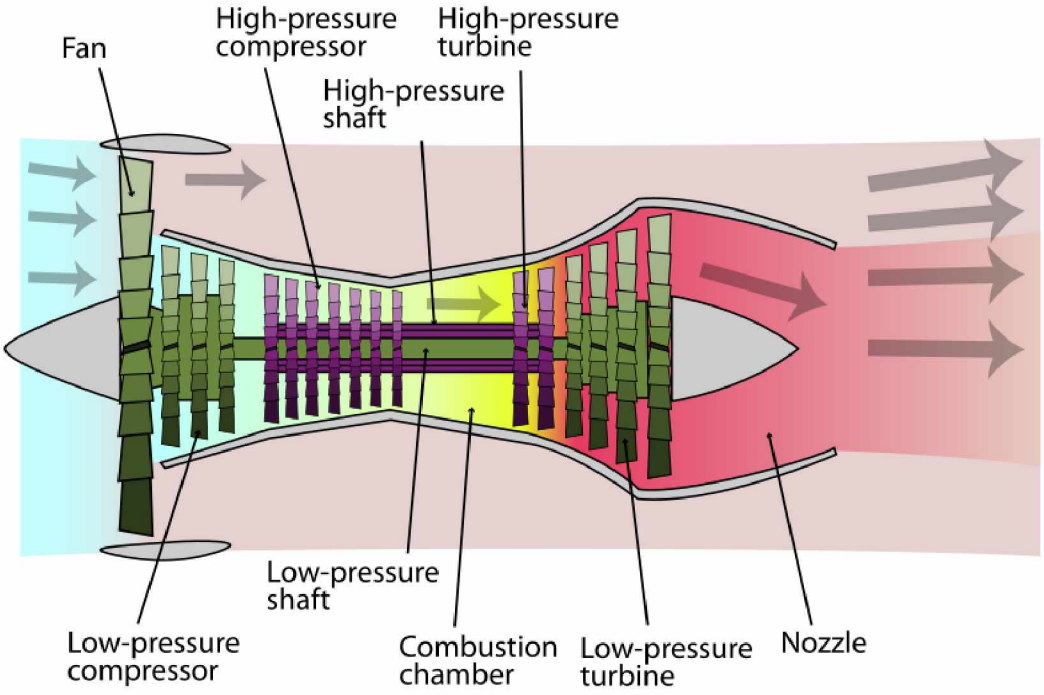


Figure 2.6: The dual spool turbofan engine .

high pressure turbine, and combustion chamber is surrounded by a fan which is in the front and an additional low pressure turbine at the rear [65]. The fan and the turbine are composed of many blades and are connected by a shaft. In this type of engine some of the fan blades turn with the shaft connected to the low pressure compressor and the low pressure turbine while some other fan blades remain stationary. The fan shaft passes through the core shaft (high speed shaft) and makes a dual spool turbofan engine. One spool connecting the fan and the low pressure compressor to the low pressure turbine and the other corresponds to the core engine and is connecting the high pressure compressor to the high pressure turbine.

In the turbofan engine the incoming air is captured by the engine inlet. A part of the incoming air passes through the fan and enters the high pressure compressor and then into the combustion chamber where it will be mixed with fuel before the combustion occurs. The hot exhaust passes through the core and the fan turbines

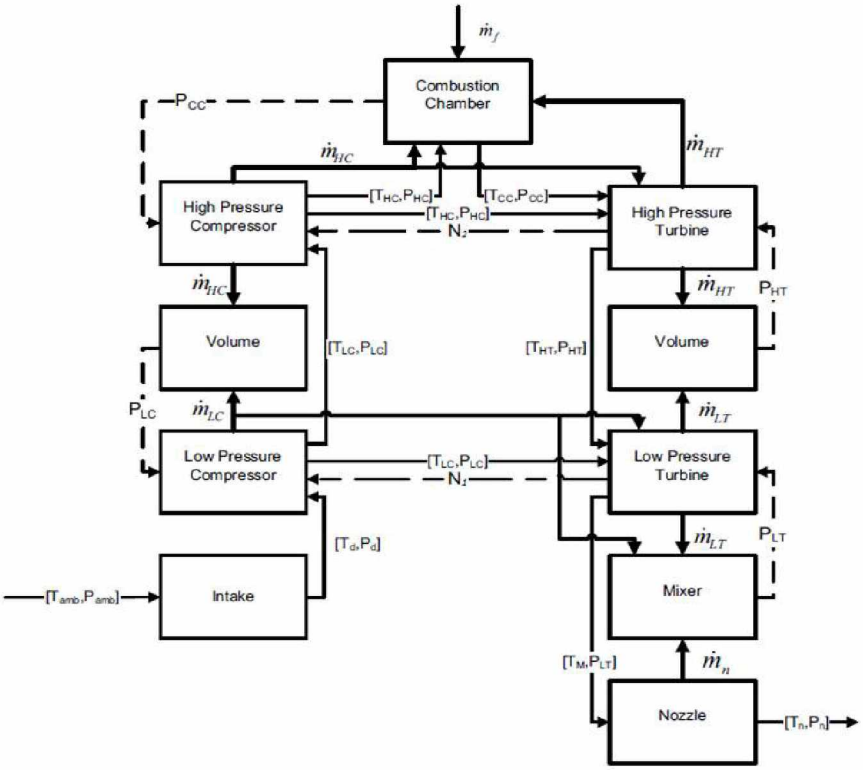


Figure 2.7: Engine modules and the information flowchart [63].

and then towards the nozzle. This airflow which is used in the dynamic equations of the turbofan engine is called the core airflow and is denoted by $\dot{m}_c$. The rest of the incoming air passes through the fan and bypasses or around the engine. The part of the air that goes through the fan has a velocity that has been increased from its free stream value. This airflow is denoted by $\dot{m}_f$ and is the fan flow, or the bypass flow. The ratio of $\dot{m}_f$ to $\dot{m}_c$ is called the bypass ratio.

2.4.1 Rotor Dynamics

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

$$\frac{dE}{dt} = \eta_{mech} W_T - W_C \tag{2.4.1}$$

where $E = \frac{J(\frac{N.2\pi}{60})^2}{2}$, W_T denotes the power generated by turbine, W_C denotes the power consumed by compressor, η_{mech} denotes the mechanical efficiency, and N denotes the rotational speed.

2.4.2 Volume Dynamics

As mentioned above, 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 homogenous properties over volumes, this dynamics can be described by the following equation:

$$\dot{P} = \frac{RT}{V}(\sum \dot{m}_{in} - \sum \dot{m}_{out}) \tag{2.4.2}$$

where P denotes the pressure, T denotes the temperature, V denotes the volume, R denotes the gas constant, $\dot{m}_{in}$ denotes the input mass flow, and $\dot{m}_{out}$ denotes the output mass flow.

2.4.3 Components

Compressor

The compressor behaviour, as a quasi-steady component, is determined by using the compressor performance map which is obtained from the commercial software package GSP (gas turbine simulation program) [66]. The GSP, a component based modelling environment, is a tool for the gas turbine engine performance analysis. GSP's flexible object-oriented architecture allows steady-state and transient simulation of any gas turbine configuration by using a user-friendly drag and drop feature. Gas turbine configurations are simulated by establishing a particular arrangement of engine component models in a model window. 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 ($\eta_C$) from a performance map by using a proper interpolation technique, where $\theta = T_i/T_0$ and $\delta = P_i/P_0$, 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:

$$T_o = T_i \left[1 + \frac{1}{\eta_C}(\pi_C^{\frac{\gamma-1}{\gamma}} - 1)\right] \tag{2.4.3}$$

$$W_C = \dot{m}_C c_p (T_o - T_i) \tag{2.4.4}$$

where T_0 denotes the output temperature, T_i denotes the input temperature, $\dot{m}_c$ denotes the compressor mass flow rate, and c_p denotes the specific heat at constant pressure.

Turbine

Similar to the compressor, the turbine behaviour is also determined by using the turbine performance map (from the software package GSP [66]). 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 ($\eta_T$) 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 are obtained as follows:

$$T_o = T_i \left[1 - \eta_T(1 - \pi_T^{\frac{\gamma-1}{\gamma}})\right] \tag{2.4.5}$$

$$W_T = \dot{m}_T c_p (T_i - T_o) \tag{2.4.6}$$

Combustion Chamber

The dynamics inside the combustion chamber is governed by equations (2.4.7) and

(2.4.8). In fact, 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. In other words, we have

$$\dot{P}_{CC} = \frac{P_{CC}}{T_{CC}}\dot{T}_{CC} + \frac{\gamma R T_{CC}}{V_{CC}}(\dot{m}_C + \dot{m}_f - \dot{m}_T) \tag{2.4.7}$$

$$\begin{aligned}\dot{T}_{CC} =& \frac{1}{c_v m_{CC}}[(c_p T_C \dot{m}_C + \eta_{CC} H_u \dot{m}_f - c_p T_{CC} \dot{m}_T) - \\ & c_v T_{CC}(\dot{m}_C + \dot{m}_f - \dot{m}_T)]\end{aligned} \tag{2.4.8}$$

where P_{CC} and T_{CC} denote the pressure and temperature of the combustion chamber, respectively, $\dot{m}_T$ denotes the turbine mass flow rate, CC denotes the combustion chamber, $\dot{m}_f$ denotes the fuel flow rate, γ denotes the heat capacity ratio, c_v denotes the specific heat at constant volume, H_u denotes the fuel specific heat, and R denotes the gas constant.

2.4.4 Nonlinear Equations of Motion

In this subsection, a set of nonlinear equations corresponding to a dual spool jet engine is provided. In the engine intakes, by assuming adiabatic process, the pressure and the temperature are computed as follows:

$$\frac{P_d}{P_{amb}} = \left[1 + \eta_d \frac{\gamma - 1}{2} M^2\right]^{\frac{\gamma}{\gamma - 1}} \tag{2.4.9}$$

$$\frac{T_d}{T_{amb}} = 1 + \frac{\gamma - 1}{2} M^2 \tag{2.4.10}$$

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 (2.4.2). The rotational speed (N_2) is obtained from the solution to equation (2.4.1) 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 map, therefore the temperature rise can be obtained from equation (2.4.3). Similar to the low pressure compressor, for high pressure compressor, pressure is obtained from the volume dynamics that is described by equation (2.4.7). The rotational speed (N_1) is obtained from equation (2.4.1) 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 calculated by using the volume dynamics after the low pressure turbine. The mass flow rate of the nozzle is computed as follows. If condition (2.4.11) holds, the mass flow rate is obtained from equation (2.4.12), otherwise, it is determined from equation (2.4.13). In other words, we have

$$\frac{P_{amb}}{P_{n_i}} < \left[1 + \frac{1-\gamma}{\eta_n(1+\gamma)}\right]^{\frac{\gamma}{\gamma-1}} \tag{2.4.11}$$

$$\frac{\dot{m}_n\sqrt{T_{n_i}}}{P_{n_i}} = \frac{u}{\sqrt{T_{n_i}}}\frac{A_n}{R}\frac{P_{amb}}{P_{n_i}}\frac{T_{n_i}}{T_{n_o}} \tag{2.4.12}$$

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

$$\frac{\dot{m}_n\sqrt{T_{n_i}}}{P_{n_i}} = \frac{u}{\sqrt{T_{n_i}}}\frac{A_n}{R}\frac{P_{crit}}{P_{n_i}}\frac{T_{n_i}}{T_{crit}} \tag{2.4.13}$$

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 $T_{n_i} = T_M$ which is obtained from the energy balance in the mixer as follows:

$$T_M = \frac{\dot{m}_{LT} T_{LT} + \beta \dot{m}_{LC} T_{LC}}{\dot{m}_{LT} + \beta \dot{m}_{LC}} \tag{2.4.14}$$

where n denotes the nozzle, n_i denotes the nozzle input, u denotes the speed, A denotes the area, n_o denotes the nozzle output, $crit$ denotes the critical and LT and LC denote the low pressure turbine and the low pressure compressor respectively.

To summarize, the nonlinear set of governing equations of the dual spool jet engine are given as follows:

$$\dot{T}_{CC} = \frac{1}{c_v m_{CC}} [(c_p T_C \dot{m}_C + \eta_{CC} H_u \dot{m}_f - c_p T_{CC} \dot{m}_T) - c_v T_{CC} (\dot{m}_C + \dot{m}_f - \dot{m}_T)] \tag{2.4.15}$$

$$\dot{N}_1 = \frac{\eta^2_{mach} \dot{m}_{LT} c_p (T_{HT} - T_{LT}) - \dot{m}_{HC} c_p (T_{LC} - T_d)}{J_1 N_1 (\frac{\pi}{30})^2} \tag{2.4.16}$$

$$\dot{N}_2 = \frac{\eta^1_{mach} \dot{m}_{HT} c_p (T_{CC} - T_{HC}) - \dot{m}_{HC} c_p (T_{HC} - T_{LC})}{J_2 N_2 (\frac{\pi}{30})^2} \tag{2.4.17}$$

$$\dot{P}_{LT} = \frac{R T_M}{V_M} (\dot{m}_{LT} + \beta \dot{m}_{LC} - \dot{m}_n) \tag{2.4.18}$$

$$\dot{P}_{CC} = \frac{P_{CC}}{T_{CC}} \dot{T}_{CC} + \frac{\gamma R T_{CC}}{V_{CC}} (\dot{m}_{HC} + \dot{m}_f - \dot{m}_{HT}) \tag{2.4.19}$$

$$\dot{P}_{LC} = \frac{R T_{LC}}{V_{LC}} ((1 - \beta) \dot{m}_{LC} - \dot{m}_{HC}) \tag{2.4.20}$$

$$\dot{P}_{HT} = \frac{R T_{HT}}{V_{HT}} (\dot{m}_{LT} + \beta \dot{m}_{LC} - \dot{m}_n) \tag{2.4.21}$$

where $\dot{m}_f = PLA \times \dot{m}_f^{max}$, and the PLA (power level angle) is the control signal or

Engine Measured Parameters	Description
LC_T	Temperature in the Low Pressure Compressor
LC_P	Pressure of the Low Pressure Compressor
HC_T	Temperature of the High Pressure Compressor
HC_P	Pressure of the High Pressure Compressor
HT_T	Temperature of the High Pressure Turbine
HT_P	Pressure of the High Pressure Turbine
LT_T	Temperature of the Low Pressure Turbine
LT_P	Pressure of the Low Pressure Turbine
N_1	The rotational speed of the spool connecting the low pressure compressor to the low pressure turbine
N_2	The rotational speed of the spool connecting the high pressure compressor to the high pressure turbine
CC_T	Temperature of the Combustion Chamber
CC_P	Pressure of the Combustion Chamber

Table 2.1: Engine measured parameters.

the input $u = PLA$ and $\dot{m}_f^{max}$ is the maximum fuel flow rate, η_{mach}^1 and η_{mach}^2 denote mechanical efficiencies, and J_1 and J_2 denote the inertia of high and low pressure shafts, respectively. Moreover, all the performance maps of the compressors and the turbines are adopted from the commercial software package GSP [66].

2.5 Engine Data Generation

All the data that is used in this work are generated by using a Simulink mathematical model of the dual spool jet engine model developed in the previous subsection as well as in [64]. The engine model is operating in the cruise mode at the low altitude of $4000m$. The engine is being controlled by the PLA (power level angel). Hence, the fuel flow rate is changing from 40% to 95% of the maximum fuel rate. The initial ambient parameters are set corresponding to the altitude of the cruise mode. This leads to the ambient condition of $T_{amb} = 262$ centigrade and $P_{amb} = 0.63$ atmosphere. The

Mach number is set to 0.7 and is assumed to be fixed in the cruise mode. To ensure that the model is as close as possible to the practical engine operation measurement noise was considered when reading the sensor data.

It is assumed there are 12 measurable parameters in the engine and by changing the input at each stage the sensor value for all the 12 parameters are collected. The parameter set is presented in Table 2.1. However, in reality depending on the engine and the application that the engine is operating the number of the measurable parameters vary. In [13] the following parameters are considered as the measurement vector, namely $[N_1, N_2, HC_P, HC_T, LC_P, LC_T, HT_T, LT_T]$. However, in this thesis we assume all the engine parameters are measurable and accessible to the user.

2.6 Faults in the Jet Engine

The faults in an engine can be classified into sensor faults, actuator faults, and physical or component faults. A sensor fault occurs when there is a sensor read that is different from the actual value. The most common sensor faults are bias, drift, noise, scaling and drop out. On the other hand, component faults may consist of a number of anomalies such as blade erosion, corrosion, tip clearance, fouling, foreign object damage, built up dirt, etc. These fault scenarios result in a change in the thermodynamic performance of the engine as measured by the adiabatic efficiencies, and flow capacities. Therefore, generally a component fault in the jet engine is modelled indirectly by a decrease in the efficiency of the flow capacities. Other fault scenarios can also be considered in the jet engine such as the stability bleed leak, start bleed leak, compressor stator and vane misrigging, which require an additional block and model to add to our jet engine model.

In this thesis, we only consider the effects of the component faults as decreases in two health parameters of the engine namely, the efficiency (η) and the mass flow

Fault Scenario	Description
FmLC	Decrease in the Mass flow capacity of Low pressure Compressor
FeLC	Decrease in the Efficiency of Low pressure Compressor
FmHC	Decrease in the Mass flow capacity of High pressure Compressor
FeHC	Decrease in the Efficiency of High pressure Compressor
FmHT	Decrease in the Mass flow capacity of High pressure Turbine
FeHT	Decrease in the Efficiency of High pressure Turbine
FmLT	Decrease in the Mass flow capacity of Low pressure Turbine
FeLT	Decrease in the Efficiency of Low pressure Turbine

Table 2.2: The component faults that are considered in the jet engine.

capacity ($\dot{m}$). The list of the faults that are considered in this thesis is presented in Table 2.2.

2.6.1 Fault Influence Matrix

In order to evaluate the importance or significance level of each fault and its effect on the engine parameters, we have generated an influence matrix of all the engine parameters and measured the effects of each fault onto all the engine parameters. The attributes that we used to evaluate the effects of the faults are the percentage of the changes in the steady state values of the signals when faults occur.

This matrix is denoted by the influence matrix and each row of the matrix represents the percentage of the change in the steady state of that parameter when a specific fault occurs.

It is assumed that the faults are occurring in steady state operation of the jet engine. The changes are measured when the transition signals have settled down. For each fault scenario we have injected a 2%, a 5% and a 10% decrease in the component

Fault	LC_T %	LC_P %	HC_T %	HC_P %	HT_T %	HT_P %	LT_T %	LT_P %	$N1$ %	N_2 %	CC_T %	CC_P %
FmLC	4.39	8.27	3.28	4.76	1.08	4.77	1.92	4.95	6.73	0.87	1.07	4.76
FeLC	4.37	7.41	4.01	3.92	3.72	4.15	4.11	3.86	2.95	1.62	3.85	3.92
FmHC	4.09	8.36	1.25	3.86	1.98	4.26	2.17	4.15	1.77	5.36	2.32	3.86
FeHC	3.62	7.31	6.40	7.21	8.10	6.79	8.48	6.53	2.63	3.52	7.44	7.21
FmHT	0.62	1.37	4.49	13.52	2.47	7.75	1.86	4.21	0.72	1.54	2.81	13.52
FeHT	4.71	10.34	1.77	10.70	11.64	8.27	12.08	7.99	2.98	4.49	6.29	10.70
FmLT	7.30	17.12	2.36	5.61	7.61	13.97	4.71	4.89	1.24	4.29	3.92	5.61
FeLT	4.34	10.00	2.01	3.90	1.05	4.25	4.37	2.58	2.80	2.67	1.31	3.90

Table 2.3: The influence matrix corresponding to the average change in the steady state of the 12 measurements subject to 2%, 5%, 10% faults.

value and measured the percentage changes in each of the engine parameters. The influence matrix shows the average corresponding to the three percentage changes for each fault and each engine parameter.

The results are presented in Table 4.1 which shows that the larger the percentage change of each fault, the larger the correlation between the two parameters. For instance, it follows that a change in the mass flow rate of the low pressure compressor (FmLC) has a significantly large effect on the pressure of the low pressure compressor (LC_p) (8.27%). This is quite predictable since the mass flow rate of the compressor has a direct relationship with the pressure of the compressor.

2.6.2 Fault Classification

Based on the influence matrix results we present a specific classification of the faults in the jet engine. This classification divides the faults into three classes namely, High, Normal, and Low severity levels. These classes categorize the faults from the severity level perspective. To determine the severity level of each fault we assume that the faults that generate larger percentage changes in all the 12 engine parameters have

Fault Scenario	Fault Severity Level
FmLC	Low
FeLC	Normal
FmHC	Low
FeHC	High
FmHT	Low
FeHT	High
FmLT	High
FeLT	Normal

Table 2.4: The faults severity levels.

higher severities than others. To evaluate this severity we consider the median of each row of the influence matrix which is an indicator of the severity of the corresponding fault. One should note that this classification is based on an overall effect a fault can have on ALL the engine parameters and is an indication on how to know which faults cause make more change in the parameters of the jet engine.

In order to generate these classes the median of the percentage changes of each fault is calculated by using the influence matrix which is basically the median of each row of the influence matrix. The faults are then classified into three classes by setting a threshold for each class of fault where the results are presented in Table 2.4.

The classification which is developed here is a simple qualitative representation of the faults to yield a better understanding of the fault isolation results that will be presented in Chapter 4. In particular, a more detailed analysis can be performed in future to yield a thorough presentation of the effects of each fault in the jet engine parameters.

2.7 Conclusions

In this chapter the governing equations of two dynamic neural networks namely, the DNM and the TDNN were presented. The DNM updating algorithm is the extended dynamic back propagation whereas the TDNN is trained based on the Levenger Marquardt updating algorithm. In order to utilize the TDNN for fault diagnosis purposes a series-parallel architecture for TDNN was presented.

All the simulations in this thesis will be carried out based on a mathematical model that is developed by using the Simulink software. The jet engine mathematical model and the nonlinear equations of motion as well as the components of the jet engine were introduced in this chapter.

The initial ambient parameters for all the simulations are set corresponding to the altitude of the cruise mode which in this thesis is assumed to be 4000m. This leads to the ambient condition of $T_{amb} = 262$ centigrade and $P_{amb} = 0.63$ atmosphere. The Mach number is also set to 0.7 and is assumed to be fixed in the cruise mode. Moreover, in this chapter a fault influence matrix was presented in order to determine the effects of each fault in all the engine parameters. This matrix shows the percentage change of each parameter corresponding to a specific fault.

Chapter 3

Dynamic Neural Network-based Fault Detection Scheme

Our aim in this chapter is to develop a fault detection scheme to determine the minimum percentage of the detectable fault for all the possible fault scenarios under consideration in the jet engine. Towards this end, two different dynamic neural network structures are used for fault detection of the jet engine, namely the DNM and the TDNN. For each method, 12 networks are trained separately for all the 12 existing engine parameters as presented in the Table 2.1 in Chapter 2. The fault detection performance of each of these 12 networks is shown and demonstrated. The engine parameters that have better capabilities in monitoring the engine health status are then determined. It is shown that by using a bank of neural networks together as the residual generators one can improve the overall performance of the general fault detection scheme.

3.1 Dynamic Neuron Unit Approach

As presented in Chapter 2, a dynamic neuron model is constructed by using an internal IIR filter in the neuron. This filter generates the dynamical behaviour between inputs and outputs. One of the advantages of this structure having internal filters is that there is no need to a series-parallel structure for identifying the nonlinear dynamics of the engine. In other words, there is a need for having a delayed sample of the output as an input. Hence, the network can be seen as a SISO system. In Figure 3.1 the fault detection scheme is depicted for the 12 engine parameters. Each neural network is working in parallel with the engine to generate a residual signal. The residual signal is then evaluated for the purpose of fault detection task. The input and the control signal to the jet engine and to each neural network models is the power level angle (PLA) or the fuel flow rate and the output would be one of the 12 engine parameters. Our fault detection scheme consists of two stages, namely (1) system identification and (2) fault detection. These are described in more details below.

3.1.1 System Identification

System identification plays an important role in the fault detection algorithm. During this task dynamic neural networks learns the dynamics of the actual turbo fan engine. Towards this end, two tasks should be performed, namely the training phase and the testing and validation phase of the proposed DNM approach.

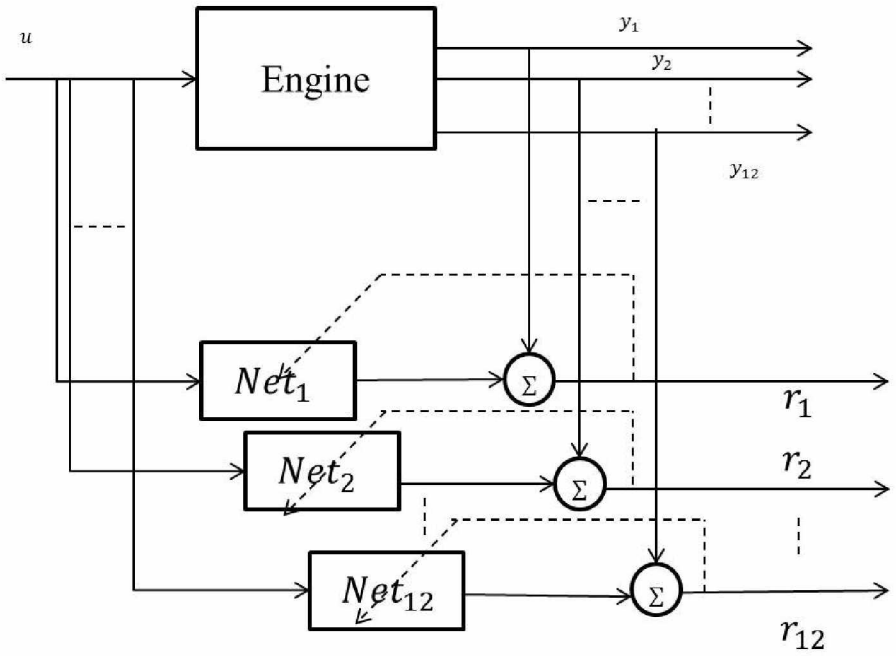


Figure 3.1: Fault detection schematic by utilizing a bank of DNM networks.

3.1.2 Training Phase

In this thesis we have developed a bank of 12 different networks where each has one input (fuel mass flow rate) and one output which is an element of the measurement vector. These networks are designated as $Net_{LC_p}, Net_{LC_T}, Net_{HC_p}, Net_{HC_T}$, $Net_{LT_T}, Net_{LT_T}, Net_{HT_p}, Net_{HT_T}, Net_{N_1}, Net_{N_2}, Net_{CC_T}, Net_{CC_P}$. It is assumed that the engine is operating in the cruise mode and is fault free during the training mode. Experimenting with several input profiles, it has been observed that by training the network using a zigzag signal ranging between 40% and 98% of the maximum fuel mass flow rate, the minimum error and the best performance can be achieved. The frequency of the signal and the slope of each ramp was considered sufficiently low to satisfy the engine dynamics.

Due to the high complexity of the engine dynamics, a large number of data is required for network to learn the dynamics of the engine. For example, the Net_{N1} has been trained by using 56,000 normalized data points which are generated by using a

Engine Parameter	Noise Percentage
LC_P	2%
LC_T	2%
HC_P	2%
CC_P	2%
HC_T	2%
LT_P	2%
HT_P	2%
CC_T	2%
$N1$	0.5%
$N2$	0.5%
HT_T	2%
LT_T	2%

Table 3.1: The measurement noise percentage applied to each engine parameter.

Simulink model of the dual spool jet engine and are the sampled data when the engine was running for about 10 min. To have a realistic model (as provided in Chapter 2) as close as possible to a practical aircraft engine, all the data for training are considered under presence of the measurement noise as presented in Table 3.1. The weight adjustment for the dynamic network was carried out by using an extended dynamic back propagation algorithm as described in Chapter 2. To make our proposed fault detection scheme as close as a realistic problem, the training method that is chosen is the incremental learning. As described in Chapter 2, in this adaptation law the weights of the $(k+1)^{th}$ sample will be updated based on the errors of the k^{th} sample. This results in a point by point update which leads to an on-line weight adjustment system.

Both the input and the output of the networks are normalized by using the max-min normalization method. It has been observed that the training procedure is sensitive to the normalization method. After several investigations it turned out that by using the max-min normalization scheme, a better performance can be achieved where both the input and the output are normalized according to:

$$X_n = 2 * \frac{(X - a)}{b - a} \tag{3.1.1}$$

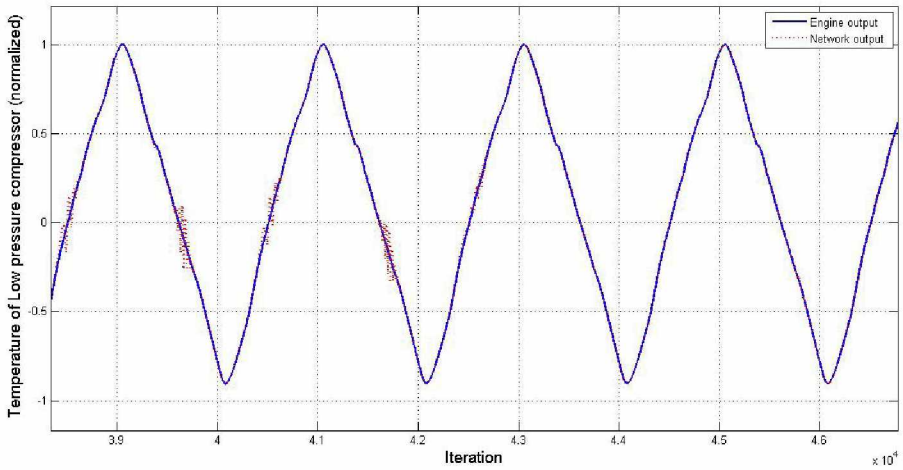
where a and b denote the maximum and the minimum of the range of the variation of the variable.

The learning algorithm is initialized with small random values for the network parameters (weights, feedback filters, activation function slopes) while the IIR filter's denominator coefficients were set to zero to ensure stable learning. All the neurons in the network are assumed to be embedded second order IIR filters. Due to the internal filter dynamics in the DNM there would be an internal feedback from the input and the output into the system, hence a second order filter has the sufficient ability to capture the dynamics of the system. The filter order is set to 2 as empirically we have found that there is no significant change in the leaning performance by increasing the filter order other than increasing the computational complexity.

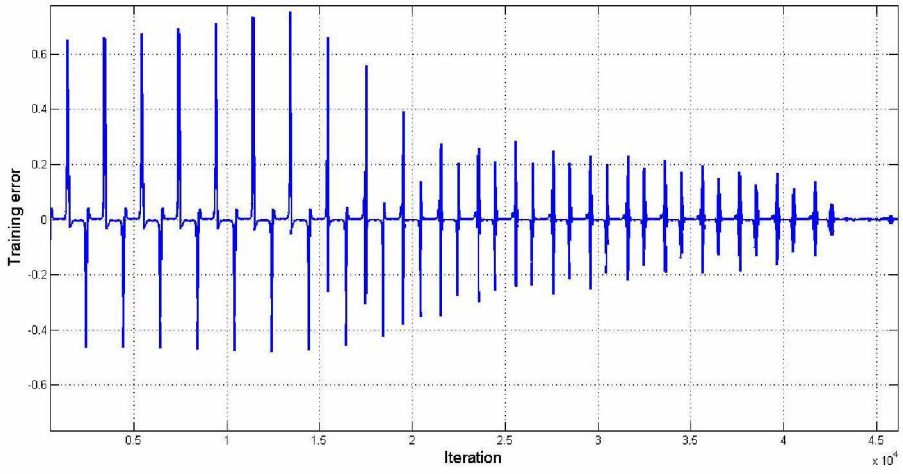
We have also observed that having only one hidden layer will unnecessarily increase the system complexity as the number of neurons should then be selected to be relatively large. Hence, we have used two hidden layers for each network to learn the dynamics of the jet engine. The activation functions in the hidden layers are the hyperbolic tangent and linear activation functions for the output layer. Starting from a relatively small structure, we developed an optimal architecture for our proposed dynamic networks by incrementally increasing the number of neurons in the hidden layers until a termination criterion (t.c.) is satisfied . The t.c. used is based on the

mean square error (mse) criterion and the convergence of the weights of the network. The termination criterion is used as t.c.=.008 and the training is stopped if the mse is smaller than the t.c. and the weights of the network have converged. Table 3.3 summarizes the characteristics and parameters of the resulting networks.

The training signals and errors as well as the network parameters updates for the neural networks Net_{LC_T}, Net_{LC_P}, Net_{HC_T} and Net_{HT_T} are depicted in Fig. 3.2-3.11. From Fig. 3.2 one can observe that the network output Net_{LC_T} is following the engine output after 45,000 iterations quite well. Fig. 3.3 and Fig. 3.4 show the network parameters updates. It follows that the network weights and error will be within a certain value after 45,000 iterations of training. This ensures that the neural network is well trained and is ready to be used in the recall phase.

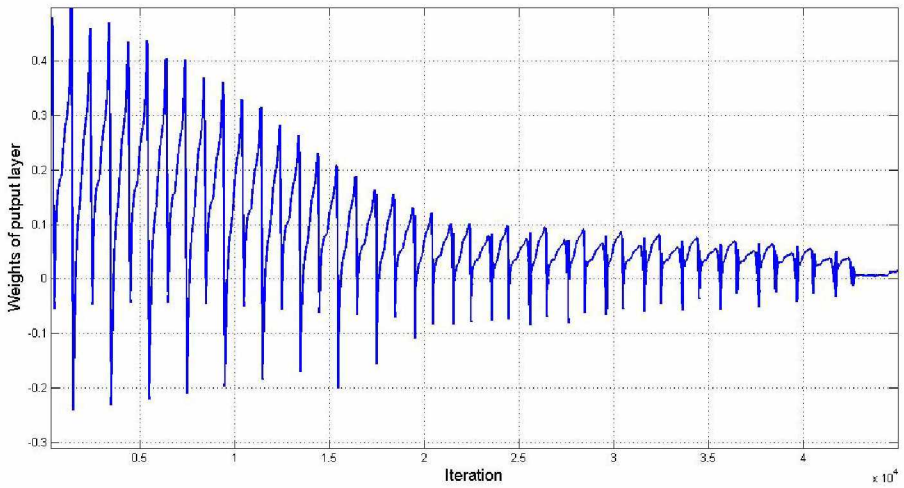


(a) The training signal for the network Net_{LC_T}.

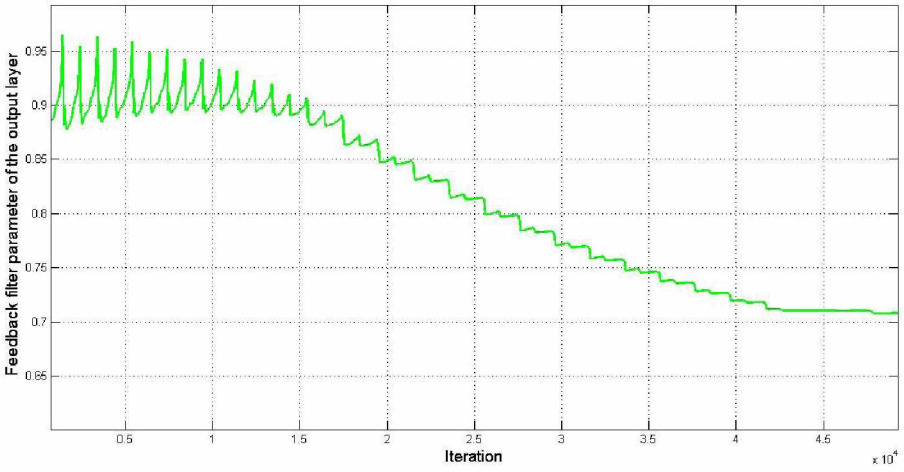


(b) The training error profile for the network Net_{LC_T}.

Figure 3.2: The training signal and the training error for the network Net_{LC_T}.

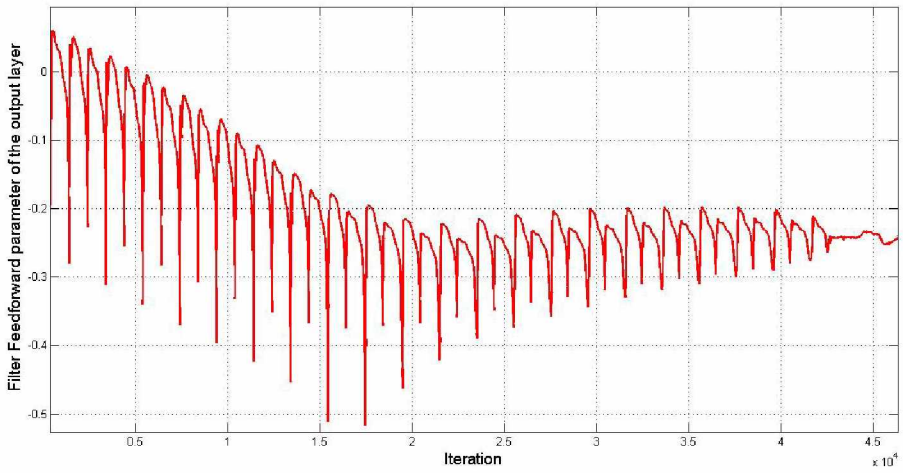


(a) Weights.

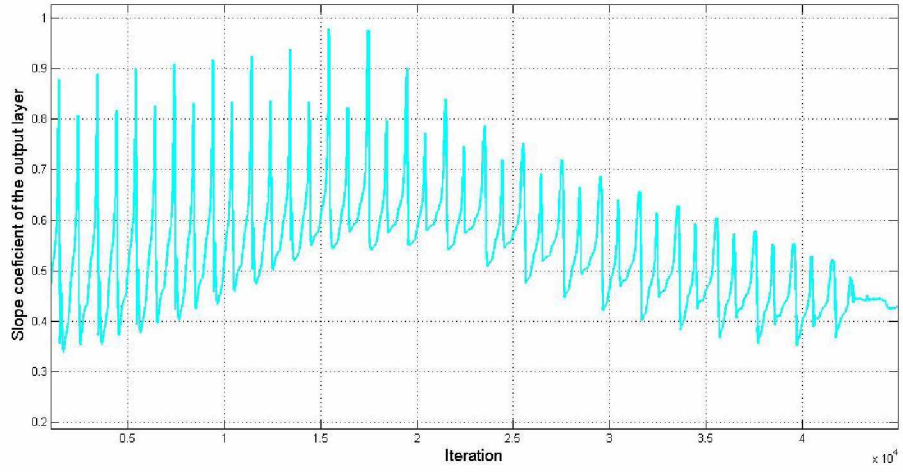


(b) Filter feedback parameters.

Figure 3.3: The evolution of the Net_{LC_T} network parameters updates.

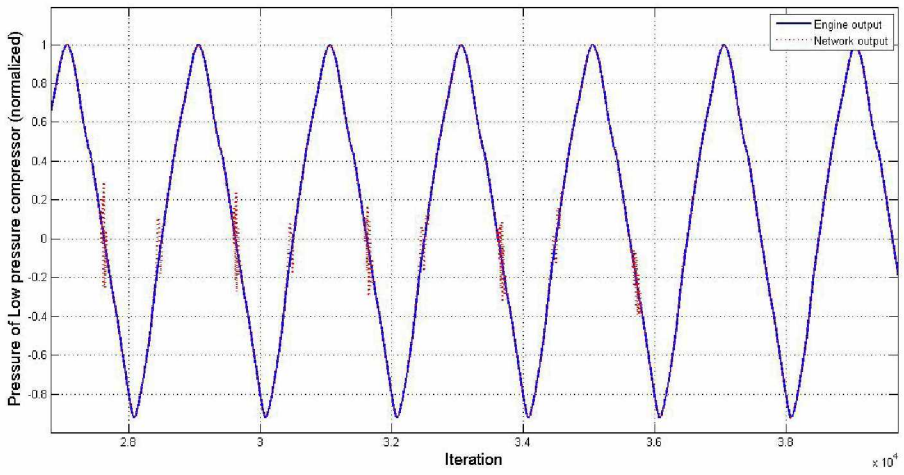


(a) Filter feedforward parameter.

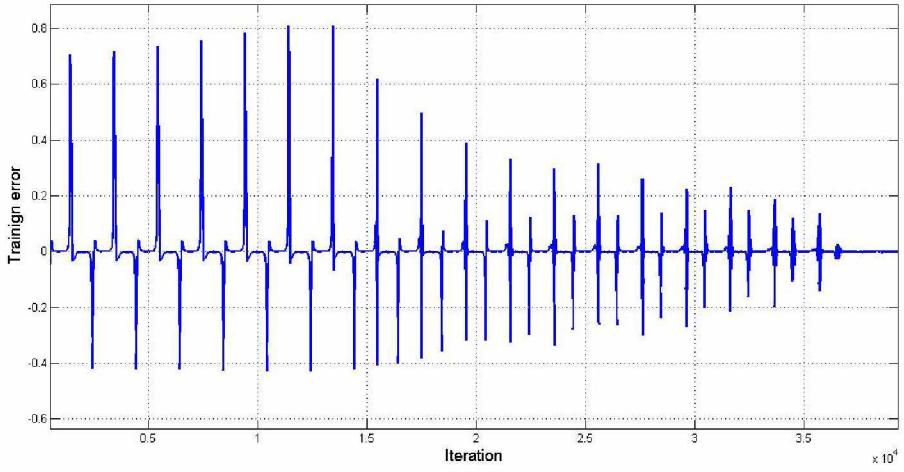


(b) Slope coefficient.

Figure 3.4: The evolution of the Net_{LC_T} network parameters updates.

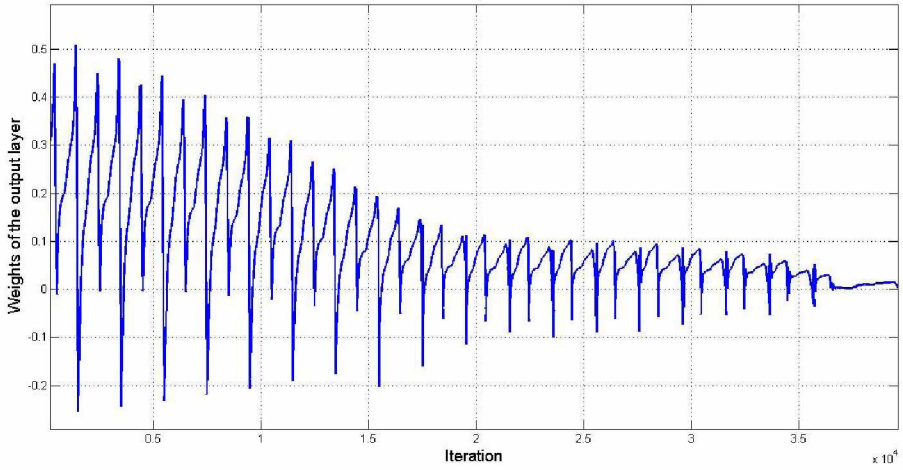


(a) The training signal for the network Net_{LC_P}.

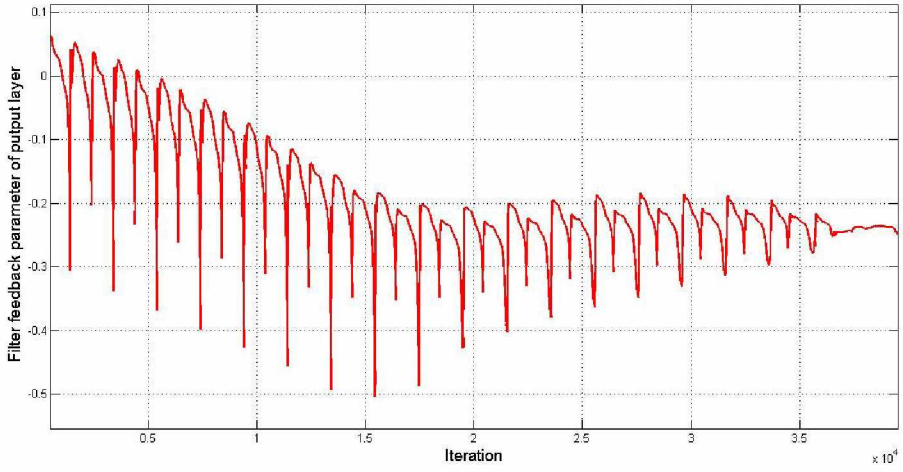


(b) The training error profile for the network Net_{LC_P}.

Figure 3.5: The training signal and the training error for the network Net_{LC_P}.

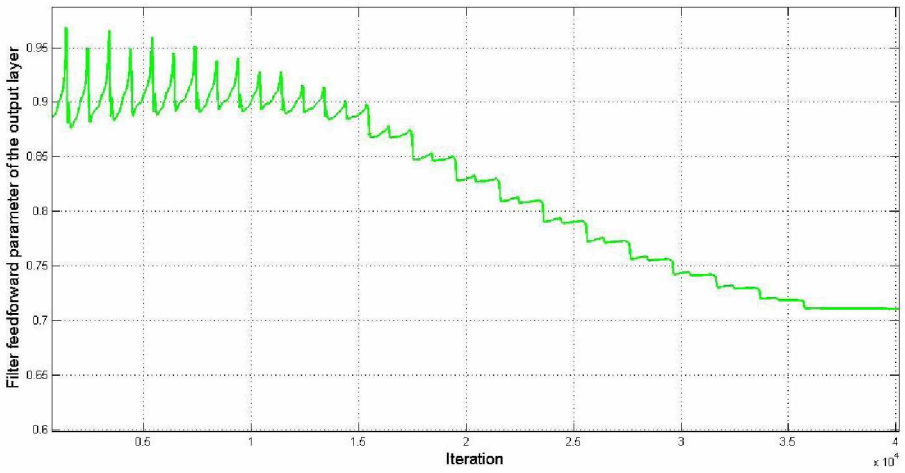


(a) Weights.

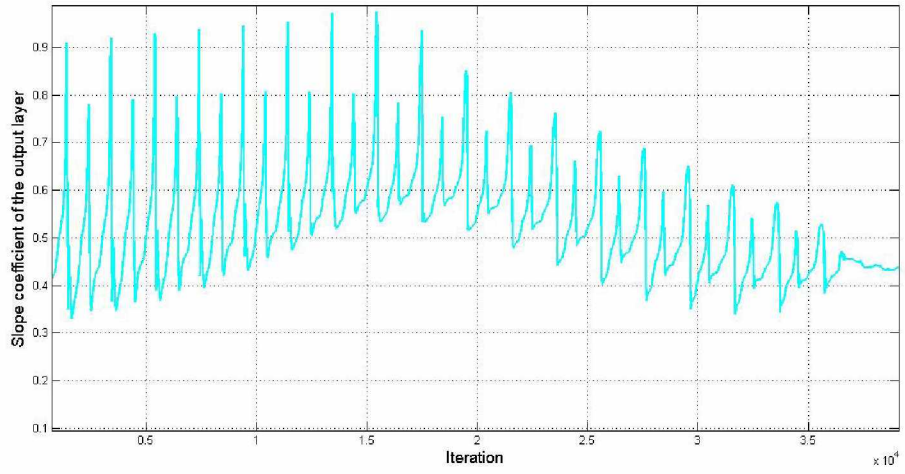


(b) Filter feedback parameters.

Figure 3.6: The evolution of the Net_{LC_P} network parameters updates.

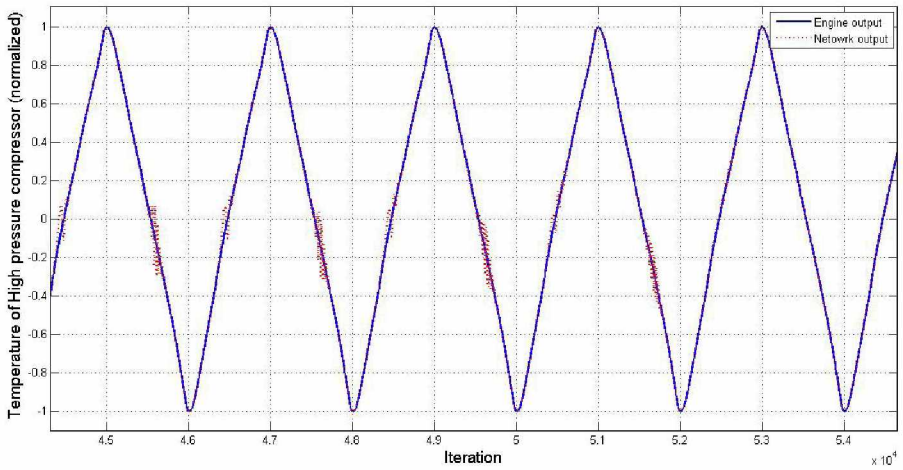


(a) Filter feedforward parameter.

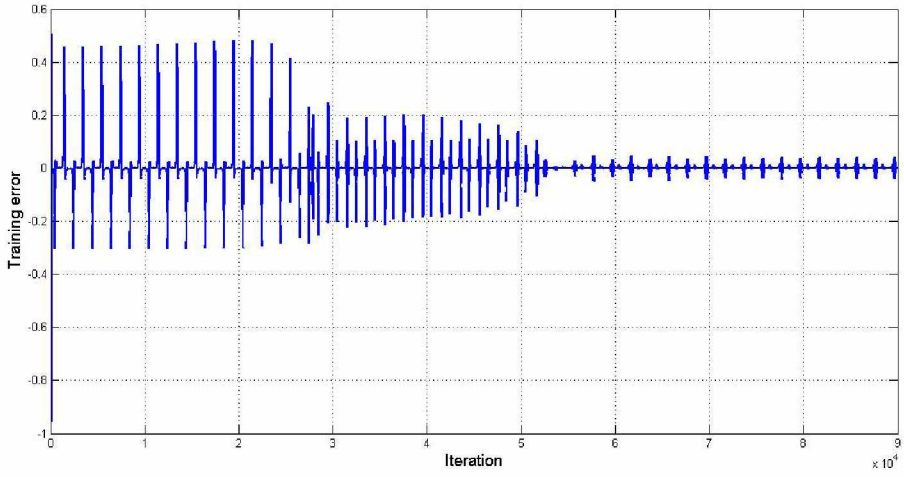


(b) Slope coefficient.

Figure 3.7: The evolution of the Net_{LC_P} network parameters updates.

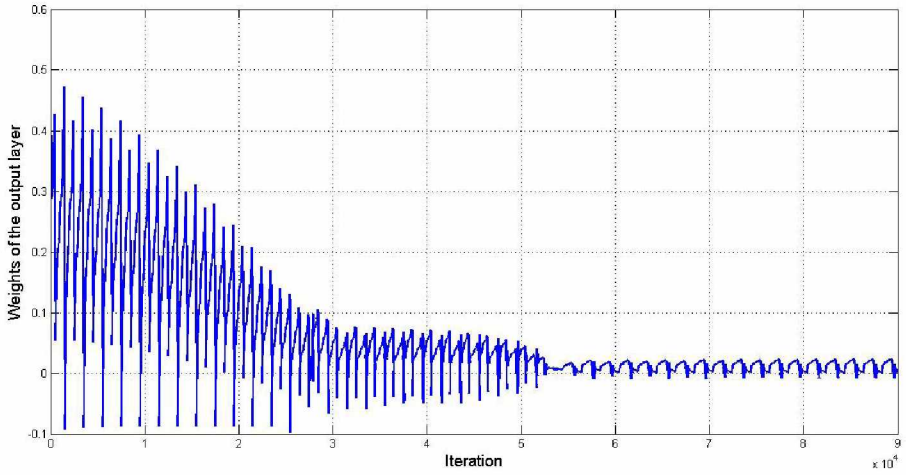


(a) The training signal for the network Net_{HC_T}.

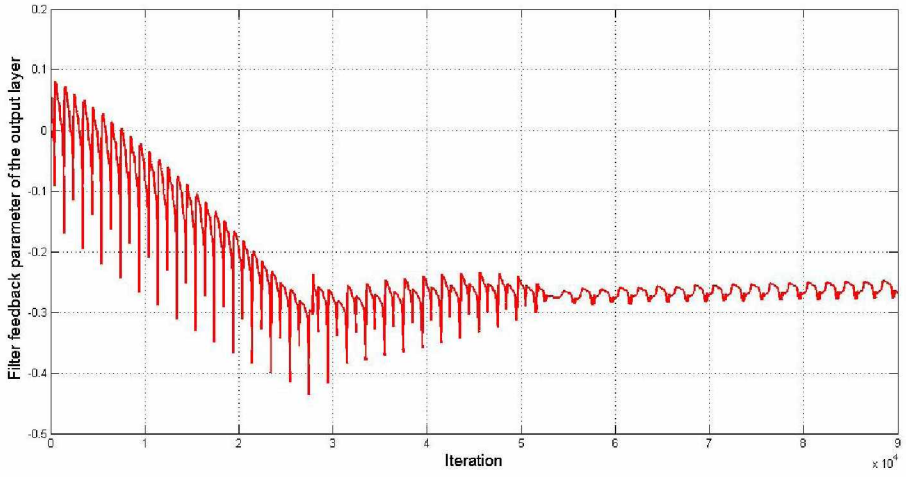


(b) The training error profile for the network Net_{HC_T}.

Figure 3.8: The training signal and the training error for the network Net_{HC_T}.

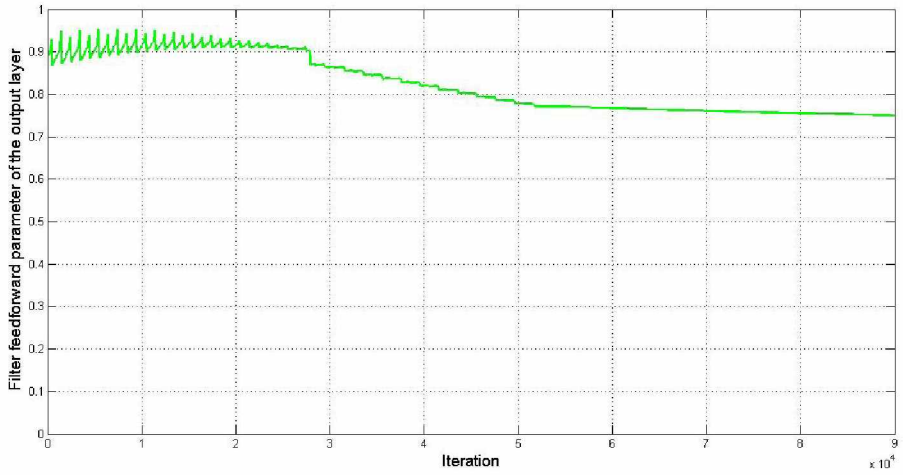


(a) Weights.

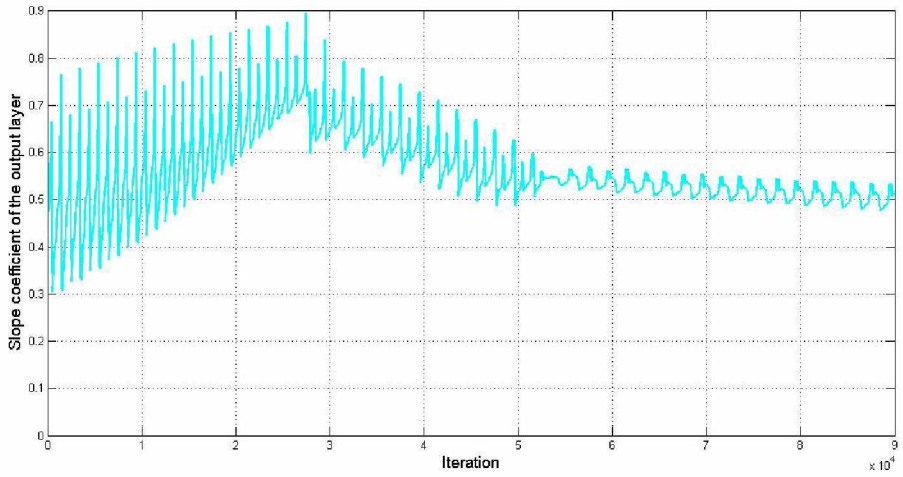


(b) Filter feedback farameters.

Figure 3.9: The evolution of the Net_{HC_T} network parameters updates.

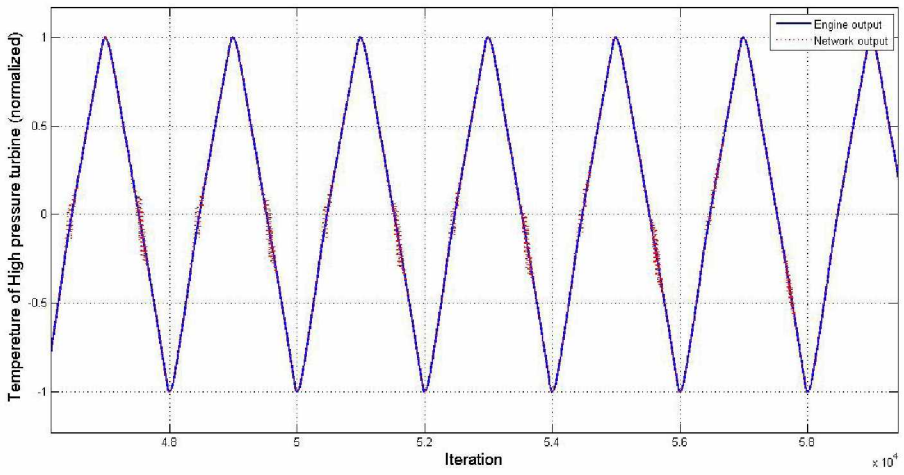


(a) Filter feedforward parameter.

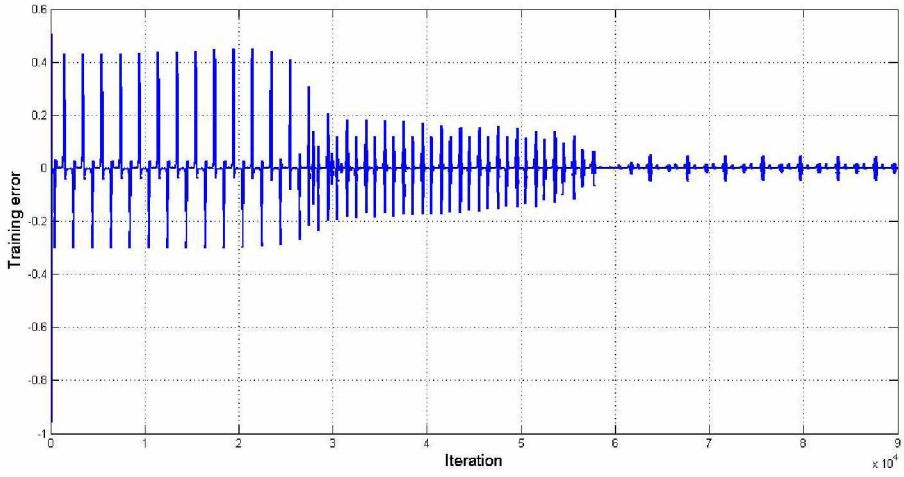


(b) Slope coefficient.

Figure 3.10: The evolution of the Net_{HC_T} network parameters updates.

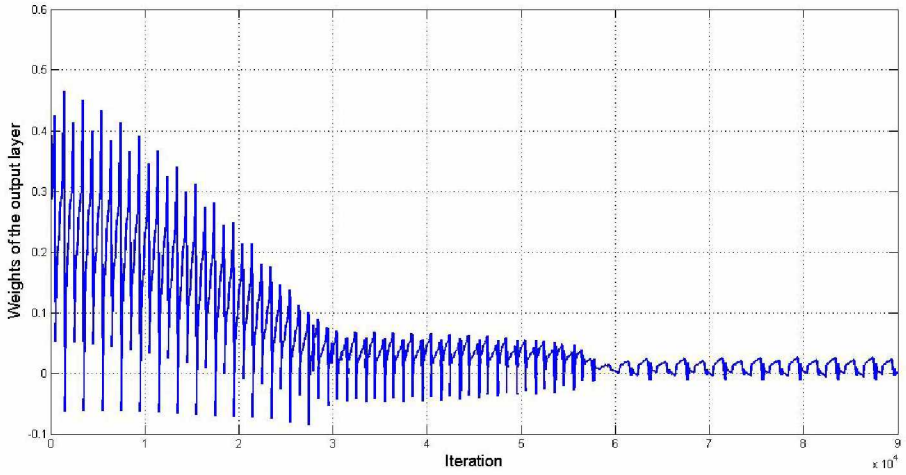


(a) The training signal for the network Net_{HT_T}.

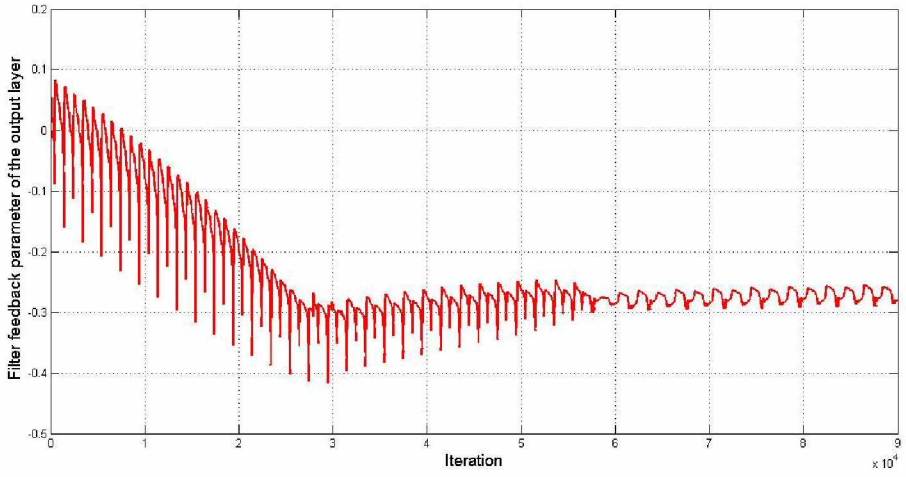


(b) The training error profile for the network Net_{HT_T}.

Figure 3.11: The training signal and the training error for the network Net_{HT_T}.

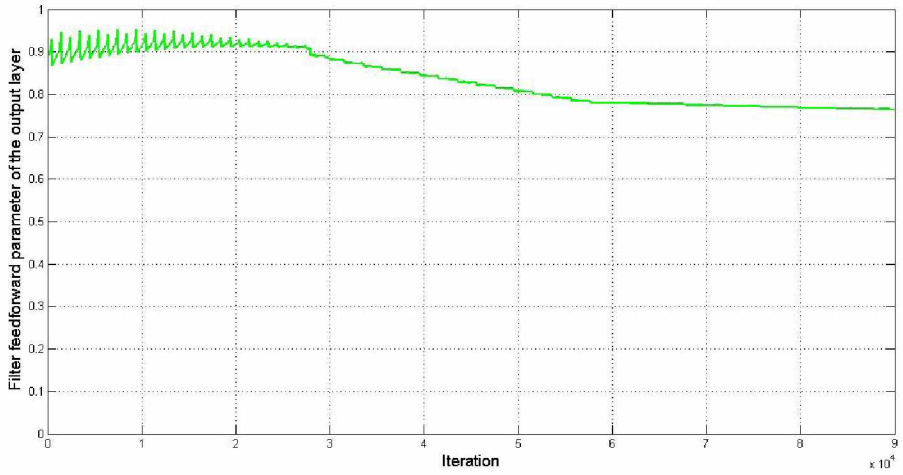


(a) Weights.

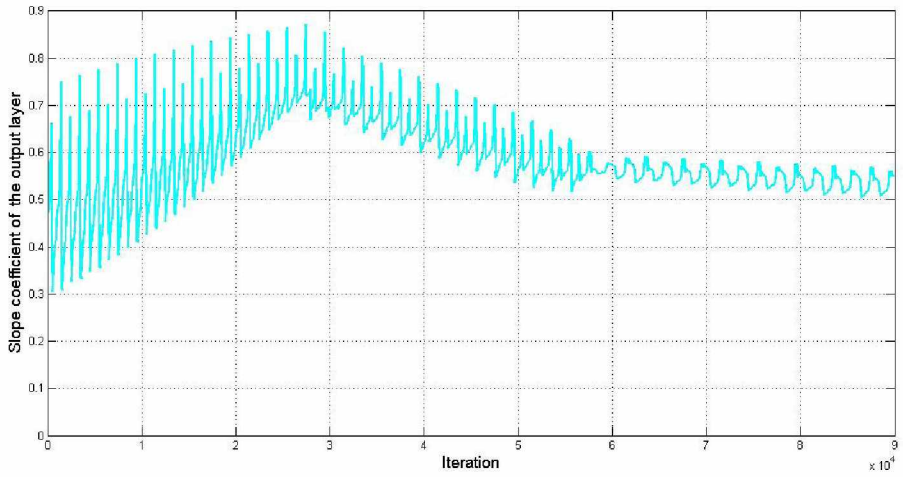


(b) Filter feedback parameters.

Figure 3.12: The evolution of the Net_{HT_T} network parameters updates.



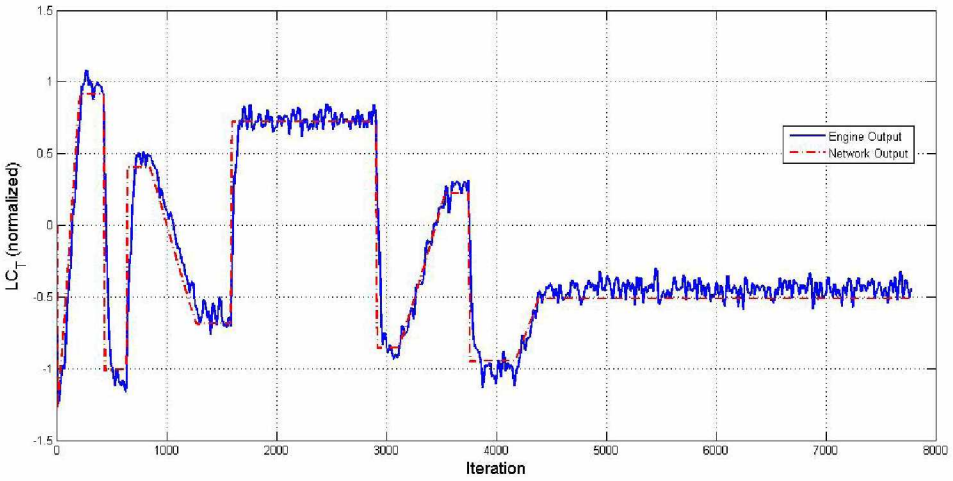
(a) Filter feedforward parameter.



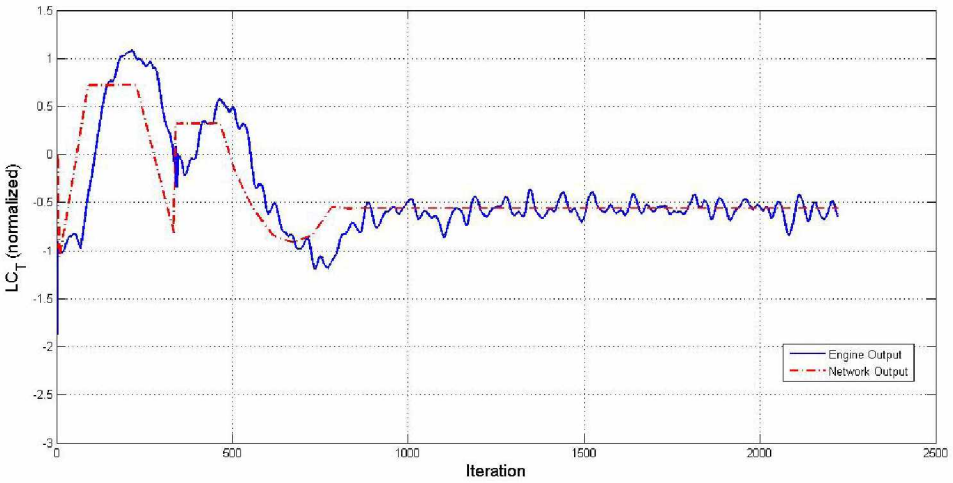
(b) Slope coefficient.

Figure 3.13: The evolution of the Net_{HT_T} network parameters updates.

To demonstrate the capability of the trained network, it was tested with another data set that has not been seen previously by the network. Fig. 3.14-3.19 show the validation/testing phase for two different input profiles defined as input a and input b. It can be seen that the output of the networks is following the actual output of the jet engine quite well.

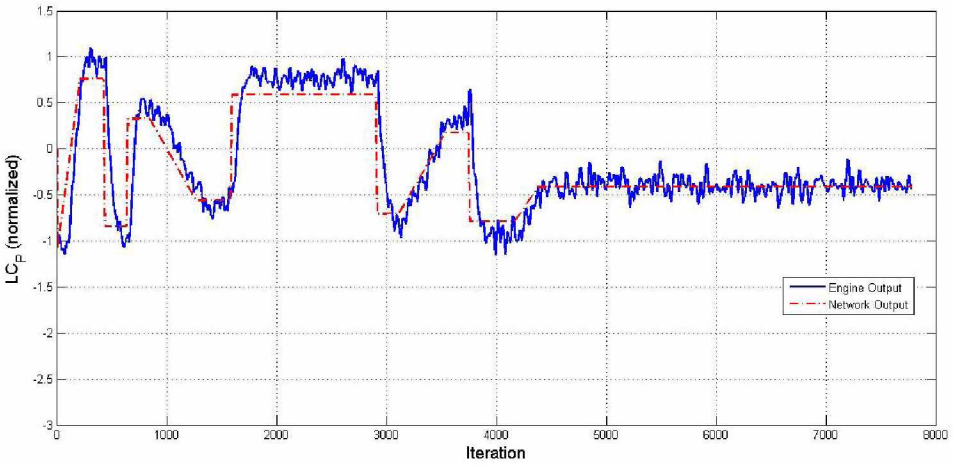


(a) Engine and network output.

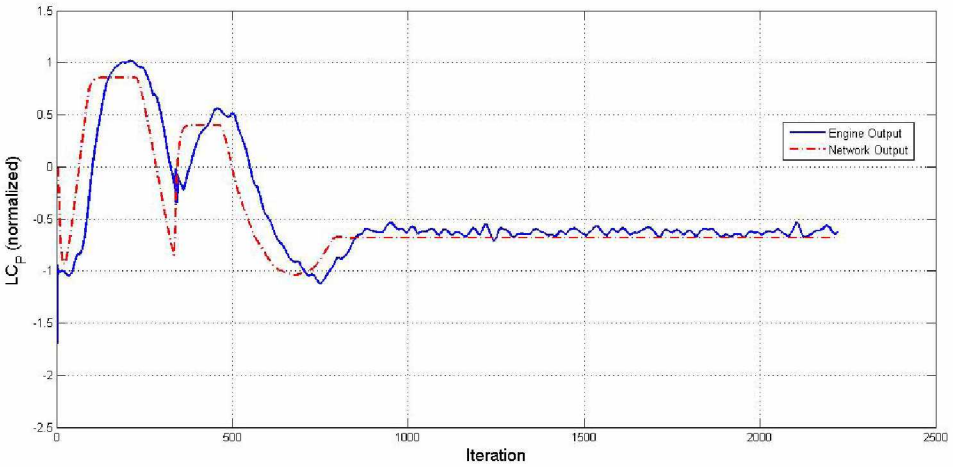


(b) Engine and network output.

Figure 3.14: Testing of the trained Net_{LC_T} network with "unseen" inputs of a and b.

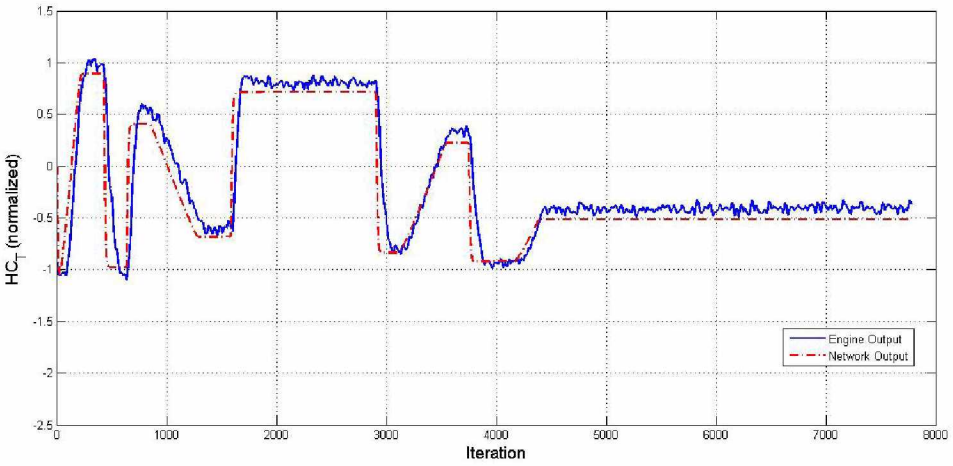


(a) Engine and network output.

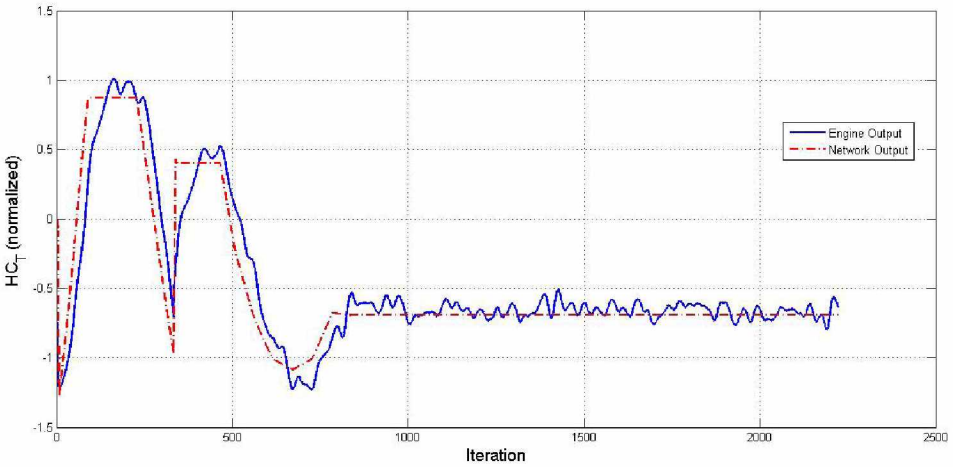


(b) Engine and network output.

Figure 3.15: Testing of the trained Net_{LC_P} network with "unseen" inputs of a and b.

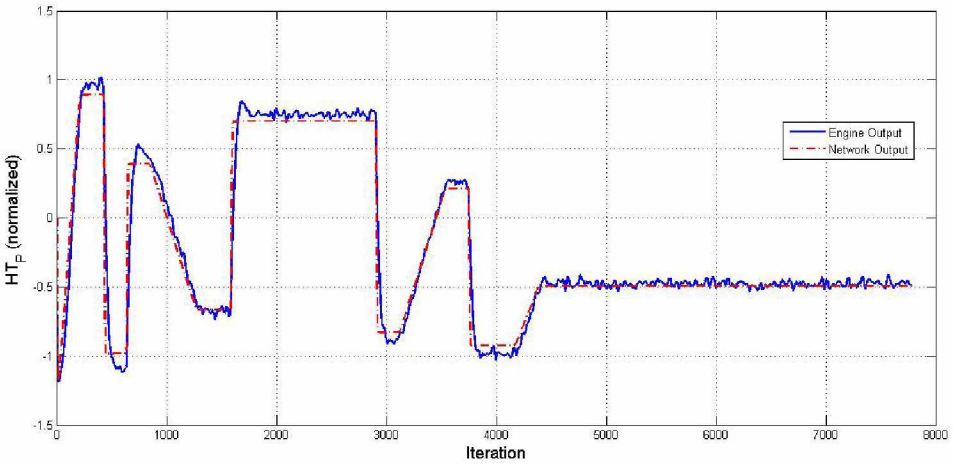


(a) Engine and network output.

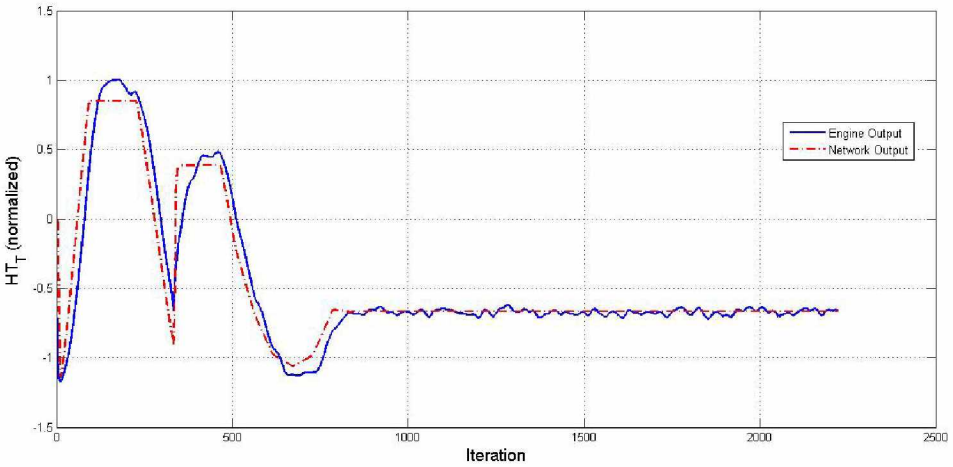


(b) Engine and network output.

Figure 3.16: Testing of the trained Net_{HC_T} network with "unseen" inputs of a and b.

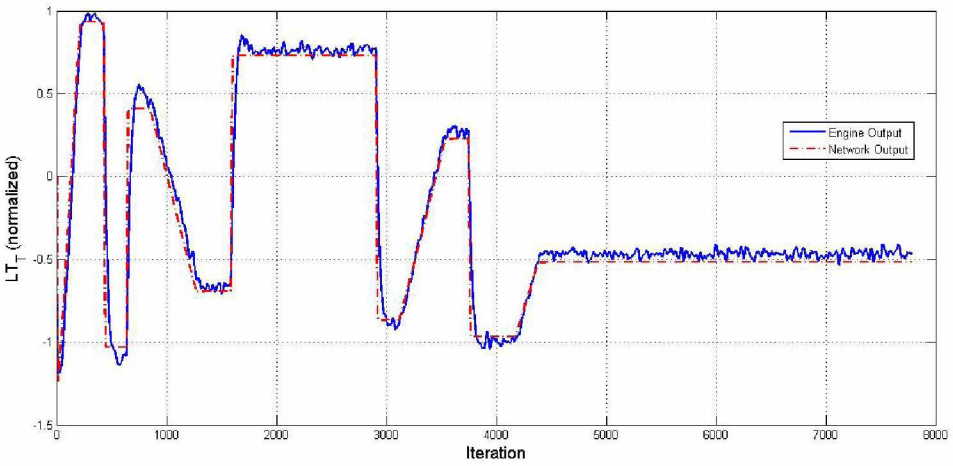


(a) Engine and network output.

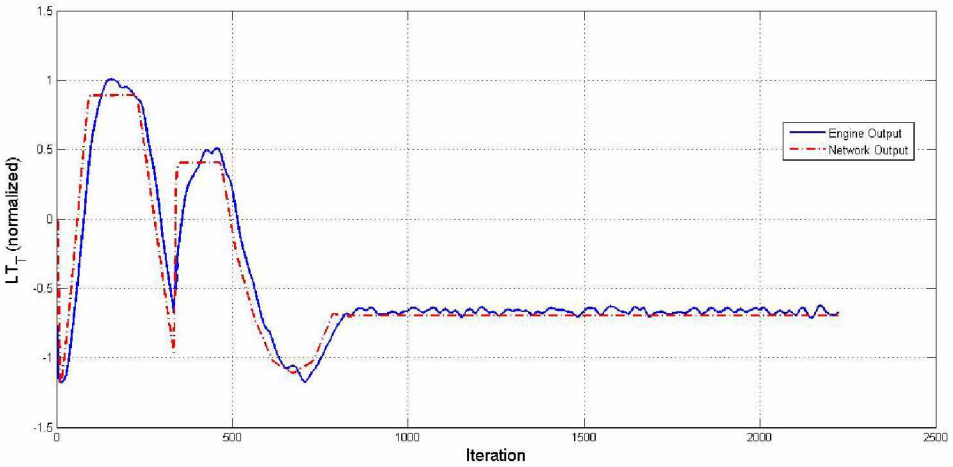


(b) Engine and network output.

Figure 3.17: Testing of the trained Net_{HT_P} network with "unseen" inputs of a and b.

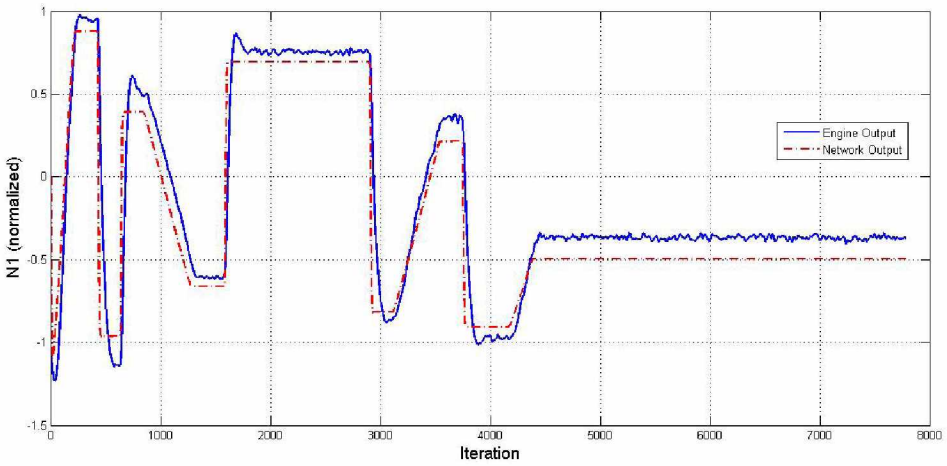


(a) Engine and network output.

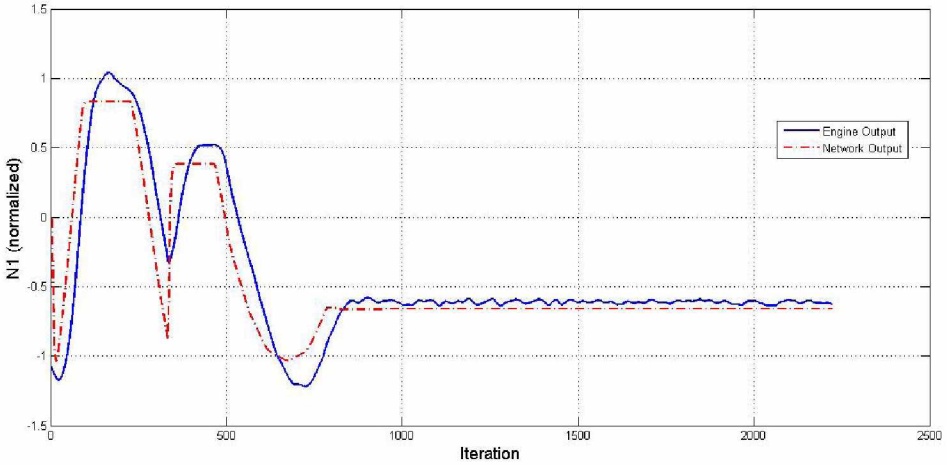


(b) Engine and network output.

Figure 3.18: Testing of the trained Net_{LT_T} network with "unseen" inputs of a and b.



(a) Engine and network output.



(b) Engine and network output.

Figure 3.19: Testing of the trained Net_{N_1} network with "unseen" inputs of a and b.

3.1.3 Fault Detection

The trained bank of neural networks are now used to generate the residual signals. The residual signal would be the difference between the actual engine output and the neural network output. When there is a fault in the system the error between these two outputs will increase and it would then be possible to detect a fault by monitoring the residual signal using an appropriate threshold level. The decision making system for our fault detection scheme is based on the concept of the confusion matrix. A confusion matrix consists of four elements, namely the true positive, the true negative, the false positive, and the false negative which are defined below:

- True positive (t.p.): the number of samples detected as healthy while the engine is operating in the healthy mode.
- True negative (t.n.): the number of samples detected as faulty while the engine is operating in the faulty mode.
- False negative (f.n.): the number of samples detected as healthy while the engine is operating in the faulty mode.
- False positive (f.p.): the number of samples detected as faulty while the engine is operating in the healthy mode.

True Positive	False Negative
False Positive	True Negative

Table 3.2: The confusion matrix.

For each residual signal a confusion matrix is constructed and the two parameters of accuracy and precision are calculated to evaluate the performance of the fault

detection scheme. These two parameters are defined by:

$$Accuracy = \frac{true\,positive + true\,negative}{true\,positive + true\,negative + false\,positive + false\,negative} \tag{3.1.2}$$

$$Precision = \frac{true\,negative}{true\,negative + false\,negative} \tag{3.1.3}$$

Different fault values ranging from 1% to 12% are injected to the aircraft engine to determine what is the minimum detectable fault severity for each network with respect to the chosen threshold. A fault scenario is considered as detectable if the averages of the accuracy and the precision of the confusion matrix corresponding to that fault severity level are more than 80%. The threshold for each network was chosen by conducting Monte Carlo simulations corresponding to random noise levels. Table 3.7 provides the selected thresholds for each network.

In this thesis, the threshold level was defined by $mean(e) + .5 * std(e)$, where e denotes the steady state error between the engine output and the network output and mean denotes the average of the signal and *std* denotes the standard deviation. The noise level was chosen as 0.5% for the N_1 and N_2 and 2% for all the other engine parameters as shown in Table 3.1 [67].

We observed that a network with 5 neurons in the first hidden layer and 5 neurons in the second hidden layer has the best performance in learning the dynamics of the jet engine. Hence, the structure of the neural network was chosen as $1 * 5 * 5 * 1$ for all the engine outputs. However, we have observed that for the Net_{N1} and Net_{N2} different structures are more desirable, and hence the structure of the network for these two parameters are selected as $1 * 20 * 10 * 1$ and $1 * 22 * 11 * 1$, respectively. The learning rate for the parameters η_b ,η_g ,η_w of all the networks are chosen as 0.6 and η_a is chosen as 0.03 for the Net_{N1} and Net_{N2} and 0.03 for all the other engine

The Trained Network	No. of Iterations	MSE
Net_{LC_T}	45000	0.0079
Net_{LC_P}	38000	0.0069
Net_{HC_T}	52000	0.0061
Net_{HC_P}	58000	0.0056
Net_{HT_T}	59000	0.0072
Net_{HT_P}	58000	0.0057
Net_{LT_T}	62000	0.0079
Net_{LT_P}	57000	0.0057
Net_{N1}	56000	0.0071
Net_{N2}	95000	0.0024
Net_{CC_T}	58000	0.0069
Net_{CC_P}	58000	.00056

Table 3.3: The DNM training parameters.

parameters.

Table 3.3 summarizes the number of the iterations and the mean square error for each network during the training phase. The threshold levels for each network are shown in Table 3.7. It is assumed that the engine is operating in the cruise mode where the PLA is varying from 50 to 60 degrees. This would bring the W_f (fuel flow) in the range of 72% to 86% of the maximum value. For any specific fault severity, each network was fed with all the input profiles in the range of 70% to 90% of the maximum fuel mass flow rate. Although, the fault detection was accomplished in the steady state of the jet engine operation, in order to show that the network has learned the aircraft engine dynamics properly the input profile was chosen as a ramp that starts from 40% of the maximum fuel mass flow rate and reaches a steady state value between 70% to 90% of the maximum fuel flow rate. Indeed, the input profile is changing from 70% to 90% at the rate of 5%, which takes the input profiles to 5 different operating settings.

Fault Scenarios Trained Network	FmLC %	FeLC %	FmHC %	FeHC %	FmHT %	FeHT %	FmLT %	FeLT %
Net_{LC_T}	N	N	N	7	N	5	3	8
Net_{LC_P}	6	8	N	7	N	4	2	6
Net_{HC_T}	N	8	12	4	4	N	N	N
Net_{HC_P}	N	12	N	5	2	3	6	N
Net_{HT_T}	N	N	N	5	N	3	5	N
Net_{HT_P}	N	N	N	6	N	5	2	N
Net_{LT_T}	N	N	N	7	N	5	12	N
Net_{LT_P}	N	N	N	6	N	5	9	N
Net_{N1}	2	N	N	N	N	N	N	N
Net_{N2}	12	10	3	3	11	2	2	6
Net_{CC_T}	N	11	N	5	N	6	9	N
Net_{CC_P}	N	12	N	5	2	3	6	N

Table 3.4: The minimum detectable fault severity level (%) using DNM approach (N implies that the fault cannot be detected).

Each fault is injected at the time t=15 seconds where the transients of engine have already settled down. Any fault is said to be detectable if it can be detected corresponding to all the 5 different input profiles ranging from 70% to 90%. A fault severity is considered as detectable if the average of the accuracy and precision parameter corresponding to the network is greater than 80%. If the fault is detected the average detection time is calculated from the 5 different input profiles. The results are shown in Tables 3.4 and 3.6. The detecting performance of the fault FeHC for the network Net_{N_2} is shown in Table 3.5. One can observe from Table 3.5 that when FeHC is 3% the average of the accuracy and the precision is greater than 80% and the fault severity is considered as detectable.

Fault severity level (%)	Accuracy (%)	Precision (%)	Confusion matrix
FmHC=1 %	68.31	61.51	40 32 10 50
FmHC=2%	78.01	77.82	39 19 11 63
FmHC=3%	95.33	93.07	49 6 1 76

Table 3.5: The detection performance for the network Net_{LC_P} (Note that the values in the entries of the confusion matrix refer to the actual number of residual signal samples representing the corresponding characteristic).

Fault Scenarios Trained Network	FmLC s	FeLC s	FmHC s	FeHC s	FmHT s	FeHT s	FmLT s	FeLT s
Net_{LC_T}	N	N	N	15.89	N	15.39	15.17	15.97
Net_{LC_P}	16.23	16.23	N	15.11	N	15.15	15.17	15.41
Net_{HC_P}	N	15.26	15.43	15.48	15.01	N	N	N
Net_{HC_T}	N	15.57	N	15.20	16.08	15.34	15.12	N
Net_{HT_T}	N	N	N	15.36	N	15.12	15.31	N
Net_{HT_P}	N	N	N	15.59	N	15.32	15.84	N
Net_{LT_T}	N	N	N	15.16	N	15.41	16.29	N
Net_{LT_P}	N	N	N	15.97	N	15.02	15.07	N
Net_{N1}	15.03	N	N	N	N	N	N	N
Net_{N2}	16.02	16.54	15.38	15.19	15.52	15.26	15.16	15.71
Net_{CC_T}	N	15.24	N	15.92	N	15.36	16.63	N
Net_{CC_P}	N	15.25	N	15.19	15.01	15.24	15.35	N

Table 3.6: The average detection time corresponding to the minimum detectable fault severity level.

Trained Network	Threshold
Net_{LC_T}	7.97 °C
Net_{LC_P}	0.079 atm
Net_{HC_T}	11.20°C
Net_{HC_P}	0.34 atm
Net_{HT_T}	24.77°C
Net_{HT_P}	0.10 atm
Net_{LT_T}	28.79°C
Net_{LT_P}	0.0392 atm
Net_{N1}	165.42 rpm
Net_{N2}	98.42 rpm
Net_{CC_T}	35.15°C
Net_{CC_P}	0.34 atm

Table 3.7: The threshold levels used for each DNM for fault detection.

3.1.4 Discussions

Table 3.4 summarizes the results of our fault detection scheme by using the DNM. The minimum detectable faults for each network are presented for all the fault scenarios. The faults considered in this study can vary between 1% to 12%. We assume any fault greater that 12% as serious failure which would need an urgent maintenance action. Hence, the notation N (Not detected) is used to denote when the network cannot detect the fault values between 1% to 12%. For instance, by monitoring the residual of the Net_{N_2} one cannot detect any fault value between 1% to 12 % decrease in the mass flow rate of the high pressure turbine.

One can observe from the results that was presented in Table 3.4 that they are in agreement with our fault severity classification that was presented in Chapter 2. Our fault detection scheme is performing satisfactorily and detecting smaller fault

severities that are labelled as High severity. One can compare the detection results corresponding to the FeHT (High) and FmHT (Low) where in general the networks are detecting smaller values of the faults for FeHT.

From Table 3.4 one can observe that in general Net_{N_2} and Net_{LC_P} have the best performance amongst the networks. On the other hand, using the pressure of the low pressure compressor as a signal to monitor the health of the jet engine one can detect smaller values of the fault in the engine. To demonstrate the function of Net_{LC_P} as a residual generator for the fault detection problem a decrease of 5% fault is injected in all the 8 fault scenarios when the fuel flow mass rate is at 70% of its maximum value. The residual of Net_{LC_P} in then monitored. Figures 3.20 and 3.21 show the residual error signals. It can be seen from the results that the faults in FeLC and FmHT and FmHC and FeLT are not detected due to the fact that the minimum detectable value for these faults as presented in Table 3.4 are greater than 5% .

Enhanced Detection Scheme

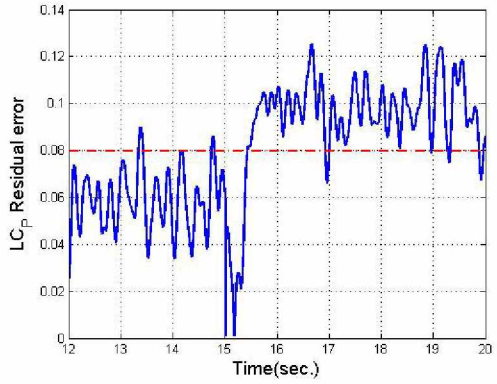
As presented in Table 3.4, each engine parameter has different capability and performance in fault detection of the jet engine. This difference may be used in order to develop a better performance in the fault detection process for a specific fault of different networks. Developing a fault detection scheme that consists of several residual generators working in parallel will improve the overall performance of the fault detection module. Towards this end, for each fault scenario we will choose the network that can detect the smaller value of a given fault. It can be easily seen from Table 3.4 that by choosing the networks Net_{LC_P}, Net_{N_1}, Net_{N_2} and Net_{HC_P} one can detect smaller fault severities than other networks. The minimum detectable fault using this new parallel structure is given in Table 3.8.

Comparing Table 3.8 with the results for the Net_{LC_P} or Net_{N_2} one can observe

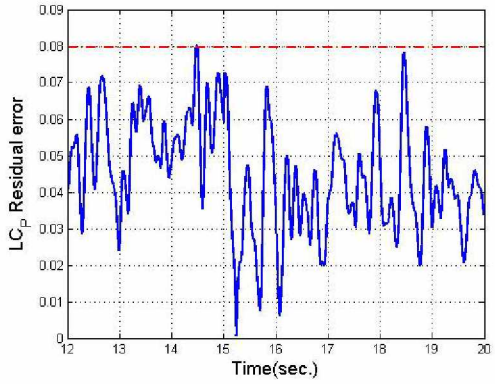
Fault Scenarios	FmLC	FeLC	FmHC	FeHC	FmHT	FeHT	FmLT	FeLT	Median
Trained Network	%	%	%	%	%	%	%	%	%
$Net_{LC_P}/Net_{HC_P}/Net_{N_1}/Net_{N_2}$	2	8	3	3	2	2	2	6	3.50

Table 3.8: The minimum detectable fault severity (%) using the enhanced fault detection system and dynamic neural models.

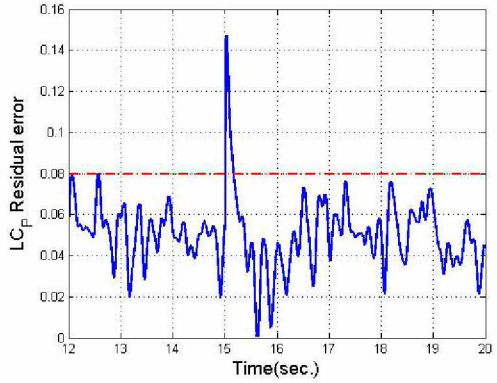
that better results are achieved and all the faults can be detected using the new structure while in the previous structure which was developed by using only one residual generator there was generally at least one fault scenario that could not have been detected.



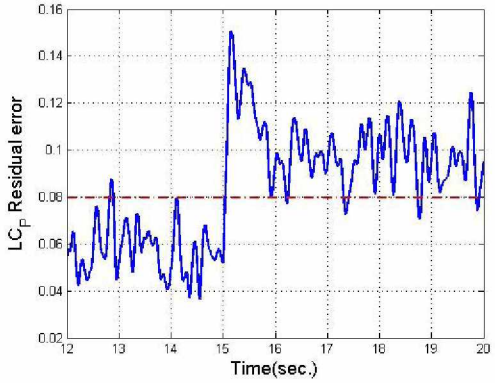
(a) Net_{LC_P} residual error signal when FmLC = 5% with accuracy=94% and precision=94% .



(b) Net_{LC_P} residual error signal when FeLC = 5% with accuracy=0% and precision=0%.

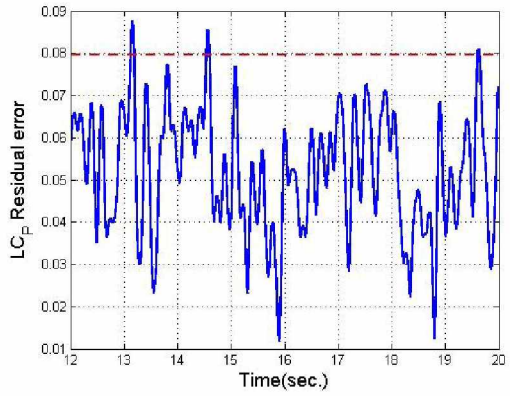


(c) Net_{LC_P} residual error signal when FmHC = 5% with accuracy=39% and precision=3%.

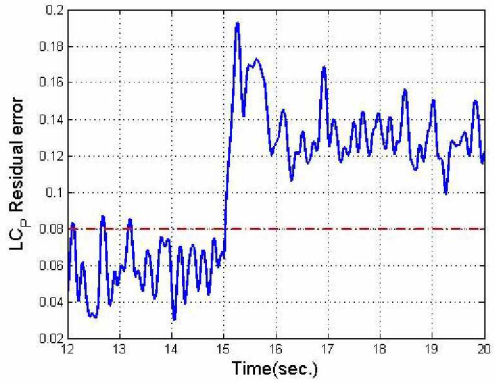


(d) Net_{LC_P} residual error signal when FeHC = 5% with accuracy=95% and precision=93%.

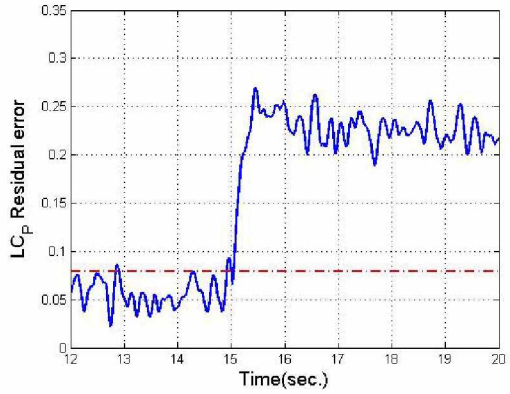
Figure 3.20: The Net_{LC_P} residuals for the fault case when the input fuel flow is at the 70% of the maximum value and the fault severity is at 5%.



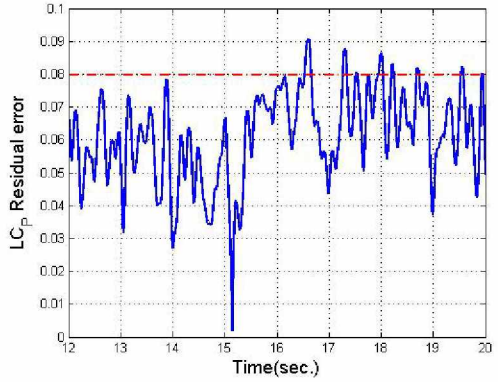
(a) Net_{LC_P} residual error signal when FmHT = 5% with accuracy=36% and precision=14%.



(b) Net_{LC_P} residual error signal when FeHT = 5% with accuracy=97% and precision=100%.



(c) Net_{LC_P} residual error signal when FmLT = 5% with accuracy=98% and precision=100%.



(d) Net_{LC_P} residual error signal when FeLT = 5% with accuracy=53% and precision=13%.

Figure 3.21: The Net_{LC_P} residuals for the fault case when the input fuel flow is at the 70% of the maximum value and the fault severity is at 5%.

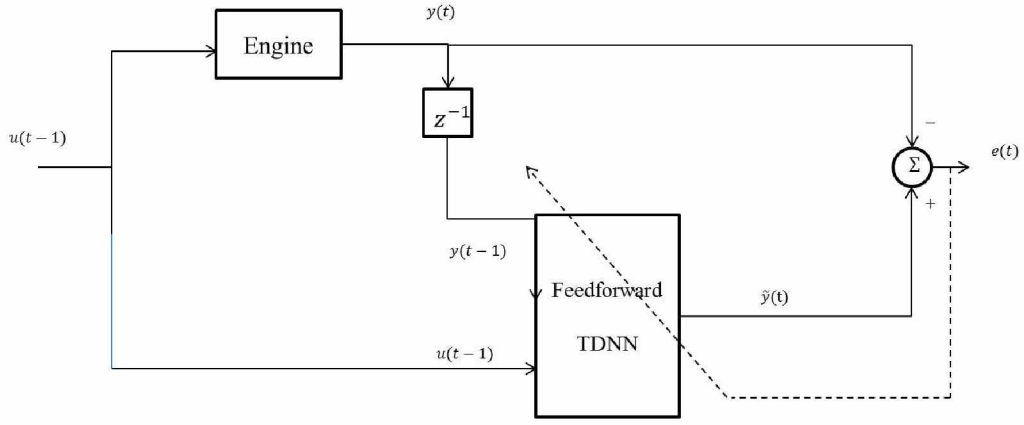


Figure 3.22: The TDNN fault detection scheme - Training phase.

3.2 TDNN Approach

As presented in Chapter 2, a TDNN should be utilized in a series-parallel structure to be able to model the dynamics of the jet engine. Hence, a delayed sample of the output is required as an additional input while the network is being trained. This can make our proposed TDNN be seen as a multi-input single-output (MISO) network as shown in Fig. 3.22.

In order to perform a fault diagnosis task with the trained neural network a modified architecture is applied as presented in Fig. 3.23. This is motivated by the observation that after the training phase, one can assume that the difference between the actual output of the system and its estimated value by the neural network is sufficiently small. Therefore, one can employ the network's own delayed output as an input to the neural network.

The development procedure for this fault detection scheme is the same as the DNM model that was presented in the previous section. In other words, this new fault detection scheme also consist of the system identification phase and the fault detection and residual generation phase.

In the ATDNN that is presented by Yazdiadeh [50] the delays are also updated

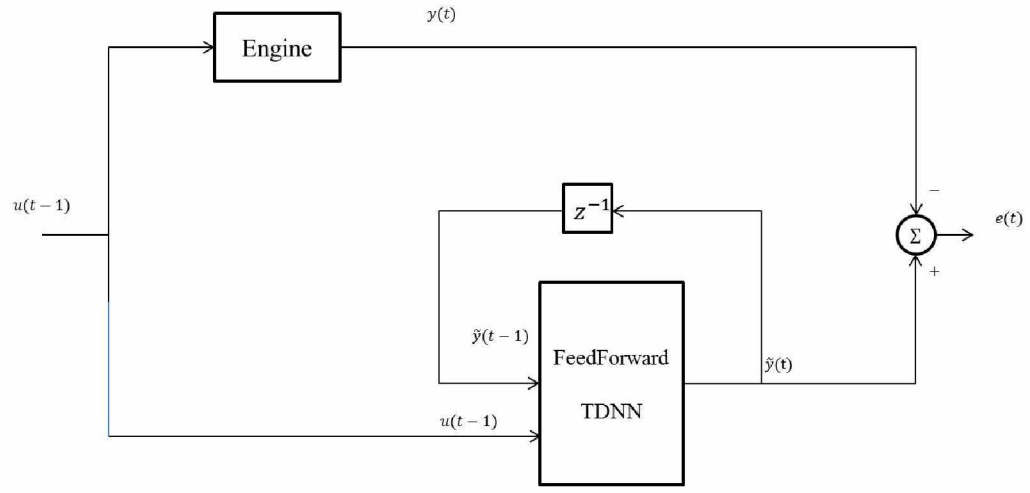


Figure 3.23: The TDNN fault detection scheme - Recall phase

along with the weights. However, there is only one connection between each two neuron in that study. In our proposed TDNN there are several connections between two neurons of different layers depending on the number of delays. Moreover, in this thesis the delay for each layer is fixed and should be defined at the beginning of training. Hence, the number of delays for each layer, the number of layers and the number of neurons all play an important role in the performance of the trained network. In our proposed TDNN each network is trained by using the Levenberg-Marquardt backpropagation algorithm [62]. The networks are trained by using training samples under the presence of noise levels as presented in Table 3.1.

The engine is considered to be operating in the cruise mode and is fault free during the training mode. A zigzag signal ranging between 40% and 98% of the maximum fuel mass flow rate is chosen for training the network. The activation functions in the hidden layers are the hyperbolic tangent and linear activation function for the output layer. Starting from a relatively small structure, we developed an optimal architecture for the proposed dynamic networks by incrementally increasing the number of neurons in the hidden layers. We observed that by using a TDNN with one hidden layer and 5 neurons in the hidden layer, the smallest training error (in the mean square error

sense) can be achieved. The termination criterion is used as t.c.=10^{-5} and the training is stopped if the mse is smaller than the t.c. or if the number of training iterations reaches to 20. The training signals and errors for the neural networks Net_{LT_T}, and Net_{N_2} are depicted in Fig. 3.24-3.25. Table 3.10 summarizes the number of the iterations and the mean square error for each network during the training phase.

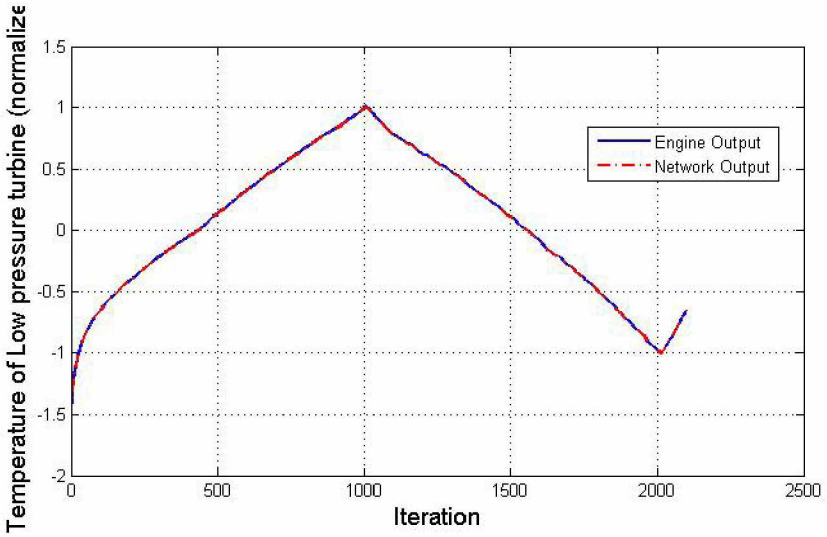
In order to make sure that the neural network is trained well, it should be tested with another "unseen" input to guarantee the stability and the performance of the neural network. Hence, to select the proper neural networks for fault detection, the trained networks were used in the recall phase where the network output is used as an additional input to the network. If the output of the trained network can follow the actual output while being used in the recall phase it would be selected as the proper neural network for our fault detection system. Towards this end, we have trained each network with a set of different delays and the trained network is then utilized in the recall structure to evaluate its performance. The structure that yields the best performance in the recall phase is then used for our fault diagnosis scheme. Table 3.9 shows the structure of the networks and the delays for each neural network.

Trained Network	Delay associated with the weights of the first layer	Delay associated with the weights of the second layer
Net_{LC_P}	3	3
Net_{LC_T}	2	3
Net_{HC_P}	2	3
Net_{HC_T}	1	2
Net_{HT_T}	2	6
Net_{HT_P}	2	1
Net_{LT_T}	3	4
Net_{LT_P}	3	5
Net_{N1}	1	4
Net_{N2}	1	3
Net_{CC_T}	6	2
Net_{CC_P}	6	3

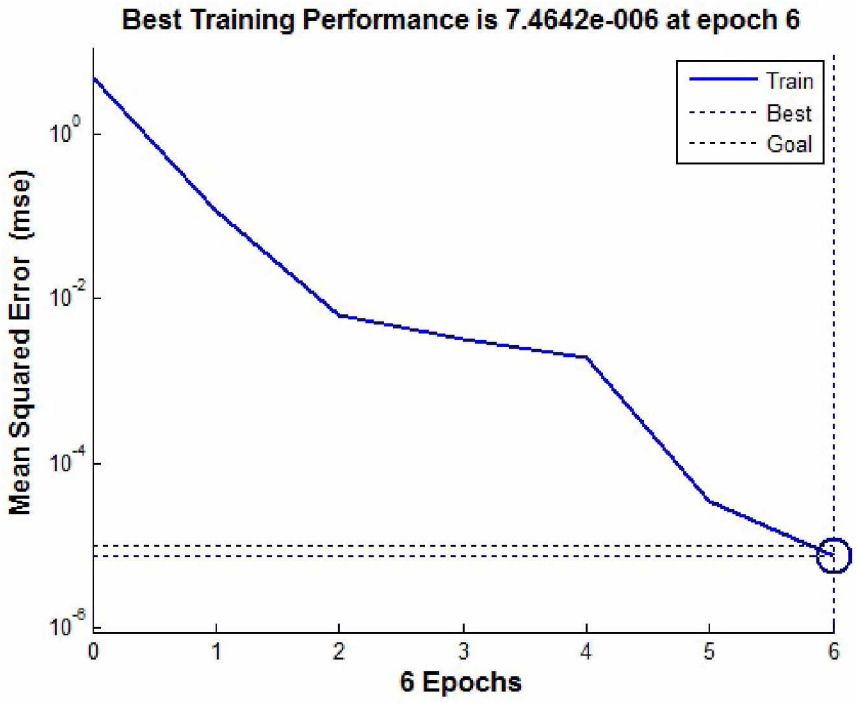
Table 3.9: The delays that are associated with each network.

The Trained Network	No. of Iterations	MSE
Net_{LC_T}	8	$9.76 * 10^{-6}$
Net_{LC_P}	20	$1.05 * 10^{-4}$
Net_{HC_T}	5	$1.38 * 10^{-6}$
Net_{HC_P}	7	$6.92 * 10^{-6}$
Net_{HT_T}	9	$5.62 * 10^{-6}$
Net_{HT_P}	7	$9.06 * 10^{-7}$
Net_{LT_T}	6	$7.46 * 10^{-6}$
Net_{LT_P}	9	$2.86 * 10^{-6}$
Net_{N1}	20	$2.33 * 10^{-5}$
Net_{N2}	15	$9.27 * 10^{-6}$
Net_{CC_T}	8	$4.09 * 10^{-6}$
Net_{CC_P}	9	$2.15 * 10^{-7}$

Table 3.10: The series-parallel TDNN training parameters.

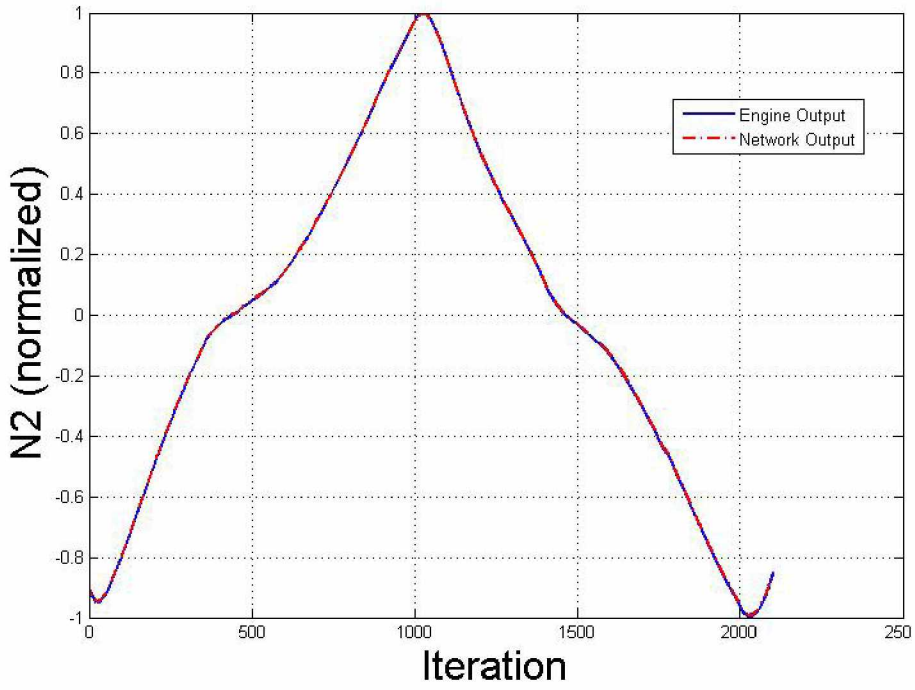


(a) The training signal for the network Net_{LT_T}.

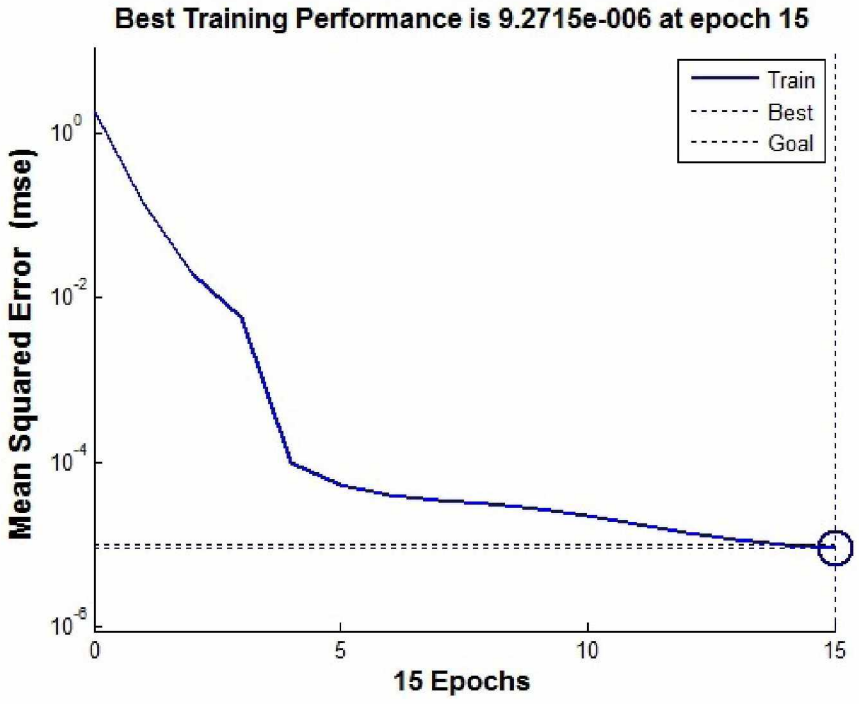


(b) The training error profile for the network Net_{LT_T}.

Figure 3.24: The training signal and the training error for the network Net_{LT_T}.



(a) The training signal for the network Net_{N_2}.



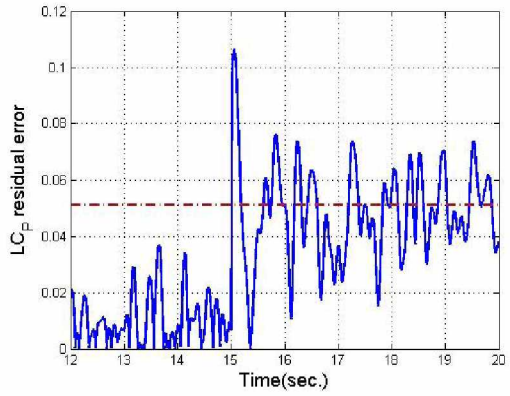
(b) The training error profile for the network Net_{N_2}.

Figure 3.25: The training signal and the training error for the network Net_{N_2}.

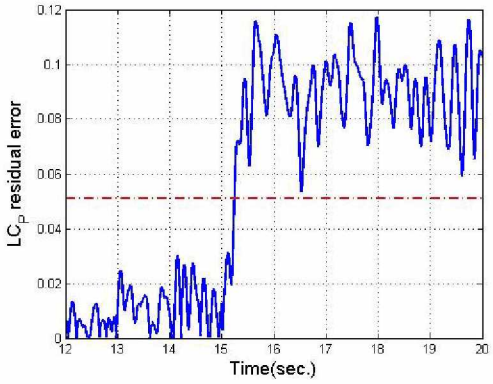
Fault Severity Level (%)	Accuracy (%)	Precision (%)	Confusion Matrix
FmLC=2%	44.69	11.25	50 73 0 9
FmLC=3%	50.75	21.77	50 65 0 17
FmLC=4%	61.28	38.52	49 51 1 31
FmLC=5%	74.28	59.75	50 34 0 48
FmLC=6%	84.3	74.9	50 21 0 61

Table 3.11: The detection performance for the network Net_{LC_P} (Note that the values in the entries of the confusion matrix refer to the actual number of residual signal samples representing the corresponding characteristic).

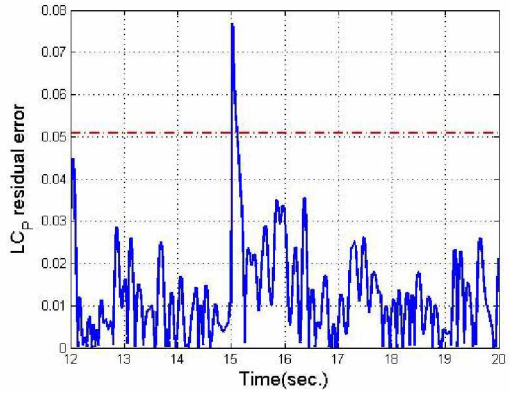
Our proposed fault diagnosis approach is used to determine the minimum detectable fault for each network. The set of 8 fault scenarios that are presented in the previous section is considered. Table 3.12 shows the performance of the networks. A fault severity is considered as detectable if the averages of the accuracy and precision parameter corresponding to the network are greater than 80%. Table 3.11 shows the detecting performance of the fault FmLC for the network Net_{LC_P}. One can observe that when the fault severity is 6%, the averages of the accuracy and precision are greater than 80% and the fault would be considered as detectable. To evaluate the results of our proposed fault detection scheme and to demonstrate the results associated with the networks, in Table 3.12 the residuals of Net_{N2} and Net_{LCP} are shown in presence of a 5% fault under various fault scenarios. Figures 3.26 and 3.28 show the residual of the networks Net_{LCP} and Net_{N2} to these faults, respectively.



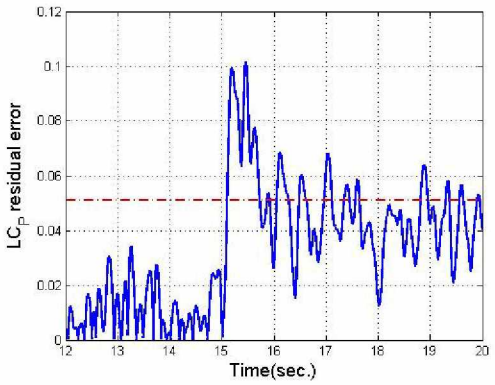
(a) Net_{LC_P} residual error signal when FmLC = 5% with accuracy=63% and precision=40%.



(b) Net_{LC_P} residual error signal when FeLC = 5% with accuracy=100% and precision=100%.

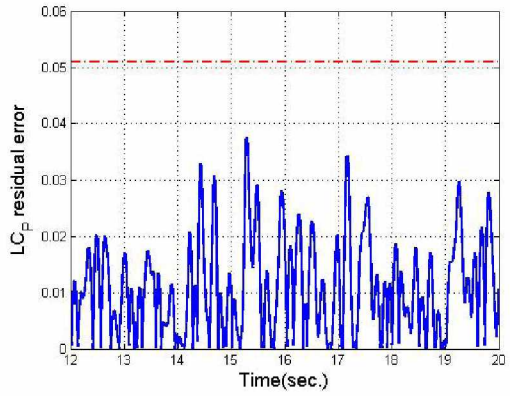


(c) Net_{LC_P} residual error signal when FmHC = 5% with accuracy=39% and precision=2%.

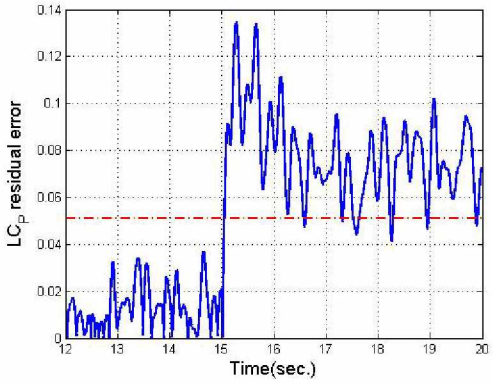


(d) Net_{LC_P} residual error signal when FeHC = 5% with accuracy=58% and precision=33%.

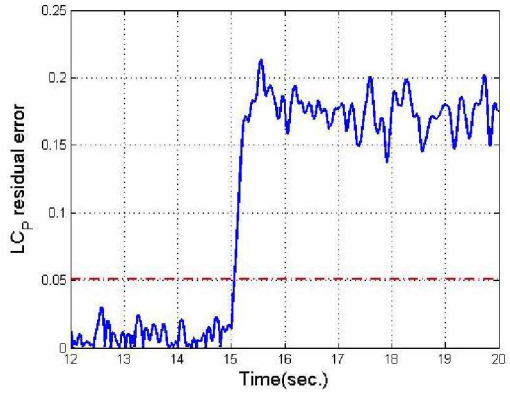
Figure 3.26: The TDNN Net_{LC_P} residuals for the fault case when the input fuel flow is at the 70% of the maximum value and the fault severity is at 5%.



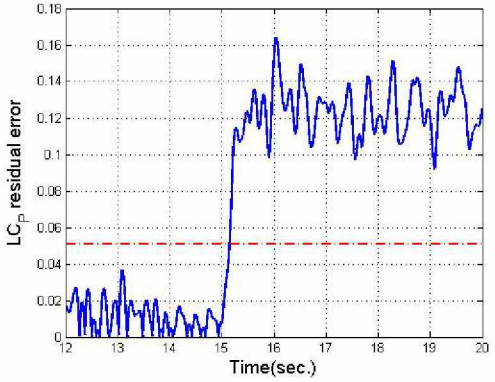
(a) Net_{LC_P} residual error signal when FmHT = 5% with accuracy=0% and precision=0%.



(b) Net_{LC_P} residual error signal when FeHT = 5% with accuracy=95% and precision=92%.

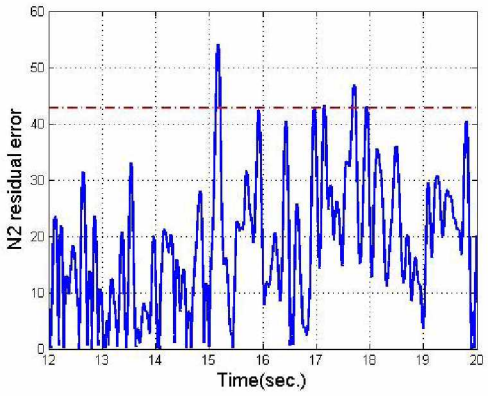


(c) Net_{LC_P} residual error signal when FmLT = 5% with accuracy=100% and precision=100%.

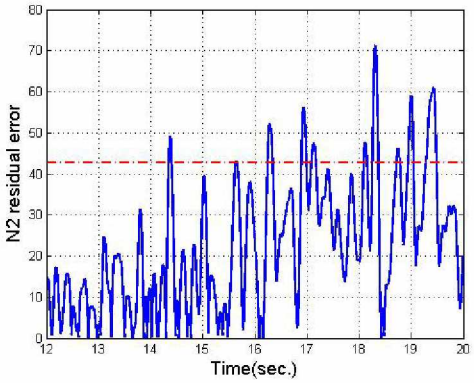


(d) Net_{LC_P} residual error signal when FeLT = 5% with accuracy=100% and precision=100%.

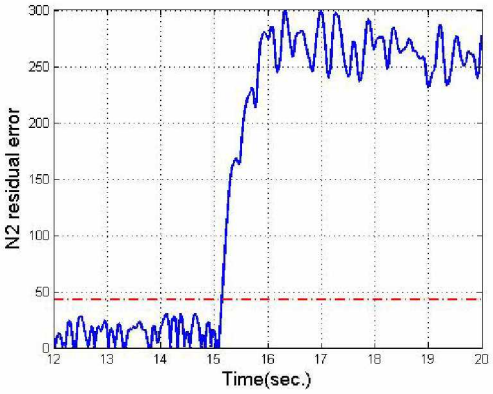
Figure 3.27: The TDNN Net_{LC_P} residuals for the fault case when the input fuel flow is at the 70% of the maximum value and the fault severity is at 5%.



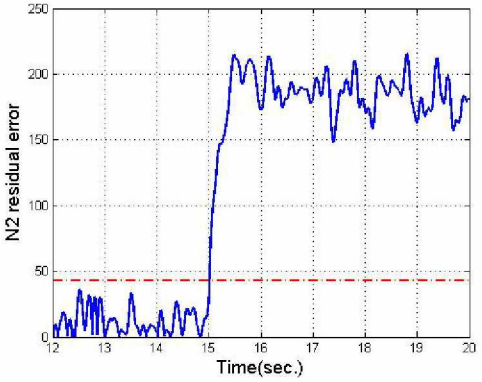
(a) Net_{N2} residual error signal when FmHC = 5% with accuracy=49% and precision=14%.



(b) Net_{N2} residual error signal when FeHC = 5% with accuracy=60% and precision=25%.

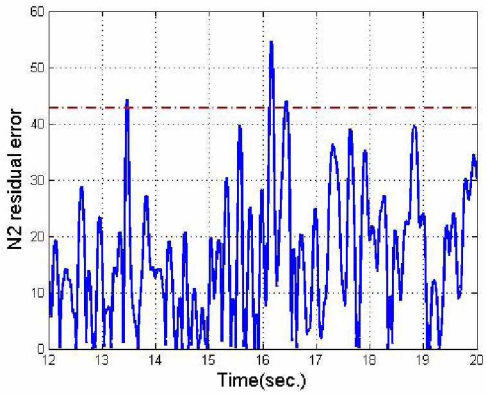


(c) Net_{N2} residual error signal when FmLC = 5% with accuracy=98% and and precision=100%.

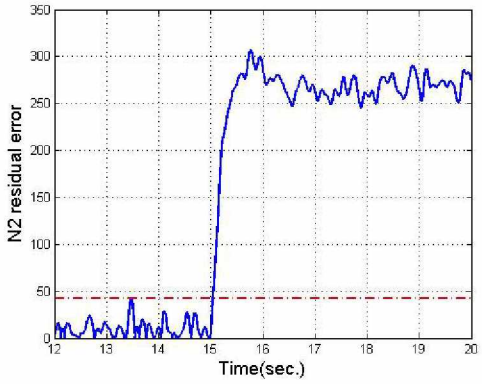


(d) Net_{N2} residual error signal when FeLC = 5% with accuracy=100% and precision=100%.

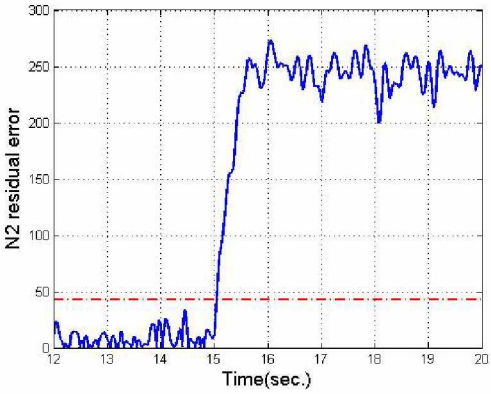
Figure 3.28: The TDNN Net_{N2} residuals for the fault case when the input fuel flow is at the 70% of the maximum value and the fault severity is at 5%.



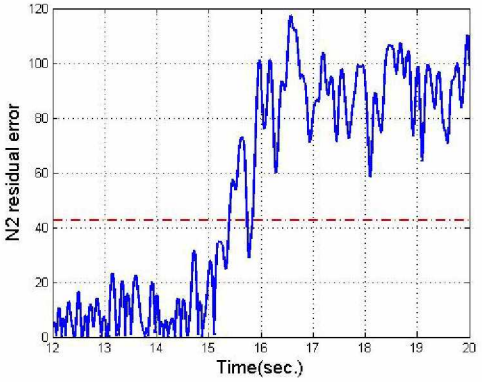
(a) Net_{N2} residual error signal when FmHT = 5% with accuracy=44% and precision=4%.



(b) Net_{N2}residual error signal when FeHT = 5% with accuracy=100% and precision=100%.



(c) Net_{N2} residual error signal when FmLT = 5% with accuracy=100% and precision=100%.



(d) Net_{N2} residual error signal when FeLT = 5% with accuracy=98% and and precision=97%.

Figure 3.29: The TDNN Net_{N2} residuals for the fault case when the input fuel flow is at the 70% of the maximum value and the fault severity is at 5%.

Fault Scenarios Trained Network	FmLC %	FeLC %	FmHC %	FeHC %	FmHT %	FeHT %	FmLT %	FeLT %
Net_{LC_T}	N	N	N	7	N	4	3	4
Net_{LC_P}	6	4	N	7	N	5	2	3
Net_{HC_T}	N	7	12	4	4	9	N	10
Net_{HC_P}	N	N	N	6	2	4	7	N
Net_{HT_T}	N	N	N	6	N	4	6	N
Net_{HT_P}	N	8	N	3	N	3	3	9
Net_{LT_T}	N	5	N	2	N	2	3	5
Net_{LT_P}	N	12	N	5	N	4	7	N
Net_{N1}	1	3	N	2	N	2	7	4
Net_{N2}	10	5	2	2	7	1	2	3
Net_{CC_T}	N	7	N	3	10	3	5	N
Net_{CC_P}	N	8	N	3	2	2	4	11

Table 3.12: The minimum detectable fault severity (%) for the series-parallel TDNN.

3.2.1 Discussions

It can be seen from the results of Table 3.12 that Net_{N1} and Net_{N2} have the best performance among all the other networks. One can conclude that having Net_{N1} as a residual generator one can detect on average faults as low severity as 3%. However, the performance of the other networks are dependent on the fault scenarios that are considered. One can also note that for instance Net_{HT_T} is only able to detect three fault scenarios out of eight.

Enhanced Fault Diagnosis Scheme

By applying our bank of parallel fault detection filters that use several residual generators as presented in the previous section from Table 3.12 it follows that by utilizing Net_{N_1}, Net_{N_2} and Net_{CC_P} one can detect lower severity faults.

The selected threshold levels and the minimum detectable faults are given by

Trained Network	Threshold
Net_{LC_T}	4.29 °C
Net_{LC_P}	0.05 atm
Net_{HC_T}	7.93 °C
Net_{HC_P}	0.44 atm
Net_{HT_T}	49.53 °C
Net_{HT_P}	0.07 atm
Net_{LT_T}	9.37 °C
Net_{LT_P}	0.045 atm
Net_{N1}	34.77 rpm
Net_{N2}	42.75 rpm
Net_{CC_T}	19.83 °C
Net_{CC_P}	0.24 atm

Table 3.13: The threshold levels for series-parallel TDNN.

Fault Scenarios Trained Network	FmLC %	FeLC %	FmHC %	FeHC %	FmHT %	FeHT %	FmLT %	FeLT %	Median %
Net_{N_1}, Net_{N_2} and Net_{CC_P}	1	3	2	2	2	1	2	3	2

Table 3.14: The minimum detectable fault severity (%) for the enhanced fault detection system.

Tables 3.13 and 3.14, respectively.

3.2.2 The SISO TDNN Fault Detection Approach

As presented earlier in this chapter, the DNM fault detection scheme is in the form of a single-input single-output (SISO) system while the TDNN fault detection scheme is constructed as a multi-input single-output (MISO) system where the additional inputs are the delayed versions of the output itself. In this section, an experiment is conducted to determine how a TDNN can perform if it is utilized in a SISO form

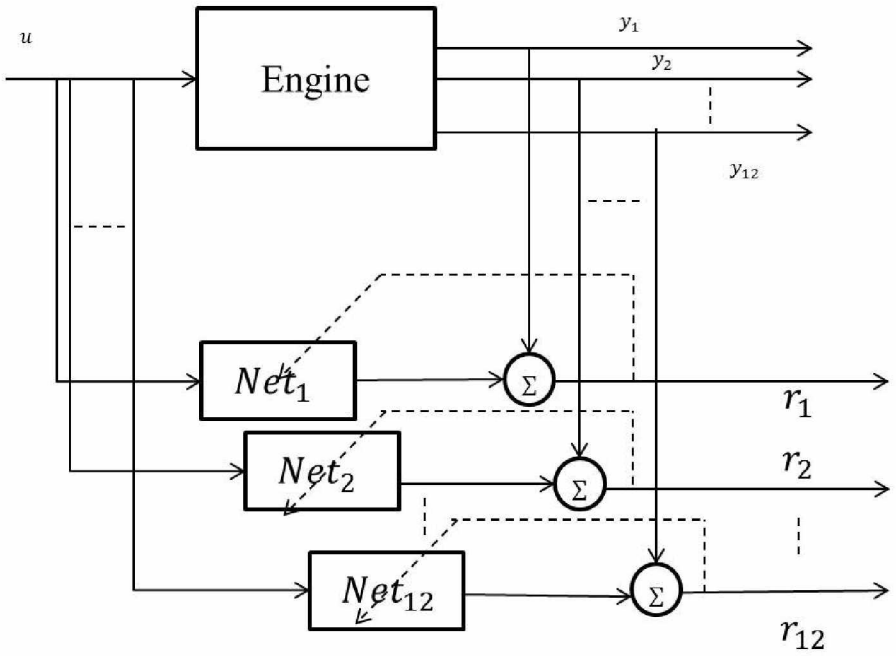


Figure 3.30: Fault detection structure by using the SISO TDNN.

similar to the DNM model. Hence, we use the TDNN that was presented in Chapter 2 as a SISO structure where the network employs only one input and one output where the input is the fuel flow rate. By eliminating the feedback from the output of the system that is shown in Fig. 3.22, the structure of the network would be the same as the DNM model and in shown in Fig. 3.30. As discussed in Chapter 2 for the DNM model, the structure of the network for the SISO TDNN would remain the same in the training and the recall phases. One may note that although the actual output is not appearing in the input-output relationship of the network, however, it is appearing in the network parameters indirectly by means of the back-propagated error which is used to update the weights. In this structure the output of the TDNN would be only a function of the input and its past values. In this section, we will discuss the performance of this structure for the problem of fault detection of the aircraft engine.

Twelve (12) networks are trained by using the zigzag signal that was utilized in

The Trained Network	No. of Iterations	MSE
Net_{LC_T}	8	0.0009
Net_{LC_P}	20	0.017
Net_{HC_T}	8	0.0006
Net_{HC_P}	4	0.0002
Net_{HT_T}	8	0.0009
Net_{HT_P}	20	0.0011
Net_{LT_T}	20	0.0013
Net_{LT_P}	7	0.0004
Net_{N1}	20	0.0035
Net_{N2}	20	0.0017
Net_{CC_T}	4	0.0004
Net_{CC_P}	20	0.0007

Table 3.15: The SISO TDNN training parameters.

the previous sections. The termination criterion is used as t.c.=10^{-3} and the training is stopped if the mse is smaller than the t.c. or if the number of training iterations reaches to 20. Table 3.15 summarizes the number of the iterations and the mean square error for each network during the training phase and Table 3.16 provides the selected thresholds for each network. The trained network is used in the recall stage to detect the minimum detectable faults in the jet engine. The results are shown in Tables 3.17 and 3.18.

3.3 Conclusions

In this chapter a comparative study is provided first between the DNM approach and the TDNN series-parallel approach, followed by discussions on the results using the SISO TDNN approach.

Trained Network	Threshold
Net_{LC_T}	7.30 °C
Net_{LC_P}	0.045 atm
Net_{HC_T}	15.41 °C
Net_{HC_P}	0.36 atm
Net_{HT_T}	21.82 °C
Net_{HT_P}	0.08 atm
Net_{LT_T}	16.24 °C
Net_{LT_P}	0.042 atm
Net_{N1}	51.054 rpm
Net_{N2}	71.71 rpm
Net_{CC_T}	29.71 °C
Net_{CC_P}	0.33 atm

Table 3.16: The threshold levels for SISO TDNN.

Fault Scenarios Trained Network	FmLC %	FeLC %	FmHC %	FeHC %	FmHT %	FeHT %	FmLT %	FeLT %
Net_{LC_T}	N	N	N	9	N	7	4	5
Net_{LC_P}	7	4	N	7	N	5	2	3
Net_{HC_T}	N	12	N	7	6	N	N	N
Net_{HC_P}	N	9	N	4	2	2	4	12
Net_{HT_T}	N	9	N	3	N	3	4	N
Net_{HT_P}	N	8	N	3	N	3	4	10
Net_{LT_T}	N	8	N	3	N	2	5	8
Net_{LT_P}	N	10	N	4	N	3	6	N
Net_{N1}	2	3	N	2	N	2	10	4
Net_{N2}	N	6	2	3	9	2	2	4
Net_{CC_T}	N	8	N	4	12	4	7	N
Net_{CC_P}	N	9	N	3	2	2	4	12

Table 3.17: The minimum detectable fault severity (%) for the SISO TDNN.

Fault Scenarios Trained Network	FmLC %	FeLC %	FmHC %	FeHC %	FmHT %	FeHT %	FmLT %	FeLT %	Median %
Net_{N_1}, Net_{N_2}, Net_{LC_P} and Net_{CC_P}	2	3	2	2	2	2	2	3	2.25

Table 3.18: The minimum detectable fault severity (%) using the enhanced fault detection system in the SISO TDNN.

In order to present our comparative study more clearly, a set of 5 engine parameters are chosen. These parameters are chosen by selecting those with better performance as well as those with worst performance. These parameters are Net_{N2}, Net_{N1}, Net_{LC_P}, Net_{CC_P}, Net_{HT_T}. The first four parameters are the outputs that have good detection capabilities and the last has the least capability in the fault detection.

A summary of the results presented for each approach is provided in Table 3.19. Comparing the results in tables for the TDNN and the DNM one can observe at first glance that the series-parallel TDNN seems to have better performance and the fault detection scheme using this approach can detect smaller fault severities in the jet engine system. For instance for the N_2 parameter it can be seen that the series-parallel TDNN method can detect a fault as small as 3% on average while for the DNM the fault is only at 5%. However, there are several other factors that should be considered for evaluating the performance of these networks.

As we have previously stated the series-parallel TDNN requires a different structure in the recall phase where it employs a feedback from the network output instead of the system output. This structure makes the system highly sensitive to the training phase, the number of neurons and the structure of the network. In other words, when the network is trained several times with the same structure and the same termination criterion, the testing results for each network might be different. Note that one should test the network in the recall phase to ensure that the network performs satisfactorily, otherwise an unacceptable and undesirable error may be possible. Hence,

Fault Scenarios Trained Network	FmLC %	FeLC %	FmHC %	FeHC %	FmHT %	FeHT %	FmLT %	FeLT %
Net_{N2}	10	5	2	2	7	1	2	3
Net_{N1}	1	3	N	2	N	2	7	4
Net_{LC_P}	6	4	N	7	N	5	2	3
Net_{CC_P}	N	8	N	3	2	2	4	11
Net_{HT_T}	N	N	N	6	N	4	6	N

Series-parallel TDNN approach.

Fault Scenarios Trained Network	FmLC %	FeLC %	FmHC %	FeHC %	FmHT %	FeHT %	FmLT %	FeLT %
Net_{N2}	N	6	2	3	9	2	2	4
Net_{N1}	2	3	N	2	N	2	10	4
Net_{LC_P}	7	4	N	7	N	5	2	3
Net_{CC_P}	N	9	N	3	2	2	4	12
Net_{HT_T}	N	9	N	3	N	3	4	N

SISO TDNN approach.

Fault Scenarios Trained Network	FmLC %	FeLC %	FmHC %	FeHC %	FmHT %	FeHT %	FmLT %	FeLT %
Net_{N2}	12	10	3	3	11	2	2	6
Net_{N1}	2	N	N	N	N	N	N	N
Net_{LC_P}	6	8	N	7	N	4	2	6
Net_{CC_P}	N	12	N	5	2	3	6	N
Net_{HT_T}	N	N	N	5	N	3	5	N

DNM appraoch.

Table 3.19: Comparison between the five (5) parameters of the engine corresponding to the three approaches.

the trained network should be examined before being used to function as a residual generator in the fault detection module. This makes its use difficult to apply in some practical fault diagnosis problems.

In contrast, the DNM network is robust to the above issue and as long as the training error is satisfied, one can be highly confident that the network is operating satisfactory in the recall phase. In other words, as long as the training error remains the same the network will almost surely have the same behaviour in the recall phase.

Another advantage of the DNM approach is that there is no need to a feedback from the output of the system as the internal IIR filters of the neurons generate the desired dynamics in the network. Hence, this network can be used as a SISO structure. Moreover, the structure of the DNM network remains the same during the training and the recall phases. In addition, the on-line parameter updating of the DNM approach makes it more suitable for practical applications where it can be used in an on-line diagnostic system.

An interesting observation can be made from our simulations of the SISO TDNN method. As stated in the previous section this structure may not be fully capable of learning the dynamics of the engine by itself as there is no delayed version of the output in the final approximated function of the network. However, simulation results show that one can achieve an acceptable performance for jet engine in terms of the fault detection performance as shown in Table 3.19. This can be explained as follows since the fault detection scheme is operating on the condition that faults are occurring in the steady state of the engine and the effects of the faults in the transient states of the engine are not significant. Hence, as far as the steady state of the fault detection scheme is considered this approach can be successfully used. However, we have observed that the SISO TDNN is not capable of learning the transient overshoots in the engine, therefore it would not be a reliable approach for fault detection in the

transient stage of the engine.

Chapter 4

Neural Network-based Fault Isolation Scheme

Our objective in this chapter is to perform the fault isolation task corresponding to different fault scenarios that are considered in this thesis. Towards this end, a multi-layer neural network classifier is employed to accomplish the fault isolation task. The residual signals are first analyzed by using a residual evaluation block that will then be used as inputs to the classifier. Our fault isolation approach is developed to work with a number of fault cases as we have limited our study to only three possible fault scenarios that can occur in the aircraft jet engine. We will show that by using the three residuals at the same time one can reach acceptable fault isolation by using a multi-layer perceptron (MLP) network. Hence, the classifier will have three inputs which are the enhanced residuals and three outputs which are the fault labels and are determined by using a binary code corresponding to the three fault scenarios that occur in the jet engine. However, in order to have concrete results we have selected five sets of fault that each consists of three fault scenarios as shown in this chapter in Table 4.2.

4.1 Proposed Fault Isolation Scheme

Based on several experiments conducted we have observed that the change in the residual percentage has valuable information for fault isolation of the jet engine while several residual generators are employed and work together in parallel. On the other hand, the percentage of change of several residuals form a pattern which corresponds to the type of the fault scenario that is generating the residuals. For example, one can observe that the pattern which is generated between the three residuals for a 2% fault of FeHT is different from the pattern of a 2% fault in FmLT. We have also observed that this pattern is also sufficiently robust to the level of the fault. This enables us to isolate the fault scenario no matter what is the percentage of the fault that is occurring in the system. One should note that in this thesis we only consider the fault isolation problem and we are not interested in the fault identification problem.

The isolation process consists of two subsystems namely, the neural network classifier and the residual evaluation block. The inputs to the neural network classifier are a static numerical value while the residuals are time-series data. Hence, in order to transform the residuals into meaningful quantities for the classifier a series of feature extraction methods should be applied to the residual signals. This process is denoted as the residual evaluation. The structure of the general fault isolation schematic is depicted in Fig. 4.1.

Based on many experiments conducted it turned out that the percentage of change in the magnitude of the residual signal when a fault occurs can yield a significant indicator to the fault isolation task. Hence, we have developed a residual evaluation block which is fed by the residual signals to detect the faults and translate the dynamic residual signal into a static numerical variable for application and use in a neuro classifier. The faults are initially detected and then the percentage of the change in the magnitude of the residuals are measured for use in the neural network classifier.

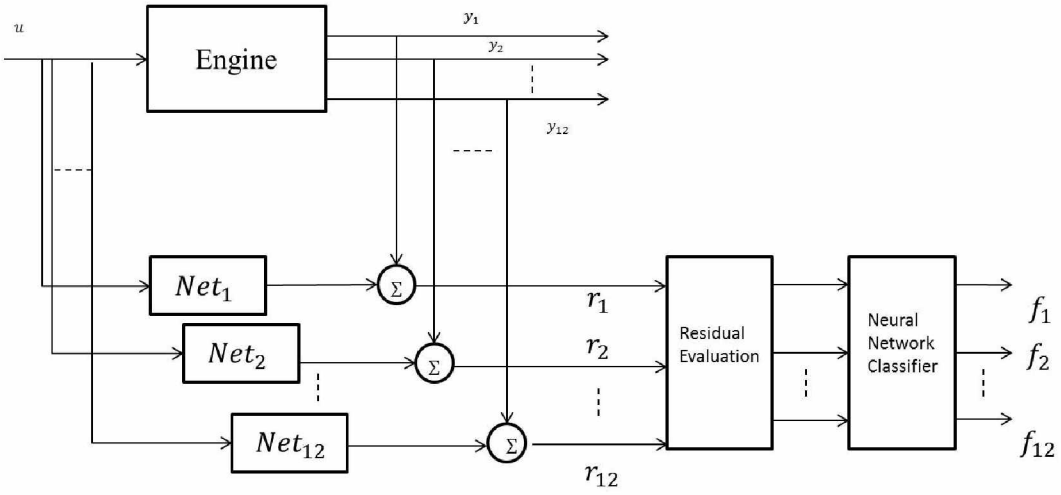


Figure 4.1: The fault isolation scheme.

4.1.1 Neural Network Classifier

We have employed a multi-layer perception (MLP) network as a neuro classifier for the task of fault isolation. The classifier employs the evaluated residual signals as inputs. The outputs of the classifier would be the fault labels corresponding to each fault class. We have observed that while several residual generators are employed together there would be a specific relationship among them corresponding to a fault class. By training the neuro classifier with several samples of data, it can learn this relationships among the inputs and can be employed as a classifier for fault isolation of the jet engine.

Our proposed fault isolation scheme is designed to isolate among three possible fault severity levels. Basically we assume that the engine is exposed to only three possible fault scenarios at each stage. However, we have examined different combinations of the faults to cover almost all the possible scenarios. Fig. 4.2 shows our fault isolation scheme.

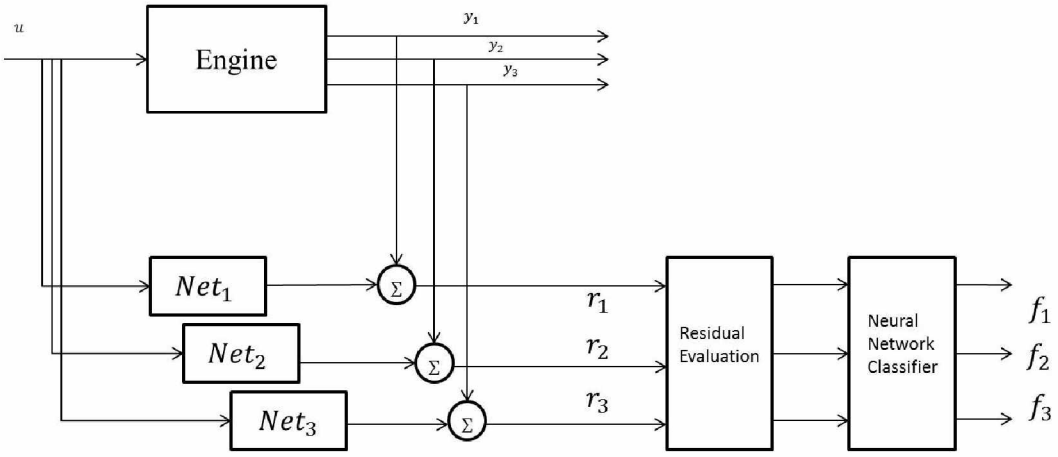


Figure 4.2: The proposed fault isolation scheme.

4.1.2 Residual Evaluation

The neural network classifier requires a numerical static value as an input whereas the residuals are time series data. In order to make the residuals meaningful to the classifier a number of feature extraction methods can be applied to the residuals.

The main task performed in the residual evaluation module is to determine the maximum change in the residual when a fault occurs. The percentage of this change from the steady state before the fault occurs would be an input to the classifier. We have observed that the maximum change in each output by occurrence of a specific fault is generally in the same range for all the input profiles (0.7 to 0.9 of the maximum fuel mass flow rate). Hence, the maximum value of the change in the residual would be a key indicator for the fault isolation. However, one should note that a residual cannot individually be used for isolation. Hence, a set of three residuals is required to construct a map for isolating the corresponding faults in the jet engine.

In the residual evaluation block the maximum residual change is calculated as the difference between the maximum change after the occurrence of a fault to the value of the residual before the fault. To consider a margin of error factor and to take into account inaccuracies due to noise and disturbances we took an average from the

residual signal after and before the fault occurs in a window of size three seconds. We have observed that the residuals normally reach to their steady state after the fault occurs in less than 2 seconds. The output of the residual evaluation module is the percentage of the change in the output. On the other hand, the input to the neural network classifier would be the the maximum change in the residual over the steady state value of the residual before the occurrence of the fault.

4.2 Fault Isolation Scheme

As described earlier, the relationships that are generated between different residuals can have valuable information on the types of the faults which are occurring. We have observed that by using three residuals an acceptable fault performance isolation can be achieved. Considering more than three residuals will increase the size of the network and the computational cost of the system and not necessarily result in more improved performance. In contrast, having less than three residuals one may then not be able to achieve a desirable fault isolation performance.

We have chosen the three residuals that are obtained in the enhanced fault detection module for both the DNM model and the SISO TDNN model for the fault isolation task. As stated earlier, in each simulation a set of three possible faults will be considered. We have selected different sets of three faults based on the fault severity levels which are labelled as High, Normal and Low, as shown in Table 4.1. We have selected different combinations of these faults as shown in Table 4.2 and applied our fault isolation approach to each case separately.

A database of all the possible residual signals for our three residual generator networks was collected for each isolation case as shown in Table 4.2. Each database contains 180 samples that are generated when the engine is operating in 5 different input levels and under the presence of faults between 1% to 12% of severities.

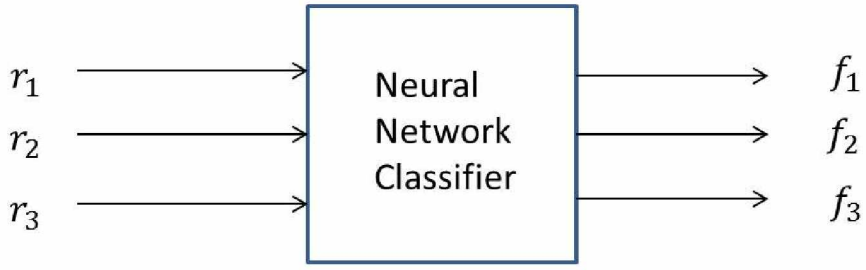


Figure 4.3: A neural network classifier.

Considering the three faults at a time, one can observe that the total samples are $5 * 12 * 3$, which generate 180 sets of residuals. Each sample (input and target pairs for the MLP) have three residual signals (namely r_1, r_2 and r_3) as inputs and the corresponding fault code targets (f_1, f_2 and f_3).

Having this database prepared, one can then perform the isolation task by using the MLP networks and standard training, testing and validation techniques. To make our proposed approach close to a practical system, we assume that certain information on the faulty events in the jet engine system that correspond to the fault scenarios and the maintenance actions that have been performed in the log and report files of the service and maintenance department. This information can be used as training samples for our MLP networks. Moreover, since in practice there are not too many log files to cover all the possible fault scenarios in the jet engine, our objective is to minimize the number of the training samples while to keep the performance acceptable. We have observed that by using 25 random samples out of the 180 for training, an acceptable isolation result can be accomplished. All the residuals are generated under the measurement noise as presented in Table 3.1 in Chapter 3.

Fault Scenario	Fault Severity Level
FmLC	Low (L)
FeLC	Normal (N)
FmHC	Low (L)
FeHC	High (H)
FmHT	Low (L)
FeHT	High (H)
FmLT	High (H)
FeLT	Normal (N)

Table 4.1: The fault severity levels.

4.2.1 DNM Isolation

In this section we use the three engine parameters that are presented as enhanced fault detection parameters in Chapter 3. These signals are Net_{LC_P} , Net_{N_1} and Net_{HC_P}. Our objective is to develop a neural network to learn the relationship among these parameters for each fault scenario. Starting from a small structure for the network and increasing the number of neurons in the network we have reached an optimal structure with two hidden layers with 25 and 2 neurons in each hidden layer, respectively. The network was trained by using 25 random sample data and the trained system is tested and validated by using the remaining 155 samples. Table 4.3 shows the fault isolation results for these three residual generators as applied to all the 5 fault sets.

4.2.2 SISO TDNN Isolation Scheme

In this section we use the residuals that are generating by using the proposed SISO TDNN for fault isolation. The structure of the fault isolation module is the same as in the DNM model and three residuals are used to learn the map corresponding to

Case 1	High	High	High
H.H.H	FeHC	FeHT	FmLT
Case 2	Low	Low	Low
L.L.L	FmLC	FmHC	FmHT
Case 3	Low	Normal	High
L.N.H	FmLC	FeLC	FeHT
Case 4	Low	Normal	High
L.N.H	FmHC	FeLC	FeHC
Case 5	Low	Normal	High
L.N.H	FmHT	FeLT	FmLT

Table 4.2: The fault sets that are selected for the fault isolation scheme.

each fault scenario. To perform a comparative study the same number of samples are generated to construct the database. Therefore, the new database would also contain 180 samples that are generated for 5 different input profiles and in presence of different fault levels ranging from 1% to 12% for each of the three faults that are considered at each stage.

The MLP network that is utilized in this section has two hidden layers and the number of neurons in the hidden layers are 25 and 2, respectively. The activation functions are selected as hyperbolic tangent for the hidden layers and linear for the output layer. The MLP network is trained with small random initial weights. The network is trained by using 25 random samples out of the 180 existing samples. The trained network is tested by using "un-seen" samples to evaluate the performance of the fault isolation module. Each network has three outputs corresponding to one class of the faults. A post processing is applied to the outputs of the network to round them to either 0 or 1 where the actual target of the network is defined. The

Case 1	FeHC	FeHT	FmLT	Mean
H.H.H	83%	62%	62%	69%
Case 2	FmLC	FmHC	FmHT	
L.L.L	88%	86%	84%	87%
Case 3	FmLC	FeLC	FeHT	
L.N.H	67%	73%	74%	72%
Case 4	FmHC	FeLC	FeHC	
L.N.H	75%	65%	73%	71%
Case 5	FmHT	FeLT	FmLT	
L.N.H	89%	63%	47%	67%

Table 4.3: The percentage of correct isolation for each set of faults using the DNM.

output of the network is rounded to 1 if it is greater than 0.5 and is rounded to 0 if it is less than 0.5. This adjustment will improve the performance of the fault isolation scheme that is presented in this section.

To achieve a realistic performance and to avoid training for specific samples, the training procedure is repeated 5 times with different sets of 25 random samples. The presented results in Table 4.4 correspond to the average of the 5 networks. All the simulations including the training and the testing are repeated for all the 5 faults sets.

4.3 Conclusions and Discussion

Tables 4.3 and 4.4 show the fault isolation results for the DNM and the SISO TDNN structures. For each case the fault scenario and the percentage of correct isolation are provided. For example, one can see from Table 4.4 that for Case 2, 98% of all the

Case 1	FeHC	FeHT	FmLT	Mean
H.H.H	88%	48%	90%	75.3%
Case 2	FmLC	FmHC	FmHT	
L.L.L	99%	98%	100%	99%
Case 3	FmLC	FeLC	FeHT	
L.N.H	92%	99%	96%	95.7%
Case 4	FmHC	FeLC	FeHC	
L.N.H	89%	92%	96%	92.3%
Case 5	FmHT	FeLT	FmLT	
L.N.H	99%	95%	93%	95.7%

Table 4.4: The percentage of correct isolation for each set of faults using the SISO TDNN.

faults occurring by a decrease in the mass flow rate of the high pressure compressor (FmHC) can be isolated from the other two cases (that is, FmHT and FmLC). It can be seen that the performance of the fault isolation is dependent on the structure of the faults that occur together. For example, one can observe from Table 4.3 that for the fault set Case 2 has a better performance yield compared to the other combination of faults. This implies that the structure of these three faults and the effects they make on the jet engine parameters are quiet distinguishable. We have observed that the majority of the samples that could not be isolated are occurring when the smaller fault severities are present. In other words, a 1% fault in the FeHT has a higher correlation to the 1% FmLT as compared to an 8% fault in the FeHT with that of the FmLT. In other words, the effects of a 1% change in the FeHT and the FmLT in the engine parameters are almost the same and indistinguishable, and the fault isolation scheme is unable to distinguish them. This will make it challenging to isolate these

cases when compared to larger fault severities. Moreover, by inspecting Tables 4.3 and 4.4 and setting aside the Case 2, one can conclude that the faults with different severity levels have better isolation capabilities. However, the nature of the faults and the types of the components where a fault has occurred have an important role in the overall fault isolation performance. Comparing the results of Tables 4.3 and 4.4, one can conclude that in general by using the SISO TDNN residuals one can yield a better fault isolation performance. Moreover, it can be observed that the results are consistent and broadly are in good agreement collectively. In other words, the faults sets of Case 2 are being isolated better in both approaches and the fault sets of Case 1 cannot be isolated quite as good in either approach.

Chapter 5

Conclusions

The objective of this thesis was to develop an intelligent-based approach for fault diagnosis of aircraft engines. Towards this end, two main dynamic neural networks and three separate structures were used to construct fault diagnosis schemes for a dual-spool turbofan jet engine. The neural networks considered in this thesis are the dynamic neural model (DNM) and the time delay neural network (TDNN). The proposed fault diagnosis schemes are based on a DNM, a series-parallel TDNN and a single-input single-output TDNN.

The jet engine considered in this thesis was assumed to have twelve (12) output measurable parameters which are mostly the thermodynamic parameters of the engine and can be affected by eight (8) different fault scenarios which are modelled by a decrease in the efficiency and the mass flow rate of different components of the jet engine.

For each developed approach, twelve (12) different networks were trained individually to learn the dynamics of all the twelve (12) engine parameters. Each network was used as a residual generator and the minimum detectable faults with that network were presented. An enhanced fault detection scheme was developed by using the top three performing networks that have the best capability in detecting the faults.

It was observed that by having a multi-network structure as residual generators the performance of the fault diagnosis scheme can significantly be improved.

The fault detection capabilities of the three structures were compared. The DNM approach has a robust structure in the training and the recall phases while the TDNN method is highly sensitive to the training and the network parameters. The DNM is more reliable and applicable to practical cases where an incremental updating is required and it can be practically used in an on-board fault diagnosis system. The TDNN network that is used in this thesis has two different structures in the training and the recall phases and cannot be used directly in an on-board fault diagnosis system. The DNM has a simpler feed-forward structure (SISO) without a need for a feedback from the outputs while in the TDNN a multi-input single-output (MISO) structure is required to learn the dynamics of the jet engine.

In contrast, the DNM scheme has a number of parameters to be updated (weight filters, feed forward and feedback coefficients, activation function slopes) and this results in more complicated and time consuming training procedure. Hence, a large number of initial parameters should be tested in order to achieve a good training. On the other hand, in the TDNN approach the only parameters that need to be adjusted are the weights of the neurons. Even in the ATDNN scheme the updating parameters are increased to only the weights and the delays of each layer, which results in faster training times.

The fault isolation task in this thesis was carried out separately by DNM and SISO TDNN network. A neural network classifier was used to evaluate the residuals that are generated from the trained networks. Our fault isolation scheme has the capability of isolating any set of three faults that can occur in the aircraft engine. It was observed that the performance of the fault isolation is dependent on the structure of the faults that occur together and for both of the isolation approaches the combination

of the faults occurring together play and important role in the isolation performance. For example, simulation results show that both isolation approaches can perform quite well on isolating the set of three low severity level faults. This shows that specific combination of the faults has better isolation capabilities or in other words, it has distinguishable effects on different engine parameters. Furthermore, from the simulation results one can conclude that in general the faults with different severity levels have better isolation capabilities. However, the nature of faults and types of the components where a fault has occurred have an important role in the overall fault isolation performance

Moreover, by comparing the fault isolation results for both the DNM and the SISO TDNN approaches, one can conclude that the TDNN based fault isolation has better performance and higher accuracy. Therefore, in order to achieve a fault isolation scheme with high performance this type of dynamic neural network might have better capabilities and reliabilities for the health monitoring of aircraft jet engines.

5.1 Future Work

Unlike static neural networks, dynamic neural networks generally have more parameters that need to be updated. In our proposed DNM model there are four sets of parameters which need to be updated, namely the feedforward coefficients and the feedforward filter coefficients, as well as the activation function slope parameters and the weights of the neurons. To obtain a well-trained neural network all these parameters should be optimized collectively. Due to high dimensionality of the updating parameter vector corresponding to the DNM one might not be able to optimize all the parameters efficiently. Hence, a neural network with more optimized parameters that can have better detection capability might still be achievable by optimizing all the parameters. The parameter optimization can be carried out in future through several

optimization methods such as simultaneous perturbation stochastic approximation (SPSA) [9].

The delays associated with the layers of the neurons in the TDNN are assumed to be fixed. One can analyze the performance of an adaptive time delay neural network (ATDNN) where the delays themselves are being updated along with the weights of the network.

From the simulation results and practical observations we have noted that a SISO TDNN can have a better performance as compared to the two other methods. Further work should be carried out to explain this performance theoretically. There are several explanations for these results that should be further examined, such as operating under a low frequency input signal, and the fact that the jet engine may be assumed to operate as an approximately linear system.

All the fault diagnosis schemes presented in this thesis have the potential capability to detect and isolate the faults in both steady state and transient modes of the aircraft engine. In this thesis, we only considered the cases where faults occur in the steady state mode of the engine while the proposed fault diagnosis schemes can be extended in future to operate in both steady state and transient modes. This capability would be an interesting subject for industrial aerospace companies since faults can occur in any arbitrary mode that the engine is operating. This results in a more secure, reliable and applicable fault diagnosis system where the fault diagnosis module can be used in a practical application.

The fault influence matrix was presented as a reference for the influence of each fault on all the engine parameters. We have considered the effects of 2%, 5% and 10% faults on the engine parameters. Further work can be carried out on the influence matrix by considering the effects of all severity level of faults on the corresponding percentage of changes they cause to the engine parameters. Considering all fault

severity levels, one can obtain a more precise fault influence matrix which would have a higher advantage for having *a priori* knowledge on the faults and their expected effects on the engine parameters even before running a simulation model.

Bibliography

[1] X. Dai, T. Breikin, Z. Gao, and H. Wang, "Dynamic modelling and robust fault detection of a gas turbine engine," in *American Control Conference*, June 2008, pp. 2160 –2165.

[2] R. Patton and J. Chen, "Detection of faulty sensors in aero jet engine systems using robust model-based methods," in *Condition Monitoring for Fault Diagnosis, IEE Colloquium on*, October 1991, pp. 211 –222.

[3] R. Patton, F. Uppal, and C. Lopez-Toribio, "Soft computing approaches to fault diagnosis for dynamic systems: A survey," *Control and Intelligent Systems*, vol. 7, no. 2-3, pp. 198–211, 2000.

[4] P. M. Frank, "Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy: A survey and some new results," *Automatica*, vol. 26, no. 3, pp. 459 – 474, 1990.

[5] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, and K. Yin, "A review of process fault detection and diagnosis: Part iii: Process history based methods," *Computers amp; Chemical Engineering*, vol. 27, no. 3, pp. 327 – 346, 2003.

[6] S. Simani, R. Patton, and C. Fantuzzi, *Model-Based Fault Diagnosis in Dynamic Systems Using Identification Techniques.* Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2003.

[7] K. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *Neural Networks, IEEE Transactions on*, vol. 1, no. 1, pp. 4 –27, March 1990.

[8] S. Leonhardt and M. Ayoubi, "Methods of fault diagnosis," *Control Engineering Practice*, vol. 5, no. 5, pp. 683 – 692, 1997.

[9] 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, 2005.

[10] I. A.-D. Al-Zyoud, "Neural network-based actuator fault detection and isolation for the attitude control subsystem of a satelite," Master's thesis, Concordia University, 2005.

[11] L. Li, L. Ma, and K. Khorasani, "A dynamic recurrent neural network fault diagnosis and isolation architecture for satellite's actuator/thruster failures," in *Advances in Neural Networks*. Springer Berlin / Heidelberg, 2005, vol. 3498, pp. 981–981.

[12] T. Kobayashi and D. L. Simon, "Application of a bank of kalman filters for aircraft engine fault diagnostics," *Volume 1 Turbo Expo 2003*, vol. 127, no. August, pp. 461–470, 2003.

[13] N. Meskin, E. Naderi, and K. Khorasani, "Fault diagnosis of jet engines by using a multiple model-based approach," *ASME Conference Proceedings*, vol. 2010, no. 43987, pp. 319–329, 2010.

[14] Y. G. Li and P. Nilkitsaranont, "A gas path diagnostic and prognostic approach for gas turbine applications," *ASME Conference Proceedings*, vol. 2007, no. 4790X, pp. 573–584, 2007.

[15] A. Volponi and B. Wood, *System Health Management.* John Wiley and Sons, Ltd, 2001.

[16] M. Barweli, "Compass-ground based engine monitoring program for general application," *SAE Technical Paper 871734*, vol. 10, 1987.

[17] A. J. 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," *Journal of Engineering for Gas Turbines and Power*, vol. 125, no. 4, pp. 917–924, 2003.

[18] D. L. Doel, "Temper—a gas-path analysis tool for commercial jet engines," *Journal of Engineering for Gas Turbines and Power*, vol. 116, no. 1, pp. 82–89, 1994.

[19] R. Ganguli, R. Verma, and N. Roy, "Soft computing application for gas path fault isolation," *ASME Conference Proceedings*, vol. 2004, no. 41677, pp. 499–508, 2004.

[20] S. R. Zedda M, "Fault diagnosis of a turbofan engine using neural-networks: a quantitative approach," *In: 34th AIAA,ASME,SAE,ASEE Joint Propulsion Conference*, 1998.

[21] C. Romessis and K. Mathioudakis, "Bayesian network approach for gas path fault diagnosis," *Journal of Engineering for Gas Turbines and Power*, vol. 128, no. 1, pp. 64–72, 2006.

[22] W. Yan and F. Xue, "Jet engine gas path fault diagnosis using dynamic fusion of multiple classifiers," in *Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, June 2008, pp. 1585 –1591.

[23] L. A. Urban, "Gas path analysis applied to turbine engine condition monitoring," *J. Aircr.*, vol. 10, no. 7, pp. 400–406, 1973.

[24] R. Patton and J. Chen, "Observer-based fault detection and isolation: Robustness and applications," *Control Engineering Practice*, vol. 5, no. 5, pp. 671 – 682, 1997.

[25] L. Trave-Massuyes and R. Milne, "Gas-turbine condition monitoring using qualitative model-based diagnosis," *IEEE Expert*, vol. 12, no. 3, pp. 22 –31, June 1997.

[26] S. Simani and C. Fantuzzi, "Fault diagnosis in power plant using neural networks," *Information Sciences*, vol. 127, no. 3-4, pp. 125 – 136, 2000.

[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] G. Torella, G — Lombardo, "Utilization of neural networks for gas turbine engines," *International Symposium on Air Breathing Engines, 12th, Melbourne, Australia*, 1995.

[29] S. Simani, C. Fantuzzi, and R. Spina, "Application of a neural network in gas turbine control sensor fault detection," in *Control Applications, 1998. Proceedings of the 1998 IEEE International Conference on*, vol. 1, September 1998, pp. 182 –186 vol.1.

[30] R. Joly, S. Ogaji, R. Singh, and S. Probert, "Gas-turbine diagnostics using artificial neural-networks for a high bypass ratio military turbofan engine," *Elsevier Science*, 2005.

[31] J. Addison, S. Wermter, and J. MacIntyre, "Effectiveness of feature extraction in neural network architectures for novelty detection," in *Artificial Neural Networks, 1999. ICANN 99. Ninth International Conference on (Conf. Publ. No. 470)*, vol. 2, 1999, pp. 976 –981 vol.2.

[32] D.-H. Seo, T.-S. Roh, and D.-W. Choi, "Defect diagnostics of gas turbine engine using hybrid svm-ann with module system in off-design condition," *Journal of Mechanical Science and Technology*, vol. 23, pp. 677–685, 2009, 10.1007/s12206-008-1120-3.

[33] S. O. T. Ogaji, "Advanced gas-path fault diagnostics for stationary gas-turbines," *PhD Thesis, School of Engineering, Cranfield University*, 2003.

[34] A. J. 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," *Journal of Engineering for Gas Turbines and Power*, vol. 125, no. 4, pp. 917–924, 2003.

[35] R. Mohammadi, S. Hashtrudi-Zad, and K. Khorasani, "Hybrid fault diagnosis: Application to a gas turbine engine," *ASME Conference Proceedings*, vol. 2009, no. 48821, pp. 719–729, 2009.

[36] S. Ogaji and R. Singh, "Artificial neural networks in fault diagnosis: A gas turbine scenario," in *Computational Intelligence in Fault Diagnosis*, ser. Advanced Information and Knowledge Processing, V. Palade, L. Jain, and C. D. Bocaniala, Eds. Springer London, 2006, pp. 179–207.

[37] L. Marinai, D. Probert, and R. Singh, "Prospects for aero gas-turbine diagnostics: a review," *Applied Energy*, vol. 79, no. 1, pp. 109 – 126, 2004.

[38] P.-J. Lu, M.-C. Zhang, T.-C. Hsu, and J. Zhang, "An evaluation of engine faults diagnostics using artificial neural networks," *Journal of Engineering for Gas Turbines and Power*, vol. 123, no. 2, pp. 340–346, 2001.

[39] A. J. 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," *Journal of Engineering for Gas Turbines and Power*, vol. 125, no. 4, pp. 917–924, 2003.

[40] Y. K. Lee, D. N. Mavris, V. V. Volovoi, M. Yuan, and T. Fisher, "A fault diagnosis method for industrial gas turbines using bayesian data analysis," *Journal of Engineering for Gas Turbines and Power*, vol. 132, no. 4, p. 041602, 2010.

[41] M. Ayoubi, "Fault diagnosis with dynamic neural structure and application to a turbo charger," *IFAC fault detection. Supervision and Safety for technicla Processes,Espo,Filand*, 1994.

[42] M. Ayoubi, M. Schäfer, and S. Sinsel, "Dynamic neural units for nonlinear dynamic systems identification," in *Proceedings of the International Workshop on Artificial Neural Networks: From Natural to Artificial Neural Computation*, ser. IWANN '96, 1995, pp. 1045–1051.

[43] R. Mohammadi, E. Naderi, K. Khorasani, and S. Hashtrudi-Zad, "Fault diagnosis of gas turbine engines by using dynamic neural networks," *ASME Conference Proceedings*, vol. 2010, no. 43987, pp. 365–376, 2010.

[44] A. Valdes, K. Khorasani, and L. Ma, "Dynamic neural network-based fault detection and isolation for thrusters in formation flying of satellites," in *Advances in Neural Networks*, ser. Lecture Notes in Computer Science, W. Yu, H. He, and N. Zhang, Eds. Springer Berlin / Heidelberg, 2009, vol. 5553, pp. 780–793.

[45] I.-D. Al-Zyoud and K. Khorasani, "Neural network-based actuator fault diagnosis for attitude control subsystem of a satellite," in *Automation Congress, 2006. WAC '06. World*, July 2006, pp. 1 –6.

[46] ——, "Detection of actuator faults using a dynamic neural network for the attitude control subsystem of a satellite," in *Neural Networks, 2005. IJCNN '05. Proceedings. 2005 IEEE International Joint Conference on*, vol. 3, July 2005, pp. 1746 – 1751 vol. 3.

[47] I. Al-Dein Al-Zyoud and K. Khorasani, "Neural network-based actuator fault diagnosis for attitude control subsystem of an unmanned space vehicle," in *Neural Networks, 2006. IJCNN '06. International Joint Conference on*, 0-0 2006, pp. 3686 –3693.

[48] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang, "Phoneme recognition using time-delay neural networks," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 37, no. 3, pp. 328 –339, March 1989.

[49] A. Yazdizadeh and K. Khorasani, "Adaptive time delay neural network structures for nonlinear system identification," *Neurocomputing*, vol. 47, no. 1-4, pp. 207 – 240, 2002.

[50] ——, "Identification of a class of nonlinear systems using dynamic neural network structures," in *Neural Networks,1997., International Conference on*, vol. 1, June 1997, pp. 194 –198.

[51] H. Talebi, K. Khorasani, and S. Tafazoli, "A recurrent neural-network-based sensor and actuator fault detection and isolation for nonlinear systems with application to the satellite's attitude control subsystem," *Neural Networks, IEEE Transactions on*, vol. 20, no. 1, pp. 45 –60, Jan 2009.

[52] Z. Li, L. Ma, and K. Khorasani, "A dynamic neural network-based reaction wheel fault diagnosis for satellites," in *Neural Networks, 2006. IJCNN '06. International Joint Conference on*, 0-0 2006, pp. 3714 –3721.

[53] L. Li, L. Ma, and K. Khorasani, "A dynamic recurrent neural network fault diagnosis and isolation architecture for satellite's actuator/thruster failures," in *Advances in Neural Networks*, ser. Lecture Notes in Computer Science, J. Wang, X.-F. Liao, and Z. Yi, Eds. Springer Berlin / Heidelberg, 2005, vol. 3498, pp. 981–981.

[54] M. Gupta, "Dynamical neural units with applications to the control of unknown nonlinear systems," *journal of intelligent and fuzzy systems*, vol. 3, no. 1, p. 73, 1994.

[55] K. P. J. Korbicz and A. Obuchowicz, "Dynamic neural networks for process modelling in fault detection and isolation systems," *International Journal of Applied Mathematics and Computer Sciences*, vol. 9, no. 3, pp. 519–546, 1999.

[56] A. Yazdizadeh, K. Khorasani, and R. Patel, "Identification of a two-link flexible manipulator using adaptive time delay neural networks," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 30, no. 1, pp. 165 –172, February 2000.

[57] C. Wohler and J. Anlauf, "An adaptable time-delay neural-network algorithm for image sequence analysis," *Neural Networks, IEEE Transactions on*, vol. 10, no. 6, pp. 1531 –1536, November 1999.

[58] D.-T. Lin, J. E. Dayhoff, and P. A. Ligomenides, "Trajectory production with the adaptive time-delay neural network," *Neural Networks*, vol. 8, no. 3, pp. 447 – 461, 1995.

[59] S. Satoh, F. Yakuwa, and Y. Dote, "Combination of radial basis function (rbf) and time delayed neural networks (tdnn) for fault diagnosis of automobile transmission gears using general parameter learning and adaptation," in *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, vol. 2, October 2003, pp. 1457 – 1462 vol.2.

[60] S. Barai and P. Pandey, "Time-delay neural networks in damage detection of railway bridges," *Advances in Engineering Software*, vol. 28, no. 1, pp. 1 – 10, 1997.

[61] D. Rao and M. Gupta, "Dynamic neural unit and function approximation," in *Neural Networks, 1993., IEEE International Conference on*, 1993, pp. 743 –748 vol.2.

[62] J. Mor, "The levenberg-marquardt algorithm: Implementation and theory," in *Numerical Analysis*, ser. Lecture Notes in Mathematics, G. Watson, Ed. Springer Berlin / Heidelberg, 1978, vol. 630, pp. 105–116.

[63] D. F. Shanno, "Conditioning of quasi-newton methods for function minimization," *Mathematics of Computation*, vol. 24, no. 111, pp. pp. 647–656, 1970.

[64] R. Mohammadi, E. Naderi, and K. Khorasani, "Fault diagnosis of gas turbine engines by using dynamic neural networks," *Proceedings of ASME Turbo Expo 2010: Power for Land, Sea and Air GT2010,Glasgow, UK*, 2010.

[65] http://en.wikipedia.org/wiki/Turbofan.

[66] W. Visser and M. Broomhead, "Gsp, a generic object-oriented gas turbine simulation environment," *Journal of engineering for gas turbines and power*, vol. 128, no. 43987, p. 13, 2000.

[67] P. C. Ten-Huei Guo, "Sensor based engine life calculation a probabilistic perspective," *16th International Symposium on Airbreathing Engines*, vol. 3, no. 3, pp. 1141–1146, , 2003.