DEGRADATION PROGNOSTICS IN GAS TURBINE ENGINES USING NEURAL NETWORKS

Ameneh Vatani

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 2013

Concordia University
School of Graduate Studies

This is to certify that the thesis prepared

By: **Ameneh Vatani**

Entitled: **Degradation Prognostics in Gas Turbine 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:

Dr. Kabir _____ Chair

Dr. Sheldon Williamson _____ Examiner

Dr. Ali Dolatabadi _____ Examiner, External

Dr. Khashayar Khorasani _____ Supervisor

Approved by Dr. William E. Lynch _____
Chair of Department of Electrical and Computer Engineering

Christopher W. Trueman, Interim Dean
Faculty of Engineering and Computer Science

Abstract

Degradation Prognostics in Gas Turbine Engines Using Neural Networks

Ameneh Vatani

In complex systems such as aircraft engines, system reliability and adequate monitoring is of high priority. The performance of all physical systems degrades over time due to aging, the working and environmental conditions. Considering both time and safety, it is important to predict the system health condition in future in order to be able to assign a suitable maintenance policy. Towards this end, two artificial intelligence based methodologies are proposed and investigated in this thesis. The main objective is to predict degradation trends by studying their effects on the engine measurable parameters such as the temperature and pressure at critical points of a gas turbine engine.

The first proposed prognostic scheme for the gas turbine engine is based on a recurrent neural networks (RNN) architecture. This closed-loop architecture enables the network to learn the increasing degradation dynamics using the collected data set. Training the neural networks and determining the suitable number of network parameters are challenging tasks. The other challenge associated with the prognostic problem is the uncertainty management. This is inherent in such schemes due to measurement noise and the fact that one is trying to project forward in time. To overcome this problem, upper and lower prediction bounds are defined and obtained in this thesis. The two bounds constitute a prediction band which helps one not to merely depend on the single point prediction. The prediction bands along with the prediction error statistical measures, allow one to decide on the goodness of the prediction results.

The second prognostic scheme is based on a nonlinear autoregressive with exogenous input (NARX) neural networks architecture. This recurrent dynamical structure takes advantage of both features which makes it easy to manage the main objective.

The network is trained with fewer examples and the prediction errors are lower as compared to the first architecture. The statistical error measures and the prediction bands are obtained for this architecture as well.

In order to evaluate and compare the prediction results from the two proposed neural networks a metric known as the normalized Akaike information criterion (NAIC) is applied in this thesis. This metric takes into account the prediction error, the number of parameters used in the neural networks architecture and the number of samples in the test data set. A smaller NAIC value shows a better, more accurate and more effective prediction result. The NAIC values are found for each case and the networks are compared at the end of the thesis.

Neural networks performance is based on the suitability of the data they are provided with. Two main causes of engine degradation are modelled in this thesis and a SIMULINK model is developed. Various scenarios and case studies are presented to illustrate and demonstrate the effectiveness of our proposed neural networks based prognostic approaches. The prognostic results can be employed for the engine health management purposes. This is a growing and an active area of research for the aircraft engines where only a few references exist in the literature.

To my beloved parents for their Love, Care and tireless support;
for teaching me the meaning of life.

Acknowledgments

I wish to thank, first and foremost my supervisor Dr. Khorasani whose sincerity and encouragement has guided me through my studies. Completion of my masters thesis would have not been possible without his patience, motivation, insightful and professional comments. This was an honour for me to work with him.

Besides my supervisor, I would like to thank the rest of my thesis committee Dr. Dolatabadi, Dr. Kabir and Dr. Williamson for their insightful comments, and evaluating my work.

I would also like to thank and appreciate the help of all my friends and colleagues who have always shared their experience with me and made my work environment friendly and productive.

My deepest gratitude goes to my loving parents. I have been blessed with the best parents who have always supported me in life and especially in this stage of my life. Their success, patience and positive attitudes towards life have always given me confidence and courage to step toward my goals with hope. Words cannot describe how fortunate I feel to have such role models in my life. My success id theirs and their happiness is mine.

Contents

List of Figures

List of Tables

List of Abbreviations and Symbols

FDI	Fault detection and identification
CBM	Condition-based monitoring
DPHM	Diagnosis, Prognosis and health management
RUL	Remaining useful life
PCA	Principal component analysis
PDF	Probability density function
ARIMA	Autoregressive integrated moving average
AI	Artificial intelligence
GA	Genetic algorithm
NN	Neural networks
MLP	Multi layer perceptron
FFNN	Feed forward neural networks

RNN	Recurrent neural networks
NARX	Nonlinear autoregressive with exogenous inputs
Φ	Nonlinear mapping of the neural network
RTRL	Real-time recurrent learning
BPTT	Back-propagation through time
GPA	Gas path analysis
C	Compressor
CC	Combustion chamber
crit	Critical
d	Intake
f	Fuel
m_f	fuel flow
i	Stage input
mech	Mechanical
n	Nozzle
n_i	Nozzle input
n_o	Nozzle output

T Turbine

TT Turbine temperature

β Bypass ratio

$\dot{m}$ Mass flow rate, $\frac{Kg}{s}$

γ Heat capacity ratio

η Efficiency

π Pressure ratio

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 Rotational speed of spool connecting the compressor to the 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
GSP	Gas turbine simulation program
FI	Fouling index
EI	Erosion index
μ_{ae}	Mean of the prediction error
σ_{ae}	Standard deviation of the prediction error (std)
$rmse$	Root mean square of the prediction error
BIC	Bayesian information criterion
NAIC	Normalized Akaike information criterion

Chapter 1

Introduction

In complex systems such as aircraft engines reliability is a necessity, therefore one is expected to predict the future behaviour of the system to assign a suitable maintenance policy. The important goal of the prediction of the system operation can be achieved through two methodologies. The first is the physics-based approach and the second is the data-driven approach. In the former approach an extensive knowledge about the entire system, various variables, exact functions of the components and their interactions together under different operating conditions are essential for a detailed calculation of the temperature, stresses, degradation modes, etc. There may still remain many unknown and neglected factors in the representation and modelling of the jet engine nonlinear system. This fact can result in unexpected catastrophic damages or costly unnecessary maintenance (system shut-down or component replacement). On the other hand one may rely on the data obtained from the engine sensors that are installed in the system.

In this research data-driven methods are preferred since the data collected from a real engine are representative of all the existing characteristics and factors. Among various data-based methods, predicting the trends in this thesis will be done by using neural networks (NN) as they are able to capture the nonlinear behaviour of a

complex system with a limited number of data points. Aircraft engines are subjected to aging process and different degradations which will increase through time. Hence, our objective is to predict degradation trends by studying their effects on the engine measurable parameters such as the temperature and the pressure at the critical points of a gas turbine engine. The results can be employed for condition-based maintenance purposes. In this chapter we will define and discuss all the above mentioned terms and shed light on the importance of prognostics and condition-based maintenance for gas turbine engines. We will also review some research conducted in these areas in the literature.

1.1 Literature review

It is a general fact that every physical component degrades over time as it may work under different levels of stress, load, temperature and varying environmental conditions. Even the products with the best design are prone to deteriorations along the time they are in service. System reliability is an important factor in all industry sectors thus it is very important to monitor components' condition and perform the necessary maintenance actions.

Maintenance actions can be divided into two groups namely conventional maintenance techniques and condition-based maintenance techniques. The traditional maintenance action is itself divided into two methods. The first one is called breakdown maintenance [1]. It is also called unplanned maintenance or run-to-failure maintenance. In this type of traditional maintenance the component is replaced only at the time that it is failed. This is not a wise method as in many cases we are dealing with people's lives such as nuclear or aerial industries and the maintenance cannot be postponed until the components' breakdown.

The second method is known as planned maintenance. This method is based on

a proactive strategy. They are also known as proactive or scheduled maintenance [18, 19]. In this method, maintenance actions are being done periodically and on a regular basis based on some previous knowledge on the component performance deterioration and there is no feedback from the current health status of the system. In other words, they are time-based, regardless of the machine health. This method can be wasteful and costly as in most of the cases the maintenance action is not necessary. There exists a trade-off between being reliable and cost-effective. Recently machinery users are adopting to a method called condition-based monitoring (CBM).

In CBM the goal is to limit the number of maintenance actions and to avoid the unnecessary ones. The first step to CBM is condition monitoring. The data collected from health monitoring will be used and analyzed later for condition-based maintenance purposes. The CBM is based on the fact that the machine does not breakdown suddenly and it often goes through a continuous degradation process. When there is enough evidence of the abnormal performance in the system, maintenance is suggested. As a result there can be a great reduction in the maintenance and repair cost by avoiding unnecessary maintenance actions and systems' downtimes and still users do not have to be concerned with the major breakdowns.

Three major steps constitute the CBM process [20] as shown in Figure 1.1:

1. Data acquisition: In this step the information is collected using the sensors that are installed throughout the system. Our goal is to collect the data that are most related to the system health. For a mechanical system these data can be vibration signals, acoustic signals, temperature data, etc.

2. Data processing: The data that is collected in step one is now analyzed. This step is the most critical among the three. The reason is that a large amount of data is available from sensors but it is not clear how the data should be handled and how exactly they are related to the system's health. It should be noted that there

exist three main categories of data processing in the literature namely, time-domain analysis, frequency-domain analysis and time-frequency analysis [1].

3. Maintenance decision making: After the data is collected and analysed, the decision will be made whether the maintenance is suggested for the current condition or not.

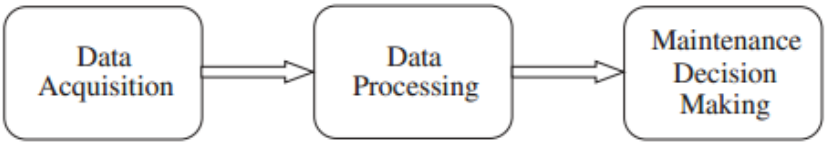

Figure 1.1: Three steps constituting the CBM system [1].

In a CBM scheme two important aspects are diagnostics and prognostics and all together they are called a diagnosis, prognosis and health management system (DPHM).

The functionality of the DPHM system mostly depends on the efficiency of the diagnostics and prognostics scheme. The diagnostics mechanism itself consists of three modules i.e. fault detection, isolation and identification (FDI scheme) [9]. These three modules are defined as follows [1, 4]. The task of fault detection is to distinguish if something abnormal is happening in the behaviour of the system under study. In the fault isolation one tries to locate the fault and declare in which part or component of the system the malfunction is occurring. Identification is the act of evaluating the nature and intensity of the fault. Figure 1.2 briefly depicts the FDI concept. The building blocks will be defined in the following sections.

On the other hand prognostics deals with prediction and prior event analysis. The incorporation of the prognostics module can enhance the diagnostics task. Prognostics can be regarded from two different perspectives [21]. It may either estimate the future

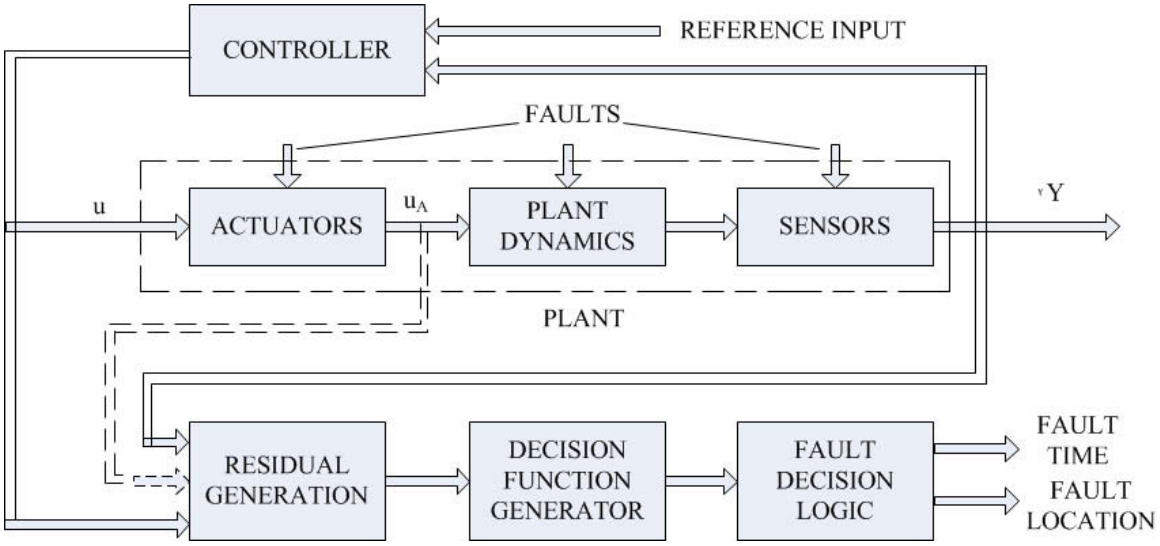

Figure 1.2: Conceptual structure of FDI [2].

health status of the system at a desired time or estimate the remaining useful life of a component or the system using the current and previous life usage data reported from the system under gradual degradations. Prognosis is a more challenging task because we are trying to predict the future based on some present-time evidence and hence our judgement is prone to errors and uncertainties and as a result, relative to the FDI, much less reference exists in the literature that have focused on the prognostics.

Many studies can be found in the literature on the subject of DPHM and CBM for the mechanical systems that have applied a variety of techniques and applications. In this literature review some recent diagnosis and prognosis schemes for physical systems will be briefly discussed.

Diagnostic methods can be divided into two categories, namely model-based and data-driven approaches [22, 23]. In order to use model-based or physics-based approaches, an explicit mathematical model of the system should be available. This needs a thorough understanding of the system, equations and relations between different variables which is not always possible. Different methods exist in the literature for fault detection, to name for example [24, 25]: fault detection with limit checking;

fault detection with process estimation methods; fault detection with parity equations; fault detection with state observers and state estimations; fault detection with principal component analysis (PCA) and finally a variety of combined methods.

To detect the fault using any of the above mentioned methods, the first step is the residual generation. Residual is often a fundamental component in a diagnosis system. Residuals are typically generated by using a mathematical model of the system and measurements from sensors and actuators. This process is referred to as residual generation. The residuals are generated by comparing system current state and the system healthy state. If the residual signal is higher than a pre-specified threshold, this is an indication of the fault presence in the system. The procedure is illustrated in Figure 1.3.

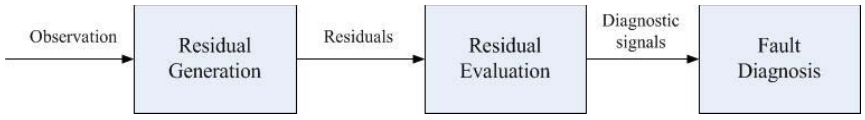

Figure 1.3: General flowchart of a model-based approach [1].

If the exact mathematical model of the system is available, the model-based approaches will give more precise results as compared to the non model-based methods, but the fact is that we usually are dealing with complex systems where having an explicit mathematical model of the system is very hard or almost impossible. In [26] the authors used the inherent analytical redundancy of multiple observers for sensor fault detection. In [27], model-based fault diagnosis of rolling element bearings is investigated using on-line vibration signals. Loparo *et al.* have developed a fault detection and isolation scheme based on a filtering approach for the same application by combining fault detection filters and sliding-mode detectors [28]. More information on different model-based techniques, residual generation and different application domains can be found in [4].

Data-driven approaches are categorized into statistical methods and artificial intelligence (AI) methods. In statistical approaches which are quantitative data-based methods [29] the data obtained from sensors and health monitoring approaches are used for hypothesis testing purposes [30]. Another statistical approach is known as cluster analysis which is a multivariate classification approach used for fault diagnosis and identification. More information on this method can be found in [31].

AI methods mostly depend on the experimental data obtained from the system under study. One can use vibration data, acoustic estimations, temperature or stress data, oil analysis, etc. and this fact makes the data-driven methods popular and able to be applied to a wide range of systems. The type of the data to be used is chosen according to the type of the system under study, available sensors and availability of the data. According to [29] the data-based or history-based methods can be also divided into qualitative and quantitative methods. One example of qualitative method is employing rule-based or fuzzy logic methods. In [32] the authors have given more insight on the use of expert systems for fault diagnosis. Li *et al.* [33] have applied the fuzzy Petri-net based methods for fault diagnosis of mechanical-electric equipment. In [34], fault diagnosis of the engine starting system is achieved by using fuzzy logic algorithm. Fuzzy observers are used in [35] for fault diagnosis of nonlinear dynamic systems.

Another popular data-driven method is the neural network (NN) method. Neural networks are powerful tools for classification and function approximation [36]. Neural networks are used in chemical engineering for process fault diagnosis [37, 38, 39]. Neural network is a quite popular method for diagnostics in the field of electrical engineering too. Reference [40] is a survey on fault diagnosis methods for electrical motors. Furthermore, Filippetti *et al.* [41] have developed an on-line scheme using neural networks for diagnostics of an induction motor rotor. In mechanical engineering

bearing diagnostics is achieved using feed-forward neural networks (FFNN) that are regarded as the most popular neural network architecture [1], for more information one can refer to [42, 43, 44]. Some other types of neural networks used for fault detection and diagnosis are cascade correlation neural networks [45], dynamical neural networks [46], recurrent neural networks [47] and wavelet neural network [48].

In some references neural networks and fuzzy logic are combined to build-up a neuro-fuzzy (NF) FDI scheme. In [49] this method is applied for the fault detection of an industrial gas turbine engine. Besides Loparo *et al.* in [28] utilized neuro-fuzzy models for fault detection of nonlinear dynamical systems. A drawback of data-driven methods is the limitations on the volume of the information that can be saved.

Despite the maturity of the research conducted in the field of fault detection, isolation and identification, failure prognostics is still in its early stages and it has drawn attention only recently, compared to the problem of fault diagnosis [50]. The definition of prognostics is given earlier in this chapter but it is apparent that yet, there is not a consistency in the literature for the definition of this term.

According to the ISO standards [51], prognostic is defined as "prediction of a system's lifetime". In other word, we are interested in finding the machine remaining useful life (RUL) using the current condition of that system or subsystem. On the other hand, in many applications we cannot wait until the failure happens in the system. The reason is that for example in nuclear power plants and air plane engines the failure can result in a catastrophic event which deals with peoples' lives. In these cases it would be more advantageous to study the fault propagation trend and find the probability or the time until the failure magnitude reaches a threshold beyond which the system must be taken off-line for maintenance. In this case, the definition of the failure will be important too.

Similar to the FDI, failure prognostic methods can be divided into two categories

namely model-based and data-driven based. Most of the research conducted on prognostics are application specific [18]. Data-driven methods can only account for the faults and conditions to which the system or component has been subjected to but the reason that non model-based approaches are more popular in this domain is that understanding of the exact failure mechanism can be infeasible or even impossible [21]. In the following, some research in the domain of prognostics will be reviewed. Figure 1.4 illustrates different prognostic approaches according to their cost, reliability and the range of applicability to different physical systems.

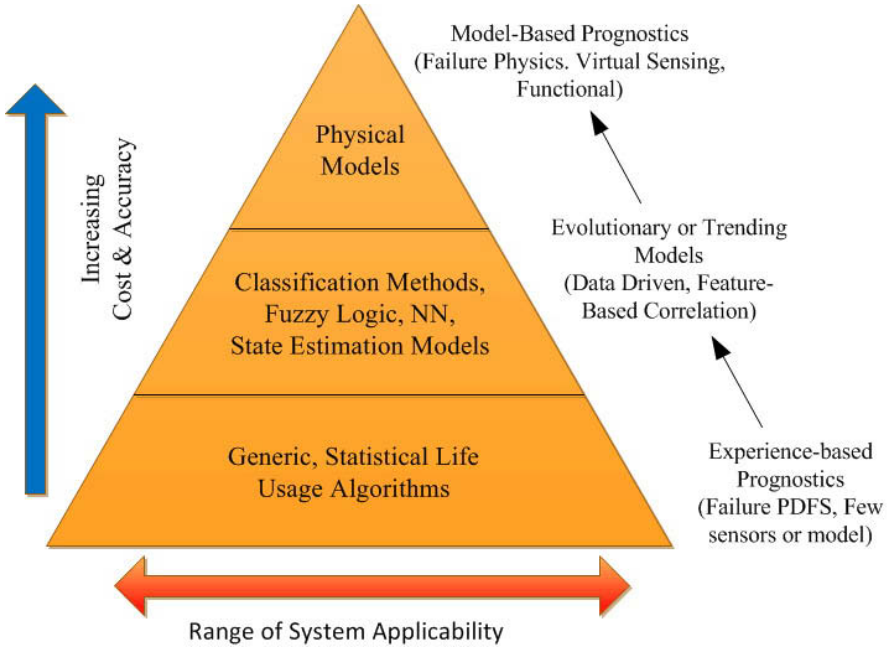

Figure 1.4: Different prognostic approaches [3].

In [52], the authors have developed an interacting multiple model-based approach and tested it on an automotive suspension system. They have achieved predicting the remaining useful life of the system through 5 steps namely, identify system model, simulation under random noise, prognostic modelling, feature estimation and finally tracking the degradation measure. Byington *et al.* [53] have applied a model-based approach to prognosis and health management of an electromechanical actuator

unit of flight control. In mechanical engineering, the initiation and propagation of fatigue as a structural damages is widely studied. In [54] the authors have applied Kalman filtering to corrosion-fatigue models. To model fatigue crack high mechanistic understanding is required. Ray *et al.* [55] have developed a stochastic damage model for the fatigue crack dynamics. They have also constructed a Kalman filtering which is able to estimate the current damage state and predict the remaining useful life of the structure. Time-dependent damage rate and damage propagation can be calculated using this stochastic model and the result is suitable to be used for an on-line decision making and real-time failure prognosis [55]. For more information on different research projects on modelling the damages in a mechanical system one can refer to [52, 56, 57, 58, 59, 60].

Similar to the FDI, data-driven approaches are divided into two general parts. The methods that rely on the expert systems and their knowledge and the methods that are data-driven. Hybrid prognostic schemes have also been developed which are combination of two methods and the goal is to use the advantages of each scheme toward a better and more accurate result. It must be noted again that prognostic is uncertain in nature and is a probabilistic method as we try to project the present information into the future [3]. In [61] fuzzy logic is used for avionic maintenance schedule planning and repair prognosis.

One domain that uses the available data is statistical based methods. In these methods the goal is to find the probability of failure using some statistic-based methods or by fitting some probabilistic model to the data from the real system working under varying conditions. Machine condition prognosis is performed in [62] which uses sequential Monte Carlo method for prognostics. This is also close to the work being done in reliability analysis. Groer has conducted time-to-failure analysis using the Weibull models [63]. Moreover, authors in [64] have presented an approach

to be served as an alternative to stochastic modelling approaches for estimating the time to failure of a machine. They fit a Weibull distribution to their data and have developed an algorithm that automatically gives the Weibull parameters suitable for each data set. Once the model is fitted, the probability density function (PDF) of the failure can be derived and utilized to find the probability of failure or equivalently the time to failure of the component under study. The component is recommended to be replaced or taken for repair when the PDF reaches a high value. This concept is summarized in the figure below (Figure 1.5). A description on application of probabilistic approaches in prognosis can be found in [60].

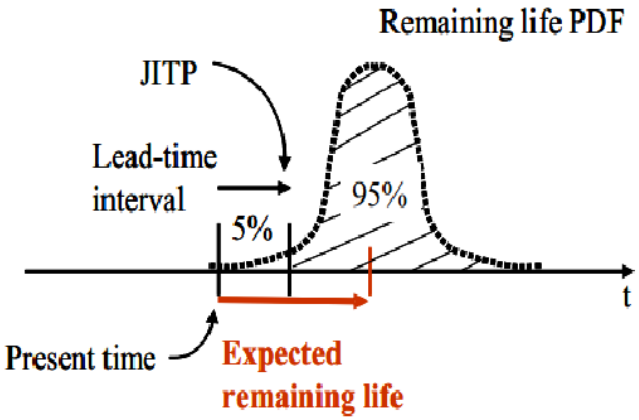

Figure 1.5: Probability density function (PDF) for prognosis [3].

One challenging fact that can improve the prediction is that the PDF is a conditional PDF, meaning that as more data becomes available *a posteriori* PDF will be computed. By taking advantage of this fact, the accuracy of the prediction can improve as time evolves, and more diagnosis data becomes available from the sensors that are installed in the system [3]. As time goes on and the component becomes closer to its failure the PDF becomes narrower with a smaller variance. The concept is depicted in Figure 1.6.

Phelps *et al.* [65] achieved promising results by finding sensor-level test-failure

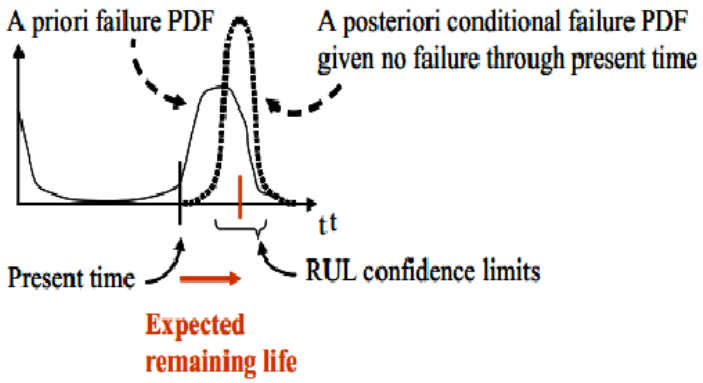

Figure 1.6: Updated PDF function for prognosis [3].

probability vectors . These probability vectors are being tracked using Kalman filters and bank of interacting multiple models associated with different fault types.

Data-driven methods depend mostly on process measurements. The challenging part in these methods is to decide which measurement to use. Furthermore, because the available data is from a real system, they contain noisy signals and first the data preprocessing has to be conducted before going through the analysis part. Besides, because of the uncertain nature of the prognostics, uncertainty management has to be considered as well, and finally based on the existing data the failure or damage propagation trends, have to be analyzed and the decision about the future condition of the system has to be made. The two popular methods in this domain are time-series analysis and neural networks (NN) analysis which is an artificial intelligence (AI) based method.

Basic theory on time-series modelling and prediction can be found in [66, 67, 68, 69, 70]. The accuracy of forecasting using this method is investigated in [71]. One popular approach is the autoregressive integrated moving average (ARIMA) method

which is applied to non-stationary observation data. In [72], an improved ARIMA-based approach is proposed for machine health status prediction via an automatic prediction algorithm. The authors have tested the effectiveness of the method by applying it to a rotor test rig which is composed of a rotor, stator, driving motor, bearings and couplings. The results are improved as compared to the basic ARIMA method and are very close to real data. Gang *et al.* [73] have proposed a novel time-series prediction methodology based on Dempster-Shafer regression and the iterative multi-step-ahead time series prediction method. A methane compressor is chosen as a test-bed to evaluate the performance of the method. The results satisfy the performance evaluation criteria [73]. Another application of time-series modelling in a chemical processes can be found in [74] where the faults are considered to be slow time-varying. The methodology is based on statistical methods and multivariate time series prediction. Denoising of the data is based on wavelet denoising technology and is applied to a continuous stirred tank reactor (CSTR).

Another very important data-driven approach to prognostics and degradation trend analysis is neural networks (NN). Here a review of the work done in this domain will be presented to give an insight into the state-of-art in research. As this method is used in this thesis, the background information will be presented in the next chapter with more details. Neural networks are nonlinear approximators [36]. Neural networks are self-adaptive and require little information about the physical model because they learn from examples. The NN methods have been developed for prediction purposes and they have become practical tools for prediction problems. The key point in this method is to be able to distinguish how the measured data is related to the engine degradation process and damage accumulation. Authors in [75] have developed a generic on-board prognostication scheme for helicopter gearbox system. Vibration data is used to train ta polynomial neural network.

Different neural network architectures and training algorithms have been proposed. In [76] Zhang *et al.* have performed multivariate trend analysis using vibration measurements and self-organizing neural networks. The method is applied to estimate the residual life of a bearing system. In [77] the rate of machine deterioration is predicted by utilizing a recurrent neural network (RNN). The method is tested in a power plant and the results are shown to be satisfactory to be used in pre-scheduling maintenance practices. Also in the field of electrical engineering backpropagation (BP) neural network prediction model is applied for condition prediction of power plant equipments [78]. Moreover, RNNs are applied for prediction of machine deterioration in [79]. Remaining bearing life is predicted in [80] using a feed forward neural network for one-step and multi-step ahead predictions.

In [80], Shao *et al.* have preformed progression-based prediction of the remaining life (PPRL). A feedforward NN with the backpropagation training algorithm is developed for bearing remaining life prediction. Neural networks are parallel computing devices that learn from examples and are fast. The results from this network were compared to those obtained from the ARIMA method and it shows that NNs outperform the traditional forecasting methods such as time-series prediction methods. Zemouri *et al.* [81] have developed a recurrent radial basis function neural network (RRBFNN) for the problem of dynamic monitoring and prognosis. Application of the network in two benchmarks gives good results.

In [82] degradation data from fatigue crack growth signals have been analyzed using a feed-forward NN (FFNN) structure and the estimate for the mean and the variance of the prediction have been derived. The authors have defined two prediction bounds to overcome the prediction uncertainty. Moreover, FFNNs have been used in [83] for diesel engine emission prediction, using the pressure data of an experimental setup. Neural networks are also capable of predicting nonlinear systems with

time-delay. The pressure trend of a continuous stirred tank reactor (STR) has been predicted in [84] by using NN and nonlinear *Smith* predictor.

It is worth mentioning that some researchers have concentrated on hybrid methods for prediction. They have either combined model-based and data-driven approaches or two non model-based approaches. Hansen *et al.* [85] have proposed a method for machinery prognostics which combines model-based methods with data-driven methods. The method is tested for operational life of a gear predictions and the authors have given some suggestions on how the method can be extended and applied to other rotary machinery such as bearings. In [86] a neuro-fuzzy prediction scheme is developed which can accurately and quickly learn the system dynamics. The method is applied to gear pitting damage and shaft misalignment data and is proven to be robust and reliable. Another neuro-fuzzy based approach is introduced in [87]. Domain expert with high experimental knowledge of the system is essential in this method and it is most useful when there is not enough information about the degradation signals or failure modes of the system. This method has been used for online health monitoring of a cutting tool. Another hybrid method for fault prognostics with application to chemical process can be found in [88].

Above, a brief review on the work conducted in the domain of diagnosis, prognosis and health monitoring (DPHM) for mechanical, electrical and chemical systems (physical systems) is given. As the focus of this thesis is on the aircraft gas turbine engines, in the following some of the work done in FDI and prognostics of gas turbine engines will be presented.

1.1.1 Gas Turbine Engine Diagnostics

Having a reliable fault detection and failure prognosis scheme is more critical and important when it comes to aircraft gas turbine engines. The reason is that this

system can be very costly if unnecessary maintenance actions are done as the jet engine components are expensive. On the other hand if the maintenance actions are not based on condition monitoring and are not done in time, there is a chance of catastrophic events such as plane crashes. For these reasons the condition-based maintenance is a highly active research subject in the academia and industry. The classification of methods is the same as the other mechanical systems which were mentioned earlier.

First we will investigate some research in the field of fault detection, isolation and identification related to gas turbine engines. There are many information sources in a gas turbine engine [89]:

Engine gas path measurements

Oil-Fuel system measurements

Vibration measurements

On-board engine models

Companion engine data

Engines gas path measurements are the data obtained from the engine's gas path such as inner-stage temperatures, pressures, fuel, spool speed, etc. The changes in these data can be studied and used as a measure to judge if the engine is working away from its desired and healthy state. The method which relies on this type of data is called gas path analysis (GPA) method. More information on GPA analysis can be found in [90, 91]. The concept is illustrated in Figure 1.7. The key point in FDI methods is defining the proper residual signals and when the signals amplitude are greater than a predefined threshold, the malfunction is detected in the system. Different methods exist in the literature for threshold determination such as adaptive thresholding [92], [93]. The type of residual signal depends on the type of measurement available.

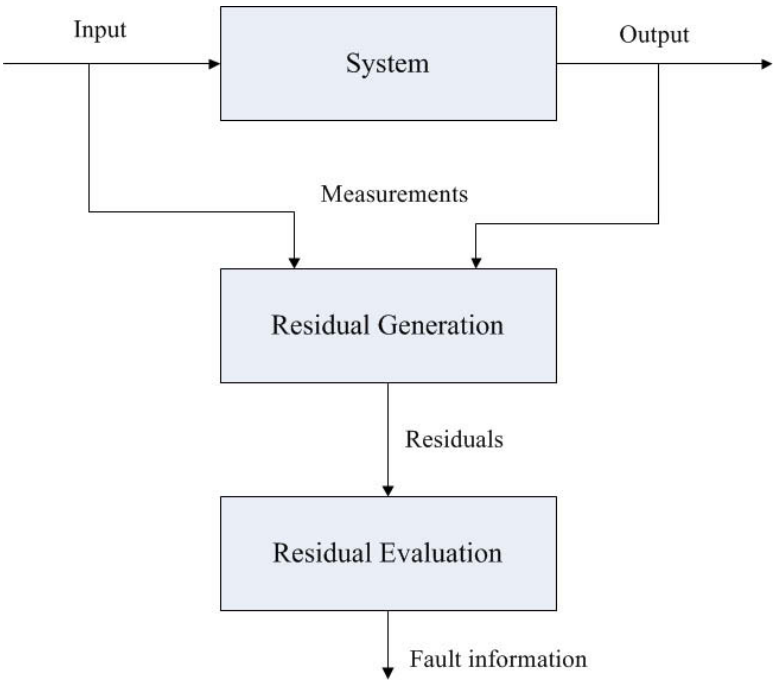

Figure 1.7: Schematic of an FDI approach [4].

Model-based approaches are one of the popular diagnosis methods for gas turbine engines. In model-based approaches one tries to mathematically model the components and then investigate how the measurements change when an anomaly exists in the jet engine. In [94] a Kalman filtering methodology is introduced for gas turbine engine fault detection and isolation. Kalman filter is a linear model-based estimator and the method has proven to be very accurate.The gas path data taken along the engine gas path are utilized here. Dewallef *et al.* have combined Kalman filters with classification techniques for engine diagnostic purposes [95]. The goal is to recursively estimate a set of health parameters associated with an engine component. Bayesian belief network (BBN) is used as the classification method and then it is combined with Kalman filters. The method is tested on a "twin-spool mixed-flow turbofan" which is a civilian aircraft. Different fault classes are then defined for different components such as high pressure compressor (HPC) or low pressure turbine (LPT). Finally the identification results for different faults are presented which shows the avoidance of false alarms and less number of undetected faults.

A bank of Kalman filters is integrated with fault isolation logic for the FDI of

sensor and actuator faults which demonstrates the popularity of Kalman filters in fault detection and isolation [96]. The method appears to be robust as the measurement noise has also been considered. The detectable faults are physical faults that can occur in the system due to different reasons such as worn seals, foreign object damage, blade erosion or corrosion and as a result they may change different engine gas path variables. This discussion is summerized in Figure 1.8.

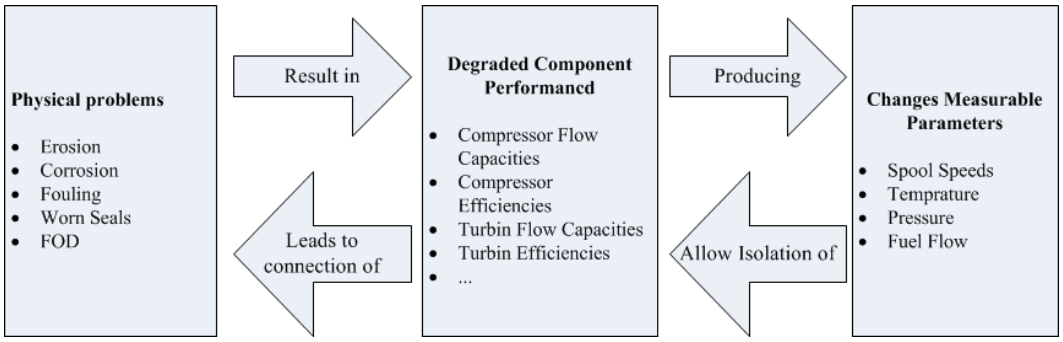

Figure 1.8: GPA principle [5].

In spite of the previous research where Kalman filters were used as linear estimator, in [9, 97] the authors have applied the extended Kalman filter concept (EKF) for nonlinear fault diagnosis of the jet engine. The developed multiple model-based approach enables single as well as concurrent fault detection and isolation in the engine. Using the maximum probability criteria the engines operating mode is determined. Also, authors in [98] have applied the method for the case of in-flight sensor fault detection.

In an aircraft engine it is vital to detect the system deviations from its nominal behaviour as early as possible [99]. Anomalous operation of the engine can affect undesirably both the engine function and the aircraft mission management. Generally speaking, if the mathematical model of the system in addition to different fault models and their progression are available, model-based approaches are more accurate and effective. It is also possible to implement them on board. But often, it is very hard

or complicated to find the explicit mathematical model as the system complexity and uncertainties increase. As a result the accuracy of the results will decrease. This problem is especially highlighted for the case of gas turbine engines, where the complexity of the system makes it challenging to have an accurate mathematical model. In this case data-based techniques have proven to be useful as the measured data have captured system's relation as well as the uncertainties [100], random error, and unbiased sensor error that are present in the system. The results from these methods can only account for the modes through which the system is operating, but they are very practical methods for complex systems. Yet, there does not exist a method that can outperform others in every aspect and there is a trade-off between generality and accuracy of the method. In the following, some data-driven work for the FDI of gas turbine engines will be presented.

The existing data-driven methods for the engine FDI are categorized into statistical approaches and artificial intelligence (AI) approaches [101]. In [102] an off-line fault diagnosis method is proposed for gas turbine engines working in the steady-state mode. The authors have realized a multiple Bayesian model in one hierarchical model for detection and identification of different fault scenarios. Single and concurrent faults are successfully detected in their scheme. Sensor bias and its magnitude can be estimated as well. The case study is a single shaft gas turbine engine with a faulty compressor and an unbiased fuel flow sensor. Four health parameters are estimated from six measurements. More information on Bayesian network approach in fault diagnosis can be found in [103, 104, 105].

In AI methods, fuzzy-logic methods and NN methods are the most popular ones [106]. In [107] a fuzzy logic system is developed for gas turbine engine fault isolation. The model is 95% accurate but it is still valid if the uncertainty level increases with the help of some additional data. Fault isolation in gas turbine engines is achieved

in [108] using fuzzy classification methods. More information on this subject can be found in [109, 110, 111, 112].

Neural networks have been frequently used in the FDI domain for a complex system such as gas turbine engine for the reasons pointed out earlier. In [113] radial basis function neural networks (RBFNN) as a class of feed-forward neural networks have been proposed as a generic method for fault detection. From the output of the network it can be determined if the system is operating under nominal condition or anomalous condition. The method is applied to advanced military aircraft subsystems. In [114] the authors have diagnosed a fault in gas turbine engines in a fleet of aircraft using neural networks. Campa *et al.* [115] have designed a sensor fault detection, isolation and accommodation (SFDIA) scheme for diesel engines. They have employed neural networks and thus the physical redundancy in the sensors in not needed.

In [116] neural networks are applied for fault diagnosis of a turbo fan engine working in the steady state condition. According to the authors there are many advantages in applying this method, namely only small number of measurements are required and the scheme is capable of detecting multiple faults. In addition to component faults, sensor faults are also detectable and it also works in the presence of measurement noise. Authors in [117] have applied a different type of neural network namely probabilistic neural network and they have investigated the effectiveness of this method for gas turbine engine fault diagnosis. Component faults are considered in a turbofan engine, shaft speed, pressures and temperatures are considered as deciding factors. Operating conditions are also provided about the ambient conditions, flight speed and fuel flow rate. Using these data, type, location and severity of faults are determined. Fault patterns contain five different noise levels which are single and concurrent fault scenarios. In [118] NN was applied for fault diagnosis in a gas turbine

engine and fuel related faults.

Neural networks are not only practical tools for fault detection in civilian aircraft engines but also powerful tools in military engine applications. In [119] three neural network architectures are applied to F16 flight line data for diagnostics purposes. More information on the data acquisition, pre-processing, types of neural networks and post-processing can be found in [119]. In [120] the concept of resource allocating network (RAN) is combined with an RBF neural network for fault detection in a gas turbine engine. Measurements from six engine sensors are utilized and all fault scenarios happen when the engine is working in the cruise mode. The scheme has been tested for several fault scenarios. Support vector machine (SVM) is another method which has been applied for gas turbine engine fault detection and isolation [121].

Another type of neural network which has proven to be useful in diagnostics, especially for nonlinear dynamical systems, is the dynamic neural network (DNN) [122, 123]. They are applied to aerospace systems such as satellite diagnostics [124, 125, 126, 127]. They have also shown promising results for gas turbine engine related applications . Fault diagnosis is accomplished in [128] for a turbo-charger applying dynamical neural network structure. In [129], Mohammadi *et al.* have proposed a dynamic neural network-based methodology for anomaly detection in the gas turbine engine. This novel architecture belongs to a class of locally recurrent feed-forward neural networks with the processing units equipped with dynamic characteristics. Different fault scenarios verify the efficacy of this novel method.

Researchers have found interest in integrating different diagnostic methods and developing a hybrid approach for this purpose. Volponi *et al.* in [94] have used Kalman filters and neural networks methodologies to find the malfunction and deviations from the normal engine behaviour. The individual methodologies have also

been compared in this study. Another possible combination is the integration of neural networks with genetic algorithm (GA) method. In [130], the neural network part of the scheme is used for engine components fault diagnostic and the GA is applied for sensor bias detection and estimation. By integration the two methods one can take benefit from each method's advantage. Neural networks enable nonlinear estimation and GA methods bring more robustness. The results are promising in terms of fewer missed detection and false alarms. Authors in [131] have developed a two stage integrated model which integrates GA and nested neural networks. The scheme is tested on an engine and has proven to increase the accuracy, reliability and consistency of the results as well as reducing the number of runs and computational loops which are important factors in data-based methods.

Neuro-fuzzy methods are also popular methods in the field of engine diagnostics. The reference s[111] and [132] are examples of integration of expert systems with neural networks. In [49] Palade *et al.* have applied neuro-fuzzy methods for both residual generation and residual evaluation. The method is mainly applied for actuator fault detection in a gas turbine.

1.1.2 Gas Turbine Engine Prognostics

After a fault is detected and isolated in the gas turbine engine, the prognostic scheme must be activated to find the failure progression trend and give an estimation about either the system's remaining useful life (RUL) or the health state of the system at a given future time. Prognostic is nowadays recognized as a key feature in maintenance strategies [133]. Once the RUL becomes predictable, maintenance actions can be planned and disastrous events can be prevented. The number of literature in this field is few as compared to the work done in the field of FDI. In a gas turbine engine - as mentioned earlier- the gas path measurements are available and the goal is to

relate these measurements to the engine health parameters, namely efficiency and mass flow rate, and to finally find how deviation from the normal performance in the system can change these parameters. Some model-based and data-driven methods for engine prognostics will be presented in the following.

In [134] a probabilistic life prediction approach is suggested for prognosis of aircraft jet engine components. This method derives a physics-based mathematical model which uses information from finite element analysis and damage accumulation algorithms. The probability of fracture of titanium compressor disk is calculated using real usage data for an actual F-16/F100. Mahulkar *et al.* [135] developed a model for degradation modelling in an engine actuator. The degradation mathematical model is used for predicting the degradation. It should be noted that the model is only valid for the specific type of fault introduced in the paper. In [136] a robust prognostic technique is introduced. It consists of two stages. In the first stage a wavelet filter based method is used for fault diagnosis and once the fault is detected in the second stage self organizing maps (SOM) are used for performance degradation assessment.

Orchard *et al.* [137] have proposed a novel prognostic scheme which is based on Bayesian estimation techniques. They have developed a particle filtering framework which has a variety of applications in the field of science and engineering. *A posterior* probability density function of the state is predicted using the state dynamic model. The method is tested for crack fault prognosis and has proven to be robust for long-term prediction. More information on model-based or physics-based prognostic can be found in the references [57, 138, 52, 56, 139]. Model-based methods are accurate and they give physical understanding of the system under study. This is s great advantage but the drawback of these methods is that in most of the complicated systems a mathematical model is not available or is inaccurate because of many simplifications resulting from ignorance in modelling or lack of information and formulation.

Data driven approaches as pointed out earlier use real data that may come either from sensor measurements or the operator measures. Available methods in the literature which are relatively rare, can be divided into AI based methods and statistical based methods. Using NNs for prediction is a suitable choice as they are adaptable, robust tools, general and open tools [133]. In [140] Wang has proposed a dynamic wavelet neural network architecture (DWNN) for fault prognostics. This architecture has storage capacity and is capable of predicting into the future. This prognosticator can be used as a link between diagnostic and maintenance scheduler part of a DPHM scheme. DWNNs are flexible in terms of nonlinear mapping and hardware implementation and parallel processing as compared to other time series prediction methods such as ARIMA. The output of the network is the fault growth and the results can always be updated as more data becomes available. More discussions are presented in this paper about stability analysis, uncertainty management and performance assessment. More information on the uncertainty associated with prognostic especially in gas turbine engines can be found in [141, 142].

A dynamic neural network model is used in [143] for engine remaining life computation. The inputs to the network are some real data from sensors installed in the engine gas path such as spool speed and gas temperature and then the stress and temperature of the future time of some critical engine components are predicted. The predicted measurements reflect the engine's health in the future and facilitate an estimation about the remaining useful life of the system or its sub-systems. Xu *et al.* [144] have investigated the predictive performance of neural networks for failure prognosis in engine systems. RBF NNs are used and the effect of hidden layer nodes are investigated. They have concluded that NNs outperform ARIMA modelling methods. [145] is another example of data-driven method for engine prognostics where fault prediction is achieved by using temporal Kohonen maps.

Some statistical and time series prediction methods have been applied for the problem of gas turbine engine trend analysis and prediction. Neural networks can also be regarded as the problem of time-series prediction as they are trying to capture the relation between the measured data and map them into the future [81]. In [146] performance deterioration of a turbofan engine is predicted using ARIMA time-series prediction methods. The authors consider one variable that reflects the effect of degradation and use its history to predict its future trend. The variance of the forecasting method serves for performance assessment. ARIMA method is more useful for short-term prediction. Some hybrid methods also exist in the literature. In addition to the above mentioned work [147, 148, 86] have combined different AI methods or AI method with model-based method or AI methods with statistical based methods .

1.2 Motivation of the Work

In this thesis the problem of gas turbine engines prognostics based on an artificial-intelligence degradation trend analysis method is studied. Prognostics is part of a diagnosis, prognosis and health Monitoring (DPHM) scheme. Condition-based maintenance systems, as opposed to time-based maintenance systems which are mostly corrective and preventive maintenance, focus on the timeliness of the maintenance action.

Prognostic is currently a key feature and game-changing technology which helps the users to avoid unnecessary maintenance costs. Prognostics has drawn the attention of the users only recently. An efficient prognostic scheme can result in the significant enhancements in the readiness and reliability of complex and dynamical systems. Prognostic can be even more important in complex systems such as aircraft engines where it allows minimization of system breakdowns and downtimes. The goal

of prognostic is to find the time to failure (TTF) of the system based on some current measurements and evidence from the system. Another approach to prognostic when it is not practical to find the TTF is instead to find the future state of the system for some steps ahead and then decide if the system will work efficiently at that time or some maintenance is required.

For the case of a gas turbine engine, performing maintenance before the system health is close to its critical point can be very inefficient economic-wise. On the other hand the operators cannot wait until the components reach their failure point as aircraft engines are safety critical machines. Thus an early prediction on the degradation levels can save lives and also can be cost-efficient. Gas turbine engine components like all other mechanical components degrade over time as they work under varying conditions such as different level of stress and temperature. These changes are even more significant when the engine is in the take-off mode compared to other modes of its mission profile. The consequence of high stress and temperature lead to changes in the components efficiency and mass flow rate which are called engines' health parameters. As the health parameters are not directly measurable, we study their effect on the gas path measurements.

Degradations can be divided into soft and hard degradations. Foreign object damage (FOD) is an example of hard degradation. In this thesis the increasing trends of the engine soft degradation i.e. fouling, erosion and corrosion are studied and predicted using the gas path measurements such as inlet temperature and pressure. Toward this end different neural networks are used such as recurrent neural networks (RNN)and nonlinear autoregressive neural networks (NARNN) and their effectiveness and accuracies in prediction are compared with the number of input parameters and computations. The goal is to predict the single spool jet engine degradation level for a number of flights ahead based on the current and past engine's gas path measurements

and decide if the next flights would be safe or maintenance actions are required.

Finally, it is worth to mention that prognostics in general is in its novice stage, especially in the area of gas turbines and to the author's best knowledge few works exist on the prediction of soft degradations using AI methods.

1.3 Thesis Contribution

In this thesis, novel solutions for the problem of the jet engine prognostics are proposed based on artificial intelligence approaches. Towards this end, the problem of the jet engine degradation trend prediction is addressed by using two different neural network architectures. Each of the proposed neural network architectures are proposed with features which enable an acceptable prediction. The obtained prediction results may be used later in an CBM scheme. Towards solving this problem, some contributions are made in this thesis which are detailed below:

After conducting an extensive literature review, it was concluded that although there has been extensive related works in the field of FDI, not much exists in the field of failure prognostics. Hence, there is an immediate need for developing related methodologies. Neural networks are applied to a large number of real-world problems with considerable complexity, but they have not been applied to the problem of the jet engine prognostics and degradation trend prediction.

Neural networks performance and reliability depend mostly on the data they are provided with. Using a set of high quality and validated data can enhance the results. This is especially important when one is projecting the results into the future. Towards this end, one should provide the neural networks with proper examples of the jet engine affected by different types and levels of degradations. In Chapter 3 of this thesis we have modelled the two main causes of the engine degradation namely fouling and erosion. These models are added to a high quality engine simulator. We

have demonstrated the validity of the data obtained from our SIMULINK model by comparing the results of different scenarios with the similar ones obtained from a gas turbine simulation program called GSP [175].

Once a suitable data set is available, the prediction problem is addressed in Chapter 4 by using a recurrent neural network (RNN) scheme. Recurrent neural networks due to their global feedback paths from the network outputs to the network inputs, can capture the dynamics of the model and temporal behaviour. Unlike some references in the literature that have used static neural networks for prediction, we have used RNN and obtained satisfactory results. This has been demonstrated through various simulation scenarios and case studies for the two degradation types with different levels, both quantitatively and qualitatively. Moreover, we have considered and overcome the uncertainty problem associated with prediction problems by defining a lower and upper prediction bounds. The bounds along with the real and predicted data points are shown for all the scenarios.

In Chapter 4, a nonlinear autoregressive with exogenous inputs (NARX) neural network architecture is proposed for an enhanced trend prediction. The proposed network is equipped with unique features which makes it capable of prediction. NARX architecture is a type of dynamical recurrent neural network that enables reliable long-term prediction by learning the long-term dependencies from the provided data set. They are used for the first time in this thesis for engine degradation prognostics. This network makes use of the benefits of both the recurrent and the dynamic neural networks. The prediction bounds are considered again for uncertainty management and handling noisy measurements.

The prediction performance of the two proposed neural networks need to be compared quantitatively. Towards this end, evaluation methods are introduced and applied to the obtained results from different scenarios of each network. The criterion

selected is called NAIC, which considers prediction error, volume of the data used for training, number of network parameters and network complexity. A smaller NAIC implies a better network performance.

The capabilities of our proposed neural networks in, make them ideal tools for prediction purposes which can later be used for condition based maintenance instead of conventional time based maintenances.

1.4 Thesis Outline

The remainder of this thesis is organized as follows. In Chapter 2, some background information which will be used in this thesis will be briefly reviewed. We will start by introducing three types of neural networks namely recurrent neural networks, time-delay neural networks and nonlinear autoregressive neural networks which will be used in the following chapters for engine prognostics. Afterwards we will present single-spool jet engine nonlinear mathematical equations to be used for developing a simulation model of the aircraft engine. We will finish the chapter by introducing the GSP software and shedding more light on the importance of performance monitoring and trend analysis. In Chapter 3, we will introduce different sources of engine deterioration and then we will focus on modelling two of the most important ones, namely fouling and erosion. The degradation models will be integrated with the model introduced in Chapter 2 and the data derived from this model will be validated by GSP software to be used later for prognostics. These data will be used and analyzed in Chapter 4 to predict the engine condition for certain flights ahead by using recurrent neural network for engine that is affected by fouling or erosion. Various simulation results sill show the prediction performance of our method and we will describe our methodology in more detail. In Chapter 5 we will continue our investigation on the capability of dynamical neural networks for engine degradation

level prediction (prognosis). More simulations will be performed to demonstrate the effectiveness of our proposed methods. At the end of this chapter the two methods will be compared with each other. Finally, the conclusion will be drawn in Chapter 6 and some future work will be discussed.

Chapter 2

Background

In this chapter we present some background information on the material which are to be used in this thesis. In this work, the problem of prognosis in a single spool jet engine using the neural network methodology is addressed. This goal is achieved by applying two different neural networks on a single spool jet engine data derived from a MATLAB/SIMULINK model. Firstly, we introduce recurrent neural networks (RNN) and then in Chapter 4 we will see how this network is used for single spool engine degradation prognostics. Moreover, we will introduce and describe a type of dynamical neural networks (DNN), namely nonlinear autoregressive neural networks (NARNN) which will be used for prognostic purposes in Chapter 5 of this thesis. Finally, we will briefly describe the single spool engine SIMULINK model to which the prognostic scheme will be applied to, as well as some of the nonlinear equations that are used for developing this model.

2.1 Artificial Neural Networks

Enabling machines to find a right solution for complex problems in a more human-like fashion is known as artificial intelligence (AI). This generally applies to computer

machine via an algorithm, which has approximated human brain characteristics. Artificial intelligence has four main branches that include case based reasoning, genetic algorithms, expert systems and neural networks [36], which can be chosen based on the specific application.

An artificial neural network (NN) is an information-processing paradigm that is inspired by the circuit of biological neurons with the aid of learning to recognize patterns from a certain data. The key point of this paradigm is the novel data analysis technique by using a solution with reasonable accuracy when the underlying data relationship is unknown [149]. Once the NN is trained on collection of data, it can predict the outcome by detecting similar patterns of the input. Thus using NN for analyzing the data stream can make the problem of detecting and predicting possible faults even if the data are imprecise and noisy. Different neural network architectures exist but in this chapter we will focus on the ones that are used in this thesis.

2.1.1 Recurrent Neural Networks (RNN)

One special form of neural networks is called recurrent neural network (RNN). The network can have single or multiple hidden layers of neurons. The fundamental difference between RNNs and feedforward neural networks is that they have one or more feedback loops. The feedback loop can appear in many forms between any two neurons or layers. Recurrent neural networks exhibit complex dynamics as they consist of a large number of feedforward and feedback connections [150]. These connections give them extra advantage over feedforward NNs for handling time-series and dynamical related problems. A recurrent network with a smaller network size may be equivalent to a large or complicated type of feedforward NN architecture.

There has been a wide application of recurrent networks in the field of intelligent

control, system identification and dynamical systems applications. In these applications, theoretical study of stability, convergence of the network and their functional approximation capabilities are regarded as important. In [151] an adaptive observer is proposed by using a generalized recurrent neural network where the learning is occurring on-line with no off-line learning. The overall adaptive observer scheme is shown to be uniformly ultimately bounded. In [152] the authors have presented a real-time learning control scheme for unknown nonlinear dynamical systems using recurrent neural networks (RNNs). A generalized real-time iterative learning algorithm is developed and used to train the RNN. The paper shows that an RNN using the real-time iterative learning algorithm can approximate any trajectory tracking to a very high degree of accuracy.

The use of recurrent neural networks as predictors and identifiers in nonlinear dynamical systems has received significant attention [6]. They can exhibit wide range of dynamics, because of feedback, and are also tractable nonlinear maps. In this thesis recurrent neural network models are considered as massively interconnected nonlinear filters with feedbacks that enables a more potential structural richness. The RNN architecture is used for prediction purposes, hence we present the material which are related to this aspect regarding the recurrent neural networks.

The basic building blocks of a discrete time predictor are adders, delays, multiplies and for the nonlinear case zero-memory nonlinearities. In order to use neural networks as nonlinear predictors, zero-memory nonlinearities such as threshold, piecewise-linear and logistic are required. These basic building blocks, form the neurons in the NN architecture.

The inputs are assumed to be the delayed version of the neuron output ($y(k)$). In the problem of prediction, the nature of inputs to the network must capture some information about the time evolution of the discrete time signal or measurement.

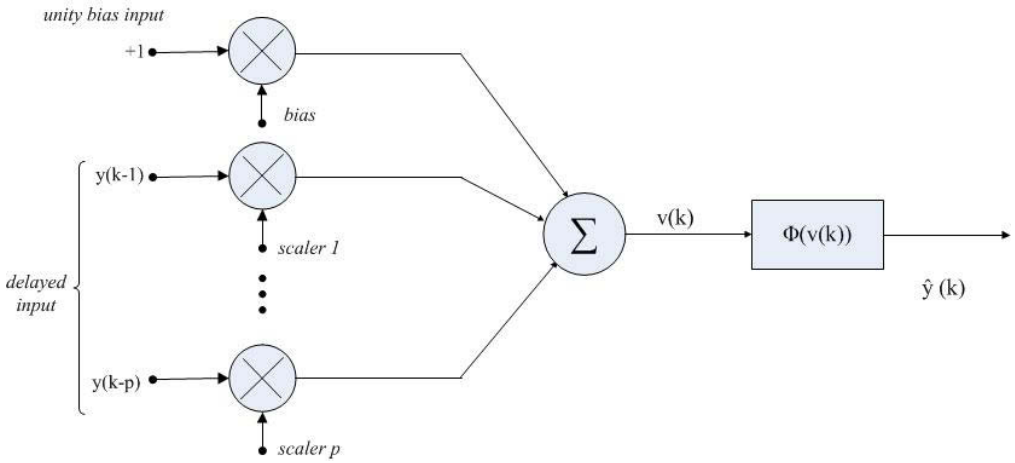

Figure 2.1: Structer of a neuron for prediction [6].

The simplest situation is for the inputs to be the time-delayed version of the output signal, i.e. $y(k-i), i = 1, 2, ..., p$, which is known as a tapped delay line. This type of network provides a short-term memory of the signal. The overall predictor can be represented as:

$$\hat{y}(k) = \phi(y(k-1), y(k-2), ..., y(k-p)) \tag{2.1.1}$$

where ϕ represents the nonlinear mapping of the neural network.

A typical recurrent neural network is depicted in Figure 2.2. If we consider connecting the delayed versions of the output $\hat{y}(k)$ of the network to its input, all together with the delays, one can introduce memory to the network and this structure becomes suitable for prediction. Information on the stability of such network can be found in [6].

As can be seen in the figure, the feedback within the network can be local or global. The global feedback is achieved by connecting the network output to the network input while the local feedback is produced by introduction of feedback within

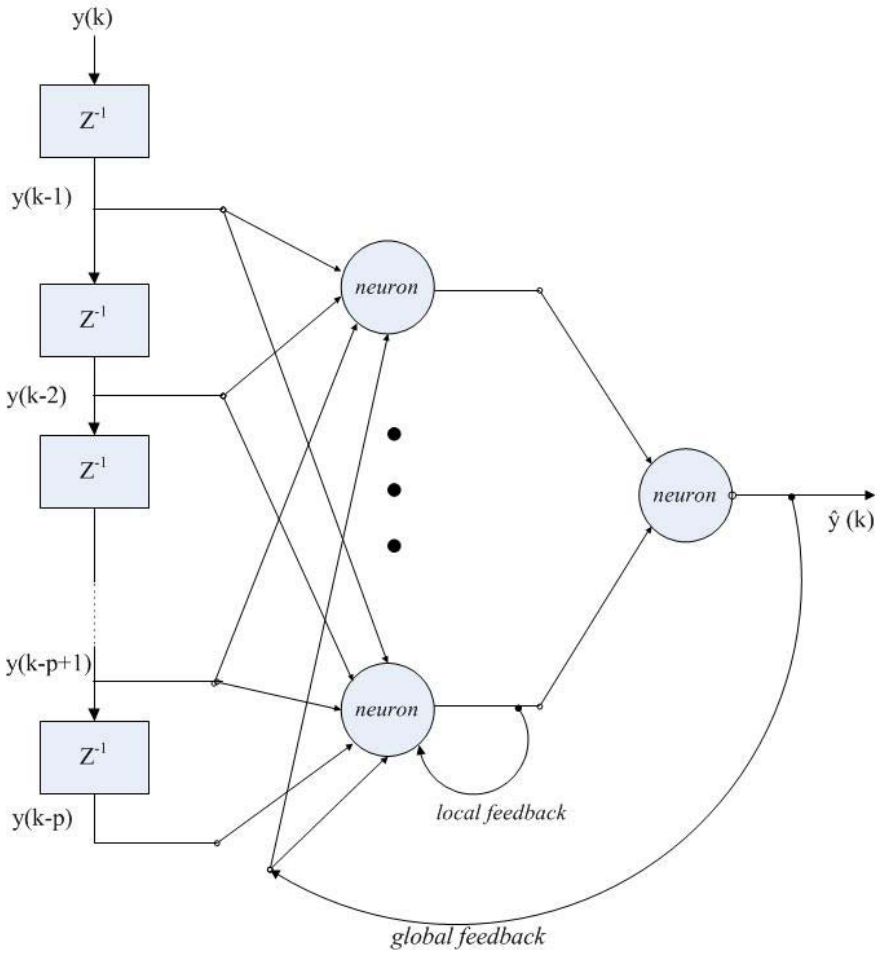

Figure 2.2: Canonical form of a recurrent neural network for prediction [6].

the hidden layer. The output of an RNN when used as a predictor, employing a global feedback can be represented by,

$$\hat{y}(k) = \phi(y(k-1), y(k-2), ..., y(k-p), \hat{e}(k-1), ..., \hat{e}(k-q)) \tag{2.1.2}$$

where $\hat{e}(k-j) = y(k-j) - \hat{y}(k-j)$. In other words, by adding the feedback and tapped-delay line to a static network architecture, we are adding memory to the network and then it becomes capable of prediction. This concept is shown in Figure 2.3.

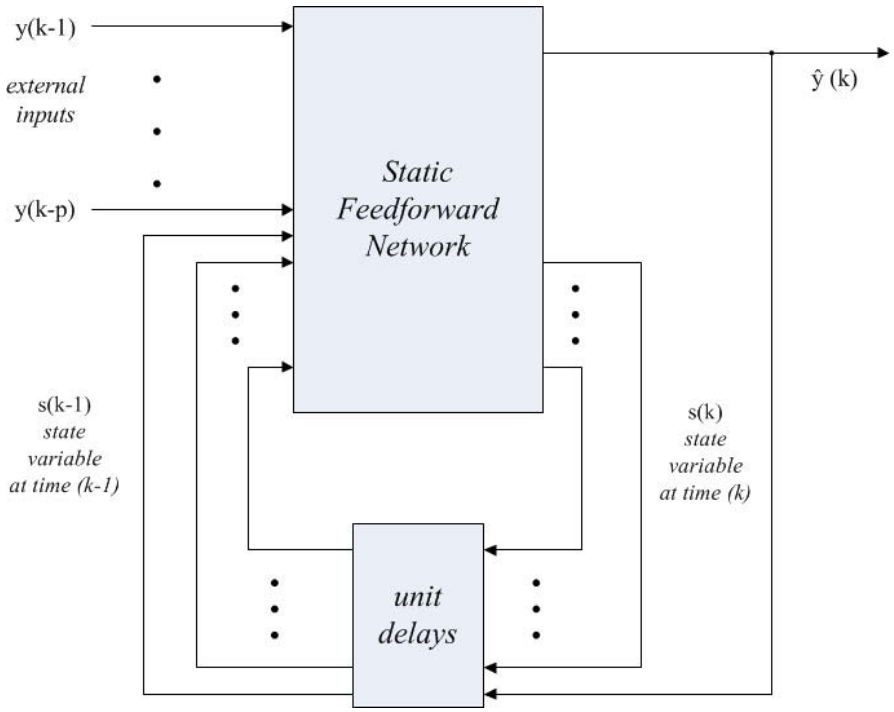

Figure 2.3: Canonical form of a recurrent neural network for prediction [6].

The choice of the structure depends on the dynamics of the signal, learning algorithm, the prediction performance and the application. It is important to note that, there is no fast rule by which the best structure can be found for a particular problem [153]. In the following we will introduce some of the activation functions used for prediction and the activation function used for our model in this work will be presented in Chapter 4.

To introduce nonlinearity to the network, we use nonlinear activation functions. Any nonlinear function help us achieve this goal, however for gradient-descent learning algorithm, this function $\sigma(.)$ should be differentiable and belong to the class of sigmoid function. Surveys of neural transfer functions can be found in [154]. Examples of sigmoidal functions are:

$$\sigma_1(x) = \frac{1}{1+e^{-\beta x}} \tag{2.1.3}$$

$$\sigma_2(x) = tanh(\beta x) = \frac{e^{\beta x} - e^{-\beta x}}{e^{\beta x} + e^{-\beta x}} \tag{2.1.4}$$

$$\sigma_3(x) = \frac{2}{\pi} arctan(\frac{1}{2}\pi\beta x) \tag{2.1.5}$$

$$\sigma_4(x) = \frac{x^2}{1+x^2} sgn(x) \tag{2.1.6}$$

where β belongs to the set of all real numbers. According to Cybenco [155] a neural network with a single hidden layer of neurons with sigmoidal functions and enough neurons can approximate an arbitrary continuous function.

Recurrent Neural Network Architectures

Two common ways of producing recurrent connections in a neural network are *activation feedback* and *output feedback* which are shown in Figure 2.4.

The output of a neuron shown in Figure 2.4 on the top is obtained as

$$\left.\begin{aligned} v(k) &= \sum_{i=0}^{M} w_{u,i}(k)u(k-i) + \sum_{j=1}^{N} w_{v,j}(k)v(k-j) \\ y(k) &= \Phi(v(k)) \end{aligned}\right\} \tag{2.1.7}$$

where $w_{u,i}$ and $w_{v,j}$ represent the weights associated with u and v, respectively. The output of a neuron shown in Figure 2.4 on the bottom is obtained as

$$\left.\begin{aligned} v(k) &= \sum_{i=0}^{M} w_{u,i}(k)u(k-i) + \sum_{j=1}^{N} w_{y,j}(k)y(k-j) \\ y(k) &= \Phi(v(k)) \end{aligned}\right\} \tag{2.1.8}$$

where $w_{y,j}$ represent the weights associated with the delayed outputs.

Each of these types when employed in a general feedforward structure, build up

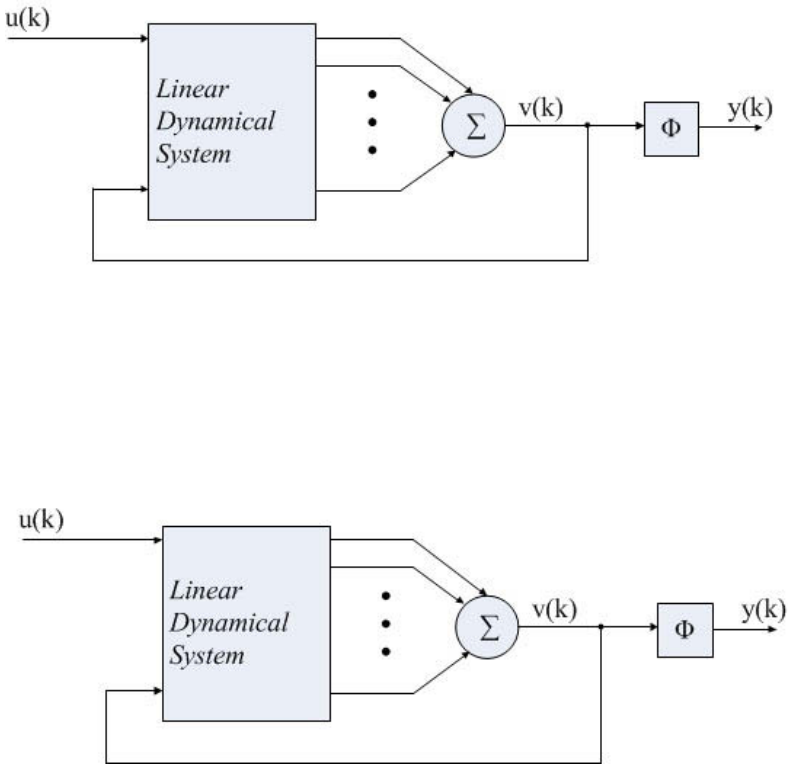

Figure 2.4: Recurrent neural network architectures. The plot on the top is the activation feedback scheme and the plot on the bottom is the output feedback scheme [6].

a type of neural network known as *locally recurrent-globally feedforward* (LRGF) architecture which is shown in Figure 2.5. This architectures allows one the use of dynamic neurons both within the input and the output feedback as represented by H_i and H_{FB}, respectively.

Another type of recurrent neural network is known as Elman network with one hidden layer. A simple example is depicted in Figure 2.6. This architecture consists of a multi layer perceptron (MLP) network with an additional delayed input.

Another architecture is the Jordan recurrent neural network which is shown in Figure 2.7. The network consists of an MLP with one hidden layer plus a feedback loop from the output layer to an additional input which is called context layer. This

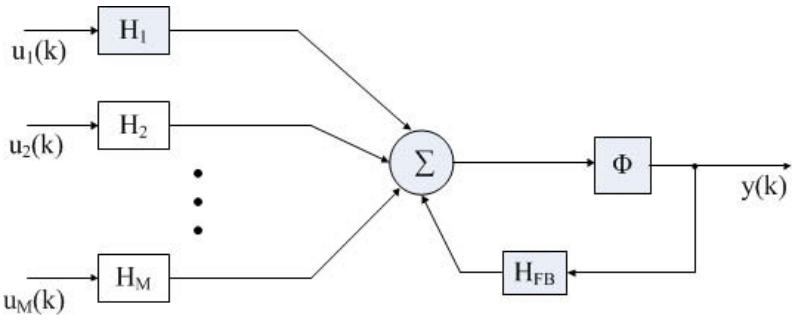

Figure 2.5: General LRGF architecture [6].

context layer contains self-recurrent loops. The structure of both Elman and Jordan networks is locally recurrent and thus they have limitations in including past information. Because of the limitations mentioned above associated with Elman and Jordan networks in this thesis we have used a fully connected recurrent neural network which has a rich representation of past outputs and is shown in Figure 2.8

The network consists of three layers, namely input layer, processing layer and the output layer. For each neuron $i, i = 1, 2, ..., N$, the elements $u_j, j = 1, 2, ..., P + N + 1$ of the input vector to a neuron$u$ are weighted, then are summed to produce an internal activation function of a neuron v, which is ultimately fed through a nonlinear activation function ϕ which yields the output of the ith neuron y_i. The function ϕ is a sigmoidal function that is monotonically increasing with slope β.

The weight of the ith neuron at the time instant k form a (P+N+1)$\times$ 1 dimensional weight vector $w_i^T(k) = [w_{i,1}(k), ..., w_{i,P+N+1}(k)]$ where P is the number of external inputs, N is the number of feedback connections and $(.)^T$ denotes the vector transpose operation. The additional element of the weight vector w is the bias input weight and the feedback consists of the delayed version of the output signals of the RNN. The above concept is presented by the following equations:

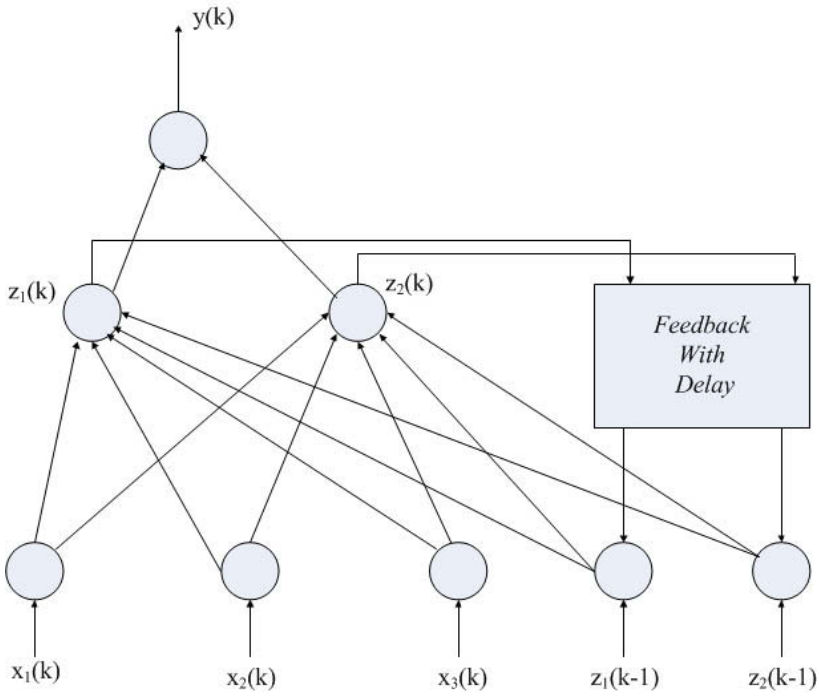

Figure 2.6: An example of Elman recurrent neural network [6].

$$y_i(k) = \Phi(v_i(k)), \quad i = 1, 2, ...N \tag{2.1.9}$$

$$v_i(k) = \sum_{l=1}^{P+N+1} w_{i,l}(k) u_l(k) \tag{2.1.10}$$

$$u_i^T(k) = [S(k-1), ..., S(k-P), 1, y_1(k-1), y_2(k-1), ..., y_N(k-1)] \tag{2.1.11}$$

where the (P+N+1)$\times$ 1 dimensional vector u includes both the unity valued constant bias input as well as the external and feedback inputs to each neuron.

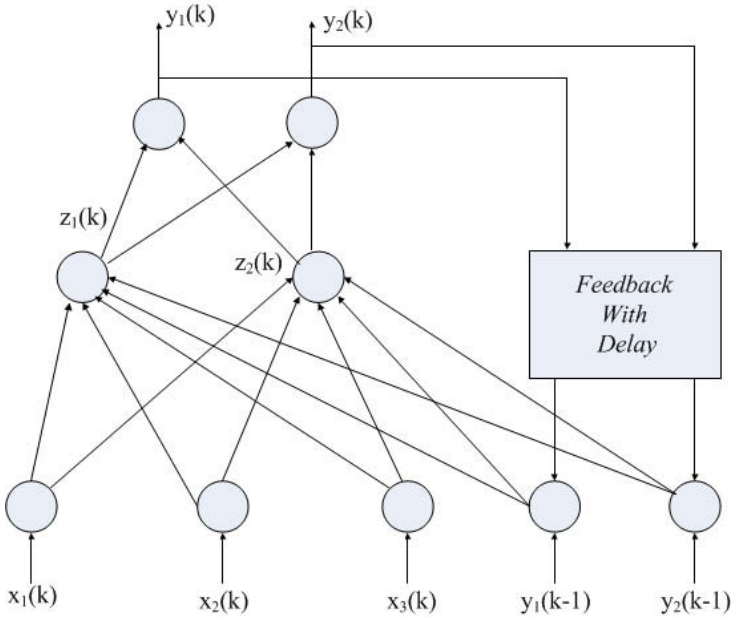

Figure 2.7: An example of Jordan recurrent neural network [6].

Learning Algorithm for the RNN

Generally in any neural network, the goal is to determine an adaptive algorithm that can adjust the parameters of the network based on the set of input-output data set. Different learning methods exist which can be categorised as supervised and unsupervised learning. In this part we present the learning algorithm for recurrent neural network shown in Figure 2.8. The neurons are depicted by circles and incorporate the operation $\Phi(.)$. For the nth neuron, the weights are $(P+N+1)\times 1$ dimensional weight vector $w_n^T = [w_{n,1}, ..., w_{n,P+N+1}]$, where P is the number of external inputs, N is the number of feedback connections, and the remaining term is the bias input weight. The feedback connections as mentioned previously represent the delayed output signals of the RNN. Equations (2.1.9) to (2.1.11) describe our RNN.

For the nonlinear prediction paradigm, there is only one output neuron for the RNN. The error is defined as the difference between the output value of the network and the target (desired output) value. The instantaneous error $e(k)$ can be positive, negative or zero. Training of the RNN is based on minimizing the instantaneous

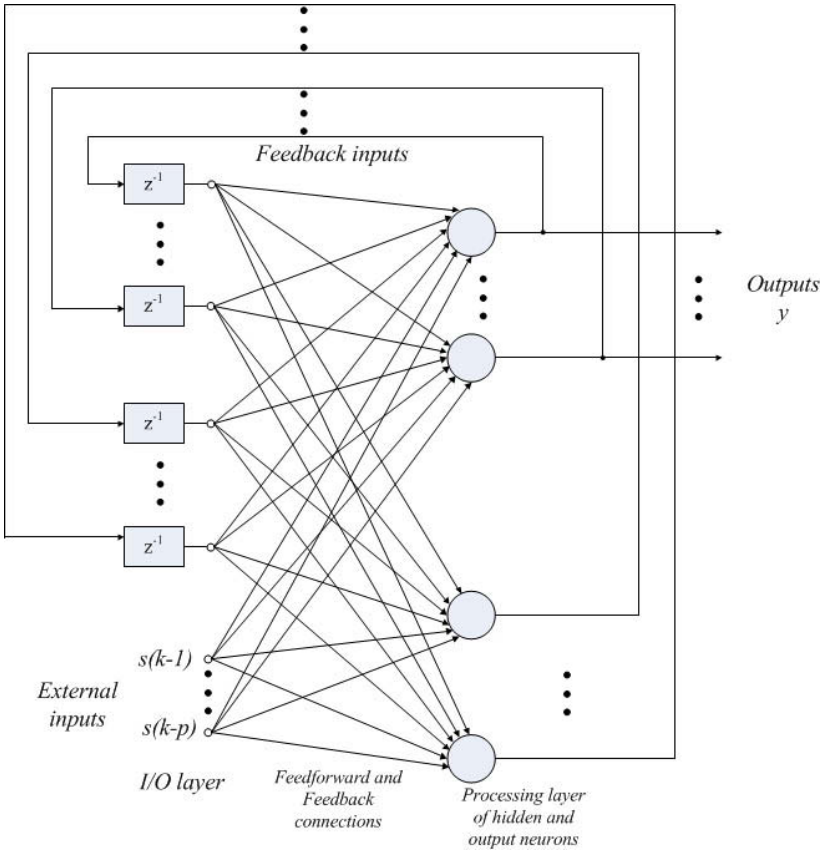

Figure 2.8: A fully connected recurrent neural network [6].

squared error at the output of the first neuron of the RNN [156]. The error to be minimized is expressed as:

$$min(\frac{1}{2}e^2(k)) = min(\frac{1}{2}[s(k) - y_1(k)]^2) \tag{2.1.12}$$

where e(k) represents the error at the output y_1 of the RNN and s(k) is the teaching signal. The above objective function has to be minimized during the training. The correction for the lth weight of the neuron n at the time instant k can be obtained

as follows:

$$\begin{aligned}\Delta w_{n,l}(k) &= -\frac{\eta}{2}\frac{\partial}{\partial w_{n,l}(k)}e^2(k) \\ &= -\eta e(k)\frac{\partial e(k)}{\partial w_{n,l}(k)}\end{aligned} \tag{2.1.13}$$

The external signal vector $S(k)$ does not depend on the elements of $w_{n,l}$, hence the gradient becomes

$$\frac{\partial e(k)}{\partial w_{n,l}(k)} = -\frac{\partial y_1(k)}{\partial w_{n,l}(k)} \tag{2.1.14}$$

Using the chain rule this can be rewritten as [157]

$$\begin{aligned}\frac{\partial y_1(k)}{\partial w_{n,l}(k)} &= \Phi'(v_1(k))\frac{\partial v_1(k)}{\partial w_{n,l}(k)} \\ &= \Phi'(v_1(k))\Big(\sum_{\alpha=1}^{n}\frac{\partial y_\alpha(k-1)}{\partial w_{n,l}(k)}w_{1,\alpha+p+1}(k) + \delta_{n1}u_l(k)\Big)\end{aligned} \tag{2.1.15}$$

where $\Phi' = \dfrac{\partial \Phi}{\partial v_1(k)}$ and Φ is a nonlinear sigmoid function as the network activation function and

$$\delta_{nl}\begin{cases}1, & n = l \\ 0, & n \neq l\end{cases} \tag{2.1.16}$$

By assumption when the learning rate η is sufficiently small, we will have

$$\frac{\partial y_\alpha(k-1)}{\partial w_{n,l}(k)} \approx \frac{\partial y_\alpha(k-1)}{\partial w_{n,l}(k-1)} \tag{2.1.17}$$

An indexed set of variables $\{\pi^j_{n,l}(k)\}$ can be introduced to characterize this algorithm for the RNN. These coefficients are called sensitivities. They are defined as

$$\pi_{n,l}^{j} = \frac{\partial y_j(k)}{\partial w_{n,l}} \quad 1 \leqslant j, n \leqslant N, 1 \leqslant l \leqslant p+1+N \tag{2.1.18}$$

The above equation is used to recursively compute the values of $\pi_{n,l}^{j}$ for every time step k and all appropriate j, n and l are as follows

$$\pi_{n,l}^{j}(k+1) = \Phi^{'}(v_j)\left[\sum_{m=1}^{N} w_{j,m+p+1}(k)\pi_{n,l}^{m}(k) + \delta_{nj}u_l(k)\right] \tag{2.1.19}$$

We assume the following initial conditions

$$\pi_{n,l}^{j}(0) = 0. \tag{2.1.20}$$

For simplification, we introduce three new matrices namely $\Pi_j(k)$, $U_j(k)$ and the diagonal matrix $F(k)$ which are $N \times (N+p+1)$, $N \times (N+p+1)$ and $N \times N$ respectively:

$$\Pi_j(k) = \frac{\partial y(k)}{\partial w_j(k)} \ , \ y = [y_1(k), ..., y_N(k)], j = 1, 2, ..., N \tag{2.1.21}$$

$$U_j(k) = \begin{bmatrix} 0 \\ \vdots \\ u(k) \\ \vdots \\ 0 \end{bmatrix} \quad j = 1, 2, ..., N, \tag{2.1.22}$$

$$F(k) = diag[\Phi^{'}(u(k)^T w_1(k)), ..., \Phi^{'}(u(k)^T w_N(k))] \tag{2.1.23}$$

Thus the gradient updating equation for the recurrent neuron can be described as follows [156]

$$\Pi_j(k+1) = F(k)[U_j(k) + W_a(K)\Pi_j(k)], \quad j = 1, 2, ..., N. \tag{2.1.24}$$

where W_a denotes the set of those entries in $\mathbf{W}$ which correspond to the feedback connections. The learning procedure described above is a supervised learning algorithm. Selecting an optimal ANN architecture is an open problem and needs to be designed as warranted by the application domain. In the following we will introduce the nonlinear autoregressive neural network (NARNN), which is used in this thesis for prognostics.

Remark

Two different learning types are generally used, namely batch learning and online learning. Batch learning is also known as epoch wise or off-line learning. In this method the weights are adapted once the entire training set has been presented to the system. The counter part is called incremental learning or pattern learning in which the weights are updated upon the instantaneous error and are updated immediately after each pattern is fed in. In this thesis batch learning is used.

Before we start reviewing the background information about the gas tubine engine and the mathematical model for a single-spool engine, we will introduce the other type of dynamical networks that is used in this thesis for engine prognostics namely nonlinear autoregressive neural network (NARNN).

2.1.2 Nonlinear Autoregressive Neural Network (NARNN)

Nonlinear autoregressive neural network (NARNN) is the other type of neural networks that we have considered in this work. This model is a combination of recurrent and dynamic network as it has both feedback path and dynamical neurons or the tapped-delay line. NARNN can be classified as a dynamic recurrent neural network

that is capable of modelling efficiently time series with long-term dependences. Feedback connections may enclose different layers of the network. The general architecture is shown in Figure 2.9. If we consider the external input, the network is called nonlinear autoregressive neural network with exogenous input (NARX). The output of

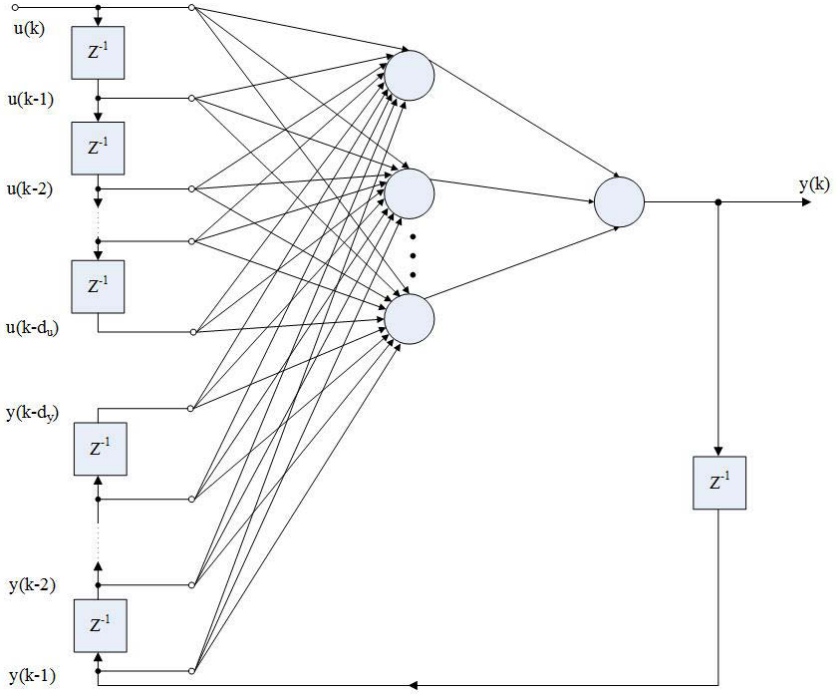

Figure 2.9: Nonlinear autoregressive neural network architecture (NARNN) [7].

this network when having an external input is derived as follows:

$$\begin{aligned} y(k+1) &= f[y(k), ..., y(k-d_y+1); u(k), u(k-1), ..., u(k-d_u+1)] \\ &= f[\mathbf{y}(k); \mathbf{u}(k)] \end{aligned} \tag{2.1.25}$$

where the next value of the dependent output signal $y(k)$ is regressed on previous values of the output signal and previous values of an independent (exogenous) input signal $u(k)$ at time k. The parameters d_u and d_y are memory delays. The NARX model is based on the linear ARX model, which is commonly used in time-series modelling. Two different architectures are defined for this type of neural network

namely parallel and series-parallel architecture.

If we consider the output of the NARNN network to be an estimate of the output of some nonlinear dynamical system that we trying to model, the output should be fed back to the input of the feedforward neural network as part of the standard NARNN architecture, as shown Figure 2.10.

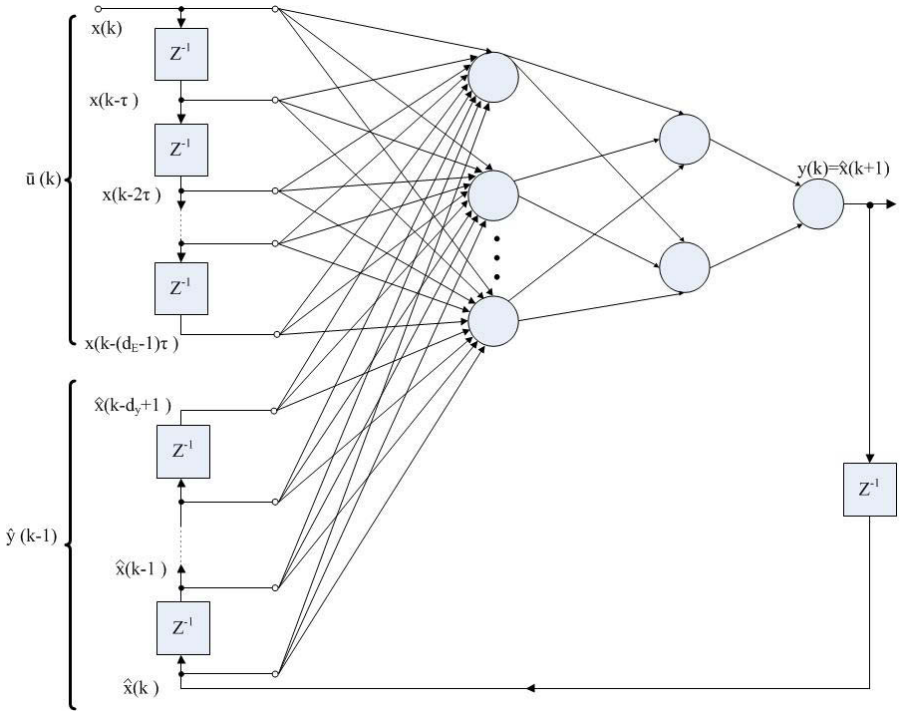

Figure 2.10: The parallel nonlinear autoregressive neural network architecture [7].

For the parallel configuration the output regressor $\mathbf{y}(n)$ is obtained as follows:

$$y_p(k) = [\hat{x}(k), ..., \hat{x}(k - d_y + 1)] \tag{2.1.26}$$

where the P-mode contains d_y past values of the estimated time series. The second configuration is called series-parallel architecture in which the true output is used instead of feeding back the estimated output, as shown in Figure 2.11.

Similar to the P-mode, in the SP-mode the output regressor $\mathbf{y}(n)$ is obtained as

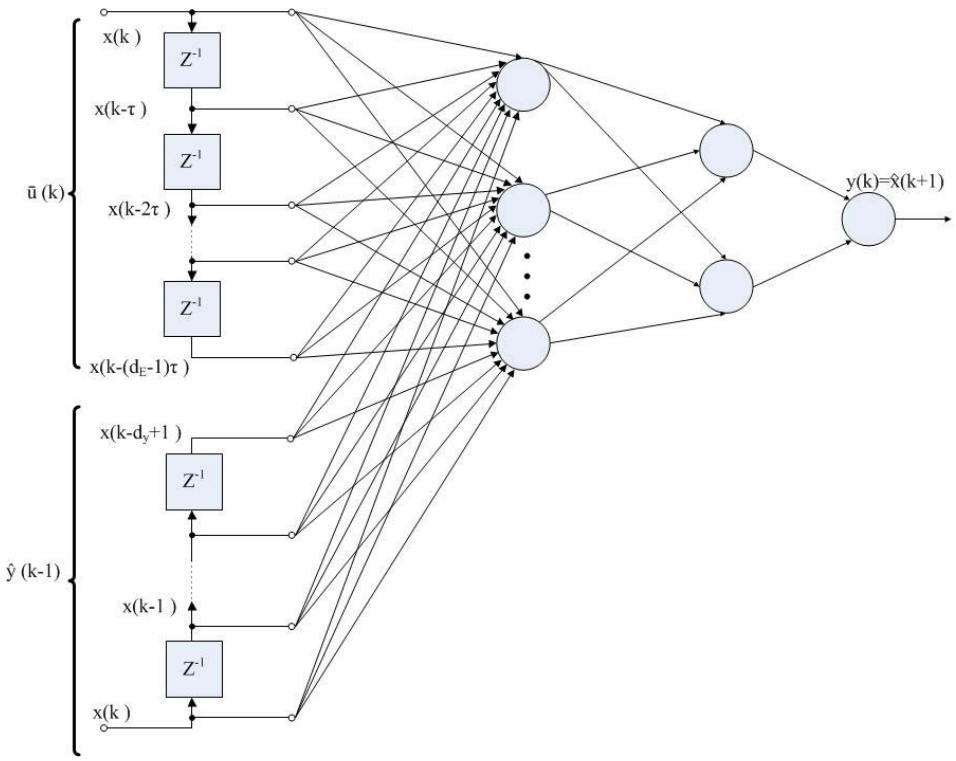

Figure 2.11: The series-parallel nonlinear autoregressive neural network architecture [7].

follows:

$$y_p(k) = [x(k), ..., x(k - d_y + 1)] \tag{2.1.27}$$

where the SP-mode contains d_y past values of the actual time series. This architecture has two advantages. Firstly, the input to the feedforward network is more accurate as it uses the past values of the actual time series not the estimated ones. Secondly the resulting network has a purely feedforward architecture, and static backpropagation can be used for training it.

The NARNN model converges easier and needs less training cycles than a fully recurrent network [158]. NARNN model provides a description of the system in terms of a nonlinear function of delayed input, output, and prediction error, and hence is general enough to approximate any nonlinear dynamical system [159, 160]. For this reason, they have been vastly applied in different applications such as prediction, time-series modelling, chaotic time-series prediction, nonlinear filtering, Control and

modelling, etc. In [160], NARNN based predictive control is proposed and tested on a continuous stirred tank reactor process (CSTR). A comparison with other approaches shows that this method can reduce network training time, improve solution accuracy and results in good control performance under different operating conditions . Gang *et al.* [161] have developed a hybrid nonlinear autoregressive neural network for permanent magnet Linear synchronous motor identification.

The main capability of NARNN is in its prediction. In [162], it has been demonstrated that this architecture is capable of representing nonlinear dynamics as well as long-term dependencies in time-series data. They have been applied in [163] for communication network traffic characterization . In [164] residual analysis using a hybrid Elman-NARNN along with embedding theorem is used to analyze and predict chaotic time series. The NARX network is used to capture the relationship among the predicted value of original time series and residuals and original time series. Filik *et al.* in [164] have investigated on short-term load forecasting using the NARNN by using the consumption value of the electricity energy.

As was defined above, a NARNN network consists of a multilayer perceptron which takes input window of past input and output values and computes the current output,

$$y(k) = \Psi\{u(k), u(k-1)..., u(k-d_u), y(k-1), ..., y(k-d_y)\} \tag{2.1.28}$$

where Ψ is the mapping performed by the multi-layer preceptron in the network. Figure 2.12 shows an example of such network with two input delays, two output delays and three neurons in the hidden layer, i.e. $d_u = 2$ and $d_y = 2$.

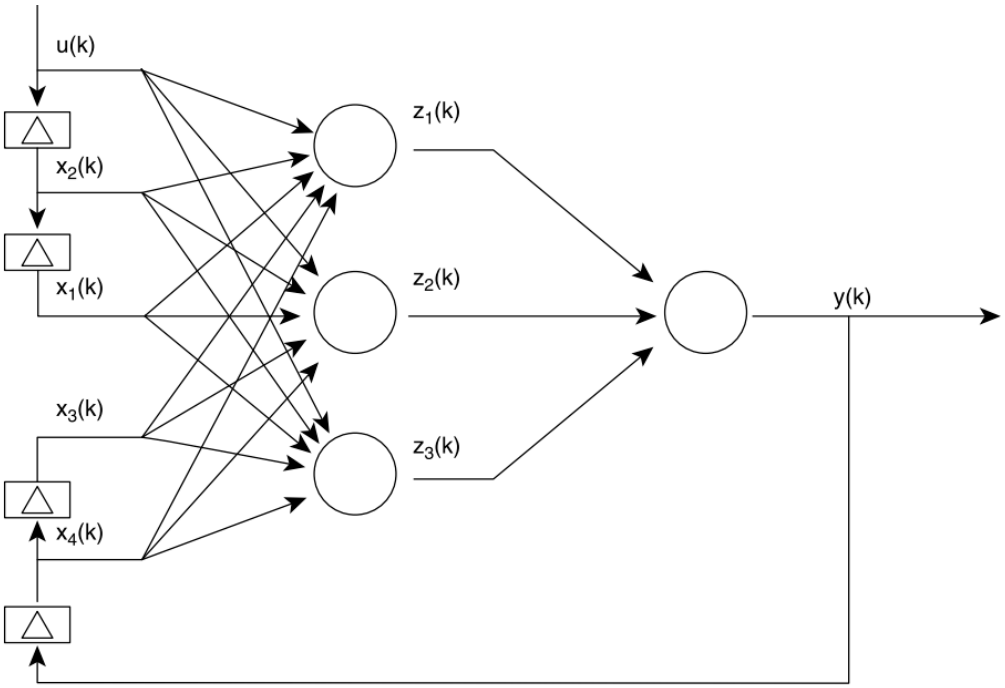

Figure 2.12: Nonlinear autoregressive neural network architecture [7].

The states are updated as follows:

$$x_i(k+1) = \begin{cases} u(k) & i = d_u \\ y(k) & i = d_u + d_y \\ x_{i+1}(k) & i < i < d_u \text{and } d_u < i < d_u + d_y \end{cases} \tag{2.1.29}$$

so that at time k the taps correspond to the values

$$x(k) = [u(k-1), ..., u(k-d_u), y(k-1), ..., y(k-d_y)] \tag{2.1.30}$$

The MLP consists of two nodes organized into two layers. There are H nodes in the first layer and this performs the following function

$$z_i(k) = \sigma\left(\sum_{j=1}^{N} a_{i,j} x_j(k) + b_i u(k) + c(i) \right), \quad i = 1, 2, ..., H \tag{2.1.31}$$

where σ is the nonlinear transfer function, $a_{i,j}$, b_i and c_i are the network fixed weights. Furthermore, the output layer consists of one linear node,

$$z_i(k) = \sum_{j=1}^{H} w_{i,j} z_j(k) + \theta_i, \quad i = 1, 2, ..., H \tag{2.1.32}$$

where $w_{i,j}$ and θ_i are the network weights with fixed value.

Generally speaking, backpropagation [165] is a commonly used technique for training feedforward neural networks whose input contains no self-feedback of its previous outputs, such as the multilayer perceptron (MLP). More information on this method can be found in the following tutorial paper [166]. This general method can be adapted and extended to become applicable to recurrent and dynamical neural networks. Real-time recurrent learning (RTRL) [167] and back-propagation through time (BPTT) [168, 169] are well-known training algorithms. Below we will briefly describe each method.

When BPTT is applied for training a NARNN, the recurrent connections are unfolded in time and the resulting network is treated as an MLP with injected errors. In the unfolded network, the recurrent connections appear as jump-ahead connections, that provide a shorter path for backpropagating the error through the network, hence diminishing the problem of the vanishing gradient [162]. After presenting the data to the network, the error is backpropagated through the unfolded network. In the output units of the recurrent network, the local error is computed and added to the backpropagated value from the subsequent input unit. As the error in the present time step is reduced while taking into account the errors made in the future, the NARX neural network is able to learn long-term dependencies in the data. The BPTT, in its basic format is very computationally intensive [170]. In the modified version one tries to reduce the number of time steps h over which the recurrent connections are unfolded (or the number of network states that is saved). This modified method is

called *Truncated* BPTT [169]. Besides, the algorithm can compute weight updates for each step, but the updates can also be delayed for an additional number of steps D. The application of both changes to the BPTT algorithm will reduce computational time, while still following the true gradient closely [169]. This new efficient algorithm is called $BPTT(h; D)$ which is the same as batch-wise BPTT when h is equal to D. If the batch size is too small, any temporal dependencies that span a longer period than the selected batch size cannot be learned, and if it is too large the algorithm converges slowly.

RTRL is a method which is of more theoretical interest as it easily generalizes [171]. To adapt the network using *sum of squared error* cost function, we need to be able to calculate

$$\frac{d\frac{1}{2}||e_k||^2}{dW} = -e_k^T \frac{dy_k}{dW} \tag{2.1.33}$$

where

$$\frac{dy_k}{dW} = \frac{\partial y_k}{\partial W} + \sum_{i=0}^{n} \frac{\partial y_k}{\partial x_{k-i}} \frac{dx_{k-i}}{dW} + \sum_{i=0}^{m} \frac{\partial y_k}{\partial y_{k-i}} \frac{dy_{k-i}}{dW} \tag{2.1.34}$$

where W is the network weight vector, e_k is the network error. The first term $\partial y_k/\partial W$ is the direct effect of a change in the weights on y_k, and is one of the Jacobians calculated by the dual-subroutine of the backpropagation algorithm. The second term is zero, since dx_k/dW is zero for all k. The final term can be broken up into two parts. The first part $(\partial y_k/\partial y_{k-i})$ is a component of the matrix $\partial y_k/\partial X$, since delayed versions of y_k are part of the networks input vector X. The dual-subroutine algorithm can be used to compute this term. The second part (dy_{k-i}/dW) is a previously calculated and stored value of dy_k/dW . Initially, dy_i/dW are set to zero for $i = 0, -1, -2, ...$ and then the rest of the terms are calculated recursively from that point on.

The dual-subroutine procedure naturally calculates the Jacobians in such a way that the weight update is done with simple matrix multiplication;

$$(d_w y)_k \triangleq \left[\left(\frac{dy_{k-1}}{dW}\right)^T \left(\frac{dy_{k-2}}{dW}\right)^T \cdots \left(\frac{dy_{k-m}}{dW}\right)^T \right]^T \tag{2.1.35}$$

$$(d_x y)_k \triangleq \left[\left(\frac{\partial y_k}{\partial y_{k-1}}\right) \left(\frac{\partial y_k}{\partial y_{k-2}}\right) \cdots \left(\frac{\partial y_k}{\partial y_{k-m}}\right) \right]^T \tag{2.1.36}$$

The second equation obtained above is simply the columns of $\frac{\partial y_k}{\partial X}$ corresponding to the feedback inputs to the network, and is directly calculated by the dual subroutine. Then, the weight update is calculated as

$$\Delta W_k = \left(\eta e_k^T \left[\frac{\partial y_k}{\partial W} + (d_x y)_k (d_w y)_k \right] \right)^T \tag{2.1.37}$$

It should be noted that if the learning rate (η) is small enough, the weights of the adaptive filter converge in a stable way.

2.1.3 Data Normalization

To obtain a better result from the neural network, data preprocessing is an important step that has to be taken before we proceed with training the network. We transform the data by normalizing them to overcome the large differences between the minimum and the maximum of the input data. Different methods exist such as Z-score normalization and min-max normalization. We also would like to modify the input data to fall within the range of an activation function, therefore we can normalize, standardize, or rescale the input data, using mean (μ), standard deviation (std) and the minimum and maximum range R_{min} and R_{max}.

To normalize the input data to $\mu = 0$ and $std = 1$, we calculate

$$\mu = \frac{\sum_{i=1}^{N} x_i}{N} \tag{2.1.38}$$

$$std = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \mu)^2}{N}} \tag{2.1.39}$$

and perform the standardization of the input data as $\tilde{x}_i = (x_i - \mu)/std$. This gives us a standard set of data with the mean close to zero and a standard deviation close to one. This statistical normalization method reduces the effect of outliers on the data. To translate the data into midrange 0 and standardize to range R, we perform

$$Z = \frac{max_i\{x_i\} + min_i\{x_i\}}{R} \tag{2.1.40}$$

$$S_x = max_i\{x_i\} + min_i\{x_i\} \tag{2.1.41}$$

$$x_i^n = \frac{x_i - Z}{S_x/R} \tag{2.1.42}$$

Data normalisation is the first step before training process as it has effects on training, but it is also important to de-normalize the network output in the analysis process.

2.2 Gas Turbine Jet Engine Mathematical Model

In this section the mathematical model of a jet engine will be descried. It is true that this research is based on data driven models but as the data from real flight missions were not available, the data generated by a single spool gas turbine engine model developed in MATLAB/SIMULINK is used and analyzed. In the following a description on gas turbine engines and the mathematical model used to develop the SIMULINK model will be presented.

A gas turbine engine is a kind of internal combustion engine which is mainly consisting of an upstream rotating compressor, which is coupled to a downstream

turbine through a combustion chamber. They are called gas turbine engines because air is mixed with fuel and is ignited in the combustor. This product is then transferred to the turbine. The gas which enters the turbine has high volume and high velocity and hence the turbine blades start spinning and this as a result powers the compressor. This produced energy can be used in form of thrust, compressed air, shaft power or a combination of all and can be used for aircraft, ships, trains or generators.

The type of the gas turbine engine that we study is used for jet engines which produce thrust form the exhaust gas. Aircraft gas turbines are categorized into turbojets and turbofans [172]. Turbojets use the direct impulse of the exhaust gasses where on the other hand turbo fans generate thrust with the addition of a ducted fan. The schematic of a gas turbine jet engine is depicted in Figure 2.13:

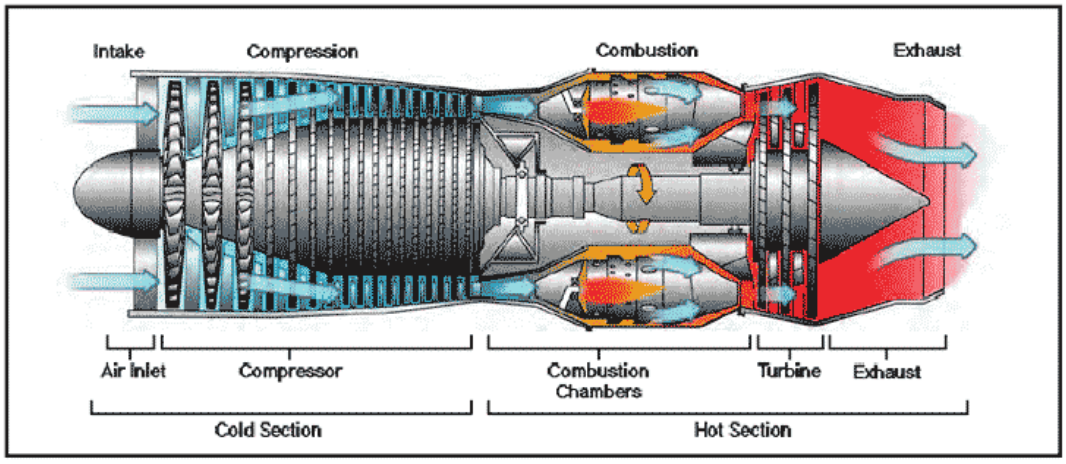

Figure 2.13: Schematic of a gas turbine jet engine [8].

Single shaft engines are considered in this thesis. They have a simple configuration which consists of a single shaft, and a high pressure compressor driven by a single turbine unit. As pointed out earlier a SIMULINK model is developed. Based on the available literature on modelling a nonlinear dynamics of a jet engine [9, 97] thermodynamics, aerodynamic and mechanical relationships of the components are used to develop the model as described below.

The thermodynamic nonlinear model describes the behaviour of the engine. It has two major steps. In the first step steady-state equations are considered and then in the second step dynamic equations are studied. Steady-state equations represent the components when working in the equilibrium point where dynamic equations consider the jet engine to be working in the transient mode. Here we are studying an engine which is working in the take-off mode. Other modes are namely, climb, cruise, descent and landing. For the transient response three dynamics are recognized i.e. rotor dynamics, volume dynamics and heat transfer dynamics.

The acceleration or deceleration of the rotor shaft (rotor dynamics) that connects the high pressure compressor and the turbine is caused by the power imbalance between the turbine and the compressor. Dynamical changes in pressure of the volumes (volume dynamics) is caused by the mass flow imbalance among different engine components.

When the engine is working in the transient mode, thermal energy is exchanged among different engine components which causes heat transfer dynamics. The effect of heat transfer dynamics on the behaviour of the engine is negligible when compared to the two other existing dynamics and thus they are not considered in this work since in this work we are concerned with a commercial single spool jet engine at normal operating conditions and we will present rotor dynamics and volume dynamics briefly where more detail can be found in [173].

2.2.1 Rotor Dynamics

The following differential equation can be derived using the concept of energy balance between the shaft and the compressor:

$$\frac{dE}{dt} = \eta_{mech} W_T - W_C \tag{2.2.1}$$

where $E = \frac{J(\frac{N.2\pi}{60})^2}{2}$, η_{mech} denotes the mechanical efficiency, W_T denotes the power generated by the turbine, W_C denotes the power consumed by the compressor and J is the rotor moment of inertia. N stands for the number of turns which is a function of time (RPM).

2.2.2 Volume Dynamics

In order to take into account 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 [174]. This assumption results in a considerable elimination of algebraic loops and makes it possible to develop a generic model based on the jet engine dynamics. Besides, we assume that the gas has zero speed and has homogeneous properties over volume. The following equation describes the volume dynamics:

$$\dot{P} = \frac{RT}{V}(\Sigma \dot{m_{in}} - \Sigma \dot{m_{out}}) \tag{2.2.2}$$

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

Now we will specify the gas turbine components' model that we use which are schematically depicted in Figure 2.13 .

2.2.3 Modelling of Engine Components

Intake duct

The position of the intake duct is before the compressor. It supplies the engine with the required airflow at the highest pressure possible. When the air reaches the compressor, its velocity decreases and on the other hand the air pressure and temperature increases. Assuming an adiabatic process, the engine inlet pressure ratio can be written as

$$\frac{p_d}{p_{amb}} = \left[1 + \eta_d \, \frac{\gamma - 1}{2} M^2\right] \frac{\gamma}{\gamma - 1} \tag{2.2.3}$$

where M denotes the Mach number and amb denotes the ambient condition for pressure and the temperature, η_d denotes the isentropic efficiency and γ stands for specific heat capacity ratio.

Compressor

The compressor in a gas turbine engine provides high-pressure air to the combustion chamber. The compressor is a quasi-steady component and for this model, its behaviour is determined by the compressor performance map. An engine simulation software known as GSP [175] (gas turbine simulation program) is used for obtaining the related maps. For a given pressure ratio (π_C) and the corrected rotational speed ($N/\sqrt{\theta}$), the corrected mass flow rate can be mapped by using a proper interpolation technique, where $\theta = \frac{T_i}{T_o}$ and $\delta = \frac{P_i}{P_o}$ *i.e.* $\dot{m}_C \frac{\sqrt{\theta}}{\delta} = f_{\dot{m}_C}(N/\sqrt{\theta}, \pi_C)$ and $\eta_C = f_{\eta_C}(N/\sqrt{\theta}, \pi_C)$. When these parameters are obtained, the compressor temperature rises and can be found from the following formula:

$$T_o = T_i \left[1 + \frac{1}{\eta_C} (\pi_C^{\frac{\gamma - 1}{\gamma}} - 1)\right] \tag{2.2.4}$$

where T_o denotes the output temperature, T_i denotes the input temperature and η_C is the isentropic efficiency of the compressor. The power consumed by the compressor can be calculated from:

$$W_C = \dot{m}_c c_p (T_o - T_i) \tag{2.2.5}$$

where $\dot{m}_c$ denotes the compressor mass flow rate and c_p denotes the specific heat at a constant pressure.

In single spool engines there is only one shaft in the system that connects the compressor and the turbine trough the combustion chamber. The speed of the engine is determined by the shaft (rotor) speed. The speed is a function of the power generated by the turbine and the total moment of inertia of the rotary system. The relation between the power consumed by the compressor and the speed of the shaft is given as follows:

$$W_c = \frac{J(2\pi N)^2}{2} \tag{2.2.6}$$

where J denotes the moment of inertia of the shaft and N is the rotor speed expressed in RPS (revolution per second).

Combustion Chamber

In the combustion chamber the fuel is mixed with the high pressure air coming from the compressor. The mixture is ignited and as a result of the burning fuel, the temperature is raised. It is desirable to keep the pressure of the gas unchanged in the combustion chamber. The combustion chamber represents both the energy accumulation and the volume dynamics between the compressor and the turbine at the same time. Combustion chamber inside pressure and temperature are derived as follows

$$\dot{P}_{CC} = \frac{P_{CC}}{T_{CC}}\dot{T}_{CC} + \frac{\gamma R T_{CC}}{V_{CC}}(\dot{m}_C + \dot{m}_f + \dot{m}_T) \tag{2.2.7}$$

$$\dot{T}_{CC} = \frac{1}{c_v m_{CC}}\left[(c_p T_C \dot{m}_C + \eta_{CC} H_u \dot{m}_f - c_p T_{CC}\dot{m}_T) - c_v T_{CC}(\dot{m}_C + \dot{m}_f + \dot{m}_T)\right] \tag{2.2.8}$$

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

Turbine

Turbine's function in a jet engine is to extract a portion of the kinetic energy from the high-temperature combustion gasses for deriving the compressor and accessories. In a typical jet engine three quarter of the internally produced power is consumed for deriving the compressor and the remaining power will be used to generate the thrust. Turbine is a rotatory component through which gas with high temperature passes. Similar to the compressor, the behaviour of a turbine is also represented by using the turbine performance map. The maps used in this work are derived from the GSP software [175].

For a given pressure ratio (π_T) and the corrected rotational speed ($N/\sqrt{\theta}$), the corrected mass flow rate $\dot{m}_T\sqrt{\theta/\delta}$ and the efficiency (η_T) are obtained from the performance map *i.e.* $\dot{m}_T\dfrac{\sqrt{\theta}}{\delta} = f_{\dot{m}_T}(N/\sqrt{\theta}, \pi_T)$ and $\eta_T = f_{\eta_T}(N/\sqrt{\theta}, \pi_T)$. The temperature drop and the turbine mechanical power which is proportional to the temperature

decrease in the turbine can be found from the following formulae:

$$T_o = T_i\left[1 - \eta_T(1 - \pi_T^{\frac{\gamma-1}{\gamma}})\right] \tag{2.2.9}$$

$$W_T = \dot{m_T} c_p (T_i - T_o) \tag{2.2.10}$$

where $\dot{m_T}$ denotes the compressor mass flow rate and c_p denotes the specific heat at a constant pressure. In a jet engine the power generated by the turbine is proportional to the power consumed by the compressor.

Nozzle

Nozzle is the last component of a jet engine. The working fluid is expanded in the nozzle to produce a high-velocity jet. The high pressure exhaust gas is accelerated in a jet pipe which is located between the turbine outlet and the nozzle throat to become close to the ambient pressure and as a result produces thrust. The nozzle exit temperature is given by:

$$T_{n_i} - T_{n_o} = \eta_n T_{n_o}\left[1 - (\frac{1}{P_{n_i}/P_{amb}})^{\frac{\gamma-1}{\gamma}}\right] \tag{2.2.11}$$

where η_n is the isentropic efficiency of the nozzle.

The mass flow rate of the nozzle is computed as follows: if condition (2.2.12) holds, the mass flow rate is derived from equation (2.2.13), otherwise it is obtained from equation (2.2.14). In other words, if

$$\frac{P_{amb}}{P_{n_i}} \langle \left[1 + \frac{1-\gamma}{\eta_n(1+\gamma)}\right]^{\frac{\gamma}{\gamma-1}} \tag{2.2.12}$$

then,

$$\frac{\dot{m}_n\sqrt{T_{n_i}}}{P_{n_i}} = \frac{u}{\sqrt{T_{n_i}}}\frac{A_n}{R}\frac{P_{amb}}{P_{n_i}}\frac{T_{n_i}}{T_{n_o}} \tag{2.2.13}$$

where $\dfrac{u}{\sqrt{T_{n_i}}} = \sqrt{2c_p\eta_n\left(1-(\dfrac{P_{amb}}{P_{n_i}})^{\dfrac{\gamma-1}{\gamma}}\right)}$, and $\dfrac{T_{n_o}}{T_{n_i}} = 1-\eta_n\left(1-(\dfrac{P_{amb}}{P_{n_i}})^{\dfrac{\gamma-1}{\gamma}}\right)$
otherwise

$$\frac{\dot{m}_n\sqrt{T_{n_i}}}{P_{n_i}} = \frac{u}{\sqrt{T_{n_i}}}\frac{A_n}{R}\frac{P_{crit}}{P_{n_i}}\frac{T_{n_i}}{T_{crit}} \tag{2.2.14}$$

where $\dfrac{u}{\sqrt{T_{n_i}}} = \dfrac{2\gamma R}{\gamma+1}$, $\dfrac{P_{crit}}{P_{n_i}} = \left(1-\dfrac{1}{\eta_n}(\dfrac{\gamma-1}{\gamma+1})\right)^{\dfrac{\gamma}{\gamma-1}}$ and $\dfrac{T_{crit}}{T_{n_i}} = \dfrac{2}{\gamma+1}$. It is assumed that $P_{n_i=P_{LT}}$ and $T_{n_i} = T_M$, which is obtained from the energy balance relation in the mixer as follows:

$$T_M = \frac{\dot{m}_T T_T + \beta\dot{m}_C T_C}{\dot{m}_T + \beta\dot{m}_C} \tag{2.2.15}$$

where β is the bypass ratio.

Set of Nonlinear Equations

In this part a set of nonlinear equations related to a single spool jet engine will be presented. In the engine intakes, the temperature and the pressure can be obtained as follows if we assume adiabatic process;

$$\frac{T_d}{T_amb} = 1 + \frac{\gamma-1}{2}M^2 \tag{2.2.16}$$

and

$$\frac{P_d}{P_{amb}} = \left[1+\eta_d\frac{\gamma-1}{2}M^2\right]^{\frac{\gamma}{\gamma-1}} \tag{2.2.17}$$

Again it must be noted that the commercial software GSP ([176]) is used for the purposes of conducting model validation studies. To summarize, the set of nonlinear equations corresponding to a single spool jet engine is given by

$$\dot{T}_{CC} = \frac{1}{c_v m_{CC}} \Big[(c_p T_C \dot{m}_C + \eta_{CC} H_u \dot{m}_f - c_p T_{CC} \dot{m}_T) - c_v T_{CC} (\dot{m}_C + \dot{m}_f + \dot{m}_T) \Big] \quad (2.2.18)$$

$$\dot{N} = \frac{\eta_{mech} \dot{m}_T c_P (T_{CC} - T_T) - \dot{m}_C c_P (T_C - T_D)}{JN(\frac{\pi}{30})^2} \quad (2.2.19)$$

where J denotes the inertia of the shaft connecting the compressor to the turbine, η_{mech} denotes the mechanical efficiency and is a rating that shows how much of the power developed by the expansion of the gasses is actually delivered as useful power.

$$\dot{P}_T = \frac{RT_M}{V_M} \big(\dot{m}_T + \frac{\beta}{1+\beta} \dot{m}_C - \dot{m}_n \big) \quad (2.2.20)$$

$$\dot{P}_{CC} = \frac{P_{CC}}{T_{CC}} \dot{T}_{CC} + \frac{\gamma R T_{CC}}{V_{CC}} (\dot{m}_C + \dot{m}_f + \dot{m}_T) \quad (2.2.21)$$

Control Input

One of the mechanisms used for controlling the parameters of the engine is the mass flow rate of the main fuel $\dot{m}_f$. The fuel mass flow rate itself is a function of the power level angle (PLA) which is set by the pilot. The following dynamics for the fuel mass flow rate is considered

$$\tau \frac{\dot{m}_f}{dt} + \dot{m}_f = G u_{fd} \quad (2.2.22)$$

where τ is the time constant, G is the gain associated with the fuel valve, and u_{fd} denotes the fuel demand, which is computed by using a feedback from the rotational

speed as described in [174]. As stated earlier the performance maps of the compressor and the turbine are adopted from the commercial software package GSP [175].

Simulating the above-mentioned jet engine nonlinear dynamics, a SIMULINK model is developed in [9] and is used in this research as well. The initial model was designed to simulate the aircraft cruise mode and for the purpose of this research work it has been transformed to work in the take-off mode and the necessary changes have been made. Figure 2.14 shows the information flow process in the developed SIMULINK model of the engine [9]. In equation (2.15) the series of steady states

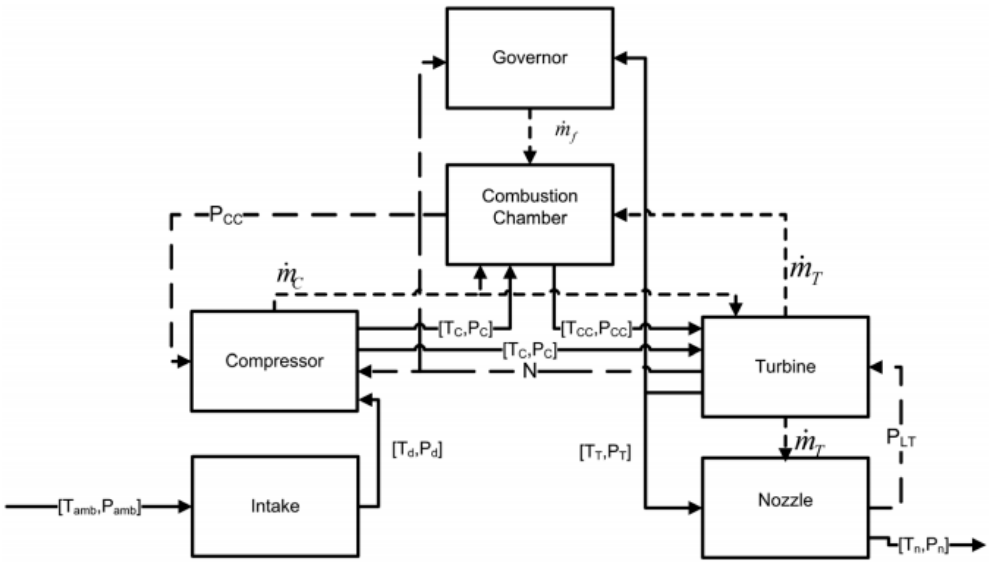

Figure 2.14: Information flow diagram in a modular modeling of the jet engine dynamics [9].

that are obtained from the developed nonlinear model for a single spool jet engine and the GSP at PLAs ranging from 0.4 to 1 are depicted. At each point, the initial condition of the PLA is set equal to 0.3 followed by a transient to reach to the steady state corresponding to the desired PLA.

As can be seen from Figure 2.15, the responses corresponding to the SIMULINK model and the GSP match each other within an acceptable error range (below 5%). Considering this model, seven different measurements can be obtained from the single

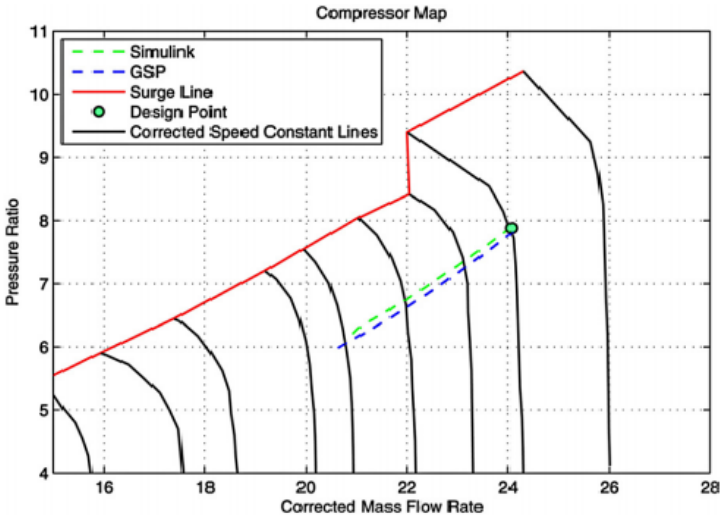

Figure 2.15: Compressor performance maps [9].

spool engine three major components. These measurements are denoted by P_C, T_C, P_{CC}, T_{CC}, P_T, T_T and N, which represent the compressor pressure, the compressor temperature, the combustion chamber pressure, the combustion chamber temperature, the turbine pressure, the turbine temperature and the spool speed.

A flight cycle (profile) can be categorized into different phases or stages namely, taxi, take-off, climb, cruise, descent and consequently according to each of these phases the engine will go through different operating conditions such as starting, idle thrust, acceleration, deceleration, cruise thrust, shut-down, etc. A sample flight envelope is depicted in Figure 2.16. In this thesis we are addressing the problem of degradation growth trend analysis and prediction when the engine is experiencing the take-off mode and this is the time when the maximum fuel is applied and the temperature is in its highest value and as we will see later, because the severity of the degradation is a function of the temperature and the variations in the temperature, the take-off mode is the most suitable mode for the prognostics study.

As explained in Chapter 1, when the engine goes trough aging or degradation

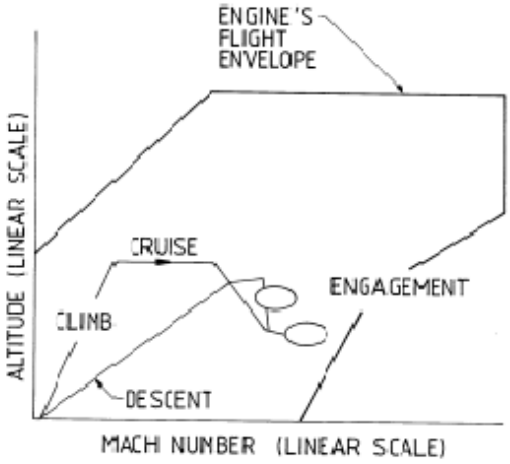

Figure 2.16: A typical duty cycle [10].

process, the engine health parameters are affected. The engine health parameters are considered to be the efficiency (η) and the mass flow rate ($\dot{m}$) of the components such as the compressor and the turbine. These changes as a result of the influence of the above measurements will help us to detect, track and predict the degradations occurring in the jet engine.

2.3 Gas Turbine Simulation Program (GSP)

The Gas turbine Simulation Program GSP [175] by National Aerospace Laboratory (NLR) was developed with flexibility as a primary objective and has successfully demonstrated the capability to model virtually any gas turbine configuration. GSP has a GUI with separate data entry windows for individual component models and a drag& drop interface. Different measurements according to the users' need can be displayed.

GSP is an off-line component-based modelling environment for both aircraft and

industrial gas turbines. Both transient and steady state simulation of any kind of gas turbine configuration can be achieved by establishing a specific arrangement of engine component models. GSP is a powerful tool for performance prediction, emission calculation, control system design, diagnostics and off-design analysis. It is especially suitable for sensitivity analysis of some variables such as ambient conditions, component deterioration and exhaust gas emissions. The type of the engine and the components used can also be customized by the user. A sample customized single-shaft jet engine in the GSP environment is depicted in Figure 2.17. More information on this software can be found in the GSP11 user manual.

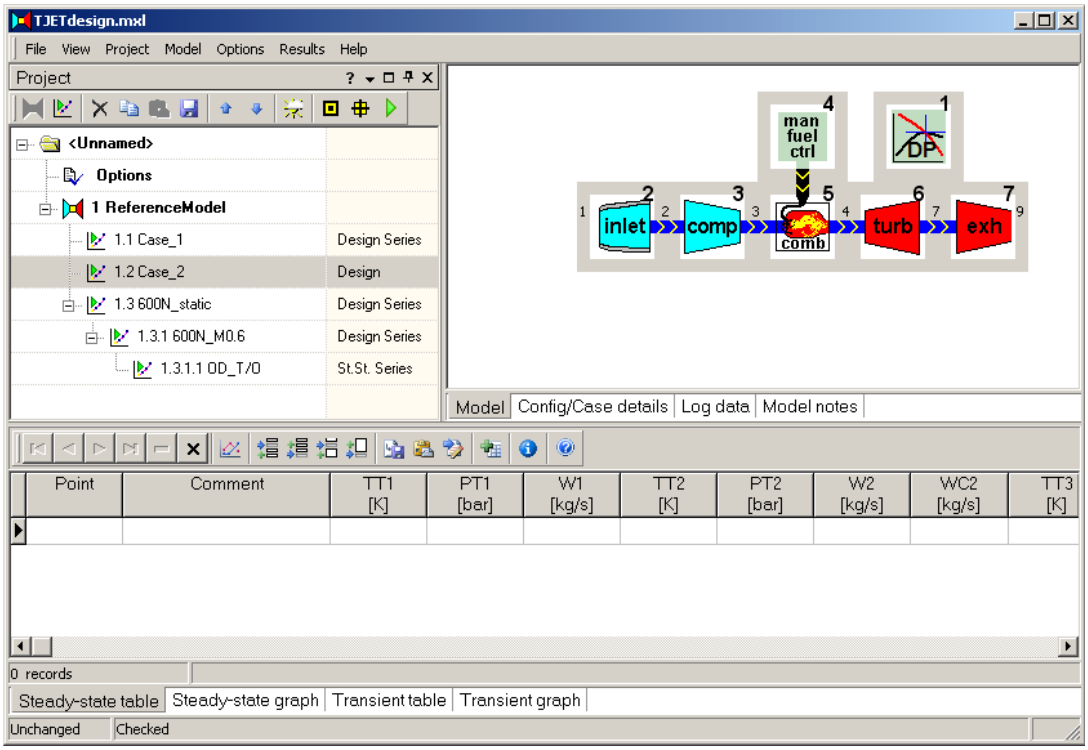

Figure 2.17: Single shaft jet engine in the GSP environment [11].

During this study two versions of the GSP were available namely GSP10 and GSP11. We have used the latest version for validating our model and the data derived from the developed model will be demonstrated in the following chapter, however in some cases in this chapter we have referred to GSP10 when citing some previous work

about the engine SIMULINK model which has been done previously. In the GSP11 version usability, productivity, data integrity, and security as well as some GUI have been improved. Other than these there are no significant changes in the two models and a model in the old version can be imported to the new version, and the version change may not affect the results.

2.4 Engine Performance Monitoring and the Problem of Trend Analysis

Engine performance monitoring involves monitoring engine operating parameters closely to determine when an engine is operating off nominal and expected condition. Some operating parameters such as the inter-turbine temperature (ITT) are monitored for deviations from the manufacturer's specified baseline [177]. The deviations from these baselines are then plotted as a function of time to form "trend plots". Significant deviations from a reference line on these trend plots are used to trigger the maintenance action [178]. This sort of trend analysis system must be accurate enough to indicate trend in engine parameters in short term. In performance scheduling we are interested in finding to what degree the performance degradation has occurred in a specific component. Turbine temperature is a good indicator of the overall system health and hence they have been monitored and studied in this research too.

Finally, it is worth to point out that there are different causes of airplane crashes such as pilot error, mechanical error, weather and other human errors. Besides, the crash can happen in any operating mode of the flight such as take-off, climb, cruise, descent and approach. Among these phases, the probability of the failure is higher in the take-off mode (30%) as can be seen in Figures 2.18 and 2.19. The statistics are valid between the years 1995- 2010 [12]. Take-off mode data are analyzed in this

thesis.

Cause	1950s	1960s	1970s	1980s	1990s	2000s	All
Pilot Error	41	34	24	26	27	30	29
Pilot Error	10	17	14	18	19	19	16
Pilot Error	6	5	5	2	5	5	5
Total Pilot Error	57	56	43	46	51	54	50
Other Human Error	2	9	9	6	9	5	70
Weather	16	9	14	14	10	8	12
Mechanical Failure	21	19	20	20	10	24	22
Sabotage	5	5	13	13	11	9	9
Other Cause	0	2	1	1	1	0	1

Figure 2.18: Causes of fatal accidents by decades [12].

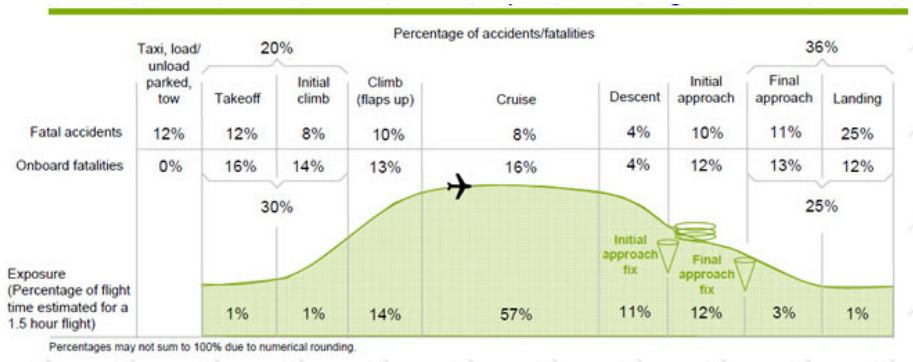

Figure 2.19: Accidents and fatalities by phase of flight [12].

In the following two figures the plane accidents are categorized if they are happening because of the engine failure or in the take-off mode. In Figure 2.20 the accidents are divided into fatal and non-fatal accidents. In Figure 2.21 the accidents because of the engine failure are categorised for single and dual spool engines. As can be seen the rate is higher in the single spool engines which is normal as they only depend on one engine during the entire working period. This can shed more light on the importance of studying single spool jet engines prognostics. The bigger the engines, the harder they fail.

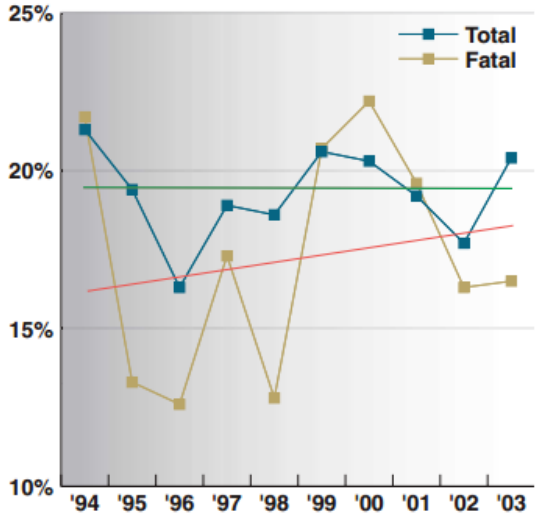

Figure 2.20: Take-off/climb accidents [13].

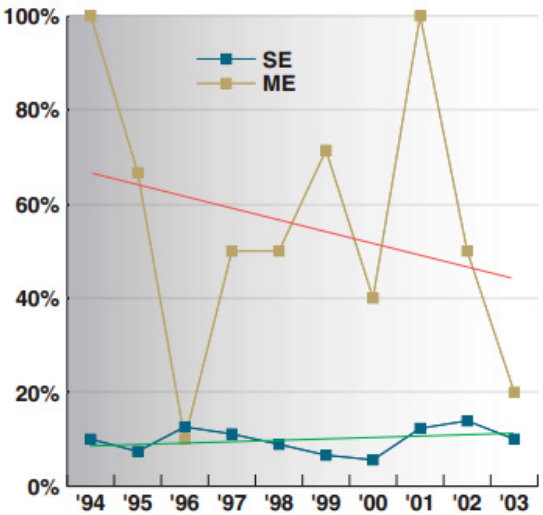

Figure 2.21: Engine failure accidents [13].

2.5 Conclusion

In this chapter some background information related to the work being done in this thesis were presented. First an introduction was given on neural networks methods and principals by focusing on the three neural network architectures that are used in this thesis. Firstly, artificial recurrent neural networks (RNN) were presented with different possible architectures and the learning algorithms. Next, time-delay neural network principles as a type of dynamical neural network were discussed and some research work in this area were discussed. Finally, nonlinear autoregressive neural networks (NARNN) were introduced as a type of recurrent dynamic neural networks architecture and some training methods were briefly studied.

In the next part, basic definitions for the gas turbine engine, different types and different components were given. Moreover, nonlinear mathematical equations for a single spool jet engine were provided which are to be used for our fault prognosis schemes.

Finally, in the last part of this chapter, we introduced GSP software which will be used for data validation in the next chapter and we finished the chapter by introducing the problem of engine performance monitoring and the problem of trend analysis and shed some light on the importance of these subjects by presenting some related statistics.

Chapter 3

Degradation Modelling and Validation

In the following chapter we will introduce and investigate the main engine degradations namely fouling, erosion and corrosion. We will define causes of jet engine degradations, show how their model are considered and integrated with the SIMULINK model described in Chapter 2 for a single spool gas turbine engine. In addition, the effects of these degradations are studied on the gas path measurements and the engine output for different levels of degradations in the take-off mode of the flight. The obtained simulation results are compared with the ones from the GSP software for data validation. These results are also presented in the following reference [179].

3.1 Degradation Model Description

In this section, the main causes of the jet engine degradation will be presented and modelled. They are integrated to the SIMULINK model developed for the single spool engine which was described in the previous chapter. We will consider the take-off mode of the flight in which the effects and growth of the degradations are more

significant than any other phase that the engine goes through in its mission profile. The results obtained from this model will be analyzed to predict the engine condition for multi-flights ahead. In practice, a jet engine like any other physical system goes through degradations such as wear and tear, where the degradation causes need to be known and future states need to be predicted as they adversely affect engine's overall performance such as specific thrust, fuel consumption, spool speed, turbine entry temperature, etc . It is desirable to understand the processes leading to each individual component's degradation. In the following we will introduce and model major causes of engine degradation, their root causes and some actions performed to remove them. If the degradations are not removed from the system they can lead to serious damages. Greater damages can adversely result in a shorter usage life of an engine.

Several mechanisms cause the degradation of an engine namely abrasion, corrosion, erosion, flaking, fouling, galling, pitting, stalling, thermal distortion, foreign object damage (FOD), etc. [172]. Quantifying performance degradations is difficult as they have a slower dynamics when compared to faults. Accumulation of these degradations can lead to different damages in the engine such as creep, low-cycle fatigue, high-cycle fatigue, oxidation, sulphidation, thermal fatigue etc. [10], and finally resulting in fatigue failure or other types of failure. In a gas turbine engine many components are subject to deterioration, however only a few of them have a significant impact on the engine behaviour. These are the rotating components that are subject to cyclic and steady-state stresses. Turbine blades are important components because they are under both highest rotating speed and gas temperature. Causes of degradation can be studied under the following categories [180]:

- Flight loads
- Thermal distortion
- Engine fouling due to deposits within the engine

- Erosion of airfoils
- In-service damage and abuse
- The type of engine operation or duty cycle implemented
- The maintenance practices employed for the engine

Performance deterioration can be classified into recoverable and non-recoverable (permanent) deterioration [181, 182]:

Recoverable deterioration: Is the performance loss that can be recovered by operational procedures such as keeping the inlet and outlet pressures low, or water washing of the compressor.

Non-recoverable (permanent) deterioration: Is the performance loss that cannot be recovered without repair or replacement of affected gas turbine components. Examples of non-recoverable degradations include: loss of surface finish on blades, increase in blade tip clearances, packing leakage of both the compressor, the turbine, and combustion system component corrosion/erosion leading to flame instabilities or increased thermal stress on the turbine.

The goal of maintenance is to minimize risk of breakdown and improve reliability. It can also repair a system and/or upgrade performance. The longer a system runs, the more performance degradation increases and the probability of breakdown and interruption of production increase [183]. Preventive maintenance protects the system from unnecessary degradation, and helps restore system performance and reliability. For a gas turbine unit the degradation is largely due to the degradation of components and depends on the unit's history.

The rate at which a turbine degrades will depend on how it has been operated, especially during start-up when the engine experiences the most severe hot end thermal gradients. Oxidation and corrosion is most severe at this time and add significantly

to the engine aging process. The more often an engine is subject to start up, possibly due to emergency trips, the more rapidly it will degrade. Also if the turbine is operated for long periods at its peak rating this will degrade performance more rapidly than an engine operating at or below base load rating. While some of these effects can be reversed by cleaning or washing the engine, others require the adjustment, repair or replacement of components. Economic considerations play an important role in determining the optimal frequency at which a gas turbine is, for example, water washed.

In this thesis we consider the effect of fouling and erosion/corrosion which have the most significant effects on the engine performance as compared to other causes. They can initiate and/or accelerate different damages. These problems are difficult to detect when the engine is in operation. In some cases, the effects of the engine degradation can be detected when the engine decreases the output power or increases the fuel consumption

Fouling

Fouling is caused by the adherence of particles to airfoils and annulus surfaces. The adherence is caused by different substances such as water, oil, carbon, smoke and salt. Fouling can be formed in different parts of the air path (stators, blades and guide vanes) and it can affect the aerodynamic behaviour of the system and changes the shape of the airfoil. The consequences of fouling in the jet engine are loss of efficiency, reduction in the power, and increase in the fuel consumption. In extreme cases fouling can also result in surge problem.

Fouling phenomena mostly occurs in the compressor. Fouling degradation in the compressor causes significant efficiency loss, which incurs operational costs through

increased fuel usage or reduced power output. This mechanism decreases the compressor isentropic efficiency and have been demonstrated to affect the thermal efficiency and output power of the engine. The accumulation of dust reduces the tip clearance and increases the surface roughness [183] which as a result affects the compressor delivery pressure (CDP) and reduces the mass flow and can result in flow reductions up to 8%. Figure 3.1 shows typical performance degradations due to fouling.

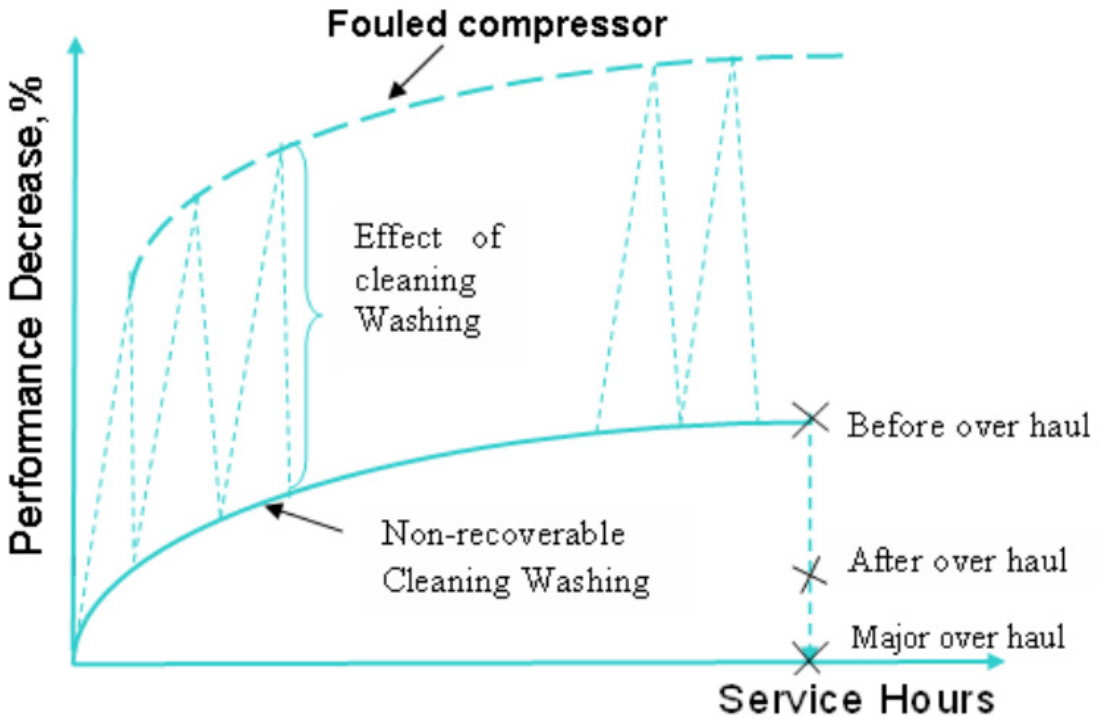

Figure 3.1: Typical performance degradations due to the fouling [14].

Compressor fouling is known as the source of about 70 - 85% of the performance degradation of gas turbine engines [184] . Although filtration system removes some of the particles, unfiltered particles enter into compressor and as a mixture of moisture and lubricants adhere to the blade surface. Severe fouling problems can be resolved by installing highly efficient filters at compressor inlet, but their use is restricted due to heavy cost and large pressure drop. Figure 3.2 depicts an example of a fouled

compressor blading. Different factors such as day temperature and humidity can

Figure 3.2: Fouled compressor blading with a mixture of oily deposits. [15].

affect the rate of fouling accumulation.

The main maintenance action to remove fouling effect from the compressor is washing the compressor. This off-line maintenance operation is generally achieved by three main methods. Figure 3.3 shows how washing can recover the compressor in terms of efficiency loss. Washes are clearly distinguished in the figure.

Fouling Degradation Modelling

Generally the effects of degradation due to fouling are as follows:

1 - Compressor component deterioration leads to performance and efficiency reduction

2 - Reduction in overall thermal efficiency

3 - Power output reduction caused by deterioration of the ingested air mass flow

4 - When constant power is required, degradation leads to increased fuel consumption and reduction in turbine blade creep life due to higher turbine entry temperatures

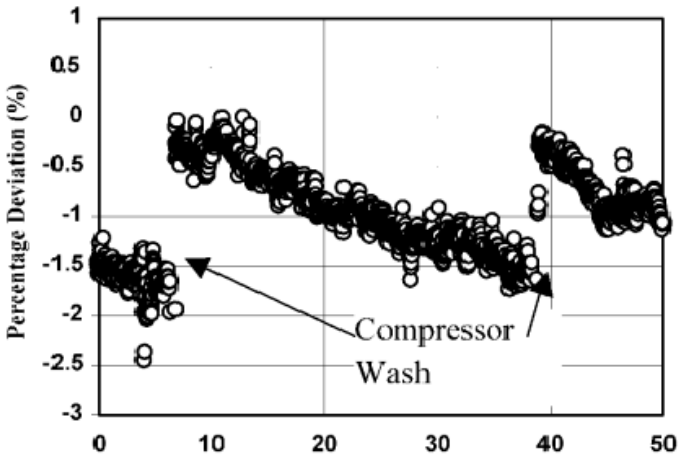

Figure 3.3: Example of the variation of a fouled compressor efficiency [11].

5 - Some changes will occur on the gas path measurements such as increase in the rotational speed and the turbine entry temperature (TET) to maintain the constant thrust.

6 - Due to all the above, economic losses will be incurred.

7 - With heavy fouling the likelihood of compressor surge and subsequent engine damage is increased.

8 - Unbalanced components due to corrosion and/or fouling could lead to unstable and critical engine operation and even vibration breakup.

All of the above factors together will cause a shorter remaining useful life of the engine [10]. To represent the fouling effects on the engine performance, a *fouling index* is used according to the work in [185]. This index is determined based on the reduction ratio of 1:2 for the compressor mass flow rate to the compressor efficiency and is denoted by FI. For example, if $FI = 1\%$ it implies a 0.5% reduction in the mass flow rate and a 1% reduction in the compressor efficiency. Fouling index is applied as a linear degradation process per cycle.

In order to model the fouling phenomena a degradation index is introduced (DI)

(which is a hypothetical parameter) and its relationship to the mass flow capacity and efficiency of the compressor for fouling is set according to equations (3.1.1) and (3.1.2), where $Flow\_\ Index$ denotes the flow capacity index and $Efficiency\_\ Index$ denotes the efficiency index, that is

$$Flow\_\ Index = 0.5 * DI \tag{3.1.1}$$

$$Efficiency\_\ Index = DI \tag{3.1.2}$$

The effects of the degradation are considered to be linear on the health parameters of the system. The resulting change slopes in each degradation phenomena are summarized in equations (3.1.3) to (3.1.6) for the fouling :

Rate of change in the mass flow capacity of the compressor ($mc\_\ slope$):

$$mc\_\ slope = Flow\_\ Index/(100 * Fouling\_\ cycles) \tag{3.1.3}$$

Degradation gain in each cycle for the compressor mass flow ($mc\_\ deg\_\ gain$):

$$mc\_\ deg\_\ gain = 1 - mc\_\ slope * i \tag{3.1.4}$$

Rate of change in efficiency of the compressor ($etac\_\ slope$):

$$etac\_\ slope = Efficiency\_\ Index/(100 * Fouling\_\ cycles) \tag{3.1.5}$$

Degradation gain in each cycle for the compressor efficiency ($etac\_\ deg\_\ gain$):

$$etac\_\ deg\_\ gain = 1 - etac\_\ slope * i \tag{3.1.6}$$

where i denotes the cycle number and $Fouling\_cycles$ denotes the total number of

cycles after which compressor fouling will be completed with an index DI.

Corrosion

Another major cause of the aero engine degradation is the corrosion. Corrosion is the loss of material from flow path components and is caused by chemical reactions between the components and certain contaminants such as mineral acids, salts or reactive gases [16]. Moreover, temperature oxidation is the chemical reaction between the components metal material and oxygen from the hot gases surrounding and entering the engine. Corrosion processes are always self propagating and will continue even if the source is removed.

Corrosion and pitting also cause aerodynamic changes in the blade behaviour, increasing the pressure losses and as a result, performance degradations or even blade failure. When corrosion is initiated, it cannot be stopped easily and will continue until it leads to failure [10]. Figure 3.4 shows an example of a corroded turbine rotor.

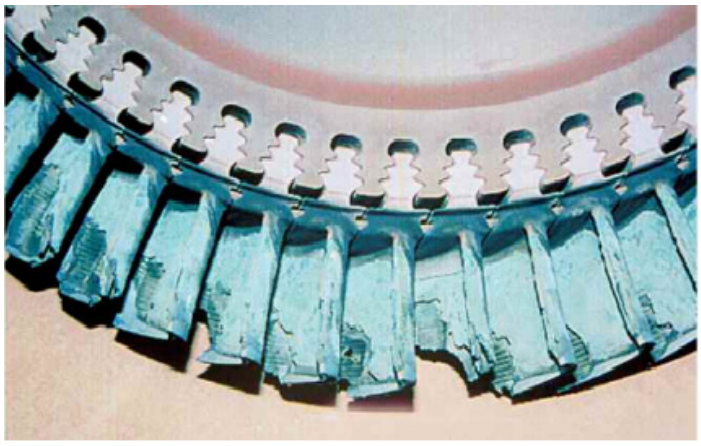

Figure 3.4: Corrosion in a turbine rotor [16].

Erosion

Erosion is another type of possible degradation in the gas turbine engine. Erosion is the removal of the material from the flow path by hard particles impinging on flow surfaces. These particles typically have to be larger than 20m in diameter to cause erosion by impact. Erosion is a problem for aero engine applications. Erosion changes airfoil shape, contours and surface finish and it contributes to 45% of the deterioration in modern turbofan engines [181].

This phenomenon will result in increase of the pressure losses, performance degradations and even blade failure. Erosion can reduce up to 5% of the performance in the compressor or the turbine and consequently the engine life [10]. Erosion is a type of non-recoverable deterioration. Typically, Erosion has the most significant effects on the blade leading edges. Erosion increases blade surface roughness, thus lowering slightly the efficiency. The effect of the erosion is studied on the turbine section and its blade for this thesis. Figure 3.5 depicts a severe erosion effect in the turbine blade which has resulted in a missing blade.

Figure 3.5: Missing blade due to erosion and burning [11].

To model the erosion we have to study their effect on the engine health parameters. As the health parameters are not always measurable, gas path measurements are

employed for trend analysis and prognostics in this work. To represent the erosion effects on the engine performance quantitatively, the Erosion Index is employed and is denoted by EI. This index is determined based on the ratio of the reduction of the turbine efficiency and increase in the turbine mass flow with the ratio of 1:2. For example, if EI=1% it implies a 0.5% increase in the mass flow rate and a 1% decrease in the turbine efficiency [10]. Similar to fouling, erosion index is also applied as a linear degradation process per cycle.

Different degradations can happen in different engine components such as airfoils, clearances, combustor, etc. . However, in this work we are considering the effect of fouling and erosion/corrosion on the compressor and the turbine as major engine components.

Erosion Degradation Modelling

As a consequence of the erosion (abrasive removal of hard particles), the nozzle throat cross-section will be wider and the mass flow rate will be increased. Moreover, the efficiency of the turbine will drop. Although with different root causes, erosion and corrosion have similar effects on the engine health parameters. Hence the corrosion will not be modelled separately. In order to model the erosion degradation phenomena a degradation index is introduced (DI) and its relationship to the mass flow capacity and efficiency of the turbine for the erosion is set according to equations (3.1.7) and (3.1.8), where $Flow\_\ Index$ denotes the flow capacity index and $Efficiency\_\ Index$ denotes the efficiency index, that is

$$Flow\_\ Index = 0.5 * DI \tag{3.1.7}$$

$$Efficiency\_\ Index = DI \tag{3.1.8}$$

The effects of the degradation, as pointed out previously, are considered to be linear on the health parameters of the system. The resulting change slopes in each degradation phenomena are summarized in equations (3.1.9) to (3.1.12) for the erosion.

Rate of change in the mass flow capacity of the turbine ($mt\_\ slope$):

$$mt\_\ slope = Flow\_\ Index/(100 * Erosion\_\ cycles) \tag{3.1.9}$$

Degradation gain in each cycle for the turbine mass flow ($mt\_\ deg\_\ gain$):

$$mt\_\ deg\_\ gain = 1 - mt\_\ slope * i \tag{3.1.10}$$

Rate of change in efficiency of the turbine ($etat\_\ slope$):

$$etat\_\ slope = Efficiency\_\ Index/(100 * Erosion\_\ cycles) \tag{3.1.11}$$

Degradation gain in each cycle for the turbine efficiency ($etat\_\ deg\_\ gain$):

$$etat\_\ deg\_\ gain = 1 - etac\_\ slope * i \tag{3.1.12}$$

where i denotes the cycle number and $Erosion\_cycles$ denotes the total number of cycles after which turbine erosion will be completed with an index DI.

It must be indicated that to maintain a constant maximum take-off thrust in the degraded engine during cycles of operation, fuel flow injection to the combustion chamber has to be increased to yield higher temperature in the turbine inlet. Therefore, the amount of increase in the fuel flow for each cycle is approximated by a second order polynomial as given by :

$$\Delta fuel\_\ flow = p_1 * i^2 + p_2 * i + p_3 \tag{3.1.13}$$

where p_1, p_2 and p_3 are set differently corresponding to each degradation index.

Thermal Distortion

The hot sections of the jet engine are more subject to thermal distortion, for example, combustor, turbine and propelling nozzles. These components are operating at high temperatures as well as high-varying stress environment. Creep and thermal fatigue (thermo-mechanical fatigue) are the most severe consequences of the thermal distortion [186]. The effect of the thermal distortion is considered in our model by introducing certain compensating coefficients.

Another important factor that may affect both the type of the engine deterioration and the rate of reduction in the engine performance is the engine duty cycle which is related to the flight mission profile. For example, in military applications, rapid throttle movements will cause unequal growths of hot end parts [187].

The developed model may be used as a test bench for studying failure prognostics of the system components when empirical data is not available. To validate the developed degradations caused by the fouling and the erosion, the GSP11 software [176] is used. This software is a versatile tool to study the behaviour of the jet engine. More information on the GSP software can be found in Chapter 2.

In this research we are considering component degradation. The effect of fouling on the compressor and erosion on the turbine blades are more significant, hence we are studying these two components and in the next section we will see how their measurements are affected by degradation.

3.2 Simulation Results

In order to study the effects of the degradation originated from fouling of the compressor or the erosion of the turbine on the engine measurable parameter which can be used later for prognostics, the dynamical model of a single spool engine that is developed in Matlab/Simulink [9] is used. The degradation model is integrated into the Simulink model. We have already described the nonlinear equations that allows us to derive such a model. The fouling and erosion effects on the system have been considered as changes in the engine health parameters, i.e. the efficiency and the mass flow rate.

To achieve a constant level of thrust for a given aircraft performance, the engine may run at higher speeds or higher turbine-entry temperatures. In our model, the strategy is to maintain the thrust constant by increasing the turbine-entry temperature through increase in the fuel consumption.

Simulations Under Different Degradation Levels

In this part we have conducted simulation scenarios to demonstrate how degradation changes the engine outputs such as spool speed and temperature and how they can adversely affect the engine performance. We have considered the fouling effect on the compressor and the erosion on the turbine separately. Moreover, the results related to the degradation modelling and the changes to the engine output parameters due to different degradation levels are compared with the GSP single spool turbojet model [175].

It is assumed that the flight mission takes 3000 seconds for each simulated cycle, from which in the first 20 seconds (the take-off mode) the engine operates from the ground idle to the maximum power. We will study and show the results associated with the take-off mode. All the simulations are performed in the standard operating

conditions. In other words, the ambient pressure and the ambient temperature are assumed to be at the standard sea level.

In order to obtain the same level of maximum take-off thrust, in presence of degradation the amount of the injected fuel to the combustion chamber is increased to maintain a constant level of thrust. The increased fuel follows the equation (3.1.13).

We have considered simulation scenarios where 100 take-off flights are simulated using 3 different severity levels of fouling index and 200 flight cycles are simulated for three (1%, 2% and 3%) erosion indices. For each scenario we will present the turbine temperature, compressor temperature, maximum spool speed and fuel consumption.

Scenario 1: Fouling Modelling

In the first scenario, the effects of the compressor fouling are modelled under three different fouling indices, namely 1%, 2% and 3% per 100 flight cycles. It is also possible for the user to change the fouling level of the compressor. It is also possible to run the simulations for other cycles which shows the versatility of our developed model. The results related to the fouling modelling are shown in Fig. 3.6. The fouling phenomenon, reduces the compressor efficiency and mass flow rate with a ratio of 1:2. It can be verified from Fig. 3.6 that the presence of the fouling in the engine causes an increase in the maximum output temperature of the compressor and the turbine and a decrease in the maximum value of the spool speed (for maintaining a constant maximum take-off thrust). Although the injected fuel level is being increased but because of the mass flow reduction, the maximum spool speed is not compensated and it drops. As the fouling index increases, these effects become more pronounced.

In the next step, we have conducted the same scenario in the GSP11 environment for a single spool engine and found the corresponding changes in the engine parameters. Per unit changes in the engine output parameters for the fouling scenarios from

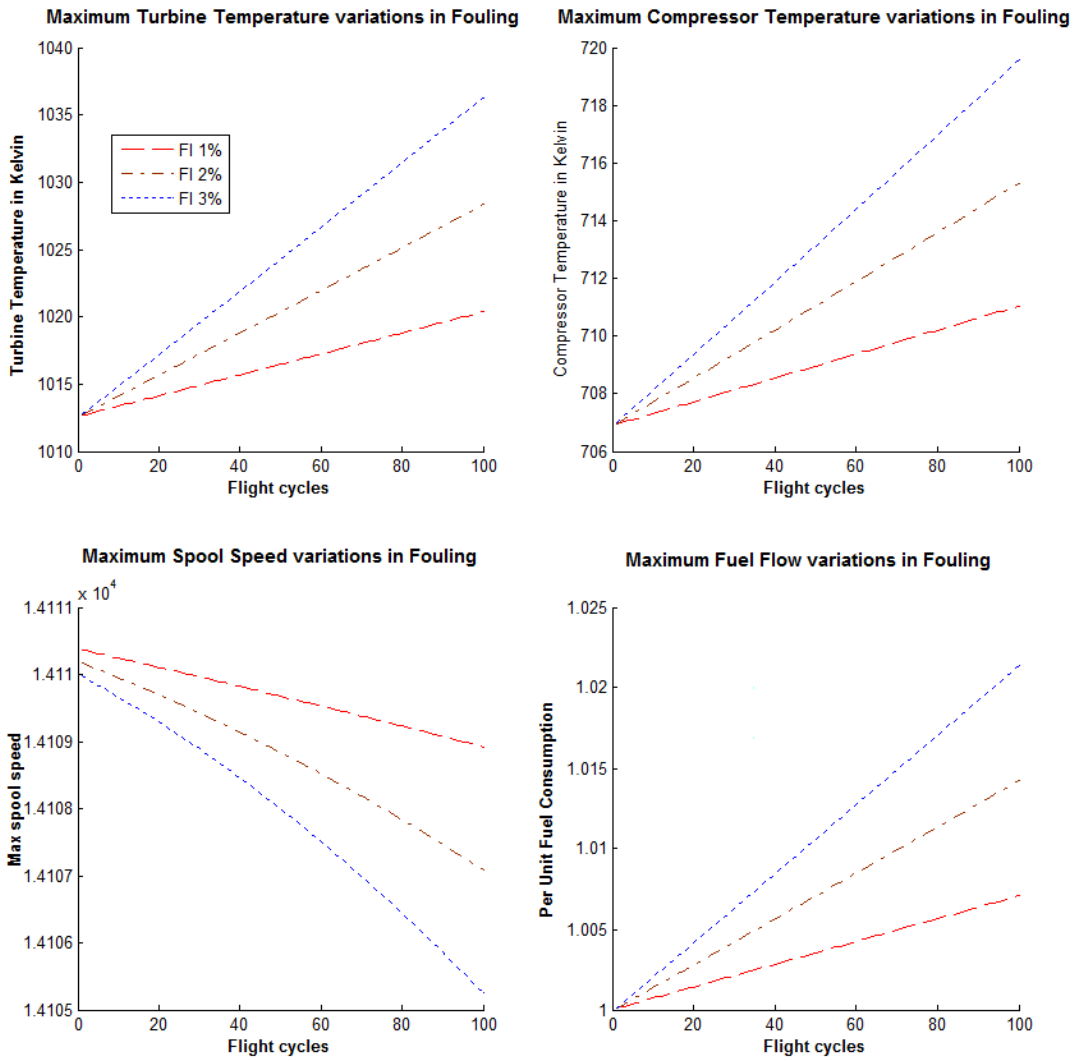

Figure 3.6: The outputs of the model corresponding to different levels of fouling degradation.

both the GSP and our developed model corresponding to the same fouling levels are summarized in Table 3.1. Although there are discrepancies between the results of the GSP and the developed model, the results from both models show the same trend. The reason for the differences is due to different fuel flow levels and compressor and turbine maps and different design points.

Table 3.1: Per unit changes in the engine parameters corresponding to the fouling scenarios.

Engine Parameter	Per Unit Change in Our Model	Per Unit Change in the GSP [176]
Fuel Flow Consumption in FI 1%	0.0699	0.01678
Fuel Flow Consumption in FI 2%	0.04855	0.02368
Fuel Flow Consumption in FI 3%	0.02127	0.03210
Compressor Temperature in FI 1%	0.05798	0.03817
Compressor Temperature in FI 2%	0.03287	0.08371
Compressor Temperature in FI 3%	0.01776	0.01187
Turbine Temperature in FI 1%	0.093	0.01151
Turbine Temperature in FI 2%	0.0652	0.02370
Turbine Temperature in FI 3%	0.02839	0.03255
Spool Speed in FI 1%	-0.007087	-0.01456
Spool Speed in FI 2%	-0.02325	-0.02795
Spool Speed in FI 3%	-0.03543	-0.04016

Scenario 2: Erosion Modelling

In the second scenario the effects of the erosion on the engine parameters are studied corresponding to the erosion indices 1%, 2% and 3% that are applied to the turbine of the model corresponding to 200 flight cycles. The related results are depicted in Fig. 3.7.

Because of the removal of material from the flow path, the efficiency of the turbine reduces however the nozzle throat will become wider which results in an increase in the turbine mass flow rate. The relation is assumed to be linear with a ratio of 1:2.

Erosion effects on the single spool engine increase the maximum turbine temperature while decrease the compressor output temperature. Furthermore, the spool

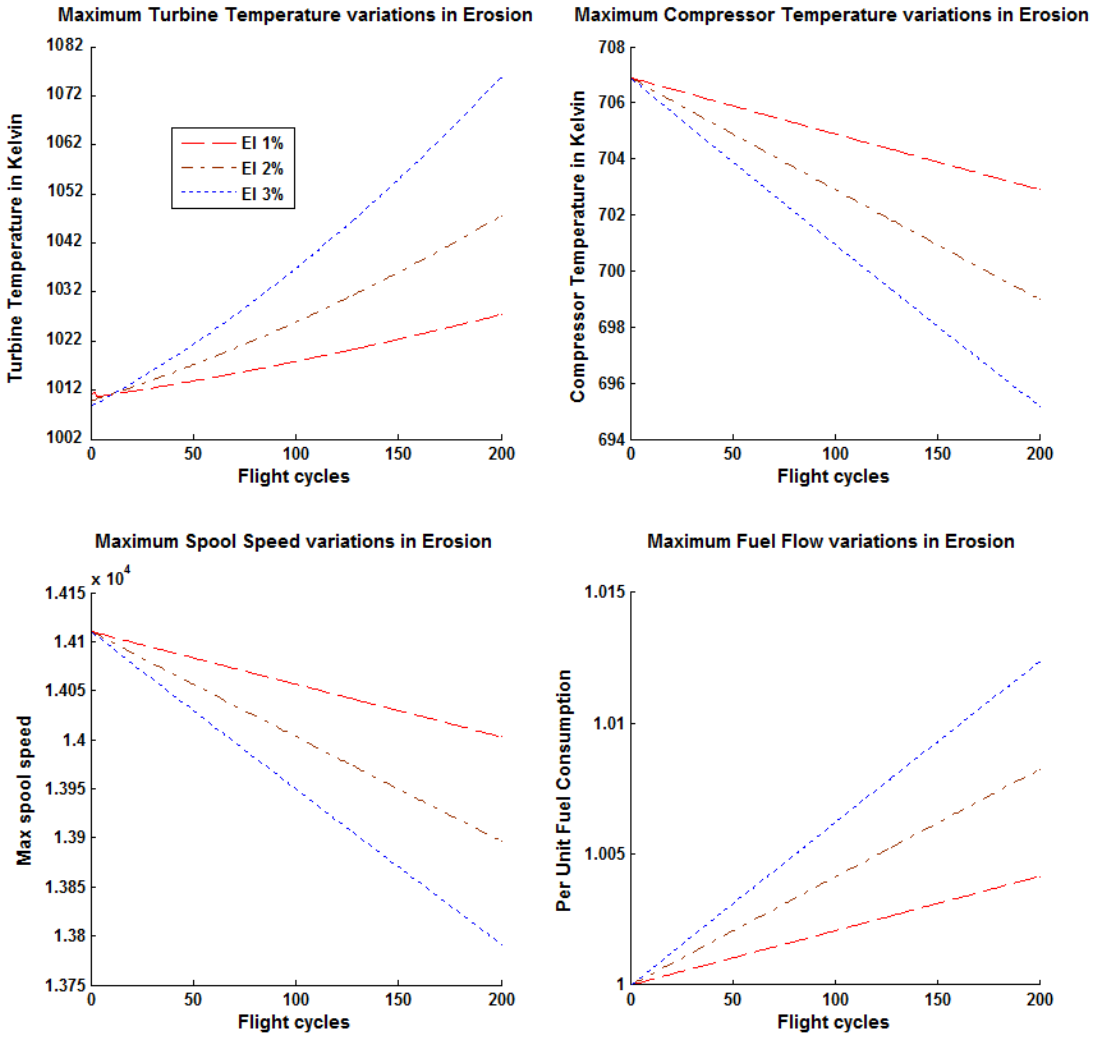

Figure 3.7: The outputs of the model corresponding to the different levels of erosion degradation.

speed has a decreasing trend again. The increase in the erosion index has more significant effects on the variations of the engine parameters. We should mention again that this integrated model is adaptable and we can adjust the number of cycles, flight duration, degradation level, etc. As the erosion level increases, these consequences become more pronounced in a nonlinear fashion.

Here again, we have conducted the same scenarios. The output of both models are shown in a tabular form (Table 3.2). One should note that the levels and values are not the same as the design specification of the single spool gas turbines used in each model are different. The compressor and the turbine maps are different and thus the fuel levels are different. But in both methods the output thrust are kept constant as compared to the healthy mode where the degradations are not initiated in the system yet. One can see the similar trend.

Table 3.2: Per unit changes in the engine parameters corresponding to the erosion scenarios.

Engine Parameter	Per Unit Change in Our Model	Per Unit Change in the GSP [176]
Fuel Flow consumption in EI 1%	0.04181	0.01763
Fuel Flow consumption in EI 2%	0.08272	0.03367
Fuel Flow consumption in EI 3%	0.01236	0.04949
Compressor Temperature in EI 1%	-0.05626	-0.04173
Compressor Temperature in EI 2%	-0.01112	-0.07378
Compressor Temperature in EI 3%	-0.01648	-0.09640
Turbine Temperature in EI 1%	0.0568	0.01771
Turbine Temperature in EI 2%	0.01086	0.03352
Turbine Temperature in EI 3%	0.01632	0.05396
Spool Speed in EI 1%	-0.07583	-0.01029
Spool Speed in EI 2%	-0.01502	-0.01854
Spool Speed in EI 3%	-0.02246	-0.02500

At the end it is worth to note that in the results shown above we chose to show them in figures or tables as they are easier to read and to study and we only showed the maximum value of each cycle. However, in the developed model, it is possible to have all the measurements from each flight cycle. For example, if we would like

to have 2% FI in the system after 100 flights, the changes in the health parameters (efficiency and mass flow rate) start to increase until they reach their ultimate point at the 100^{th} cycle. We will save all the available gas path measurements associated with this level of degradation and use them to train and test the neural network for trend analysis. This will be discussed in details in the following chapter.

3.3 Conclusion

In this chapter, possible degradations in the jet engine were introduced. Our goal in this thesis is to track and predict the degradation propagation. We presented different causes for the degradations and the components which can potentially get affected by these, i.e. fouling/erosion. A solution was then presented on how to model the degradation as a function of the engine health parameters (efficiency and the mass flow rate) and how to relate them to the gas path measurements.

The equations are used to integrate the degradation modelling part to a previously developed single spool jet engine model. We studied fouling effect on the compressor and erosion effect on the turbine separately. In each scenarios three different levels of deterioration were considered and the goal was to keep the output thrust constant. The data derived from this model will be analyzed for the engine prognosis purposes. The model is compared and validated with the GSP11 software and the result from the two models were tabulated and were proven to have the same trend. Some result are presented in the following work.

N. Daroogheh, A. Vatani, M. Gholamhossein, K. Khorasani, " Engine Life Evaluation Based on a Probabilistic Approach", ASME congress, IMECE2012, Houston, TX, USA.

Chapter 4

Jet Engine Degradation Prognostics Using Recurrent Neural Networks

In this chapter the problem of trend analysis and prognostics for a single spool aircraft engine is addressed. Towards this end, recurrent neural networks (RNN) are developed and trained to learn the degradation growth dynamics and then to predict the pattern for some flight steps ahead. This work does not consider diagnostics or anomaly detection and instead focuses on prognostic aspects. The aim of the prognosis function is to predict a sensor signal evolution, where its function is strongly dependent on the dynamic behaviour of the process.

RNNs have proven to be suitable for the task of prediction. The base of this type of network is a feedforward multi layer perceptron (MPL) and the dynamics is achieved by feeding the output of the network back to its input layer which introduces memory to the network. We introduced the RNNs in Chapter 2, in addition to the different architectures, training algorithms, etc. In this chapter we investigate on

the type of the recurrent networks we have proposed for predicting the gas turbine degradation trends by studying their effect on the gas path measurements. Various simulations under different scenarios are carried out to demonstrate the performance and accuracy of the proposed network.

Application of RNNs for prognostics has become quite popular recently. RNNs incorporate temporal information and store these into their functionality. In [140] Wang *et al.* have utilized recurrent wavelet neural networks to predict the future dynamics. The network is used to prognosticate the remaining useful lifetime of a defective bearing with a crack in its inner race. In [86] RNNs are combined with a fuzzy-based (NF) approach to build a reliable machine fault prognostic system. The performance of the system is evaluated by using two benchmarks. Tse *et al.* [79] have introduced a prognostic method for forecasting the rate of machine deterioration using recurrent neural networks. Vibration-based fault trends have been analyzed for industrial gas turbine and the results are promising for predicting the remaining life span of defective components.

Due to adding feedback connections to the feedforward network, there are at times the risk of network instability and unrobustness. This problem has been addressed in [188, 6]. It should be noted that in RNNs as with other models, the input configuration is critical to good prediction performance, implying that the training data should be sufficiently rich so that it contains all the relevant information including noise that is necessary for learning.

4.1 Recurrent Neural Network Prognosis Approach

In this section, we will see how RNNs are used for predicting the future value of a parameter and afterwards we will see how knowing the future value of some engine parameters will help us to study the degradation and its propagation through time.

Note that the main aim of this chapter is to develop an RNN-based engine degradation prognostic system.

RNNs have shown potential in temporal forecasting and neural networks in general are flexible models for nonlinear prediction, especially in certain complex dynamical systems where a comprehensive expert system is not available. These types of predictors are build automatically by training. They usually do not need any *a priori* statistical information from the data and do not require any identification of the model structure or the parameters. According to [6], the main reasons for using neural networks for prediction rather than statistical time series analysis are:

- NNs are computationally as fast or even faster than most available statistical techniques;
- NNs are as accurate or even more accurate than most of the available statistical techniques;
- NNs are self-monitoring meaning that they learn how to make accurate predictions;
- NNs provide iterative forecast; and
- NNs are capable of coping with nonlinearity and nonstationarity of input processes.

Although some researches have used feedforward neural networks for prediction [189, 190], RNNs have better performance in learning the dynamic behavior and the projecting them to the future [188, 191, 192].

As described in Chapter 2, recurrent neural networks are similar to feed-forward neural networks but with an additional feedback path. In other words, RNN is a closed network which is now suitable for learning temporal behaviour [79]. Hence, in

a recurrent neural network the current output and its activation state is a function of the previous activation states as well as the current input. For instance, for a one-step-ahead prediction at time k, the activation of the output node at time $k+1$ is a function of activations of the neurons in the hidden layer and the input layer at the previous steps. There are no explicit equation that can determine the number of input nodes, neurons in the hidden layer, delays etc. and the choice of these parameters can affect the goodness of the prediction and possible steps-ahead. This is an open subject and we will investigate more on this matter in the rest of this chapter until we get our satisfactory results.

A summary of the work done in this chapter is provided below. For each parameter i.e. pressure and temperature we have data which shows the changes and evolution in the parameters under study through time or flight cycles for a degraded engine. We divide a portion of the data and feed it to the input layer of the RNN. The neurons are connected to each other by weights. The input neurons distribute the signals forward to the next layer. In the next layer, which is the hidden layer, each neuron receives a signal which is the weighted sum of the output of the neurons in the input layer. The total output in a neuron is obtained through a nodal activation function. The first portion of the data is used for training. In this phase the goal is to find an input-output relationship between the set of training data.

We have chosen a batch supervised learning method which consists of processing and learning phases. In learning the result or the output pattern is compared with the target pattern, and the error is computed as the difference between these two values. The error or the residual is then propagated back to the previous layer and all the weights are adjusted based on this error. Different error functions can be defined and minimized such as the sum of the squares of the errors. This formulation is presented in Chapter 2. Once the error goal is reached the network is said to be trained. The

final error can help us determine the percentage of the data that are used for training the network.

Once our recurrent network is trained we use the rest of the available data for the same parameter to test the network. We do so by feeding this data that has never been seen by the network to the input layer and then find the network output and compare it to the test output. If the test error is high while the training phase error is low, we take it as a sign that the network was over trained implying that too much information was fed in the training phase. This is a common challenge with the employment of neural networks and one has to be careful when one divides the data or selects the network parameters. In the case that the test error is high the percentage of data used for training has to be readjusted and the network has to be tested and trained again until the desired requirements are satisfied. Otherwise, if the error is acceptable we state that the network has learned the process dynamics well and can be used for trend prediction.

4.2 Engine Data Generation

All neural network based methods are data-driven methods and their performance relies on the set of data they are provided with. In this chapter we investigate data generation and the type of data that was used in this thesis. The data that we use for testing and training our neural network are derived from a single spool jet engine model. The original model, nonlinear equations and validation with the GSP software were described in Chapter 2. Since the purpose of this thesis is to study the degradations in the system, we added certain important causes of engine deterioration to this model in Chapter 3. The formulation and data validation with GSP can be found there.

We utilize the knowledge about the measured variables taken along the engine's

Table 4.1: Available engine measurements and their abbreviations.

Gas Path Measurement	**Abbreviation**
Fuel flow rate	$\dot{m}_f$
Spool speed	N (RPM)
Compressor temperature	CT (K)
Compressor pressure	CP
Turbine exit temperature	TT (K)
Turbine pressure	TP

gas path and then by studying their trends we can determine the engine's health state and the time to failure or probable maintenance actions that have to be taken. For a single spool engine temperature and pressure of both the compressor and the turbine are available as well as the spool speed connecting these components together and the fuel flow rate. In addition to the fuel flow ($w_f(t)$) which is the system input we have five measurements. These measurements are summarized in Table 4.1.

Each engine component has an efficiency value and a mass flow rate. The changes in the turbine and compressor efficiency and mass flow rate allow one to model the fouling and the erosion. We will refer to these values by the notations presented in Table 4.2. Efficiency and the mass flow rate are also referred as the engine health parameters, as changes from the nominal value can be a sign of an unhealthy engine. Variations in the health parameters affect the gas path measurements. In this thesis we will analyze and predict these trends when they are only due to soft degradation and not hard degradations such as FOD.

The gas path measurements can be generated for different phases of the flight such as take-off, cruise, etc. We have assumed that for each simulated cycle the flight mission takes 3000 seconds. Generally the take-ff mode can be divided to two individual phases, namely the ground roll phase and the air phase. The former is from the break release to the lift-off of the last aircraft wheel. The air phase is from

Table 4.2: Engine health parameters and their notations.

Engine Health Parameter	**Notation**
Compressor efficiency	η_C
Compressor mass flow rate	$\dot{m}_C$
Turbine efficiency	η_T
Turbine mass flow rate	$\dot{m}_T$

the lift-off until the aircraft reaches an arbitrary altitude. In the ground roll phase the engine operates from its idle condition to its maximum power and the effect of degradation is more significant in this phase, therefore we will consider this phase of the flight for our investigation.

If we assume the take-off time to be at t seconds, using our simulation software we can choose our desired sampling time s. This value determines the number of samples taken at each second. After each simulation the number of data points for one parameter is $t \times s$. We considered a standard take-off time to be 20 seconds. We can apply different fuel rates to the system and determine the flight phase by adjusting the Mach number and the altitude. A general turbine temperature variation (in Kelvin) during the take-off is depicted in Figure 4.1 for a healthy engine. Healthy engine in this context refers to an engine which does not suffer from any type of degradation.

After determining the take-off duration, we can set the cycle or the number of flights being taken. Now let us assume we would like to study the effect of fouling on the system. We will observe the turbine temperature (TT) variation as it is a suitable indicator. Since the turbine is at the last stage of the engine, its output temperature entails valuable information. As an example, if we want a 2% fouling to affect the compressor at the end of the 100^{th} flight, which is equivalent to 2% reduction in the compressor efficiency (η_C) and 1% increase in the compressor mass flow rate ($\dot{m}_C$),

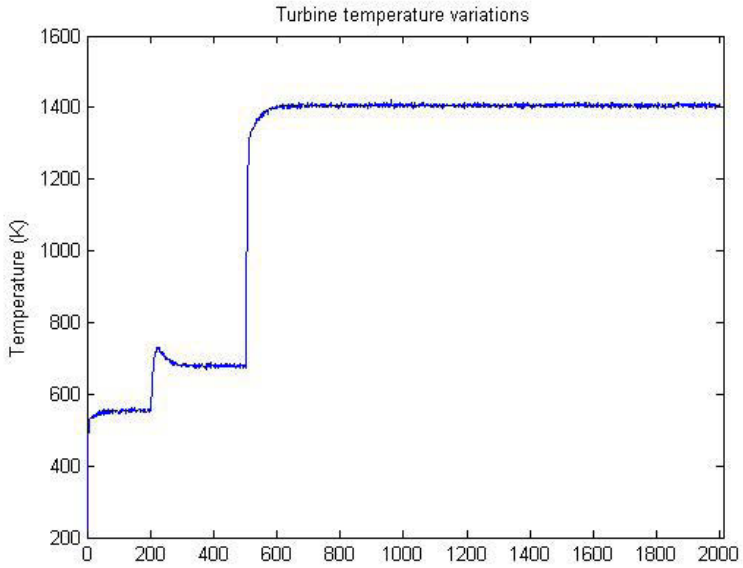

Figure 4.1: Turbine temperature variations during take-off for a healthy engine.

these changes have to be linearly applied. Since we want to keep the output thrust constant at all times, even when the engine is degraded, we have to increase the fuel. The details can be found in Chapter 3.

The procedure for erosion data generation is the same, except that erosion mostly happens in the turbine, affecting the turbine blades, and causes a reduction in the turbine mass flow rate ($\dot{m}_T$). Erosion causes an efficiency drop in the turbine.

In this thesis our objective is not to train the neural network to develop the dynamics of the engine, instead we are interested in training the network to learn and predict the dynamics of degradation and how it grows in the system as the flight cycles continue. Towards this end, we take only one sample from each flight rather than using all the data points. Since fouling and erosion are soft degradations they do not change the system's behaviour drastically in only one flight cycle. Therefore, from each flight cycle we collect the 12^{th} second data. In fact, prognosis can be accomplished in either time or frequency domain or even the event domain, as these domains are made up of ordered points.

In reality all the measurements and readings are affected by noise. Therefore we

Table 4.3: Noise standard deviations.

CT	CP	TT	TP	N	$\dot{w}_f$
0.23	0.164	0.097	0.164	0.051	0.51

have conducted all our analysis in presence of noise. The nominal values of noise levels are given in Table 4.3 where the standard deviations are given as a percentage of the nominal values [97] at our typical take-off profile when reaching the steady-state condition.

4.3 Simulation Results

In order to study the effects of degradation originated from fouling of the compressor or the erosion of the turbine on the engine measurable parameters which can be used later for prognostics, the dynamical model of a single spool engine that is developed in Matlab/Simulink [9] is used. The degradation model is integrated into the Simulink model. We have already described the nonlinear equations that allows us to derive such a model. The fouling and erosion effects in the system have been considered as changes in the engine health parameters, i.e. in the efficiency and the mass flow rate.

To achieve a constant level of thrust for a given aircraft performance, the engine may run at higher speeds or higher turbine-entry temperatures. In our model, the strategy is to maintain the thrust constant by increasing the turbine-entry temperature through increase in the fuel consumption.

4.3.1 Simulation Results for Fouling Scenarios

The general approach for prediction using RNNs was explained at the beginning of this chapter. Our general goal in prognosis is to identify the degradation level at

a certain point in time in future. When applying neural networks, there are many parameters that we have to adjust. Some of these parameters are the number of layers, the number of neurons in the hidden layer, the number of input neurons, the number of delays, activation functions, the number of epochs, the percentage of data for training the network, etc. The choice of neural networks to represent a physical process depends on the dynamics and complexity of the network that is best for representing the problem in hand. There is no optimal value to all the above parameters and the optima have to be found through trial and error which can be tedious. Below we demonstrate different choices for these parameters and show how they can affect the capability of the prediction. The prediction horizon can be time or cycle and we have chosen cycle numbers for our prediction horizon. Finally, we will present the neural network that gives the best prediction performance.

In the following subsections we will show our simulation results for different scenarios. In our scenarios the effects of fouling and erosion degradations are studied separately on the system. In each scenario the degradation level is increased from 1% to 3%. We do not go beyond this level as washing and maintenance is recommended beyond this level. Each scenario itself consists of a number of cases. By following the cases one can observe how we have decided on the data size and the network parameters and how the results are improved by adjusting different variables. One can also note how many steps-ahead prediction is viable by using our proposed network. Toward these ends, we will compare the actual and predicted values which are obtained as the NN outputs, study statistically and depict the errors. Now let us start by the first fouling scenario.

4.3.1.1 First Scenario: FI = 1%

FI = 1%: Case 1

As mentioned above, we collect the 12^{th} second of each take-off cycle for an engine degraded by fouling. The fouling level is fully effective at the end of 100^{th} cycle and hence we have 100 data points. The data is depicted in Figure 5.1. We assume that according to the accuracy expectation and the network architecture we have to decide on the percentage of data which are to be fed to the network for training. In the first case, let us consider a RNN with one hidden layer. The approach is schematically depicted in Figure 4.3.

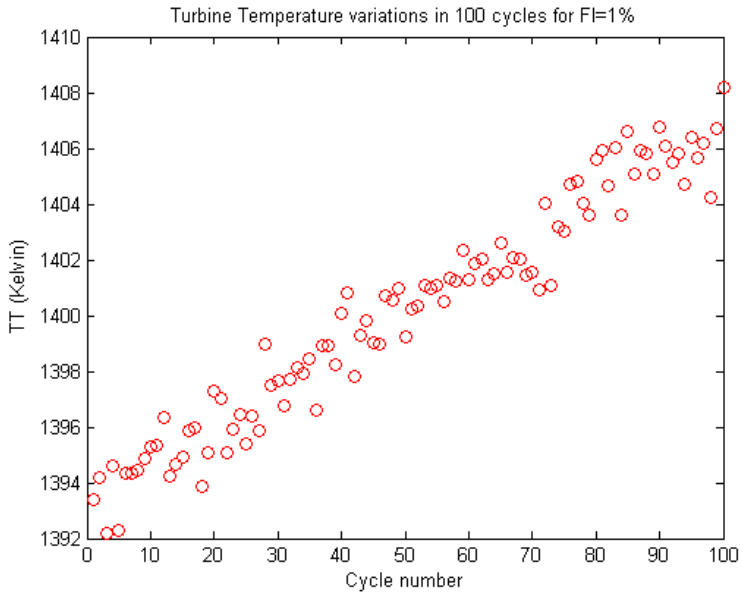

Figure 4.2: Turbine temperature variations due to a fouled compressor.

Our goal is to achieve the best result for three-step-ahead prediction ($l = 3$) that here implies three cycles ahead. This three-step ahead is a starting point case and once we have achieved this number of steps, we can widen our prediction horizon. As mentioned earlier, no specific rule exists for the minimum or the maximum number of steps-ahead feasible and the results are to be determined experimentally. In the first

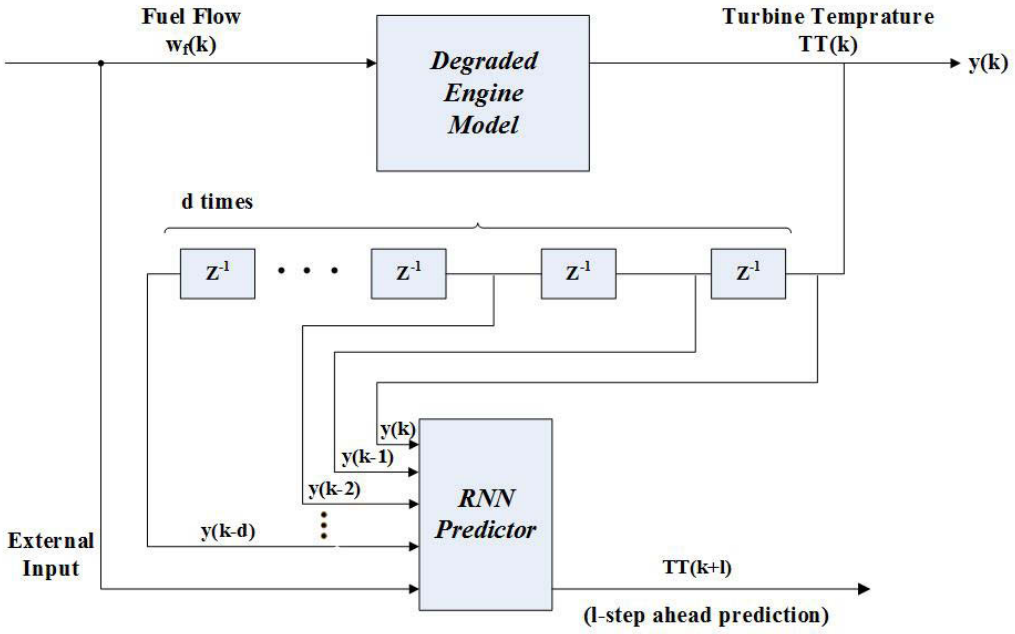

Figure 4.3: Schematic of the RNN-based prediction approach.

case we use half of the available data for training the network and the rest for testing the network. The number of neurons in the hidden layer are set to be five and the network epoch is 20. The input to the network will be the previous values of turbine temperature and the fuel flow rate. The number of the delays is set to be 2 ($d = 2$), this results in a 4-5-1 network architecture ($4 = 1{+}1{+}2$ where input, output and two delayed inputs serve as the input to the RNN). The error is defined as the difference between the network output in the testing phase and the real turbine temperature data, which defines the actual prediction error. We will represent the error graphically and quantitatively by calculating the mean (μ_{ae}) and standard deviation (σ_{ae}) of the absolute error (ae). Absolute error is defined as the difference between the real data and the network output. Another quantitative way is to find the root mean square of the error ($rmse$). In Figure 4.4, the actual and the predicted turbine temperature, which is in fact the output of the network, are presented.

In Figure 4.4, the *circles* denote the actual temperature incremental trend and

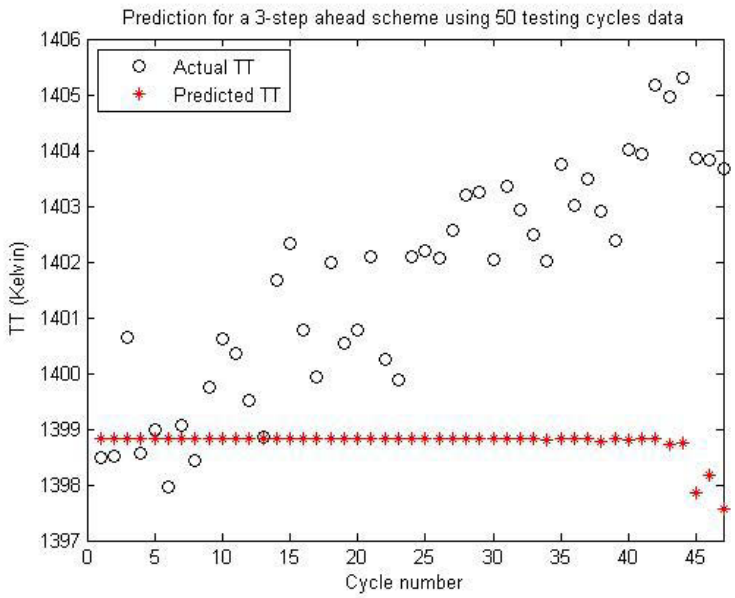

Figure 4.4: Actual vs. predicted TT for a 1% fouled compressor (3-step ahead).

Table 4.4: Prediction error for FI=1% case 1.

μ_{ae}	4.901 (K)
σ_{ae}	2.187 (K)
$rmse$	5.319 (K)

the *stars* show the predicted ones. It is clear that the network is not able of learning the degradation dynamics and predict it. The cause of this poor performance can be either due to insufficiency of the input data or the poor adjustment of the network parameters. In Figure 4.5 the prediction error is shown which is increasing and does not show improvement.

The mean of the prediction error in this case is $\mu_{ae} = 4.556K$, the prediction standard deviation is $\sigma_{ae} = 2.187K$ and the error root mean square is $rmse = 5.043K$. These error results are summarized in Table 4.4.

We conclude that this prediction performance is not acceptable and the network needs more samples to learn the dynamics of the degradation. In the next case we provide the network with 70% of the available data.

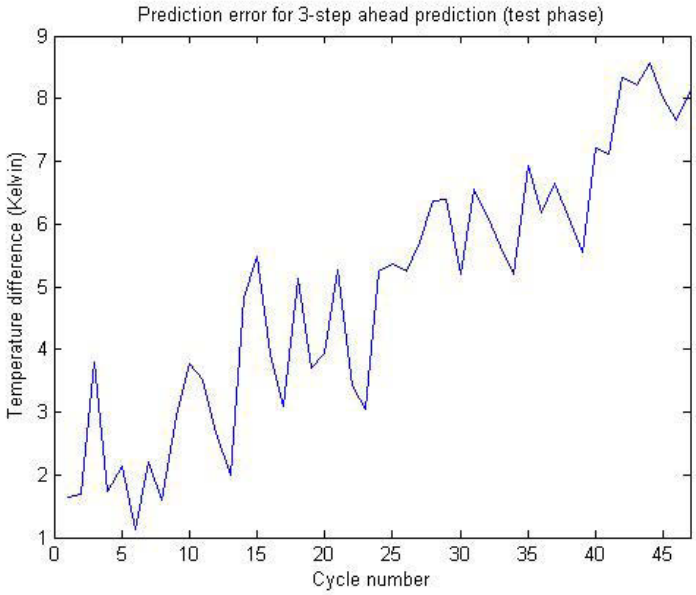

Figure 4.5: Temperature prediction error for a 1% fouled compressor (3-step ahead).

FI = 1%: Case 2

In the second case we still consider a fouled compressor in which the fouling level is 1%. We try to predict the turbine temperature over 100 take-off cycles which was depicted in Figure 5.1. This time we provide the network with 70% of the data and use the remaining 30% for testing the network performance. The number of delays is the same as in case 1 ($d = 2$) which results in the same 4-5-1 network. The comparison of the actual data and the predicted value are depicted in Figure 4.6.

It can be observed from Figure 4.6 that now with additional amount of training data, the network has become able to distinguish and learn the increasing trend of the turbine temperature due to the fouling. The error level shown in Figure 4.7 is also decreased compared to the Case 1.

In addition to the graphical representation of the error we use the statistical performance measure to judge the results in this second case. The mean of the prediction error is $\mu_{ae} = 1.177K$, the prediction standard deviation is $\sigma_{ae} = 1.073K$

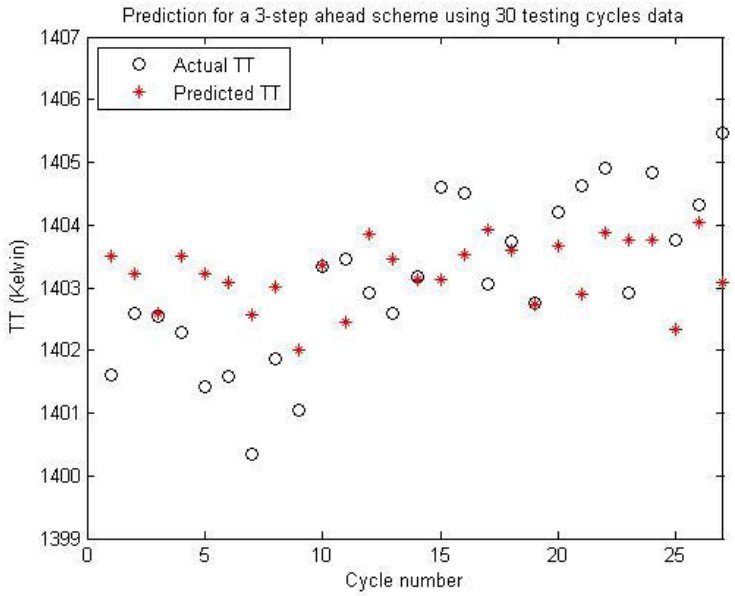

Figure 4.6: Actual vs. predicted TT for a 1% fouled compressor for three-step ahead prognostication.

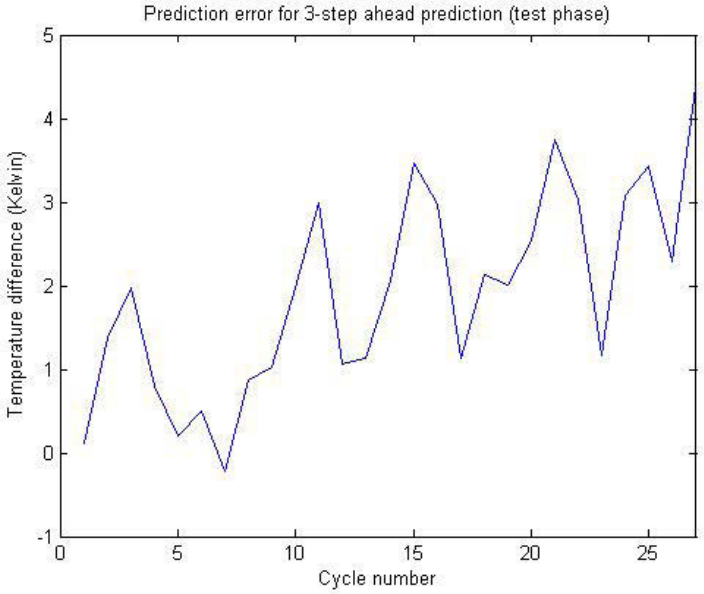

Figure 4.7: Temperature prediction error for a 1% fouled compressor (3-step ahead).

Table 4.5: Prediction error for FI=1% case 2.

μ_{ae}	1.177 (K)
σ_{ae}	1.073 (K)
$rmse$	1.579 (K)

and the error root mean square is $rmse = 1.579K$. These error results are summarized in Table 4.5.

Referring to these results, one can note the improvement in the prediction performance, however one would still need to achieve a more accurate prediction before performing a more-steps ahead prediction. In the next case we adjust the network parameters through the number of the neurons in the hidden layer.

Remark. One important concept in prognosis is uncertainty and uncertainty management. According to [140], uncertainty in prognosis is the rule rather than exception and it manifest itself at different levels of this procedure such as data level and decision level. Dealing with uncertainties is inevitable. As one extends the prediction horizon, the uncertainty will increase consequently. The prognosis procedure operates over time horizon from the past, through the present and to the future. It is not always straightforward to identify the uncertainty sources and model them. These uncertainties could be originated from insufficient data or the changes in the operating conditions. Thus to manage the data uncertainty, in this thesis we define upper and lower bounds for reporting prediction performance as well as point prediction. When the upper bound of the prediction variable meets a specified threshold, one may declare that the engine should be taken for maintenance. To determine the confidence bounds for evaluating the prediction performance of the model, according to normal theory [193] a multiple of standard deviations of the prediction error (for a given confidence level, that is 95%) are added and subtracted from the prediction values. More details are given in the next *remark*.

Remark on defining the lower and upper bands.

A prediction or confidence band is used in statistical analysis to represent uncertainty about the value of a new (predicted) data point which is subjected to noise. The bands are usually used as part of the graphical representation of the prediction, estimation or regression results (See Figure 4.8 as an example) [194].

The prediction bands build up a prediction interval. In other words, a prediction interval is an estimate of an interval in which future observations will fall, with a certain probability, given what has already been observed. In the case involving a single independent variable, results can be presented in the form of a plot showing the predicted data points along with the point-wise prediction bands. Prediction intervals are also present in forecasts. The concept of prediction intervals need not be restricted to inference just a single future sample value but can be extended to more complicated cases. This shows the ample application of them [195].

The construction of these bands needs to be formulated. In the data set that was shown above, and throughout this thesis we consider the effect of measurement noise on our data points. Without loss of generality we assume that the measurement noise follows a normal distribution. We define a significance level denoted by α. The confidence level is determined by $(1-\alpha)$ [196]. If for example we set $\alpha = 0.05$, the confidence intervals covers the corresponding data points with the probability 0.95 .

If we are interested in finding the lower and the upper prediction bounds denoted by l and u, respectively, a prediction interval $[l, u]$ for a future data point in a normal distribution may easily be calculated from [195]:

$$\gamma = P(l < X < u) = P\left(\frac{l-\mu}{\sigma} < \frac{X-\mu}{\sigma} < \frac{u-\mu}{\sigma}\right) = P\left(\frac{l-\mu}{\sigma} < Z < \frac{u-\mu}{\sigma}\right) \tag{4.3.1}$$

Table 4.6: z values [17].

Cofidence level	z
50%	0.67
90%	1.64
95%	1.96
99%	2.58

where $Z = \dfrac{X - \mu}{\sigma}$ is a standard normal distribution. Hence,

$$\frac{l - \mu}{\sigma} = -z; \frac{u - \mu}{\sigma} = z \tag{4.3.2}$$

or

$$l = \mu - z\sigma; u = \mu + z\sigma \tag{4.3.3}$$

Equation (4.3.1) can be rewritten as:

$$\gamma = P\big(-z < Z < z\big) \tag{4.3.4}$$

The prediction interval is conventionally written as:

$$\big[\mu - z\sigma, \mu + z\sigma\big] \tag{4.3.5}$$

In this thesis, we have presented the error statistics in tables. To find the lower and upper bounds we need to find the value of z. The values of z for different confidence levels are given in Table 4.6.

For instance, to calculate the 95% prediction interval for a normal distribution with a mean μ_{ae} and a standard deviation σ_{ae}, then z is approximately 2 and therefore our lower and upper bounds are calculated as:

$$\left[\mu_{ae} - 2\sigma_{ae}, \mu_{ae} + 2\sigma_{ae}\right] \tag{4.3.6}$$

where *ae* refers to the prediction error. The above calculation is used in this thesis to construct a 95% confidence band.

Taking the above *remarks* into account, we are now able to define our prediction bounds. The goal is to evaluate the network for a 95% confidence level. We first find the standard deviation of the prediction error which for Case 2 is 1.073 K. The resulting upper and lower bounds are depicted in Figure 4.8. The actual turbine temperature during 100 cycles for a fouled compressor is shown with *circles*, the predicted temperatures are indicated by *stars* and the two dashed lines represent the prediction upper and lower bounds. Although this prediction capability is improved compared Case 1, but only half of the data points are close enough to their real values.

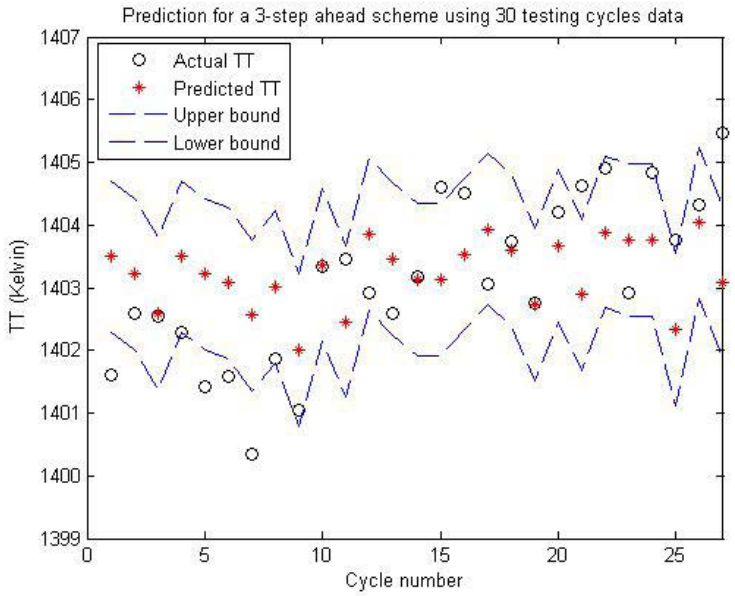

Figure 4.8: Actual vs. predicted TT for a 1% fouled compressor considering prediction bounds (3-step ahead).

Before proceeding we again emphasize that to be able to judge the goodness of

the prediction one has to study both the predication bands and the error statistics represented in the tables. With only one of them a correct conclusion cannot be drawn.

FI = 1%: Case 3

In the third case, we consider the same inputs and data (a 4-5-1 network) and try to improve the three-steps-ahead prediction by adjusting the network parameters. The number of neurons are set to five and the epoch number to 30. The resulting prediction performance is shown in Figure 4.9. Furthermore, the prediction error for

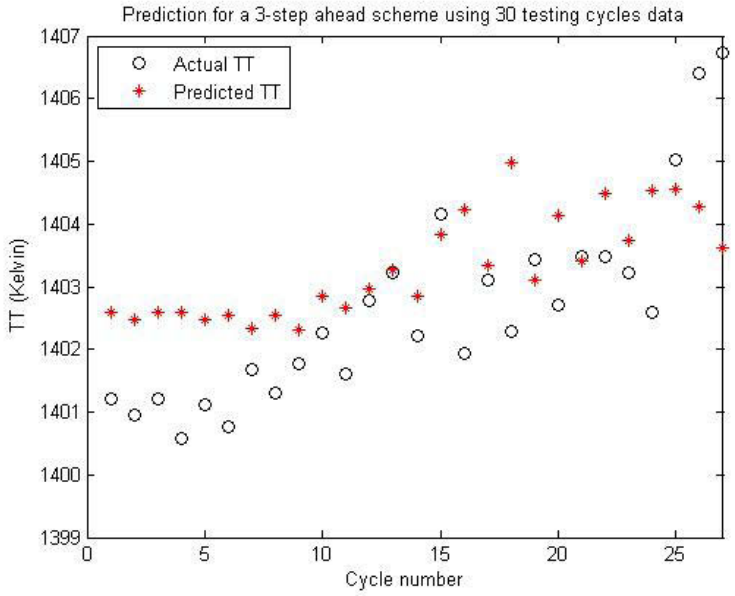

Figure 4.9: Actual vs. predicted TT for a 1% fouled compressor (3-step ahead).

the 30 test data point is shown graphically in Figure 4.10.

The network errors after the above changes are as follows. The mean of the prediction error is $\mu_{ae} = 0.177K$, the prediction standard deviation is $\sigma_{ae} = 1.384K$ and the error root mean square is $rmse = 1.343K$. The negative mean implies that the predicted data are always above the real value and so that the maintenance

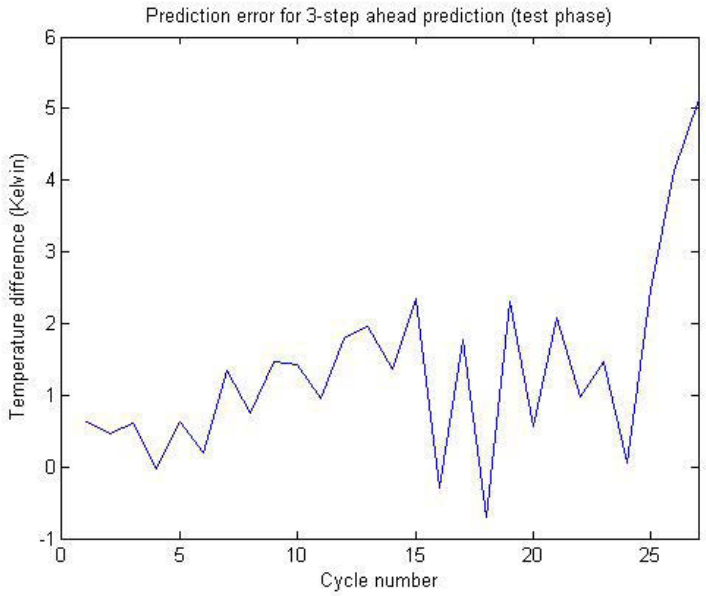

Figure 4.10: Temperature prediction error for a 1% fouled compressor (3-step ahead).

Table 4.7: Prediction error for FI=1% case 3.

μ_{ae}	0.177 (K)
σ_{ae}	1.384 (K)
$rmse$	1.343 (K)

should take place before the real severity level reaches a critical value. However, when compared to the previous case the standard deviation is larger which implies that the data are farther away from the mean . These error results are also given in Table 4.7. Finally, if we predict the upper and lower prediction bounds the result are shown in Figure 4.11.

When one considers predicting the bounds for the future turbine temperature when the engine is subjected to 1% fouling it is understood that although the standard deviation of the error has increased, but most of the real values lie in the prediction bounds as predicted by our recurrent neural network. Most of the data implies that only 2 out of the 27 data points are out of the bands. This shows that 92.6% of the

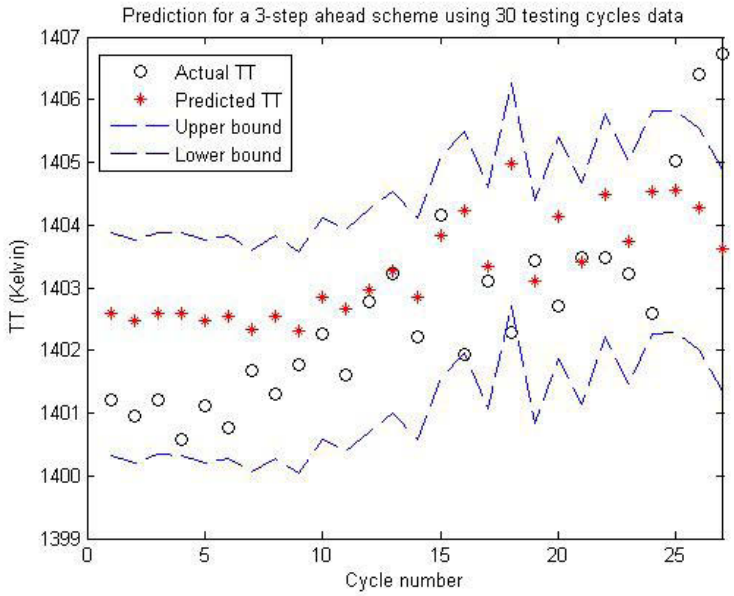

Figure 4.11: Actual vs. predicted TT for a 1% fouled compressor considering prediction bounds (3-step ahead).

data are inside in the bands.

We fix the resulting neural network in *case 3* to be the most suitable network for predicting a 1% fouling effect on the TT. In the following case that we consider for this level of fouling our goal is to extend the prediction horizon beyond a three-step-ahead or cycle-ahead range using the same trained predictor.

FI = 1%: Case 4

The network developed in *Case 3* is considered to be a suitable network that has learned the dynamics of the degradation propagation through flight cycles, we now use it to extend the prediction to beyond three-step-ahead as the results deemed satisfactory. For the next try, let us consider a six-step ahead prediction, implying that we are interested in predicting the turbine temperature for the 6^{th} flight from now by having the fuel mass flow rate ($\dot{m}_f$) and previous values of the turbine temperature which are fed back to the network. The number of output delay is $d = 4$ (TT) and

the current state of the fuel flow rate (with no delay) is set as the input which gives a 6-5-1 network architecture. In other words from the 30 data points available for testing, if one starts with the first point then the network will predict 6 steps further and thus gives 24 results for prediction. That is, in the test phase one provides the network with data point which has never been seen by the network and wants to verify how six-step-further in the future is predicted by the network. Starting with the first point, the 7^{th} point is predicted, providing us with 24 temperature data. The predicted versus the actual values are depicted in Figure 4.12.

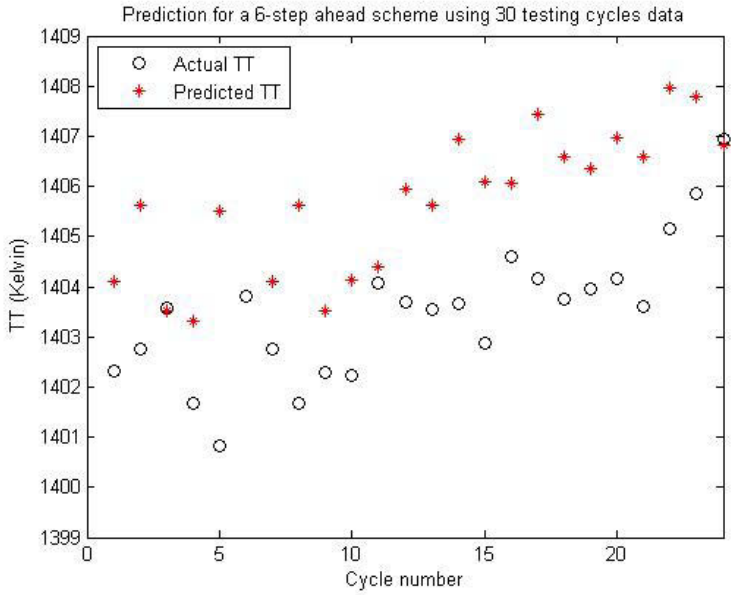

Figure 4.12: Actual vs. predicted TT for a 1% fouled compressor (6-step ahead).

The test error is shown in Figure 4.13. The error levels are not high and the obtained six-steps-ahead prediction is reasonably good. The mean of the prediction error is 1.968 K, the standard deviation is equal to 1.829 K and the *rmse* is 2.658 K. The error mean and the *rmse* are both increasing. A summary of error results for *Case 4* are shown in Table 4.8.

Figure 4.14 shows the results of predicting with the lower and upper prediction

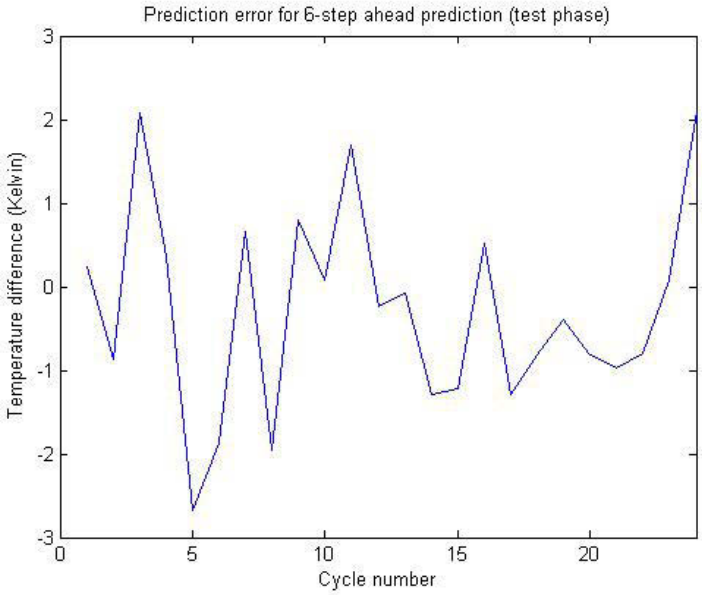

Figure 4.13: Temperature prediction error for a 1% fouled compressor (6-step ahead).

Table 4.8: Prediction error for FI=1% case 4.

μ_{ae}	1.968 (K)
σ_{ae}	1.829 (K)
$rmse$	2.658 (K)

bounds. A large percentage of data are within these bounds (more than 70%) which demonstrates the capability of our recurrent neural network for predicting the turbine temperature for six cycles ahead under a 1% *fouling index*.

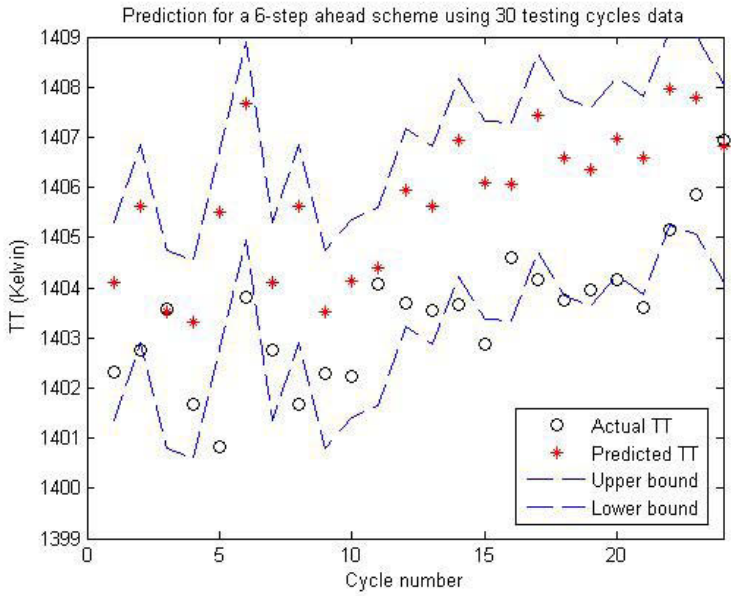

Figure 4.14: Actual vs. predicted TT for a 1% fouled compressor considering prediction bounds (6-step ahead).

FI = 1%: Case 5

In this Case we study a 1% fouled compressor for a ten-steps ahead prediction of the turbine temperature using the same 6-5-1 network as in the previous case. The prediction results as well as the test error are presented in Figures 4.15 and 4.16, respectively. It is observed that as we increase the prediction steps, more output delays are needed for a satisfactory result. The error statistical results for the ten-steps-ahead prediction are shown in Table 4.9 and they indicate reasonable error levels. We state that our error level is reasonable as the maximum error is 4 degrees which is less than 1% of the range of 1400K implying that the prediction is accurate by 99%.

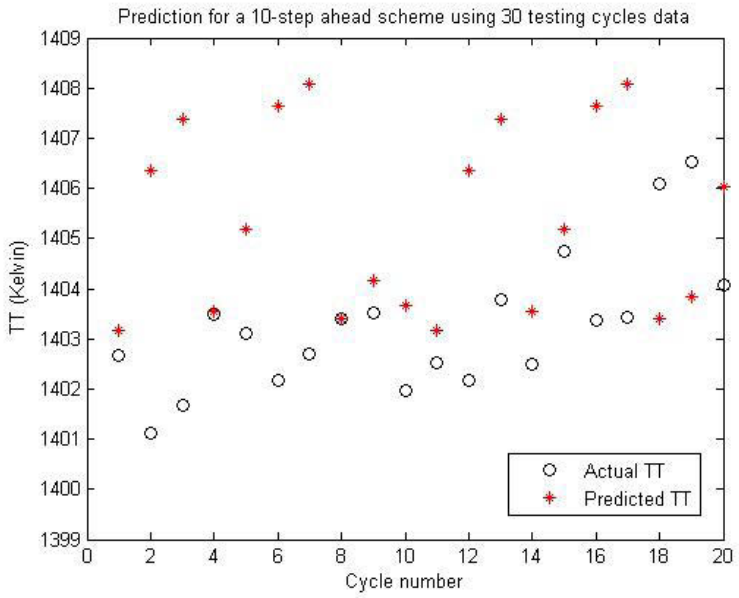

Figure 4.15: Actual vs. predicted TT for a 1% fouled compressor (10-step ahead).

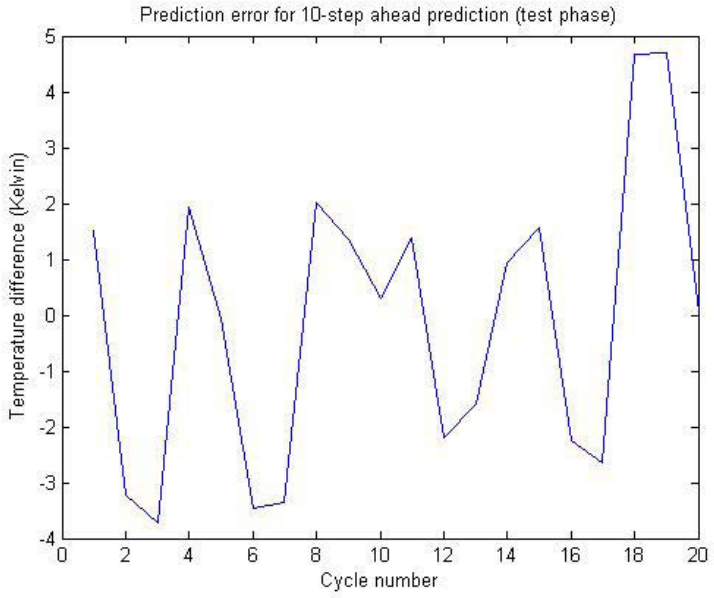

Figure 4.16: Temperature prediction error for a 1% fouled compressor (10-step ahead).

Table 4.9: Prediction error for FI=1% case 5.

μ_{ae}	-0.338 (K)
σ_{ae}	1.810 (K)
$rmse$	2.786 (K)

If we also consider the prediction bounds, as depicted in Figure 4.17, one can see that the actual data points have exceeded beyond the bound (65% are within the bands). Hence, one can conclude that this network is capable of predicting the turbine temperature for ten cycles ahead and can provide us with accurate results to be used for condition-based maintenance purposes.

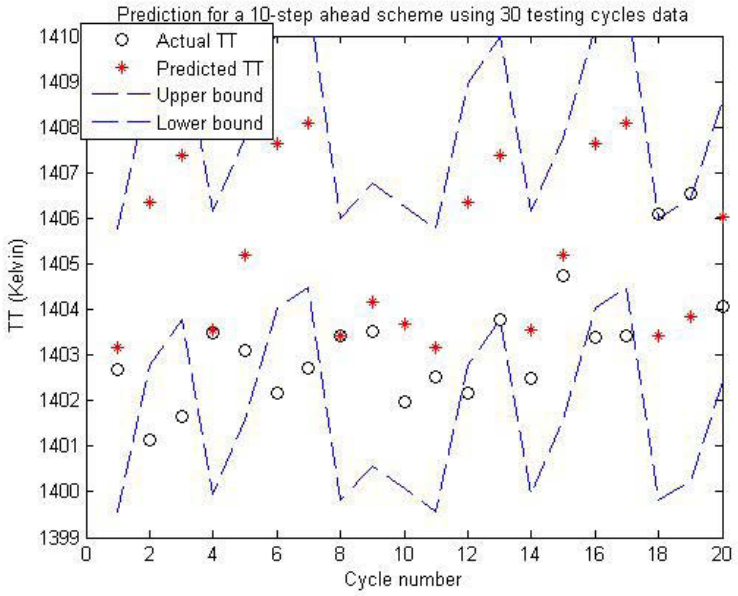

Figure 4.17: Actual vs. predicted TT for a 1% fouled compressor considering prediction bounds (10-step ahead).

4.3.1.2 Second Scenario: FI = 2%

In the second scenario that we consider in this chapter a 2% fouling index that is effective at the end of the 100$^{\text{th}}$ cycle is applied. This implies that at the end the

engine efficiency will drop by 2% and the mass flow rate will increase by 1%. The data generation method follows the same routine that was described for the case of 1% FI. The difference is that because the degradation level is increased, the changes in the turbine temperature to the compressor fouling will be more significant, i.e. the slope of the change will increase. The turbine temperature variations over 100 take-off cycles are depicted in Figure 5.11. The following cases will be discussed.

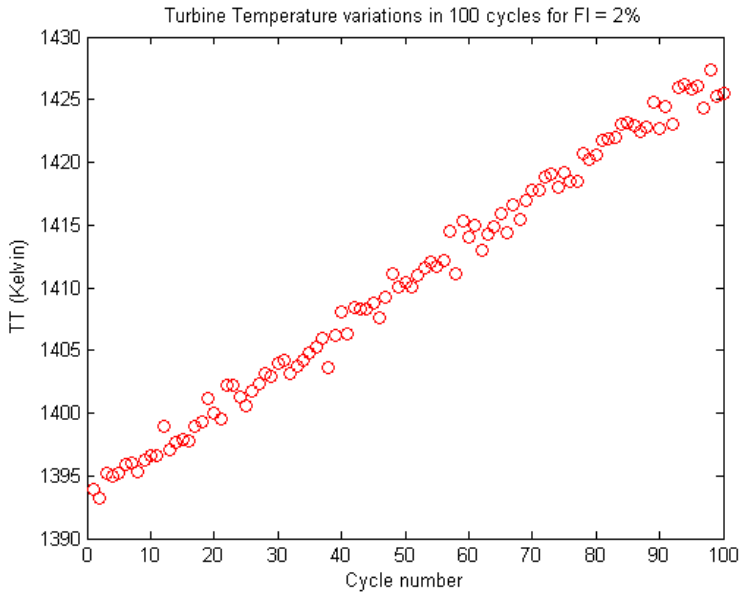

Figure 4.18: Turbine temperature variations due to a 2% fouled compressor.

FI = 2%: Case 1

We start by performing a three-steps-ahead prediction using the same network that was used in *Case 3* of FI = 1%. The network inputs are the same thus using a 4-5-1 network. The fuel flow rate ($\dot{m}_f$) will serve as the network input which is adjusted by the pilot and the current value of the TT is fed back to the network to help us predict its values in future, d is equal to two. In Figure 4.19 we have presented the prediction results versus the actual turbine temperature that is derived from our SIMULINK model (see Chapter 3). The predicted turbine temperatures are indicated with *stars*

Table 4.10: Prediction error for FI=2% case 1.

μ_{ae}	1.600 (K)
σ_{ae}	1.948 (K)
$rmse$	2.717 (K)

and the real turbine temperatures are indicated with *circles*.

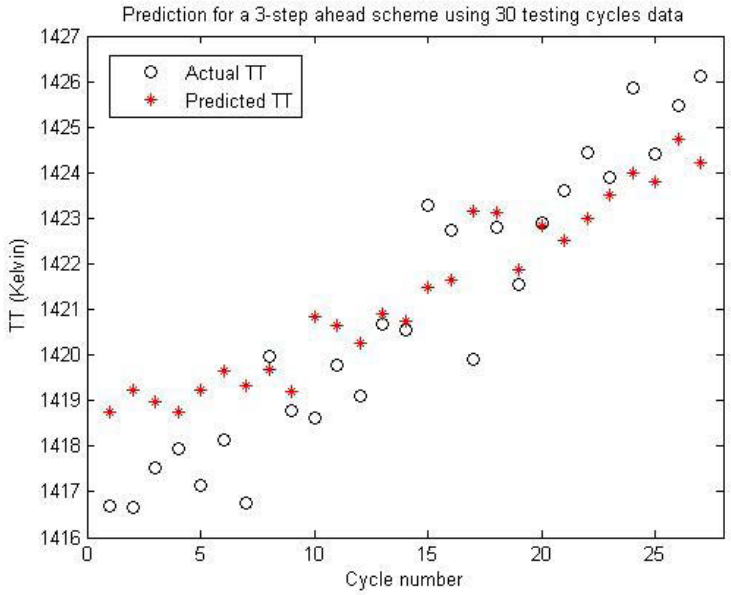

Figure 4.19: Actual vs. predicted TT for a 2% fouled compressor (3-step ahead).

It is verified from the Figure 4.19 that the network has learnt the increasing trend for the fouled compressor. The prediction error is depicted in Figure 4.20. At the early stages the error values are higher, but it becomes smoother as cycles proceed further on.

The predicted data points are in average $1.6K$ different from the actual values which is a very good prediction result as the nominal turbine temperature during take-off is about $1400K$. The statistics of the network performance are shown in Table 4.10

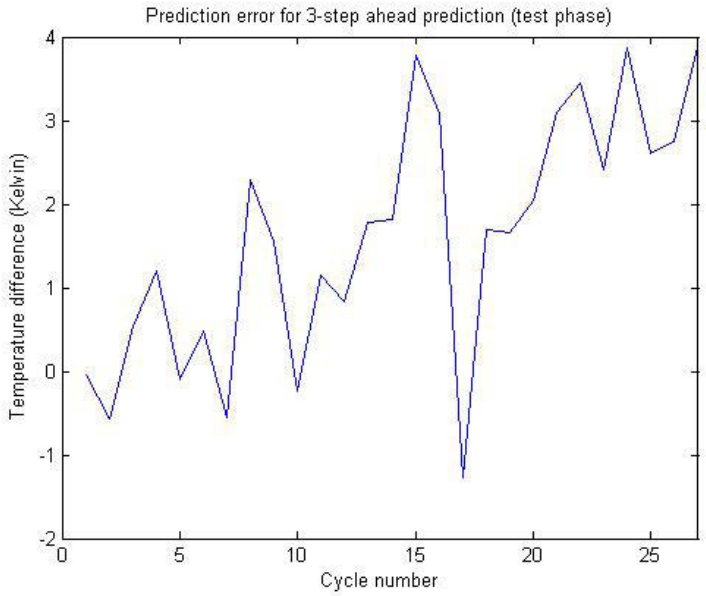

Figure 4.20: Temperature prediction error for a 2% fouled compressor (3-step ahead).

FI = 2%: Case 2

In the second case where the engine is under 2% fouling, we use the network that was used in *Case 1* of FI = 2% to determine how many steps-ahead prediction one can accomplish. The inputs are the same as before, the number of neurons in the hidden layer are ten, 70% of the available data are used for training and $d = 4$. The architecture is now 6-10-1. By tuning the network through many simulations the best prediction results are now reported. It was concluded that ten-step ahead prediction yield satisfactory results, considering the training data amount, training time, errors and whether or not the actual data lie in the prediction bounds. The comparison of the real and predicted turbine temperatures for a 2% fouled compressor are depicted in Figure 4.21.

The error results are graphically shown in Figure 4.22. Similar to the previous Case, the error has an increasing trend. The reason is that when one looks further through the cycles the network uses the previously predicted point which are

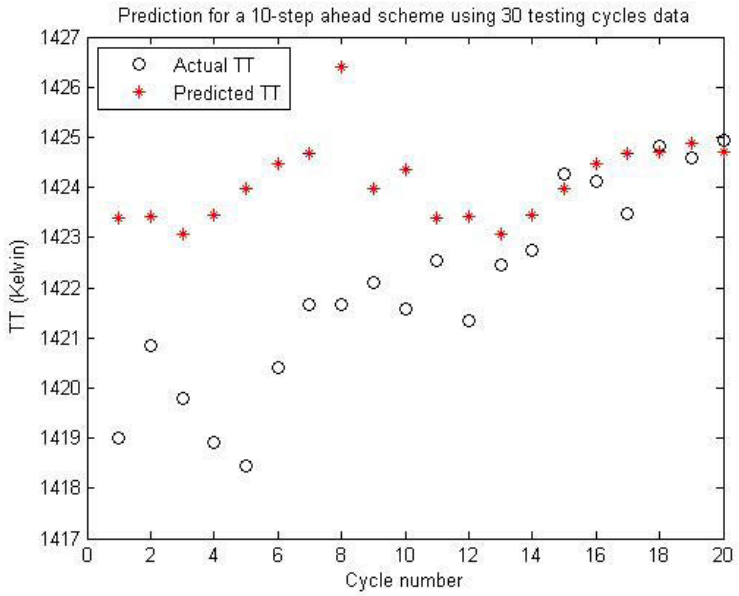

Figure 4.21: Actual vs. predicted TT for a 2% fouled compressor (10-step ahead).

Table 4.11: Prediction error for FI=2% case 2.

μ_{ae}	-1.815 (K)
σ_{ae}	3.632 (K)
$rmse$	5.792 (K)

themselves prone to error and this causes an increase in the overall error.

By studying the error statistics for performance assessment, as presented in Table 4.11 one can conclude that the error mean and variance are low and the overall error is less than 1%. This demonstrates the effectiveness of using RNNs for prognostics. The obtained prediction bounds as depicted in Figure 4.23 confirm this conclusion. 65% of the data points are within the predicted bands.

4.3.1.3 Third Scenario: FI = 3%

In the third scenario, we increase the fouling level in the compressor by decreasing the efficiency and increasing the mass flow rate even further. As a consequence, higher

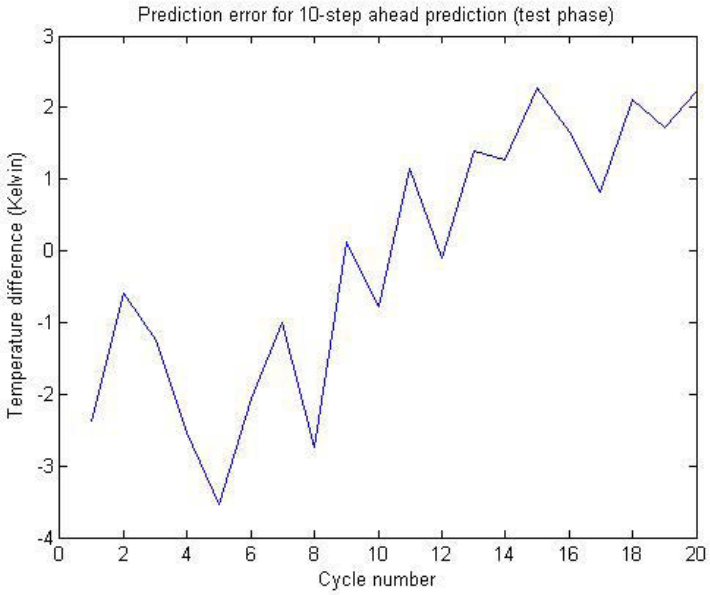

Figure 4.22: Temperature prediction error for a 2% fouled compressor (10-step ahead).

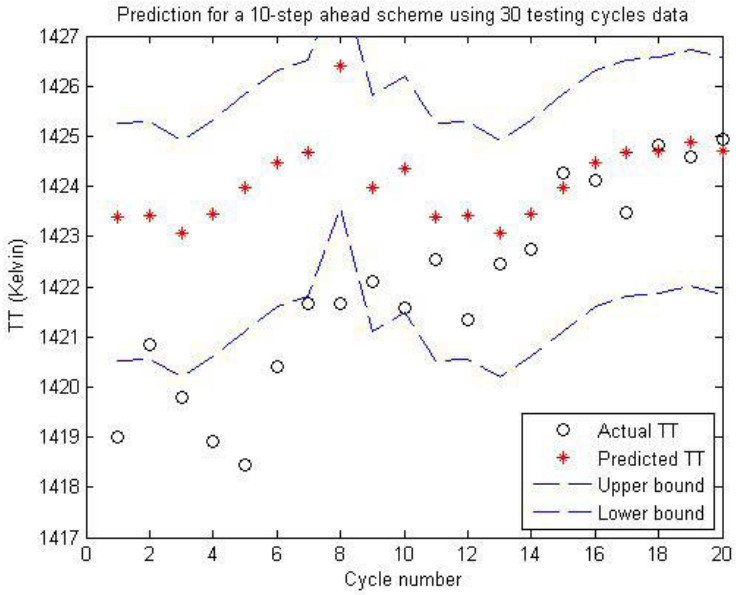

Figure 4.23: Actual vs. predicted TT for a 2% fouled compressor considering prediction bounds (10-step ahead).

changes in the turbine temperature will occur as depicted in Figure 5.21.

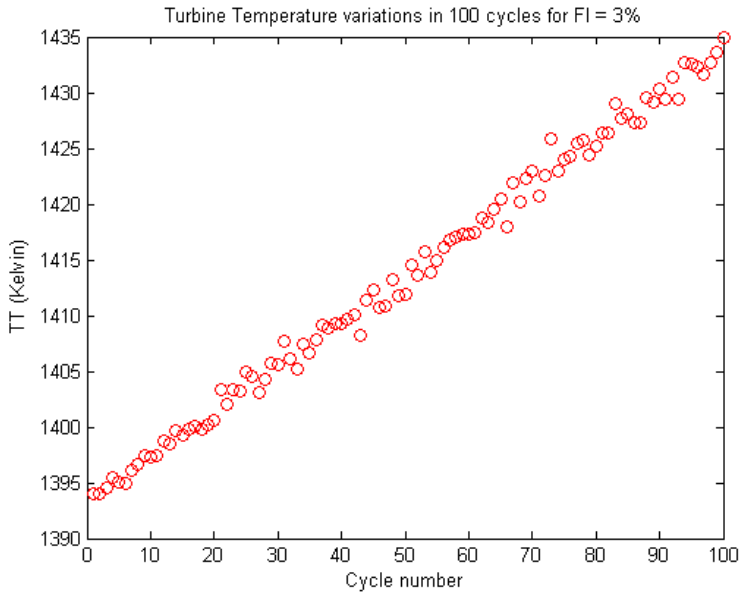

Figure 4.24: Turbine temperature variations due to a 3% fouled compressor.

FI = 3%: Case 1

We use the developed network in *Case 3* of FI = 1% to predict the turbine temperature variations where 70% of the data is used for training the network and the rest are used for testing the network prediction performance resulting in a 4-5-1 network. The actual and predicted temperatures are shown in Figure 4.25. The resulting error which is the difference between the actual and the predicted temperatures at each point are shown in Figure 4.26.

The statistical result for the error are given in Table 4.12. One should note that compared to 1400 K which is the nominal turbine temperature in kelvin during the take-off mode this is less than 1% error which is quite acceptable. The prediction results when combined with the prediction of upper and lower bounds are depicted in Figure 4.27. One can verify that all 100% of the data points are within the bands. This confirms the accuracy of the prediction, obtained by RNN for this case using

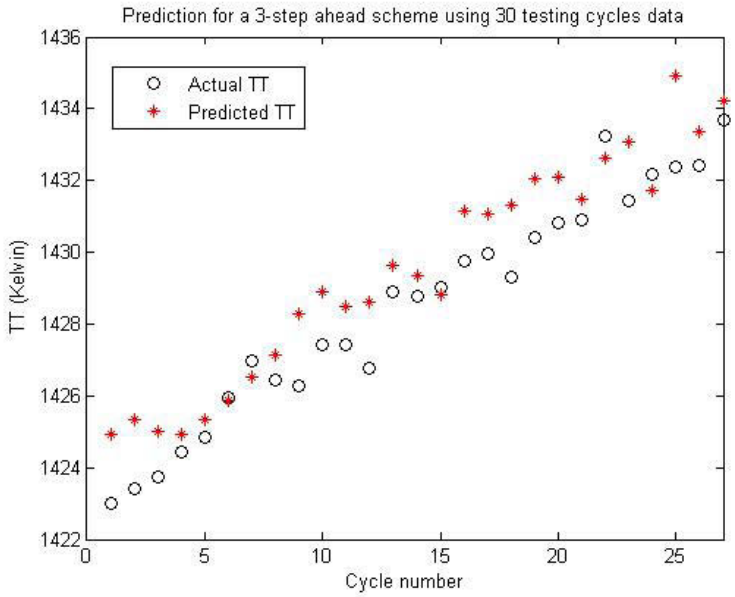

Figure 4.25: Actual vs. predicted TT for a 3% fouled compressor (3-step ahead).

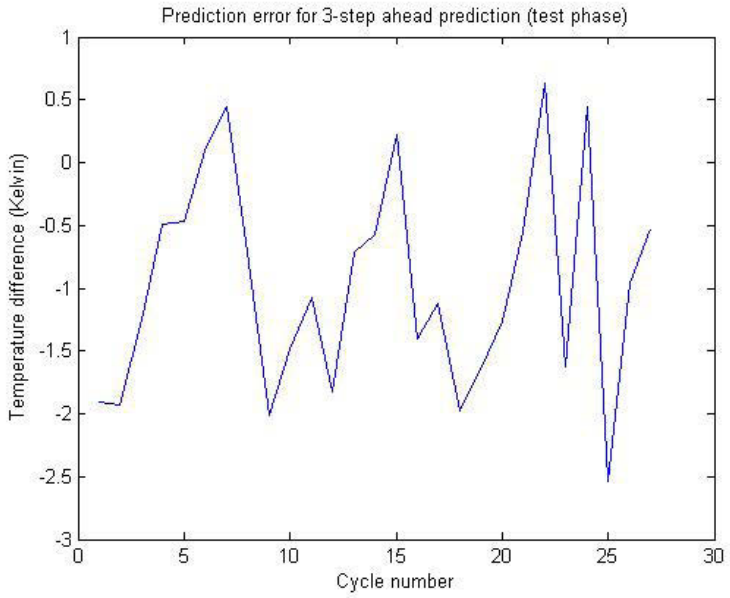

Figure 4.26: Temperature prediction error for a 3% fouled compressor (3-step ahead).

Table 4.12: Prediction error for FI=3% case 1.

μ_{ae}	-1.812 (K)
σ_{ae}	1.705 (K)
$rmse$	2.610 (K)

the aforementioned number of neurons, delays, etc.

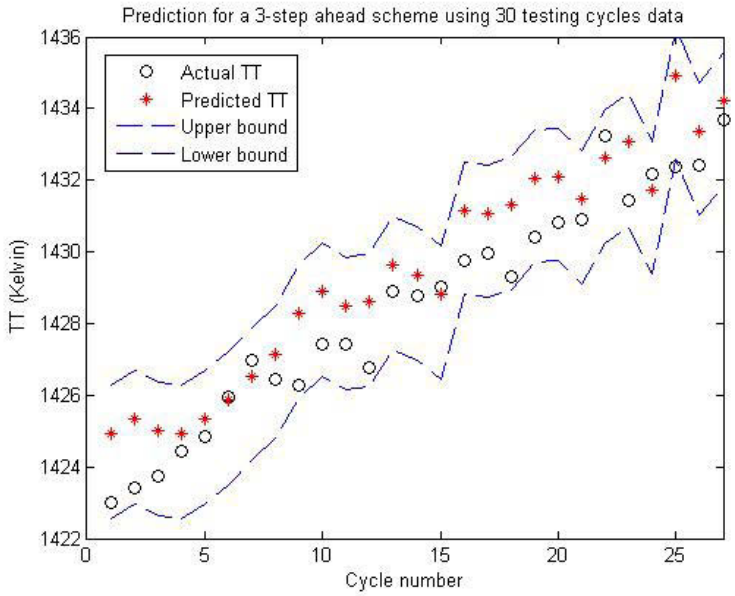

Figure 4.27: Actual vs. predicted TT for a 3% fouled compressor considering prediction bounds (3-step ahead).

In the next two cases we will try to improve the prediction results by changing the network parameters and then increase the prediction horizon.

FI = 3%: Case 2

As changes in the turbine temperature are more significant due to higher levels of degradation, the results obtained in the previous case deemed satisfactory but to go further in the prediction horizon, after some trials and errors it was concluded that one needs to provide the network with more examples. This time we feed the network

Table 4.13: Prediction error for FI=3% case 2.

μ_{ae}	-0.410 (K)
σ_{ae}	5.369 (K)
$rmse$	5.187 (K)

with 80% of the data for training and verify if the prediction results for five-step-ahead is improved. Furthermore , one more delayed version of the output is needed ($d = 3$). This results in a 5-5-1 NN. The results are depicted in Figure 4.28. Furthermore the error is shown in Figure 4.29.

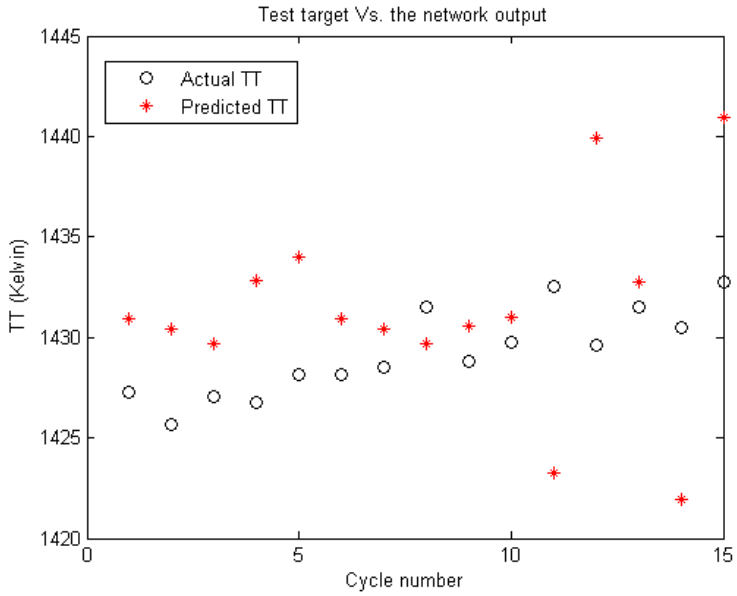

Figure 4.28: Actual vs. predicted TT for a 3% fouled compressor (5-step ahead).

In this case the network has distinguished the trend. The errors have decreased after the changes are in the portion of training data made as can be verified quantitatively in Table 4.13.

To overcome the problem of uncertainty associated with prediction, the lower and upper bound are found and depicted in Figure 4.30. The results shown in this figure confirm that we have achieved a good prediction result. Three of the 15 data are out

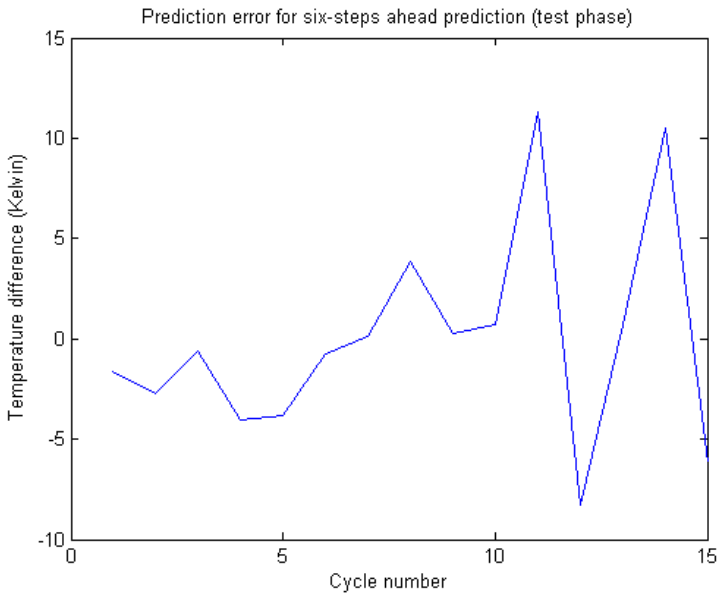

Figure 4.29: Temperature prediction error for a 3% fouled compressor (5-step ahead).

the band made by the lower and the upper prediction bounds. This correspond to only 20% of the data points.

FI = 3%: Case 3

In the third case which is the last case associated with fouling scenarios, the developed recurrent neural network is tested to find a larger prediction horizon while yielding satisfactory results (A 5-5-1 network). Through examining many cases we have concluded that eight-steps ahead prediction is feasible implying that from the 20 data point available for testing, if one starts with the first point then the network will predict 8 steps further and thus provides 12 results for prediction. We have compared the actual and the predicted turbine temperatures in Figure 4.31. The prediction errors for the test data are shown in Figure 4.32. In this case the mean of the error is 1.410 K, the standard deviation is 1.100 and the *rmse* is equal to 1.760 K as summarized in Table 4.14.

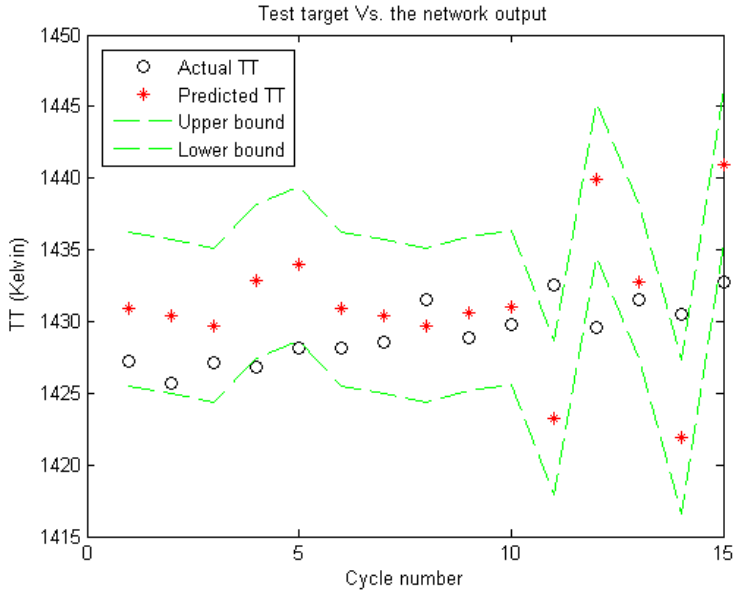

Figure 4.30: Actual vs. predicted TT for a 3% fouled compressor considering prediction bounds (5-step ahead).

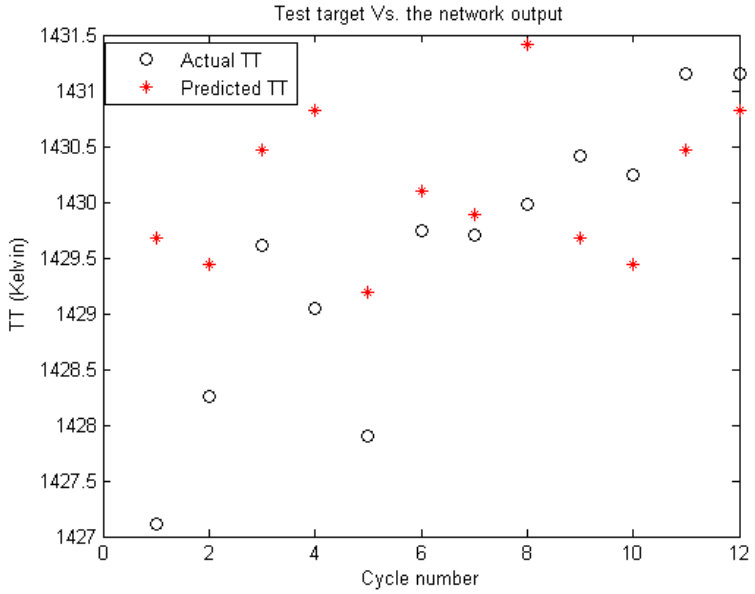

Figure 4.31: Actual vs. predicted TT for a 3% fouled compressor (8-step ahead).

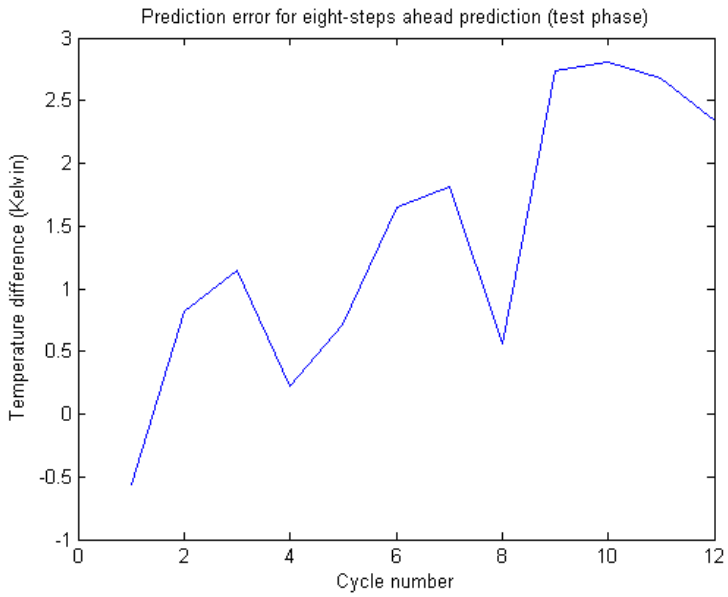

Figure 4.32: Temperature prediction error for a 3% fouled compressor (8-step ahead).

Table 4.14: Prediction error for FI=3% case 3.

μ_{ae}	1.410 (K)
σ_{ae}	1.100 (K)
$rmse$	1.760 (K)

Determining two bounds for the prediction results and managing uncertainty is even more critical as the prediction steps increase through the cycles as the probability of uncertainty modelling becomes higher. Again we have predicted and shown these in Figure 4.33. Note that 8 of the 12 predicted data points are inside in the bands. This corresponds to more than 65% of them.

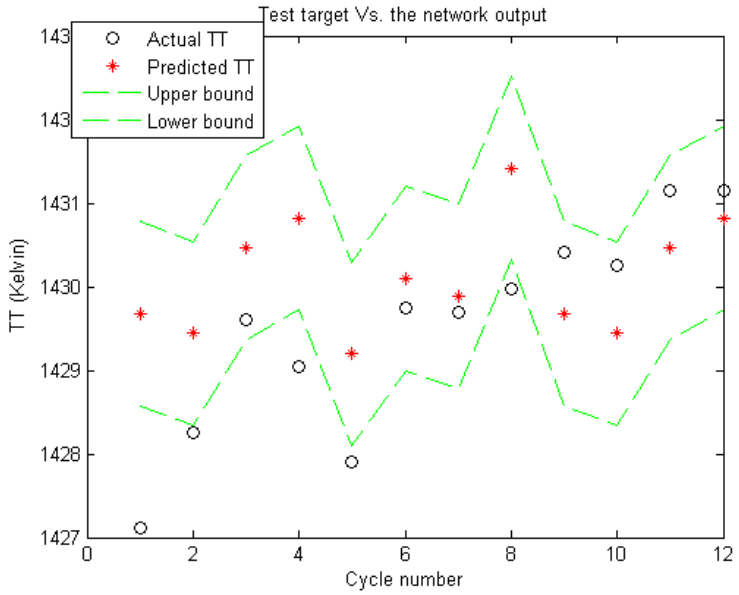

Figure 4.33: Actual vs. predicted TT for a 3% fouled compressor considering prediction bounds (8-step ahead).

Through many trial and error attempts and network adjustments it is concluded that the network is capable of eight-cycles-ahead prediction. To go beyond this horizon one needs a higher percentage of the data for training, besides the prediction error has an increasing trend which can be regarded as a sign for the limit of the step-ahead possible. Moreover, as we considered flight cycles for our prediction horizon, this number of step-ahead is good enough and gives enough time to the operators to decide about the necessary maintenance actions while the engine condition is still safe. In the next sections, we study and discuss the cases in which the turbine is affected by the erosion.

4.3.2 Simulation Results for Erosion Scenarios

In this section we will focus on erosion and its effect on the aircraft engine. Erosion is the removal of material from the flow path components with hard particles. Erosion mostly occurs in the turbine component and can cause aerodynamic changes in the behaviour of blades. We will also consider erosion to occur in the turbine section. Pressure losses, performance degradations and even blade failure are consequences of the erosion. It is very important to track the effects of the erosion on the gas path measurements and be able to predict them.

Erosion changes the engine health parameters. It will cause the turbine efficiency and mass flow rate to drop. It is assumed that a 2:1 linear relationship exists between the turbine efficiency and the mass flow rate, respectively. To quantify these effects we use the EI that is explained in Chapter 3. Changes in the efficiency and the mass flow rate due to erosion increase the turbine exit temperature (TT). In the following scenarios we will use RNNs for predicting the turbine temperature through various simulations. Three levels of erosion constitute as basis for our scenarios. The process of making the scenarios and cases are similar to the ones in the fouling cases.

4.3.2.1 First Scenario: EI = 1%

If erosion occurs and stays in the turbine it can change the spool speed, turbine temperature, etc. The rate of these changes were studied and validated with the GSP software in Chapter 3. The simulations for a system under erosion have been performed for 200 take-off cycles. In the take-off mode, since the engine is operating from the ground idle condition to the maximum level of fuel, the degradation initiation and propagation is more significant and therefore are studied here. For EI = 1%, this 1% is equivalent to 1% drop in the turbine efficiency and 0.5% drop in the turbine mass flow rate. Recurrent neural networks are employed here to learn this evolution

trough flight cycles. Thus it is not necessary to save and use all the data points from the 200 cycles. Degradations have low dynamics and do not change the system abruptly or significantly in only one cycle. Hence, instead one can pick the same time from each flight cycle and put them in a vector. From this vector a portion will serve as the training data set and a portion for testing purposes.

The turbine temperature evolution for a 1% eroded turbine during 200 cycles is depicted in Figure 5.28. The effect of the measurement noise on the data is clear.

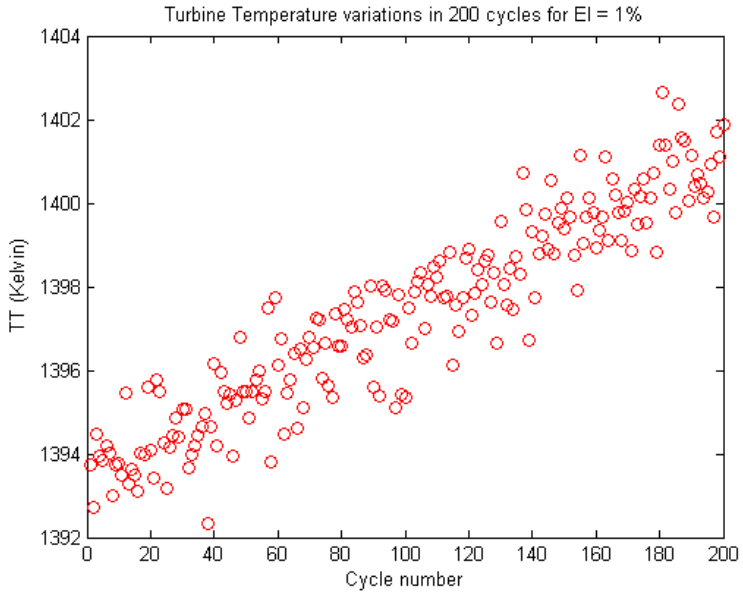

Figure 4.34: Turbine temperature variations due to a 1% eroded turbine.

EI = 1%: Case 1

Given an appropriate data set one can train the neural network for the following objective. If a 1% erosion occurs in the turbine and remains there implying that no maintenance is performed, what would be the engine condition in some flights ahead. Once the turbine temperature is predicted for some flights ahead, one can then decide if the next flights will be safe or the temperature has reached certain thresholds that makes it necessary to take the engine off-line for maintenance.

First we start training our recurrent neural network that is provided with a global feedback path from the network output to the network input. Using the present value of the input which is the fuel flow rate, the current value of the temperature which is the network output and three of its previous values ($d = 3$) the network is trained. In the first case 50% of the data shown in Figure 5.28 are used to train a 5-5-1 RNN network. This implies that we have two inputs namely the current value of the fuel flow rate and past value of the turbine temperature (TT), five neurons in the hidden layer and one output which is the turbine output temperature. The actual versus the predicted data for a five-step-ahead prediction are compared in Figure 4.35. It is understood from the figure that that network is not able to learn the turbine

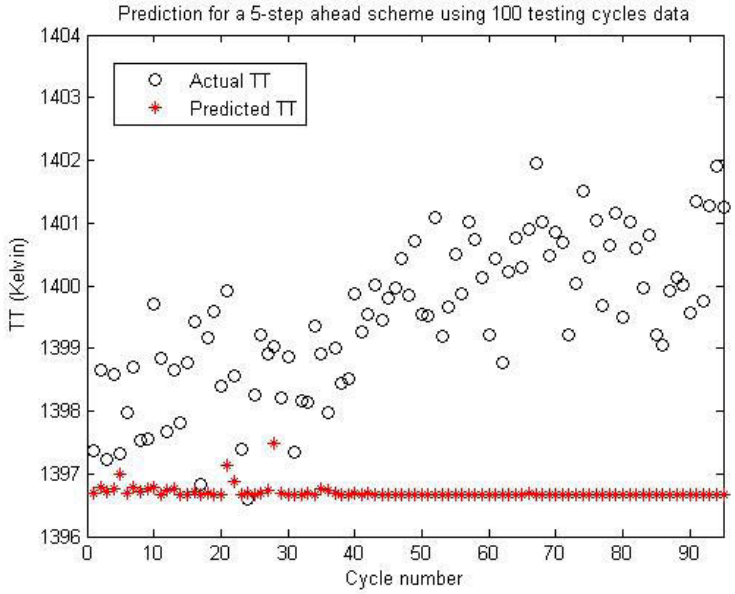

Figure 4.35: Actual vs. predicted TT for a 1% eroded turbine (5-step ahead).

temperature trend for a 1% eroded turbine. Especially as one proceeds further in time the predicted points are not updated and they are remaining constant. This network is unable to perform the prediction task and we either have to provide the network with more examples or change the network parameters. The resulting error is depicted in Figure 4.36. The increase in the error confirms that this network is

Table 4.15: Prediction error for EI=1% case 1.

μ_{ae}	6.664 (K)
σ_{ae}	8.537 (K)
$rmse$	8.703 (K)

unable to predict the turbine temperature.

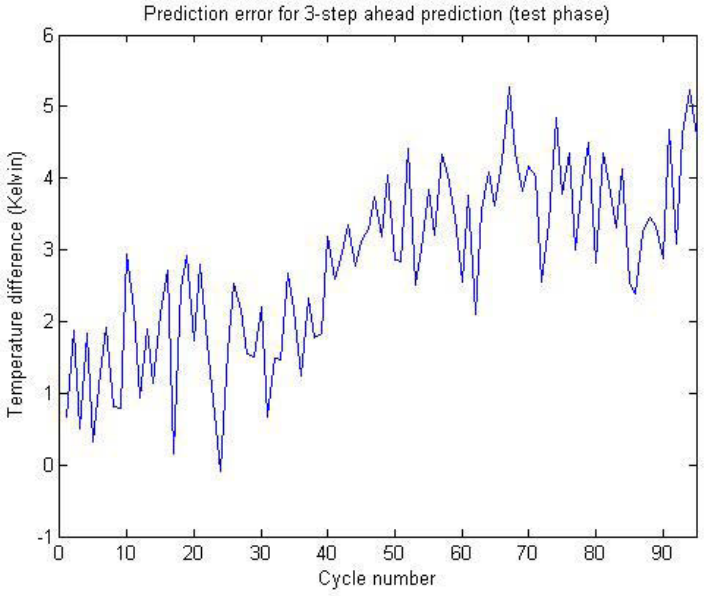

Figure 4.36: Temperature prediction error for a 1% eroded turbine (5-step ahead).

Similar to the fouling cases, we will calculate some error statistics which help us judge the prediction performance. We obtain the mean (μ) and the standard deviation (std) of the error in addition to the root mean square of the instantaneous error ($rmse$). For the *Case 1* the mean of the prediction error is equal to 6.664 K, the standard deviation is 8.537 K, and $rmse$=8.703 K. The values are summarized in Table 4.15. We will increase the number of input data in the next case.

EI = 1%: Case 2

Our recurrent neural network requires more examples to learn the dynamic of the turbine temperature increase when the turbine is affected by erosion. In this second case, 70% of the data vector is assigned for network training. Moreover, the network architecture is changed to 5-10-1 ($d = 3$ and number of neurons in the hidden layer is set to 10). The comparison results are shown in Figure 4.37 where the *circles* represent the actual data and the *stars*represent the predicted values (neural network output).

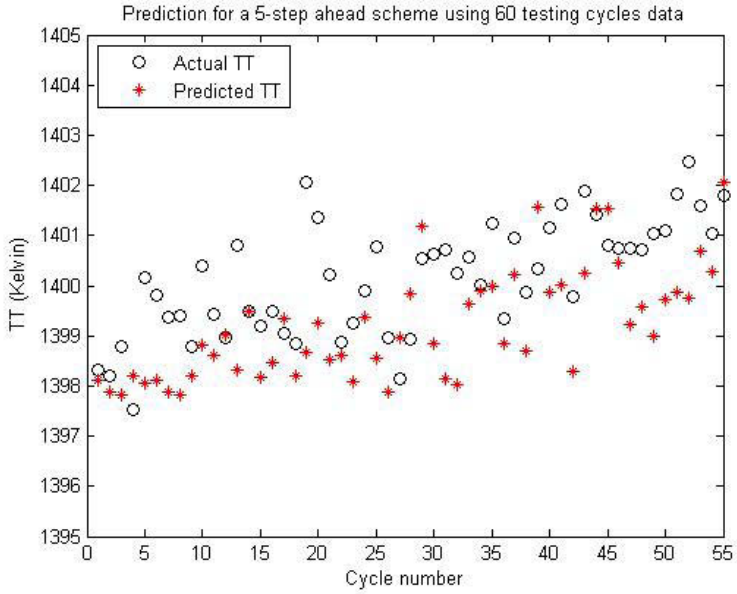

Figure 4.37: Actual vs. predicted TT for a 1% eroded turbine (5-step ahead).

Furthermore, the prediction error for the test data is shown graphically in Figure 4.38 and as expected the error level increases as the prediction horizon increases. The test error mean, standard deviation and *rmse* are -0.198 K, 2.735 K, and 2.718 K respectively which are also tabulated in Table 4.16. The errors are sufficiently small enough (less than 1% error).

The importance of uncertainty management and defining some bounds for the

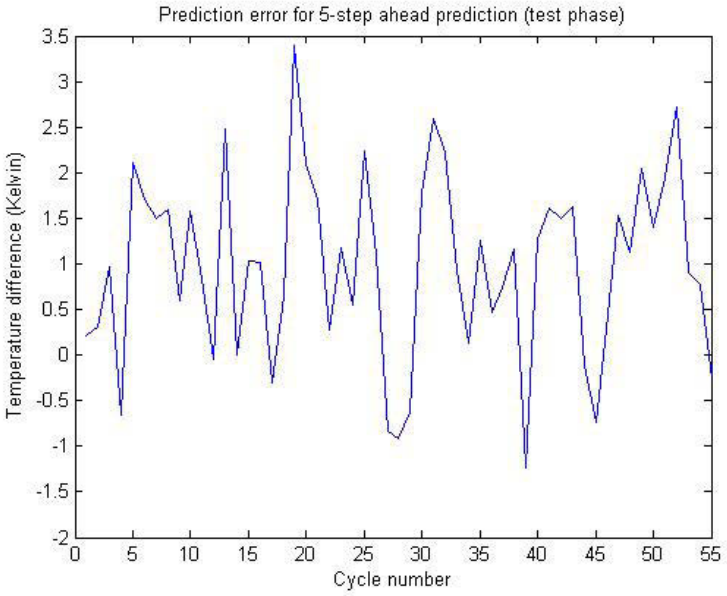

Figure 4.38: Temperature prediction error for a 1% eroded turbine (5-step ahead).

Table 4.16: Prediction error for EI=1% case 2.

μ_{ae}	1.016 (K)
σ_{ae}	1.971 (K)
$rmse$	1.599 (K)

prediction instead of just relying on the point prediction has been highlighted in the previous section. In Figure 4.39, the actual data, the network output and the predicted bounds are shown in the same diagram. The graph shows that most of the data points are within the bounds and this implies an acceptable five-step-ahead prediction (87% of the data points are inside the area made by the lower and the upper prediction bounds).

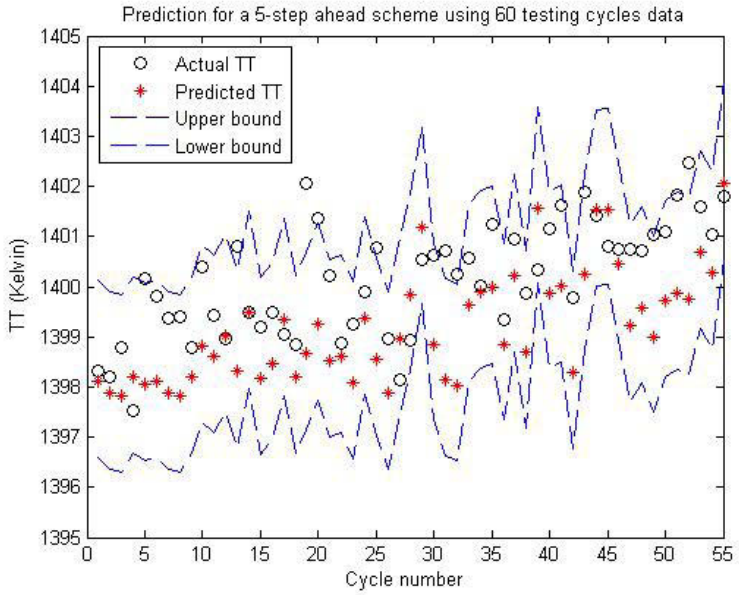

Figure 4.39: Actual vs. predicted TT for a 1% eroded turbine considering prediction bounds (5-step ahead).

To summarize put the results of this case both the error results and the prediction bounds show that the 5-10-1 RNN is suitable for predicting the turbine temperature changes when it is due to 1% erosion. Thus we consider this network for our prediction and will attempt to extend the number of cycles ahead in the next case.

Remark. One should note that in order to be able to judge the neural network performance and the goodness of the prediction, we use both the prediction bands and the quantitative error statistics. First of all we verify if most of the data points are in the predicted band. Then, we study the error mean, standard deviation and

the mean-squared error. If all of these values are small enough i.e. the absolute mean plus three times the standard deviation (this is a common measure) is less than 1% of the nominal value of the temperature, we conclude that the prediction error is less that 1%. In the other words, the network is capable of predicting the turbine temperature for that certain level of the degradation with 99% accuracy. This 1% measure is being studied in all the cases and scenarios, it does not depend on the level of degradation and the prediction preciseness is important for us.

EI = 1%: Case 3

In this case, we assume that the turbine is still having the same level of erosion. We examine the network to determine the most achievable step-ahead for the turbine temperature prediction using the same 5-10-1 network architecture. We test our RNN for an eight-step-ahead temperature prediction. The prediction results are depicted in Figure 4.40. Moreover, the errors between the actual and the predicted temperatures are shown in Figure 4.41.

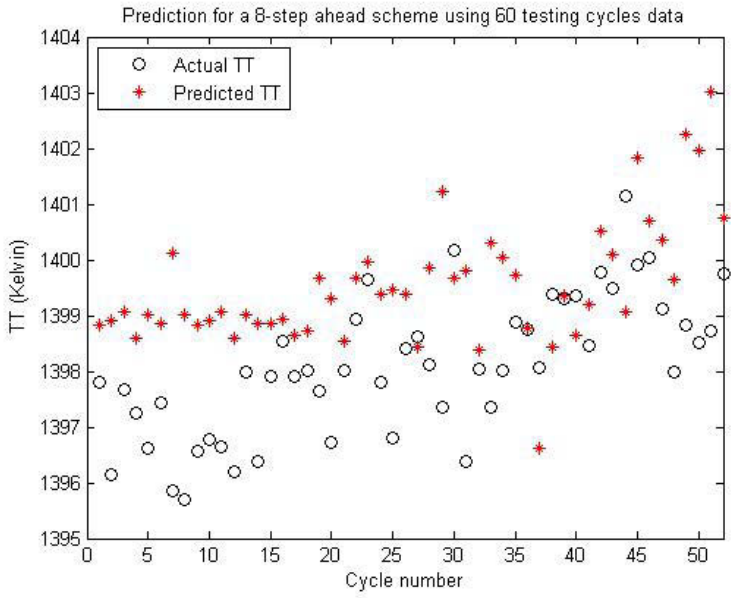

Figure 4.40: Actual vs. predicted TT for a 1% eroded turbine (8-step ahead).

Table 4.17: Prediction error for EI=1% case 3.

μ_{ae}	1.578 (K)
σ_{ae}	2.410 (K)
$rmse$	2.511 (K)

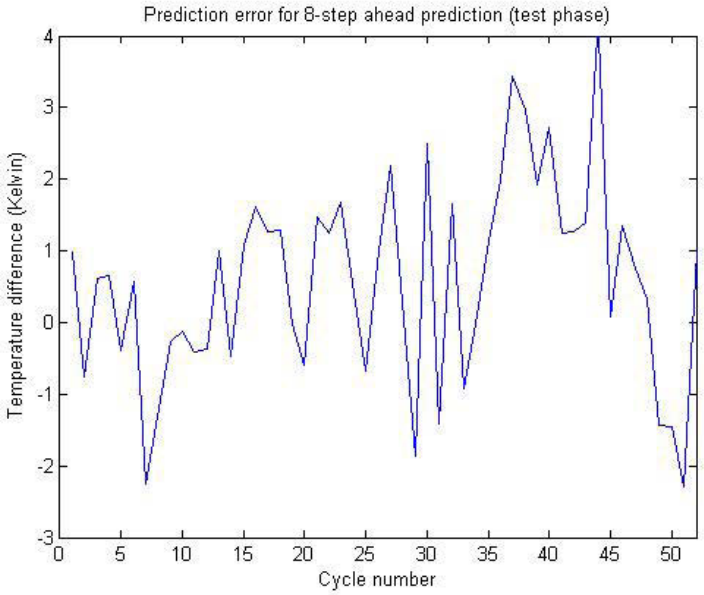

Figure 4.41: Temperature prediction error for a 1% eroded turbine (8-step ahead).

We have also found the error statistics quantitatively and presented them in table 4.17. The error is still within the reasonable and acceptable ranges. This demonstrates that we can rely on this network for an eight-step-ahead prediction.

Finally, we have also predicted two lower and upper prediction bounds to overcome uncertainties associated with the prediction and determine if the data points are within these ranges. The bounds and the predicted temperatures for this case are depicted in Figure 4.42. One can see that 4 data points are not inside the bands and in other words 92% are within these bands.

It follows that error results are quite good. In the next case we try to find the largest cycle possible for which an acceptable prediction horizon can be achieved.

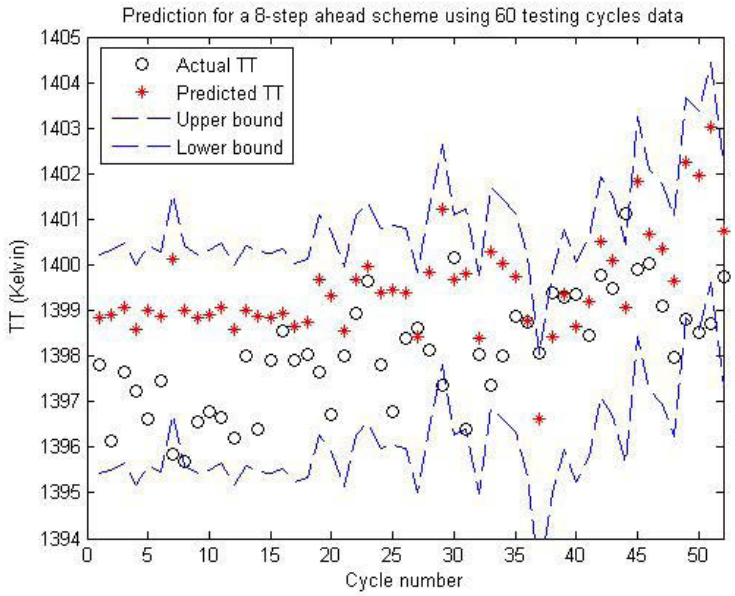

Figure 4.42: Actual vs. predicted TT for a 1% eroded turbine considering prediction bounds (8-step ahead).

EI = 1%: Case 4

For *Case 4* we have conducted many simulations to find the maximum allowable time step horizon for the turbine temperature prediction. Two performance measures have to be monitored carefully. The first is the set of error statistics such as the mean and the *std* of the prediction error and the second one is the set of prediction bounds. It is both important that the errors are low and also the actual values lie within the prediction bounds. After various trials and errors we conclude that for the case of 1% erosion in the turbine component, with adjusting the number of delays to 4 ($d = 4$), the 6-10-1 RNN gives us the temperature in the 15^{th} cycle from the present time. The results are shown in the following Figures 4.43 - 4.45 and Table 5.11.

Considering the error levels and the position of the data with respect to the prediction bounds (77% of them are inside the bands), a 15-step ahead is the maximum number of cycles that can be projected satisfactorily. We could obtain better results

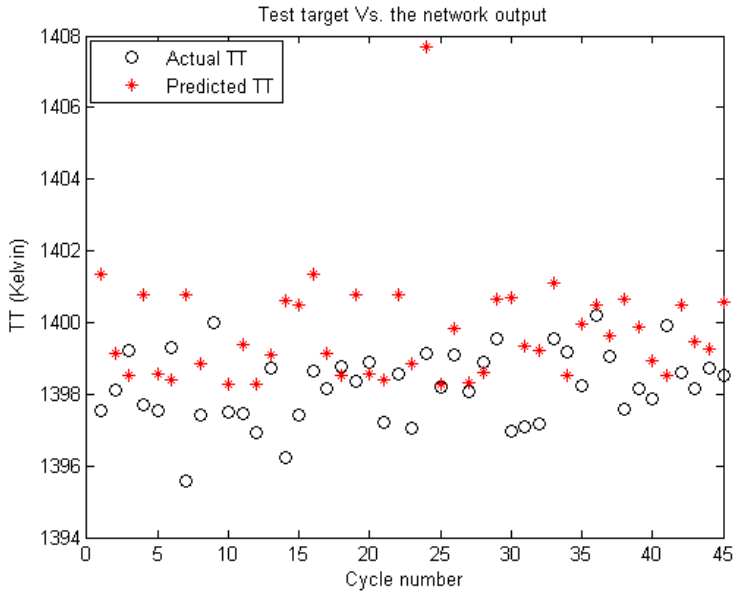

Figure 4.43: Actual vs. predicted TT for a 1% eroded turbine (15-step ahead).

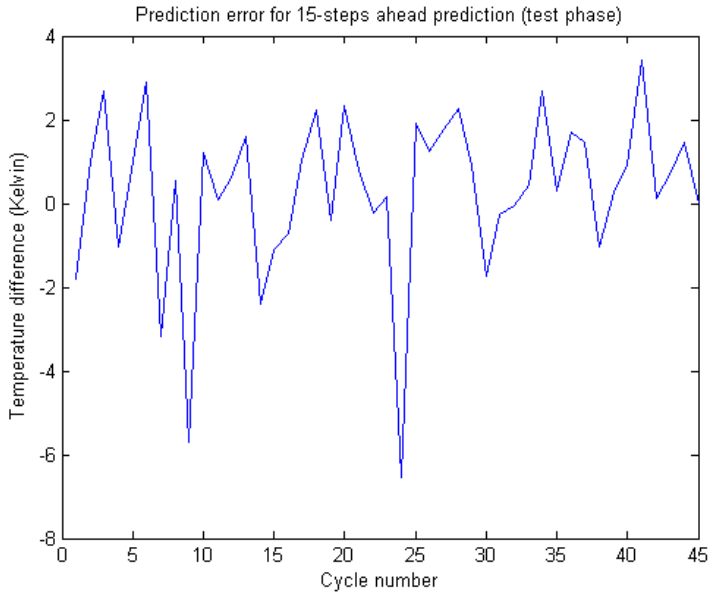

Figure 4.44: Temperature prediction error for a 1% eroded turbine (15-step ahead).

Table 4.18: Prediction error for EI=1% case 4.

μ_{ae}	2.301 (K)
σ_{ae}	3.989 (K)
$rmse$	3.990 (K)

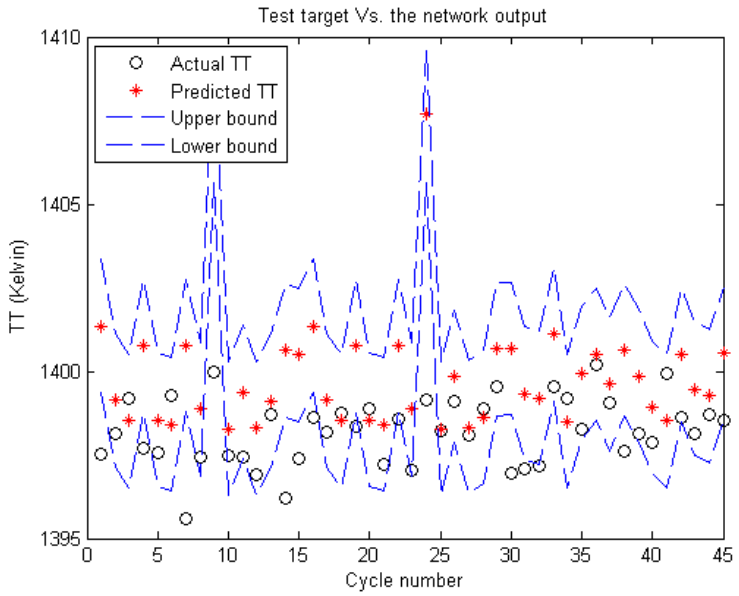

Figure 4.45: Actual vs. predicted TT for a 1% eroded turbine considering prediction bounds (15-step ahead).

by increasing the number of delays but it will cost us computational complexity and thus we stop going further at this point.

4.3.2.2 Second Scenario: EI = 2%

In this scenario we follow the procedure that was followed in the first scenario with EI = 1%. First we have to generate the proper data. The standard take-off duration is assumed to be 20 seconds. We run the simulations 200 times (200 cycles) and pick up the 12^{th} second of each take-off and construct the data set. We assume that a 2% EI is effective in the entire simulation. This is equivalent to a 2% drop in the efficiency and a 1% drop in the turbine mass flow rate. We also assume that the erosion levels remain the same. If the erosion remains in the system it causes and increases the effects in the turbine temperature. We would like to train the neural network to predict the temperature of an eroded turbine for certain steps ahead. Our data set considering the measurement noise are depicted in Figure 5.38 where all the temperatures are measured in Kelvin.

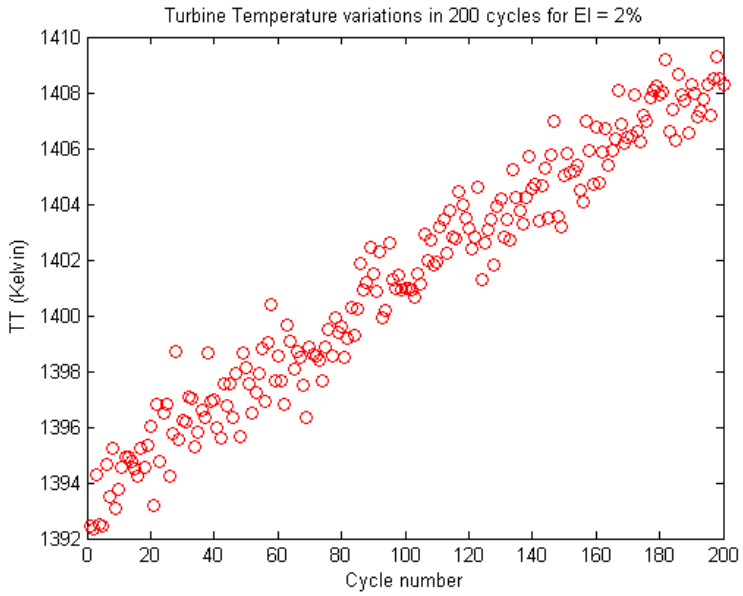

Figure 4.46: Turbine temperature variations due to an eroded turbine with EI= 2%.

EI = 2%: Case 1

In the first case that we investigate the goal is to achieve a reliable five-step-ahead prediction with the same "optimum" network that was derived in the previous scenario. We have a 5-10-1 recurrent neural network where 70% of our data set are given as training examples to the neural network. We use the rest of the data to test the neural network as shown in Figure 4.47. However in this case to achieve good results we had to use 80% of the data points for training purposes. The actual and

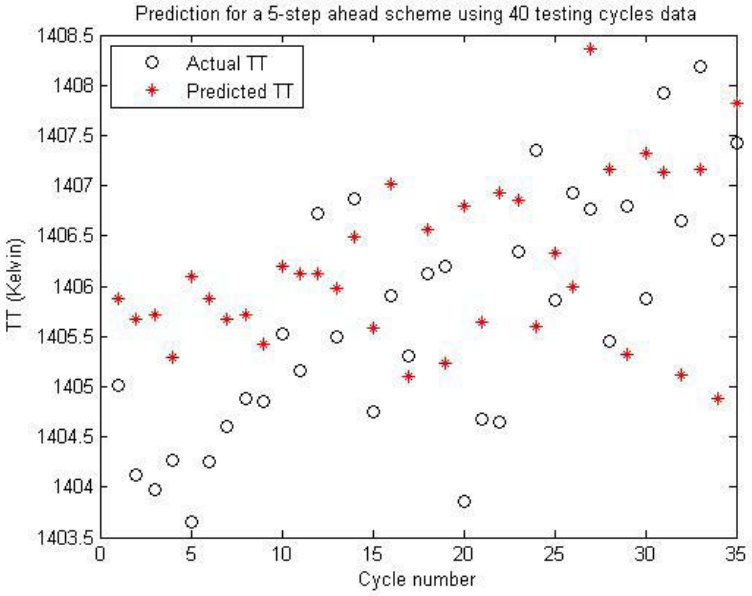

Figure 4.47: Actual vs. predicted TT for a 2% eroded turbine (5-step ahead).

the predicted data deviate from each other as can be seen in Figure 4.48 where the error increases.

The predicted data points are in average $1.5K$ different from the actual values which is a good prediction result as the nominal turbine temperature during the take-off is about $1400K$. The statistics of the performance are shown in Table 4.19

In addition to the error values, the prediction bounds confirm that with our RNN architecture, a five-step-ahead prediction is possible for the turbine temperature. The

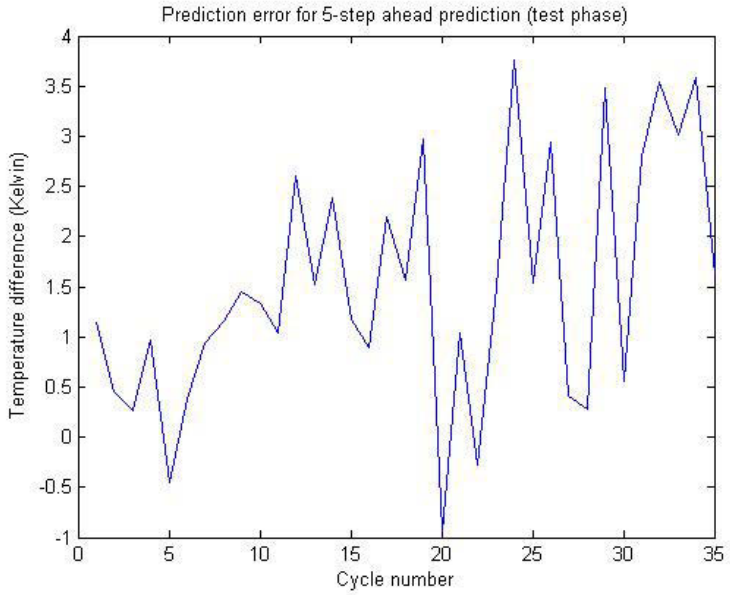

Figure 4.48: Temperature prediction error for a 2% eroded turbine (5-step ahead).

Table 4.19: Prediction error for EI=2% case 1.

μ_{ae}	1.507 (K)
σ_{ae}	1.216 (K)
$rmse$	1.926 (K)

bounds when compared to the predicted and actual temperatures are depicted in Figure 4.49. One can see that 80% of the data points are inside the created bands.

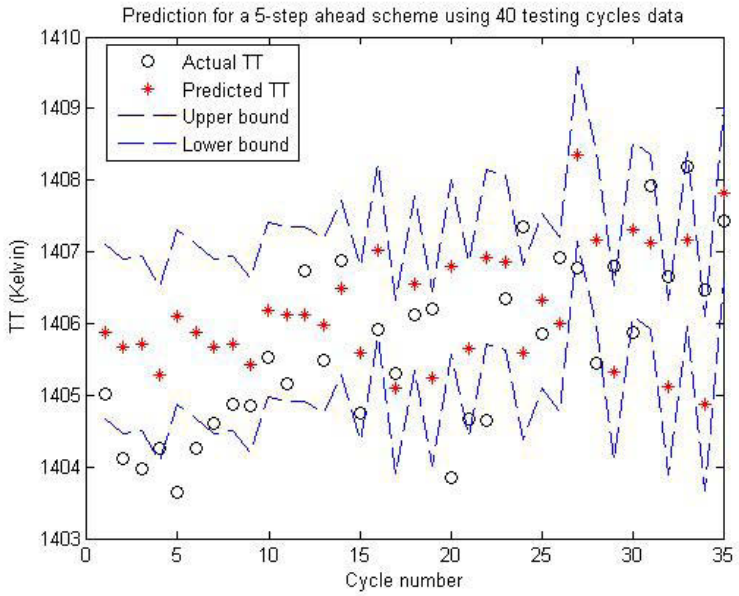

Figure 4.49: Actual vs. predicted TT for a 2% eroded turbine considering prediction bounds (5-step ahead).

FI = 2%: Case 2

We finish the second scenario which considers EI to be equal to 2% by investigating the maximum step-ahead horizon for the prediction. We use the same network architecture as in Case 4 of the previous scenario. d is set to four which gives a 6-10-1 NN. We want the mean, standard deviation and *rmse* of the error to be as low as possible, implying that the predicted point are close to the real temperature as derived from our SIMULINK model. Moreover, we want the predictions to be inside the bounds to overcome uncertainty. The comparison results are presented in Figure 4.50.

The error level is within the range (less than 1% error) when compared to the nominal value for the turbine temperature in the take-off. The error is depicted in

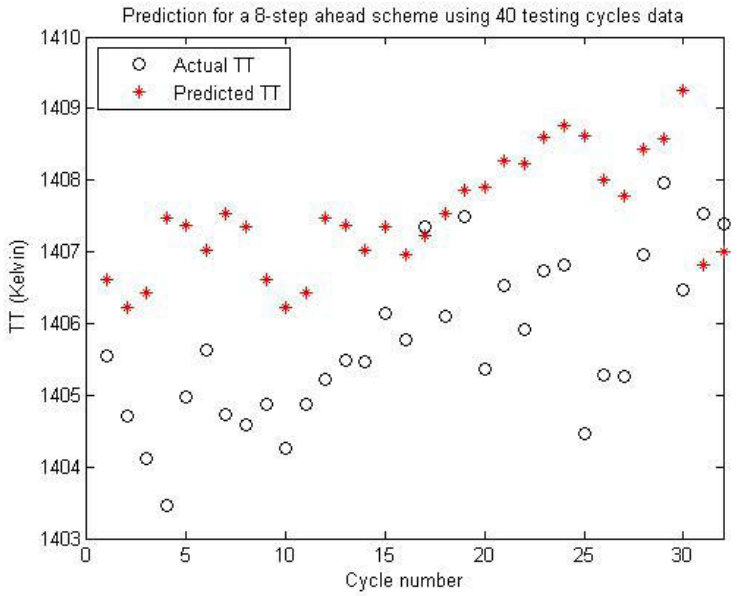

Figure 4.50: Actual vs. predicted TT for a 2% eroded turbine (8-step ahead).

Table 4.20: Prediction error for EI=2% case 2.

μ_{ae}	0.325 (K)
σ_{ae}	2.085 (K)
$rmse$	2.091 (K)

Figure 4.51 for the network testing phase. Moreover, the mean, *std* and *rmse* of the error are found in Table 4.20.

The bounds are predicted and depicted in Figure 4.52. Defining the bounds is very important especially when we are taking larger steps ahead. Since the error level is low and also most of the points stay between the lower and the upper prediction bounds (22 of the 32 data point which corresponds to 70% of them) one can conclude that the trend has been learned satisfactorily. A reliable eight-step-ahead prediction can be performed with our RNN when 2% erosion is effective in the turbine.

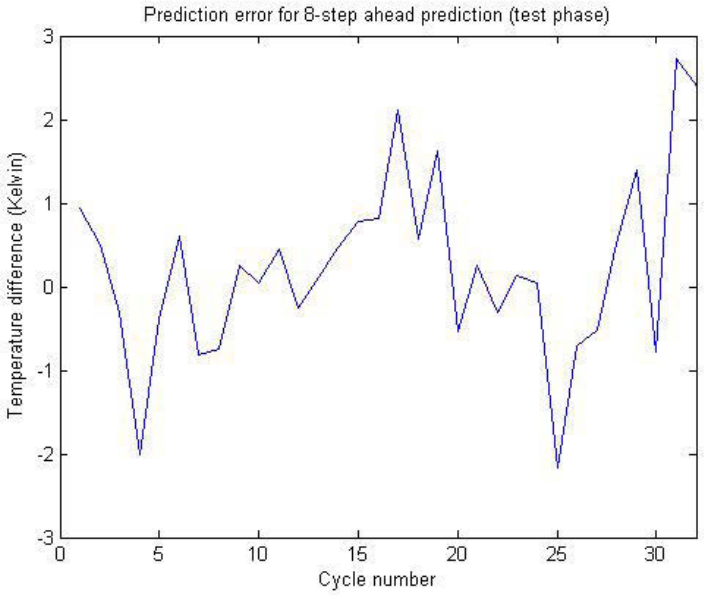

Figure 4.51: Temperature prediction error for a 2% eroded turbine (8 step-ahead).

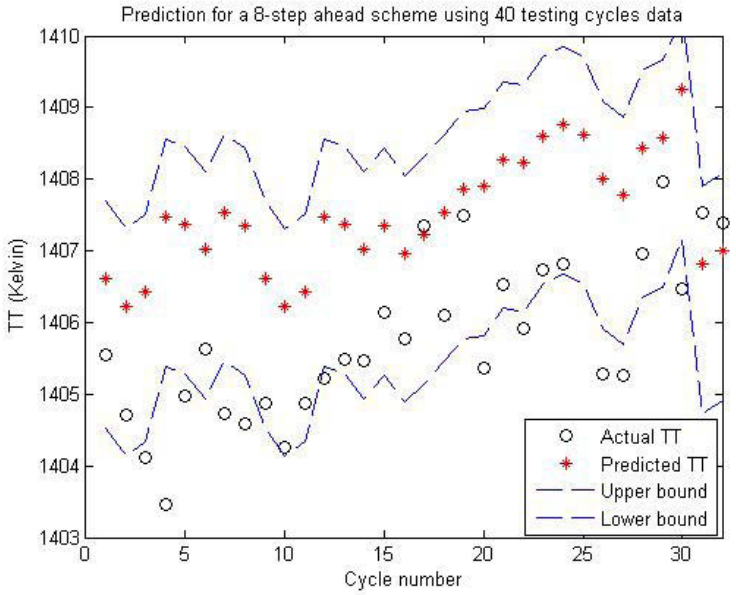

Figure 4.52: Actual vs. predicted TT for a 2% eroded turbine considering prediction bounds (8 step-ahead).

4.3.2.3 Third Scenario: EI = 3%

In our final scenario we assume that a 3% erosion has occurred in the turbine section. This is equivalent to 3% drop in the turbine efficiency and 1.5% drop in the turbine mass flow rate. We need to generate a proper data set as it can affect the performance of the neural network. We have considered the measurement noise as well. The resulting data set are depicted in Figure 5.45. In the following we will investigate our

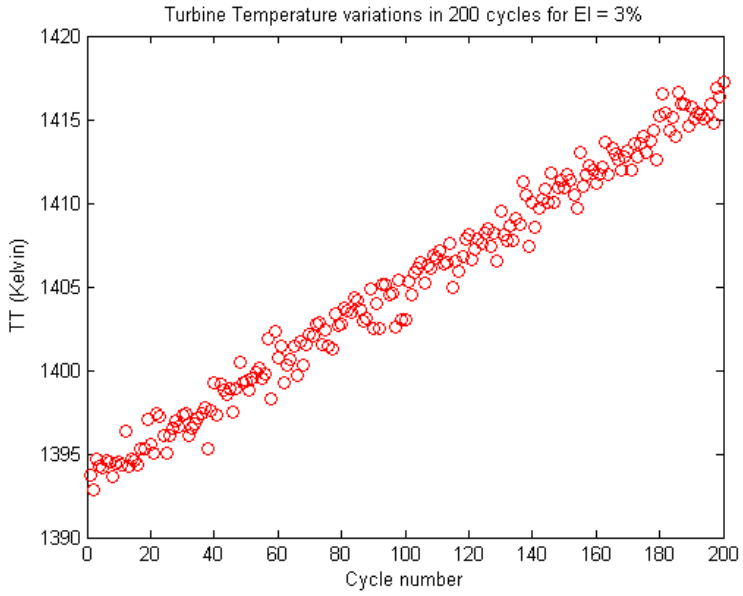

Figure 4.53: Turbine temperature variations due to a 3% eroded turbine.

final scenario.

EI = 3%: Case 1

In the fist case, we assume that a 3% erosion has occurred in the turbine. We have modelled the erosion by changing the engine health parameters, varying the health parameters, and changing the gas path measurements. The turbine temperature is the best candidate for prognostics study as it contains important information about the changes in the entire system. This level of erosion is fully effective in the system

at the end of 200th cycle. We will start training our 5-10-1 RNN implying 3 delayed outputs plus the turbine temperature itself as the out pit and the mass flow rate as the input for this case. Once the training phase is completed, in the testing phase we compare the network output with the obtained SIMULINK model output. The results are shown in Figure 4.54.

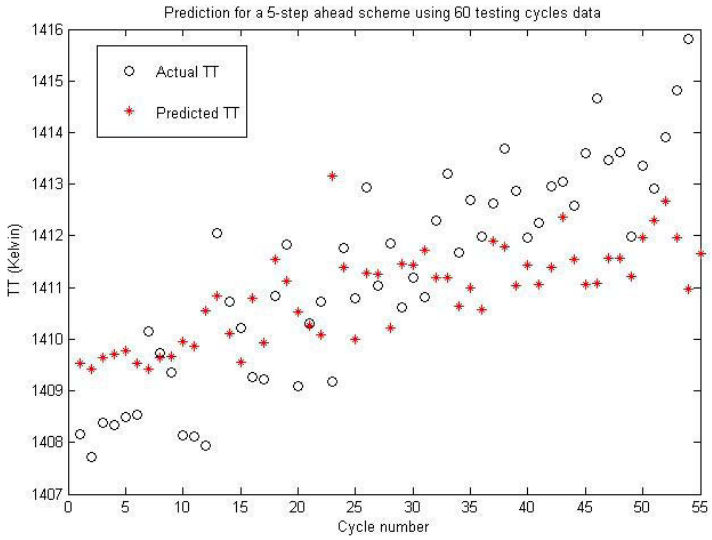

Figure 4.54: Actual vs. predicted TT for a 3% eroded turbine (5-step ahead).

The difference between the network output and the real data at each point is depicted in Figure 4.55. The learning is more accurate at the beginning but as one reaches to the end the error levels increase and this shows the need for updating the information that we feed to the network.

The error statistics are provided in Table 4.21 and the prediction bounds are shown in Figure 4.56. Combining these two measures one can see that error levels are very low and most of the data points (78% of the entire data set) are within the bounds which show that an effective 5-step ahead can be performed with our RNN.

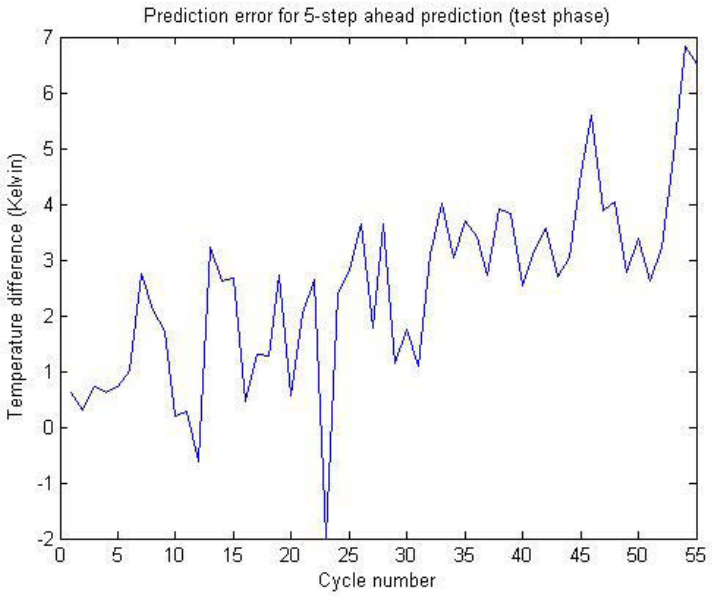

Figure 4.55: Temperature prediction error for a 3% eroded turbine (5-step ahead).

Table 4.21: Prediction error for EI=3% case 1.

μ_{ae}	2.495 (K)
σ_{ae}	1.674 (K)
$rmse$	2.996 (K)

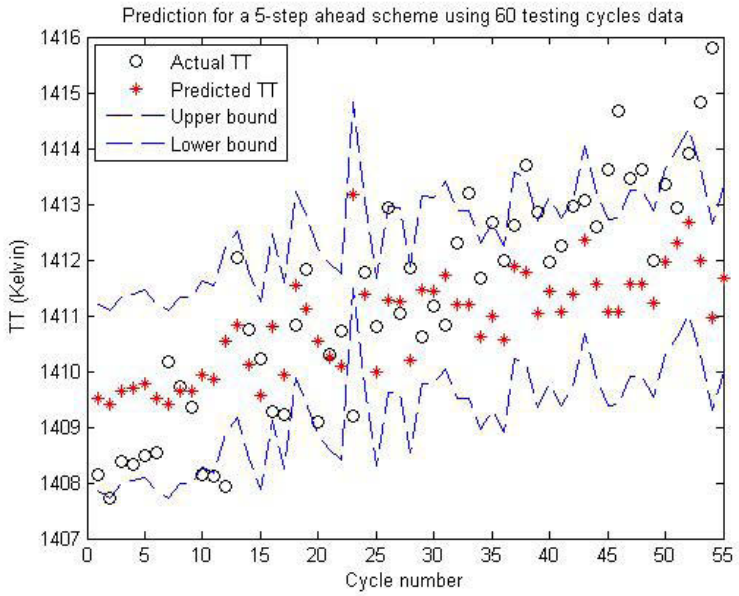

Figure 4.56: Actual vs. predicted TT for a 3% eroded turbine considering prediction bounds (5-step ahead) .

Table 4.22: Prediction error for EI=3% case 2.

μ_{ae}	0.747 (K)
σ_{ae}	2.349 (K)
$rmse$	2.339 (K)

EI = 3%: Case 2

In the last case that we present in this chapter, our RNN will be tested to find its limit for cycles ahead prediction horizon. We use the 6-10-1 RNN with global feedback from the network output to verify the results for a five-step ahead prediction when the erosion index is equal to 3%. After feeding the network with 80% of the data for training, the following results were obtained in the testing phase as shown in Figure 4.57.

The error results for ten-steps-ahead prediction are shown in Figure 4.58. Moreover, the error mean, *std* and *rmse* are tabulated in Table 4.22.

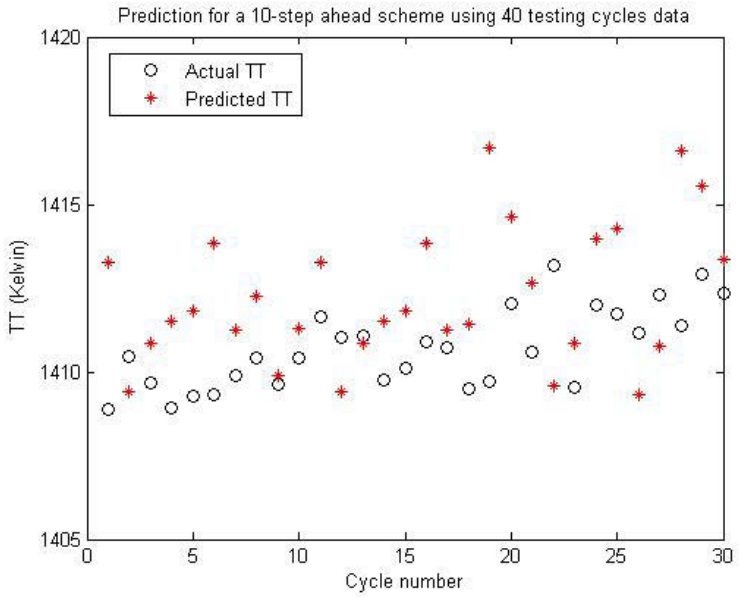

Figure 4.57: Actual vs. predicted TT for a 3% eroded turbine (ten-step ahead).

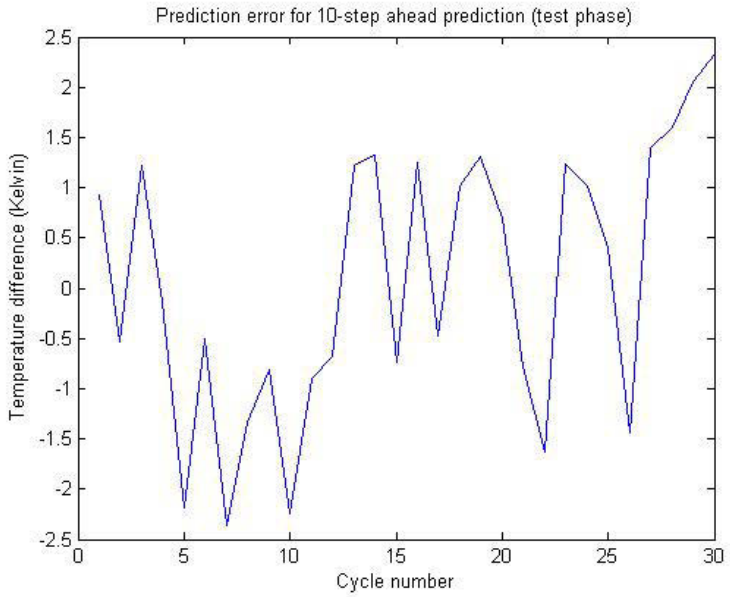

Figure 4.58: Temperature prediction error for a 3% eroded turbine (ten-step ahead.

The error results are reasonable. We also have to check the prediction bounds to be able to comment on the prediction performance (Figure 4.59). One can see that 60% of the data are within the bands created by the upper and the lower prediction bounds. The ten-step-ahead prediction is an accurate prediction and we can ensure that on average our prediction of the turbine temperature is quite close to the real value with only a 1% error.

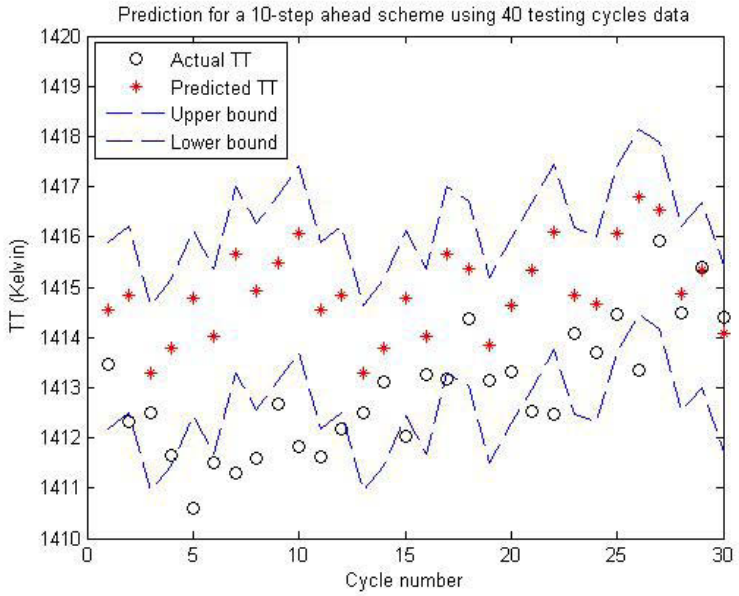

Figure 4.59: Actual vs. predicted TT for a 3% eroded turbine considering prediction bounds (ten-step ahead).

In several scenarios that we have presented in this chapter the prediction results using our recurrent neural network are showing promising results. For a gas turbine engine that is due to either fouling or erosion, if we provide the network with some data points from the temperature evolution, the network will be able to predict the temperature values for future cycles with a good error percentage. Knowing the future values of the temperature allows one to schedule the maintenance based on the predicted condition of the engine.

4.4 Summary of the Simulation results

In this section we Summarize the simulation results obtained in the previous section, in a tabular form. In all of the simulation results, We studied the results both qualitatively and quantitatively. We compared the predicted values which were the output of our recurrent NN with the test data vector that we had kept for comparison and network testing purposes. In addition, we verified if the data points were within the bands created by the upper and the lower prediction bands. Finally, we found the prediction error which is defined as the difference between the real and the predicted turbine temperatures. The error is found for each point and the number of error points depend on the size of the test vector and number of cycles ahead predicted. Utilizing the points, Some error statistics are calculated. Error mean (μ_{ae}), error standard deviation (σ_{ae}) and the mean squared-error ($rmse$) were calculated and presented in tables as well as the network structure for each case.

These quantitative measures were calculated for all Scenarios and cases for different degradation types, different degradation levels (1% to 3%) and different steps ahead. These error statistics helped us judge if the prediction was reliable (less than 1% error). To achieve satisfactory prediction result, We had to change the networks parameters such as the number of delays fed into the network input. l is the number of prediction steps ahead, d is the number of output delays fed back to the network input.

4.4.1 Summary of the results for fouling scenarios

The results for the fouling scenarios (a fouled compressor) are presented in the following tables. We do not go beyond 3% of fouling as engine washing is recommended after this level. It is understood form the tables that if we want to keep the error low and increase the cycles ahead in time, we have to increase the number of neurons

Table 4.23: Summary of the results for FI=1% scenarios.

FI = 1%	Network Structure	μ_{ae}	σ_{ae}	$rmse$
d = 2 / l = 3	4-5-1	1.177	1.073	1.579
d = 2 / l = 3	4-5-1	0.177	1.384	1.343
d = 4 / l = 6	6-5-1	1.968	1.829	2.658
d = 4 / l = 10	6-5-1	-0.338	1.810	2.786

Table 4.24: Summary of the results for FI=2% scenarios.

FI = 2%	Network Structure	μ_{ae}	σ_{ae}	$rmse$
d = 2 / l = 3	4-5-1	1.600	1.948	2.717
d = 4 / l = 10	6-10-1	-1.815	3.632	5.792

in the hidden layer as well as increasing the number of output delays d given to the network as input.

4.4.2 Summary of the results for erosion scenarios

In this subsection we summarize the prediction results when the engine is under different levels of erosion. Erosion in the turbine causes efficiency drop and increases the mass flow rate by a linear relation of 2:1. By studying these three tables one can see that to achieve more steps ahead, more delayed outputs need to be fed back to the RNN input otherwise the error level would go high and beyond our ideal error level (Less than 1%).

Table 4.25: Summary of the results for FI=3% scenarios.

FI = 3%	Network Structure	μ_{ae}	σ_{ae}	$rmse$
d = 2 / l = 3	4-5-1	-1.812	1.705	2.610
d = 3 / l = 5	5-5-1	-0.410	5.369	5.187
d = 3 / l = 8	5-5-1	1.410	1.100	1.760

Table 4.26: Summary of the results for EI=1% scenarios.

EI = 1%	Network Structure	μ_{ae}	σ_{ae}	$rmse$
d = 3 / l = 5	5-10-1	1.016	1.971	1.599
d = 3 / l = 8	5-10-1	1.578	2.410	2.511
d = 4 / l = 15	6-10-1	2.301	3.989	3.990

Table 4.27: Summary of the results for EI=2% scenarios.

EI = 2%	Network Structure	μ_{ae}	σ_{ae}	$rmse$
d = 3 / l = 5	5-10-1	1.507	1.216	1.926
d = 4 / l = 8	6-10-1	0.325	2.085	2.091

4.5 Conclusion

In this chapter we have carried out various simulation results using our developed recurrent neural network. We considered two main causes of engine degradation namely fouling and erosion, which are soft degradation causes. They have a slow dynamics and their effect has to be studied along relatively large number of cycles. Among the available gas path measurements, the turbine temperature was chosen to be observed and predicted. the degradation initiate in different engine components and stays in the system. The RNN is provided with a portion of this data (we keep one data from each cycle at the same time) and the rest are used for testing the network. All of the temperatures are derived from our developed SIMULINK model. The engine is working in the take-off mode. Different scenarios were considered for

Table 4.28: Summary of the results for EI=3% scenarios.

EI = 3%	Network Structure	μ_{ae}	σ_{ae}	$rmse$
d = 3 / l = 5	5-10-1	2.495	1.674	2.996
d = 4 / l = 10	6-10-1	0.747	2.349	2.339

different levels of fouling and erosion. To evaluate the performance of our network for each scenario, we found the mean, standard deviation and the root mean square of the error. Furthermore, we introduced two prediction bounds at each simulation for uncertainty management associated with the prediction problem and we presented the formulation. The obtained results demonstrate that this type of neural network can be used for prediction and the prognosis results can be used for condition based maintenance. A summary of all the simulation cases is tabulated at the end of this chapter.

Chapter 5

Jet Engine Degradation Prognostics Using Dynamic Neural Networks

In this chapter, the investigation on degradation prognostics using another neural network will be presented. The network to be used is the nonlinear autoregressive neural networks (NARNNs). This is a combination of recurrent and dynamic neural networks which enjoys both features. The engine soft degradations are being considered. Their dynamics is slow and show their effects in the engine system after some cycles. They are similar to abrupt faults in the sense that once they occur in the system they remain there unless the engine is taken for maintenance. We are interested in predicting their future values for scheduling condition-based maintenance. This goal is to be achieved by using the proposed neural network. At the end of this chapter we compare the prediction performance of the two neural networks that are considered in this thesis.

5.1 Degradation Trend Prognostics Using Nonlinear Autoregressive Neural Networks (NARNNs)

In this section, we predict the degradation trends and their effects on the turbine measurable data using the nonlinear autoregressive neural networks (NARNNs). It was shown in Chapter 4 that recurrent neural networks (RNNs) are capable of learning short-time dependencies due to their global feedback from the network out put to the network input. On the other hand, NARNNs are capable of learning long-term dependencies. This implies that the output at the present time is dependent on the present and past values of the input as well as the past vales of the output itself. This is explained in [197] as to why learning long-term dependencies is not a feasible task.

The general type of the NARNNs that are used in this chapter belongs to a class of architectures that is based upon nonlinear autoregressive models with exogenous inputs (NARX model). NARX model is in fact a recurrent neural network capable of modelling and predicting efficiently time-series data. NARX is based on the linear ARX model that is commonly used in time-series modelling. Therefore, it benefits from the advantages of both recurrent and dynamical neural networks. The defining equation of the NARX network was presented earlier in Chapter 2. In NARX NN, the next value of the dependent output signal $y(n)$ is regressed on previous values of the output signal and previous values of an independent (exogenous) input signal as follows

$$\begin{aligned} y(n+1) &= f[y(n), ..., y(n-d_y+1); u(n), u(n-1), ..., u(n-d_u+1)] \\ &= f[\mathbf{y}(n); \mathbf{u}(n)] \end{aligned} \tag{5.1.1}$$

where f is a nonlinear function, u and y are the input and output regerssors, respectively, d_u and d_y are the number of input and output delays, respectively. In other

words, we are dealing with two tapped delay lines, one sliding over the input signal and the other sliding over the network's output. The set of time-delay operators serves as memory elements. To achieve our prediction purpose, especially in multi-steps-ahead prediction where we are interested in a wider prediction horizon, the model's output should be fed back to the input regressor for a fixed but finite number of time steps. It should be pointed out that NARX networks are also powerful tools for nonlinear system identification.

NARX networks normally converge much faster and generalize better than other networks [159]. Moreover, NARX networks are shown to be universal computational devices. In [198], NARX networks have been compared with nine other recurrent neural networks and these capabilities of the NARX neural networks have been confirmed. The challenge associated with NARX networks is determining the optimum values of the autoregressive model and the exogenous input order. They have to be determined empirically and through trial and error. Normally the performance will improve with increasing the model orders, but this comes at the cost of additional parameters and learning time.

NARX neural networks have applications in chaotic time-series prediction ([199]). They have also been applied in some non-engineering fields such as variable bit rate (VBR) video traffic time series prediction ([200]), and nearshore sandbar behaviour modelling ([170]). Although very useful, they have not found their way in engineering prediction problems. In [143], the authors have demonstrated a model reduction technique for computing critical engine component parameters for remaining life prediction. In this chapter, our goal is to demonstrate the effectiveness of this approach for predicting the degradation trend in a jet engine. In the following section we describe our NARX approach followed by the obtained simulation results.

5.1.1 NARX Neural Network Prognosis Approach

In this section we follow the same steps as in the previous chapter. Jet engine, like any other physical system degrades thorough time due to operational and environmental conditions. This in return can change the engine health parameters such as the efficiency and the mass flow rate. These quantities are not directly measured by sensors but they consequently change the gas path measurements such as the spool speed, pressure and the temperature at the engine critical points. One can track these changes by monitoring gas path data. After studying certain data points and feeding them into the NARX NN for training, we expect the neural network to predict the future value of them for certain steps ahead in time of the engine flight cycle. The challenge would be the choice of the network architecture such as the number of hidden layers, the number of the neurons in the hidden layers, the activation functions, and the number of input and output delays.

We also have to decide about the portion of the data used for training and the portion used for testing and validation. Once we are confident about our network (called the trained predictor) and methodology (by studying the errors and statistical measures), we will demonstrate the approach by applying it to different engine degradation scenarios for two different degradation types. When working with neural networks one should always make sure that the network is trained enough and at the same time it is not over trained. We try to simulate the same scenarios and cases in order to be able to make a proper comparison. We also use the same inputs and predict the same gas path measurement, i.e. the turbine temperature.

For the data set, we use the same ones as in Chapter 4. Similar approach is followed to build these data sets, to be used for training and testing our proposed NARX neural network. When our degraded engine model is simulated certain number of data points become available. The data points are the gas path measurements. We consider the

two main causes of engine degradation namely the fouling and the erosion. They are called soft as they have as low dynamics and do not change the system abruptly. They affect the efficiency and the mass flow rate of the engine components which in return change the gas path measurements. Fouling mostly affects the compressor and erosion changes the turbine performance more, as compared to the other engine components.

The resulting changes are not observable with only one flight cycle, that is why one must run the simulations for the take-off mode for hundred cycles and pick one point (the same point) from each flight cycle. The procedure for generating the simulations is described in the previous chapter. Measurement noise has been considered as well. The thrust level is kept constant in spite of degradations by increasing the fuel flow rate. The data are then used to train our proposed NARX NN architecture toward learning the fouling/erosion dynamics and the turbine temperature increasing effects caused by these phenomena.

5.1.2 Simulation Results

Once the data sets (ordered points) are selected, they are used toward training a suitable NN predictor and then testing it. To achieve prognostics and to be able to identify the degradation level at a certain point in time in future, the time evolution of the turbine temperature has to be learnt. The task in the training phase is to adjust the network parameters such as the number of hidden layers, the number of neurons, the data portion for training and testing are selected to obtain acceptable training error. This shows the ability of the trained NARX NN predictor to dynamically map the historical and current data into the future. When the network is trained properly, the prediction task starts for predicting the future TT development.

In the following sections we first demonstrate different cases and scenarios for a

fouled compressor and then scenarios and cases for an eroded turbine. By following these cases the selection of the data and the network parameters become more clear and how the results are improved by adjusting different variables. One can also see how many step-ahead prediction is achievable by using our proposed network. For that we will compare the actual and the predicted values which are the NN outputs. We also study them statistically and depict the errors.

5.1.3 Simulation Results for Fouling Scenarios

The first degradation to be studied is the fouling. Fouling mostly occurs in the compressor and adversely impact its functionality by reducing the efficiency and the mass flow rate. In the following cases and scenarios, the fouling level varies from 1% to 3%. We do not exceed this level of fouling. As when the fouling goes beyond this level, the engine is recommended to be take off-line for washing. The simulations for a fouled engine are run 100 times (100 cycles) and one point from each cycle at the same time is chosen to build-up our data vectors to be used for neural network training and testing phases. We pick the 12^{th} second of each take-off cycle for an engine degraded by fouling.

5.1.3.1 First Scenario: FI = 1%

At the 1% fouling level, the compressor efficiency drops 1% and the mass flow rate decreases 1%. The turbine temperature for 100 cycles (already shown in Chapter 4) are depicted here again in Figure 5.1.

FI = 1%: Case 1

To examine the capability of our NARX NN for prediction, we feed the network with the fuel flow as the input and the turbine temperature (TT) is predicted as the

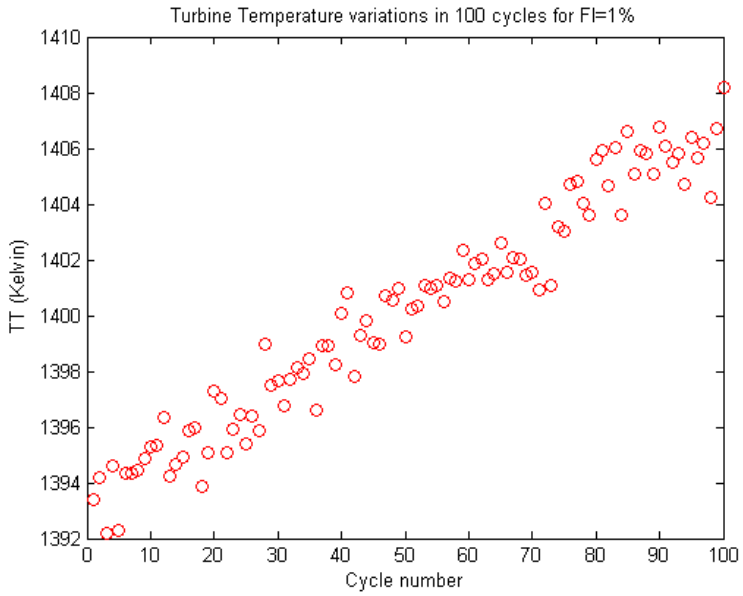

Figure 5.1: Turbine temperature variations due to a fouled compressor.

output. The previous values of the input and the output are regressed and used for neural network training. We start by setting the input and output delays to be equal to three ($d_u = d_y = 3$) as after some trials and errors these seem to be the proper delay values. Out of the 100 available data points half of them are used for training and the other half is used for testing the network. The number of neurons in the hidden layer is five. The above mentioned parameters result in a 8-5-1 network (8 = 1+1+3+3 where we have fuel flow as input, turbine temperature as out put and three delayed versions of each).

The approach for demonstrating and evaluating the prediction result is the same as before. First, we depict the real values and the predicted values in the same figure. Then the prediction error obtained from the NN training phase are plotted point-wise. Afterwards, as explained earlier, we add the upper and the lower prediction bounds to overcome the uncertainty associated with the problem of prognostics. Finally, the statistical error measures such as the prediction error mean (μ_{ae}), prediction error standard deviation (σ_{ae}) and root mean squared error are presented in a tabular form

to provide quantitative measures about the goodness of the prediction results.

Our goal is to achieve the best result for a three-step ahead prediction that here implies three cycles ahead ($l = 3$). Once we achieve this number of steps, we can widen our prediction horizon. The results are depicted in Figure 5.2. The *circles* denote the actual temperature values and the *stars* show the predicted ones. The increasing trend is learnt and predicted by our network. By studying Figure 5.2, one can see that 97 data points are being compared although the test data vector has 100 entries. The reason is that we are conducting a 3-step ahead prediction and thus starting from the first point of the test data set, 97 points can be predicted.

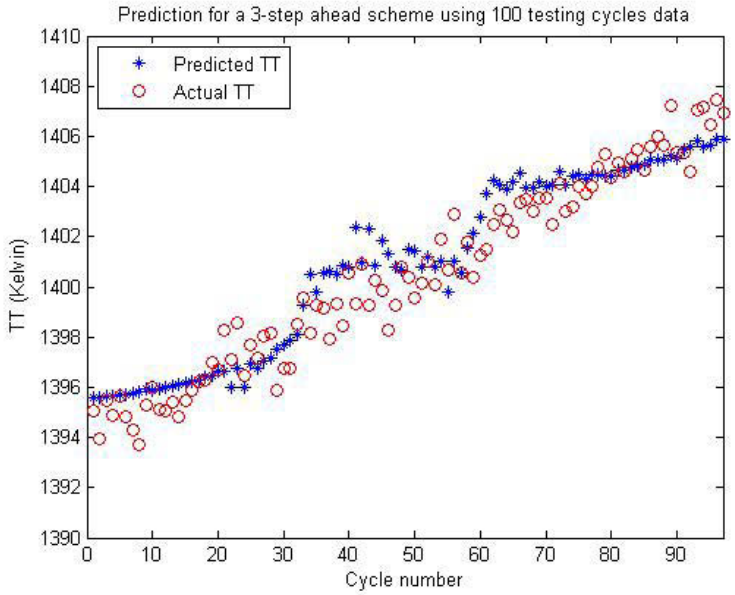

Figure 5.2: Actual vs. predicted TT for a 1% fouled compressor (3-step ahead).

Furthermore, the prediction error for the 30 test data point is shown graphically in Figure 5.3. The error varies between -3K and 2K. Again, we have considered the upper and the lower prediction bounds for uncertainty management and the result are shown in Figure 5.4. One can observe from the Figure 5.4 that only 6 of the 97 data points are outside of the bands which yield that 93.8% are inside the bands.

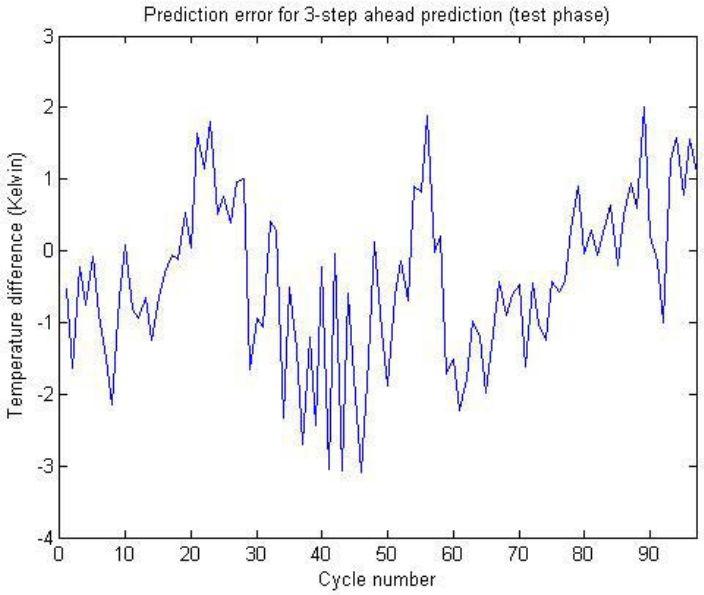

Figure 5.3: Temperature prediction error for a 1% fouled compressor (3-step ahead).

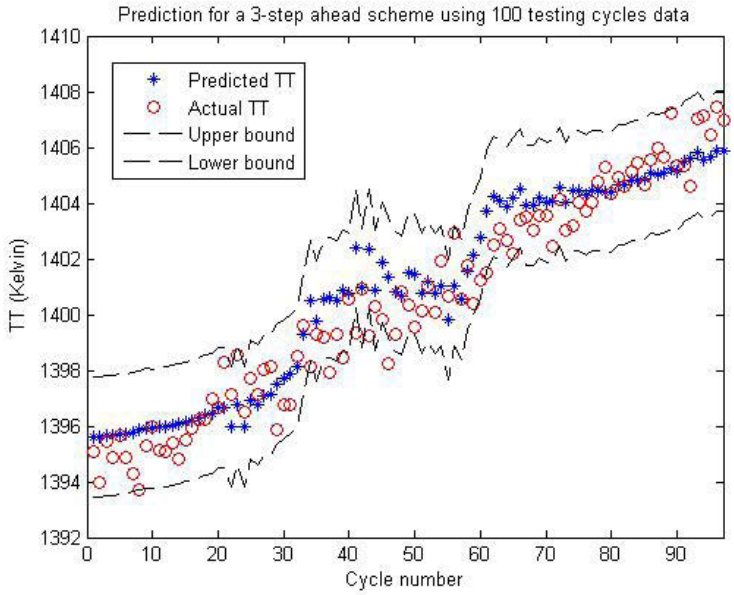

Figure 5.4: Actual vs. predicted TT for a 1% fouled compressor considering prediction bounds (3-step ahead).

Table 5.1: Prediction error for FI=1% case 1.

μ_{ae}	0.970 (K)
σ_{ae}	1.149 (K)
$rmse$	1.220 (K)

The above three figures demonstrate the NARX network prediction capability graphically. One also needs a quantitative measure of performance. These measure are presented in Table 5.1. The mean of the prediction error (absolute value) is $\mu_{ae} = 0.970K$, the prediction standard deviation is $\sigma_{ae} = 1.149K$ and the error root mean square is $rmse = 1.220K$.

Remark. If one compares the error results obtained in this case with the error results obtained in Case 3 of Chapter 4, one can see that they are quite close to each other. When comparing the results, this fact should be highlighted that for training our RNN, 70% of the data were used to train the network where only half of the data are used in training our NARX NN scheme. Moreover, training the NARX network is relatively faster, as compared to our other proposed scheme.

FI = 1%: Case 2

In the second case, we still investigate turbine temperature prediction for a 1% fouled compressor. In this case, 50% of the available data are used for training our NN and 50% are used for testing it. It is not possible to always follow certain predetermined rules for neural network parameter selection. One has to try different number of hidden neurons, different input delays, and different output delays. Following the trial and error process for 10-step ahead prediction, we concluded that a 11-5-1 network with four input delays d_u (fuel flow) and five output delays d_y (turbine temperature) can give us suitable prediction results (11 = 1+1+4+5 where we have the fuel flow as input, turbine temperature as output, four delayed version of the input and five

Table 5.2: Prediction error for FI=1% case 2.

μ_{ae}	1.535 (K)
σ_{ae}	1.721 (K)
$rmse$	1.860 (K)

delayed version of the output). The prediction results are depicted in Figure 5.5.

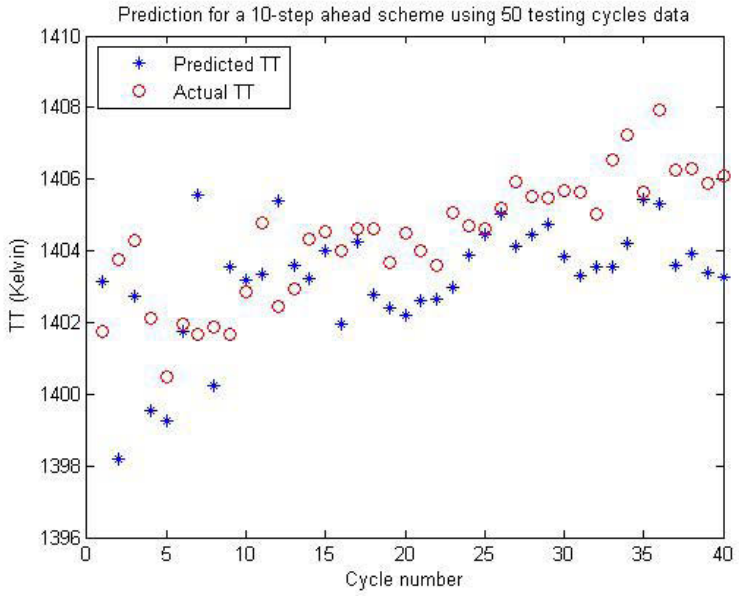

Figure 5.5: Actual vs. predicted TT for a 1% fouled compressor (10-step ahead).

Moreover, the prediction error for the 40 predicted data points is shown point-wise in Figure 5.6. The mean, standard deviation and $rmse$ of the prediction error are 1.535, 1.721 and 1.860, respectively. The values are also presented in Table 5.2. One can verify that the prediction error is less than 1% and the results are even better than the same case as obtained in Chapter 4.

We have also depicted the prediction bounds in Figure 5.7. Full explanation on deriving these bounds is given as a form of a Remark in Chapter 4. We follow the same methodology in this chapter. One can observe that the points are within the prediction bounds to a very good extent (at the rate of 87.5%).

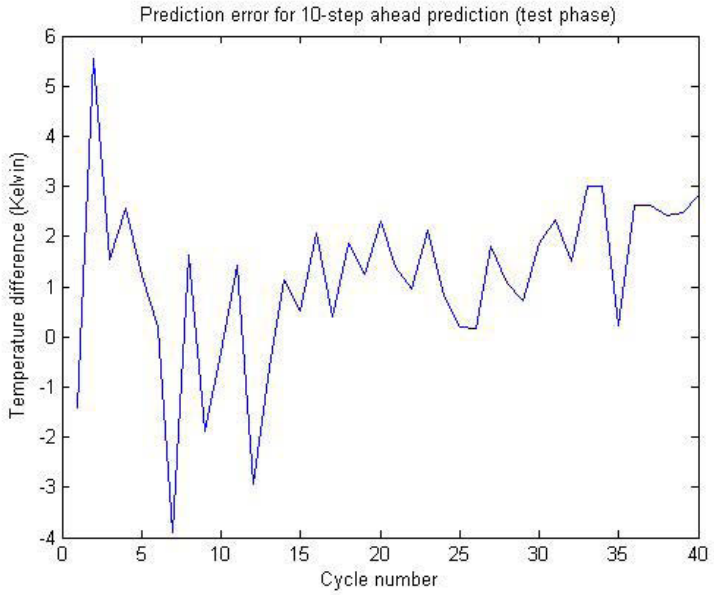

Figure 5.6: Temperature prediction error for a 1% fouled compressor (10-step ahead).

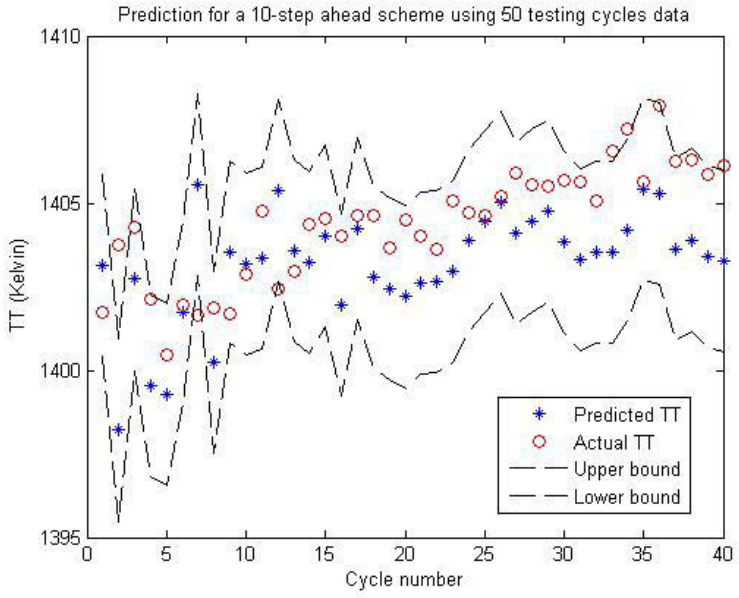

Figure 5.7: Actual vs. predicted TT for a 1% fouled compressor considering prediction bounds (10-step ahead).

FI = 1%: Case 3

In the third case of the first scenario, we widen the prediction horizon further in time. Many trials are conducted and only the best results are depicted. The aim of the investigation in this case is to find the maximum prediction capability of this NARX NN scheme. By changing the number of neurons in the hidden layer or increasing the size of training data vector, one may be able to have ma larger cycles ahead prediction but as the goal is to examine our proposed network developed in Case 2, we keep the same network parameters and data portion and just change the step ahead factor. The 11-5-1 network is capable of predicting the turbine temperature for a fouled compressor for 16 cycles ahead with reasonable error values (less than 1%). The comparison between the actual and the predicted temperatures is depicted in Figure 5.8.

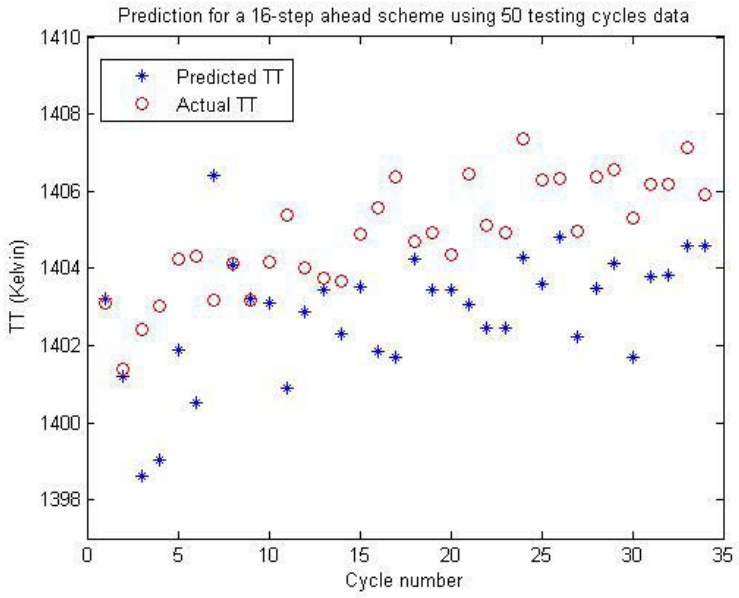

Figure 5.8: Actual vs. predicted TT for a 1% fouled compressor (16-step ahead).

The error is depicted graphically in Figure 5.9 for the 34 predicted points. This represents the difference between the actual and the predicted temperature for each

Table 5.3: Prediction error for FI=1% case 3.

μ_{ae}	2.055 (K)
σ_{ae}	2.373 (K)
$rmse$	2.479 (K)

data in the testing set. Furthermore, the error statistics for the 16-step ahead are shown in Table 5.3. Finally, one can see from Figure 5.10 the predicted versus the actual values that are within the zone made by the lower and upper prediction bounds that confirms the goodness of this prediction. Specifically, 82.8% of the data points are within the bands.

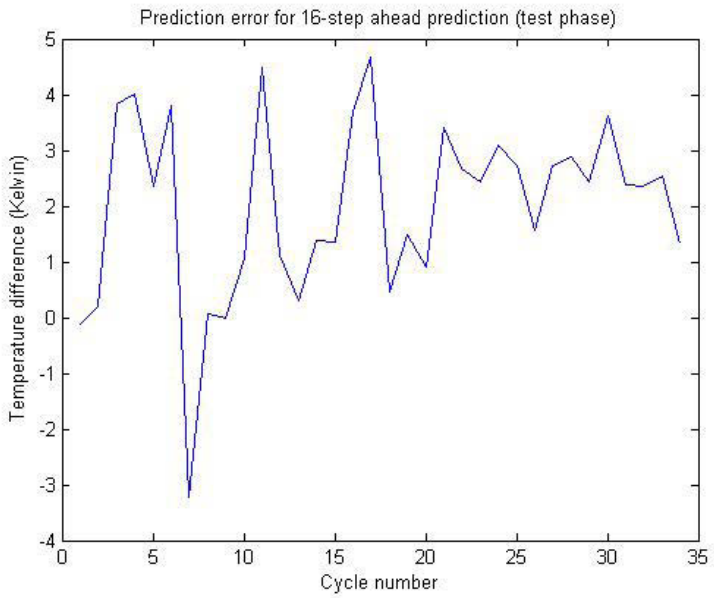

Figure 5.9: Temperature prediction error for a 1% fouled compressor (16-step ahead).

We emphasize again that prediction results by using the NARX network show that training and handling this network is easier as compared to the other architecture and less data are needed for the same level of preciseness. We now proceed to the second scenario which considers a 2% fouling in the compressor.

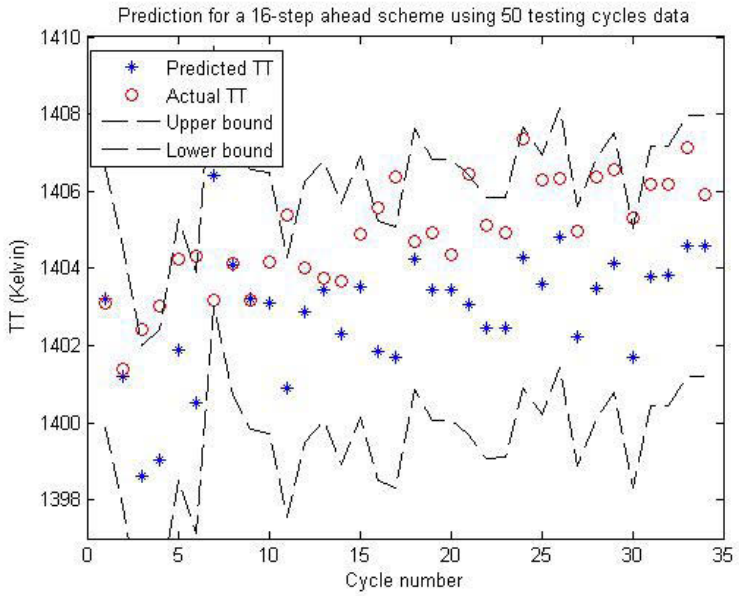

Figure 5.10: Actual vs. predicted TT for a 1% fouled compressor considering prediction bounds (16-step ahead).

5.1.3.2 Second Scenario: FI = 2%

In the second scenario that we consider in this section a 2% fouling is considered. This implies that at the end the engine efficiency will drop by 2% and the mass flow rate will increase by 1% following the 1:0.5 rule. The data generation method follows the same routine that was described for the case of 1% FI. The difference is that because the degradation level is now higher, the changes in the turbine temperature to the compressor fouling will be more significant, i.e. the slope of the change will increase. The turbine temperature variations over 100 take-off cycles are depicted in Figure 5.11. The following cases will be discussed using the depicted data.

FI = 2%: Case 1

In the first case we perform a 5-step ahead prediction using the neural network proposed in case 1 of the first scenario. The fuel flow and its three delayed values are the

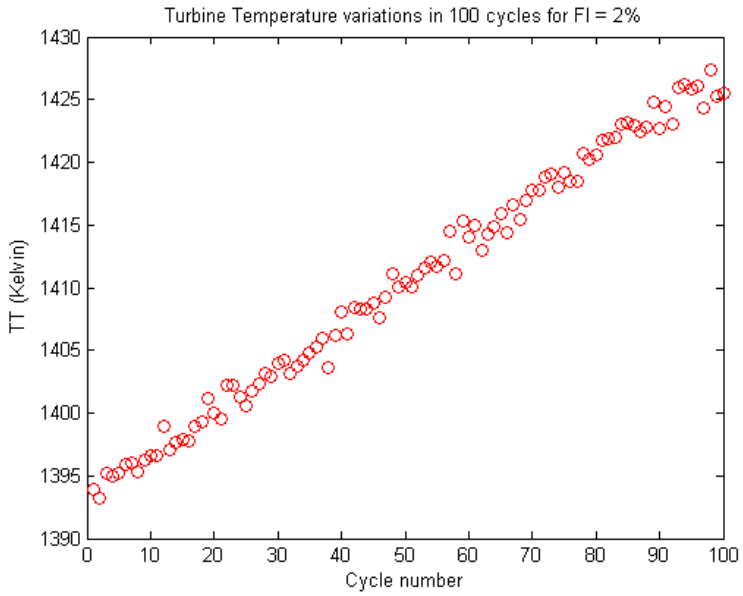

Figure 5.11: Turbine temperature variations due to a 2% fouled compressor.

network inputs in addition to the three delayed turbine temperature ($d_u = d_y = 3$). We used three as trials and error showed us that one can obtain satisfactory results while still keeping the parameters as low as possibles. Five neurons exist in the hidden layer of the proposed NARX network which results in a 8-5-1 architecture (8 is obtained as the sum of the input, the output and three delayed version of each : 8+1+1+3+3). We train the network using only half of the available data and the rest are used for testing the prediction performance.

In Figure 5.12 we have presented the prediction results versus the actual turbine temperature that is derived from our SIMULINK model (see Chapter 3). The predicted turbine temperatures are indicated with *stars* and the real turbine temperatures are indicated with *circles*. One can observe that the predicted temperatures properly follow the real ones.

The point-wise error is depicted in Figure 5.13. By studying Figures 5.12 and 5.13 and the error statistical results all together, confirm the effectiveness of this method for a 5-step ahead prediction in time for a 2% fouled compressor.

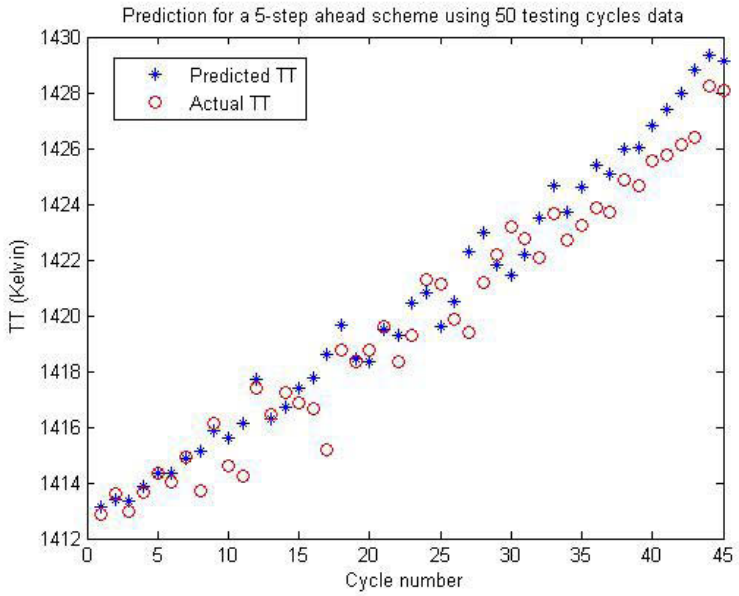

Figure 5.12: Actual vs. predicted TT for a 2% fouled compressor (5-step ahead).

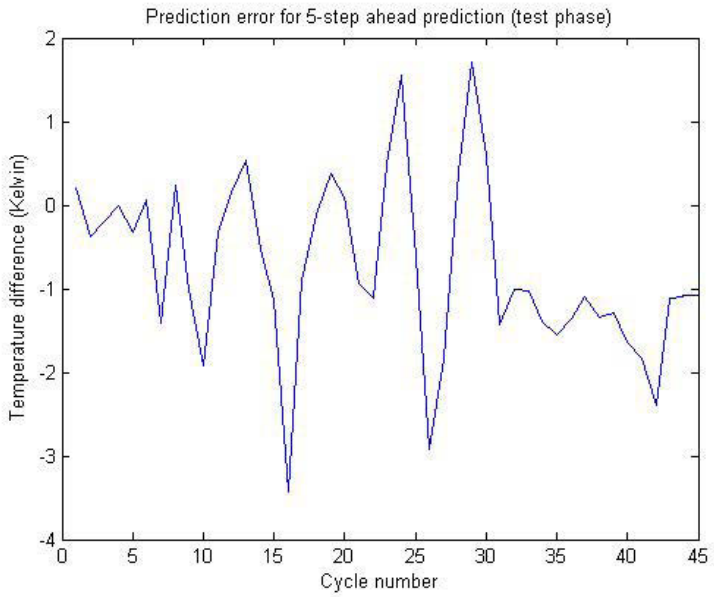

Figure 5.13: Temperature prediction error for a 2% fouled compressor (5-step ahead).

Table 5.4: Prediction error for FI=2% case 1.

μ_{ae}	1.764 (K)
σ_{ae}	1.073 (K)
$rmse$	1.293 (K)

The predicted data points are in average $1.7K$ different from the actual values (absolute value) which is a very good prediction result as the nominal turbine temperature during take-off is about $1400K$. The statistics of the network performance are shown in Table 5.4

One can note that the predicted versus the actual values mostly stay within the zone made by the lower and upper prediction bands. Only 5 of the 45 data points are outside the bands which means more that 88% of them are inside. This confirms the goodness of this prediction scheme. The results are shown in Figure 5.14.

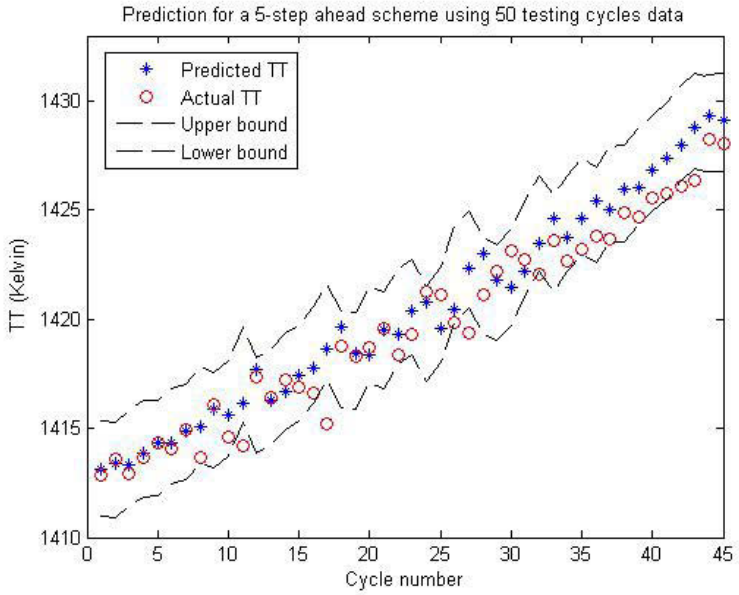

Figure 5.14: Actual vs. predicted TT for a 2% fouled compressor considering prediction bounds (5-step ahead).

FI = 2%: Case 2

In the second case that we consider in the scenario of a 2% fouled compressor, the prediction horizon is extended to 10 flight cycle ahead. To achieve this goal the same type of inputs were used except that d_u was set to 4, d_y was set to four as well. Achieving good results were viable using three delays. Besides, we do not want to increase the delays and number of network parameters unboundedly, hence four appeared to give satisfactory results. We do not change the number of neurons in the hidden layer or the data portion used for training and testing purposes. This results in a 10-5-1 NARX NN (10 = 1+1+4+4 which is the input, and the output and four delayed values of input and output). One can compare the actual and predicted values (network output) in Figure 5.15.

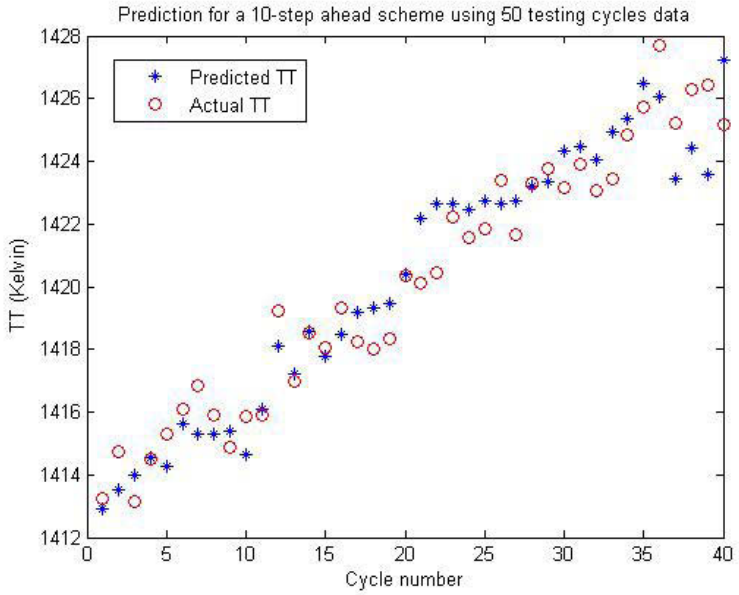

Figure 5.15: Actual vs. predicted TT for a 2% fouled compressor (10-step ahead).

The resulting error is shown in Figure 5.16 followed by Table 5.5 which summarizes the results of this case. One can find the mean of the absolute error, the error standard deviation and the rmse of the prediction error in this table.

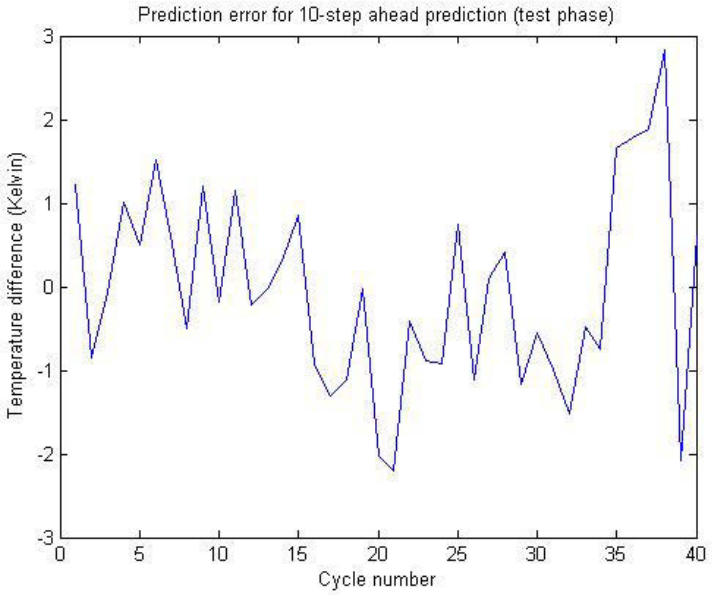

Figure 5.16: Temperature prediction error for a 2% fouled compressor (10-step ahead).

Table 5.5: Prediction error for FI=2% case 2.

μ_{ae}	0.873 (K)
σ_{ae}	0.989 (K)
$rmse$	0.987 (K)

Furthermore, in Figure 5.17 we have added the upper and the lower prediction bounds to overcome prediction uncertainty and one can note that the data points are within these two bands to a great extent. One can verify this by observing that 34 of the 40 predicted point are inside the band with corresponds to 85% of them.

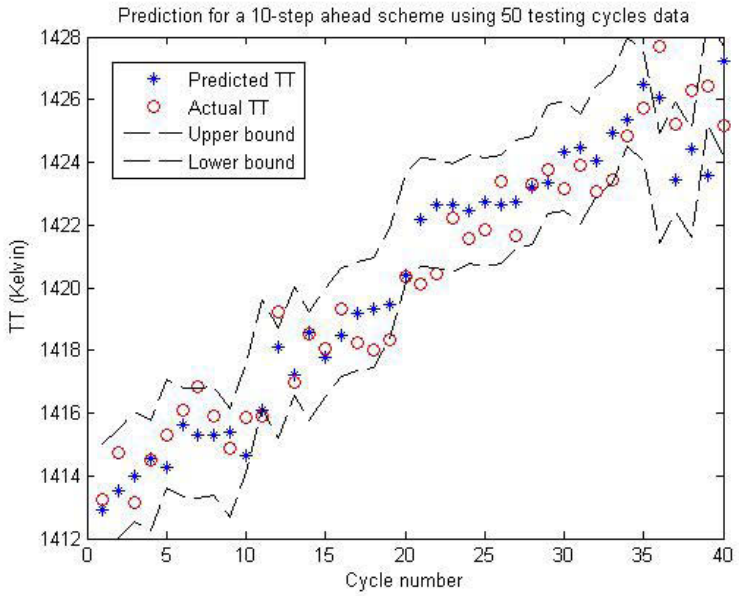

Figure 5.17: Actual vs. predicted TT for a 2% fouled compressor considering prediction bounds (10-step ahead).

FI = 2%: Case 3

In the third case we go further in time to find out the maximum allowable cycles ahead which still gives good prediction results, implying that the error statistics are low (less than 1% as compared to the nominal turbine temperature) and the values are mostly within the prediction bands. We keep the same number of neurons in the hidden layer (set to 5) and use 50% of the data for our NARX NN training. After many trials and errors we concluded that in order to keep these parameters constant, and achieve a 16-step ahead prediction (l = 16), d_u should be equal to 7 and d_y

Table 5.6: Prediction error for FI=2% case 3.

μ_{ae}	1.215 (K)
σ_{ae}	1.810 (K)
$rmse$	1.900 (K)

should be equal to 8. Taking all of these parameters, we have a 17-5-1 NARX NN. The comparison results are depicted in Figure 5.18.

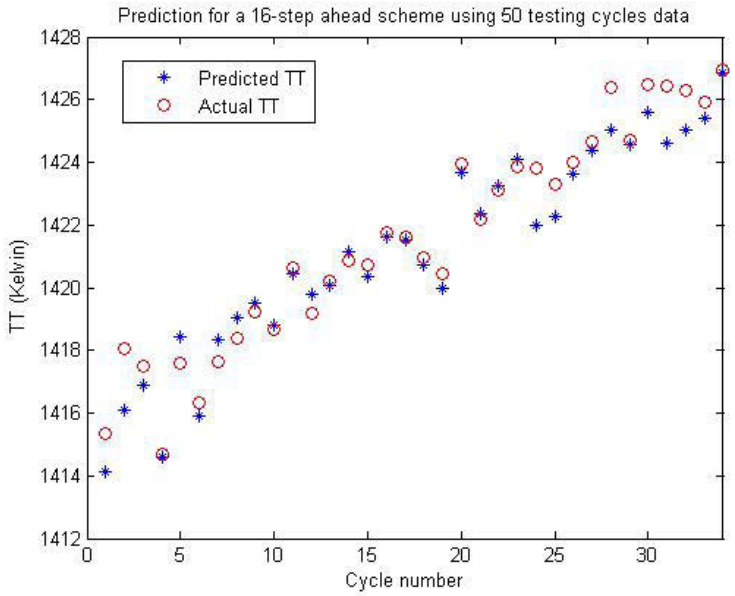

Figure 5.18: Actual vs. predicted TT for a 2% fouled compressor (16-step ahead).

One can observe the error results graphically in Figure 5.19 and quantitatively in Table 5.6. Note that half of the data were used for testing implying that 50 points are available. As the network uses the current and previous data to predict 16 cycles ahead in time, 34 predicted values are obtained.

Moreover, in Figure 5.20, the prediction bounds are added to the comparison figure. One can verify that 92% of the data are within the bands made by the lower and the upper prediction bounds. One can go further than a 16-step prediction but the amount of computation regarding the time and the cost is one issue that limits

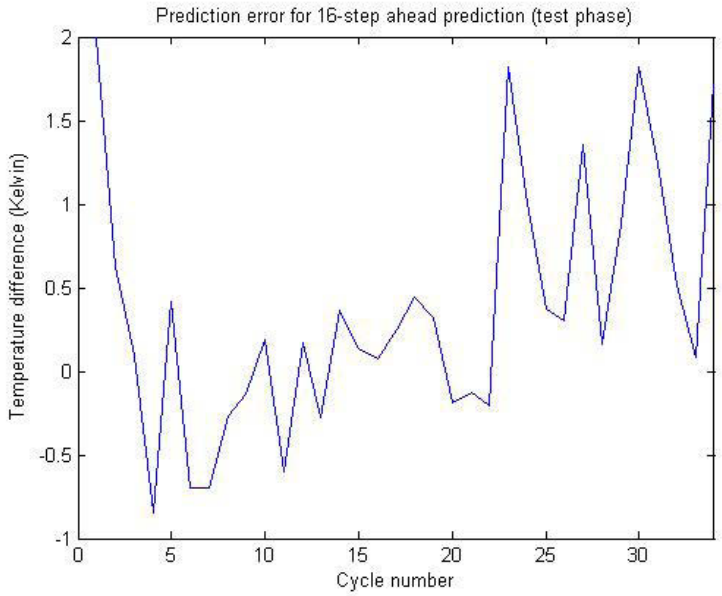

Figure 5.19: Temperature prediction error for a 2% fouled compressor (16-step ahead).

the number of cycles ahead possible to be predicted by our proposed NARX NN.

5.1.3.3 Third Scenario: FI = 3%

In the third scenario, we increase the fouling level in the compressor by decreasing the efficiency and increasing the mass flow rate even further. The efficiency drops by 3%. As a consequence, higher changes in the turbine temperature will occur as depicted in Figure 5.21. Temperature variations are more significant as compared to previous case with less level of degradation.

In the following case we use a portion of these data to train and the rest to test our proposed NN. The network performance is demonstrated graphically and quantitatively in each case.

FI = 3%: Case 1

Similar to the previous scenario, we start the first case by a 3-step ahead prediction and by finding the suitable network structure and parameters to achieve this goal. By

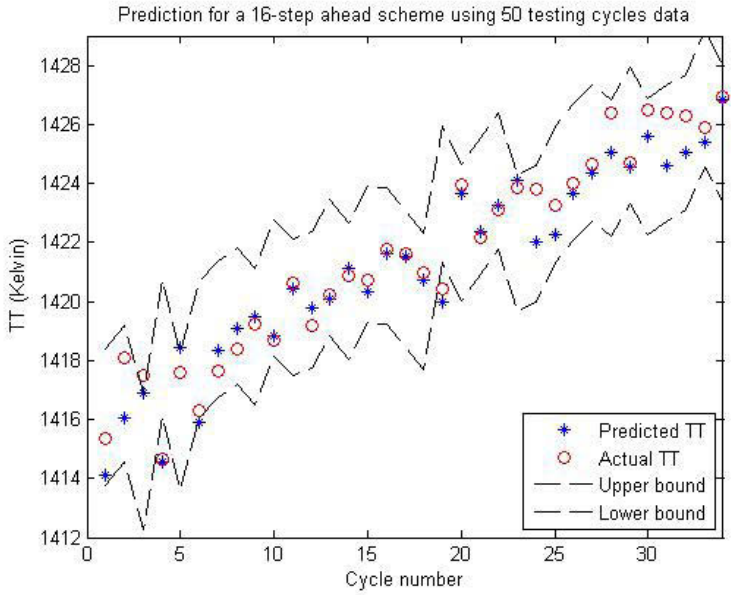

Figure 5.20: Actual vs. predicted TT for a 2% fouled compressor considering prediction bounds (16-step ahead).

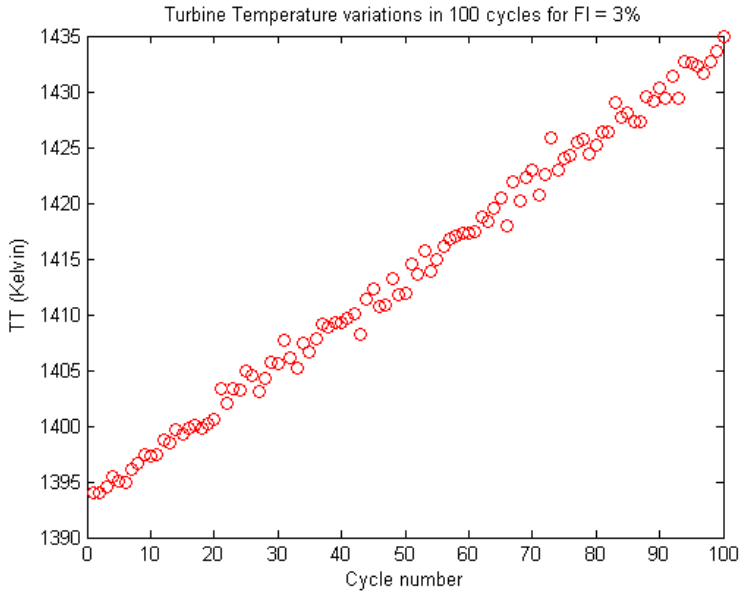

Figure 5.21: Turbine temperature variations due to a 3% fouled compressor.

Table 5.7: Prediction error for FI=3% case 1.

μ_{ae}	0.991 (K)
σ_{ae}	1.451 (K)
$rmse$	1.749 (K)

setting $d_u = d_y = 3$, and the number of hidden neurons to five, we have a 8-5-1 neural network. The network is trained with 50% of the data points and as can be seen in the following figures and table, the prediction result is very good. The predicted and real temperatures are compared in Figure 5.22.

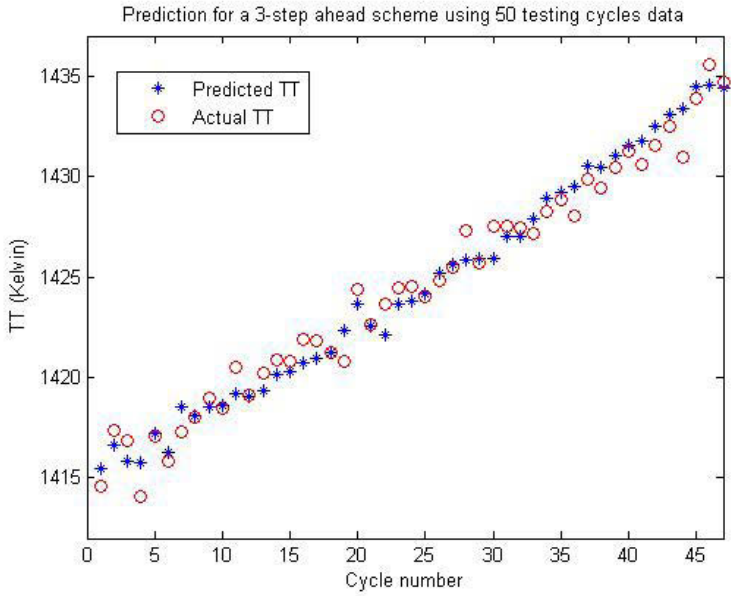

Figure 5.22: Actual vs. predicted TT for a 3% fouled compressor (3-step ahead).

The error for the 47 predicted points which is the difference between the real and predicted value (NN output) at each time is depicted in Figure 5.23. The absolute error mean, error standard deviation and the rmse are $0.991K$, $1.451K$ and $1.749K$, respectively which confirms the suitability of the prediction.

One should note that although tight, most of the data are in the prediction band as can be verified from Figure 5.24. By most we mean that 41 of the 47 data points

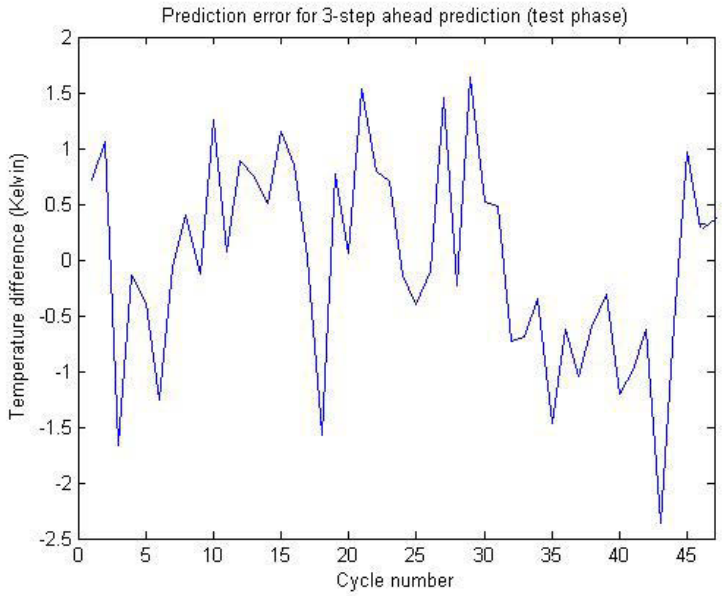

Figure 5.23: Temperature prediction error for a 3% fouled compressor (3-step ahead).

which results in 87.2% of them being inside the bands.

FI = 3%: Case 2

As the results in the previous case deemed satisfactory, we now examine the obtained network in that case and verify whether it can be used for further cycles ahead, or some modifications on the network parameters are needed. After performing a number of simulations we came to the conclusion that the number of input and output delays must be increased to four ($d_u = d_y = 4$), providing us with a 10-5-1 NARX NN. Note that we are still using 5 neurons in the hidden layer of our proposed network and the amount of data needed for training this network is less as compared to the RNN developed in the previous chapter. Both real and predicted values for 8-step ahead prediction are depicted in Figure 5.25.

Similar to the previous case we have tabulated the error statistics in Table 5.8 (μ_{ae}, σ_{ae} , $rmse$). The error for all the 42 predicted points versus the real points are shown in Figure 5.26. By verifying Figure 5.27, one can see that both points at each

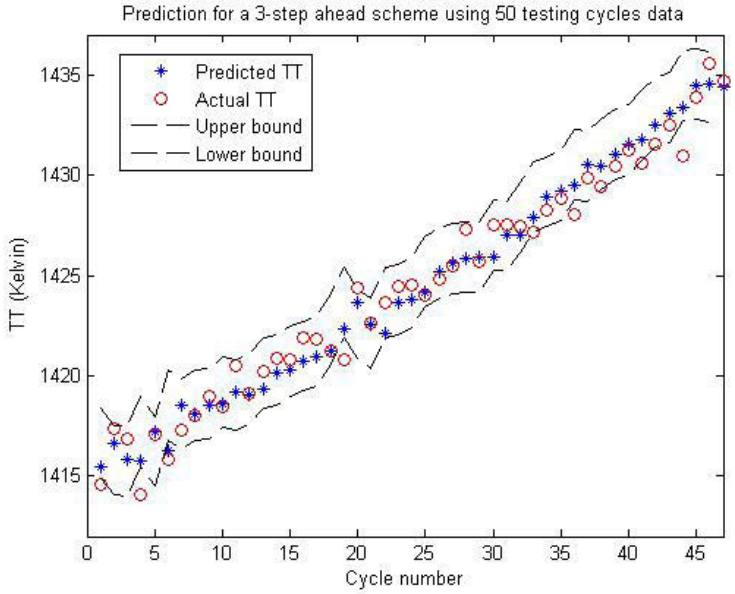

Figure 5.24: Actual vs. predicted TT for a 3% fouled compressor considering prediction bounds (3-step ahead).

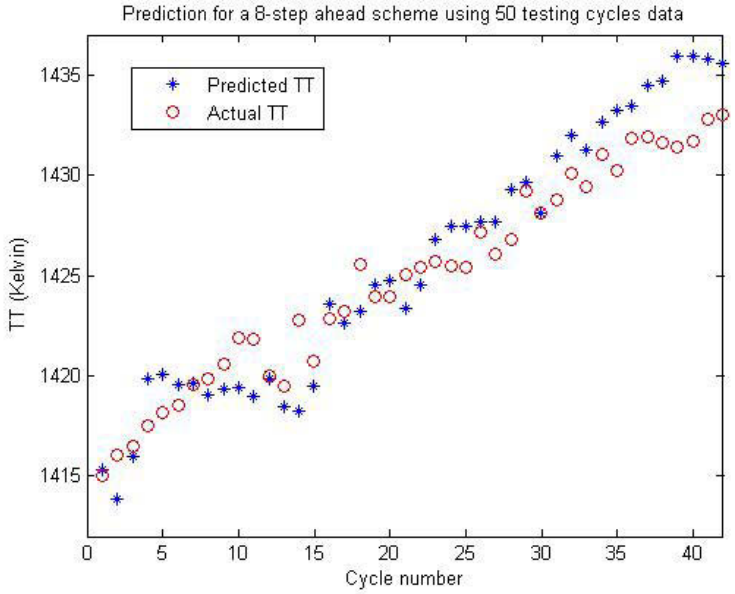

Figure 5.25: Actual vs. predicted TT for a 3% fouled compressor (8-step ahead).

Table 5.8: Prediction error for FI=3% case 2.

μ_{ae}	1.023 (K)
σ_{ae}	1.078 (K)
$rmse$	1.339 (K)

time are close and in the bands although the points start to move out of the bands at the end. Note that 6 of the 42 data points are outside of the bands which amounts to only 14% of them (86% are inside the bands).

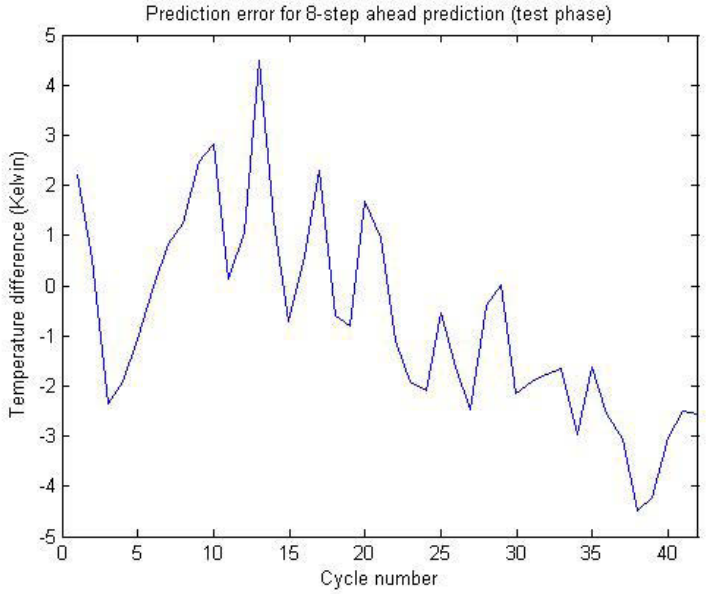

Figure 5.26: Temperature prediction error for a 3% fouled compressor (8-step ahead).

Having obtained satisfactory results from our developed NARX neural network, we conclude our investigation on the cases associated with fouling scenarios. If one wants to have a larger prediction horizon while yielding satisfactory results, the network parameters such as the number of input and output delays have to be adjusted as well as the number of neurons and data percentage used for training. One has to always consider the amount of computation time as there is a trade-off between the accuracy and the complexity. Note that since we considered flight cycles for our

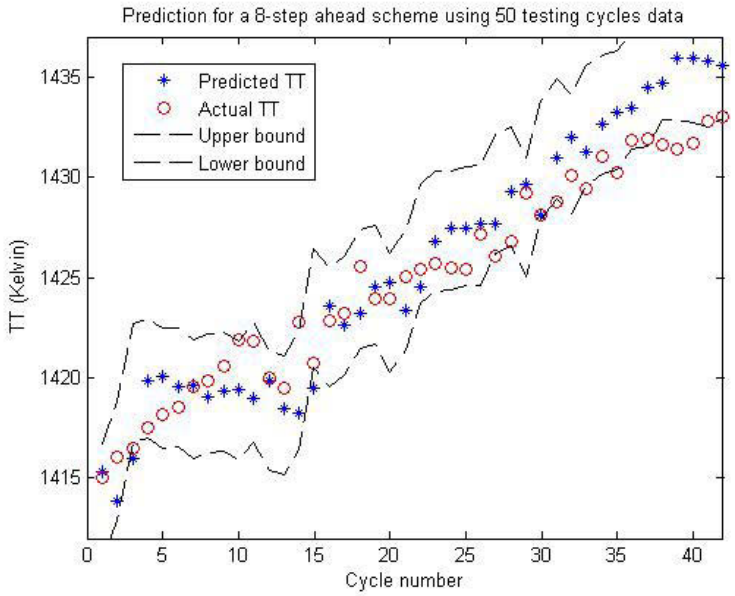

Figure 5.27: Actual vs. predicted TT for a 3% fouled compressor considering prediction bounds (8-step ahead).

prediction horizon, the number of step-ahead is good enough and gives enough time to the operators to decide about the necessity of the maintenance actions while the engine condition is still safe. In the next sections, we study and discuss the cases in which the turbine is affected by the erosion.

5.1.4 Simulation Results for Erosion Scenarios

In this section, we investigate a number of scenarios associated with the erosion. Erosion mostly occurs in the turbine section and can change the behaviour of the blade. According to its definition, erosion is the removal of the material from the flow path components with hard particles. This removal results in loss of efficiency and increase in the mass flow rate. We assume a linear relationship in the form of 2:1 between the turbine efficiency and the mass flow rate, respectively. If the erosion occurs and then remains in the system, it will degrade the engine performance and

in long term it can even lead to blade failure, hence it is important to keep track of this degradation and predict it.

The erosion index (EI), as explained in Chapter 3 is used here. For different levels of erosion we study the resulting effects on the turbine temperature (TT). To predict the future values of the turbine temperature we used our proposed NARX NN, which is a type of recurrent dynamic neural network. We have tried to produce the same scenarios for the three levels of fouling, as in previous subsections, to be able to make a comparison between the cases.

5.1.4.1 First Scenario: EI = 1%

If erosion occurs and stays in the turbine it can change the spool speed, turbine temperature, etc. The rates of these changes were studied and validated with the GSP software in Chapter 3. The simulations for a system under erosion have been performed for 200 take-off cycles. In the take-off mode, since the engine is operating from the ground idle condition to the maximum level of fuel, the degradation initiation and propagation is more significant and therefore it is studied here. For EI = 1%, this 1% is equivalent to 1% drop in the turbine efficiency and 0.5% drop in the turbine mass flow rate. Nonlinear autoregressive neural networks are employed here to learn this evolution trough flight cycles. Thus it is not necessary to save and use all the data points from the 200 cycles. Degradations have slow dynamics and do not change the system abruptly or significantly in only one cycle. Hence, instead one can pick the same time from each flight cycle and put them in a vector. From this vector a portion will serve as the training data set and a portion for testing purposes.

The turbine temperature evolution for a 1% eroded turbine during 200 cycles is depicted in Figure 5.28. The effects of the measurement noise on the data have been considered as well. In the following three cases, we consider three different prediction

steps.

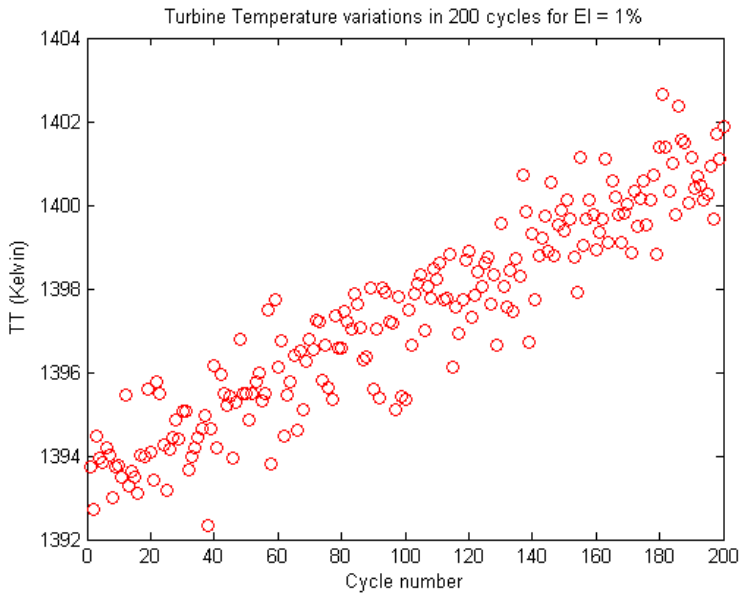

Figure 5.28: Turbine temperature variations due to a 1% eroded turbine.

EI = 1%: Case 1

Our goal in this case is to predict the future turbine temperature considering the following assumptions. Firstly, we assume that a suitable data set is available which contains information about the engine and the degradation dynamics (an engine subjected to soft degradations and erosions for this case). Secondly, the 1% erosion has occurred in the system and remains there implying that no maintenance action has been done. Taking these two conditions into account one is able to predict the turbine temperature for certain cycles ahead and the results are shown in the following subsections. Once the turbine temperature is predicted for some flights ahead, one can then decide if the next flights will be safe or the temperature has reached certain thresholds that makes it necessary to take the engine off-line for maintenance.

The fuel flow rate and the turbine temperature are the NARX NN inputs. In addition, the delayed version of these two measurements will serve as network input

for the training phase so that the network learns the system dynamics. We study the case for a 5-step ahead prediction. We set $d_u = d_y = 3$ and the number of neurons in the hidden layer to five. We explained earlier that the number of delays is determined through trials and errors. We obtain a 8-5-1 NARX NN architecture where 50% of the data shown in Figure 5.28 are used to train the network. The results for the five-step-ahead prediction are depicted in Figure 5.29.

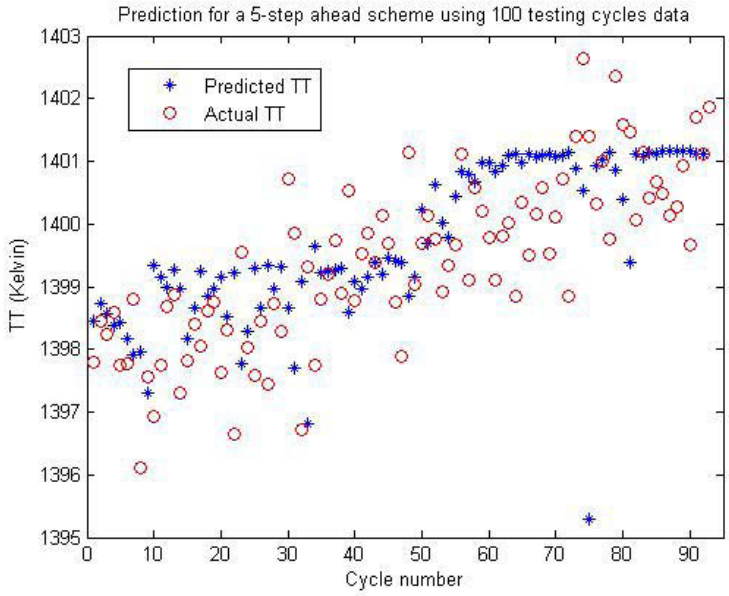

Figure 5.29: Actual vs. predicted TT for a 1% eroded turbine (5-step ahead).

The prediction error for the test data is shown graphically in Figure 5.30. The test error mean, standard deviation and *rmse* are 1.078 K, 1.081 K, and 1.155 K, respectively which are also tabulated in Table 5.9. The errors are sufficiently small enough (less than 1% error).

In the previous chapter we have highlighted the importance of uncertainty management. To overcome this problem we add prediction bounds instead of merely relying on predicting the points. If most of the data are within these bands, the prediction is deemed satisfactory. For more information on how we find these bounds

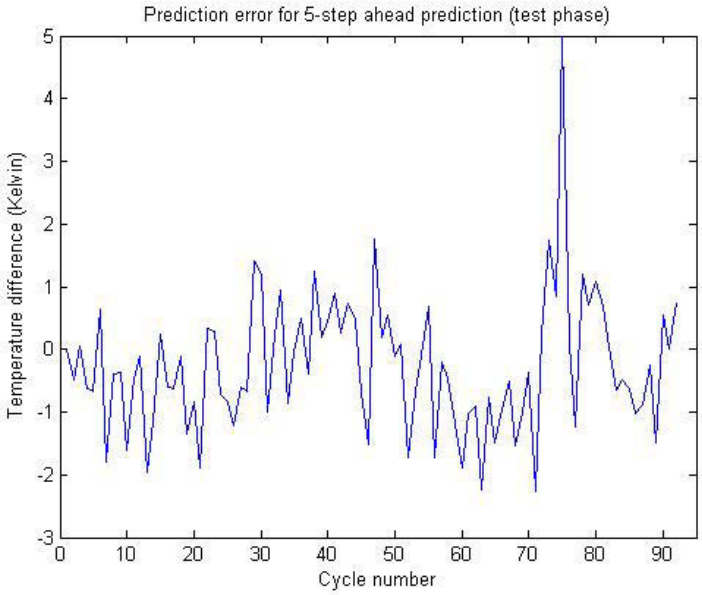

Figure 5.30: Temperature prediction error for a 1% eroded turbine (5-step ahead).

Table 5.9: Prediction error for EI=1% case 1.

μ_{ae}	1.078 (K)
σ_{ae}	1.081 (K)
$rmse$	1.155 (K)

one can refer to Chapter 4. The results are depicted in Figure 5.31. This confirms a satisfactory five-step ahead prediction when we take all the metrics into account. The data points are mostly within the bands (only 10 of the 95 data are outside which implies that 89.4% of them are inside the lower and upper bands) and error is less than 1% as compared to the nominal turbine temperature.

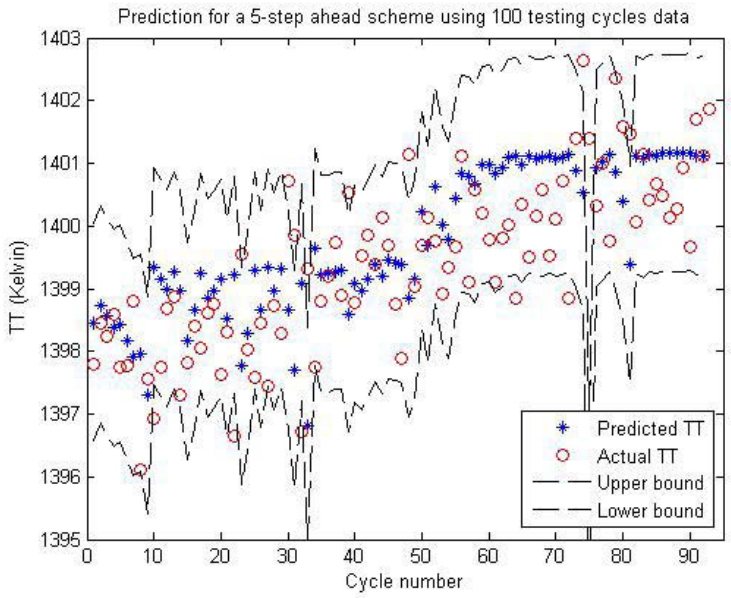

Figure 5.31: Actual vs. predicted TT for a 1% eroded turbine considering prediction bounds (5-step ahead).

EI = 1%: Case 2

In this case, we assume that the turbine has still the same level of erosion. We test our NARX NN for an eight-step-ahead temperature prediction and see if any change in the parameters is necessary. After a number of trials and errors it is understood that all the parameters can be kept the same except for the delays, which are changed to $d_u = 3$ and $d_y = 4$, and this results in a 9-5-1 NARX NN. The prediction results are depicted in Figure 5.32. Moreover, the errors between the actual and the predicted temperatures are shown in Figure 5.33.

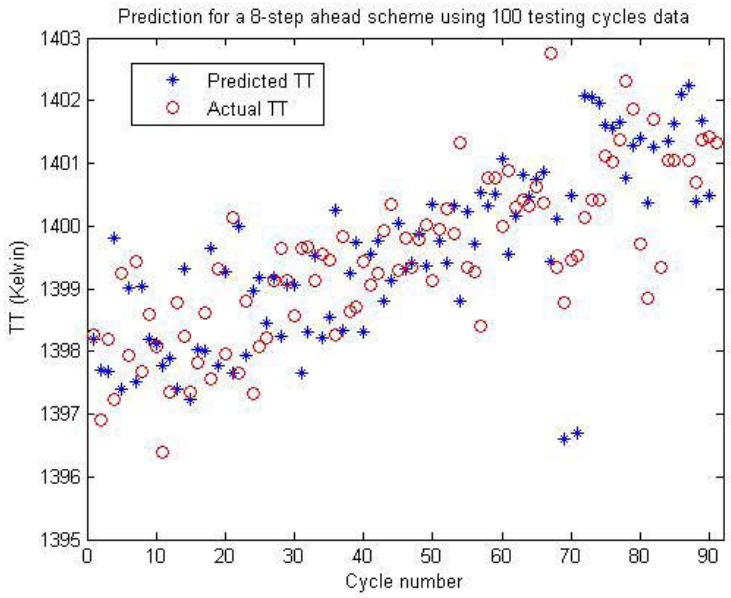

Figure 5.32: Actual vs. predicted TT for a 1% eroded turbine (8-step ahead).

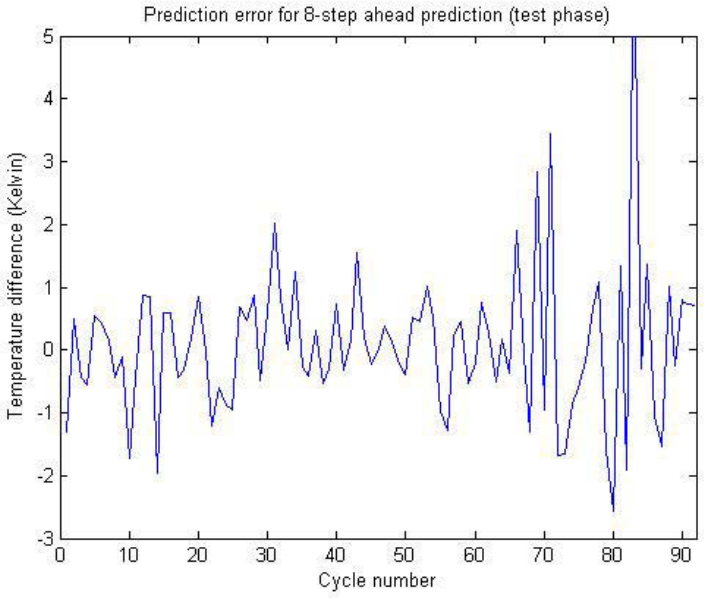

Figure 5.33: Temperature prediction error for a 1% eroded turbine (8-step ahead).

Table 5.10: Prediction error for EI=1% case 2.

μ_{ae}	0.951 (K)
σ_{ae}	1.175 (K)
$rmse$	1.229 (K)

We have also found the error statistics quantitatively and presented them in Table 5.10. The actual and predicted values are in average $1K$ different. The error is still within the reasonable and acceptable ranges (less than 1% error in prediction). This demonstrates that we can rely on this network for an eight-step-ahead prediction.

To complete this case, we have also added the lower and upper prediction bounds to overcome uncertainties associated with the prediction and determine if the data points are within these ranges. The bounds and the predicted temperatures for this case are depicted in Figure 5.34.

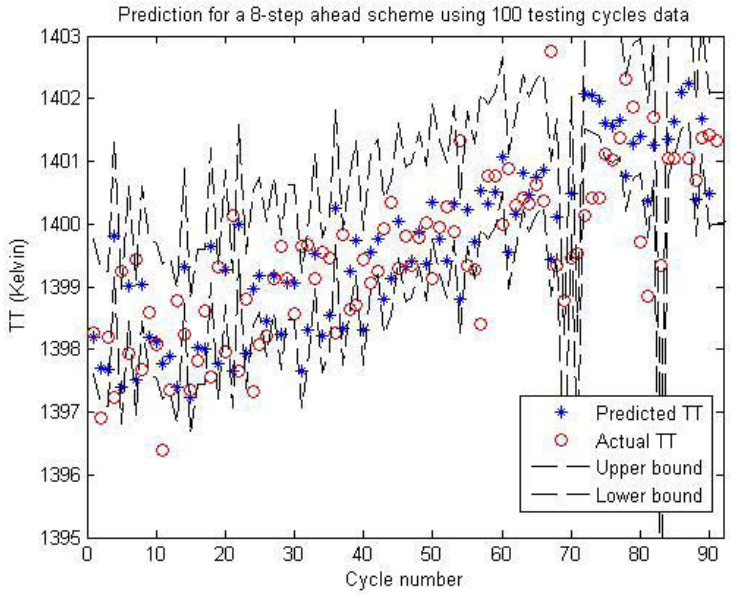

Figure 5.34: Actual vs. predicted TT for a 1% eroded turbine considering prediction bounds (8-step ahead).

It follows that error results are quite good. Note that 80 of the 92 data points are

inside the bands which is equal to 87% of them. In the next case we try to find the largest cycle possible for which an acceptable prediction horizon can be achieved.

EI = 1%: Case 3

In this case we keep the same number of input and output delays and find the maximum number of cycles ahead possible. The measure for that limit is obtained by observing the mean and the standard deviation of the error. We do not go beyond the 1% error level. Our developed 9-5-1 NN is capable of 17-step ahead prediction and the comparison results confirm this conclusion in Figure 5.35.

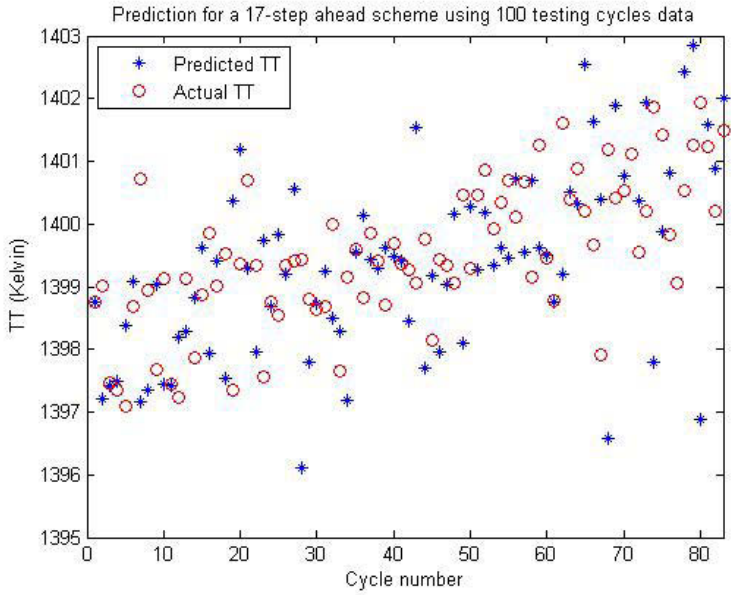

Figure 5.35: Actual vs. predicted TT for a 1% eroded turbine (17-step ahead).

It can be seen that our 9-5-1 NARX NN is capable of giving the temperature in the 17^{th} cycle from the present time. The error is depicted in Figure 5.36. Note that the error starts increasing at the end and we stop increasing the step ahead beyond this point.

The error mean, standard deviation and rmse are shown in Table 5.11, followed by the prediction bounds depicted in Figure 5.37. Note that 93.9% of the data points are

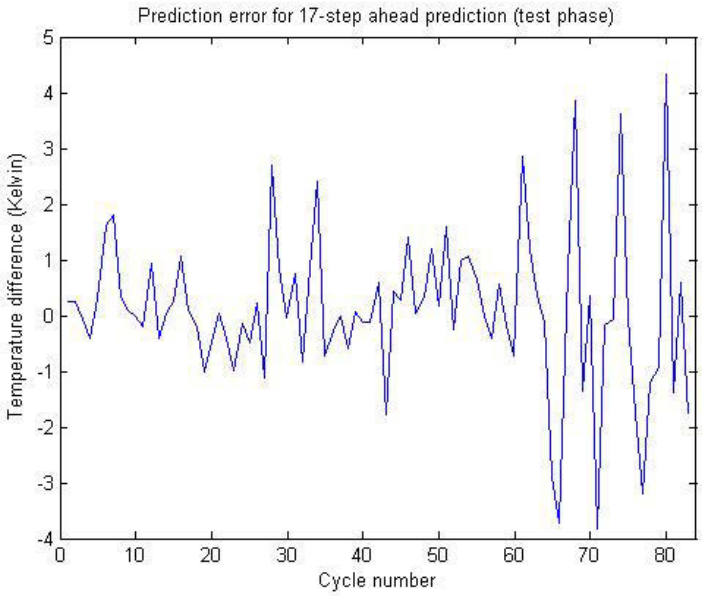

Figure 5.36: Temperature prediction error for a 1% eroded turbine (17-step ahead).

Table 5.11: Prediction error for EI=1% case 3.

μ_{ae}	0.735 (K)
σ_{ae}	2.080 (K)
$rmse$	2.078 (K)

inside the prediction bands. We could obtain better results by increasing the number of delays but it will cost us computational complexity and thus we stop going further at this point.

5.1.4.2 Second Scenario: EI = 2%

In this scenario we follow the procedure that was followed in the first scenario. First, we have to generate a proper data set. The standard take-off duration is assumed to be 20 seconds. We run the simulations 200 times (200 cycles) and pick up the 12^{th} second of each take-off and construct the data set. A 2% erosion index is equivalent to a 2% drop in the efficiency and a 1% drop in the turbine mass flow rate. We

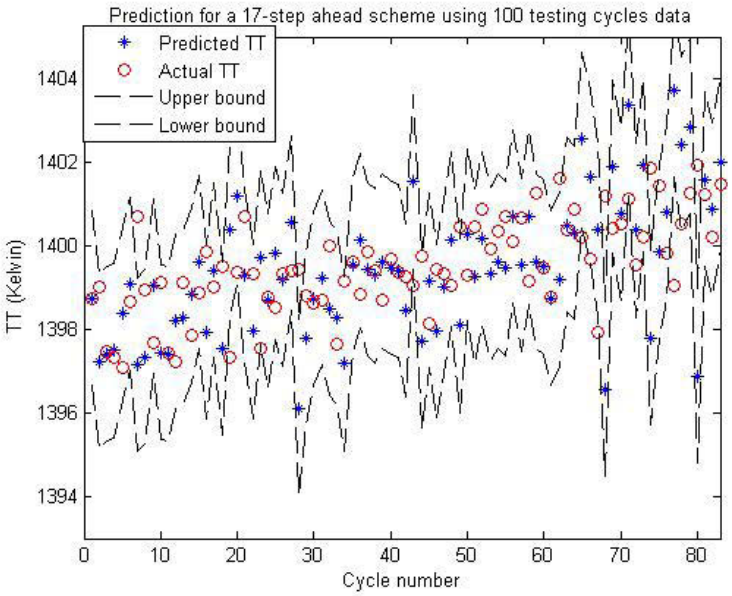

Figure 5.37: Actual vs. predicted TT for a 1% eroded turbine considering prediction bounds (17-step ahead).

also assume that the erosion levels remain the same. If the erosion remains in the system it causes and increases the effects in the turbine temperature. We train the NARX neural network to predict the temperature of an eroded turbine for certain steps ahead. Our data set considering the measurement noise is depicted in Figure 5.38 where all the temperatures are measured in Kelvin.

EI = 2%: Case 1

We start off the first case by a five-step ahead prediction. We use the same 8-5-1 NARX NN derived in Case one of the previous scenario to verify if this network is able to earn faster dynamics and more significant changes occurring in the system. The 8-5-1 structure means that in addition to the fuel flow and the turbine temperature, three delayed values of each is fed into the NARX neural network as inputs which all together are 8 inputs. We have five neurons in the hidden layer and the number of output is one which is the turbine temperature subjected to 2% erosion. We still

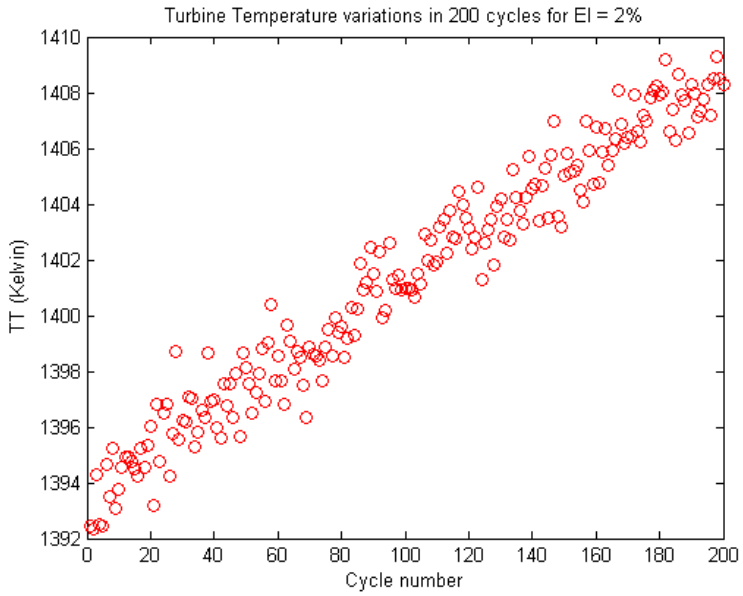

Figure 5.38: Turbine temperature variations due to an eroded turbine with EI= 2%.

Table 5.12: Prediction error for EI=2% case 1.

μ_{ae}	1.132 (K)
σ_{ae}	1.400 (K)
$rmse$	1.566 (K)

use only half of the data for network training. This shows one of the advantages of the NARX network which requires a shorter training time. This advantage is more highlighted when working with larger data sets. The remaining half of the data are used for testing the prediction results and are depicted and compared in Figure 5.39 and the prediction error is shown in Figure 5.40.

By studying the error statistics it is understood that the two values at each point are $1.132k$ in average different from each other. The summary of the statistics is given in Table 5.12. Lower than 1% error confirms a proper prediction result. In addition to the error values, the prediction bounds confirm that with the proposed NARX NN architecture, a five-step-ahead prediction is possible for the turbine temperature. The bounds when compared to the predicted and actual temperatures are depicted

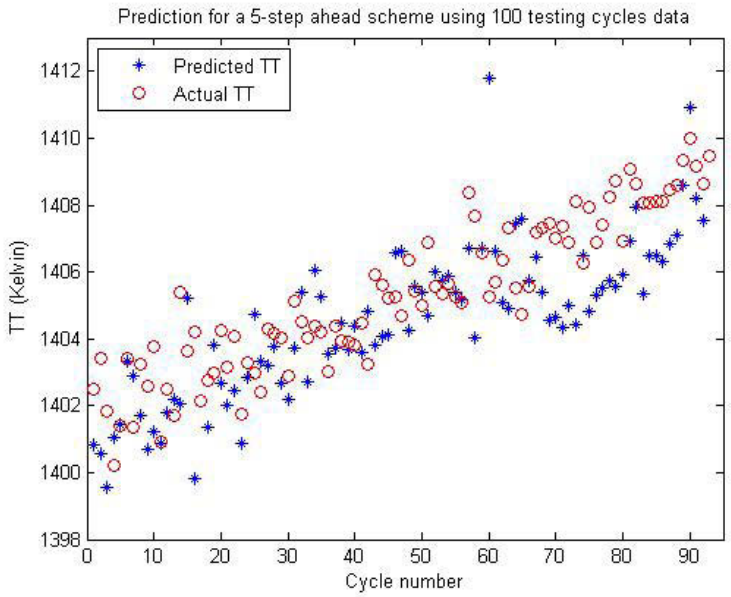

Figure 5.39: Actual vs. predicted TT for a 2% eroded turbine (5-step ahead).

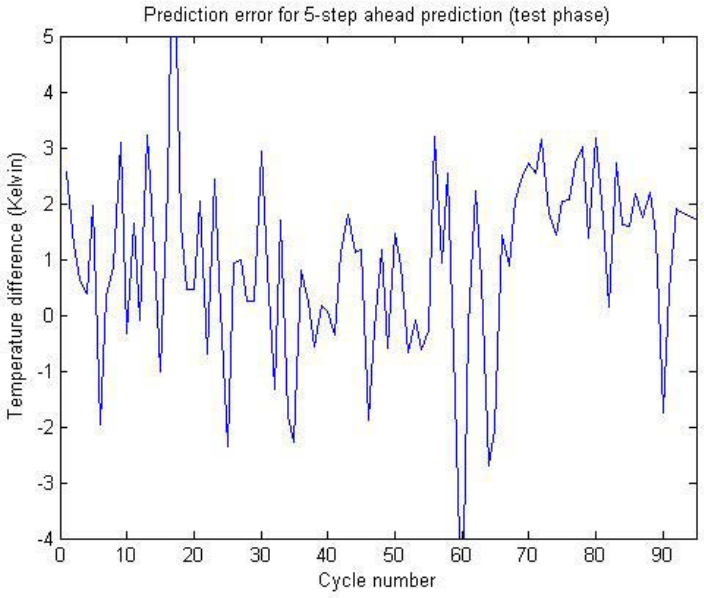

Figure 5.40: Temperature prediction error for a 2% eroded turbine (5-step ahead).

in Figure 5.41. Note that 88.4% of the data points are inside the prediction bands.

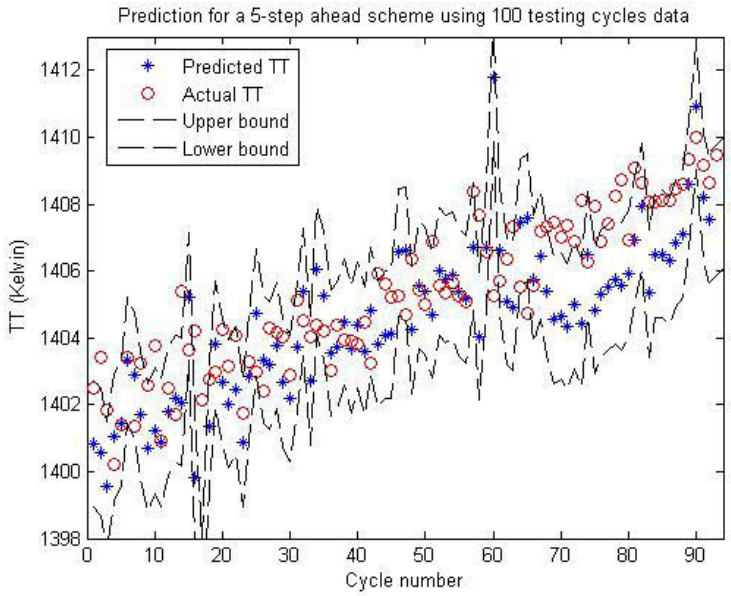

Figure 5.41: Actual vs. predicted TT for a 2% eroded turbine considering prediction bounds (5-step ahead).

FI = 2%: Case 2

In the second case the goal is to obtain the results for a ten-step-ahead prediction. The engine is still under 2% erosion. After comparing the values and studying the error we concluded that one more delayed version of the measurements is needed for the network input to obtain satisfactory training and testing results.

The prediction results for the 10-5-1 network implying $d_u = 4$ and $d_y = 4$ are depicted in Figure 5.42. It is required the mean, standard deviation and *rmse* of the error to be as low as possible. The resulting errors are depicted in Figure 5.43. One should note that the predicted points are mostly above the real turbine temperatures in this case. The negative error and higher absolute error mean confirm this observation. Moreover, the absolute mean, *std* and *rmse* of the error are found in Table 5.13.

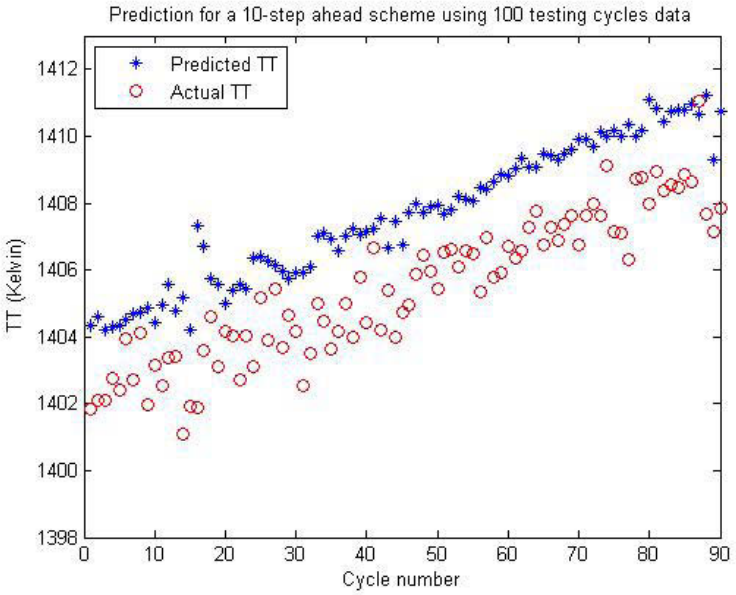

Figure 5.42: Actual vs. predicted TT for a 2% eroded turbine (10-step ahead).

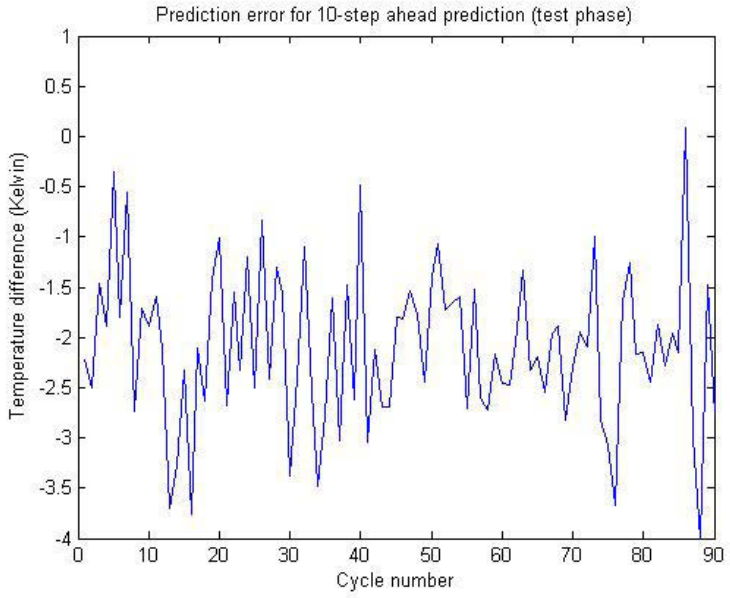

Figure 5.43: Temperature prediction error for a 2% eroded turbine (10-step ahead).

Table 5.13: Prediction error for EI=2% case 2.

μ_{ae}	2.172 (K)
σ_{ae}	0.984 (K)
$rmse$	2.379 (K)

The bounds are predicted and depicted in Figure 5.44. More than half of the data are inside the band. Defining the bounds is very important especially when we are taking larger steps ahead. We now complete the second scenario and proceed to cases where the erosion level is increased to 3%.

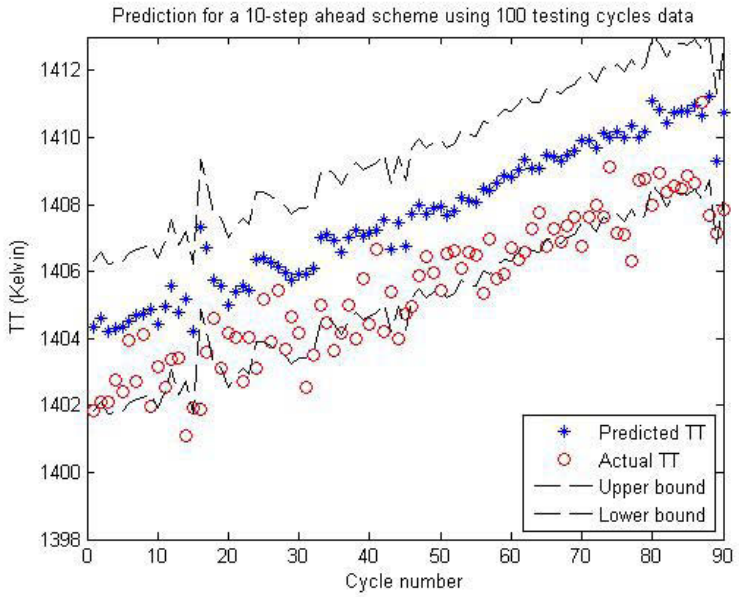

Figure 5.44: Actual vs. predicted TT for a 2% eroded turbine considering prediction bounds (10-step ahead).

5.1.4.3 Third Scenario: EI = 3%

In our final scenario we assume that a 3% erosion has occurred in the turbine section. This is equivalent to 3% drop in the turbine efficiency and 1.5% drop in the turbine mass flow rate. A proper data set is vital as it can affect the performance of the

neural network. We have considered the measurement noise as well. The resulting data set are depicted in Figure 5.45. In the following, we perform the last series of

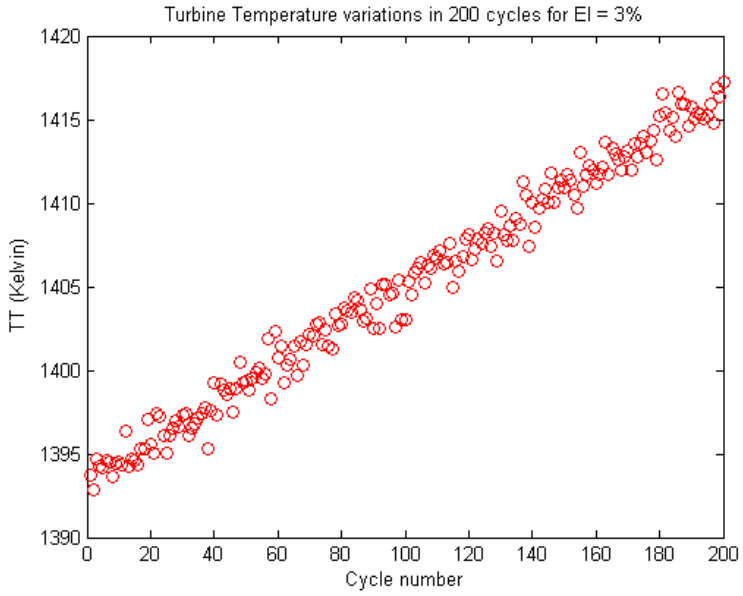

Figure 5.45: Turbine temperature variations due to a 3% eroded turbine.

our investigation.

EI = 3%: Case 1

In this case the engine is subjected to a 3% erosion. When the degradation level is higher, the changes in the turbine temperature are higher as a consequence. We should verify if our NARX network is still capable of learning the dynamics of the degradation and projecting the information into the future. We start by examining the same 8-5-1 neural network used in the first case of the two previous erosion scenarios with the same number of input and output delays. Half of the data are used for training and compared to the similar cases shown earlier with different networks. It follows that that the NARX networks are trained easier and with less amount of data. The test phase results are depicted in Figure 5.46.

The comparison results in addition to the errors are depicted in Figure 5.47, which

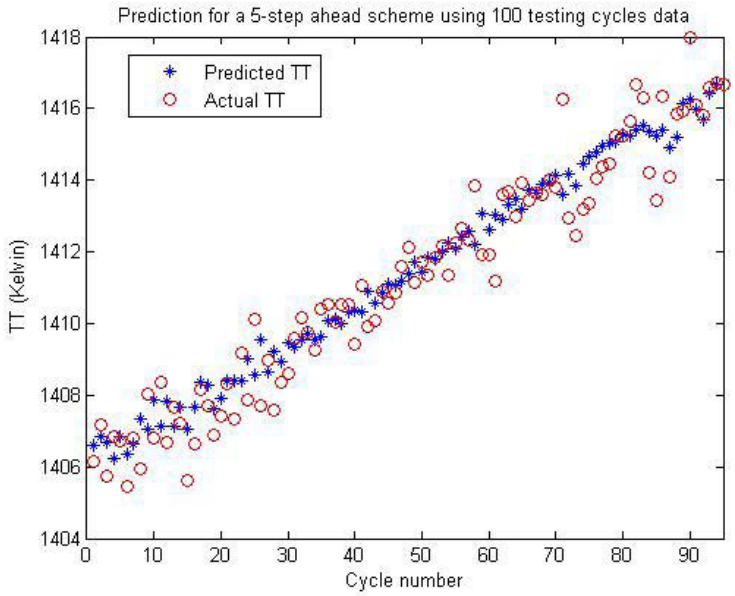

Figure 5.46: Actual vs. predicted TT for a 3% eroded turbine (5-step ahead).

Table 5.14: Prediction error for EI=3% case 1.

μ_{ae}	0.708 (K)
σ_{ae}	0.825 (K)
$rmse$	0.883 (K)

show that the network has properly learnt the erosion dynamics and its effects on the turbine temperature. According to the error results, the dynamics is learnt quite well for the entire testing phase. The mean, standard deviation and $rmse$ of the error are $0.7084K$, 0.8254, and 0.8833, respectively as shown in Table 5.14.

The above represents a very good error level as it is less that 1% as compared to the nominal turbine temperature in the take-off mode. Moreover, when one adds the lower and upper prediction bounds, one can observe that 92% of the data points are within the bands as only 7 data points are outside. The bands are depicted in Figure 5.48.

We will complete our investigation by extending the prediction horizon to ten

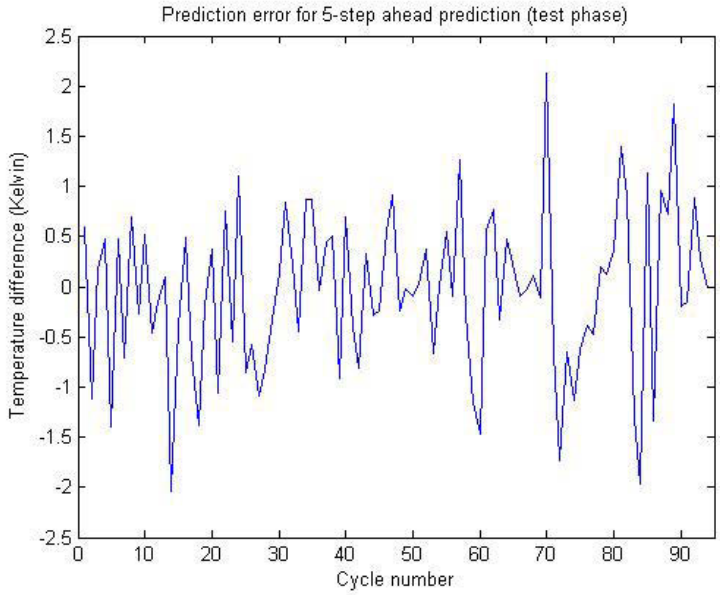

Figure 5.47: Temperature prediction error for a 3% eroded turbine (5-step ahead).

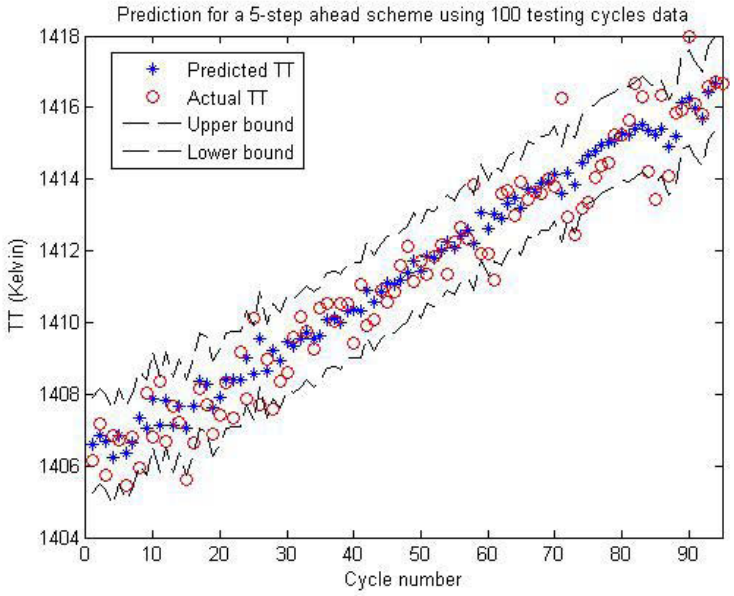

Figure 5.48: Actual vs. predicted TT for a 3% eroded turbine considering prediction bounds (5-step ahead).

cycles ahead in the next case.

EI = 3%: Case 2

In the last case that we present in this chapter, to be consistent with the previous examples we test the network for a 10-step ahead prediction. We start examining the 8-5-1 NARX network. If the results are not satisfactory we make some changes in the parameters of our proposed network. The prediction results (test phase) are depicted in Figure 5.49. The results are satisfactory but not as good as the previous case as the error starts to increase at the end of the prediction horizon.

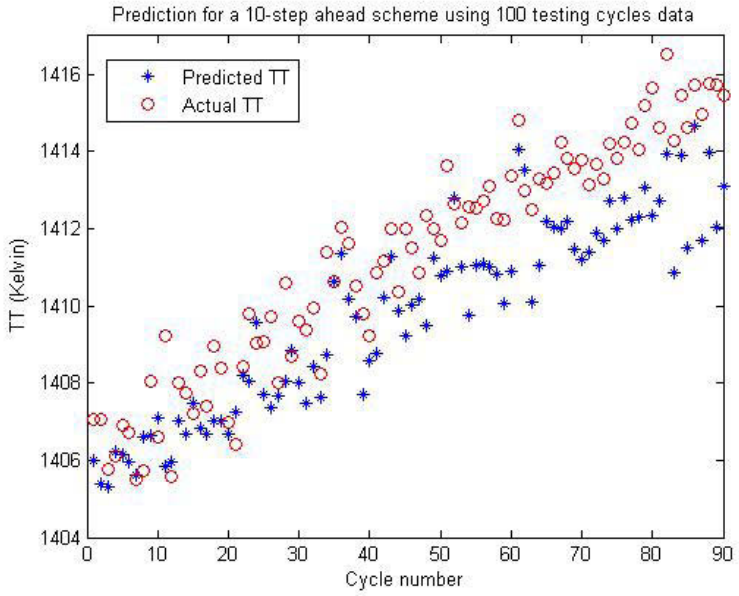

Figure 5.49: Actual vs. predicted TT for a 3% eroded turbine (10-step ahead).

The error results for a ten-steps-ahead prediction are shown in Figure 5.50. Moreover, the error mean, *std* and *rmse* are tabulated in Table 5.15.

The above error results are reasonable. We have also checked the prediction bounds to be able to comment on the prediction performance (Figure 5.51). Almost all of the points are inside except 6 of them. This implies that 93% of them are within

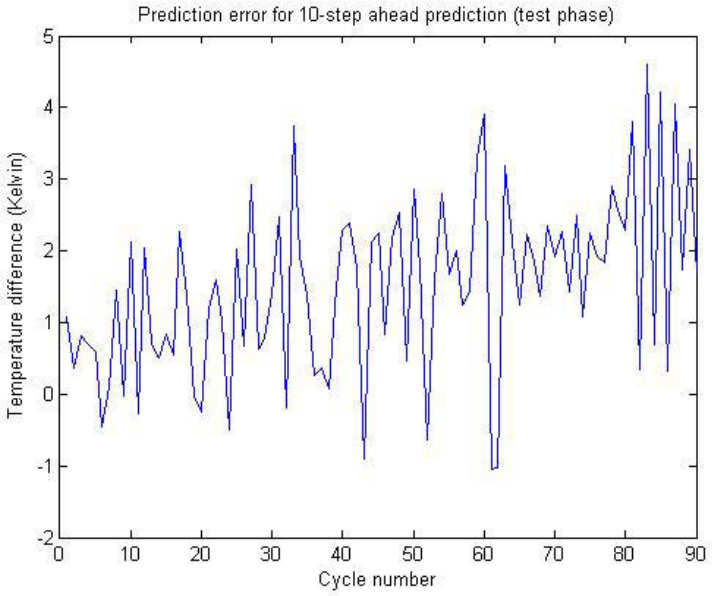

Figure 5.50: Temperature prediction error for a 3% eroded turbine (10-step ahead).

Table 5.15: Prediction error for EI=3% case 2.

μ_{ae}	1.144 (K)
σ_{ae}	1.167 (K)
$rmse$	1.481 (K)

the bands. The ten-step-ahead prediction is an accurate prediction and we can ensure that on average our prediction of the turbine temperature is quite close to the real value by only a 1% error.

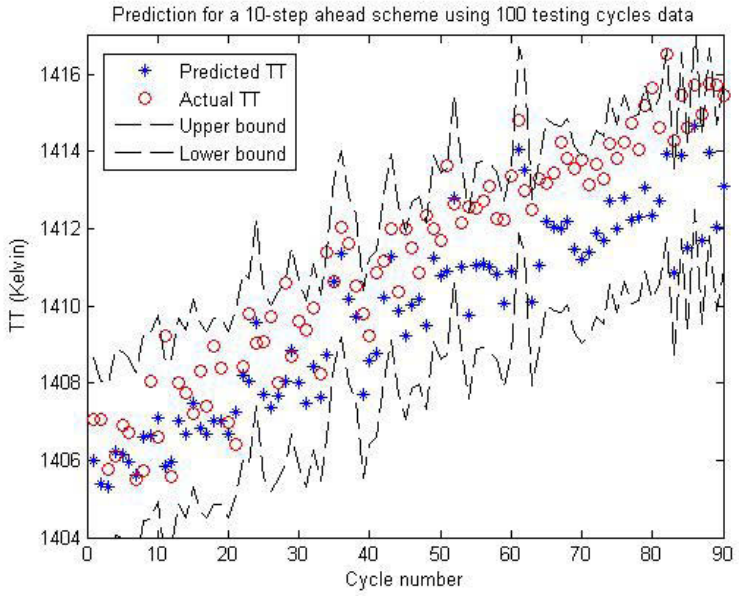

Figure 5.51: Actual vs. predicted TT for a 3% eroded turbine considering prediction bounds (5-step ahead).

This completes our simulation case using the NARX neural network. As mentioned earlier after trials and errors it is concluded that this network is easier to handle and to work with as compared to the other network architecture (RNN). Besides, fewer amount of data are needed to achieve the same accuracy. We have obtained promising results in the numerous cases presented in this chapter. Knowing the future values of the temperature allows one to schedule the maintenance based on the predicted condition of the engine. In the next section, we will compare all the proposed networks and compare their prediction performance. Before we proceed, let us summarize all the simulation results obtained using the NARX NN in the following subsection.

Table 5.16: Summary of the results for FI=1% scenarios using NARX NN.

FI = 1%	Network Structure	μ_{ae}	σ_{ae}	$rmse$
$d_u = 3$; $d_y = 3$ / l = 3	8-5-1	0.970	1.149	1.220
$d_u = 4$; $d_y = 5$ / l = 11	11-5-1	1.535	1.721	1.860
$d_u = 4$; $d_y = 5$ / l = 16	11-5-1	2.075	2.373	2.479

5.1.5 Summary of the Simulation Results

A summary of the obtained results for all the case are tabulated in the following two parts. These correspond to the quantitative measures that show how changing the number of input and output delays can let us increase the number of cycles ahead to be predicted. Error mean (μ_{ae}), error standard deviation (σ_{ae}) and the mean squared-error ($rmse$) are calculated and compared in these tables.

One can note how changing the number of the delays can improve the results. Our goal was to maintain lower than 1% error, l is the number of prediction steps ahead, d_u is the number of input delays and d_y is the number of output delays fed back to the network input.

Summary of the results for the fouling scenarios

The results for the fouling scenarios are presented in Tables 5.16 - 5.18. The results are categorized according to the number of delays, network structure and steps ahead prediction, The prediction error mean, the standard deviation and the $rmse$ are presented as well . When fouling occurs, the system efficiency drops and the mass flow rate is dropped as well in a linear 2:1 fashion. We did not go beyond the 3% of fouling as engine washing is recommended after this level. One can verify that if one wants to keep the error low and increase the cycles ahead in time, one has to increase the number of input and output delays given to the network as inputs.

Table 5.17: Summary of the results for FI=2% scenarios using NARX NN.

FI = 2%	Network Structure	μ_{ae}	σ_{ae}	$rmse$
$d_u = 3$; $d_y = 3$ / l = 5	8-5-1	1.762	1.073	1.293
$d_u = 4$; $d_y = 4$ / l = 10	10-5-1	0.873	0.989	0.987
$d_u = 7$; $d_y = 8$ / l = 16	17-5-1	1.215	1.810	1.900

Table 5.18: Summary of the results for FI=3% scenarios using NARX NN.

FI = 3%	Network Structure	μ_{ae}	σ_{ae}	$rmse$
$d_u = 3$; $d_y = 3$ / l = 3	8-8-1	0.991	1.451	1.749
$d_u = 4$; $d_y = 4$ / l = 8	10-5-1	1.023	1.078	1.339

Summary of the results for erosion scenarios

In this part we summarize the prediction results when the engine is under different levels of erosion from 1% to 3%. Erosion mostly occurs in the turbine and causes efficiency drop and increases the mass flow rate by a linear relation of 2:1. By studying Tables 5.19 - 5.21 one can observe that to achieve higher steps ahead, more delayed inputs and outputs need to be fed back to the RNN inputs otherwise the error level would become high and goes beyond our desired error level (less than 1%).

Table 5.19: Summary of the results for EI=1% scenarios using NARX NN.

EI = 1%	Network Structure	μ_{ae}	σ_{ae}	$rmse$
$d_u = 3$; $d_y = 3$ / l = 5	8-5-1	1.078	1.081	1.155
$d_u = 3$; $d_y = 4$ / l = 8	9-5-1	0.951	1.175	1.229
$d_u = 3$; $d_y = 4$ / l = 17	9-5-1	0.735	2.080	2.078

Table 5.20: Summary of the results for EI=2% scenarios using NARX NN.

EI = 2%	Network Structure	μ_{ae}	σ_{ae}	$rmse$
$d_u = 3$; $d_y = 3$ / l = 5	8-5-1	1.132	1.400	1.566
$d_u = 4$; $d_y = 4$ / l = 10	10-5-1	2.172	0.984	2.379

Table 5.21: Summary of the results for EI=3% scenarios using NARX NN.

EI = 3%	Network Structure	μ_{ae}	σ_{ae}	$rmse$
$d_u = 3$; $d_y = 3$ / l = 5	8-5-1	0.708	0.825	0.883
$d_u = 3$; $d_y = 3$ / l = 10	8-5-1	1.144	1.167	1.481

5.2 Evaluation of the Results Using the Normalized Akaike Information Criterion (NAIC)

After presenting all the simulation results, we now need a measure or criterion to enable us compare the different neural network architectures used for prediction. It is worth mentioning the fact that any model of evolution we can construct is never going to be the "true model" that generated the data we observed [201]. In other words there is always a deviation between the real and the predicted values from our model.

Different model selection criteria exist in the literature such as the Bayesian information criterion (BIC) and the Akaike information criterion (AIC) [202]. We are interested in finding the best fit using the available data. A method called the normalized Akaike information criterion (NAIC) is employed here which is based on the classical maximum likelihood estimation procedure [203]. According to [202], NAIC is a "versatile procedure for statistical model identification which is free from the ambiguities inherent in the application of conventional hypothesis testing procedure." Akaike criterion has a large number of uses in different model selection applications

such as curve-fitting [204].

One can test the goodness of the prediction model by using the NAIC. We need to show which of the neural networks presented in Chapters 4 and 5 outperform the others. An investigation is therefore performed in this section to compare the effectiveness of the prediction results provided by different neural network architectures [79]. Two main issues should be addressed in the selection of suitable models for prediction purposes. Firstly, we need a proper data set for investigating the prediction capabilities. Secondly, an evaluation method is necessary for the comparison study. Our data set for network testing and training is obtained from a developed engine simulation program which was described in Chapter 2. The second main issue is addressed by employing the popular normalized Akaikeinformation criterion (NAIC). The NAIC is used as a basis for comparing the accuracy of several prediction models and selecting the model with the best performance [205].

The notion of the NAIC metric is defined as follows:

$$NAIC = ln\sigma^2 + \frac{2\rho}{N} \tag{5.2.1}$$

where σ^2 is the variance of the prediction error calculated by squaring the *std* of the error presented in tables in this thesis, ρ is the total number of parameters of the NN model and N is the total number of samples in the predicted data set.

A smaller NAIC value implies a better prediction model. Hence one can find the most appropriate prediction method (model) by using this criterion as the one with the smallest NAIC. In the following we compare our proposed neural networks using the NAIC metric for some of the prediction results presented in this thesis.

5.2.1 Evaluation of the Prediction Results

The prediction results by using the RNN are summarized in Tables 4.23 - 4.28 for both degradation types. Moreover, the prediction results using the NARX NN are presented in Tables 5.16 - 5.21. We clarify the NAIC calculations by giving one example.

Let us consider the first row of the entry in Table 5.16. This entry corresponds to a 1% of fouling effective in the engine compressor. According to the table, to achieve a 3-step ahead prediction, 3 input delays and 3 output delays were regarded as the NN input in addition to the current input and output to predict the turbine temperature in 3 cycles ahead from present time. The number of neurons in the hidden layer is set as five resulting in an 8-5-1 NN. The number of parameters in this case is 54. This is the sum of the 8 parameters from the input, 40 parameters represent the connections between the 8 input nodes and five hidden nodes ($8 \times 5 = 40$), 5 parameters represent the connections from the five hidden neurons to the one network output, and finally one parameter represents the feedback connection from the output neuron to the input node. Therefore, in this case $\rho = 54$.

As defined earlier σ_{ae} is the standard deviation of the prediction error which is the difference between the output of the NN and the target output corresponding to the test data set. The variance or *var* is obtained by squaring the *std*. In this case the standard deviation is equal to 1.149 and the variance is thus equal to 1.320. N is the number of data used for prediction which is 50. Therefore, $NAIC = 2.437$ (Table 5.28, first row). The same procedure is followed for finding the other NAIC values for the other cases.

Remark. One can observe that in the NAIC presented by equation (5.2.1), the contribution of the first term which is the error variance to the NAIC value is more

Table 5.22: NAIC values for FI=1% scenarios using RNN.

FI = 1%	Error *std*	Error *var*	Parameters used	test data set size (N)	NAIC
Case 1	1.384	1.915	30	30	2.650
Case 2	1.829	3.345	42	30	4.007
Case 3	1.810	3.276	42	30	3.986

Table 5.23: NAIC values for FI=2% scenarios using RNN.

FI = 2%	Error *std*	Error *var*	Parameters used	test data set size (N)	NAIC
Case 1	1.948	3.794	30	30	3.333
Case 2	3.632	13.191	77	30	7.712

effective and more significant than the second term which uses the number of parameters and the number of observations. This shows that a lower error variance is more important than having llower number of parameters. In other words, increasing the number of parameters, as long as the computational cost is acceptable may result in a better and more accurate prediction model. We will verify this argument by comparing some of the same prediction results from our different developed neural networks. NAIC is calculated and presented in the following tables for all the simulation cases.

NAIC Values Using the RNN

NAIC values for the fouling scenarios obtained from our RNN are presented in Tables 5.22 - 5.24. NAIC values for the erosion scenarios and cases obtained from our RNN are presented in Tables 5.25 - 5.27.

Table 5.24: NAIC values for FI=3% scenarios using RNN.

FI = 3%	Error *std*	Error *var*	Parameters used	test data set size (N)	NAIC
Case 1	1.705	2.907	30	30	3.067
Case 2	5.369	28.826	36	20	6.961
Case 3	1.100	1.210	42	20	3.790

Table 5.25: NAIC values for EI=1% scenarios using RNN.

EI = 1%	Error *std*	Error *var*	Parameters used	test data set size (N)	NAIC
Case 2	1.971	3.884	66	30	5.757
Case 3	2.410	5.808	66	30	6.159
Case 4	3.989	15.912	77	30	7.900

Table 5.26: NAIC values for EI=2% scenarios using RNN.

EI = 2%	Error *std*	Error *var*	Parameters used	test data set size (N)	NAIC
Case 1	1.216	1.478	66	20	6.991
Case 2	2.085	4.347	77	20	9.169

Table 5.27: NAIC values for EI=3% scenarios using RNN.

EI = 3%	Error *std*	Error *var*	Parameters used	test data set size (N)	NAIC
Case 1	1.674	2.802	66	30	7.630
Case 2	2.349	5.517	77	30	9.408

Table 5.28: NAIC values for FI=1% scenarios using NARX NN.

FI = 1%	Error *std*	Error *var*	Parameters used	test data set size (N)	NAIC
Case 1	1.149	1.320	54	50	2.437
Case 2	1.721	2.961	72	50	3.965
Case 3	1.810	3.276	42	50	3.986

Table 5.29: NAIC values for FI=2% scenarios using NARX NN.

FI = 2%	Error *std*	Error *var*	Parameters used	test data set size (N)	NAIC
Case 1	1.073	1.151	54	50	2.300
Case 2	0.989	0.978	66	50	2.617
Case 3	1.810	3.276	108	50	5.506

NAIC Values Using the NARX NN

In Tables 5.28 through 5.30 one can find the NAIC values for a fouled engine, when the prediction results in each case are obtained from our proposed NARX neural network. NAIC values for the erosion scenarios and cases obtained from our NARX neural network are presented in Tables 5.31 - 5.33.

5.2.2 Comparison of the Results

After calculating the normalized Akaike information criterion (NAIC) value for different cases corresponding to all simulation results presented in this thesis we now

Table 5.30: NAIC values for FI=3% scenarios using NARX NN.

FI = 3%	Error *std*	Error *var*	Parameters used	test data set size (N)	NAIC
Case 1	1.451	2.105	54	50	2.904
Case 2	1.078	1.162	66	50	2.790

Table 5.31: NAIC values for EI=1% scenarios using NARX NN.

EI = 1%	Error *std*	Error *var*	Parameters used	test data set size (N)	NAIC
Case 1	1.081	1.168	54	50	2.315
Case 2	1.175	1.380	60	50	2.722
Case 3	2.080	4.326	60	50	3.864

Table 5.32: NAIC values for EI=2% scenarios using NARX NN.

EI = 2%	Error *std*	Error *var*	Parameters used	test data set size (N)	NAIC
Case 1	1.400	1.960	54	50	2.832
Case 2	0.984	0.968	66	50	2.607

Table 5.33: NAIC values for EI=3% scenarios using NARX NN.

EI = 3%	Error *std*	Error *var*	Parameters used	test data set size (N)	NAIC
Case 1	0.825	0.680	54	50	1.775
Case 2	1.167	1.361	54	50	2.468

compare the two network structures utilized here for prediction purposes. In all simulation results the time evolution of the turbine temperature for an engine subjected to different levels of degradations were obtained. The two degradations that are studied here are fouling and erosion. Fouling and erosion are the two main causes of engine soft degradation. They are called soft because they do not cause abrupt changes in the dynamics of the system and their effects can become sever throughout the time. They are modelled by using the changes in the component efficiency and mass flow rate.

Changes in the efficiency and mass flow rate in return change the gas path measurements. Turbine temperature is considered as a suitable candidate which reflects the effects of these degradations on the engine system. Three levels of degradations are considered. To be able to compare the prediction capability of our proposed neural networks for degradation trend prognostics, one should compare the same cases. An evaluation measure is needed. The NAIC is a suitable and popular evaluation method that was introduced earlier in this section.

We may now compare the results of similar cases acquired from the proposed neural networks in Chapters 4 and 5. We compare the results for different degradations, type by type and for each level separately. The comparison results are shown in tables followed by the discussions where in the tables l denotes the number of steps ahead in time.

In the first scenario we considered a fouled compressor under 1% of fouling. For a 3-step ahead prediction the NAIC value using the RNN is 2.650 and the NAIC for the NARX NN is 2.437 (Table 5.34). The smaller the NAIC value, the more effective our prediction is. NAIC combines both the preciseness and the computational cost by using both the prediction error variance and the number of parameters used. The results obtained from the NARX network are slightly better. In the RNN the

Table 5.34: Comparison of the NAIC values for FI=1%.

FI=1%	NAIC RNN	NAIC NARX NN
$l = 3$	2.650	2.437
$l = 10/11$	3.986	3.965

Table 5.35: Comparison of the NAIC values for FI=2%.

FI=2%	NAIC RNN	NAIC NARX NN
$l = 3/5$	3.333	2.300
$l = 10$	7.712	32.617

number of parameters is 30 and in the NARX NN it is equal to 45. However, the NAIC is smaller when applying the NARX NN. This confirms our earlier *remark* that the contribution of the number of parameters is less in this performance evaluation measure, implying that obtaining a lower level of error is more important in the case of prediction.

The last entry of Table 5.34 again shows that the NARX neural network has a better prediction performance (although the number of steps ahead are increased) thus resulting in a more reliable prognostics scheme.

The comparison results for the second fouling scenario are found in Table 5.35. The NAIC value is smaller when we use the NARX architecture for both step-ahead cases. Although some of the cases are not exactly the same, the results are still comparable. The reason is that in the NARX the step-ahead value is higher, and still the NAIC is smaller. This highlights the superiority of the NARX network even more.

In Table 5.36, the comparison of the results of the third scenario for a 3% fouled engine are shown. We have considered a 3-step and an 8-step ahead prediction using

Table 5.36: Comparison of the NAIC values for FI=3%.

FI=3%	NAIC RNN	NAIC NARX NN
$l = 3$	3.067	2.904
$l = 8$	3.970	2.790

Table 5.37: Comparison of the NAIC values for EI=1%.

EI=1%	NAIC RNN	NAIC NARX NN
$l = 5$	5.757	2.315
$l = 8$	6.159	2.722
$l = 15/17$	7.900	3.860

the two schemes. One can see that in contrast with FI=1% results, now the difference in the NAIC values from the two methods is increased. This shows us the predominance of the NARX methodology when dealing with faster dynamics (larger changes). This shows that NARX NNs are capable of learning long-term dependencies as well.

In the following three tables, the results for the three erosion scenarios are compared and tabulated. For EI=1% presented in Table 5.37, we have compared three cases. Again, the NARX prediction results are more accurate as for all cases the NAIC values stay in the same range but the RNN NAIC values increase more significantly. The prediction horizon is increased even larger, employing the NARX NN which confirms this better capability.

For an increased level of erosion, the NARX network shows its advantage over the RNN. This can be verified from Table 5.38 where we are dealing with a 2% of fouling.

In our final investigation, the NAIC comparison results for the EI=3% are shown in Table 5.39. There is a significant difference between the calculated NAICs for the two different networks. This shows that by utilizing the NARX NN is especially more

Table 5.38: Comparison of the NAIC values for EI=2%.

EI=2%	NAIC RNN	NAIC NARX NN
$l = 5$	6.991	2.832
$l = 8/10$	9.169	2.607

Table 5.39: Comparison of the NAIC values for EI=3%.

EI=3%	NAIC RNN	NAIC NARX NN
$l = 5$	7.630	1.775
$l = 10$	9.408	2.468

advantageous for predicting the effects of the erosion on the turbine temperature as the error levels are much less.

In all the cases, the evaluation of both methods show that the NARX NN performance in prediction is better. In brief, we prefer to utilize nonlinear autoregressive neural networks with exogenous input to achieve a more reliable prognostics. This conclusion makes sense as the NARX NN is a more developed version of the recurrent neural network and is in fact a dynamical recurrent neural network. It is equipped with the advantageous of the RNNs. Moreover, it has some delay units (feeding input delays to the NN input) in addition to the global feedback paths from the network output to the network input. Our investigation in this thesis confirms this matter. Hence, a novel method is proposed in this thesis for degradation trend prognostics as part of a health monitoring scheme which can be later used for condition-based maintenance.

5.3 Conclusion

In this chapter, we proposed a novel neural network architecture for prediction purposes. The nonlinear autoregressive neural network (NARX) with exogenous input was employed to predict the degradation trends and the increase in the turbine temperature level for certain steps ahead in time. Towards this end, we conducted various simulation cases where the engine was under different types and levels of degradations. The fouling is considered to occur in the compressor and the erosion shows its effects mostly in the turbine. The challenge associated with the NARX architecture is determining the number of input and output delays. One has to perform trials and error to obtain acceptable error levels while the network parameters are kept low as well. We determine the suitability of the results by studying the error results. For each case the error mean, the standard deviation and the root mean square error are tabulated. Furthermore, to overcome uncertainty and to obtain a better insight into the goodness of the prediction, lower and upper prediction bounds were depicted for each case.

At the end of this chapter a metric called the normalized Akaike information criterion (NAIC) was introduced. Using such metrics enables us to compare the prediction results obtained from our two neural network architectures. The smaller NAIC value implies better prediction results. Akaike metric entails a trade-off between the prediction error and the number of parameters used in the network under study. The NARX network shows to be superior to the RNN especially when the degradation level is higher (faster dynamics) and also for erosion scenarios. This confirms that NARX neural network since is a combination of recurrent and dynamic networks, benefits from the features and advantages of both models.

Chapter 6

Conclusions and Future Work

6.1 Summary of the Thesis

Prognostics of the nonlinear dynamics of an aircraft engine is addressed in this thesis. Two artificial intelligence based approaches are employed to achieve this goal. Neural networks that were used in this thesis belong to data-based approaches. They are powerful tools that can learn the system dynamics from examples. They have a wide range of applications especially when the dynamical system model is not available and merely system measurements are available. In an aircraft engine, it is critical to predict the future health condition of the system as for the safety issues are concerned. Aging process and different degradations caused by environmental conditions are inevitable in a gas turbine engine, as in any other physical system. Moreover, condition-based maintenance can result in financial benefits as well. Hence, developing a reliable prognostics, condition monitoring and health management system is highly important.

performance of the neural networks is dependent on the data set they are provided with for training and testing. Hence, before training the network one needs a proper set of data that contains the degradation dynamics. Towards this end, we have

modelled the fouling and the erosion as two main causes of soft engine degradations. They were modelled by changing the efficiency and the mass flow rate of different engine components. These changes in return change the engine gas path measurements which are directly measurable by sensors located in different parts of the engine. Fouling occurs mostly in the compressor by decreasing the efficiency and decreasing the mass flow rate, whereas erosion occurs mostly in the turbine by decreasing the efficiency and increasing the mass flow rate. These models are investigated with a MatLab-based engine simulator and the results were validated by the GSP which is a gas turbine engine simulation program. Gas path measurement changes such as the component temperature, pressure and spool speed for an engine working in the take-off mode were studied when the engine is under different levels of fouling and erosion.

The first proposed neural network for prediction was the recurrent neural network (RNN). This network architecture is capable of learning short-term dependencies and therefore is suitable for prediction. The dynamics is learnt by the additional feedback from the network outputs to the network inputs. We had to adjust the number of the delays to be fed back to the network outputs. This usually was tedious and challenging as there is no general rule for determining the neural network parameters and architectures. We obtained suitable number of neurons in the hidden layer, data set size and delays through trials and errors. The goal was to make sure that the degradation dynamic is learnt while the number of delays are kept as low as possible to prevent larger computational storage and cost. Many simulation cases were presented to demonstrate the network prediction capabilities for both fouling and erosion degradations. For each case the prediction error was calculated and the error mean, error standard deviation and the mean square error were tabulated. Furthermore, because of the uncertainty and measurement noise we could not merely

depend on the point prediction. To overcome this problem, prediction bounds were calculated and added to allow us to observe what percentage of the data stay within the band constructed by the lower and the upper prediction bounds.

The second proposed neural network architecture was the nonlinear autoregressive neural network (NARX) with exogenous inputs. This is a type of recurrent dynamic neural network capable of learning long-term dependencies. We were able to change the number of input and output delays fed to the input of the NARX neural network. Similar to the RNN determining the type and the number of input and output delays, number of the neurons in the hidden layer, percentage of the data to be used for training and testing the network were challenges associated with this work. NARX NNs are equipped with good features of both recurrent neural networks and dynamic neural networks. As a result, they were easier to handle and training was viable using less number of examples as compared to the RNNs. Simulation cases were investigated similar to the RNNs. They outperform RNNs regarding many aspects mentioned above. Both the prediction bounds and the error statistics were obtained for each case.

Finally, we have evaluated each neural network performance for all of the cases using the normalized Akaike information criterion (NAIC) metric. The presented metric combines the effects of the prediction error with the number of the network parameters and the size of the test data set. It was observed that the error value plays a more significant role than the number of network parameters. Thus, NARX NNs, in spite of using more parameters due to the added input delays, resulted in a better performance prediction as the error levels were lower and the predicted data (NN output) were closer to the real data. Smaller NAIC values demonstrate better prediction performance. Our conclusion is that the NARX NN performance is superior to RNNs regarding the prediction task. Finally, it is worth mentioning

that the prediction results can be used for calculation of the system time to failure (TTF) when the gas path measurements reach pre-specified thresholds. Moreover, condition-based maintenance actions can be planned using the accurate and reliable prognostic schemes that are proposed in this thesis.

6.2 Suggestions for Future Work

The problem of modelling and predicting the degradation trends in a gas turbine engine has been investigated for the first time in this thesis, to the author's best knowledge. The research presented in this thesis can provide a foundation for future research in the field of gas turbine prognostics using artificial intelligence based methods. A number of potential future work emerge as extension to the present work as it has a lot of potential for further development.

In Chapter 3, regarding the modelling and validation of the engine degradation, other sources of engine degradation such as thermal distortion or other chemical-based degradations can be modelled mathematically and added to the model in order to have a more general models which contain the effects of a wider range of possible engine degradation causes. This developed model can be used as a bench mark for data generation and prognostics study. Moreover, we considered the engine to be working in the take-off mode as the changes in the engine condition, fuel consumption level and degradation growth are more significant. This work can be extended to other modes of the flight profile such as the cruise or the landing phase.

For Chapter 4, another recommendation for further studies is pursuing the trend prediction using different types of recurrent neural networks such as the Elman or Jordan neural networks. One can verify if training the neural network and obtaining acceptable error levels is viable using a network architecture with less number of parameters. Although the turbine temperature is one of the best candidates to be

studied as it reflects the effects of degradation very well, one could monitor, study and predict other gas path measurements such as the pressure. Accurate prognostics and maintenance policies are an invaluable assets for engine systems, hence one can increase the prediction accuracy by combining two gas path measurements implying that one can train a multiple input and multiple output network as an extension. In this thesis we considered training and testing multiple input and single output neural networks. Besides, in the present work we considered the effects of each degradation on the system separately. A suggestion is to consider concurrent degradations occurring in the aircraft engine system.

The results obtained in Chapter 5 by using the NARX neural network were quite satisfactory. It is suggested to employ other types of dynamical or hybrid neural network architectures such as time-delay neural networks and see if they can outperform NARX NN. One can also address the problem of determining network parameters in a more explicit way rather than our trial and error methodology.

Uncertainty management is an important issue in an effective prognostics scheme. It is important to know not only how the uncertainties propagate but also how they could be reduced or managed. Other uncertainty management methods for determining lower and upper prediction bounds are suggested such as fuzzy auto regression to establish probabilistic boundaries. We may also want to try not limiting ourself to the noisy data that are merely normally distributed.

NAIC is a useful general metric. It is also possible to try different performance assessment metrics and error measures in our planning future work such as calculating and comparing the prediction target error, the prediction behaviour error, prediction similarity error and the Bayesian information criterion (BIC).

By integrating the methodology and the results obtained in this thesis with FDI and health management schemes, a reliable DPHM system can be developed for the

gas turbine engine. First of all, by starting with a well-trained FDI scheme, the presence of slow dynamic degradations can be detected and identified in the system. This can trigger the prognostics scheme. This is beneficial when we are working with a large data set and computational costs are important for us. The prognostics scheme will not be working at all times, and it is only triggered when there are some alarms from the FDI subsystem. Once the predicted values for certain cycles ahead reach some pre-specified or adaptive threshold levels, the health management subsystem is triggered and proper maintenance schedule and actions can be planned for the engine system.

Bibliography

[1] A. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mechanical Systems and Signal Processing*, vol. 20, no. 7, pp. 1483–1510, 2006.

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

[3] M. Roemer, C. Byington, G. Kacprzynski, and G. Vachtsevanos, "An overview of selected prognostic technologies with application to engine health management," in *ASME Turbo Expo*, 2006, pp. 707–715.

[4] S. Simani, C. Fantuzzi, and R. Patton, "Model-based fault diagnosis in dynamic systems using identification techniques," *Advances in Industrial Control*, no. one, November 2003.

[5] A. Volponi, "Foundation of gas path analysis (part i and ii)," *Gas Turbine Condition Monitoring and Fault Diagnosis*, 2003.

[6] D. Mandic and J. Chambers, *Recurrent neural networks for prediction: Learning algorithms, architectures and stability.* John Wiley & Sons, Inc., 2001.

[7] J. Menezes Jr and G. Barreto, "A new look at nonlinear time series prediction with narx recurrent neural network," in *Proceedings of the Ninth Brazilian Symposium on Neural Networks*. IEEE, 2006, pp. 160–165.

[8] (2013). [Online]. Available: http://www.wikipedia.org/

[9] E. Naderi, N. Meskin, and K. Khorasani, "Nonlinear fault diagnosis of jet engines by using a multiple model-based approach," in *Journal of Engineering for Gas Turbines and Power*. ASME, 2011, pp. 4–267.

[10] M. Naeem, R. Singh, and D. Probert, "Implications of engine deterioration for creep life," *Applied Energy*, vol. 60, no. 4, pp. 183–223, 1998.

[11] K. Mathioudakis, A. Stamatis, A. Tsalavoutas, and N. Aretakis, "Performance analysis of industrial gas turbines for engine condition monitoring," *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy*, vol. 215, no. 2, pp. 173–184, 2001.

[12] (2013). [Online]. Available: http://PlaneCrashInfo.com/

[13] B. Landsberg, "General aviation accidents 10-year trend," 2005.

[14] H. Selanduray and M. Boosroh, "Power plant optimization in a regulated environment electricity supply industry: A least cost generation approach," in *2nd International Power and Energy Conference*. IEEE, 2008, pp. 1245–1250.

[15] C. Meher-Homji and A. Bromley, "Gas turbine axial compressor fouling and washing," in *Proceedings of the 33rd Turbomachinery Symposium*, 2004, pp. 20–23.

[16] R. Kurz and K. Brun, "Gas turbine tutorialmaintenance and operating practices effects on degradation and life," in *Proceedings of the 36th Turbomachinery Symposium,*, 2007, pp. 11–13.

[17] B. Kirkwood and J. Sterne, "Essential medical statistics. 2nd," *Massachusetts: Blackwell Science Ltd*, 2003.

[18] J. Lee, J. Ni, D. Djurdjanovic, H. Qiu, and H. Liao, "Intelligent prognostics tools and e-maintenance," *Computers in Industry*, vol. 57, no. 6, pp. 476–489, 2006.

[19] A. Saxena, B. Wu, and G. Vachtsevanos, "Integrated diagnosis and prognosis architecture for fleet vehicles using dynamic case-based reasoning," in *Autotestcon,*. IEEE, 2005, pp. 96–102.

[20] J. Lee, R. Abujamra, A. Jardine, D. Lin, and D. Banjevic, "An integrated platform for diagnostics, prognostics and maintenance optimization," *Proceedings of the Intelligent Maintenance Systems*, pp. 15–27, 2004.

[21] L. Hao, Y. Jinsong, Z. Ping, and L. Xingshan, "A review on fault prognostics in integrated health management," in *9th International Conference on Electric measurment and instrumentation*. IEEE, 2009, pp. 4–267.

[22] J. Korbicz, J. Koscielny, Z. Kowalczuk, W. Cholewa, and J. Karbicz, "Fault diagnosis: Models," in *Artificial Intelligence and Applications*. Springer, Berlin, Germany,, 2004, p. 920.

[23] J. Williams, A. Davies, and P. Drake, *Condition-based maintenance and machine diagnostics*. Springer, 1994.

[24] R. Isermann, *Fault-diagnosis systems: an introduction from fault detection to fault tolerance*. Springer, 2005.

[25] ——, "Model base fault detection and diagnosis methods," in *Proceedings of the American Control Conference,*, vol. 3. IEEE, 1995, pp. 1605–1609.

[26] R. Clark, "Instrument fault detection," *IEEE Transactions on Aerospace and Electronic Systems* ,, no. 3, pp. 456–465, 1978.

[27] D. Baillie and J. Mathew, "Nonlinear model-based fault diagnosis of bearings," *NDT and E International*, vol. 30, no. 5, pp. 328–328, 1997.

[28] K. Loparo, M. Adams, W. Lin, M. Abdel-Magied, and N. Afshari, "Fault detection and diagnosis of rotating machinery," *IEEE Transactions on Industrial Electronics,*, vol. 47, no. 5, pp. 1005–1014, 2000.

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

[30] M. Nyberg, "A general framework for fault diagnosis based on statistical hypothesis testing," in *Proc. 12th Int. Workshop on Principles of Diagnosis*, 2001.

[31] V. Skormin, L. Popyack, V. Gorodetski, M. Araiza, and J. Michel, "Applications of cluster analysis in diagnostics-related problems," in *Aerospace Conference,*, vol. 3. IEEE, 1999, pp. 161–168.

[32] H. Kumamoto, K. Ikenchi, K. Inoue, and E. Henley, "Application of expert system techniques to fault diagnosis," *The Chemical Engineering Journal*, vol. 29, no. 1, pp. 1–9, 1984.

[33] Q. Li, L. Zhu, and Z. Xu, "Fuzzy petri-nets based fault diagnosis for mechanical - electric equipment," in *IEEE International Conference on Control and Automation ICCA*, june 2007, pp. 2539 –2543.

[34] R. Bayir and O. Bay, "A fault diagnosis of engine starting system via starter motors using fuzzy logic algorithm," *Gazi University Journal of Science*, vol. 24, no. 3, pp. 437–449, 2011.

[35] R. Patton, J. Chen, and C. Lopez-Toribio, "Fuzzy observers for nonlinear dynamic systems fault diagnosis," in *Proceedings of the 37th IEEE Conference on Decision and Control,*, vol. 1. IEEE, 1998, pp. 84–89.

[36] M. Hagan, H. Demuth, M. Beale *et al.*, *Neural network design*. Thomson Learning Stamford, CT, 1996.

[37] V. Venkatasubramanian and K. Chan, "A neural network methodology for process fault diagnosis," *AIChE Journal*, vol. 35, no. 12, pp. 1993–2002, 1989.

[38] K. Watanabe, I. Matsuura, M. Abe, M. Kubota, and D. Himmelblau, "Incipient fault diagnosis of chemical processes via artificial neural networks," *AIChE Journal*, vol. 35, no. 11, pp. 1803–1812, 1989.

[39] J. Hoskins, K. Kaliyur, and D. Himmelblau, "Fault diagnosis in complex chemical plants using artificial neural networks," *AIChE Journal*, vol. 37, no. 1, pp. 137–141, 1991.

[40] S. Nandi, H. Toliyat, and X. Li, "Condition monitoring and fault diagnosis of electrical motors-a review," *IEEE Transactions on Energy Conversion,*, vol. 20, no. 4, pp. 719–729, 2005.

[41] F. Filippetti, G. Franceschini, and C. Tassoni, "Neural networks aided on-line diagnostics of induction motor rotor faults," *IEEE Transactions on Industry Applications,*, vol. 31, no. 4, pp. 892–899, 1995.

[42] B. Li, M. Chow, Y. Tipsuwan, and J. Hung, "Neural-network-based motor rolling bearing fault diagnosis," *IEEE Transactions on Industrial Electronics,*, vol. 47, no. 5, pp. 1060–1069, 2000.

[43] E. Larson, D. Wipf, and B. Parker, "Gear and bearing diagnostics using neural network-based amplitude and phase demodulation," in *Proceedings of the 51st Meeting of the Society for Machinery Failure Prevention Technology, Virginia Beach, VA*, 1997, pp. 511–521.

[44] B. Samanta and K. Al-Balushi, "Artificial neural network based fault diagnostics of rolling element bearings using time-domain features," *Mechanical Systems and Signal Processing*, vol. 17, no. 2, pp. 317–328, 2003.

[45] J. Spoerre, "Application of the cascade correlation algorithm (cca) to bearing fault classification problems," *Computers in industry*, vol. 32, no. 3, pp. 295–304, 1997.

[46] T. Marcu, L. Mirea, and P. Frank, "Development of dynamic neural networks with application to observer-based fault detection and isolation," *Applied Mathematics and Computer Science*, vol. 9, pp. 547–570, 1999.

[47] D. Dong, J. Hopfield, and K. Unnikrishnan, "Neural networks for engine fault diagnostics," in *Proceedings of the VII IEEE Workshop on Neural Networks for Signal Processing*. IEEE, 1997, pp. 636–644.

[48] H. Berenji and Y. Wang, "Wavelet neural networks for fault diagnosis and prognosis," in *IEEE International Conference on Fuzzy Systems,*. IEEE, 2006, pp. 1334–1339.

[49] V. Palade, R. Patton, F. Uppal, J. Quevedo, and S. Daley, "Fault diagnosis of an industrial gas turbine using neuro-fuzzy methods," in *Proceedings of the 15th IFAC world congress*, 2002, pp. 2477–2482.

[50] D. Tobon-Mejia, K. Medjaher, and N. Zerhouni, "The iso 13381-1 standard's failure prognostics process through an example," in *Prognostics and Health Management Conference, PHM'10.* IEEE, 2010, pp. 1–12.

[51] L. Hitchcock, "Iso standards for condition monitoring," *Engineering Asset Management*, pp. 606–613, 2006.

[52] J. Luo, A. Bixby, K. Pattipati, L. Qiao, M. Kawamoto, and S. Chigusa, "An interacting multiple model approach to model-based prognostics," in *IEEE International Conference on Systems, Man and Cybernetics,*, vol. 1. IEEE, 2003, pp. 189–194.

[53] C. Byington, M. Watson, D. Edwards, and P. Stoelting, "A model-based approach to prognostics and health management for flight control actuators," in *IEEE Proceedings on Aerospace Conference,*, vol. 6. IEEE, 2004, pp. 3551–3562.

[54] D. Muench, G. Kacprzynski, A. Liberson, A. Sarlashkar, and M. Roemer, "Model and sensor fusion for prognosis, example: Kalman filtering as applied to corrosion-fatigue and fe models," *SIPS Quarterly review presentation*, 2004.

[55] A. Ray and S. Tangirala, "Stochastic modeling of fatigue crack dynamics for on-line failure prognostics," *IEEE Transactions on Control Systems Technology,*, vol. 4, no. 4, pp. 443–451, 1996.

[56] C. Oppenheimer and K. Loparo, "Physically based diagnosis and prognosis of cracked rotor shafts," in *AeroSense*. International Society for Optics and Photonics, 2002, pp. 122–132.

[57] G. Kacprzynski, A. Sarlashkar, M. Roemer, A. Hess, and B. Hardman, "Predicting remaining life by fusing the physics of failure modeling with diagnostics," *JOM Journal of the Minerals, Metals and Materials Society*, vol. 56, no. 3, pp. 29–35, 2004.

[58] J. Qiu, B. Seth, S. Liang, and C. Zhang, "Damage mechanics approach for bearing lifetime prognostics," *Mechanical systems and signal processing*, vol. 16, no. 5, pp. 817–829, 2002.

[59] G. Lesieutre, L. Fang, and U. Lee, "Hierarchical failure simulation for machinery prognostics," *Critical Link: Diagnosis to Prognosis, Haymarket*, pp. 103–110, 1997.

[60] S. Engel, B. Gilmartin, K. Bongort, and A. Hess, "Prognostics, the real issues involved with predicting life remaining," in *IEEE Proceedings on Aerospace Conference*, vol. 6. IEEE, 2000, pp. 457–469.

[61] C. Byington, P. Kalgren, B. Donovan, and A. Thompson, "Streamlined avionics phm utilizing portable information and reasoning," in *IEEE Aerospace Conference,*. IEEE, 2005, pp. 3547–3554.

[62] W. Caesarendra, G. Niu, and B.-S. Yang, "Machine condition prognosis based on sequential monte carlo method," *Expert Systems with Applications*, vol. 37, no. 3, pp. 2412–2420, 2010.

[63] P. Groer, "Analysis of time-to-failure with a weibull model," in *Proceedings of the Maintenance and Reliability Conference*, 2000, pp. 59–01.

[64] A. Schömig *et al.*, "On the suitability of the weibull distribution for the approximation of machine failures," 2003.

[65] E. Phelps, P. Willett, and T. Kirubarajan, "A statistical approach to prognostics," *Component and Systems Diagnostics, Prognosis and Health Management*, vol. 4389, pp. 23–34, 2001.

[66] G. Box, G. Jenkins, and G. Reinsel, *Time series analysis: forecasting and control*. Wiley, 2011, vol. 734.

[67] S. Pandit and S. Wu, "Time series and system analysis with application," *New York*, 1983.

[68] C. Chatfield, *Time-series forecasting*. Chapman & Hall/CRC, 2000.

[69] E. George, *Time series analysis: forecasting and control*. Holden-D., 1970.

[70] M. Priestley, "Spectral analysis and time series," 1981.

[71] S. Makridakis, A. Andersen, R. Carbone, R. Fildes, M. Hibon, R. Lewandowski, J. Newton, E. Parzen, and R. Winkler, *The forecasting accuracy of major time series methods*. Wiley Chichester et al., 1984.

[72] W. Wu, J. Hu, and J. Zhang, "Prognostics of machine health condition using an improved arima-based prediction method," in *2nd IEEE Conference on Industrial Electronics and Applications, ICIEA*. IEEE, 2007, pp. 1062–1067.

[73] G. Niu and B. Yang, "Dempster–shafer regression for multi-step-ahead time-series prediction towards data-driven machinery prognosis," *Mechanical Systems and Signal Processing*, vol. 23, no. 3, pp. 740–751, 2009.

[74] G. Li, S. J. Qin, Y. Ji, and D. Zhou, "Reconstruction based fault prognosis for continuous processes," *Control Engineering Practice*, vol. 18, no. 10, pp. 1211–1219, 2010.

[75] B. Parker Jr, T. Nigro, M. Carley, R. Barron, D. Ward, H. Poor, D. Rock, and T. DuBois, "Helicopter gearbox diagnostics and prognostics using vibration signature analysis," in *Optical Engineering and Photonics in Aerospace Sensing*. International Society for Optics and Photonics, 1993, pp. 531–542.

[76] S. Zhang and R. Ganesan, "Multivariable trend analysis using neural networks for intelligent diagnostics of rotating machinery," *Journal of Engineering for Gas Turbines and Power*, vol. 119, no. 2, pp. 378–384, 1997.

[77] R. Yam, P. Tse, L. Li, and P. Tu, "Intelligent predictive decision support system for condition-based maintenance," *The International Journal of Advanced Manufacturing Technology*, vol. 17, no. 5, pp. 383–391, 2001.

[78] Y. Dong, Y. Gu, K. Yang, and W. Zhang, "A combining condition prediction model and its application in power plant," in *Proceedings of 2004 International Conference on Machine Learning and Cybernetics,*, vol. 6. IEEE, 2004, pp. 3474–3478.

[79] P. Tse and D. Atherton, "Prediction of machine deterioration using vibration based fault trends and recurrent neural networks," *Journal of Vibration and Acoustics*, vol. 121, no. 3, pp. 355–362, 1999.

[80] Y. Shao and K. Nezu, "Prognosis of remaining bearing life using neural networks," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 214, no. 3, pp. 217–230, 2000.

[81] R. Zemouri, D. Racoceanu, and N. Zerhouni, "Recurrent radial basis function network for time-series prediction," *Engineering Applications of Artificial Intelligence*, vol. 16, no. 5, pp. 453–463, 2003.

[82] T. Girish, S. Lam, and J. Jayaram, "Reliability prediction using degradation data–a preliminary study using neural network-based approach," in *Proc. of European Safety and Reliability Conference (ESREL)*, 2003, pp. 15–18.

[83] M. Traver, R. Atkinson, and C. Atkinson, "Neural network-based diesel engine emissions prediction using in-cylinder combustion pressure," *SAE transactions*, vol. 108, no. 4, pp. 1166–1180, 1999.

[84] Y. Tan and A. Van Cauwenberghe, "Neural-network-based¡ i¿ d¡/i¿-step-ahead predictors for nonlinear systems with time delay," *Engineering Applications of Artificial Intelligence*, vol. 12, no. 1, pp. 21–35, 1999.

[85] R. Hansen, D. Hall, and S. Kurtz, "A new approach to the challenge of machinery prognostics," *Journal of Engineering for Gas Turbines and Power*, vol. 117, no. 2, pp. 320–325, 1995.

[86] W. Wang, M. Golnaraghi, and F. Ismail, "Prognosis of machine health condition using neuro-fuzzy systems," *Mechanical Systems and Signal Processing*, vol. 18, no. 4, pp. 813–831, 2004.

[87] R. Chinnam and P. Baruah, "A neuro-fuzzy approach for estimating mean residual life in condition-based maintenance systems," *International Journal of Materials and Product Technology*, vol. 20, no. 1, pp. 166–179, 2004.

[88] X. Tian, Y. Cao, and S. Chen, "Process fault prognosis using a fuzzy-adaptive unscented kalman predictor," *International Journal of Adaptive Control and Signal Processing*, vol. 25, no. 9, pp. 813–830, 2011.

[89] A. Volponi, T. Brotherton, R. Luppold, and D. Simon, "Development of an information fusion system for engine diagnostics and health management," DTIC Document, Tech. Rep., 2004.

[90] D. Doel, "Tempera gas-path analysis tool for commercial jet engines," *Journal of Engineering for Gas Turbines and Power*, vol. 116, p. 82, 1994.

[91] L. Urban, "Gas path analysis applied to turbine engine condition monitoring," in *8 th Conference of American Institute of Aeronautics and Astronautics and Society of Automotive Engineers on Joint Propulsion Specialist*, 1972, p. 1972.

[92] M. Tagina *et al.*, "A novel fault detection approach combining adaptive thresholding and fuzzy reasoning," *arXiv preprint*, 2012.

[93] B. Hansen, "Sample splitting and threshold estimation," *Econometrica*, vol. 68, no. 3, pp. 575–603, 2003.

[94] A. Volponi, H. DePold, R. Ganguli, and C. Daguang, "The use of kalman filter and neural network methodologies in gas turbine performance diagnostics: a comparative study," *Journal of Engineering for Gas Turbines and Power*, vol. 125, no. 4, pp. 917–924, 2003.

[95] P. Dewallef, C. Romessis, O. Léonard, and K. Mathioudakis, "Combining classification techniques with kalman filters for aircraft engine diagnostics," *Journal of Engineering for Gas Turbines & Power*, vol. 128, no. 2, 2006.

[96] T. Kobayashi and D. Simon, "Application of a bank of kalman filters for aircraft engine fault diagnostics," DTIC Document, Tech. Rep., 2003.

[97] N. Meskin, E. Naderi, and K. Khorasani, "Fault diagnosis of jet engines by using a multiple model-based approach." ASME, 2010.

[98] T. Kobayashi and D. Simon, "Hybrid kalman filter approach for aircraft engine in-flight diagnostics: sensor fault detection case," *ASME Paper GT2006–90870*, 2006.

[99] D. Tolani, M. Yasar, A. Ray, and V. Yang, "Anomaly detection in aircraft gas turbine engines," *Journal of Aerospace Computational Information Communication*, vol. 3, no. 2, pp. 44–51, 2006.

[100] R. Abernathy, B. Powell, D. Colbert, D. Sanders, and J. Thompson Jr, "Handbook, uncertainty in gas turbine measurements," DTIC Document, Tech. Rep., 1973.

[101] S. Gupta, A. Ray, S. Sarkar, and M. Yasar, "Fault detection and isolation in aircraft gas turbine engines. part 1: underlying concept," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 222, no. 3, pp. 307–318, 2008.

[102] Y. Lee, D. Mavris, 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, p. 041602, 2010.

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

[104] K. Mathioudakis, C. Romessis, and A. Stamatis, "Probabilistic methods for gas turbine fault diagnostics," *Gas Turbine Condition Monitoring and Fault Diagnosis,*, 2003.

[105] U. Lerner, R. Parr, D. Koller, G. Biswas *et al.*, "Bayesian fault detection and diagnosis in dynamic systems," in *Proceedings of the National Conference on*

Artificial Intelligence. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2000, pp. 531–537.

[106] D. Wang, D. Yang, J. Xu, and K. Xu, "Computational intelligence based machine fault diagnosis," in *Proceedings of The IEEE International Conference on Industrial Technology, (ICIT'96),*. IEEE, 1996, pp. 465–469.

[107] 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.

[108] E. Applebaum, "Fuzzy classification for fault isolation in gas turbine engines," in *Joint 9th IFSA World Congress and 20th NAFIPS International Conference,*, vol. 1. IEEE, 2001, pp. 292–297.

[109] C. Siu, Q. Shen, and R. Milne, "Tmdoctor: A fuzzy rule-and case-based expert system for turbomachinery diagnosis," Ph.D. dissertation, University of Edinburgh, 1996.

[110] P. Fuster, A. Ligeza, and J. Martin, "Abductive diagnostic procedure based on an and/or/not graph for expected behaviour: Application to a gas turbine," in *VTT Symposium*, vol. 171. Valotion Teknillinen Tutkimuskeskus, 1997, pp. 511–520.

[111] H. DePold and F. Gass, "The application of expert systems and neural networks to gas turbine prognostics and diagnostics," *Journal of Engineering for Gas Turbines and Power*, vol. 121, no. 4, pp. 607–612, 1999.

[112] E. Mo, M. Jie, C. Kim, and K. Lee, "Gas turbine engine fault isolation based fuzzy inference logic with ecm trend plot report," in *International Conference on Control, Automation and Systems, ICCAS'07.* IEEE, 2007, pp. 454–458.

[113] T. Brotherton and T. Johnson, "Anomaly detection for advanced military aircraft using neural networks," in *IEEE Proceedings on Aerospace Conference,*, vol. 6. IEEE, 2001, pp. 3113–3123.

[114] R. Eustace and G. Merrington, "Fault diagnosis of fleet engines using neural networks," in *12 th International Symposium on Air Breathing Engines,*. ISABE, 1995, pp. 926–936.

[115] G. Campa, M. Krishnamurty, M. Gautam, M. Napolitano, and M. Perhinschi, "A neural network based sensor validation scheme for heavy-duty diesel engines," in *14th Mediterranean Conference on Control and Automation, MED'06.* IEEE, 2006, pp. 1–6.

[116] M. Zedda and R. Singh, "Fault diagnosis of a turbofan engine using neural networks: a quantitative approach," in *The 34th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibition*, 1998.

[117] C. Romessis, A. Stamatis, and K. Mathioudakis, "A parametric investigation of the diagnostic ability of probabilistic neural networks on turbofan engines," *ASME paper*, no. 2001-GT, p. 0011, 2001.

[118] O. Illi, F. Greitzer, L. Kangas, and T. Reeve, "An artificial neural network system for diagnosing gas turbine engine fuel faults," in *48th Meeting of the Mechanical Failures Prevention Group (MFPG48),*, 1994.

[119] R. McDuff, P. Simpson, and D. Gunning, "An investigation of neural networks for f-16 fault diagnosis. i. system description," in *AUTOTESTCON'89. IEEE Automatic Testing Conference on The Systems Readiness Technology Conference. Automatic Testing in the Next Decade and the 21st Century. Conference Record.* IEEE, 1989, pp. 351–357.

[120] V. Patel, V. Kadirkamanathan, and H. Thompson, "A novel self-learning fault detection system for gas turbine engines," in *International Conference on Control'96, UKACC*, vol. 2. IET, 1996, pp. 867–872.

[121] X. WEI, C. LU, C. Wang, J. LU, and Y. LI, "Applications of support vector machines to aeroengine fault diagnosis," *Journal of Aerospace Power*, vol. 6, p. 020, 2004.

[122] N. Sinha, M. Gupta, and D. Rao, "Dynamic neural networks: An overview," in *Proceedings of IEEE International Conference on Industrial Technology*, vol. 1. IEEE, 2000, pp. 491–496.

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

[124] I. Al-Zyoud and K. Khorasani, "Detection of actuator faults using a dynamic neural network for the attitude control subsystem of a satellite," in *IEEE International Joint Conference on Neural Networks, IJCNN'05.*, vol. 3. IEEE, 2005, pp. 1746–1751.

[125] E. Tehrani, K. Khorasani, and S. Tafazoli, "Dynamic neural network-based estimator for fault diagnosis in reaction wheel actuator of satellite attitude control system," in *IEEE International Joint Conference on Neural Networks, IJCNN'05. Proceedings.*, vol. 4. IEEE, 2005, pp. 2347–2352.

[126] A. Valdes and K. Khorasani, "Dynamic neural network-based pulsed plasma thruster (ppt) fault detection and isolation for the attitude control system of a satellite," in *IEEE World Congress on Computational Intelligence, Neural Networks, IJCNN*. IEEE, 2008, pp. 2689–2695.

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

[128] M. Ayoubi, "Fault diagnosis with dynamic neural structure and application to a turbo-charger," in *Proceeding of the International Symposium Fault Detection Supervision and Safety for Technical Processes, SAFEPROCESS*, vol. 94, 1994, pp. 618–623.

[129] R. Mohammadi, E. Naderi, K. Khorasani, and S. Hashtrudi-Zad, "Fault diagnosis of gas turbine engines by using dynamic neural networks," in *IEEE International Conference on Quality and Reliability (ICQR),.* IEEE, 2011, pp. 25–30.

[130] T. Kobayashi and D. Simon, "A hybrid neural network-genetic algorithm technique for aircraft engine performance diagnostics," DTIC Document, Tech. Rep., 2001.

[131] S. Sampath and R. Singh, "An integrated fault diagnostics model using genetic algorithm and neural networks," *Journal of Engineering for Gas Turbines and Power*, vol. 128, no. 1, pp. 49–56, 2006.

[132] B. Köppen-Seliger and P. Frank, "Fuzzy logic and neural networks in fault detection," *Fusion of Neural Networks, Fuzzy Sets, and Genetic Algorithms,*, pp. 169–209, 1999.

[133] O. Dragomir, R. Gouriveau, F. Dragomir, E. Minca, N. Zerhouni *et al.*, "Review of prognostic problem in condition-based maintenance." in *European Control Conference, ECC'09.*, 2009, pp. 1585–1592.

[134] M. Enright, S. Hudak, R. McClung, and H. Millwater, "Application of probabilistic fracture mechanics to prognosis of aircraft engine components," *AIAA journal*, vol. 44, no. 2, pp. 311–316, 2006.

[135] V. Mahulkar, D. Adams, and M. Derriso, "Minimization of degradation through prognosis based control for a damaged aircraft actuator." ASME, 2009.

[136] H. Qiu, J. Lee, J. Lin, and G. Yu, "Robust performance degradation assessment methods for enhanced rolling element bearing prognostics," *Advanced Engineering Informatics*, vol. 17, no. 3, pp. 127–140, 2003.

[137] M. Orchard, B. Wu, and G. Vachtsevanos, "A particle filtering framework for failure prognosis," in *Proceedings of the World Tribology Congress*, 2005.

[138] D. Chelidze and J. Cusumano, "A dynamical systems approach to failure prognosis," *TRANSACTIONS- Journal of Vibration and Acoustics*, vol. 126, no. 1, pp. 2–8, 2004.

[139] D. Adams, "Nonlinear damage models for diagnosis and prognosis in structural dynamic systems," in *SPIE Conference Proceedings*, vol. 4733, 2002, pp. 180–191.

[140] P. Wang and G. Vachtsevanos, "Fault prognostics using dynamic wavelet neural networks," *AI EDAM*, vol. 15, no. 04, pp. 349–365, 2001.

[141] S. Hudak Jr, B. Lanning, G. Light, J. Major, M. Enright, R. McClung, and H. Millwater, "The influence of uncertainty in usage and fatigue damage sensing on turbine engine prognosis," in *Proceedings of the Minerals, Metals, and Materials Society Materials Science and Technology Symposium on Materials Damage Prognosis*, 2004.

[142] T. Khawaja, G. Vachtsevanos, and B. Wu, "Reasoning about uncertainty in prognosis: a confidence prediction neural network approach," in *Annual Meeting of the North American Fuzzy Information Processing Society, NAFIPS.* IEEE, 2005, pp. 7–12.

[143] G. Parthasarathy, S. Menon, K. Richardson, A. Jameel, D. McNamee, T. Desper, M. Gorelik, and C. Hickenbottom, "Neural network models for usage based remaining life computation," *Journal of Engineering for Gas Turbines and Power*, vol. 130, no. 1, 2008.

[144] K. Xu, M. Xie, L. Tang, and S. Ho, "Application of neural networks in forecasting engine systems reliability," *Applied Soft Computing*, vol. 2, no. 4, pp. 255–268, 2003.

[145] M. Cottrell, P. Gaubert, C. Eloy, D. François, G. Hallaux, J. Lacaille, and M. Verleysen, "Fault prediction in aircraft engines using self-organizing maps," in *Advances in Self-Organizing Maps.* Springer, 2009, pp. 37–44.

[146] L. MARINAI, R. SINGH, B. CURNOCK, and D. PROBERT, "Detection and prediction of the performance deterioration of a turbofan engine," in *Proceedings of the International Gas Turbine Congress*, 2003.

[147] R. Sekhon, H. Bassily, and J. Wagner, "A comparison of two trending strategies for gas turbine performance prediction," *Journal of Engineering for Gas Turbines and Power*, vol. 130, no. 4, 2008.

[148] P. Shetty, D. Mylaraswamy, and T. Ekambaram, "A hybrid prognostic model formulation system identification and health estimation of auxiliary power units," in *IEEE Aerospace Conference,.* IEEE, 2008, pp. 10–pp.

[149] G. Kumar, *Artificial Neural Nework*, 2007, vol. 2, no. 1.

[150] T. Chow and S. Cho, *Neural Networks and Computing: Learning Algorithms and Applications*. World Scientific Publishing Company Incorporated, 2007, vol. 7.

[151] Y. Kim, F. Lewis, and C. Abdallah, "A dynamic recurrent neural-network-based adaptive observer for a class of nonlinear systems," *Automatica*, vol. 33, no. 8, pp. 1539–1543, 1997.

[152] T. Chow and Y. Fang, "A recurrent neural-network-based real-time learning control strategy applying to nonlinear systems with unknown dynamics," *IEEE Transactions on Industrial Electronics,*, vol. 45, no. 1, pp. 151–161, 1998.

[153] A. Tsoi and A. Back, "Discrete time recurrent neural network architectures: A unifying review," *Neurocomputing*, vol. 15, no. 3, pp. 183–223, 1997.

[154] W. Duch and N. Jankowski, "Survey of neural transfer functions," *Neural Computing Surveys*, vol. 2, no. 1, pp. 163–212, 1999.

[155] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems (MCSS)*, vol. 2, no. 4, pp. 303–314, 1989.

[156] H. Simon, *Neural networks: a comprehensive foundation*. Prentice Hall, 1999.

[157] R. Williams and D. Zipser, "Gradient-based learning algorithms for recurrent networks and their computational complexity," *Back-propagation: Theory, architectures and applications*, pp. 433–486, 1995.

[158] V. Kodogiannis, P. Lisboa, and J. Lucas, "Neural network modelling and control for underwater vehicles," *Artificial Intelligence in Engineering*, vol. 10, no. 3, pp. 203–212, 1996.

[159] H. Siegelmann, B. Horne, and C. Giles, "Computational capabilities of recurrent narx neural networks."

[160] R. Al Seyab and Y. Cao, "Differential recurrent neural network based predictive control," *Computers & Chemical Engineering*, vol. 32, no. 7, pp. 1533–1545, 2008.

[161] L. Gang and F. Yu, "A hybrid nonlinear autoregressive neural network for permanent-magnet linear synchronous motor identification," in *Proceedings of the Eighth International Conference on Electrical Machines and Systems, ICEMS*, vol. 1. IEEE, 2005, pp. 310–314.

[162] T. Lin, B. Horne, P. Tino, and C. Giles, "Learning long-term dependencies in narx recurrent neural networks," *IEEE Transactions on Neural Networks,*, vol. 7, no. 6, pp. 1329–1338, 1996.

[163] M. Grossglauser and J. Bolot, "On the relevance of long-range dependence in network traffic," *IEEE/ACM Transactions on Networking (TON)*, vol. 7, no. 5, pp. 629–640, 1999.

[164] U. Filik and M. Kurban, "A new approach for the short-term load forecasting with autoregressive and artificial neural network models," *International Journal of Computational Intelligence Research*, vol. 3, no. 1, pp. 66–71, 2007.

[165] D. Rumelhart, G. Hintont, and R. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[166] B. Widrow and M. Lehr, "30 years of adaptive neural networks: Perceptron, madaline, and backpropagation," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1415–1442, 1990.

[167] J. RONALD and D. Zipser, "Experimental analysis of the real-time recurrent learning algorithm," *Connection Science*, vol. 1, no. 1, pp. 87–111, 1989.

[168] D. Nguyen and B. Widrow, "Neural networks for self-learning control systems," *IEEE Control Systems Magazine,*, vol. 10, no. 3, pp. 18–23, 1990.

[169] R. Williams and J. Peng, "An efficient gradient-based algorithm for on-line training of recurrent network trajectories," *Neural Computation*, vol. 2, no. 4, pp. 490–501, 1990.

[170] L. Pape, B. Ruessink, M. Wiering, and I. Turner, "Recurrent neural network modeling of nearshore sandbar behavior," *Neural Networks*, vol. 20, no. 4, pp. 509–518, 2007.

[171] G. Plett, "Adaptive inverse control of linear and nonlinear systems using dynamic neural networks," *IEEE Transactions on Neural Networks,*, vol. 14, no. 2, pp. 360–376, 2003.

[172] I. Treager, *Aircraft gas turbine engine technology.* Glencoe, 1996.

[173] H. Saravanamuttoo, G. Rogers, and H. Cohen, *Gas turbine theory.* Pearson Education, 2001.

[174] S. Camporeale, B. Fortunato, and M. Mastrovito, "A modular code for real time dynamic simulation of gas turbines in simulink," *Journal of Engineering for Gas Turbines and Power*, vol. 128, no. 3, pp. 506–517, 2006.

[175] W. Visser and M. Broonhead, "Gsp, a generic object-oriented gas turbine simulation environment," *ASME Turbo Expo*, 2000.

[176] W. Visser, O. Kogenhop, and M. Oostveen, "A generic approach for gas turbine adaptive modeling," *Proceedings of ASME Turbo Expo.*, 2004.

[177] G. Aker and H. Saravanamuttoo, "Predicting gas turbine performance degradation due to compressor fouling using computer simulation techniques," *Journal of Engineering for Gas Turbines and Power*, vol. 111, pp. 343–350, 1989.

[178] D. Karanjia and H. Saravanamuttoo, "Cost effective engine health monitoring system for on-board use on hovercraft," Tech. Rep., 1980.

[179] N. Daroogheh, A. Vatani, M. Gholamhossein, and K. Khorasani, "Engine life evaluation based on a probabilistic approach," in *Proceedings of the 2012 ASME congress (IMECE2012)*.

[180] M. Naeem, "Fuel usage and its effects on operational performance for ge-f404/f-18," Ph.D. dissertation, M. Sc. thesis, Cranfield University, UK, 1996.

[181] C. Meher-Homji, M. Chaker, and H. Motiwalla, "Gas turbine performance deterioration," in *Proceedings of the 30th Turbomachinery Symposium*, 2001, pp. 17–20.

[182] J. Petek and P. Hamilton, "Performance monitoring of gas turbines," *Journal of Orbit*, vol. 25, no. 1, pp. 64–74, 2005.

[183] H. Ben Hariz, "The optimisation of the usage of gas turbine generation sets for oil and gas production using genetic algorithms," Ph.D. dissertation, Cranfield University, 2010.

[184] T. Song, J. Sohn, K. Tong Seop, K. Jae Hwan, and S. Ro, "An improved analytic model to predict fouling phenomena in the axial compressor of gas turbine engines," in *Proceedings of the International Gas Turbine Congress,*, 2003.

[185] M. Naeem, R. Singh, and D. Probert, "Implications of engine deterioration for a high-pressure turbine-blade's low-cycle fatigue (lcf) life-consumption," *International Journal of Fatigue*, vol. 21, no. 8, pp. 831 – 847, 1999.

[186] M. Naeem, "Implications of turbine erosion for an aero-engine's high-pressure-turbine blade's low-cycle-fatigue life-consumption," *Journal of Engineering for Gas Turbines and Power*, vol. 131, no. 5, 2009.

[187] ——, "Implications of day temperature variation for an aero-engine's hp turbine-blade's creep life-consumption," *Aerospace Science and Technology*, vol. 13, no. 1, pp. 27–35, 2009.

[188] J. T. Connor, R. D. Martin, and L. Atlas, "Recurrent neural networks and robust time series prediction."

[189] S. D. Strudwick and R. Ferryman, "Trend analysis of remote standby generating sets using parameter-estimation neural networks," *Techniques and Applications of Neural Networks*, p. 117, 1993.

[190] P. Tse and D. Wang, "A hybrid neural networks based machine condition forecaster and classifier by using multiple vibration parameters," in *IEEE International Conference on Neural Networks,*, vol. 4. IEEE, 1996, pp. 2096–2100.

[191] A. S. Weigend and N. A. Gershenfeld, "Results of the time series prediction competition at the santa fe institute," in *IEEE International Conference on Neural Networks,*. IEEE, 1993, pp. 1786–1793.

[192] J.-M. Kuo and B. de Vries, "Prediction of chaotic time series using recurrent neural networks," in *Proceedings of the IEEE-SP Workshop on Neural Networks for Signal Processing II.,*. IEEE, 1992, pp. 436–443.

[193] S. A. Vander Wiel and W. Q. Meeker, "Accuracy of approx confidence bounds using censored weibull regression data from accelerated life tests," *IEEE Transactions on Reliability,*, vol. 39, no. 3, pp. 346–351, 1990.

[194] W. Härdle, *Nonparametric and semiparametric models*. Springer Verlag, 2004.

[195] S. Geisser, *Predictive inference: an introduction*. CRC Press, 1993, vol. 55.

[196] D. C. Montgomery and G. C. Runger, *Applied statistics and probability for engineers*. Wiley, 2010.

[197] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[198] B. G. Horne and C. L. Giles, "An experimental comparison of recurrent neural networks," *Advances in Neural Information Processing Systems*, pp. 697–704, 1995.

[199] M. Ardalani-Farsa and S. Zolfaghari, "Chaotic time series prediction with residual analysis method using hybrid elman–narx neural networks," *Neurocomputing*, vol. 73, no. 13, pp. 2540–2553, 2010.

[200] J. M. P. Menezes Jr and G. A. Barreto, "Long-term time series prediction with the narx network: An empirical evaluation," *Neurocomputing*, vol. 71, no. 16, pp. 3335–3343, 2008.

[201] D. Posada and T. R. Buckley, "Model selection and model averaging in phylogenetics: advantages of akaike information criterion and bayesian approaches over likelihood ratio tests," *Systematic Biology*, vol. 53, no. 5, pp. 793–808, 2004.

[202] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control,*, vol. 19, no. 6, pp. 716–723, 1974.

[203] ——, "Information theory and an extension of the maximum likelihood principle," in *Selected Papers of Hirotugu Akaike*. Springer, 1998, pp. 199–213.

[204] I. A. Kieseppä, "Akaike information criterion, curve-fitting, and the philosophical problem of simplicity," *The British Journal for the Philosophy of Science*, vol. 48, no. 1, pp. 21–48, 1997.

[205] Y. Sakamoto, M. Ishiguro, and G. Kitagawa, *AKAIKE information criterion statistics*. KTK scientific publishers, 1986.