CRANFIELD UNIVERSITY

LIHUI WANG

ROUGH SET BASED GAS TURBINE FAULT ISOLATION STUDY

SCHOOL OF ENGINEERING
GAS TURBINE ENGINEERING

PHD
ACADEMIC YEAR: 2007 - 2010

SUPERVISOR: YI-GUANG LI
07/2010

CRANFIELD UNIVERSITY

SCHOOL OF ENGINEERING
GAS TURBINE ENGINEERING

PHD

ACADEMIC YEAR 2007 - 2010

LIHUI WANG

ROUGH SET BASED GAS TURBINE FAULT ISOLATION STUDY

SUPERVISOR: YI-GUANG LI

07/2010

THIS THESIS IS SUBMITTED IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

ABSTRACT

Gas path fault isolation is one of the key techniques in Engine Health Management systems. In order to accomplish gas path fault isolation successfully for a gas turbine engine, both an accurate off-design performance model and an effective fault isolation approach are necessary. In this thesis, two original and useful contributions to knowledge are presented: a new gas turbine off-design performance model adaptation approach and a new gas turbine fault isolation approach.

This new adaptation approach uses optimal multiple scaling factors obtained by using a Genetic Algorithm to scale inaccurate component characteristic maps in gas turbine performance models to improve their prediction accuracy in different off-design conditions. The major feature of this approach is that it provides non-linear map scaling and therefore is able to provide more effective adaptation. The new fault isolation approach can be used to discover knowledge hidden in engine fault samples, transfers that knowledge into rules, and then uses those rules for fault isolation. In addition, it is also capable of selecting appropriate measurements for fault isolation, dealing with uncertainty caused by measurement noise. Enhanced fault signatures, which are represented by the measurement deviations and their ranking pattern in terms of magnitude, are developed to make gas turbine faults easier to distinguish and hence make this fault isolation approach more effective.

The new adaptation approach was applied to the off-design performance model adaptation of a gas turbine, while the new fault isolation approach was employed for fault isolation in a gas turbine. The results show that the new adaptation approach is very effective in improving the prediction accuracy of off-design performance models and the new fault isolation approach is not only effective in fault isolation but also in selecting measurements for isolation and generating fault isolation rules.

Acknowledgements

I would like to thank various people who have helped, supported, and encouraged me in the past three years.

I am very grateful to my supervisor Dr Yi-Guang Li, who guided me through the past three years and provided me with valuable suggestions.

I would also like to thank Professor Riti Singh, Dr Stephen Ogaji, Dr Kenneth Ramsden and Dr Vassilios Pachidis from the department of Power and Propulsion of Cranfield University for their help during my studies at Cranfield.

In addition, I want to express my gratitude to Aviation Industry Corporation of China (AVIC) for funding this work, and I also would like to take this opportunity to thank Mr Xing Feng, Mr Kaiming Huang and Mr Wei Zhang from AVIC for their supports to this research.

Special thanks go to my dear friends, Mohammad Fahmi B Abdul Ghafir, Uyioghosa Igie, Daniel Kamunge, Jizhao Li, Dr Georgios Doulgeris, and Dongman Hong. Thanks for making my life in UK wonderful.

Finally I'd like to express my gratitude to my family – my parents, my wife and my brother for supporting and encouraging me. Especially to my father, the greatest man in my heart: thank you for all the sacrifices you have made for me and always believing in me; this work is specifically dedicated to you.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

a	=	Object
A	=	Off-design point
*A**	=	Modified off-design point
At	=	Attribute set
$At\_(W)$	=	Lower approximation of object set *W* with respect to attribute set *At*
b	=	Object
B	=	Off-design point
*B**	=	Modified off-design point
c	=	Condition attribute
C	=	Condition attribute set
D	=	Decision attribute
d	=	Discrete value
e	=	Error between predicted measurement and actual measurements
$\hat{e}$	=	Error between predicted and actual measurement vectors
E	=	State error covariance
$Fault_h$	=	h^{th} type of fault
f	=	Attribute
F	=	Subset of attribute set *At*
*F**	=	Subset of attribute set *F*
$F\_(W)$	=	Lower approximation of object set *W* with respect to attribute set *F*

$F^{-}(W)$	=	Upper approximation of object set *W* with respect to attribute set *F*
$\lvert F\_(W)\rvert$	=	Cardinality of $F\_(W)$
$\lvert F^{-}(W)\rvert$	=	Cardinality of $F^{-}(W)$
H	=	Influence coefficient matrix
H^{-1}	=	Fault coefficient matrix
I	=	Total number of different measurement values
i	=	Total number of measurements
J	=	Cost function
K	=	Kalman gain
k	=	Time index
L	=	Total number of measurements
l	=	Total number of health parameters
M	=	Total number of fault samples
m	=	Cut point
No	=	Indicator to indicate that two objects cannot be distinguished
N	=	Number of measurements
N_2	=	Rotational speed of high pressure compressor
O	=	Total number of different faults
ΔP_k	=	k^{th} measurement deviation
P	=	Measurement
P_x	=	Measurement deviation
P_y	=	Measurement deviation
Q	=	Process covariance

R	=	Covariance of measurement noise
$R(P_xP_y)$	=	Ranking-parameter-difference between two measurement deviations
S_i	=	i^{th} Fault sample
SR_{half}	=	Success rates under half assumed ranges
SR_{double}	=	Success rates under double assumed ranges
Support	=	Total number of fault samples matching with a rule
T	=	Decision table
U	=	Universe of objects
u	=	State control vector
V_1	=	State transition model
V_2	=	Control-input model which is applied to the state control vector
v	=	Attribute value
w	=	Object
$[w]_F$	=	Equivalent class of an object w with respect to an attribute f
$[w]_F$	=	Equivalent class of an object w with respect to attribute set F
W	=	Object set
W^C	=	Object groups classified by the condition attributes
W^D	=	Object groups classified by the decision attribute
We	=	Weight of neuron
$\hat{x}$	=	Health parameter vector

y	=	An object
Yes	=	Indicator to indicate that two objects can be distinguished
$\hat{z}$	=	Measurement vector
z_k	=	Measurement at time k
Superscript		
-	=	Denotes the predicted state
-1	=	Denotes the matrix inverse
T	=	Denotes transpose of the vector or matrix
Subscript		
A	=	Off-design point index
*A**	=	Modified off-design point index
E	=	Efficiency
FC	=	Flow capacity
O	=	Optimal
PR	=	Pressure ratio
Cout	=	Compressor outlet
h	=	An integer
i	=	An integer
j	=	An integer
k	=	Time
Greek letter		
$\alpha_F(W)$	=	Accuracy of approximation of object set *W* with respect to attribute set *F*
$\hat{\varepsilon}$	=	Measurement noise vector

$\hat{\mu}$	=	Sensor bias vector

Symbol

Δ	=	Deviation

ABBREVIATIONS

AI	=	Artificial intelligent
BBN	=	Bayesian-belief network
BIP	=	Burner inlet pressure
CN	=	Corrected non-dimensional rotational speed
Comp	=	Compressor
DH	=	Enthalpy drop ratio
DI	=	Degradation index
DP	=	Design point
EA	=	Exponential average
ES	=	Expert system
ETA	=	Isentropic efficiency
FCM	=	Fault coefficient matrix
FF	=	Fuel flow rate
FW	=	Framework
GA	=	Genetic algorithm
GPA	=	Gas path analysis
HP	=	High pressure
HPC	=	High pressure compressor
HPT	=	High pressure turbine
ICM	=	Influence coefficient matrix
IP	=	Intermediate pressure
IPC	=	Intermediate pressure compressor

IS	=	Information system
ISA	=	International standard atmosphere
KF	=	Kalman filter
LP	=	Low pressure
LPC	=	Low pressure compressor
LPT	=	Low pressure turbine
NN	=	Neural network
OD	=	Off-design
OF	=	Objective function
PR	=	Pressure ratio
RA	=	Rolling average
RESE	=	Rough set exploration system
SF	=	Scaling factor
Turb	=	Turbine
WAC	=	Corrected non-dimensional mass flow rate
WLS	=	Weighted-Least-Squares

1 Introduction

1.1 Gas turbine maintenance

With the increasing application of gas turbines in civil, military and industry fields, gas turbine maintenance becomes more important as it is able to improve the availability, safety and economics of gas turbines. Therefore much effort has been put into developing an effective maintenance technique. To date, maintenance techniques have been developed for three generations: these are listed as follows:

1. First generation: run to failure maintenance which only carries out maintenance when gas turbines are broken.
2. Second generation: preventive maintenance which executes scheduled overhauls in order to prevent component failure.
3. Third generation: predictive maintenance which performs maintenance only when it is necessary.

The first maintenance method is simple and less costly than the others; however it is risky and not applicable at the present time as it may lead to disasters if failures occur during flight. Thus it can neither ensure the reliability nor the safety of gas turbine operation. The second method is able to offer more insurance in preventing failures. However it still has several shortcomings: firstly, it is very costly since some overhauls may be unnecessary; secondly unexpected failures may occur between scheduled overhauls, thirdly, the downtime caused by overhauls leads to profit losses. As a result of this, the third method was introduced to overcome the shortcomings that the first and second methods have and maximally ensure availability, safety and good economics. This can be implemented by monitoring the health of gas turbines detect and predict faults, and hence appropriate maintenance can be scheduled and carried out. The system used to monitor and analyse the health of the gas turbine and optimise the gas turbine's operation is usually referred to as the Engine Health Management (EHM) system.

1.2 Engine health monitoring

An EHM system is usually used to deal with the monitoring, detection, isolation, predictive trending and accommodation of engine degradation, faults and failures [1]. It determines a state of awareness of the engine using tools such as data fusion physics based models, data driven models and feature extraction techniques, which determine accurate fault isolation for more efficient maintenance, predict future faults and deteriorating equipment health, and propose maintenance actions to obviate on-board equipment malfunction.

The main purposes for employing an EHM system are to keep the gas turbine operating reliably, minimise total cost of gas turbine operation, asset availability and mission readiness while maintaining, and even increasing, safety [1]. These above-mentioned purposes are shared by different EHM systems for aero-derivative, industrial and military engines; however, different EHM systems usually have different emphases. In the commercial sector, it is strongly desired to keep the total cost of engine operation as low as possible; this can be achieved by using EHM systems to reduce the possibility of in-flight shutdowns, unplanned engine removals, and cancellations and delays through planned maintenance. For military engine users, the prime causes of the implementation of EHM systems are to improve the availability of engines in order to reduce aborted missions, and provide real-time detection, prediction and evaluation of faults which may lead to critical failures. In the industrial sector, the main consideration is the down time of gas turbines, which may result in direct economic losses.

The development of EHM systems is almost as old as that of the gas turbine engine. Thirty years ago, the term Engine Health Monitoring was used rather than Engine Health Management. The difference between them is that the former refers to passive observations whereas the latter is an active pursuit which is more robust and advanced [1]. To date, two types of EHM systems have been developed; ground-based and engine-hosted. The former is usually

used for aero-derivative and industrial gas turbines while the latter is more suitable to be introduced in the military sector.

An EHM system usually employs several different techniques, as no single technique is able to satisfy all the requirements for detection, isolation, prediction and accommodation of all engine faults and failures. Some common EHM techniques are listed below [2]:

Gas path analysis (GPA)

Gas turbine component deterioration induces a change in configuration of the components. Such a change leads to performance degradation of the gas turbine, and simultaneously gas path measurement deviations which can be considered to be gas path fault signatures. The aim of GPA is to detect and isolate the faulty component(s), and quantify deterioration severity based on the fault signatures.

Oil system monitoring

a. Oil condition monitoring: a change in operating conditions of a gas turbine caused by deterioration can cause a change in aeration, temperature, oil consumption, and consequently induce a variation in the chemical properties of the oil used for combustion in the gas turbine. Such a variation can be utilised for analysis of the deterioration.

b. Oil debris monitoring: some of the rotary components of gas turbines, such as bearings and gears, need to be lubricated in order to reduce their temperatures, and the friction between them and the components connected to them. The deterioration of these oiled components, which leads to their size change, may result in a gradual increase of usage of the oil used for lubrication. Such a variation in oil usage can be considered as a signature of component deterioration and used for deterioration analysis.

Vibration monitoring

The rotation of components of gas turbines usually induces the components to vibrate at certain frequencies, unless there is deterioration or change of operating conditions; hence, monitoring the vibration of both the components and the overall vibration can provide useful information to detect many faults such as compressor or turbine blade fracture, misaligned coupling, bearing defect, malfunction of gears and so on. These faults usually cannot be detected easily by other health monitoring techniques.

Engine usage monitoring

This technique is used for estimating component life and providing very useful information to users of gas turbines about when they need to replace components or order new ones, and consequently this improves the reliability of the gas turbines, reduces their maintenance costs and off line maintenance periods. Normally, the life of gas turbine components is determined by analysing creep and low cycle fatigue.

Visual condition monitoring

Visual condition monitoring is a very simple, accurate and effective technique, since many component faults are visible, and do not require any sensors, hence avoiding the positive influence resulting from measurement noise or sensor bias. Gas turbine internal inspection can be implemented by endoscopes.

Turbine exit spread monitoring

This technology is usually used to detect faults taking place in combustors, such as blocking and carking of the combustion chamber, a blocked burner, and asymmetric distribution of fuel in the combustors.

Transient monitoring

During some transient operating conditions (i.e. start up and shut down) of gas turbines, some transient condition parameters, such as maximum gas temperature, time required to light, and time needed for the start up or shut

down, can be utilised to analyse fuel system, starting system, or combustion system faults.

Acoustic monitoring

Rotary components always produce acoustic emissions, and any physical change of components caused by deterioration can lead to a change of level in their acoustic emissions which can be considered to be fault signatures. Hence the acoustic fault signatures can be used to detect component faults.

Some thought needs to be given to the employment of health monitoring techniques, such as the choice of which components need to be monitored, and the appropriate selection of health monitoring techniques. These considerations could compromise the performance of cost of the EHM system. If an EHM system is too complex, it is possible that it may actually cause an increase of maintenance costs. In contrast, if it is too simple, the system may only be capable of dealing with a limited number of gas turbine faults and failures.

Among the above-mentioned techniques, GPA is one of the most powerful as it is capable of dealing with many component faults effectively. It is included in most EHM systems and thus plays an important role.

1.3 Gas path diagnostics

An effective GPA method is a key to building an EHM system because of the important role this method plays in the system. Fault detection is relatively easier than fault isolation and quantification as it can be done by monitoring anomalies in the trend of measurement deviations. Usually fault isolation needs to be carried out prior to fault quantification to identify faulty component(s) (some GPA methods execute both of isolation and quantification simultaneously); hence successful fault isolation is essential to GPA. No matter how successful the fault quantification is, it can be misleading if a wrong fault isolation solution is provided.

In addition, the gas turbine performance model, which is able to simulate thermodynamic behaviour of clean and degraded gas turbines (gas turbines with component faults at different operating conditions, has a significant influence on the quality of gas path diagnostics. The gas turbine performance model is required by almost every existing method. Some of the methods use the model directly in the diagnostic process while others need it to generate essential gas turbine deterioration data, and then analyse such data to obtain useful information for diagnostic purposes.

Therefore, the fault isolation method and the performance model are two of the most important aspects that need to be considered in gas path diagnostics.

1.4 Gas path analysis based fault isolation

The prime driver of the implementation of gas path analysis based fault isolation is to identify faulty components; however, there are also other important problems that may need to be dealt with. The relevant problems that a gas path diagnostic method may face are listed as follows:

- *Fault isolation*: to identify the faulty component.
- *Measurement noise and sensor bias*: GPA is to implement diagnostics based on measurement data, and measurement data usually have uncertainty due to measurement noise, sensor bias or both. Such an uncertainty can be reduced using measurement noise reduction approaches; however, they cannot be eliminated completely.
- *Limitation of measurement number*: in practice, the number of available measurements for diagnostics is always limited due to high cost and installation difficulties.
- *Measurement selection*: selecting appropriate set of measurements for fault isolation is important since it has great impact on the quality of diagnostics.

- *Complexity*: the high complexity of the diagnostic method can lead to higher costs.
- *Computational speed*: in order to make a diagnostic approach suitable for on-wing applications, the computational speed is one of the key factors needed to be considered, especially for aircrafts not equipped with high performance computers.
- *Capability to provide expert knowledge*: users of diagnostic systems can benefit from the expert knowledge to understand and solve diagnostics problems.

An effective gas path based fault isolation method should be able to solve as many as possible of the problems listed above.

1.5 Research objectives and strategies

The research objective of this study is to develop two approaches for achieving good fault isolation: the first approach is to ensure the prediction accuracy of the engine performance model; the second approach is able to accurately isolate component faults and also solve most the fault isolation problems listed in the previous section.

The strategy to achieve the research objective contains the following steps:

1. To define the requirements for an effective gas turbine off-design performance model adaptation and a gas path analysis based fault isolation method.
2. To carry out a literature review of existing gas path based fault isolation and gas turbine performance model adaptation techniques, and obtain a good understanding of their mechanisms, advantages and shortcomings.
3. To construct a novel gas turbine off-design performance model adaptation method and a gas path based fault isolation method

4. To test the novel methods by applying them to performance model adaptation and fault isolation of gas turbines respectively.

5. To analyse the results and check if the requirements have been met or not, if not, improve the methods and go back to step 4.

6. To evaluate the potential of the methods and make recommendations for future studies.

1.6 A guide to this thesis

Section 1 introduces gas turbine maintenance, Engine Health Management systems, gas path diagnostics, and the importance of gas turbine performance model adaptation and gas turbine fault isolation techniques, and also states the reasons why this research needed to carried out and the main research objective.

Section 2 focuses on different aspects: one is a literature review of existing gas path diagnostic techniques, including a study of their advantages and shortcomings, and the other is a literature review of gas turbine performance model adaptation approaches. This section also contains summaries for gas path diagnostics and gas turbine performance model adaptation.

A new gas turbine performance model adaptation method developed by the author is presented in Section 3. In this section, the methodology, application, verification and evaluation of this approach are described.

In Section 4, three different versions of a new rough set based fault isolation introduced by the author and different fault isolation frameworks integrating with the developed approach are described. The application, verification and analysis of the methods and frameworks are also presented in this section. Conclusions for the developed gas turbine performance model adaptation and fault isolation methods are made in Section 5 which states the pros and cons of these methods and also the contribution to knowledge made by the author in this research.

In Section 5, several conclusions are made on the basis of the work presented in this thesis.

The last section, and 6, some recommendations for future works, which, may be worth carrying out to further improve the developed methods or integrate the methods with other techniques to form more robust gas path diagnostic techniques, are presented.

2 State of the art

2.1 Gas path diagnostics

2.1.1 Introduction

When a gas turbine is in service, some of their components need to work under high pressure, high temperature and high stress conditions in long periods. This may result in component fouling, erosion, corrosion, rubbing wear, labyrinth deal damage, and hot end component damage, and consequently change of size of the components. Such a change will lead to performance degradation of the gas turbine, and simultaneously measurement deviations which can be considered to be fault signatures. The aim of GPA is to detect, and isolate the faulty component(s), and quantify degradation level of the faulty component(s) based on the fault signatures. The general process of GPA based gas turbine diagnostics is illustrated in Figure 2-1.

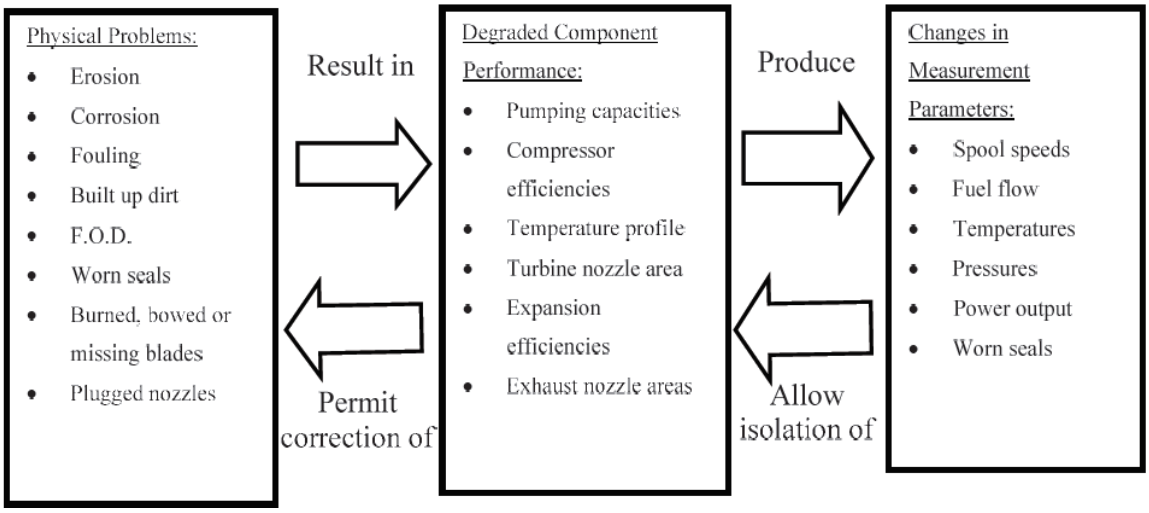

Figure 2-1 Gas turbine diagnostics based on GPA [3]

2.1.2 Component faults

There are many different sorts of gas turbine component faults. Among them, the following faults [2] [4] are major factors causing degradation in performance of gas turbine components and cause gas path fault signatures (gas path measurement deviations): fouling, erosion, corrosion, rubbing wear, labyrinth

seal damage, and foreign object damage. Therefore the scope of the faults that gas path diagnostics is able to effectively deal with is limited to the above-mentioned faults. Each of the faults is described in this section.

Fouling

When gas turbines are in service, large amount of air needs to pass thought their cores, and particles in the air are likely to adhere to their components. In addition, remains of combustion resulted from impure fuels may adhere to turbines. Both the phenomena are usually referred to as fouling which leads to degradation of flow capacity and efficiency of the components. Compressor fouling is the most common performance deterioration although fouling may take place in both the compressor and the turbine. Compressor fouling usually can be reduced by washing; however washing is not applicable for turbine fouling due to the extremely high temperature of and the location of turbines, which makes turbine fouling cleaning more difficult.

Erosion

Hard particles in the fluid passing through gas turbines can result in erosion on surfaces of some components. The erosion process can be speed up if gas turbines are operating in polluted atmospheres with solid particles. Erosion always leads to decrease in flow capacity and efficiency for compressors, and decrease in efficiency but increase in flow capacity for turbines.

Corrosion

Like erosion, corrosion also causes material loss of components, but it normally caused by chemical reaction between component materials and some elements or impurities in the fluid passing through the components. Compressor corrosion can by caused by salt, mineral acids and reactive gases, while more serious corrosion resulted from elements and high temperature can occur on turbines. The corrosion causes similar variation in health parameters of component to erosion does. Corrosion prevention can be carried out by coating of blades of turbines and compressors.

Rubbing wear

Several factors, such as relative thermal growth between the static and rotating members, centrifugal growth of the rotating member, axial movement of rotating pars, distortion of casing relative to the rotors, and engine operating procedure, can induce contact between static parts and rotating parts with very high speed, and subsequently can result in material loss of rotor blade tips or knife edge seal. In order to reduce the influence of rubbing wear, abradable surfaces are employed by many gas turbines.

Labyrinth seal damage

Labyrinth seal damage doesn't cause material losses on compressors or turbines; however it causes leakages on cavities connecting to compressors or turbines. Such leakages usually have larger influences on the efficiency of components affected by the leakages than on the flow capacity.

Foreign object damage

The faults mentioned earlier in this section, whose severity increases as time goes on, induce gradual deteriorations. Larger or hard objects such as birds sucked into gas turbines can cause serious damages to the gas turbines, and the deteriorations caused by the damages are instant. Foreign object damage usually causes the requirement to replace the damages components with new ones since it is unlikely to recover the damage caused by it. It can result in sudden reduction of efficiency of the damaged components or mass flow capacity of them.

2.1.3 Component characteristics

The component faults mentioned in the previous section usually cause variations of the performance characteristics of the faulty components which can be compressors, burner, and turbines. The characteristics of the components are usually represented by their corresponding characteristic parameters such as corrected mass flow rate (*WAC*) (i.e. flow capacity),

pressure ratio (*PR*), isentropic efficiency (*ETA*), enthalpy drop ratio (*DH*), burner inlet pressure (*BIP*), and burner temperature rise (ΔT). The characteristics of components varies with different operating conditions; hence component characteristic maps are usually used to describe the variation of the characteristics caused by the change of operating conditions. Examples of compressor, burner and turbine characteristic maps are shown in Figure 2-2, Figure 2-4 and Figure 2-3 where CN is corrected rotational speed.

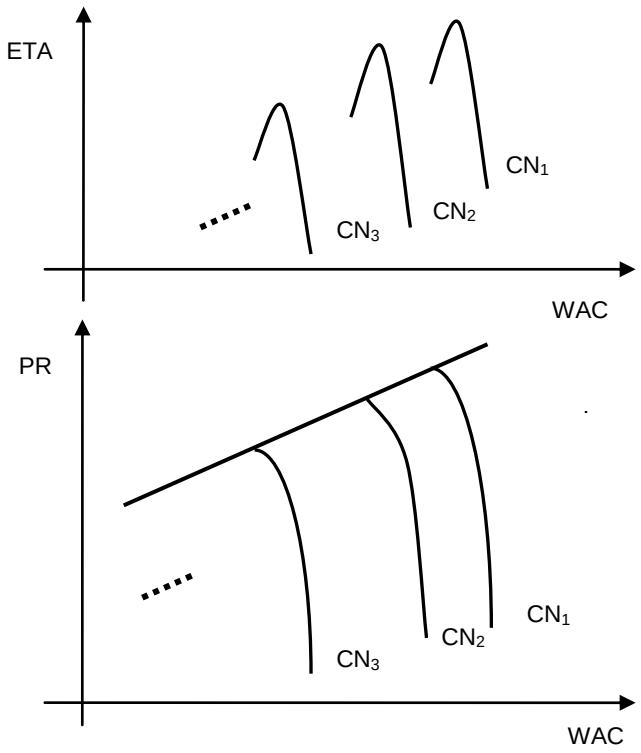

Figure 2-2 Compressor characteristic map

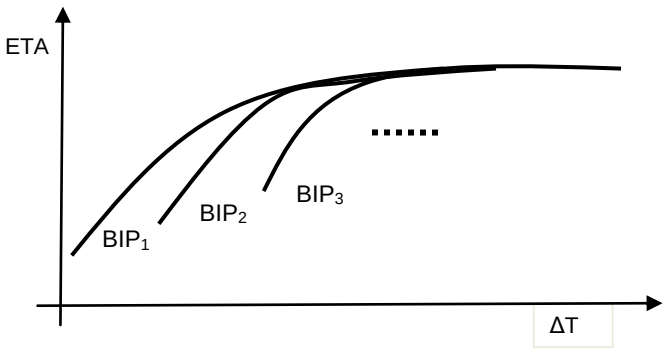

Figure 2-3 Burner characteristic map

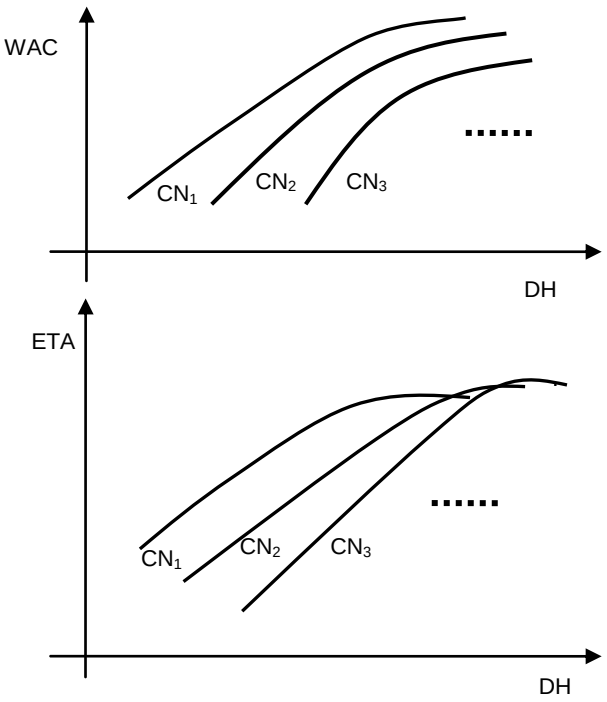

Figure 2-4 Turbine characteristic map

In this study, the severity of performance degradation of components caused by the faults described in Section 2.2.2 are represented by degradation indices including flow capacity index (DI_{FC}), pressure ratio index (DI_{PR}) and isentropic efficiency index (DI_{E}) (the details of the degradation indices can be found in [5]). Each of the degradation indices represents shift of speed lines in a certain direction (i.e. variation of its corresponding characteristic parameter) on compressor, burner, or turbine characteristic maps caused by their degradation. For example, if the degradation index DI_{FC} of a faulty compressor is 5%, it means its flow capacity drops 5% and the points on each of the speed lines will be moved to new locations where the mass flow rates are 5% lower than before, as shown in Figure 2-5 where point A is a point on a speed line of the clean component map and point A* is the new location of point A after reducing the compressor's flow capacity by 5%.

As mentioned earlier, different components have different performance characteristics parameters; hence the level of performance degradation of each

of them can be represented by different degradation indices. The lists of degradation indices of major components are given in Table 2-1.

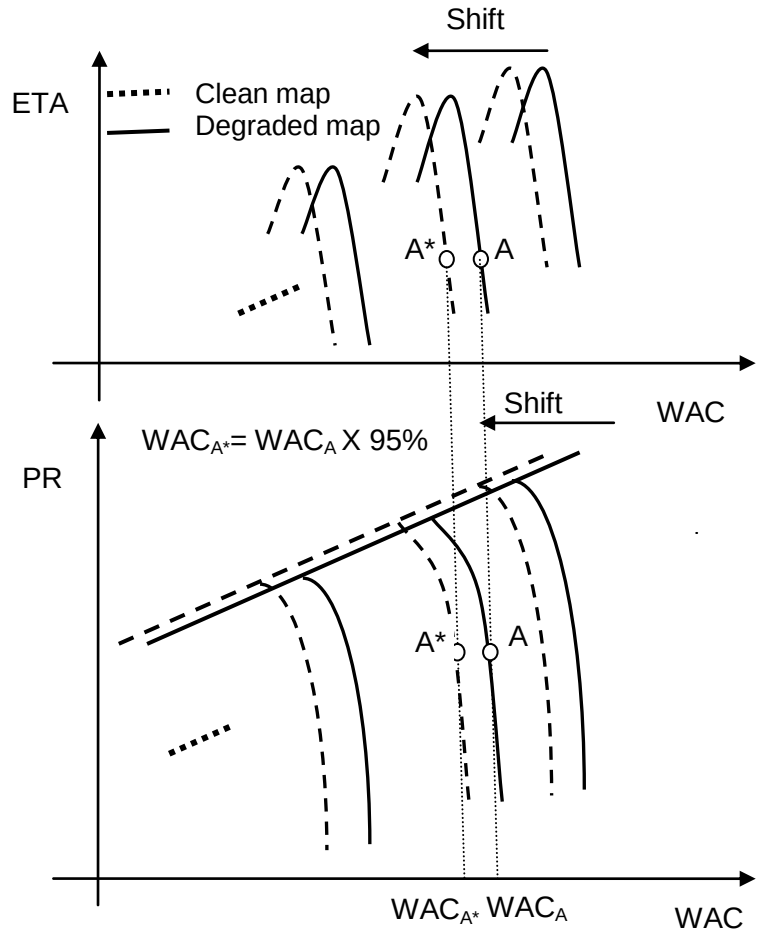

Figure 2-5 Degradation of flow capacity of a compressor

Table 2-1 Characteristic parameters and degradation indices of different components

Component	Characteristic Parameter	Degradation indices
Compressor	WAC, ETA, and PR	DI_{FC} , DI_E , DI_{PR}
Turbine	WAC, ETA and DH	DI_{FC} , DI_E , DI_{DH}
Burner	ETA	DI_E

2.1.4 Measurement uncertainty

Due to measurement noises and sensor biases, in practice, gas turbine gas path measurements are always subject to uncertainty. This uncertainty always results in differences (errors) between what we measure and what is true. Measurement noises and sensor biases usually lead to random errors and constant errors respectively. GPA approaches isolate faults by analyzing the deviation of measurements caused by deterioration, and constant errors caused by sensor biases are more likely to influence the true values of measurements rather than their deviations, hence in this study, only measurement noises are considered. The order of magnitude of the measurement errors due to noises can be the same as that of measurement deviations resulted from gas turbine component faults. Hence measurement errors can easily lead to misdiagnosis or poor diagnostic results. Much effort has been made to create effective approaches to reduce gas path measurement uncertainty. As a result of this, different approaches have been developed, such as Rolling Average, Exponential Average and auto-associative ANN. A brief introduction of the three above-mentioned approaches is included in this section.

Rolling average

It reduces noise for a measurement by taking a rolling average (RA) as expressed in Equation (2-1) of its values in certain last period:

$$RA(z_k) = \frac{1}{I}(z_k + z_{k-1} + z_{k-2} + \cdots + z_{k-I+1}) \qquad (2\text{-}1)$$

where z_k is the value of a measurement at time *k*, is the rolling average of the measurement, *I* is the total number of measurement values that involved in the rolling average calculation process.

Exponential average

In this method, an exponential average (*EA*) as expressed in Equation (2-2) of a measurement is calculated by its current value and the average of the last ten values, where the current value is associated with a weight with 15% and the average with a weight with 85%.

$$EA(z_k)=\frac{1}{10}(z_{k-1}+\cdots+z_{k-10})*0.85+z_k*0.15 \quad (2\text{-}2)$$

Auto-associative ANN

Auto-associative ANN can be used as a filter for removing noise from measurements. More details about noise reduction using auto-associative ANN can be found in [6]. Compared with the above-mentioned two methods, auto-associative ANN has an advantage that it can filter noise with much less time delay.

2.1.5 Measurement selection

GPA is used to analysis engine component deterioration based gas path measurement deviations caused by the deterioration. Usually the number of gas path measurements that a gas turbine engine can be equipped with is limited in practice due to high cost or installation difficulties. Thus selecting appropriate measurements affects GPA base diagnostics approaches to provide reliable diagnostic results.

So far, many measurement selection methods have been developed. Stamatis et al. [7] used singular value decomposition to analyze the relationships between measurements and component health parameters and developed a measurement selection method based on sensitivity analysis. Provost [8] introduced a selection method based on observability and correlation analysis. Ogaji et al. [9] studied the impact of measurement selection on the quality of diagnostic results by ranking all potential measurements by their sensitivities to different faults and using different combinations of measurements having higher sensitivities for diagnostics. However the number of possible combinations can

be very high and hence the selecting process may be very time-consuming. Sowers et al. [10] introduced a measurement selection approach based a GA which is able to avoid combinatorial selection process and search for an optimal selection solution. However the GA searching process sometimes can be computationally expensive. Jasmani et al [11] developed a sensitivity and correlation analysis based method to select appropriate measurements for multiple-fault diagnostics.

Most of the existing measurement selections focus on selecting appropriate measurements to achieve optimal fault quantification as many GPA based diagnostic approaches carry out fault isolation and quantification simultaneously. However there are also approaches which isolate faults prior to fault quantification. Thus it is may be worth to study a proper method for selecting measurements for fault isolation.

2.1.6 Gas path analysis technologies

Urban [3] made a breakthrough in gas turbine diagnostics in 1967 by introducing a linear model based approach, which generally called linear Gas Path Analysis (LGPA). This approach relies on the assumption of linearity of gas turbine thermodynamic behaviour. Many optimal estimation theories, such as weighted-least squares [12], and Kalman filter [13] have been employed to help LGPA to deal with measurement noises and sensor biases. Weighted-least-squares is used by General Electric in their diagnostic tool (TEMPER) [14]. Pratt and Whitney employed Kalman filter with several adaptations in their diagnostic tools (MAPIII, TEAMIII, EHM and ADEM). Rolls Royce employed a modified version of Kalman filter which was developed by Provost [8] and is called “Concentrator” in their diagnostic tool (COMPASS). In order to take the non-linearity into account, benefiting from the development of engine performance model non-linear GPA diagnostic approaches [15][16] [17] were developed. However non-linear model based approaches have a shortcoming that it may get stuck at a local optimum. Hence Zedda et al. [18] introduced non-linear diagnostic approach based on Genetic Algorithm (GA) to overcome

this shortcoming. Since the 80th there has been a rapid growth in the interest of the development of diagnosis approaches without involving complex performance models in the process (non-model based methods), this leaded to the successful applications of some artificial intelligence (AI) approaches in gas path diagnostics, such as artificial neural networks (ANNs) [19], rule based fuzzy logic [20], Bayesian-belief networks (BBNs) [21] and expert system (ES) [22]. However, these AI based approaches usually need to learn from the dataset which contains large information regarding to engine degradation, in order to obtain essential knowledge for fault detection, isolation, or quantification, and engine performance models are always still required to simulate degraded engines' performance to construct such a dataset. Most of the above-mentioned diagnostic approaches will be described in detail in the following sections.

2.1.7 Linear gas path analysis

In 1967, a linear model based diagnostic approach generally called Linear Gas GPA was proposed by Urban [23]. In this approach, the relationship between the changes in measurements (independent engine parameters) and in health parameters is assumed to be linear. Such a linear relationship is expressed in Equation (2-3) in a matrix form:

$$\Delta\hat{z} = H \cdot \Delta\hat{x} \tag{2-3}$$

The above equation can be converted to:

$$\Delta\hat{x} = H^{-1} \cdot \Delta\hat{z} \tag{2-4}$$

Where $\Delta\hat{z}$ is the measurement vector, $\Delta\hat{x}$ is the health parameter vector, H is usually called Influence Coefficient Matrix (ICM) and it inverse matrix H^{-1} is referred to as Fault Coefficient Matrix (FCM).

The Equation (2-4) can be used to calculate the change in health parameters caused based on the deviation in measurements in real engine. Linear GPA has the following advantages:

1. It is able to perform multiple fault diagnosis, which is a breakthrough in gas turbine diagnosis.
2. It is also capable of implement of fault quantification which can be represented by the change in health parameters.
3. It has low complexity.

However it also has the following limitations to its application in gas turbine diagnosis:

1. ICM must be available; hence an engine performance model is necessary in order to obtain such a matrix.
2. ICM is invertible in order to obtain FCM; therefore the number of measurements must not be less than the number of health parameters.
3. It cannot deal with measurement noises and sensor biases.
4. The assumption of linearity limits its application as when the deviation in health parameters increases the relationship between the health parameters and measurements become more non-linear and the effectiveness of LGPA reduces.

2.1.8 Non-linear gas path analysis

Nonlinear GPA was developed to overcome the limitations caused by the assumption of linearity in linear GPA by solving the nonlinear relationship between the changes in measurements and in health parameters. House et al. [16] developed an iterative non-linear GPA method. Escher [17] did further development by introducing a nonlinear GPA method with an iterative method called Newton Raphson.

As discussed in the previous section, the linear relationship between measurement vector and health parameter vector can be expressed by Equation (2-5) below:

$$\Delta\hat{x} = H \cdot \Delta\hat{z} \tag{2-5}$$

In the nonlinear GPA method, a correction $\Delta\vec{x}$ is introduced to the solution vector:

$$\hat{x}_{new} = \hat{x}_{old} + \Delta\hat{x} \quad (2\text{-}6)$$

where $\hat{x}_{new}$ is the new predicted health parameter vector after an iteration, and $\hat{x}_{old}$ is the previous predicted health parameter vector.

By using Newton Raphson method, as shown in Figure 2-6, the process in Equation (2-6) is repeated, after each iteration, according to $\hat{x}_{new}$, the predicted $\Delta\hat{z}$ can be calculated by Equation (2-3), if the difference between the predicted and actual measurement vectors is too large, then another iteration will be carried out until the difference become less than a threshold defined by the user.

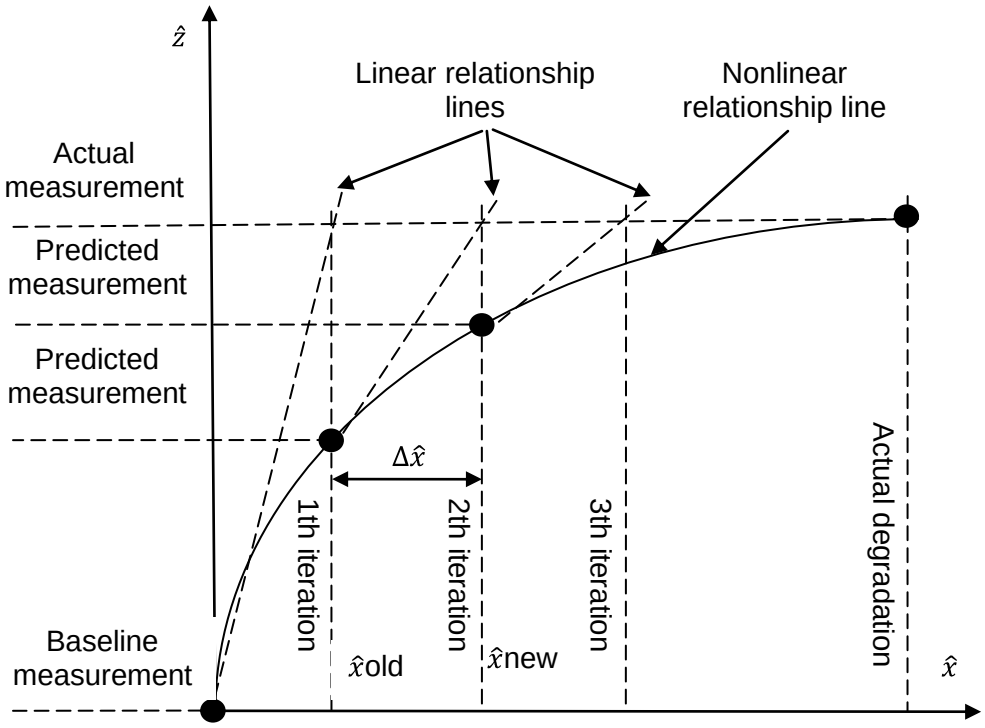

Figure 2-6 Illustration of nonlinear GPA [17]

The accuracy of degradation prediction can be largely improved by the introduction of the iteration process. Although the introduction of the iteration process addresses the limitations caused the assumption of linearity, nonlinear GPA still inherits several shortcomings from linear GPA, such as:

1. Influence Coefficient Matrix must be available.

2. The number of measurements must not be less than the number of health parameters.
3. It cannot deal with measurement noise and sensor bias.

In addition, the iteration process also causes other problems:

1. It may get stuck in local optimums.
2. It may not be able to converge at all.
3. Increase of computational time and complexity of the diagnostic system

2.1.9 Kalman filter based GPA: linear approach

Kalman Filter (KF) [24] is an optimal estimation method, which processes measurements to minimize estimation error of the stat of a dynamic system with noisy measurements. KF has been successfully applied in lots of different fields. The concept of Kalman Filter, a linear filtering approach, is based on the following assumptions, noises are time-independent, and are Gaussian distributed. The following five equations are employed in KF to calculate an optimal solution (i.e. optimal health parameter vector) at certain time:

State Equation:

$$\hat{x}_k^- = V_1 \hat{x}_{k-1} + V_2 u_k \tag{2-7}$$

Predicted State Error Covariance Equation:

$$\mathbf{E}_k^- = A\,\mathbf{E}_{k-1} A^T + Q \tag{2-8}$$

Kalman Gain Equation:

$$K_k = \mathbf{E}_k^- H^T (H \mathbf{E}_k^- H^T + R)^{-1} \tag{2-9}$$

Measurement Equation:

$$\hat{x}_k = \hat{x}_k^- + K_k(\hat{z}_k - H\hat{x}_k^-) \tag{2-10}$$

Updated State Error Covariance Equation:

$$\mathbf{E}_k = (I - K_k H)\,\bar{\mathbf{E}}_k \tag{2-11}$$

Where:

$\hat{x}_k^-$ is predicted state vector (i.e. health parameter vector) at time k

$\hat{x}_{k-1}$ is state vector after optimization at time k-1

$\hat{x}_k$ is state vector after optimization at time k

V_1 is the state transition model

V_2 is the control-input model which is applied to the state control vector

H is the Influence Coefficient Matrix

$\bar{\mathbf{E}}_k$ is predicted state error covariance at time k

$\mathbf{E}_k$ is state error covariance after optimization at time k

$\mathbf{E}_{k-1}$ is state error covariance after optimization at time k-1

K_k is the Kalman gain

u_k is state control vector

Q is process covariance

R is covariance of measurement noise

$\hat{z}_k$ is measurement vector at time k

The process of optimization based on Kalman filter is shown in Figure 2-7, it usually contains 2 steps:

- Step1: Prediction of State (i.e. degradation): Equation (2-7)) is used to predict the state of a system. Next, based on Equation (2-8)), calculate predict state error covariance at time k.
- Step2: Optimization of State: Then in Equation (2-9) it uses the predict state error covariance and covariance of measurement noise to calculate a factor called Kalman gain which defines how much the predicted state vector and the state vector calculated by measurement vector should be weighted in order to obtain a optimal state vector. Next, calculate the

optimal state error covariance in Equation (2-11). The optimal state vector and optimal state error covariance at time k then will be used to predict state vector and stat error covariance for next cycle. The estimation error will keeps reducing as the operation cycle goes on.

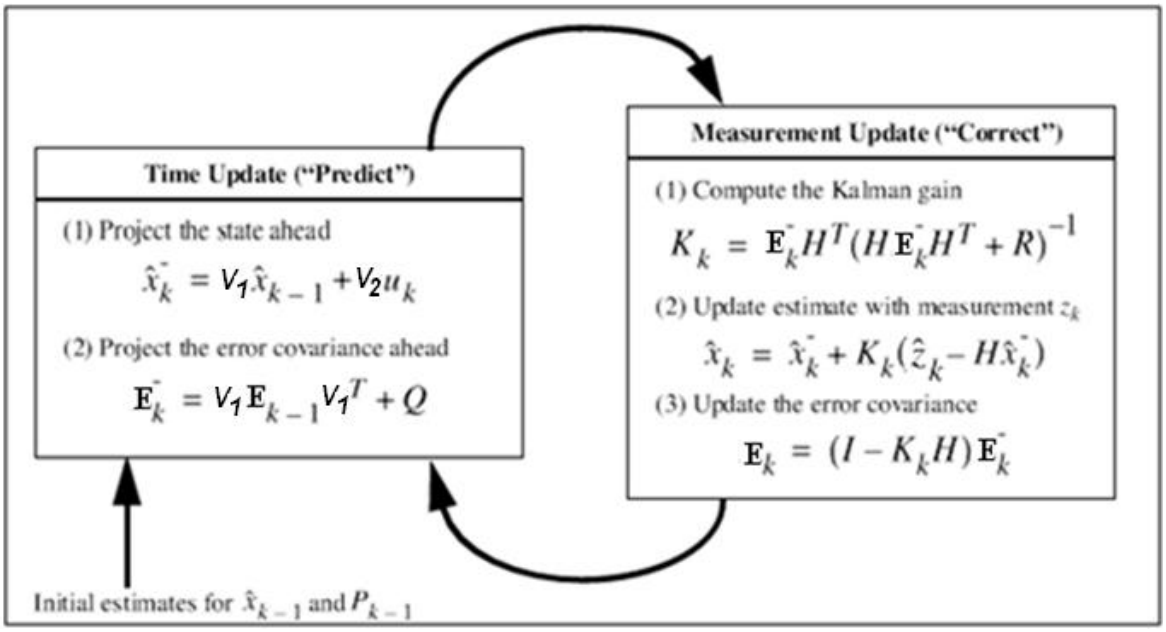

Figure 2-7 Operation cycle of Kalman Filter [24]

The advantages to apply Kalman filter in GPA are listed as follows [17] [18]:

1. It can deal with measurement noises which are Gaussian distributed.
2. It is an efficient optimal estimation method, which leads to low cost.
3. Its operation process is recursive, only information from one time before the operation time is needed. Therefore it has low requirements on computer memory.
4. It also can deal with sensor error by introducing sensor biases in operation cycle

However it also faces the following problems [17] [18]:

1. Kalman Filter always tries o "smear" the faults caused by few gas turbine components or sensors over many components or sensors. This problem is usually referred to as "smearing" effect.

2. It is based on an assumption that the process noise covariance is constant in the theory, but in reality this value is always not a constant value.
3. It is still a linear approach; hence it shares one of the shortcomings that linear GPA has, which is that when the deviation in health parameters increases the prediction error will increase.
4. The estimation process sometimes does not converge.

2.1.10 Kalman filter based GPA: nonlinear approach

The KF mentioned in the previous section is based on a linear assumption. However gas turbine thermodynamic behaviour is usually nonlinear. The intention to improve the performance of linear KF based GPA by modelling the engine thermodynamic behaviour more accurately leaded to the employment of nonlinear KF. Two of the most common nonlinear KF are extended Kalman Filter and iterated extended Kalman Filter. However, according to Marinai [25], both of them produce biased and sub-optimal estimates due the linearization of the cost function and this may lead to low accuracy estimation, apart from that, Zedda [18] discussed another potential solution which divides the cost function minimization (e.g. engine health state prediction error minimization) into two steps, a linear step followed by a non-linear step, by introducing new non-linear combinations of the unknown states in the first step.

2.1.11 Weighted-Least-Squares: linear approach

Weighted-Least-Squares (WLS) [26] is another optimization method which has been applied in GPA. It is able to take measurement noises and sensor biases into account. In this approach, WLS is employed to minimize the error between the actual measurement deviations and the predicted. This is accomplished by minimizing a cost function denoted by J and it is expressed in Equation (2-12) below:

$$J = \frac{1}{2}[\left(\frac{\Delta HP_1}{\sigma_{HP_1}}\right)^2 + \left(\frac{\Delta HP_2}{\sigma_{HP_2}}\right)^2 + \cdots \left(\frac{\Delta HP_l}{\sigma_{HP_l}}\right)^2 + \left(\frac{e_{Z_1}}{\sigma_{Z_1}}\right)^2 + \left(\frac{e_{Z_2}}{\sigma_{Z_2}}\right)^2 + \cdots + \left(\frac{e_{Z_i}}{\sigma_{Z_i}}\right)^2] \quad (2\text{-}12)$$

where HP_1, HP_2, … , HP_i, are health parameters, σ is standard deviation (a standard deviation can be used to represent the statistical variation of a health parameter or a measurement for overhauled engines [25]), e is a measurement error, Z_1, Z_2, … Z_i, are measurements, the subscript l is the total number of health parameters, and the subscript i denotes the number of measurements.

In Equation (2-12), there two types of unknowns: health parameter and measurement error, however each measurement error can be expressed by the difference between the predicted measurement which can be represented by the health parameter deviations and the ICM and the actual measurement. Therefore the minimum J can be calculated by solving:

$$\frac{\partial J}{\partial \Delta HP_1} = 0 \quad (2\text{-}13)$$

$$\frac{\partial J}{\partial \Delta HP_2} = 0 \quad (2\text{-}14)$$

……. ……

$$\frac{\partial J}{\partial \Delta HP_l} = 0 \quad (2\text{-}15)$$

From above equations a set of optimal health parameter deviations in terms of minimizing the total error can be calculated out:

From Equation (2-13)=> Optimal ΔHP_1

From Equation (2-14) => Optimal ΔHP_2

……

From Equation (2-15) => Optimal $\Delta \mathrm{HP_l}$

The cost function *J* can also be expressed in Equation (2-16) by matrix nomenclature:

$$J = \frac{1}{2}\{(\hat{x}^T E^{-1} \hat{X}) + (\hat{z} - H\hat{x})^T R^{-1} (\hat{z} - H\hat{x})\} \quad (2\text{-}16)$$

where $\hat{x}$, $\hat{z}$, E, H and *R* have been defined in the previous section, and the solution is shown in Equation (2-17):

$$\hat{x}_o = (E^{-1} H^T R^{-1} H)^{-1} H^T R^{-1} \hat{z} \quad (2\text{-}17)$$

where $\hat{x}_o$ is the optimal health parameter vector to minimise *J*.

The major advantage of WLS based GPA is very obvious: it is able to deal with measurement noises and sensor biases. According to Doel [26], its major limitations are:

1. This is a linear approach; hence it cannot provide accuracy solution for large degradation caused component(s).
2. The “smearing” effect, described in the previous section, also occurs in this approach

2.1.12 Genetic Algorithm

GA is a robust optimization algorithm, which is used to search for optimized approximate solutions for optimization problems. Unlike traditional methods that possibly lead to local optimized solutions, GA is a global searching method, and it also can be categorized as one of stochastic search methods. Zedda et al. [27] first introduced a gas turbine sensor fault diagnostic system based on GA which takes measurement noises and biases into account. Later on, Gulati et al [28] made further development on the application of GA in diagnostics and developed a multiple-operation-point analysis method based on GA, which is

able to perform diagnostics even there is only limited available measurements. GA theory is developed based on the Theory of Evolution of Darwin's and the Theory of Heredity of Mendel's. So the following concepts are employed from these two theories:

- String: chromosome in the Theory of Heredity of Mendel's and stands for a potential solution to a problem.
- Population: a set of potential solutions (Strings).
- Gene: It is used to describe characteristics of strings.
- Gene Position: It represents location of genes in a string.
- Fitness: It is used to evaluate the effectiveness of each solution.
- Selection (Reproduction): It is a process to select strings that have higher fitness to produce next generation (new strings). It is also referred to as reproduction.
- Crossover: It is a process to switch genes that have the same gene positions in two strings to produce two new strings.
- Mutation: It is a process to randomly change the value of genes of strings.

Generally, as shown in Figure 2-8, the searching process of GA has the following steps.

- Coding: convert potential solutions of the real problem to strings. Coding is the bridge connecting real problems and GA.
- Initialization: randomly choose strings to form an initial population. The size of the population depends on the nature of the problem.
- Selection (Reproduction): calculate a fitness for each string, then based on the fitness of each string, select the strings with higher fitness, and eliminate the rest.
- Crossover: those selected strings are randomly paired off, and switch some genes in the same gene position to produce the same number of new strings as the number of the selected strings (strings have higher fitness are more likely to be selected for crossover).

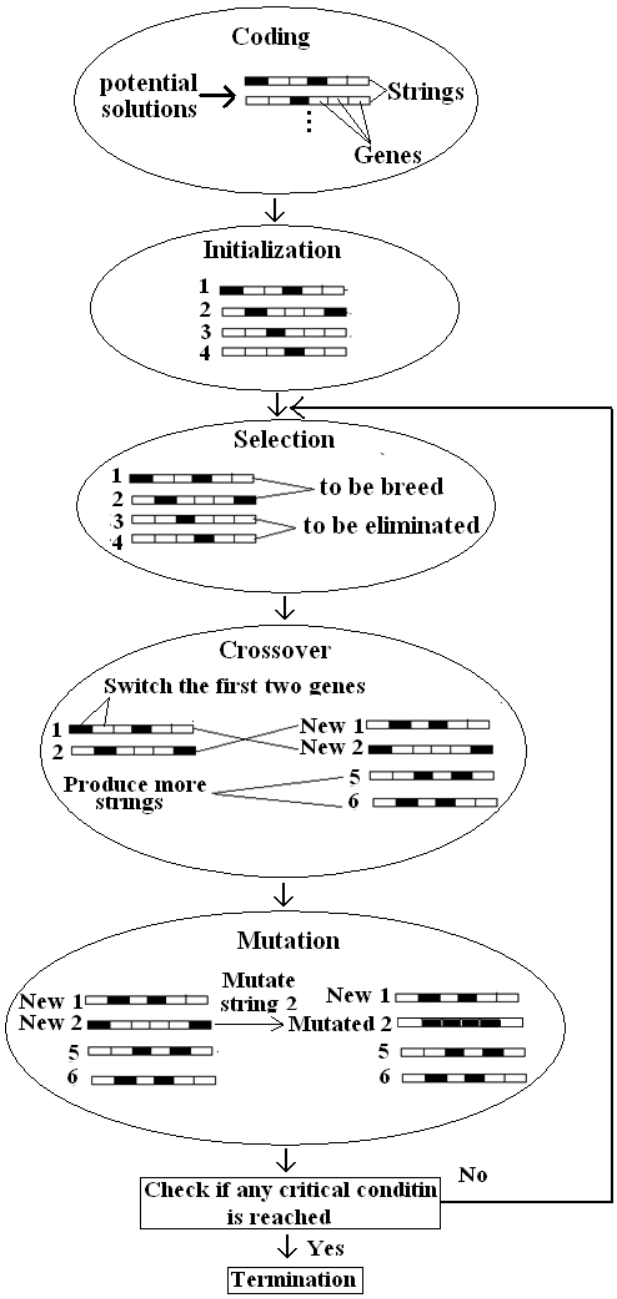

Figure 2-8 Illustration of a generation cycle of Genetic Algorithm

- Mutation: randomly change value of some gene(s) of string(s) to generate new strings. The intention of mutation is to enhance the diversity from the generation of previous population to the next.
- This generation cycle will be repeated until one of the following conditions has been satisfied: a string with a desired fitness has been found; limitation of computing time or cost has been reached; the number

of generation has reached the value defined by operator; a global optimal solution has been found.

The gas turbine faults diagnosis process using the Genetic Algorithm is similar to the process shown above. Equation (2-18) can be considered as the real problem that needs to be solved.

$$\hat{z} = f(\hat{x}) + \hat{\varepsilon} + \hat{\mu} \tag{2-18}$$

where $\hat{z}$ and $\hat{x}$ have been already defined in previous sections, $\hat{\varepsilon}$ is measurement noise vector, and $\hat{\mu}$ is sensor bias vector.

Each different $\hat{x}$ is a potential solution, and hence can be considered to be a string. In order to represent the fitness of string, an objective function, which measures the difference between the actual measurement parameter vector and the simulated measurement parameter vector, can be introduced, and the value of the fitness of string should be inversely proportional to the value of objective function. A classic objective function is expressed in equation (2-19).

$$OF=\sum \frac{(\hat{z}_{actual}-\hat{z}_{predictied})^2}{(\hat{z}_{predicted})^2} \tag{2-19}$$

where $\hat{z}_{actual}$ the actual measurement is vector and $\hat{z}_{predicted}$ is the predicted vector resulted from a solution ($\hat{x}$) obtained from a Genetic Algorithm. In order to calculate this $\hat{z}_{predicted}$, an engine model is needed.

Compared with traditional calculation based diagnostic methods, GA has following features [27]:

1. It is a global search algorithm; hence it is unlikely to get stuck in local optimizations.
2. It is capable of dealing with non-smooth functions or functions with multiple peaks.
3. GA directly deals with strings rather than numbers, leads to wide application of GA since the strings not only can represent numbers, but also structure object such as matrices, sets and so on.

However GA is also facing few limitations:

1. The GA searching process can be very time-consuming.
2. It is not very efficient since it still a stochastic search method.
3. It can find approximate location of optimal solution but cannot locate optimal solution very accurately within a short period.
4. Despite the crossover and mutation methods, the quality of the optimal solution provided by GA is largely depends on the number of generation, population size, and the size of searching area (i.e. the amount of all the potential solutions). The larger number of generation and population size, the better solutions can be found. For constant number of generation and population size, if the size of search-space can be reduced and simultaneously make sure that the optimal solution still locates in the reduced search-space) the possibility to obtain better solutions in each generation increases. Therefore due to these features, it is difficult to operate GA.

2.1.13 Expert system

ES is the first application of artificial intelligence (AI). The first ES was proposed in 1956. After then, many ESs were developed, but most of them were used for research purposes. In 70^{th}, as the development of computer languages, ES was further completed. In 1990, ES was applied in gas turbine diagnostics by Doel [29]. Further applications of ES have been done by Torella [30], Winston et al. [31], Dundas et al. [32], and etc. So far, the application of expert systems in gas turbine diagnostics is still popular. The existing ESs include rule-based, model-based, and case-based systems. Some of the developed expert systems are listed as follows [33]: ENGDOC, TEXMAS, HELIX, XMAN, TIGER, IFDIS, SHERLOCK, and etc. In an ES based on diagnostic approach, existing knowledge of gas turbine diagnostics is employed and usually represented in a way that similar to human beings' logic, and then such knowledge can be used to solve gas turbine diagnostic problems.

An ES normally consists of three parts, knowledge base, inference engine and user interface, the definition of them is shown below. A configuration of a simple expert system is illustrated in Figure 2-9.

- Knowledge base: It uses knowledge representation formalism to capture and codify knowledge of human experts or other sources. Then the knowledge being codified, which can be used by inference engine to search right solutions for certain problems, will be stored in knowledge base.

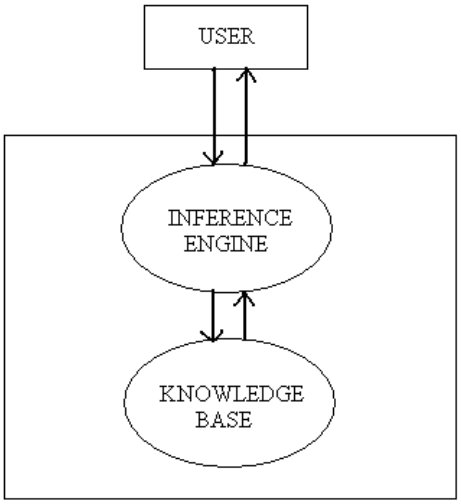

Figure 2-9 Illustration of a simple ES

- Inference Engine: By using inference rules (e.g. if-then rules) and knowledge from knowledge base, inference engine is not only able to find solutions for certain tasks, but also able to produce an explanation to each task.
- Interface: allow user to input data and the system output solutions or explanation.

ES based diagnostic approaches have two major advantages:

1. It can provide interpretations to each solution, which helps user to understand the problems they are dealing with.
2. It can deal with imprecise and uncertain information.

and the shortcomings of them are:

1. It cannot be adapted unless knowledge is modified or updated.
2. Its performance relies on the quality of the knowledge provided.
3. It cannot deal with the problems that are not encountered in knowledge base

2.1.14 Artificial neural network

Artificial neural networks (ANNs) are a robust information process system. Early effort to apply the ANN in gas turbine diagnostics was made by Dietz et al. and Denny [40]. So far, many different ANNs have been applied in gas turbine measurement noise reduction, sensor fault detection, fault detection, isolation and quantification. Li [33] described existing applications of the ANN in gas turbine diagnostics, and pointed out that the most popular ANN used in gas turbine diagnostics is Feed-Forward Back-Propagation Networks (FFBPNs). Many researchers have made contributions to the application of FFBPNs in diagnostics, such as Torella et al. [41], Eustace [42], Ogaji et al.[19] and so on. Some of other types of ANNs have also been employed. Eustace et al. [43] introduced a probabilistic ANN for diagnostics, and a self-organizing ANN was used in an EHM system by Roemer [44].

The human brain, which roughly consists of 10^{10} neurons, is a highly complex and very robust information analysis system. Simulation of functions human brain for solving real problems has been a popular research field for many years; this leaded to the development of artificial neural networks, which has similar structure to biological neural networks (e.g. the human brain). An illustration of a single neuron is shown in Figure 2-10 below.

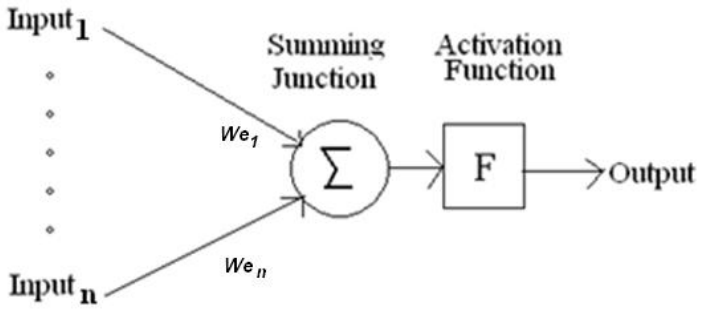

Figure 2-10 Illustration of a single neuron

In Figure 2-10, We_1, … We_n, are weights, which used to store knowledge. Each neuron performs a weighted sum of its own inputs, and consequently passes the sum through an activation function, which controls the activation level of this neuron. An ANN consists of many neurons; and different networks can be form by connecting neurons in different ways. The way that the ANN works is to learn from samples regarding to the problems needed to be solve, and store knowledge that obtained from the learning process in the ANN by adjusting weights connecting neurons, and once the knowledge has been appropriately stored (i.e. appropriate weights have been found), the neural network can be used to solve new problems in the range of the samples. The process to adjust weights is usually referred to as leaning.

Existing neural networks can be classified by the type of the learning algorithm employed by them, or by the way signals are transferred in the networks. There are majorly three types of learning algorithms: supervised, unsupervised, and reinforcement; and there are two ways to transfer signals in networks: feed-forward and feedback (also known as interactive or recurrent). For feed forward neural networks where connections between the neurons do not form cycles, signals always transfer on one direction, and it doesn't move backwards, while in feedback neural networks, signals can move in both directions since the connection cycles are included. Unlike feed forward neural networks which fixed state, in order to reach an equilibrium state, feedback neural networks need to change their states continuously; therefore their complexity can be very high.

Take the FFBPN as an example to demonstrate how it is used for diagnostics. An illustration of a multiplayer FFBPN is shown in Figure 2-11. Normally, The FFBPN operates in two steps:

- Learning: in this step, samples (i.e. training data) are used for adjusting weights by minimizing the error between predicted outputs from the network and the actual outputs in the samples.
- Operation: once the learning or training step is accomplished, an output can be predicted by the network for each new input.

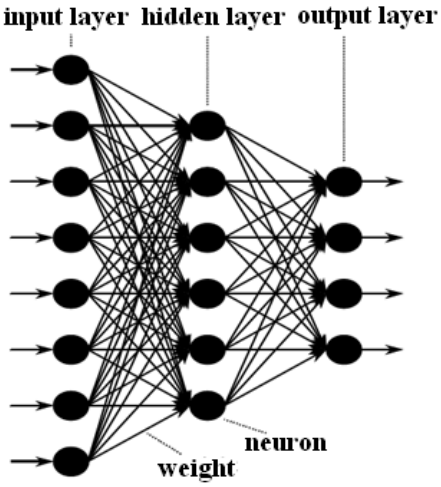

Figure 2-11 Illustration of a multiplayer FFBPN

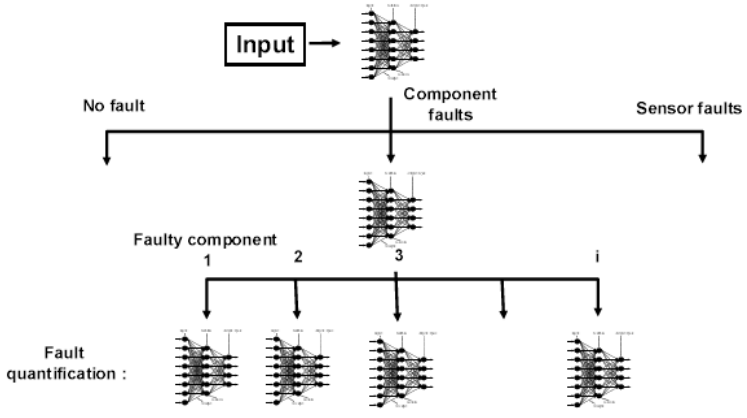

Figure 2-12 Illustration of a gas turbine diagnostic framework based on FFBPNs [19]

A diagnostic system based on ANN usually involves more than one network as shown in Figure 2-12. In this figure, the system contains different networks for different tasks, e.g. different networks need to be built for fault detection, isolation, and quantification respectively. Having a network for each task, such a diagnostic system performs can provide better performance [19]. The ANN based GPA has following advantages:

1. It addresses the non-linearity problem of gas turbine diagnostics.
2. It can deal with the uncertainty of measurements.
3. The ANN can be used to form filters to pre-process measurements to reduce measurement noises.

But it also has some limitations:

1. Large amount of training samples are required to train ANNs.
2. The learning or training process can be time-consuming; however this issue may be becoming resolvable as the fast development of the computer.
3. It cannot deal with the problems that are outside the range of the samples used for learning [45].
4. It can only provide solutions to diagnostic problems but it is difficult for it to give interpretations to the solutions [45], since it is a “black-box” modeling approach.

2.1.15 Fuzzy logic

Fuzzy logic is a generalization of fuzzy set theory that introduced by Zadeh in 1965. Tang [46] developed an ANN and fuzzy logic based gas turbine diagnostics system. A fuzzy rule and case based expert system was introduced by Siu et al. [47]. Ganguli [48] presented a fuzzy logic based helicopter rotor fault isolation system. Karvounis et al. [49] introduced a diagnostic system using hybrid model-based and fuzzy logic approach. Marinai et al [45] developed a non-linear model based fuzzy logic system at Cranfield University.

Fuzzy logic is an approach that solves problems by formalizing the human's reasoning capability that is approximate rather than precise [33]. Usually, the conventional way (e.g. crisp set) to describe affiliation between en element and a set is to use true and false logics, i.e. an element either belongs or doesn't belong to a set. In fuzzy set, a notation called membership function is introduced to describe affiliations between elements and sets instead of true and false logics. For example, assuming there is a car with a speed of 100 km/h, in order to define how fast this speed is, the conventional way is to set up a boundary between the crisp sets of "fast speed" and "slow speed" and the speed of car can be classified into to one of the sets, while in fuzzy set theory, the sets of "fast speed" and "slow speed" doesn't have such a boundary, and the speed can have two separate membership functions (one for "fast speed" and the other for "slow speed") to describe the speed of the car. Each membership function maps the same speed to a degree of membership in the 0 to 1 range. These memberships can then be used to determine how fast or slow the speed is. An illustration of comparison between the crisp set and the fuzzy set using the above-mentioned sample is shown in Figure 2-13.

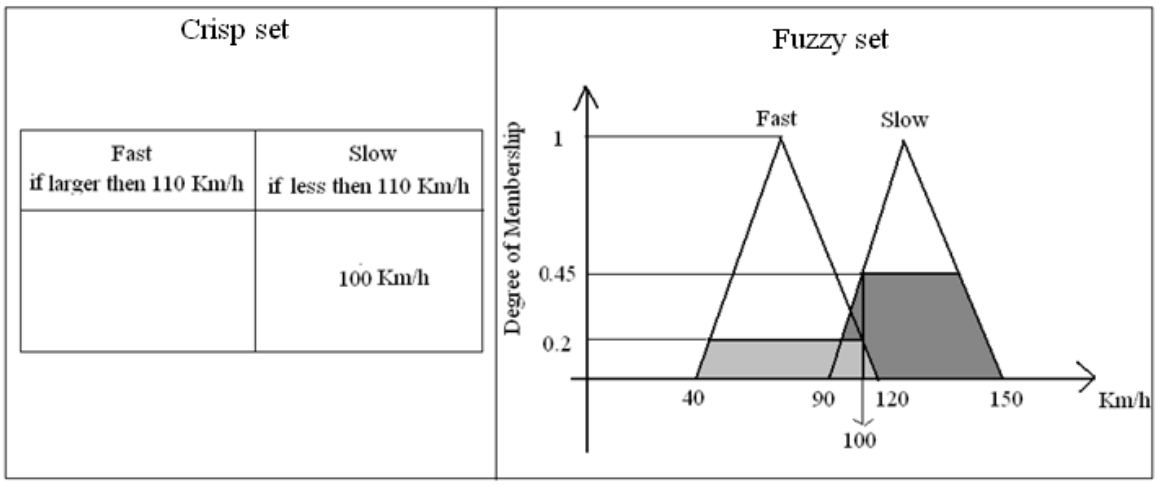

Figure 2-13 Comparison between crisp set and fuzzy set

A fuzzy logic system, which normally involves fuzzification, rule library, fuzzy inference and defuzzificaiton, is graphically represented in Figure 2-14. Fuzzy logic based GPA usually has the following steps [25].

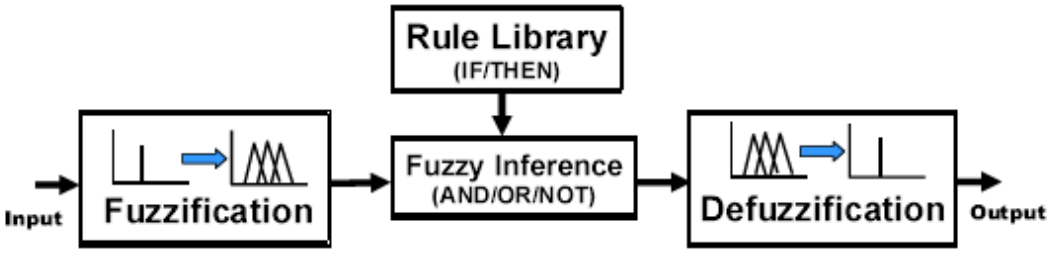

Figure 2-14 Configuration of a fuzzy logic system [44]

- Input fuzzification: construct membership functions for real engine measurements (input membership functions) and health parameters (output membership functions), and convert real values of the input functions into degrees of memberships.
- Application of Rules: the rules, which need to be obtained from other sources, representing relationships between measurements and health parameters stored in the system can be then used by a fuzzy inference engine to map the degrees of output functions.
- Output Defuzzification: calculate deviation of health parameters by converting the degrees of output membership functions to real values.

The features of fuzzy Logic based diagnostics can be highlighted as follows:

1. Just like ANN, it has the capability to deal with the non-linear nature of gas turbine diagnostic problems.
2. It is also able to deal with uncertainty.
3. The rules can be modified and updated in operations.
4. It is fast and hence can be used for on-line applications.
5. No engine performance models are required in the diagnostic process.

Marinai [45] summarized limitation of fuzzy logic based diagnostics:

1. It is not able to deal with the problems that are outside the range of its dataset. Therefore a large amount of fault samples covering all potential faults are required.
2. As the complexity of the problem needed to solved increases (e.g. the potential faults increases), the number of rules required increases as well.

3. Computational speed and burden can influence the quality of the solutions provided by the fuzzy logic system
4. Faults need to be accurately isolated using rules first.

2.1.16 Rough set

Rough set theory initially proposed by Pawlak in 1982 [34] establishes a sound basis for knowledge discovery in datasets. It is an extension of conventional set theory that supports approximations in decision making. The rough set itself is the approximation of a vague set by using a pair of precise sets called lower and upper approximations, which are an isolation of the domain of interest into disjoint categories. The lower approximation is a description of the domain objects which are known with certainty to belong to the subset of interest, whereas the upper approximation is a description of the objects which possibly belong to the subset [35]. For example, the conventional set theory is not suitable for describing a vague concept such as “good-looking people” as it is always hard to say if a person is good-looking or not. Therefore in rough set, two precise sets can be used to approximate this vague concept; one is the lower approximation set including all the people who can be definitely classified into the “good-looking” group, and the other is the upper approximation set including all the people who possibly can be classified into the group.

Rough set is a useful tool for pattern extraction, attribute reduction, data reduction, and decision rule generation. It has been applied widely in many different fields, such as machine learning, artificial strategic decision, pattern recognition, and knowledge discovery. Rough set also have been used for machine diagnostics. In machine diagnostics, if a dataset containing fault samples of a machine is available, rough set can be utilized as a tool to select appropriate attributes (representing the characteristics of faults of the machine) and remove necessary attributes from the database, and generate decision rules based on the reduced database. The generated rules, representing logic relationships between the selected attributes and the faults, then can be used for the diagnostics of the machine.

Although Chen and et al.[36] studied the possibility of applying rough set for gas turbine fault isolation, rough set is still novel technique for gas turbine diagnostics and more works need to be carried out to construct an effective fault isolation system based on rough set. In his method, rough set is used in a similar way as it is used in machine diagnostics. They applied the rough set based method for single fault isolation of a two-spool turbofan engine. Chen et al.'s study on application of rough set for fault isolation proves the possibility of using rough set for measurement selection and gas turbine fault isolation, however their method only has been applied for single fault isolation and the accuracy of fault isolation obtained by his method and the effectiveness of rough set based fault isolation are not discussed in the publication. Therefore there are still many problems need to be studied and solved in order to build a effective fault isolation system based on rough set, such as:

1. The effectiveness of rough set based fault isolation on single and multiple faults isolation
2. The effectiveness of rough set for selecting measurements for fault isolation.
3. The capability of the rough se based system to deal with measurement noises.

More descriptions of rough set theory and Chen et al.'s study on rough set are included in Section 4.2.1.

2.1.17 Hybrid system

In order to develop an effective diagnostic method, much effort has been made by various researchers to build up a hybrid system which combines two or even more different GPA methods. Volponi et al. [50] developed a hybrid ANN where part of the model was replaced by influence coefficients to improve the accuracy of diagnostics. Another hybrid ANN developed by Sun et al.[51] used a training rule to improve its convergence. A feed forward ANN and genetic algorithm based system was developed by Kobayahshi et al.[52], where the ANN were used to for prediction of deviation of health parameters, and the

genetic algorithm was for dealing with sensor biases. Green [53] discussed the incorporation of ANN with other AI approaches or gas turbine diagnostics and prognostics. Breese et al [54] described a model based expert system combining a Bayesian belief network [55]. Torella [56] combined an expert system and ANN for fault isolation. In addition, as mentioned earlier, a hybrid system combing fuzzy-logic and expert system was developed by Siu et al. [47], and Tang [46] introduced a fuzzy-logic and ANN based diagnostic method. Chen and et al.[36] introduced a hybrid system based on rough set and ANN for diagnostics where rough set is used for fault isolation and the ANN for fault quantification.

2.1.18 Summary

Although there are many mature technologies among the GPA approaches described earlier, none of them is capable of solving all the problems defined earlier regarding to gas path diagnostics [25], each of them has several limitations which has been described earlier. Li [33] carried out a review of GPA approaches and comparison of them in terms of computation speed and complexity (see Figure 2-15, please note rough set has also been added in to the figure by the author). Later on, Marinai et al. [25] also made a review of GPA approaches and summarized the pros and cons of them as shown in Table 2-2, X denotes that the approach has that feature, and MFI and SFI denote multiple fault isolation and single fault isolation respectively..

Basically, linear model based methods, such as LGPA, WLS, and Linear KF, are simple and fast solutions to diagnostic problems, but they are not able to solve the non-linearity of the diagnostic problems. Although non-linear model based methods, such as non-linear GPA, non-linear KF and GA, are able to overcome this shortcoming, the non-linear models also induce problems like high complexity and the possibility to get stuck at local optimums or un-convergence. Non-model based methods like ANN, ES, and fuzzy Logic do not have the problems caused by engine performance models, and most of them have the capability to deal with measurement noises and perform diagnostics

quickly once they have been set up, however their performance to deal with the problems outside their datasets is low. Therefore, in order to develop a more effective approach, it may be necessary to make more efforts to improve the existing diagnostic systems or develop a new more effective diagnostic system or a new hybrid diagnostic system combing two or more GPA based diagnostic methods which can cover the limitations of the other's.

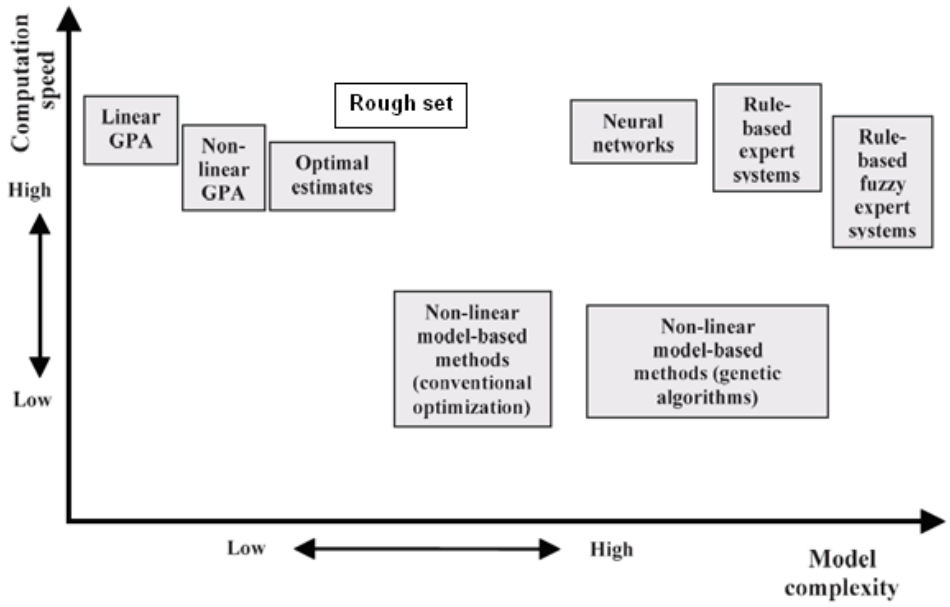

Figure 2-15 Comparison of gas path analysis approaches [33]

Table 2-2 Summary of gas path analysis approaches [25]

Strategy involved	Methodology									
	Linear GPA with ICM inversion	Non-linear GPA with ICM inversion	Linear Kalman-filter	Linear WLS	Non-linear Kalman-filter	Non-linear model-based with GA	Artificial neural-networks	Bayesian-belief networks	Expert systems	Fuzzy-logic
Linear/non-linear model	Linear	Non-linear	Linear	Linear	Non-linear	Non-linear	Non-linear	Non-linear		Non-linear
Small changes of health parameters	X		X	X						
Measurements of random noise			X	X	X	X	X	X	X	X
Bias			X	X	X	X	X		X	X
N parameters, M measurements	$M \geqslant N$	$M \geqslant N$	$M < N$	$M < N$	$M < N$	$M < N$	$M < N$	$M < N$	$M < N$	$M < N$
Single/multiple fault(s)	MFI	MFI	MFI	MFI	MFI	SFI/limited MFI	SFI/limited MFI	SFI/limited MFI	SFI/limited MFI	SFI/limited MFI
Smearing versus concentration			Smearing	Smearing	Smearing	Concentration	Concentration	Concentration	Concentration	Concentration
Difficulty and dependence on training/tuning			Prior knowledge	Prior knowledge	Prior knowledge	Number of string assignment	Long training and data selection	Effort in gathering info for setting-up		Rules explosion
Artificial-intelligence based						X	X	X	X	X
Computational burden (slow in recall mode)						X		X		
Model-free							X	X	X	X
High dimensionality problems						X	X	X	X	X
Data-fusion capability							X	X	X	X
Black-box not observable							X			
Good accuracy in predefined ranges only						X	X	X	X	X
Expert knowledge capability								X	X	X
On-wing			X	X	X		X			X

2.2 Gas turbine performance model adaptation

2.2.1 Introduction

Accurate gas turbine performance models are necessary for gas turbine performance analysis and gas path diagnosis. The accuracy of a performance model largely depends on its component characteristic maps. These maps are usually built by carrying out rigorous engine rig tests at different operation conditions. Hence, it is not realistic for the engine manufacturers to produce individual set of component characteristic maps for each gas turbine manufactured by them due to that the tests are not only very time-consuming but also costly. As a result of this, engine manufacturers usually only provide their customers a set of characteristic maps of major components (i.e. compressors and turbines) of the engines they received in the same fleet. However, the gas turbines in the same fleet usually slightly differ from each other in their performance due to manufacturing or assembly tolerances. Apart from that, after maintenances or overhauls engines normally need to be disassembled or re-built, this will lead to variation of their performance. In addition, for some gas turbine users, they even do not have the component characteristic maps of the fleet engines because the maps are proprietary information of the engine manufacturers, hence they may use generic maps for the performance simulation.

The performance difference between gas turbines in the same fleet, the performance variation after maintenances or overhaul, or the absence of the component characteristic maps produced by the engine manufacturers can influence the performance models' prediction accuracy. Most of existing gas path analysis approaches either are performance model based or require models to produce their dataset of engine deterioration, for this reason, inaccuracy engine performance models may lead to poor gas path diagnostic results. Therefore, so far, much effort has been made to establish an effective approach to improve the prediction accuracy of engine performance models by

tuning the existing component characteristic maps or generating new component characteristic maps based on engine test data.

2.2.2 Performance model adaptation techniques

Stamatic et al. [57] developed a gas turbine performance model adaptation method which improves the accuracy of engine performance models by using an optimal set of modification factors to modify component characteristic maps. Such an optimal set can be found by an optimization procedure. This method was further developed by Lambiris [58]. Later on, Kong et al. [59] introduced a new map scaling method based on system identification. In this method, component characteristics and scaling factors obtained at design point (DP) based on engine test data are utilized to derive new component characteristics at off-design (OD) points. In order to address the problems like low accuracies of scaled characteristic maps or even the absence of characteristic maps, Kong [60] developed an approach to produce new component characteristic maps using a Genetic Algorithm (GA) based on engine test data. Li et al. [61] and Roth et al. [62] [63] separately developed two different DP performance model adaptation approaches by minimising the difference between DP performance predicted by engine models and engine test data. To minimise such a difference, Li employed a Newton-Raphson based approach, while Roth utilized a minimum variance optimal estimator. Lo Gatto et al. [64] studied to use a GA to search for an optimal set of scaling factors to scale a component characteristic map based on engine test data at a single OD point. Marinai et al. [65] [66] further developed this method by seeking an optimal set of scaling factors for map scaling based on engine test data at multiple OD points.

2.2.3 Summary

Most of the off-design performance adaptation approaches described earlier used only a set of scaling factors to scale one component characteristics map. However the nature of the engine performance is usually non-linear, the difference between the engine model performance and test data at different off-design points may vary non-linearly. Due to this non-linear behaviour, only use

a set of optimised scaling factors to modify each component characteristic map to minimize the errors between the engine model performance and test data can be difficult.

3 Off-design performance adaptation using a Genetic Algorithm

3.1 Introduction

Engine off-design (OD) prediction accuracy of engine performance models is mainly depended on the quality of engine component characteristic maps. If inaccurate component characteristic maps used in an engine performance model, the predicted of off-design performance of the engine model will differ from the engine's actual performance thus producing prediction errors. Therefore, it is importance to adapt inaccurate component maps in order to match the predicted with actual performance of engines at corresponding off-design operating conditions.

In order to do so, Gatto et al. [64] introduced a GA based adaptation method. This method utilizes a GA to search for an optimized set of scaling factors (*SF*s) to scale a component map to minimize the difference between the predicted performance resulted from the engine model with the map to engine test data at one OD point. Marinai et al. [65] [66] further developed this method. In the further developed method, it still uses a GA to search for an optimized set of scaling factors; however this optimized set is able to be used to minimize the average performance prediction error at several OD points.

A common shortcoming of these two methods is that it ignores the non-linearity nature of the engine performance model hence making it difficult to achieve high prediction accuracy when only one identical set of scaling factors is used to scale different speed lines in a component map.

To overcome this limitation, different speed lines in a component map may need different sets of scaling factors. Therefore in this study, a new GA based adaptation method was proposed. The proposed method is able to search for an optimized set of scaling factors for each speed line a component map in order to minimise the performance prediction errors at different OD points.

3.2 Methodology

In this study, a GA based OD performance model adaptation approach, which modifies the component characteristic maps using multiple sets of scaling factors, is presented. It is worth mentioning that this study focuses on the OD performance model adaptation only and hence all the performance models used are assumed to be accurate at design point. Considering turbines are likely to work under choked conditions and consequently the quality of turbine maps doesn't affect the prediction accuracy largely, therefore this study focuses on modifying only the compressor maps to improve the prediction accuracy at OD points of engine performance models.

Sections 3.2.1 to 3.2.5 describe the important factors involved in this method, and the process of using this method for OD performance model adaptation is described in Section 3.2.6.

3.2.1 Engine test data

The main purpose of the developed adaptation approach is to minimize the predicted performance of the engine performance model and the real engine data at different OD points. Therefore deciding which OD points' performance need to be improve is the first step to use the developed approach to adapt an engine model. Users then can select the speed lines on compressor maps(s) of the model need to be scaled based on the corrected rotational speeds (*CNs*) covered the selected OD points. The *CN* can be calculated by Equation (3-1).

$$CN = PCN \times \frac{\sqrt{T_{DP}}}{\sqrt{T_{OD}}} \qquad \textbf{(3-1)}$$

where *CN* is the relative rotational speed (the ratio between actual absolute rotational speed and design point absolute rotational speed), and T_{DP} and T_{OD} are the inlet total temperatures at design point and at the actual operating condition respectively.

Once the speed lines need to be adapted are known, the next step is to gather engine test data at the *CNs* that the speed lines need to be adapted have. The engine test data will be used as the target performance data for finding appropriate scaling factors to each of the speed lines needed to be scaled so that the error between the predicted performance and the performance and the test data will be minimal at the *CNs*.

3.2.2 Scaling factor

As mentioned earlier in Section 2.1.3, a component may have more than one characteristic parameter; in order to in order to define the whole characteristic of the compressor, three characteristic parameters are required and they are listed in Table 3-1. Because of this, three scaling factors are required to scale the entire compressor map and they are also listed in Table 3-1

Each scaling factor is used to scale one characteristic parameter, and for all the points on a speed line in a component map, an identical scaling factor will be used for each of the characteristic parameters to scale the speed line. However such a scaling factor will differ for different speed lines. The scaling factors required for scaling compressor maps are also shown in Table 3-1.

Table 3-1 Characteristic parameters and scaling factors for compressor maps

Characteristic parameter	Scaling factor
WAC	SF_{WAC}
ETA	SF_{ETA}
PR	SF_{PR}

The definitions of the scaling factors are expressed in Equations (3-2) to (3-4).

$$SF_{WAC,OD} = \frac{WAC_{A*}}{WAC_A} \tag{3-2}$$

$$SF_{PR,OD} = \frac{PR_{A*}}{PR_A} \tag{3-3}$$

$$SF_{ETA,OD} = \frac{ETA_{A*}}{ETA_A} \tag{3-4}$$

where the subscript A and $A*$ represent the original and the modified characteristic parameters respectively.

3.2.3 Map scaling

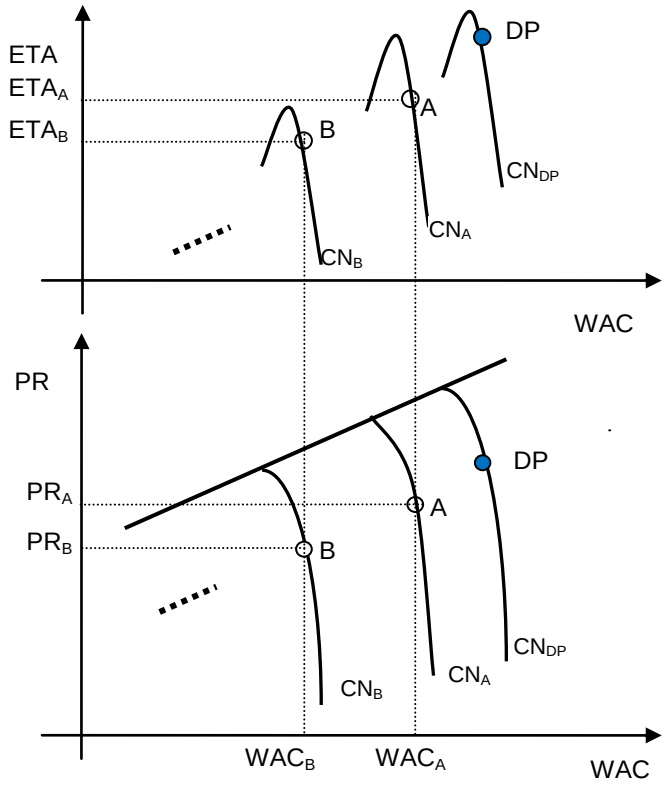

Figure 3-1 Illustration of a compressor map

In order to scale a component map to achieve minimum overall prediction error at OD points, different speed lines in each component map may need different sets of scaling factors. In this study, a set of optimal scaling factors will be generated for each of the speed lines needed to be scaled. Using the scaling factor, the speed lines in the original component map can then be modified according to the scaling factors, resulting in better performance prediction accuracy.

The following example is used to schematically illustrate the map scaling for a compressor map. Assume that the compressor map shown in Figure 3-1 is the

map used in an engine performance model and the speed lines CN_A and CN_B need to be scaled to improve the model's prediction accuracy. In addition also assume that there are available engine test data, which covers the engine performance at one operating point (point A) on the speed line CN_A and *also* one operating point (point B) on the speed line CN_B. The map scaling then can be carried out by the following two steps:

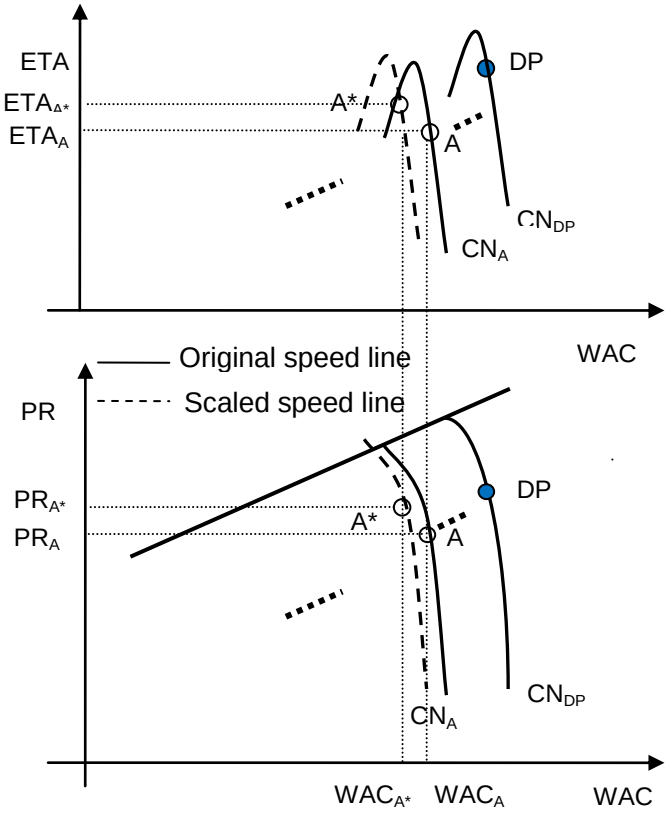

Figure 3-2 Illustration of scaling of a speed line in a compressor map

Step 1: find a set of optimal scaling factors SF_{WAC}, SF_{PR} and SF_{ETA} for the speed line CN_A by a GA (the process to find optimal scaling factors will be introduced in Section 3.2.5). The optimal scaling factors are able to give minimal performance prediction error at point A. The three characteristic parameters *WAC, PR*, and *ETA* on the speed line CN_A will then be modified according to their corresponding scaling factors to form a scaled speed line as shown in Figure 3-2, where the speed lines CN_B in original component map represented by the solid lines are shifted to form new scaled line represented by the dotted

line, and the adapted characteristic parameters marked by subscript "*" are calculated by equations (3-2) to (3-4). It is worth noting that the DP speed line is always fixed during the scaling.

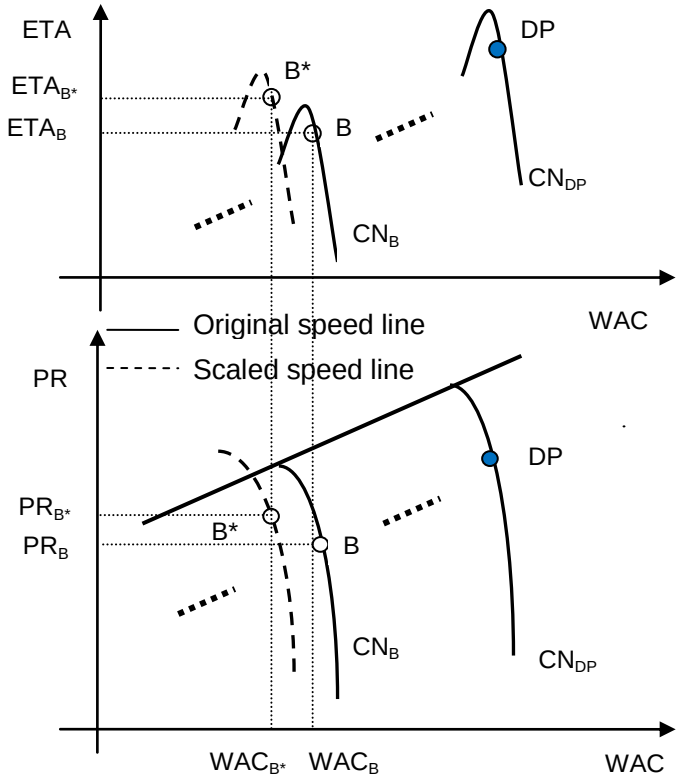

Figure 3-3 Illustration of scaling of a speed line in a compressor map

Once the scaling has been carried out to all the speed lines need to be scaled (i.e. in this case, the speed lines CN_A and CN_B), a scaled compressor map then can be obtained as shown in Figure 3-4. The scaled map then can be used by the engine performance model to provide more accurate performance prediction at OD points having corresponding operating points near the scaled speed lines.

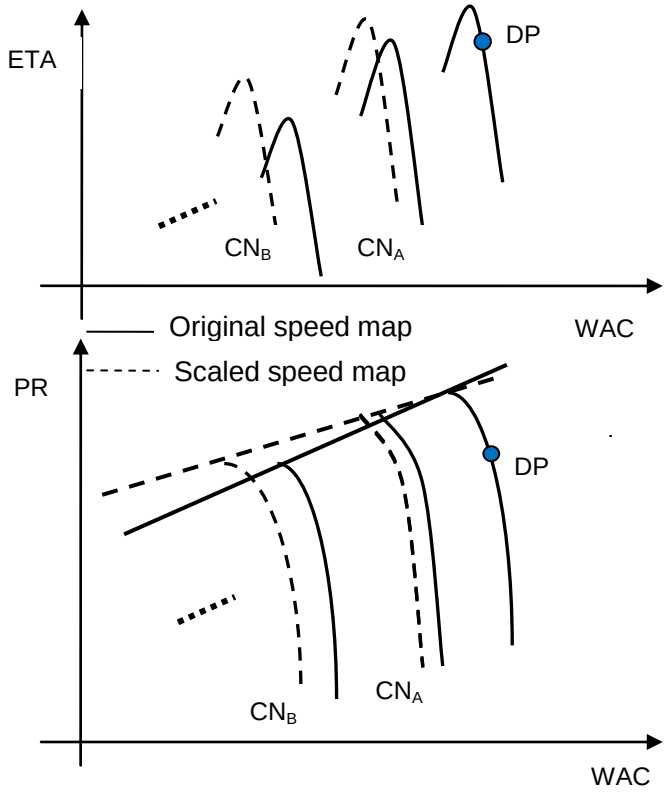

Figure 3-4 Original and scaled compressor maps

3.2.4 Objective function

In this study, the accuracy of prediction of an engine performance model at an OD point is measured by an objective function representing the average difference between the predicted gas path measurements and actual gas path measurements (i.e. engine test data) at this point. Such an objective function is defined in Equation (3-5).

$$OF = \frac{1}{L}\sum_{i=1}^{L}\left|\frac{P_{(predicted)i} - P_{(actual)i}}{P_{(actual)i}}\right| * 100 \qquad (3\text{-}5)$$

where L denotes the number of measurements, *P* denotes measurement, subscript *i* is measurement index.

In order to evaluate the effectiveness of each set of scaling factors in minimizing prediction error at its corresponding CN, each obtained set of scaling factors will be used to scale the speed line corresponding to the CN, and this scaled map will then be used by the engine model to predict engine performance at the CN.

Then based on the predicted performance and the accuracy performance data from the engine data, an objective function can be calculated to represent the effectiveness of the obtained set(s) of scaling factors to improve performance prediction accuracy. The higher objective function indicates higher effectiveness.

3.2.5 Genetic Algorithm

In order to search for an optimal set of scaling factor for each speed line having corresponding engine data in a compressor map, a GA is used in this study because of the following two advantages it has: firstly, it does not get stuck in local optima since it is a global search algorithm, secondly, non-smooth functions can be optimized because no derivatives are required [27]. A GA is able to find the fittest solution (string) among all potential solutions using three GA operations which are reproduction (selection), crossover and mutation.
Generally, as shown in Figure 2-7, the searching process of GA has the following steps.

- Initialization: generate an initial population of strings where each string represents a potential solution (i.e. in this study, a string represent a set of scaling factors) to the problem needed to be solved (i.e. in this study, minimizing prediction error of an engine performance model at a operating point on a speed line of a compressor map). Then each potential solution is evaluated by calculating its fitness (i.e. in this study, the fitness is calculated using equation (3-6).

$$Fitness = \frac{1}{1 + objective\ function} \tag{3-6}$$

- Selection (Reproduction): calculate fitness for each potential solution for evaluations, and then select the strings with higher fitness for crossover and mutation, eliminate the rest.
- Crossover: select a part of strings and randomly pair them off, and switch some genes in the same gene position to produce the same number of

new strings as the number of the selected strings (strings with higher fitness are more likely to be selected for crossover).

- Mutation: randomly mutate some of the strings to generate new strings with the intention to enhance the diversity from current generation to the next.

This generation cycle will be repeated until one of the following conditions has been satisfied: (1) a string with a desired fitness (i.e. target objective function) has been found; (2) limitation of computing time or cost has been reached; (3) the number of generation has reached the value defined by operator; or (4) a global optimal solution has been found (i.e. minimized objective function).

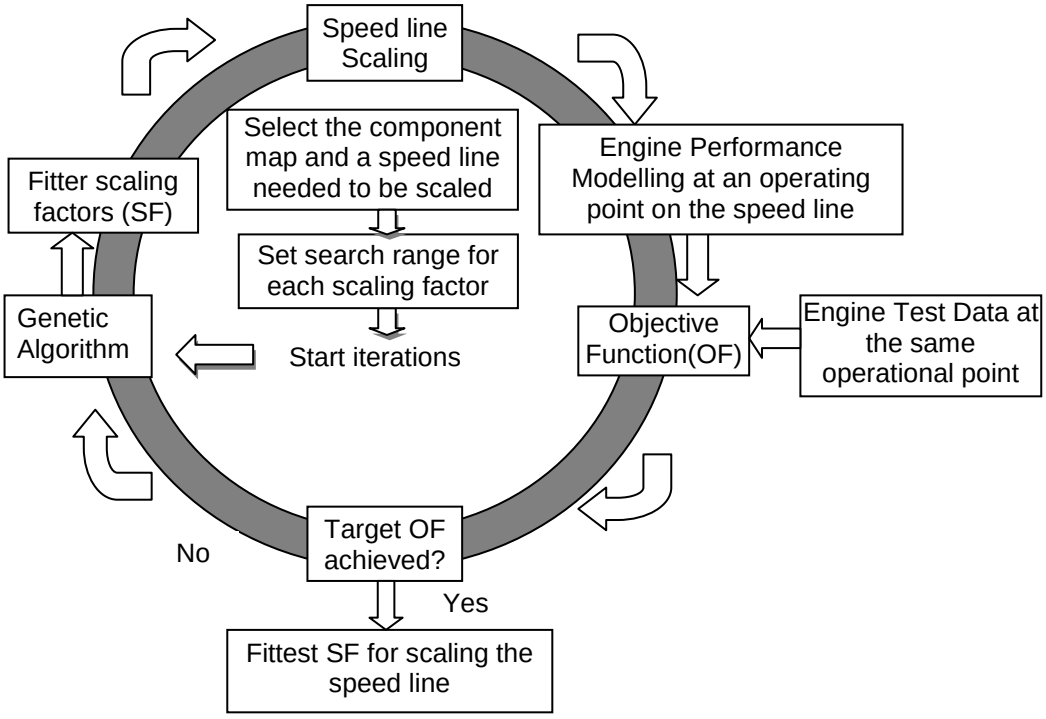

Figure 3-5 A flow chart of searching for optimal scaling factors for a speed line

Details of the process of searching for an optimal set of scaling factors for a speed line having corresponding engine test data (i.e. the data cover an operating point on the speed line) in a component map using a GA is illustrated in Figure 3-5. Once the to-be-adapted component map and speed line, and an operating point (having corresponding engine test data) on the speed line are selected, and the search range for each scaling factor (i.e. minimum and maximum values of each scaling factor) are defined, the GA will start to search for fitter sets of scaling factors within the defined ranges by the three GA

operations described earlier in this section. Each set will then be applied to scale the map(s), and the scaled map is then used by the engine performance model to produce simulated measurements. Based on these simulated measurements and engine test data at this operating point, an objective function can be calculated for the operating point. This process will continue until one of the conditions mentioned earlier has been satisfied.

It is worth mentioning that setting of the search ranges for each scaling factors is very essential to find an optimized set of scaling factors as the GA only searches for solutions within the pre-defined scope (i.e. search ranges of scaling factors). If the best solution is not included in the pre-defined scope, the GA will only provide a local optimized solution. Therefore it is very important to check the optimized scaling factors found by the GA to see if any of them has a value close to lower or upper boundary of its pre-defined search range. If so, the search ranges need to be adjusted to avoid local optimum. This done by modifying the search ranges so that the value of each of the optimized scaling factors will roughly lies in the middle of its new search range. After the modification the search process described in last paragraph needs to be carried out again. This modification and the search process need to be repeated until the value of each optimized scaling factor lies in the middle of its search range.

3.2.6 Adaptation process

Generally the adaptation process for a compressor map of an engine performance model based on the developed method consists of the following steps:

1) Choose the speed lines needed to be adapted in the compressor map and gather sufficient engine test data for the speed lines needed to be adapted
2) For each of the speed lines, use the GA to search for a set of optimized scaling factors by the process described in Figure 3-5.
3) Scale all the speed lines in the component map according to their corresponding scaling factors to form a scaled map which can then be

used by the engine performance model to provide better OD performance prediction for the engine.

3.3 Application and analysis

To evaluate the effectiveness of the developed adaptation method, the method was applied to a turbo-shaft gas turbine engine. In addition, because this method is developed based on the studied carried out by Gatto et al. [64] and Marinai et al. [65], it is necessary to compare this method with their methods to see weather multiple sets of scaling factors can lead to better adaptation results. As Marinai have already proved that his method provides better results than the work of Gatto's, in this study, in this application, Marinai's method was also applied to the turbo-shaft gas turbine for comparisons. All the adaptations in this application were carried out by a gas turbine performance and diagnostics software called PYTHIA [67], of Cranfield University which has been developed over many years. PYTHIA is able to carry out adaptations by either the developed method or the old methods (i.e. Gatto and Marinai's methods)

3.3.1 Engine model

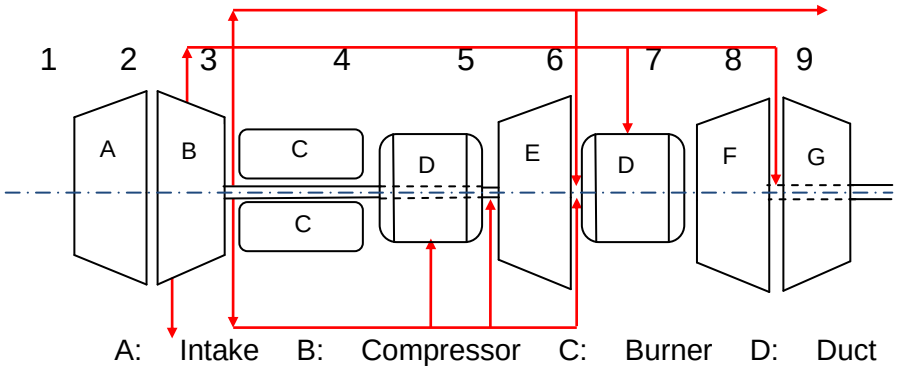

Figure 3-6 Engine model configuration

The configuration of the turbo-shaft gas turbine engine being studied is given in Figure 3-6, where the red arrows stand for flows splitting from the main flow. An engine performance model for this engine was constructed using PYTHIA. It consists of a three axial and one centrifugal stages-compressor, a reverse flow annular combustion chamber, a single stage compressor turbine and a two

stages-power turbine. Due to the confidentiality reasons, the engine characteristics is not given in this thesis.

A default compressor map is used in the engine model for performance simulation. The purpose of this application is to use the developed adaptation approach to adapt this engine model so that its predicted performance can match with target engine data. In practice, the target engine data is purposed to be real engine test data. However since the engine used in this application is a model engine and there are no available engine test data, a set of simulated engine data given by the same engine performance model but with another compressor map was employed as target engine data. In this application, the default compressor map is referred to as “inaccurate” map and the compressor map used to generate the target engine data is referred to as “accurate” map. Hence the developed adaptation approach was used to scale the “inaccurate” map so that the predicted performance given by it can match with that given by the “accurate map”. Configuration of the “inaccurate” compressor map is shown in Figure A -6-1and Figure A -6-2 in Appendix A. The “accurate” compressor map’s configuration is shown in Figure A -6-3 and Figure A -6-4 in Appendix A. The measurements shown in Table 3-2 were chosen for OD adaptations.

Table 3-2 Measurements chosen for adaptations

Measurement	Symbol	Unit
Compressor outlet total pressure	P3	atm
Compressor outlet total temperature	T3	k
Power turbine inlet pressure	P7	atm
Power turbine inlet temperature	T7	k
Power turbine outlet total pressure	P8	atm
Power turbine outlet total temperature	T8	k
Fuel flow rate	FF	kg/s
Shaft power	SP	Watt

3.3.2 Engine data simulation

Using the same engine model, two sets of engine performance data were generated at the International standard atmosphere (ISA) sea level ambient condition. The first set of data were generated using the “inaccurate” compressor map while the second set were generated using the “accurate” compressor map. The set of data resulted from the “accurate” compressor map was used as target data while the OD adaptation methods were used to scale the “inaccurate” map to provide predicted performance matching to the target data.

Table 3-3 Rotational speeds for adaptation and verification

	Case 1	Case 2	Case3
CN	new developed method		old method
1.2	**Adaptation**	**Adaptation**	**Adaptation**
1.15	Verification	Verification	Verification
1.1	Verification	Verification	Verification
1.05	Verification	Verification	Verification
1	Design point	Design point	Design point
0.97	Verification	Verification	Verification
0.95	Verification	**Adaptation**	**Adaptation**
0.93	Verification	Verification	Verification
0.9	**Adaptation**	**Adaptation**	**Adaptation**

* The point with CN =1 is the design point

Eight different CNs representing eight OD points as listed in Table 3-3 were used in this study where for each CN; measurements listed in Table 3-2 were simulated. It is important to note that throughout the study, the CN (i.e. CN and PCN are equivalent since the ambient condition is ISA sea level condition during the simulation of performance data and it is assumed that the intake is adiabatic (see Equation (3-1)) was used as the engine handle in order to have the control when the values were changed to the desired OD point.

Three cases of OD adaptation were studied, see Table 3-3. Cases 1 and 2 used the new developed method while Case 3 has the same adaptation points as Case 2 has but the adaptation was performed using Marinai’s method (old

method). Case 1 has two adaptation points (CNs with 1.2 and 0.9), while Case 2 and Case 3 have three adaptation points (CNs with 1.2, 0.95, and 0.9). Therefore In the application, in Case 1, only the two speed lines with CNs 1.2, and 0.9 were scaled while in Cases 2 and 3 all the three speed line with CNs 1.2, 0.95 and 0.9 were scaled.

The reason for having both Case 1 and 2 is to investigate how the number of adaptation points can affect the adaptation result. In Case 1, two CNs (1.2 and 0.9) were chosen for OD adaptation while another six were chosen to validate the effectiveness of this method. Meanwhile in Cases 2 and 3, only three OD points (points with CNs 1.2, 0.95 and 0.9) were chosen for OD adaptation while another five were used for the verification of the effectiveness of the adaptation. The reason that the points with CNs 1.2, 0.95 and 0.9 are chosen for adaptations is that in the “inaccurate” map it has three speed lines with CNs 1.2, 0.95, and 0.9.

3.3.3 Searching for scaling factors

Table 3-4 GA setting

GA Setting	Value
Max No. of Generations	20
Population Size	50
Probability of Crossover	0.35
Probability of Mutation	0.3
Convergence Criterion (Fitness)	0.99

In order to search for the scaling factors, in both Case 1 and Case 2, firstly, a search range for each scaling factor was set, then following the process described in Section 3.2.5, a GA was used to search for an optimized set of scaling factors (i.e. SF_{WAC}, $SF_{PR,}$ and SF_{ETA}) for each speed line selected for adaptation. The setting of maximum number of generations, population size,

probability of crossover, probability of mutation, and convergence criterion (fitness) for each searching process is shown in Table 3-4.

As shown in Table 3-4, the maximum number of generations was set to 20, however it is possible that the best set of scaling factors found within the 20 generations may not be the global optimal scaling factors since the scaling factors in the global optimal set may not be in the pre-defined search ranges. Therefore in the process of searching scaling factors for each speed line, after each 20 generations, the scaling factors in the fittest set were checked against their pre-defined search ranges. If any of the scaling factors had a value close to boundaries of its search range, the search range would be modified so that the value would lie in the middle of the new search range and another 20 generations would be carried out to search for better scaling factors. This was repeated until the value of each of the scaling factors in the fittest set found after the 20 generations lie in the middle of it search range.

In Case 3, according to Marinai's method, the GA was used to search for an identical set of scaling factors for scaling all the three speed lines (speed lines with CNs =1.2, 0.95 and 0.9) using the same setting shown in Table 3-4. This can be done by the same process described in Section 3.2.5, however the fitness of a set of scaling factors is represented by the average prediction accuracy over the three CNs resulted from the map scaled by this set of scaling factors.

3.3.4 Map scaling

Once the scaling factors required by each case have been obtained by the GA, a scaled map was then produced for each case. For Case 1, scaled map 1 was produced by using obtained optimized scaling factors to scale the speed lines with CNs 1.2 and 0.9; for Case 2 and Case 3, scaled maps 2 and 3 was produced respectively by scaling the speed lines with CNs 1.2, 0.95 and 0.9. Note that in Case 2, the three speed lines are scaled by different sets of scaling factors while in Case 3 they all scaled by the same set of scaling factors.

3.3.5 Verification

In order to evaluate the effectiveness of the adaptation methods, each of the scaled maps was used to predict the model engine's OD performance at all the CNs shown in Table 3-3. The predicted performance resulted from each of the scaled maps was then compared with the predicted performance with the "accurate" map at the CNs and discussed in the next section.

3.3.6 Simulation results and analysis

Initial prediction errors of measurements, which represent the relative differences between measurements resulted from the "inaccurate" map and the "accurate" map as shown in Equation (3-7), are shown in Figure 3-7.

$$Prediciton\ error = \frac{\left|P_{target} - P_{predicted}\right|}{P_{target}} \tag{3-7}$$

where $\boldsymbol{P_{target}}$ is a measurement simulated using the "accurate" map, and $\boldsymbol{P_{predicted}}$ is the same measurement simulated using the "inaccurate" map.

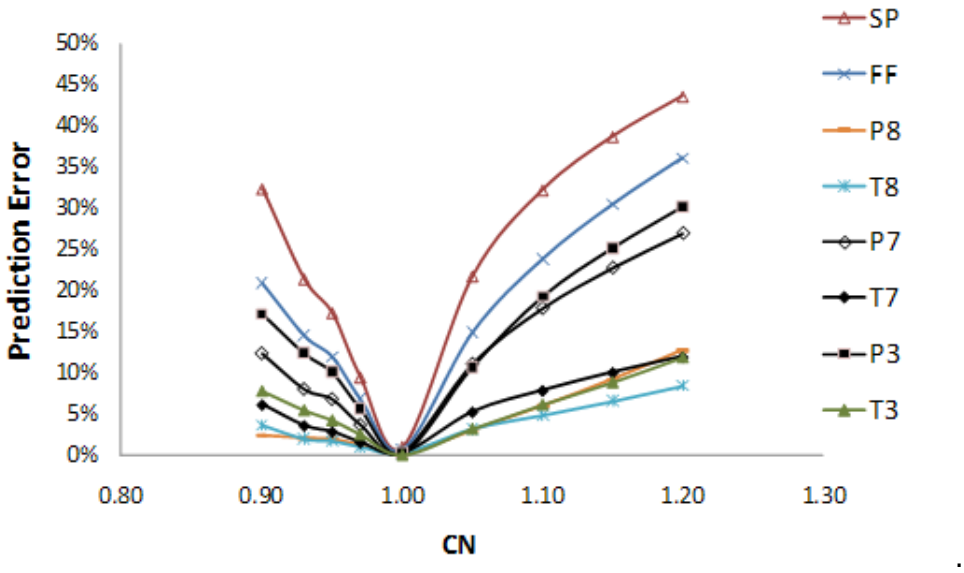

Figure 3-7 Initial prediction errors

By analyzing Figure 3-7, the following can be seen:

- The performance prediction error for each measurement becomes larger as the CN moves away from the design point (Design point selected is at CN = 1). The reason caused this phenomenon is the non-linearity of the OD performance model which induces the prediction error increase as the operating point is move away from the design point.
- The increase of CN leads to larger prediction error than the decrease of CN. This may because the similarity of between the speed lines with higher CNs from the “inaccurate” and “accurate” maps is lower than that of between the speed lines with lower CNs from the maps (higher similarity may lead to lower prediction errors).
- The prediction of SP, FF, P7 and P3 have very low accuracy, while relatively T3, T7, T8, and P8 have relatively less prediction errors. Therefore it seems that the prediction accuracy of temperature measurements is higher than most of the other measurements.
- The SP seems to be the most sensitive measurement to the variation of compressor’s characteristics as it’s the one having the largest prediction error in this figure. This is because the SP is largely influenced by mass flow rate and the change of the compressor characteristic map changes the flow capacity of the compressor and consequently affects the mass flow rate. The variation of characteristics of the compressor seems have less influence on T8 and P8, as these two measurements have lower prediction errors than the other measurements have.
- The initial prediction errors are unacceptable for diagnostics as most of prediction errors may have levels higher than measurement deviations caused by component degradation. Especially for SP and FF, they both have very large prediction errors and they also are very important measurements for diagnostics.

A graph of average of measurement prediction error at each power against CN is plotted and shown in Figure 3-8. The average measurement prediction error at a CN is the average of the prediction errors of the eight measurements at this CN.

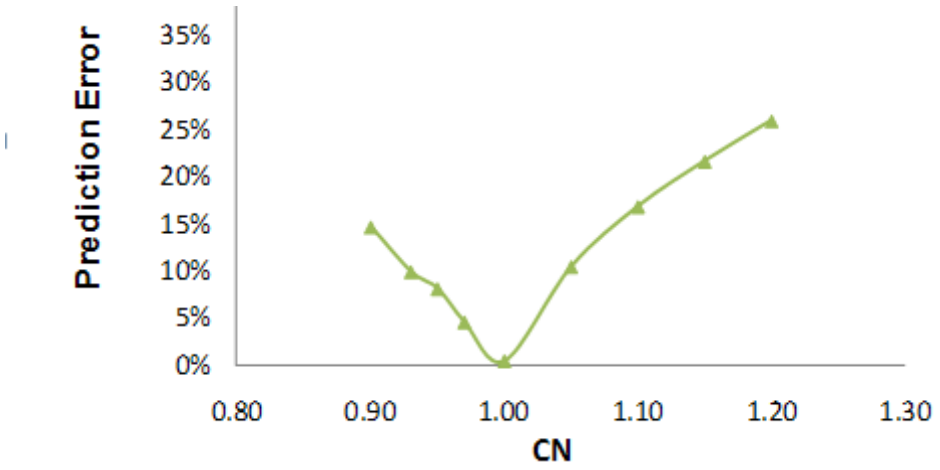

Figure 3-8 Average measurement prediction errors

Note that the average prediction error line is V shape as all the all prediction errors are absolute values. In addition, it is also worth mentioning that for design point performance simulation the compressor map is not needed then at the design point both the performance simulation using the “inaccurate” and “accurate” map lead to the same result and hence there is no prediction error at design point. The average prediction errors shown in Figure 3-8 indicate that the prediction accuracy of the engine OD performance is very poor. Hence this model is not qualified to be used for diagnostics and it is necessary to carry out OD adaptation to improve the quality of prediction.

3.3.7 Adaptation results and analysis

Table A -6-1 to Table A -6-3 shown in Appendix C list the amount of attempts made (each attempt is denoted by “TRY”) to modify the scaling factor search ranges for the speed lines with CNs 1.2, 0.95, and 0.9 respectively. In addition the tables also contain the maximum fitness obtained within 20 generations by the GA and the scaling factors (corresponding to the maximum fitness) in each attempt (in each attempt 20 generations were carried out).

Note in Case 1 and Case 2, the scaling factors obtained for the speed lines with CNs .2, and 0.9 are the same (the only different between the adaptations in Case 1 and Case 2 is that the latter scale an extra speed line which is the one with CN 0.95).

Bar charts of the maximum fitness found within each 20 generations for different speed lines in each case are shown in Figure 3-9, Figure 3-10, and Figure 3-11.

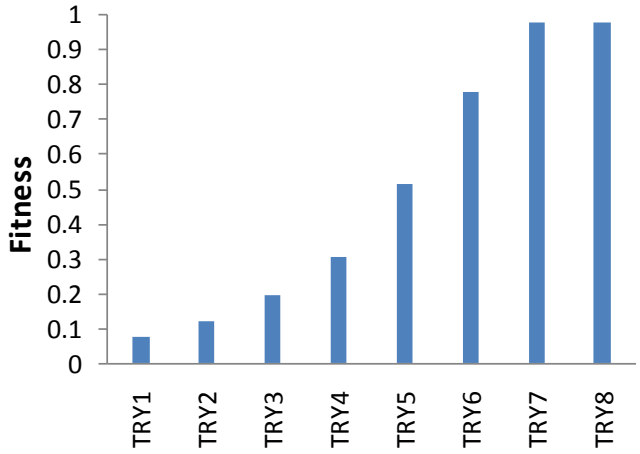

Figure 3-9 Maximum fitness found within each 20-generation for speed line with CN 0.9 in both Case 1 and Case 2

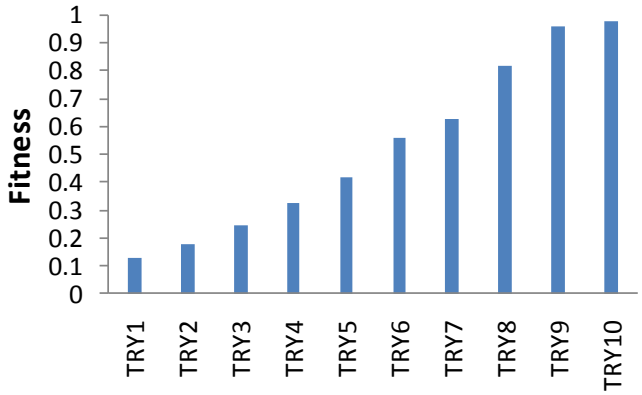

Figure 3-10 Maximum fitness found within each 20-generation for speed line with CN 0.95 in Case 2

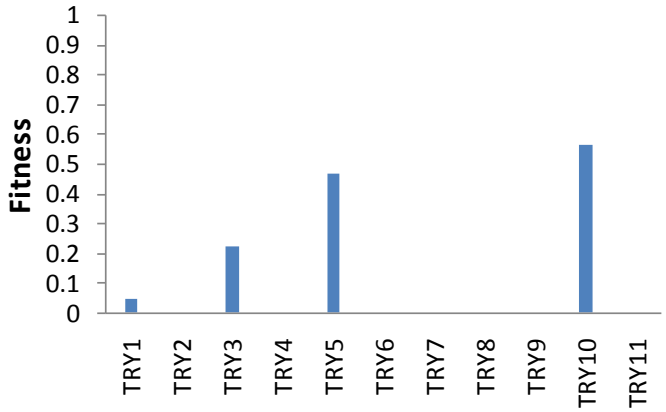

Figure 3-11 Maximum fitness found within each 20-generation for speed line with CN 1.2 in both Case 1 and Case 2

From the results shown in above Figure 3-9 to Figure 3-11, the following observations can be made:

- The attempt to adjust search ranges can largely improve the maximum fitness can be obtained within 20 generations. In the adaptations for the speed lines with CN 0.93 and 0.9, each of the attempts leads to a rise of the maximum fitness.

- The adaptations for the speed lines with CNs 0.9 and 0.95 are quite successful. Each of the adaption obtained a final maximum fitness (i.e. the highest maximum fitness obtained within the TRYs) close to 1 after several TYRs. However the adaptation for the speed line with CN 1.2 are relative less effective as the best result it obtained with in eleven TRYs is a maximum fitness 0.563. This is because the initial average prediction error at CN 1.2 is much higher than the error at CN 0.9 or 0.95 (refer to Figure 3-8) and the increase of the average initial prediction error raises the difficulty to achieve good adaptation Therefore the final maximum fitness obtained for the speed line the CN 1.2 is much low than for the other two. The absence of fitness for some TRYs in Figure 3-11 and Table A -6-3 because in these TRYs the GA code in PYTHIA crashed. The reason caused the crash of the code is due to that the operating

point may lie beyond the surge line of the scale map resulted from the scaling factors found by the GA in these tries.

In addition, the final optimized set of scaling factors for each of the three speed lines in Case 1 and Case 2 and an optimized set of scaling factors for all the three speed line obtained in Case 3 are shown in Table 3-5.

Table 3-5 Final optimal scaling factors for different speed lines

	Scaling Factor								
	Case 1			Case 2			Case 2		
CN	ETA	WAC	PR	ETA	WAC	PR	ETA	WAC	PR
1.2	0.87	1.273	1.175	0.87	1.273	1.175	0.901	1.178	1.134
0.95	x	x	x	0.951	1.088	1.059	0.901	1.178	1.134
0.9	0.908	1.145	1.126	0.908	1.145	1.126	0.901	1.178	1.134

The results shown in Table 3-5 indicate:

- The obtained final optimized scaling factors for each of the speed lines in each case shown in Table 3-5 shows that ETA of all the three speed lines in the “inaccurate” map’s needed to be scaled down (scaling factor less than 1 indicates a scale down), and in contrast WAC and PR need to be scaled up (scaling factor higher than 1 indicates a scale up) .

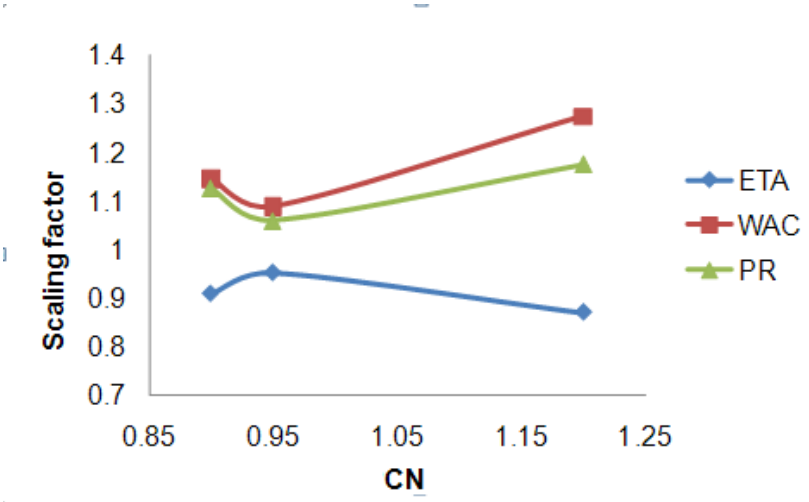

Figure 3-12 Scaling factors obtained in Case 2

- The optimal scaling factors required to minimize prediction errors at different CNs varies with CN non-linearly, this can be observed from Figure 3-12 where the scaling factors obtained in Case 2 is plotted against CN. It can be seen from Figure 3-12, all the three trend lines of the three scaling factors (ETA, WAC, and PR) are non-linear lines.

In each case, the obtained scaling factors obtained were then employed to scale the “inaccurate” map to produce a scaled map. Hence three scaled maps were generated in three cases: scaled map 1 generated by scaling two speed lines with CNs 1.2 and 0.9 in Case 1, scaled map 2 generated by scaling thee speed lines with CNs 1.2, 0.95, and 0.9 in Case 2, scaled map 3 generated by scaling thee speed lines with CNs 1.2, 0.95, and 0.9 in Case 3. The scaling of the three speed lines in Case 2 is shown in Figure A 6-5 and Figure A 6-6 in 6Appendix C.

3.3.8 Verification results and analysis

In order to evaluate the effectiveness of each of the scaled maps to improve OD performance prediction, each of them was used to predict performance at all the CNs shown in Table 3-3, and the simulation errors resulted from using the scaled maps 1 and 2 are shown in Figure 3-13 to Figure 3-20.

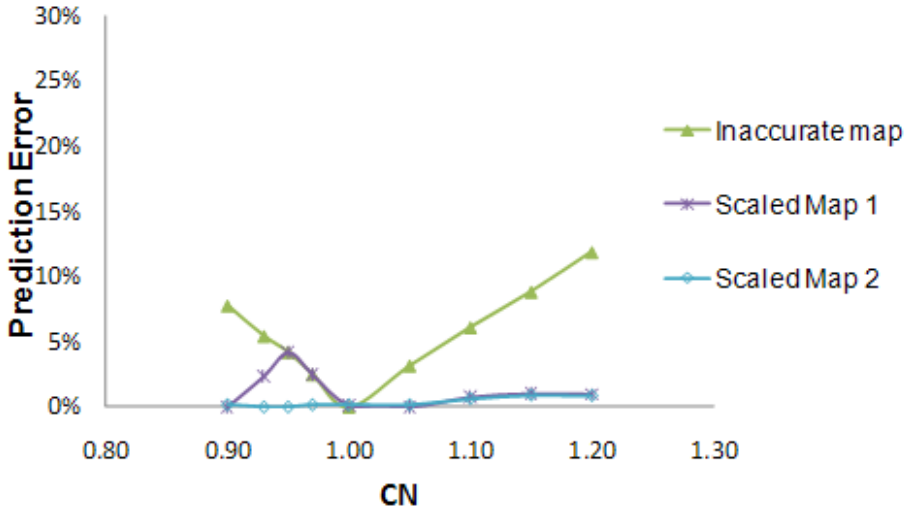

Figure 3-13 Prediction errors of T3 in Case1 and Case 2

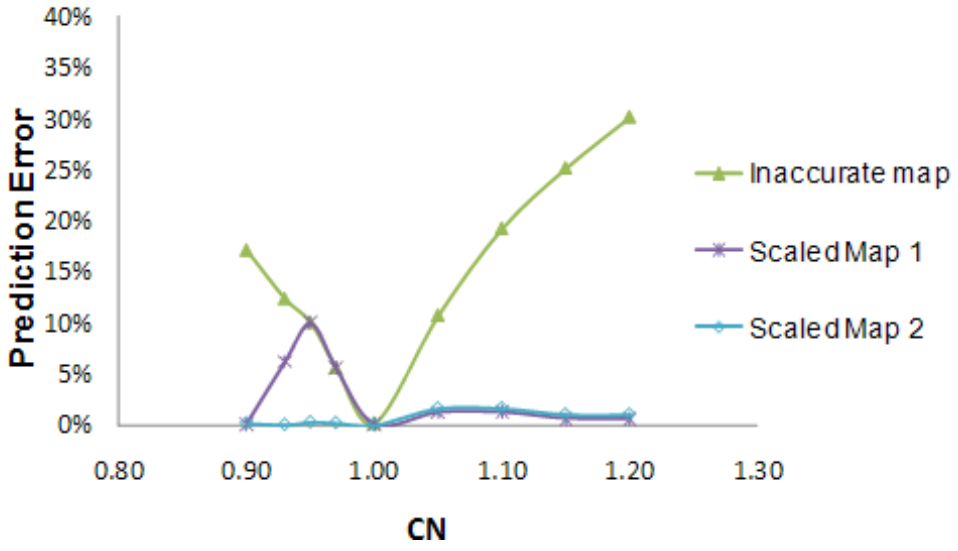

Figure 3-14 Prediction errors of P3 in Case1 and Case 2

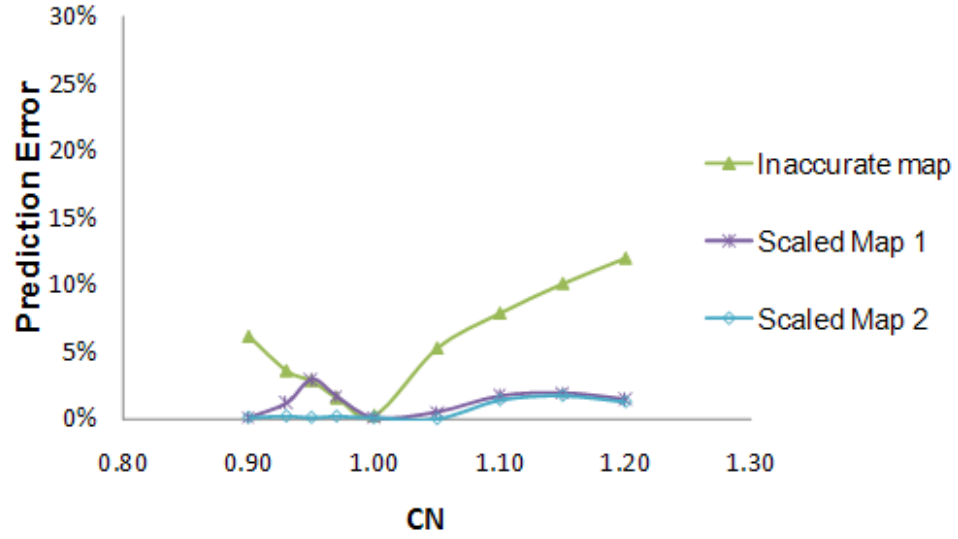

Figure 3-15 Prediction errors of T7 in Case1 and Case 2

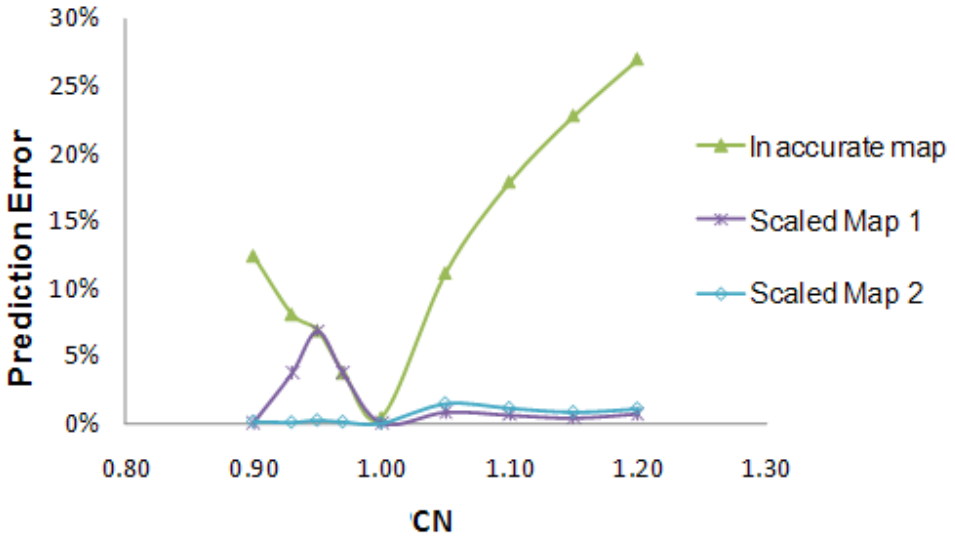

Figure 3-16 Prediction errors of P7 in Case1 and Case 2

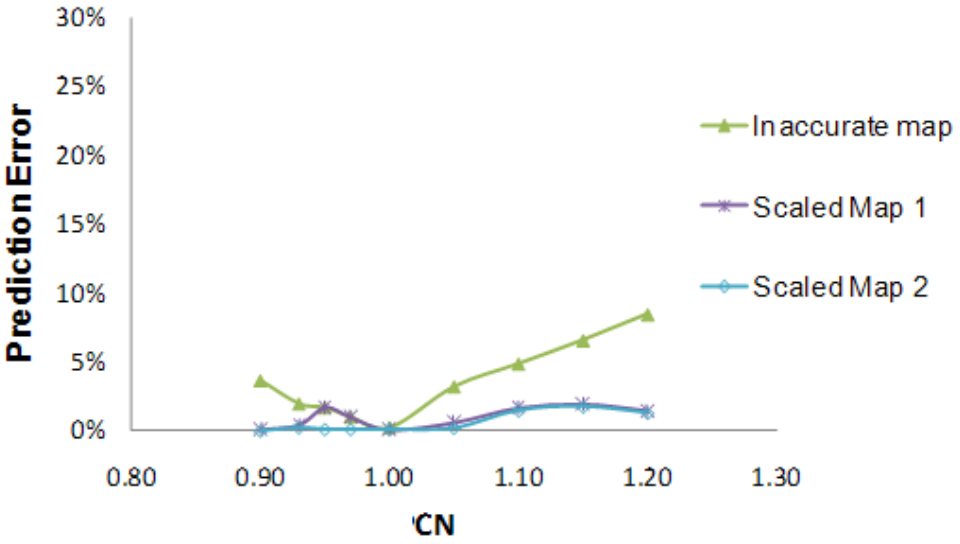

Figure 3-17 Prediction errors of T8 in Case1 and Case 2

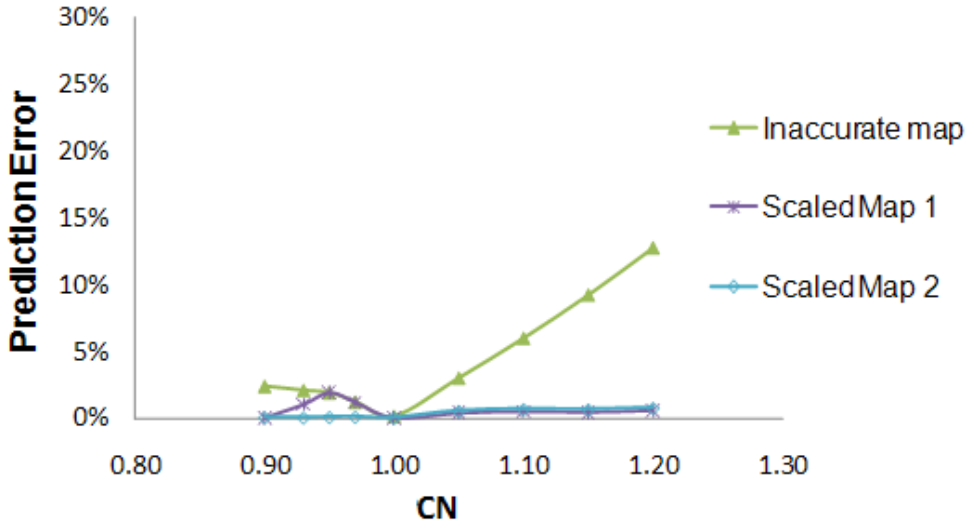

Figure 3-18 Prediction errors of P8 in Case1 and Case 2

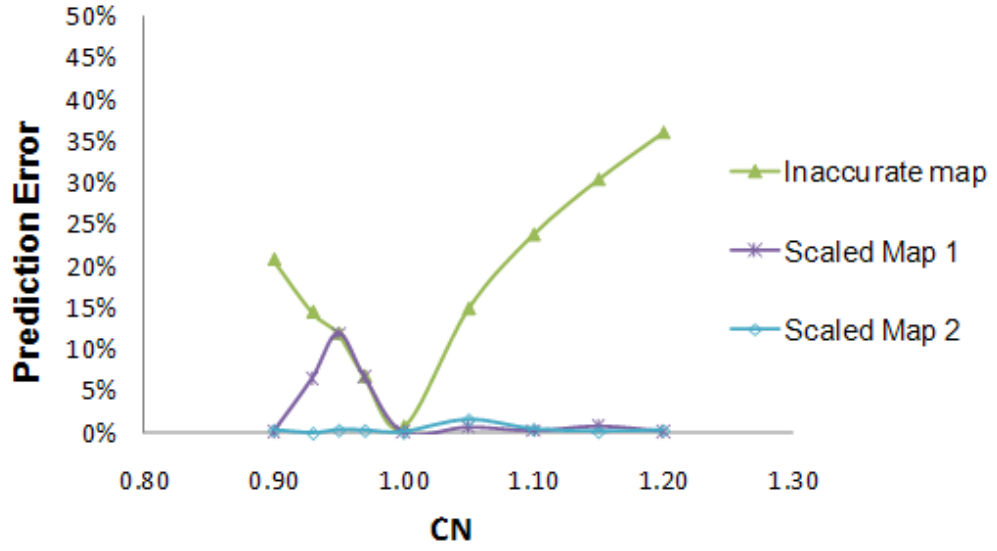

Figure 3-19 Prediction errors of FF in Case1 and Case 2

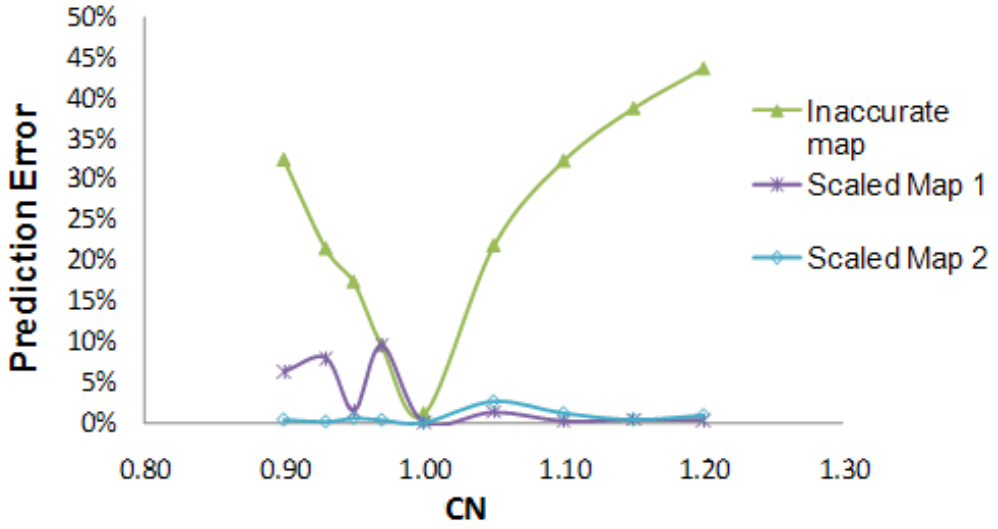

Figure 3-20 Prediction errors of SP in Case1 and Case 2

From Figure 3-13 to Figure 3-20, some observations can be made:

- At CNs 1 to 1.2 the use of either of the scaled maps (scaled maps 1 and 2) the prediction error of each measurement can be reduced significantly. Especially shaft power, its highest initial prediction error is 43.59% at CN 1.2. (refer to Figure 3-20), and this error is reduced to 0.93% in Case 1 and 0.35 % in Case 2 respectively
- The figures also show that in Case 2 prediction accuracies for all the measurements different CNs can be largely improved at CNs 1 to 0.9, while in Case 2 only the prediction errors at CNs between 0.9 and1.0 cannot be reduced to very low levels. This is because in Case 1 the speed line with CN 0.95 was not included in the adaptation and hence the prediction errors at CNs near 0.95 were very low. However since the speed line with CN 0.9 was included in the adaptation in Case 1 and CN 1.0 is the design point, Case 1 was still able to provide accurate performance prediction accuracies at CNs near 0.9 and 1.0.
- By analyzing the adaptation results from Case 1 and Case 2, it can be seen that the developed adaptation approach is very effective in improving simulation accuracy of engine performance models. In addition, the choice on the speed lines needed to be scaled is very

important. The more speed lines selected for adaptation, the better simulation accuracy can be obtained at the operating points near the selected speed lines.

Average error of each measurement over the nine CNs is calculated and shown in Table 3-6.

Table 3-6 Average error of each measurement over nine different CNs

	Average error		
	inaccurate map	scaled map 1	scaled map 2
T3	6.24%	1.49%	0.36%
P3	16.34%	3.24%	0.81%
T7	6.19%	1.43%	0.62%
P7	13.77%	2.14%	0.65%
T8	3.95%	1.11%	0.66%
P8	4.80%	0.75%	0.35%
FF	20.02%	3.44%	0.47%
SP	27.23%	3.43%	0.83%

Table 3-6 shows that the average error of each measurement can be reduced significantly using either of the scaled map 1 or 2, however the scaled map 2 is seen to be more effective than scaled map 1. This can be seen clearly as the scaled map 2 is able to reduce the average error of each measurement to lower than 1 % while in scaled map 1 only measurement P8 is seen to have a measurement prediction error of less than 1%.

Average prediction errors (i.e. average of eight measurement prediction errors) resulted from different scaled maps at each CN are shown in Table 3-7 and a graph of average prediction error against CN is plotted and shown in Figure 3-21.

Table 3-7 Average prediction errors at different CNs

	Average prediction error over all measurements			
CN	Inaccurate map	Scaled map 1	Scaled map 2	scaled map 3
1.2	25.97%	0.92%	1.08%	4.82%
1.15	21.67%	1.09%	1.07%	5.22%
1.1	16.85%	1.01%	1.22%	6.13%
1.05	10.45%	0.81%	1.18%	4.33%
1	0.00%	0.00%	0.00%	0.00%
0.97	4.56%	4.58%	0.23%	3.81%
0.95	8.11%	5.86%	0.25%	4.20%
0.93	9.91%	4.22%	0.11%	5.40%
0.9	14.66%	0.98%	0.21%	4.70%

Figure 3-21 provides useful information for evaluating the effectiveness of the scaled maps in improving accuracy OD performance prediction. By analyzing this figure the following can be observed:

- It can be seen from this figure that scaled map 2 is able to reduce prediction errors at all the eight different CNs significantly, while scaled map 1 only able to improve the prediction accuracy at some of the CNs and it leads to high prediction error at CNs near 0.95 (as shown in Table 3-6, the top three highest average prediction errors of scaled map 1 are 4.58% at CN 0.97, 5.96 % at CN 0.95, and 4.22% at CN 0.93). This again proves that the absence of speed line 0.95 in the adaptation Case 2 results in the low prediction accuracy at CNs near 0.95 and hence the choice and the number of speed lines need to be adapted is very importance. Therefore for an engine model needs to be adapted at a certain CN range, all the speed lines within the range are needed to be scaled.

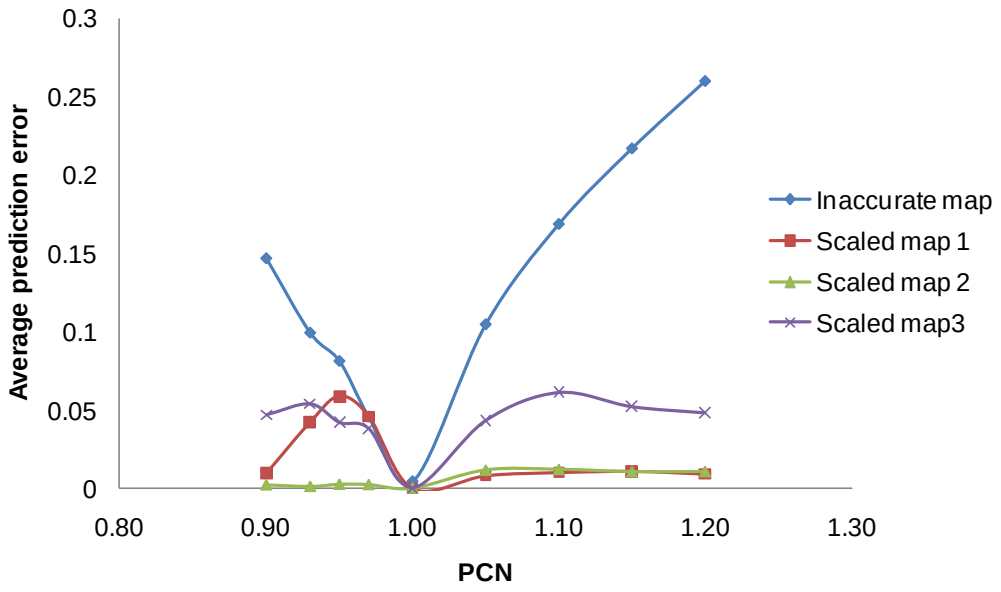

Figure 3-21 Average prediction errors resulted from different maps

- Apart from that, Figure 3-21 also shows that the prediction accuracy of simulation using scaled map 2 is not only very high but also quite stable over a range of CNs unlike that of inaccurate map which changes dramatically with the variation of CN.
- It also can be observed from Figure 3-21 that compared with the scale map 3 the scaled map 2 is able to provide more accuracy performance prediction at all the eight CNs included in this application. This indicates that with the same adaptation points included in the adaptation the new developed method is capable of providing better adaptation results. The reason for this can be found in Table 3-5 which indicates that the scaling factors required to minimize performance prediction error vary with different CNs and it is necessary to scale the component map non-linearly, hence using an identical set of scaling factors to scaling different speed lines cannot maximally reduce the prediction error at different CNs.
- In addition, the results in Figure 3-21 show that using the new developed method, an “inaccurate” map, which is originally total unacceptable for gas path diagnostics of the model engine, can be scaled to a map which provides good performance prediction. In this study, the highest and lowest prediction errors resulted from the an “inaccurate” map at the

eight CNs are 25.97% and 4.56%, and the new developed managed to reduce these two errors to 1.22 % and 0.11 % respectively (see the prediction errors of scaled map2 in Table 3-7)

3.4 Chapter conclusions:

In this research, a new gas turbine performance model adaptation using multiple non-linear scaling factors is presented with the intention of improving the prediction accuracy of gas turbine engine performance models so that they can be used for diagnostics. This is implemented by using different optimised sets of scaling factors to scale different speed lines on a component map that needed to be adapted. A GA is utilised to search for an optimal set of scaling factors for each speed line that needs to be scaled. Application of the developed non-linear off-design adaptation approach to a model single spool turbo-shaft aero-engine has proved that:

- This new adaptation method has great potential to improve the off-design prediction accuracy of engine performance models.
- It is capable of searching for a set of optimal scaling factors for the speed lines needed to be scaled in a component characteristic map and scaling component map non-linearly and effectively.
- The GA is very effective in finding optimal solutions of the scaling factors.

In addition, limitations of the new method have also been found:

- It can be time-consuming when the number of speed lines that need to be scaled is high as the adaptation for each speed line needs to be carried out individually.
- In order to scale a compressor map using the new method, engine test data at different off-design conditions are needed. The absence of necessary data to cover different speed lines could lead to low prediction accuracy at certain off-design conditions.

Based on the advantages and the limitations of the developed off-design adaptation approach listed in this section, comparisons between this approach and those which are also able to carry out off-design adaptation can be made:

All of the off-design adaptation approaches mentioned in Section 2.2 are only able to scale component characteristic maps linearly, but as mentioned earlier in this section the developed approach can carry out non-linearly map scaling and hence match the non-linearity of the gas turbine off-design performance model. However the developed method may require more engine test data for adaptation than the other approaches do.

4 Rough set based fault isolation

4.1 Introduction

Economic, availability and safety concerns have led to a rapid growth in the application of Engine Health Monitoring (EHM) in recent years. Gas path diagnostic techniques for the isolation of component faults based on gas path fault signatures, are important techniques in condition monitoring. An effective fault isolation approach should be able to solve most the problems listed in Section 1.4.

In the past half-century, many model-based gas path diagnostic techniques, such as linear gas path analysis, non-linear gas path analysis, Kalman filter, genetic algorithm and weighted least squares, have been developed by employing gas turbine performance models in the diagnostic process. Since 1980 there has been a rapid growth in interest in the development of non-model based diagnostic approaches where no complex performance models are required in the diagnostic process. This has led to the successful development of neural networks, rule based fuzzy logic, and expert systems based gas path diagnostic techniques. All these techniques have been introduced in Section 2.1.

It is also mentioned in Section 2.1.6 that in the most recent development rough set is introduced for fault isolation. Rough set is a mathematical approach in dealing with vagueness, imprecision and uncertainty. It can be used for pattern extraction, attribute selection, data reduction, and decision rule generation. It has been applied widely in many different fields, such as machine learning, artificial strategic decisions, pattern recognition, and knowledge discovery. Therefore rough set has the potential to be a good tool for gas turbine fault isolation.

In addition, conventional fault signatures are represented by measurement deviations due to gas turbine component faults. As the complexity of potential faults increases, similarity of the conventional fault signatures caused by different faults may increase, which results in difficulties in distinguishing the

faults. Thus, enhanced fault signatures, derived from the conventional fault signatures and including both the measurement deviations and the ranking patterns of the measurement deviations in terms of magnitude, are developed and integrated with the rough set based fault isolation approach to make this approach more robust.

Therefore in this research, a new rough set based gas turbine fault isolation approach was created. Three different versions of the new rough set based approach were developed to make it suitable for fault isolations using conventional fault signatures, using enhanced fault signatures, and only limited measurements respectively. In addition, a study of creating an effective fault isolation framework for this developed rough set based approach was also carried out. This was done by studying three different fault isolation frameworks.

All the three versions of the developed approach integrated with one fault isolation framework were applied to a model two-spool turbofan gas turbine engine for the isolation of single and dual faults under different levels of measurement noises in order to test the effectiveness of the approach. In addition, in order to study the other two frameworks, one of the versions was integrated with the other two frameworks separately and also tested by applying it to the isolation of single and dual faults of the same model engine under different levels of measurement noises. Based on the study of the three frameworks a new framework was created for the developed fault isolation approach.

4.2 Methodologies

Rough set theory is introduced in Section 4.2.1 along with the process of using rough set for machine diagnostics. Three different versions of the developed rough set based fault isolation approach are presented in 4.2: the first one using conventional fault signatures is described in Section 4.2.2, the second one using enhanced fault signatures is presented in Section 4.2.3, and the last one presented in Section 4.2.4 is suitable in the circumstances when only limited measurements are available. Section 4.2.5 illustrates the study being carried

out to create an effective fault isolation framework for the developed isolation system.

4.2.1 Rough set theory

As mentioned in Section 2.1.6, rough set establishes a sound basis for knowledge discovery in datasets, and can be used to reduce redundancy of attributes and generate rules from datasets for machine diagnostics. The following concepts are important to understand how rough set can be used to approximate a vague concept reduce attributes in a dataset, and generate rules from that dataset.

4.2.1.1 Information system

In rough set theory, the dataset that needs to be analysed is usually referred to as an information system and represented by *IS: IS* = (*U, At*), where *U* is the set of objects in the dataset and *At* is a set of attributes used to describe the characteristics of the objects (e.g. in this study, objects can be gas turbine fault samples, and attributes of each fault sample can be the fault type and measurement deviations corresponding to the fault sample). The value of an attribute *f* (*f* $\in$ *At*) of an object *w* (*w* $\in$ *U*) is denoted by *w(f)*.

4.2.1.2 Discretization

Due to the fact that rough set is not able to deal with continuous data, discretization needs to be carried out to discrete attribute values of objects in the information system before the system can be analysed by rough set. An example of discretization is illustrated in Figure 4-1, where v_1 and v_2 are two threshold values, and d_1, d_2 and d_3 are discrete values.

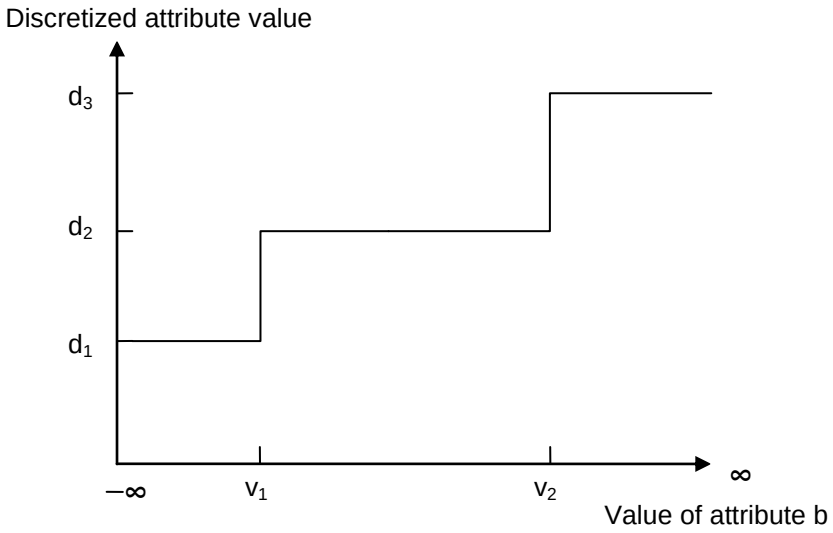

Figure 4-1 An example of discretization

In this example, it is assumed that the discretization for the value of an attribute *b* follows the following three rules listed below:

If the original value of attribute $f < v_1$ then the discrete value = d_1

If the original value of attribute $f < v_2$ and $> v_1$ then the discrete value = d_2

If the original value of attribute $f > v_2$ then the discrete value = d_3

It can be seen from Figure 4-1 that before discretization the value of attribute *f* can be any value from infinity to finity, but after discretization it can only be d_1, d_2, or d_3. These three discrete values actually represent intervals of the value domain of attribute *f* (i.e. d_1, d_2, or d_3 represent intervals $(-\infty, v_1]$, $(v_1, v_1]$, $(v_1, \infty]$, of the value domain of attribute *f* respectively). The choice of discretization method is very important as it can directly affect the characteristics of objects in the information system. Well-established discretization methods have been described by Øhrn [68], [69]; they include: manual discretization, equal interval width, equal frequency interval, entropy algorithm, naïve algorithm, semi-naïve algorithm, Boolean reasoning and rough set based algorithm, and so on. Basically, these algorithms can be classified into two groups; one which needs users to provide preliminary information (i.e. discretization rules shown earlier in the section) to carry out discretization such as the first three algorithms mentioned earlier in this paragraph, and the other which does not require any

preliminary information and is able to implement discretization based on the information given in the decision table such as the last four algorithms also mentioned earlier in this paragraph.

4.2.1.3 Indiscernibility

In an information system, if two objects *w* and *y* (*m, y* $\in$ *U*) have the same value of an attribute *f* (i.e. w(f) = y(f)) or the same discretized value if discretization has been carried out to the attribute value, then they are equivalent with respect to attribute *f*, and they are *f*-indiscernible. All objects, which have the same discrete value of the attribute *f* as the object *w* has, are equivalent to the object *w* with respect to attribute *f*, and they can be represented by a equivalence-class set denoted by $[w]_f$. For an attribute set *F* (*F* $\subseteq$ *A*), $[w]_F$ denotes the set of objects which are equivalence to object *w* with respect to the attribute set *F*, and is expressed in Equation (4-1):

$$[w]_F = \{y \in U \mid y(f) = w(f)\ for\ each\ f \in F\} \tag{4-1}$$

For individual object *w*, its equivalence-class set $[w]_F$, with respect to a certain attribute set *F* provides the information that which objects in the decision table are equivalent to the object *w*. The smaller the size of this equivalence-class set, the easier it is to distinguish this object from the rest of the objects using the attribute set *F*.

4.2.1.4 Lower & upper approximations and boundary region

As far as a group of objects is concerned, say a set of objects *W* (*W*$\subseteq$*U*), equivalence-class sets can be used to calculate lower and upper approximations of this set with respect to a certain attribute set. These approximations provide the information about how effective the objects in this group can be distinguished from the rest in the information system by using a certain attribute set.

As expressed in Equation (4-2), the lower approximation of a set *W* (denoted by $F\_(W)$) with respect to an attribute set *F* consists of those objects in *W* which definitely can be distinguished from the rest of the objects in the information system using attribute set *F*. In other words, the objects in the lower approximation are those that definitely can be classified into the set *W* using the attribute set *F*. The way to identify whether an object *w* in the universe of objects *U* (in an information system) belongs to the set *W'* s low approximation follows two steps: firstly, calculate the equivalence-class set with respect to attribute set *F* for the object *w*; secondly, check if all the objects in this equivalence-class set are in the set *W*, and if so, this object should be classified into the lower approximation of the set *W* with respect to attribute *F*.

$$F\_(W) = \{w \mid w \in U \; and \; [w]_F \subseteq W\} \tag{4-2}$$

As expressed in Equation (4-3), the upper approximation of a set *W* with respect to attribute set *F* denoted by $F^{-}(W)$ contains those objects in the object universe *U* which possibly can be classified into the set *W* using attribute set *F*. For an object *w* from the universe of objects, if any of the objects in the equivalence-class set of this object *w* is in the set *W*, then object *w* will be classified into the upper approximation of set *W*. It is worth noting that the lower approximation of set *W* is actually a subset of the upper approximation.

$$F^{-}(W) = \{w \mid w \in U \; and \; [w]_F \cap W \neq \emptyset\} \tag{4-3}$$

The tuple ($F\_(W)$, $F^{-}(W)$) composed of the lower and upper approximation is called a rough set.

The difference between the lower and upper approximation sets of a set *W* with respect to a certain attribute set *F* is called the boundary region, as expressed in Equation (4-4), which consists of those objects having equivalence-class sets that do not belong to the set *W* but have intersection with it.

$$\textit{Boundary Region of approximation of } W = F^{-}(W) - F\_(W) \tag{4-4}$$

The effectiveness of using an attribute set *F* to discern objects in set *W* from the rest of the objects in the same information system can be evaluated by a so-called accuracy of the approximation which is denoted by $\alpha_F(W)$ and expressed as:

$$\alpha_F(W) = \frac{|F\_(W)|}{|F^-(W)|} \quad (4\text{-}5)$$

Where $|F\_(W)|$ and $|F^-(W)|$ represent the cardinalities of $F\_(W)$ and $F^-(W)$ respectively.

4.2.1.5 Reduct

Rough set can be used for attribute reduction (to analyse all attributes and remove unnecessary ones*)*. Assuming it is desired to distinguish a set of *W* (*W* ⊂ *U*) from the others in the universe *U*, for any subset *F* of the original attribute set *At* (*F* ⊆ *At*), if $F\_(W) = At\_(W)$ and $F *\_ (W) \neq At\_(W)$ (for any subset *F** ⊂ *F*), then the attribute set *F* can be defined as a reduct of the original attribute set *A*. Such a reduct has the same capability to distinguish the objects in the set *W* from other objects in the information system as the original attribute set *A* has, thus the rest of the attribute in *A* will not be needed. It is possible to have more than one reduct for distinguishing the set *W*, and the intersection of all reducts is called the core. The core is the set of attributes (or a single attribute) which is the most important in order to distinguish the object set *W* from the rest in the object universe *U.*

4.2.1.6 Decision table

An information system usually can also be represented by a so-called decision table. For diagnostic purposes, the decision table is more appropriate to generate decision rules for diagnostic purposes. Such a decision table contains all the objects and their corresponding attributes and attribute values, in the dataset. In a decision table, attributes are usually divided into two types: the condition attribute and the decision attribute. The major difference between them is that the latter is the most important characteristics of the object and is

actually used by users of the decision table to distinguish the objects (e.g. in this study, fault type should be the decision attribute as the purpose of this study is to identify the fault type when gas turbine faults occur, and measurements or measurement deviations can be the condition attributes). The decision table then can be expressed as $T=(U,C\cup D)$, where C is a set of condition attributes and D is a decision attribute (although the decision attribute can be more one - in this study, the number of decision attribute is always one). An example of a decision table is illustrated in Table 4-1, where fault samples are considered as objects, gas path measurements (i.e. temperatures, pressures, power output, fuel flow rate and so on) as condition attributes, measurement deviation values as condition attribute values, and fault type as a decision attribute.

Table 4-1 A gas turbine degradation decision table

Fault Sample (objects)	Condition attribute (C)				Decision attribute (D)
	P_1	P_2	…	P_L	Fault type
S1	*-1.1%*	*3.1%*	…	*-0.9%*	$Fault_1$
S2	*2.3%*	*-0.2%*	…	*4.7%*	$Fault_1$
S3	*1.3%*	*5.6%*	…	*2.5%*	$Fault_1$
S4	*-0.4%*	*1.4%*	…	*3.1%*	$Fault_2$
…	…	…	…	…	…
S_M	*-2.1%*	*2.3%*	…	*-3.6%*	$Fault_O$

* S_1, S_2,… S_M are fault sample indices, P_1, P_2, …, P_L are measurements, $Fault_1$, $Fault_2$, …, and $Fault_O$ represent different fault types, and subscripts *M, L* and *O* are the total number of fault samples, total number of measurements and total number of fault types respectively.

4.2.1.7 Discretization for decision table

Similar to attribute values in an information system, condition attribute values in a decision table also need to be discretized in order to analyse the decision table by rough set; however, it may not be necessary to discretize decision

attribute value (e.g. in Table 4-1, the decision attribute is fault type and does not have a numerical attribute value; hence there is no needed to carry out discretization). For example, using a simple manual discretization method to discretize condition attribute values in Table 4-1, a discretized table as shown in Table 4-2 can be obtained. It is assumed that this method follows two discretization rules:

1. If the original value of an attribute < 0 then the discrete value = *0*
2. If the original value of an attribute >= 0 then the discrete value = *1*

Table 4-2 A discretized decision table

Fault Sample (objects)	Condition attribute (C)				Decision attribute (D)
	P_1	P_2	…	P_L	Fault type
S1	*0*	*1*	…	*0*	*$Fault_1$*
S2	*1*	*0*	…	*1*	*$Fault_1$*
S3	*1*	*1*	…	*1*	*$Fault_1$*
S4	*0*	*1*	…	*1*	*$Fault_2$*
…	…	…	…	…	…
S_M	*0*	*1*	…	*0*	*$Fault_O$*

4.2.1.8 Attribute reduction for decision table

As far as a decision table being discretized is concerned, all the fault samples (objects) can be classified into groups by the fault type (the decision attribute), and this is the way the users of the decision table want to classify the objects (e.g. in this study, the actual way to classify fault samples (objects) is to use their fault types (decision attribute values)). The measurements (the condition attributes) can also be used to classify the fault samples into groups. If the groups resulting from the measurements are the same as (or very similar to) the groups obtained by the fault type, it means that it is possible to use the measurements to identify the fault type of a new fault sample with known measurement deviation values and unknown fault type. It is important to note

that this is possible only if the fault type of the new fault sample is covered by the decision table.

The concepts of lower and upper approximation concepts can now be utilised to evaluate how well the fault sample groups resulting from the measurements can match the groups given by the fault type (i.e. how well the measurements can be used to distinguish different fault types in the decision table).

If, in a decision table, T (T=(U,C∪D)), assuming all the fault samples in U can be divided into several groups denoted by W^D (W^D represents the set of those groups) by their fault types (the decision attribute *values*), say W^D_1, W^D_2, W^D_3,… (W_1, W_2, W^D_3, … ⊂ U). In addition, assuming that they also can be classified into several groups denoted by W^C by the measurements (the condition attribute set C), say W^C_1, W^C_2, W^C_3, … (W^C_1, W^C_2, W^C_3, … ⊂ U), then fault samples in the same group have the same measurement deviations.

If any group from W^C belongs to one of the groups in W^D; then all the fault samples in this group form W^C will be classified into the lower approximation $C\_(W^D)$ of the set of groups W^D with respect to the measurements (C). If any group from W^C is interactive with one of the groups in W'; then all the objects in this group form W^C will be classified into the upper approximation $C^-(W^D)$ of the set of groups W^D with respect to the measurements (C). In other words, fault samples in the lower approximation of W^D with respect to the measurements (C) are those which can be definitely classified into the correct fault types using the measurements; while objects in the upper approximation are those which possibly can be classified into the correct fault types using the measurements (C).

If a measurement subset F of the original measurement set C, ($F \subseteq C$), leads to $F\_(W^D) = C\_(W^D)$ and $F^*\_(W^D) \neq C\_(W^D)$ (for any measurement subset $F^* \subset F$), then F is a reduct of C. Once a reduct of the original measurements has been found, unnecessary measurements (condition attributes) can then be removed from the decision table to reduce the redundancy in the decision table.

This is actually how rough set can be used to select appropriate measurements (i.e. measurements in the reduct) for fault isolation.

Once a reduct of condition attributes (measurements) can be found, measurements not included in the reduct can then be removed from the discretized decision table to create a discretized and reduced decision table. Taking the attributes in Table 4-2 as an example, assuming that P_2 and P_L are not included in the reduct, then a discretized and reduced table as shown in Table 4-3 can be formed by removing these two attributes from the table.

Table 4-3 A discretized and reduced decision table

Fault Sample (objects)	Condition attribute (C)				Decision attribute (D)
	P_1	P_3	…	P_{L-1}	Fault type
S1	*0*	*1*	…	*1*	*Fault$_1$*
S2	*1*	*1*	…	*1*	*Fault$_1$*
S3	*1*	*0*	…	*1*	*Fault$_1$*
S4	*0*	*1*	…	*1*	*Fault$_2$*
…	…	…	…	…	…
S_M	*0*	*1*	…	*0*	*Fault$_O$*

4.2.1.9 Decision rules

One of the common purposes of the implementation of rough set is to generate decision rules from a decision table. The decision rules can be used to identify decision values of new objects (e.g. identify the fault type of a new fault sample) with known condition attribute values (e.g. measurement deviation values). Rule generation usually should be carried out after discretization and attribute reduction of the decision table, as, compared with the original decision table a discretized and reduced decision table contains more simple and useful information. Each rule contains a conditional part “if …” and a decision part “then…” in the form as “if … then …” or “if…→…”, where the conditional and decision parts are the descriptions of condition attribute(s) and decision attribute

respectively of the same object in a discretized and reduced decision table. More detail of rule generation will be described in 4.2.2.4.

4.2.1.10 Reasoning

Once rules have been generated from a discretized and reduced decision table, the only problem left is how to use the rules to identify the decision attribute values of new objects. This is usually done by using a reasoning method. This method decides which rules should be used for each new object and how to solve conflicts if different rules lead to different solutions for the same new object (e.g. rules gives different decision attribute values for the same new object).

4.2.1.11 Rough set in gas turbine fault isolation

As mentioned in Section 2.1.6. Chen et al. studied the possibility of using rough set for fault isolation. The process of fault isolation is based on Chen et al.'s method and contains 5 major steps as shown in Figure 4-2. Chen et al. applied their method for the single fault isolation of a twin spool turbofan.

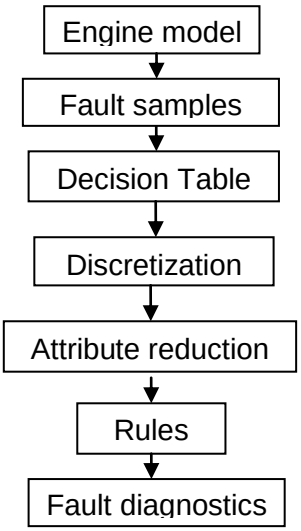

Figure 4-2 A fault isolation method based on rough set [36]

Step 1 Decision table construction

Generate single fault samples using an engine performance model to build a decision table for the engine, which considers fault samples as objects, measurements as condition attributes, measurement deviations caused by faults as condition attribute values and the fault type as a decision attribute, an example of such a decision table has already been illustrated in Table 4-1.

Step 2 Discretization

In this method, the discretization algorithm [37], is used to split the domain of the value of each condition attribute into two intervals; therefore two discrete values, are used to replace the original values of the condition attributes. This algorithm is similar to the manual discretization method used for Table 4-1 which also split the domain of each value into two intervals by discretization rules defined by the author; however, for the former the discretization rules are automatically provided by the algorithm.

Step 3 Attribute reduction

An attribute reduction method based on a GA is then used to calculate a reduct of the measurements (condition attributes) to select appropriate measurements for fault isolation (measurements in the reduct). In this method, each possible solution (a subset of the attributes) to the attribute reduction problem is considered to be a string (refer to the description of GA in Section 2.1.12), and by following a searching process similar to the one described in Section 2.1.12, the fittest string (i.e. an optimal reduct) can be found. The fitness of each string is calculated by evaluating the number of measurements included in the string and the capability of the measurements to distinguish fault samples in decision table. More detail of this method can be found in [38].Once a reduct has been found, measurements not included in the reduct can then be removed from the discretized decision table and a discretized and reduced table can be formed.

Step 3 Rule generation

In Chen et al.'s method, rules are generated from the discretized and reduced table, and used for single fault isolation (the reasoning algorithm they used in their study is not described in their publication[36] regarding this method).

4.2.2 Fault isolation based on rough set using conventional fault signatures

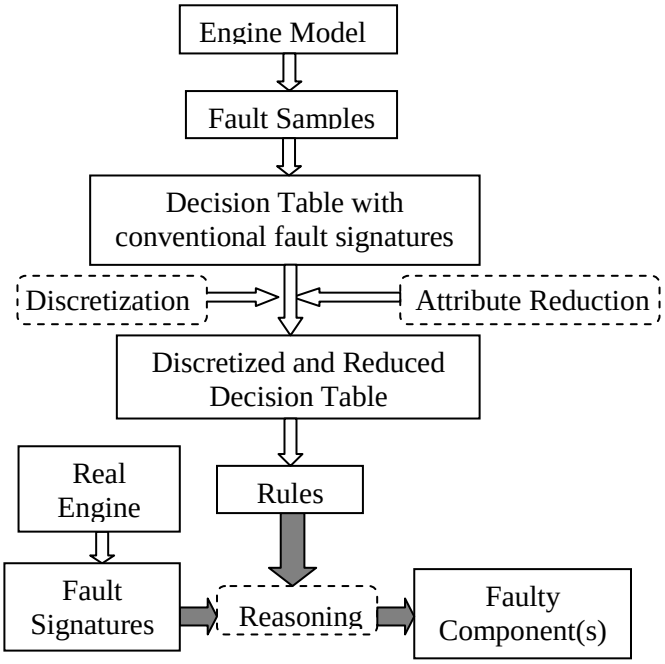

Figure 4-3 Flow chart of the developed rough set based fault isolation approach using conventional fault signatures

The main objective of this study is to develop a rough set based gas turbine diagnostic approach where knowledge, which is hidden in fault samples, is discovered and transferred into rules that can be used to isolate gas turbine faults. In this section a new rough set based fault isolation approach using conventional fault signatures (i.e. each signatures is represented by the deviation of a measurement caused by the gas turbine component fault) is presented. The novelty of this approach is that the way to carried out discretization and attribute reduction is totally different to that of in Chen et al.'s method described in Section 4.2.1.11. The generation and application of such a rough set diagnostic approach for a gas turbine engine is illustrated in Figure 4-3 and it contains steps 1 - 5:

1. Build a performance model for the engine and generate sufficient fault samples from an engine performance model for all potential faults. Each fault sample is represented by conventional fault signatures caused by the fault (i.e. measurement deviations) and the fault type.
2. Represent the generated fault samples in a decision table according to rough set theory. Such a decision table considers fault samples as objects, conventional fault signatures as condition attributes (Unlike described in Section 4.2.1.6, measurement deviations are used as condition attributes instead of measurements, however measurement deviations are equivalence to measurements as each measurement produces one measurement deviation) and fault type as a decision attribute.
3. Discretization and attribute reduction are then carried out simultaneously (unlike the process shown in Figure 4-2 in Section 4.2.1.11 which carries out discretization and attribute reduction separately) to form a *discretized and reduced decision table.* Here the attribute reduction is utilized to select appropriate measurements for fault isolation.
4. Generate rules for fault isolation from the *discretized and reduced decision table.*
5. Once real fault signatures have obtained from real faulty engine, the generated rules can be used for identifying the faulty component by a reasoning algorithm.

The details of above five steps are explained in Sections 4.2.2.1, 4.2.2.2, 4.2.2.3, 4.2.2.4, and 4.2.2.5 respectively.

4.2.2.1 Gas turbine fault samples

It is unrealistic in practice to generate all fault samples for a gas turbine using engine tests. Therefore, gas turbine thermodynamic performance models may be used to produce fault samples that cover all potential faults including different types and different levels of degradation. Each of the samples contains information of the fault type and corresponding fault signatures. The more samples provided, the more knowledge may be discovered and consequently

better rules may be generated. Each fault sample may be represented in the following form:

Sample i (S_i) (i=1 to M): $\Delta P_k = v_{ik}$ (k=1 to N), $Fault_h$ (h =1, 2,… or O)

where S_i represents the i^{th} fault Sample, M is the total number of samples, ΔP_k is the deviation of the k^{th} measurement P_k, v_{ik} is the value of ΔP_k, N is the total number of measurements, $Fault_h$ is the h^{th} type of fault, and O is the total number of different fault types.

4.2.2.2 Decision table

Once sufficient fault samples are obtained, they need to be represented in a decision table acceptable by rough set. In rough set theory, a decision table denoted by T may be represented as $T=(U, C \cup D)$, where U is the universe of objects, and C and D denote condition and decision attributes respectively. The attributes are used to represent the characteristics of the objects, and each object has its own attribute values. In this approach, fault samples are considered as the objects, conventional fault signatures (measurement deviations) as the condition attributes C, measurement deviation values as condition attribute values, and fault type as the decision attribute D. For example, an example of gas turbine fault decision table is shown in Table 4-4, which includes four fault samples two measurement deviations and two types of fault. Note that although some of the samples have the same fault, they have different levels of degradation.

Table 4-4 An example of gas turbine fault decision table

Fault Sample	Condition attributes (C)		Decision attribute (D)
	ΔP_1(%)	ΔP_2(%)	Fault type
S1	v_{11}	v_{12}	$Fault_1$
S2	v_{21}	v_{22}	$Fault_1$
S3	v_{31}	v_{32}	$Fault_2$
S4	v_{41}	v_{42}	$Fault_2$

where ΔP_1 and ΔP_2 are two measurement deviations, v_{11} …. v_{41} and v_{12} … v_{42} are measurement deviation values, $Fault_1$ and $Fault_2$ are faulty component's names.

4.2.2.3 Discretization and attribute reduction

After the construction of the decision table, discretization needs to be carried out to discrete condition attribute values in the decision table since rough set is only able to deal with discrete data. Apart from that, in order to select appropriate measurements for fault isolation using rough set, attribute reduction needs to be carried out. Attribute reduction is a process to select an informative condition attribute subset called a reduct from the decision table being discretized. In a more technical term, the reduct is a representative subset of the original condition attributes in a decision table, which is able to be used to discern the same objects that can be distinguished by using the original condition attributes. If each condition attribute in the decision table is represented by a measurement deviation, then the measurements involved in the reduct can be selected as appropriate measurements for isolation of those faults included in the decision table.

In this study, an algorithm [70] which is based on the combination of rough set and Boolean reasoning is employed for both discretization and attribute reduction. This algorithm is able to carry out discretization while simultaneously calculate a reduct for the original condition attributes in a decision table for the attributes reduction. The main reason this algorithm is chosen in this research due to the unique feature it has: it is able to carry out discretization with minimal information loss (the downside of discretization is that it induces information loss to the decision table).

Taking the decision table shown in Table 4-4 as an example, the process of carrying out discretization and finding a reduct using this algorithm follows the three steps below:

Step 1- Initial discretization

In order to demonstrate the discretization process clearly, each fault sample in the decision table is schematically denoted by a point in *N*-dimensional space, where *N* is the total number of the measurement deviations (i.e. the total number of measurements) in the decision table. Assuming $v_{41} < v_{11} < v_{31} < v_{21}$ and $v_{32} < v_{22} < v_{42} < v_{12}$, the four samples in Table 4-4 now can be denoted by four points in a two-dimensional space as shown in Figure 4-4.

To carry out discretization, the axis of each measurement deviation in the space is divided into intervals by cut(s). A cut is denoted by $C(c_i , m_{ip})$, where c_i denotes that this is a cut of the i^{th} measurement deviation, m_{ip} is the position of this cut and its subscript *ip* denotes this is the p^{th} cut of the i^{th} measurement deviation; it divides the axis of the measurement deviation c_i into two intervals $(-\infty, m_{ip}]$, and (m_{ip}, ∞), and the position the cut should be taken (i.e. the value of m_{ip}) is defined by an average of two adjacent measurement deviation values in the space. All fault samples, which have values of a condition attribute falling into the same interval of this condition attribute, will be assigned an identical discrete value d_{ig} to replace the actual values, where d_{ig} denotes the g^{th} interval of the i^{th} measurement deviation. For example, in Figure 4-4, ΔP_1 is divided into two intervals by taking a cut at the point where the value of ΔP_1 is the average of v_{41} and v_{11}, and the cut is denoted by $C(\Delta P_1, m_{11})$, where m_{11} is the average of v_{41} and v_{11}. Therefore two different discrete values will be assigned to the four samples to replace their actual values of ΔP_1, say discrete values d_{11} and d_{12} as shown in Figure 4-4, where in this case they represent intervals $(-\infty, m_{11}]$ and (m_{11}, ∞) of the axis of the measurement deviation ΔP_1 respectively.

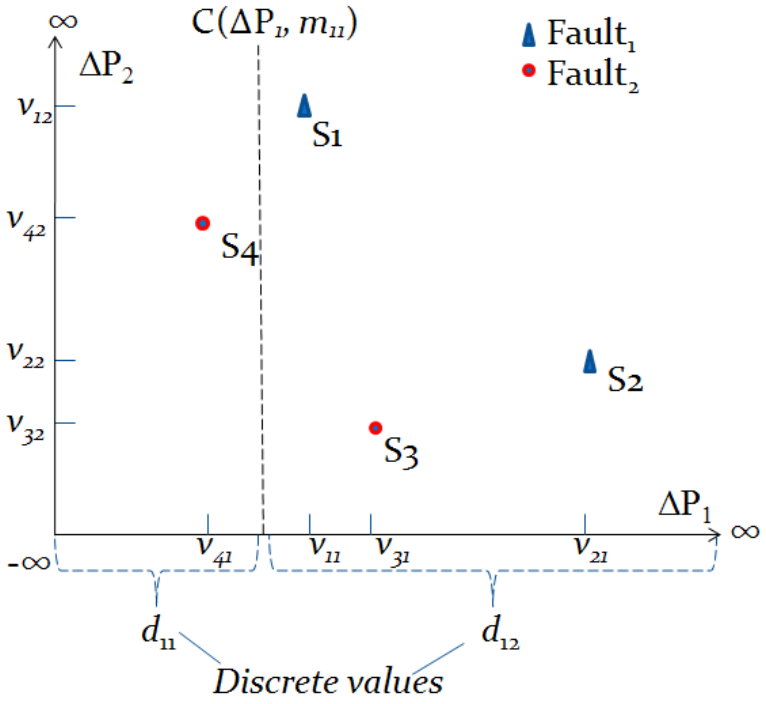

Figure 4-4 Faults in a two-dimension space

In order to replace the actual measurement deviation values with discrete values (i.e. carry out discretization), while ensuring that all the faults in the decision table can still be distinguished by these discrete values, an set of initial cuts of measurement deviations is formed by taking a cut in the middle of each two adjacent values on each measurement deviation axis in the space. For example, as shown in Figure 4-5, the set of initial cuts of ΔP_1 and ΔP_2 in Table 4-4 should be: {C(ΔP_1 , m_{11}), C(ΔP_1 , m_{12}), C(ΔP_1 , m_{13}), C(ΔP_2 , m_{21}), C(ΔP_2 , m_{22}), (ΔP_2 , m_{23})}, where m_{11}, m_{12},…, m_{23} are the averages of two adjacent condition values. According to the set of initial cuts, new discrete values will be assigned to replace the original values as illustrated in Figure 4-5, where d_{11}, d_{12},…, d_{24} are new discrete values, and they represent different intervals of measurement deviation axes, for example, now d_{11} represents interval ($-\infty$, m_{11}] of ΔP_1 axis, and d_{22} represents interval (m_{21}, m_{22}] of ΔP_2 axis.

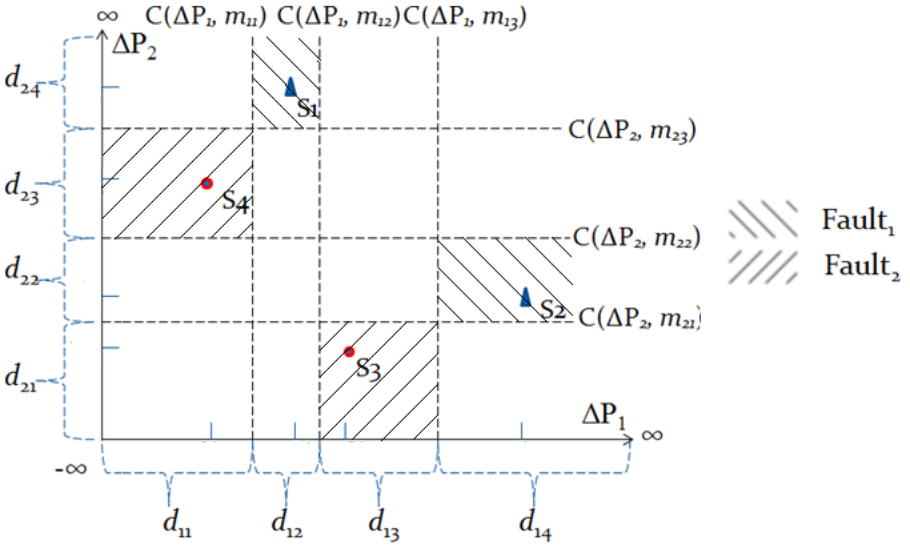

Figure 4-5 Initial discretization

By comparing Figure 4-4 with Figure 4-5, it can be seen that after dividing the space by cuts each fault sample can provide much more useful information (before discretization each fault sample only represent a single point in the space and after discretization a area).

Step 2-Cut selection

The next step is to select appropriate cuts from the set of initial cuts to form a set of final cuts since not all the initial cuts are necessary to distinguish all the faults in the decision table. To do so, a cut selection table needs to be constructed. In the cut selection table, all the initial cuts are listed in a row, and all possible pairs of the fault samples that have different types of faults are listed in a column. The set of final cuts should be able to distinguish all faults covered in the decision table with a minimum number of cuts (i.e. be able to distinguish the fault samples in each pair listed in the cut selection table).

In the cut selection table, if two samples in a pair can be distinguished using the discrete values resulting from a cut, the word "*Yes*" will be placed in the intersection cell between the column of this cut and the row of this pair; otherwise the word "*No*" will be placed in the intersection cell. For example, for fault samples shown in Table 4-4, a cut selection table is constructed and

shown in Table 4-5. Taking the cut $C(\Delta P_1, m_{11})$ as an example, it can be observed in Figure 4-4 that it leads to samples S_1, S_2 and S_3 having the same discrete value. Therefore, the fault samples in pair (S_1, S_3) or (S_2, S_3) are not distinguishable with respect to the discrete value given by the cut $C(\Delta P_1, m_{11})$, as a result of this, the intersection cells between these two pairs and the cut should be marked by "*No*".

Table 4-5 Cut selection table

	Initial cuts of measurement deviations					
Pair of fault samples	$C(\Delta P_1, m_{11})$	$C(\Delta P_1, m_{12})$	$C(\Delta P_1, m_{13})$	$C(\Delta P_2, m_{21})$	$C(\Delta P_2, m_{22})$	$C(\Delta P_2, m_{23})$
(S_1,S_3)	*No*	*Yes*	*No*	*Yes*	*Yes*	*Yes*
(S_1,S_4)	*Yes*	*No*	*No*	*No*	*No*	*Yes*
(S_2,S_3)	*No*	*No*	*Yes*	*Yes*	*No*	*No*
(S_2,S_4)	*Yes*	*Yes*	*Yes*	*No*	*Yes*	*No*

After the construction of the cut selection table, the three steps shown below need to be taken to find a set of final cuts:

1. In the cut selection table, find a cut corresponding to the column with the largest number of "*Yes*" (i.e. a cut that is able to distinguish the largest number of fault samples in the decision table).
2. Remove the cut and the pairs of fault samples that can be distinguished by this cut from the cut selection table.
3. Repeat the above two steps for the remaining cuts and pairs of fault samples until all the pairs have been removed, and the set of final cuts is the one with all the cuts have been removed.

Following the above steps, a set of final cuts can be obtained in Table 4-5 is {(ΔP_1 , m_{13}), (ΔP_2 , m_{23})}. It is worth mentioning that, in this case, there is more than one solution to the set of final cuts.

Step 3- Final discretization and attribute reduction

Once a set of final cuts has been found, those measurement deviations that are not involved in the final cuts will be removed from the decision, and discretization based on the final cuts will be carried out to the remaining measurement deviations to form a discretized and reduced decision table. Taking Figure 4-5 as an example, after the set of final cuts {(ΔP_1, m_{13}), (ΔP_2, m_{23})} has been found, the two-dimensional space can be re-divided as shown in Figure 4-6.

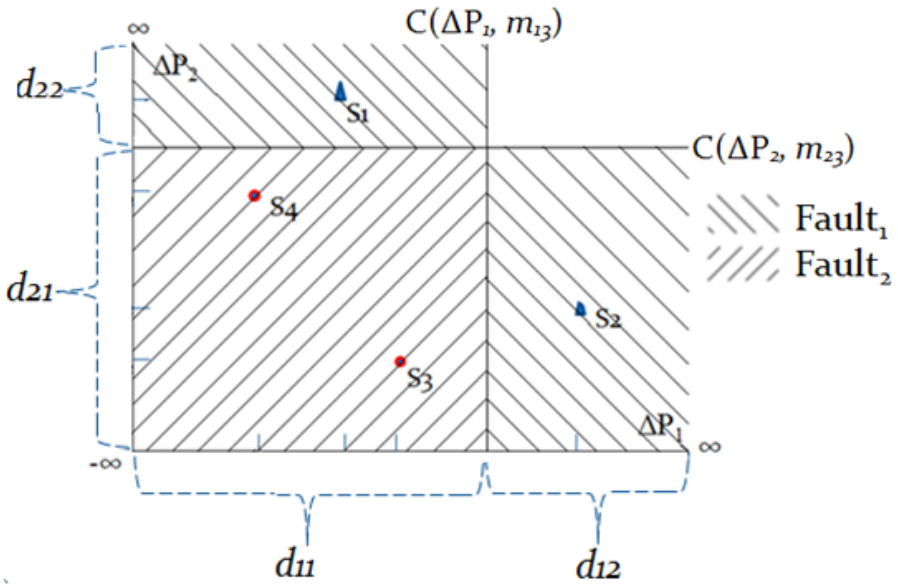

Figure 4-6 Final discretization

The reduct of the original condition attributes is the set of condition attributes included in the final cuts. table. In this case, the reduct includes ΔP_1 and ΔP_2. Notice that the condition attributes are the same as the initial ones although it is possible to distinguish the fault samples with only one attribute (by taking three cuts on axis of ΔP_1 the fault samples still can be discerned and ΔP_1 will be the only attribute in the reduct). The reason for this is because that this algorithm

tends to use as fewer cuts as possible to achieve discretization, and in this case using both condition attributes obviously leads to fewer cuts (only two cuts). Measurements included in the reduct are selected for classifying the faults in the decision table, since each condition attribute is represented by a measurement deviation.

A discretized and reduced decision table that only contains the condition attributes involved in the final cuts and discrete values given by the final cuts can be constructed for Table 4-4 and is shown in Table 4-6.

Table 4-6 A discretized and reduced decision table

Fault Sample	Condition attribute		Decision attribute
	ΔP_1 (%)	ΔP_2 (%)	Fault Type
S1	d_{11}	d_{22}	*Fault*$_1$
S2	d_{12}	d_{21}	*Fault*$_1$
S3	d_{11}	d_{21}	*Fault*$_2$
S4	d_{11}	d_{21}	*Fault*$_2$

*Note that now d_{11} and d_{12} represent intervals $(-\infty, m_{13}]$ and (m_{13}, ∞) of the axis of ΔP_1 in the space respectively, and d_{21} and d_{22} represent intervals $(-\infty, m_{23}]$ and (m_{23}, ∞) of the axis of ΔP_2 respectively.

4.2.2.4 Rule generation

Any decision table being discretized and reduced can be used to generate decision rules which represent the logical relationships between measurement deviations (the condition attributes) and the fault type (the decision table) in the decision table. Each rule contains a conditional part “if …” and a decision part “then…” in the form of “if … then …” or “if…→…”, where the conditional and decision parts are the descriptions of measurement deviation(s) and the fault type of the same fault sample in the decision table respectively. In this study, the principle to generating rules from any decision table is the same [71]: each rule is formed by matching a fault sample against its measurement deviations

and fault type respectively in the table, however such a rule only contains a minimal subset of the measurement deviations which is sufficient to be used to distinguish this fault sample from those having different faults. It is also likely to generate more than one rule from a fault sample. Apart from that, identical rules may be generated from different fault samples, and hence to evaluate the importance of each generate rule a parameter called "*Support"* will be calculated for each rule. The "*Support"* of each rule equals the number of fault samples in the decision table matching the rule.

Taking Table 4-6 as an example, by the rule generation method described earlier in this section three rules can be generated as follows:

Rule1 from sample 1: If $\Delta P_2 = d_{22} \rightarrow \text{Fault}_1$ (*Support* = 1)

Rule2 from sample 2: If $\Delta P_1 = d_{12} \rightarrow \text{Fault}_1$ (*Support* = 1).

Rule3 from sample 3 or 4: If $\Delta P_1 = d_{11}$ and $\Delta P_2 = d_{21} \rightarrow \text{Fault}_2$ (*Support* = 2)

For fault sample *S1*, a minimal subset to distinguish *S1* from *S3* and *S4* (*S1* has different fault type to the fault *S3* and *S4* have) would only be ΔP_2 although ΔP_1 is also available. This is because the value of ΔP_2 of S1 is different to both the values of *S3* and *S4*. This is similar to the rule generated from sample *S2* where ΔP_1 is used as the minimal subset. Both rules from sample *S1* and *S2* has a "*Support"* equalling to one, as there are no other fault samples matching the rules. On the other hand, both samples *S3* and *S4* have the same rule thus the "*Support"* of this rule is two. In addition the minimal subset for both *S3* and *S4* are the same (ΔP_1 and ΔP_2) because both ΔP_1 and ΔP_2 are needed to distinguish each of *S3 and S4* from both *S1* and *S2*.

4.2.2.5 Reasoning

The generated rules then can be utilized for classifying component faults of the real engine by a reasoning algorithm called Standard Voting [71]. It carries out reasoning by three steps:

Firstly, once the real engine degrades and the fault signatures caused by the degradation are obtained, the Standard Voting algorithm will first compare the obtained fault signatures with the rules generated. The algorithm will then select the rules with correct matches between the conditional part of the rules and the signatures for reasoning. For example, taking the rules (i.e. Rule1, Rule2 and Rule3) presented in Section 4.2.2.4 as an example, assuming that a fault occurs in the real engine and causes deviations ΔP_1 and ΔP_2 (i.e. fault signatures) to the measurements P_1 and P_2, where ΔP_1 and ΔP_2 have values *5%* and *-3%* respectively, if *5%* is in the interval d_{12} of ΔP_1 (i.e. if m_{13} >*5%* since d_{11} represent interval (m_{13}, ∞), refer to Table 4-6) then the fault signatures is matched by the Rule2, similarly if *-3%* is in the interval d_{22} (i.e. if m_{23}> or =*-3%* since d_{22} represent interval (-∞,m_{23}]) then the fault signatures is matched by the Rule1, and if *5%* is in the interval d_{12} and *-3%* is in the interval d_{21} then the fault signatures is matched by the Rule3. The faults described in the decision parts of the selected rules are potential faults which may cause the degradation to the real engine. For example, if the Rule2 is selected, then $Fault_1$ is a potential fault.

Next, a certain number of "*votes*", representing the probability of each potential fault, will be assigned to each potential fault based on the number of selected rules supporting (having the potential fault in its decision part) the potential fault and the importance ("*Support"*) of the rules. If a potential fault is supported by a selected rule, then from this rule, this fault can obtain the number of "*votes*" equalling to the "*Support*" of this rule.

Finally, after the calculation of total "*votes*" that each potential fault obtained from the selected rules, the total "votes" of each potential fault will be normalized by dividing the number of total "votes" of this potential fault by the number of total "votes" that all potential faults obtained. The real fault will be classified as the potential fault with the highest normalized total "*votes*".

Basically, the fault isolation process based on this version of the developed approach follows the five steps described in previous five sections respectively The application of this version of the developed rough set based fault isolation approach presented in this section will be described in Section 4.3.1.

4.2.3 Fault isolation based on rough set using enhanced fault signatures

In order to make the developed fault isolation approach more effective, enhanced fault signatures are developed to enhance the characteristics of faults and hence increase the discernibility of faults and consequently improve fault isolation accuracy. More details of enhanced fault signatures will be presented in Section 4.2.3.1. The process of fault isolation based on the developed rough set approach using enhanced fault signatures is shown Figure 4-7.

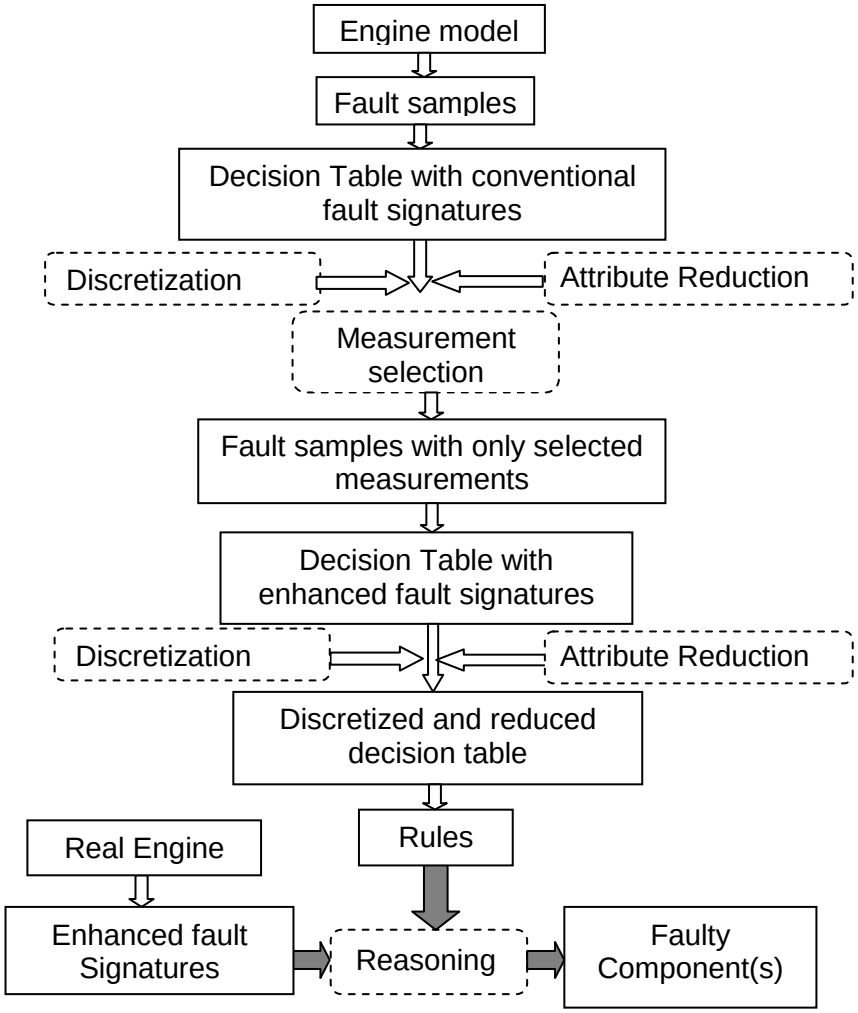

Figure 4-7 Flow chart of the developed rough set based fault isolation approach using enhanced fault signatures

As shown in the figure above, basically this flow chart consists of three major steps:

1. Measurement selection: this process is exactly the same as the one described in Section 4.2.2: firstly, generate fault samples to cover potential faults, secondly, use the samples to build a decision table with conventional fault signatures which consider fault samples as objects and measurement deviations as condition attributes and the fault type as the decision table, and finally carry out discretization and attribute reduction to calculate a reduct to of measurement deviations and select measurements involved in the reduct for fault isolation. However now the decision table constructed with conventional fault signatures is only used for measurement selection but not for rule generation.

2. Rule generation: another decision table, which is similar to the decision table with conventional fault signatures except that it uses enhanced fault signatures derived from the select measurements as condition attributes, will be built. After carrying out discretization and attribute reduction to this new decision table (note the attribute reduction is utilized to select appropriate enhanced fault signatures for fault isolation), rules will be generated from this table by the same method mentioned in 4.2.2.4.

3. Reasoning: the reasoning process is exactly the same as the one described in Section 4.2.2.5.

Actually, the major different between using conventional and enhanced fault signatures for rough set based fault isolation is that the latter uses a extra decision table with enhanced fault signatures for rule generation. More details about the step 2 (i.e. rule generation using enhanced fault signatures) will be presented in Section 4.2.3.3).

4.2.3.1 Enhanced fault signatures

A measurement deviation caused by a gas turbine component fault, is conventionally called a fault signature. Thus the number of condition attributes

is consistent with the number of measurements if conventional fault signatures are used as condition attributes in decision tables. Due to the limited number of measurements available, the rough set based fault isolation approach may not be able to isolate engine faults effectively when engines are complicated in configuration. This is because when the complexity and number of faults needed to dealt with increase, the similarity of conventional fault signatures caused by different faults may increase as well, which results in rise of the difficulty to distinguish different faults and consequently low fault isolation accuracy.

To make the developed fault isolation approach more robust in dealing with complexity faults, a concept of enhanced fault signatures is introduced. Enhanced fault signatures are developed by the author with the intention to enhance the characteristics of faults, which may make it easier to distinguish them, and consequently improve the possibility of successful fault isolation. The enhanced fault signatures, which include both the measurement deviations and the information of the ranking pattern of the same measurement deviations in terms of magnitude, are derived from conventional fault signatures. For instance, if a component fault induces conventional fault signatures (i.e. measurement deviations) ΔP_1, ΔP_2 and ΔP_3, where $\Delta P_1 > \Delta P_2 > \Delta P_3$, then the corresponding enhanced fault signatures should include both the measurement deviations ΔP_1, ΔP_2 and ΔP_3 and also the information representing the ranking pattern of these deviations, i.e. $\Delta P_1 > \Delta P_2 > \Delta P_3$.

The introduction of such a ranking pattern is due to the consideration that such a ranking pattern of some measurement deviations may be consistent or similar for the same faults with different levels of degradations but different for other types of faults. Therefore such a ranking pattern may provide extra useful information to distinguish the faults, and the extra information can be analyzed by rough set and transferred to extra rules for fault isolation. In order to convert the information of the ranking pattern into condition attributes that rough set can deal with, a so-called ranking parameter denoted by $R(P_x P_y)$ is created and

used. It represents the difference between two measurement deviations as expressed in Equation (4-6).

$$R(P_xP_y)= \Delta P_x-\Delta P_y \tag{4-6}$$

where both P_x and P_y denote measurements, and ΔP_x and ΔP_y are deviations of them respectively. Each ranking parameter's value is able to reflect the relationship between two measurement deviations in terms of magnitude; hence sufficient ranking parameters are able to represent a ranking pattern of all measurement deviations. The number of ranking parameters required to represent a ranking pattern of *N* measurement deviations can be calculated using a combination equation (see Equation (4-7)). Taking the ranking pattern "$\Delta P_1 > \Delta P_2 > \Delta P_3$" as an example, three ranking-parameters are needed: $R(P_1P_2)$, $R(P_1 P_3)$ and $R(P_2P_3)$.

$$No.of\ ranking\ parameters = \frac{N!}{(N-2)!\,2!} \tag{4-7}$$

4.2.3.2 Measurement selection

As shown in Figure 4-7, Fault isolation based on rough set using enhanced fault signatures follows the same measurement selection process described in Section 4.2.2 : build a decision table using conventional fault signatures (refer to Section 4.2.2.2), then use the algorithm based on the combination of rough set and Boolean to calculate a reduct (refer to Section 4.2.2.3), and measurements involved in the reduct are selected for fault isolation.

4.2.3.3 Rule generation using enhanced fault signatures

Once measurement selection is finished, the selected measurements will then be used to derive enhanced fault signatures (i.e. to calculate measurement deviations and ranking parameters based on the selected measurements) for each fault sample.

Both the measurement deviations and the ranking parameters then can be used to form a new decision table which is the same to the one with conventional fault signatures (i.e. Table 4-4) but using both the measurement deviations and ranking parameters as condition attributes. Taking the decision table with conventional fault signatures shown in Table 4-4 as an example, two measurements *P1*, *P2* were selected as appropriate measurements for fault isolation (refer to Section 4.2.2.3), then now enhanced fault signatures based on the two measurements can be derived which contains measurement deviations ΔP_1 and ΔP_2, and a ranking parameter $R(P_1P_2)$. Using the derived enhanced fault signatures and the same fault samples included in Table 4-4, another decision table can be constructed as shown in Table 4-7.

Table 4-7 Decision table using enhanced fault signatures

Fault Sample	Condition attribute			Decision attribute
	$\Delta P_1(\%)$	$\Delta P_2(\%)$	$R(P_1P_2)$	Fault type
S1	v_{11}	v_{12}	v_{11}-v_{12}	$Fault_1$
S2	v_{21}	v_{22}	v_{21}-v_{22}	$Fault_1$
S3	v_{31}	v_{32}	v_{31}-v_{32}	$Fault_2$
S4	v_{41}	v_{42}	v_{41}-v_{42}	$Fault_2$

After the construction of such a decision table, the table will then be discretized and reduced using the same discretization and attribute reduction algorithm described in Section 4.2.2.3, and this will result in a new discretized and reduced decision table. Note that here the attribute reduction is utilized to select appropriate enhanced fault signatures for fault isolation as not all the enhanced fault signatures are useful. By applying the same principle presented in 4.2.2.4, fault isolation rules can be generated from this new discretized and reduced decision table.

4.2.3.4 Reasoning

By the same reasoning algorithm mentioned in 4.2.2.5, rules generated from the discretized and reduced decision table with enhanced fault signatures can be utilized for fault isolation. It worth mentioning that here the fault signatures obtained from the real engine need to be enhanced fault signatures which can be simply derived from measurement deviations.

The application of this version of the developed rough set based fault isolation approach (using enhanced fault signatures) will be described in Section 4.3.1.

4.2.4 Fault isolation based on rough set with limited measurements

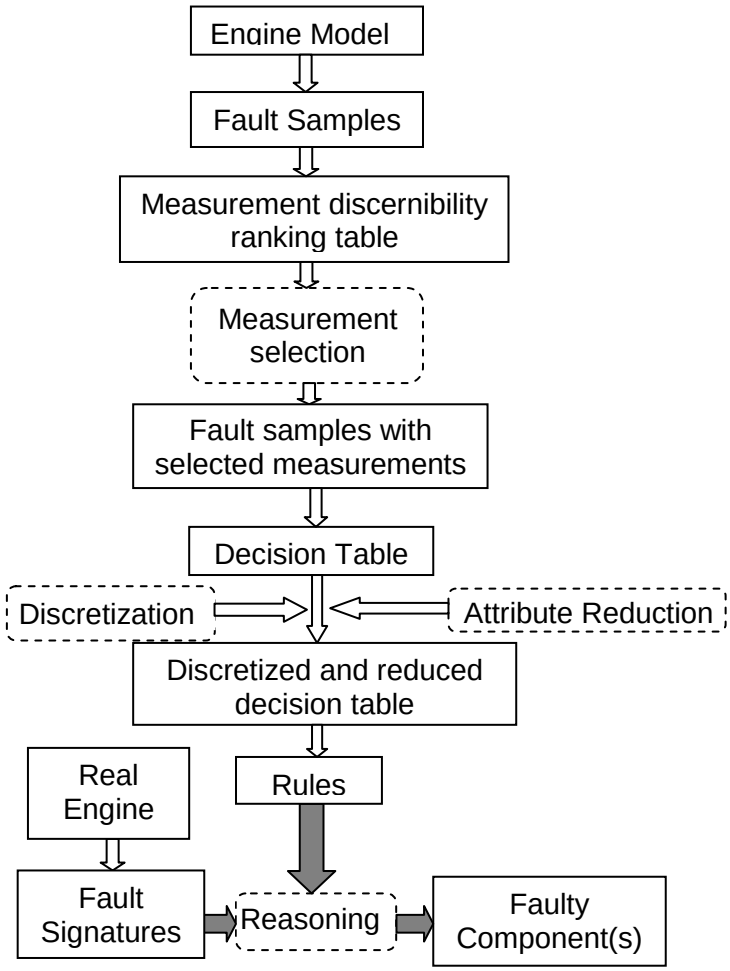

Figure 4-8 Flow chart of the developed rough set based fault isolation approach with limited measurements

One of the common between the first two versions of the developed fault isolation approach (i.e. the one using conventional and the other one using enhanced fault signatures, as presented in Sections 4.2.2 and 4.2.3) is that they select an appropriate set of measurements for fault isolation from all potential measurements that can be used for fault isolation. However if the appropriate set given by them contains too many measurements then it may be unacceptable to engine users as the higher number of sensors may lead to a higher cost. Therefore, in order to make the developed fault isolation approach more flexible in measurement selection, a new measurement selection approach was proposed and integrated with the developed fault isolation approach to form another version, which is able to provide an appropriate set of measurements according to the number of measurements that users want to use for isolation. This new measurement selection method is suitable for either fault isolation using conventional fault signatures or using enhanced fault signatures. The flow chart of fault isolation based on this new version is illustrated in Figure 4-8.

From Figure 4-8, it can be observed that this flow chart also contains three major steps: measurement selection, rule generation, and reasoning. These steps will be described in the following three sections respectively.

4.2.4.1 Measurement selection

In order to carry out measurement selection under the circumstance when the number of measurements can be used for isolation is fixed, a measurement discernibility (the capability to discern potential faults) ranking table needs to be constructed, and then users can select a certain number of measurements that have higher ranks for fault isolation. The flow chart of building such a measurement discernibility table is illustrated in Figure 4-9, where *C* is the condition attribute, W^D is a set of groups that all the fault samples in the decision table can be divided into by the fault type (the decision attribute *D)*, and $C\_(W^D)$, which indicates the discernibility of the condition attribute *C* to distinguish faults in the decision table, is the low approximation of W^D with

respect to the condition attribute *C* (Please refer to Section 4.2.1 for more details of $C\_(W^D)$).

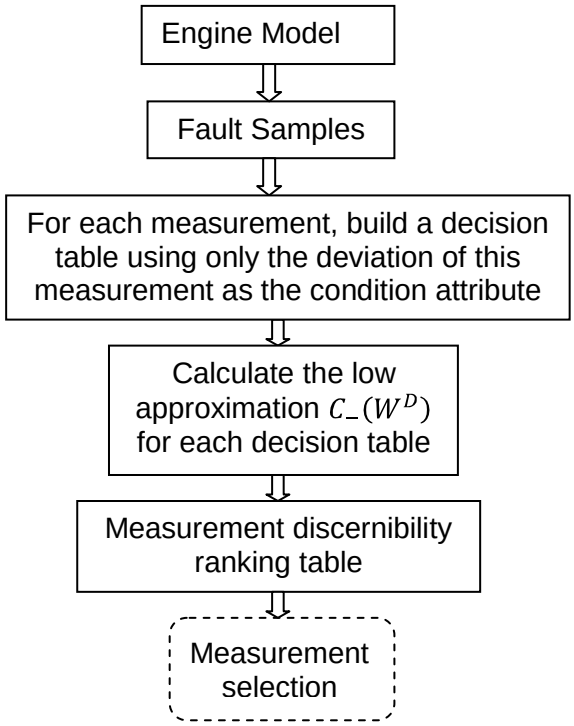

Figure 4-9 Measurement selection based on fault discernibility

As shown in Figure 4-9, such a measurement selection contains the following steps:

- Firstly, use an engine model to generated fault samples to cover all potential faults, and use the generated samples to build a decision table for each measurement. Each decision table considers fault samples as objects, the fault type as the decision table, and only the deviation of the measurement as the condition attribute. Taking one of the measurements (ΔP_1) in Table 4-4 in Section 4.2.2.2 as an example, a decision table shown in Table 4-8 can be built for this measurement using the fault sample included in Table 4-4.

Table 4-8 An example of a decision table with one condition attribute

Fault Sample	Condition attribute (C)	Decision attribute (D)
	$\Delta P_1(\%)$	Fault
S1	v_{11}	$Fault_1$
S2	v_{21}	$Fault_1$
S3	v_{31}	$Fault_2$
S4	v_{41}	$Fault_2$

- Secondly, calculate $C_(W^D)$for each decision table built in the first step, which indicates the discernibility of the measurement included in this decision table.
- Thirdly, rank all the measurements according to the fault discernibility of each of them (i.e. the size of the low approximation $C\_(W^D)$ that each measurement corresponds to) to form a measurement discernibility table.
- Finally, according to the measurement number that the user wants to use for fault isolation, select the same number of measurements having higher ranks from the measurement discernibility table for fault isolation.

4.2.4.2 Rule generation

Once measurements for fault isolation have been selected by the method mentioned in the previous section, using the generated fault samples also mentioned in the previous section with only the selected measurements, either a decision table with conventional fault signature or with enhanced fault signatures can be constructed (since the measurement selection method is suitable for either fault isolation using conventional fault signatures or using enhanced fault signatures). In addition, rules can be generated from either of them by the same method for rule generation described 4.2.2.4 after carrying out discretization and attribute reduction to them by the same algorithm introduced in 4.2.2.3.

4.2.4.3 Reasoning

By the same reasoning algorithm mentioned in Section 4.2.2.5, rules generated from the discretized and reduced decision table can be utilised for fault isolation.

The application of this version of the developed rough set based fault isolation approach presented in this section will be described in Section 4.3.2.

4.2.5 Fault isolation frameworks

Apart from the choice of the fault isolation method, the fault isolation framework, which describes the process employed to classify faults, is also one of the major factors influencing fault isolation accuracy. In this research, three different frameworks shown in Figure 4-10, Figure 4-11, and Figure 4-12 were studied, and they are:

(a) Framework 1 that is simple and the most commonly used framework, and this is the framework used in Sections 4.2.2., 4.2.3 and 4.2.4.

(b) Framework 2 that has been used by several researches for gas path diagnostics [1] [19] [45].

(c) Framework 3 which is proposed by the author.

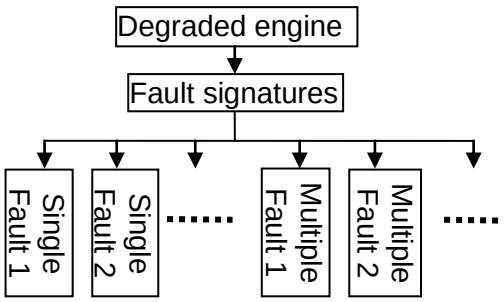

Figure 4-10 Framework 1

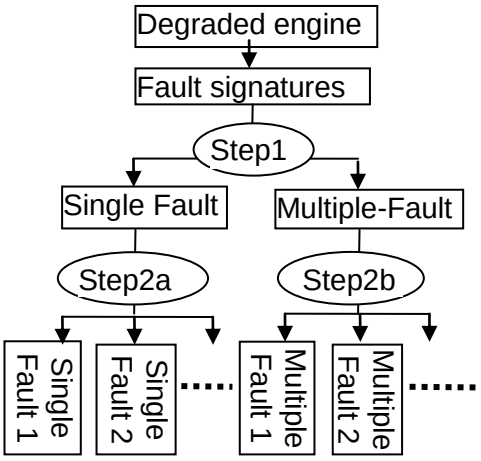

Figure 4-11 Framework 2

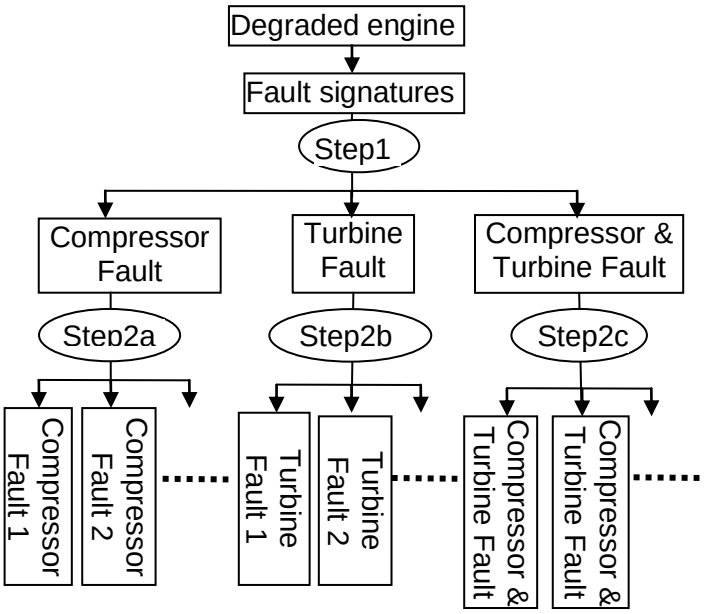

Figure 4-12 Framework 3

In Framework 1, when a component fault occurs in an engine, the faulty component(s) will be identified directly. This framework seems to be the simplest one. However, it also means that the fault isolation method needs to be capable of accurately identifying the actual fault from all potential faults. This may raise difficulties in remaining the accuracy of isolation as the number or the complexity of potential faults increases, especially if multiple-faults are involved in the isolation. The fault isolation processes of the three different versions of

the developed rough set based fault isolation approach mentioned in Sections 4.2.2, 4.2.3, and 4.2.4 actually follow the process in Framework 1.

Framework 2 splits the fault isolation process into two steps. The first step is to identify whether the fault is a single fault or multiple-fault before the actual fault is identified at step 2 which has two options (one for identifying single faults and the other for multiple-faults). Having the two steps may help to reduce the isolation complexity because in each step the complexity and number of potential faults that need to be analysed are low. Apart from that, this framework is seen to be more flexible than Framework 1 in the following two circumstances: (1) when users just want to know whether the fault causes rapid (short-term) deterioration or gradual (long-term) deterioration since the single fault and multiple-fault usually cause rapid and gradual deteriorations respectively [8] (in this case only step 1 in framework 2 needs to be carried out); and (2) when the fault is known as a single fault (users may be able to detect single faults by monitoring rapid trend variation of measurements), only Step 2a in framework 2 needs to carry out the isolation, hence in return the isolation can be much easier as there are no complex multiple-faults involved.

Framework 3 also splits the process into two steps. Like Framework 2, this framework may also be able to reduce the difficulty of isolation and provide more options for users. The first step of this framework is to determine which type(s) of components (i.e. compressors or turbines) is(are) involved with the fault. For example, when a fault occurs and every faulty component is a compressor then the fault will be classified as a compressor fault. Similarly if every faulty component is a turbine then it is a turbine fault and if the fault components contain both compressor(s) and turbine(s) then it is a combined compressor and turbine fault. This kind of information can be valuable to gas turbine operators, because some of the most common faults such as fouling, corrosion and erosion can be induced by different causes, and each of the causes may only lead to deterioration to the same type components (i.e. either only compressors or only turbines). For example, if gas turbines are operating in dusty localities, fouling is more likely to occur in all compressors. Resultants of

combustion can also cause fouling (or corrosion) to and only to turbines. Salts in the air lead to fouling (or corrosion) to compressors of the gas turbines operating in offshore oil rigs, but this is less likely happened to turbines. Therefore under any of the above-mentioned circumstances in this paragraph, only Step1 in Framework 3 needs to be carried out. Even it is needed to identify the faulty components, once step 1 has been carried out, the actual faulty component(s) can then be identified by one of the options at step 2 (steps 2a, 2b or 2c).

4.2.5.1 Integrating the developed rough set based fault isolation approach with different frameworks

To study and test the effectiveness of the introduced three frameworks, they are integrated with one of the versions of the developed rough set based fault isolation approach using enhanced fault signatures. As mentioned in Section 4.2.3, the fault isolation based on rough set using enhanced fault signatures has three steps which are (a) measurement selection, (b) rule generation, and (c) using the generated rule to classify faults by reasoning. Sections 4.2.5.2, 4.2.5.3, and 4.2.5.4 describe how the above steps can be integrated with the frameworks.

4.2.5.2 Measurement selection

The process for measurement selection is consistent for all three frameworks, and is the same as the process presented in Section 4.2.3.2: use all generated fault samples to build a decision table considering conventional fault signatures as condition attributes, and calculate a reduct of the condition attributes in the table, and consequently measurements involved in the reduct will be chosen for fault isolation.

4.2.5.3 Rule generation

Enhanced fault signatures derived from selected measurements are used as condition attributes for decision tables for rule generation. Further, since the fault isolation purposes at different steps in different frameworks are different, a

Table 4-9 Decision tables used for rule generation in different frameworks

Decision table No.	Framework / step that the decision table is built for		Decision table properties	
			Fault samples required	Decision attribute values
1	FW1		All fault samples	faulty component(s)'s names
2	FW2	Step1	All fault samples	"Single Fault" or "Multiple-Fault"
3		Step2a	Single fault samples	faulty component's names
4		Step2b	Multiple-fault samples	faulty components' names
5	FW3	Step1	All fault samples	"Compressor Fault", "Turbine Fault" or "Comp & Turb Fault"
6		Step2a	Compressor fault samples	faulty component(s)'s names
7		Step2b	Turbine fault samples	faulty component(s)'s names
8		Step2c	Comp & Turb fault samples	faulty component(s)'s names

decision table needed to be constructed to generate a unique set of rules for each step. In addition, apart from condition attributes (all the decision tables for different steps using the enhanced fault signatures derived from selected measurements as condition attributes), both the choices on fault samples and decision attribute values for constructing such a decision table may be different for different steps. Framework 1 for example only requires one decision table

since it only needs rules to identify the faulty component(s). In Framework 2 or 3, at each step, the isolation purpose is different and hence a different decision table is needed to generate specific rules for the fault isolation at each step. Table 4-9 provides the properties of different decision tables used in Frameworks 1, 2 and 3 where *Comp* and *Turb* denote the compressor and turbine respectively, and *FW* stands for framework.

After building the decision tables, the tables will then be discretized and reduced using the same discretization and attribute reduction algorithm described in Section 4.2.2.3, and hence each of them will result in a new discretized and reduced decision. By applying the same rule generation method introduced in 4.2.2.4, a set of fault isolation rules can then be generated from each of the discretized and reduced decision tables.

4.2.5.4 Reasoning

Each set of fault isolation rules generated will be employed for fault isolation at its responding framework or step by the same reasoning algorithm mentioned in 4.2.2.5. Note that only the set of rules generated for each step will be involved in the reasoning at the step and once the enhanced fault signatures have been obtained from the real engine, they will be used for both fault isolations at steps 1 and 2 in Frameworks 2 and 3.

The application of the developed rough set based fault isolation approach using enhanced fault signatures integrated with the three different frameworks will be presented in Section 4.3.3.

4.3 Application and analysis

This section presents the application and analysis of the different versions of the developed fault isolation approach presented in Sections 4.2.2, 4.2.3, and 4.2.4, and the three different frameworks mentioned in 4.2.5. Section 4.3.1 describes fault isolations using different fault signatures (conventional and enhanced fault signatures). Section 4.3.2 discusses the application of the developed fault isolation approach using enhanced fault signatures with limited measurements. Section 4.3.3 studies the three different frameworks and proposes a suitable framework for the developed fault isolation approach.

Rough Set Exploration System (RSES) 2.2 [72] developed by Warsaw University, which is a rough set toolkit can be used to analyze of table data, is employed in this study as it is capable of carrying out discretization & attribute reduction, rule generation and reasoning by the methods described in Sections 4.2.2.3, 4.2.2.4, and 4.2.2.5 respectively. The user's guide of RSES 2.2 can be found in [73].

4.3.1 Fault isolation using different fault signatures

This section describes the application of two versions (the one using conventional fault signatures and another one using enhanced fault signatures) of the develop fault isolation approach (referred to as Approach 1 in this section) and the corresponding results and analysis. In addition, in order to test the effectiveness of enhanced fault signatures on other rough set based approaches, another rough set based approach (referred to as Approach 2) is also applied for the fault isolation using conventional and enhanced fault signatures respectively. The Approach 2, which follows the same fault isolation process of Chen et al.'s approach [36] presented in Section 4.2.1, uses the manual discretization method (the one used to create Table 4-2) and the Genetic Algorithm [38] introduced in Section 4.2.1, and the same rule generation and reasoning methods described in Sections 4.2.2.4 and 4.2.2.5 respectively. Hence this section also describes the application and analysis for the Approach 2. In the application of the Approach 2, RSES 2.2 was also

employed for attribute reduction, rule generation and reasoning (RSES 2.2 is also able to carry out attribute reduction based on the Genetic Algorithm method mentioned in this paragraph). All the two versions of the Approach 1 and the Approach 2 were applied to a two-spool turbofan model engine for single and dual-component faults isolation. Therefore six fault isolation test cases shown in Table 4-10 were studied.

Table 4-10 Six different test cases of fault isolation

	Fault isolation approach		Fault signatures		Potential faults considered	
	Approach 1	Approach 2	Conventional	Enhanced	single faults only	single & dual faults
Case1	X		X		X	
Case2	X			X	X	
Case3	X		X			X
Case4	X			X		X
Case5		X	X			X
Case6		X		X		X

Cases 1 and 2 only carried out the same single fault isolation using the same approach (Approach 1) but different types of fault signatures.

Cases 3 and 4 carried out isolation of the same single and dual faults using the same approach (Approach 1) but different types of fault signatures.

Cases 5 and 6 carried out isolation of the same single and dual faults using the same approach (Approach 2) but different types of fault signatures.

4.3.1.1 Model engine

The two-spool turbofan model engine used in this study, which is similar to the Pratt & Whitney PW4000-94 engine, comprises a fan, an intermediate (IP) compressor, a high pressure (HP) compressor, an annular combustor, a HP turbine and a low pressure (LP) turbine. The LP shaft rotational speed was used

as the handle to control the status of the engine in this study. Using Cranfield University gas turbine performance and diagnostics software called PYTHIA, a non-linear thermodynamic performance model was built for the engine and used for generating fault samples. The model engine configuration is shown in Figure 4-13.

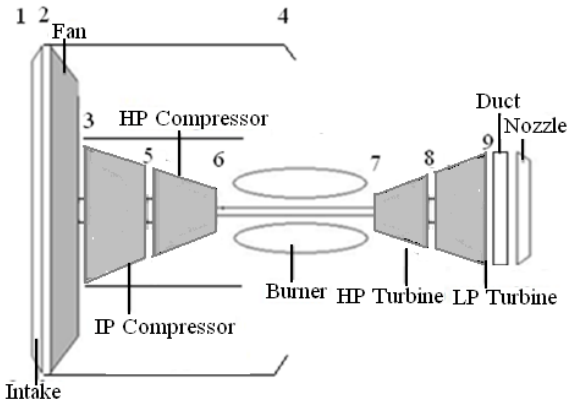

Figure 4-13 Model engine configuration

The characteristics of this model engine is shown in Table 4-11.

Table 4-11 Engine characteristics

Engine Characteristics	Value	Unit
Mass flow rate	700	kg/s
Bypass ratio	5	
Fan PR	1.7	
IP Compressor PR	1.5	
HP Compressor PR	12	
Overall PR	31	
TET	1700	k
Thrust	278	KN
Fuel flow rate	3.8	kg/s

Assuming the 12 measurements shown in Table 4-12 are potential measurements that can be used for diagnostics, the measurement selection method described in Section 2.2.1 was utilized to select appropriate ones for

two different fault isolations (i.e. single faults isolation only and both single and dual-faults isolation), since some of them may not effectively contribute to the fault isolations.

Table 4-12 Potential measurements for fault isolation

Measurement	Symbol	Unit
Fan outlet total temperature	T3	k
Fan outlet total pressure	P3	atm
IP compressor outlet total temperature	T5	k
IP compressor outlet total pressure	P5	atm
HP compressor outlet total temperature	T6	k
HP compressor outlet total pressure	P6	atm
HP turbine outlet total temperature	T8	k
HP turbine outlet total pressure	P8	atm
LP turbine outlet total temperature	T9	k
LP turbine outlet total pressure	P9	atm
Fuel flow rate	FF	kg/s
HP compressor rotational speed	N2	rpm

4.3.1.2 Fault samples

In this study, degradation indices including flow capacity index (SF_{FC}) and isentropic efficiency index (SF_E), which represent the shift of speed lines on compressor or turbine characteristic maps caused by their degradation, are employed to represent the level of the degradation of compressors and turbines. The details of the degradation indices can be found in [5]. Usually burner degradation does not induce large variations in gas path measurements; therefore it is ignored in this study. Table 4-13 shows the ranges of the levels of component degradations considered in this study.

Table 4-13 Ranges of component degradations

Fault	Health Parameter	Range (%)
Compressor	Flow Capacity Index SF_{FC}	-0.5 to -5
	Isentropic Efficiency Index SF_E	-0.5 to -5
Turbine	Flow Capacity Index SF_{FC}	-0.5 to -5 and 0.5 to 5
	Isentropic Efficiency Index SF_E	-0.5 to -5

As shown in Figure 4-14, the degradation of either compressor or turbine can be represented in a two-dimensional degradation since the degradation of compressors or turbines can be represented by two degradation indices. Hence each point in a degradation space can be selected to produce a fault sample, and fault samples for constructing decision tables or verification can be selected from degradation spaces.

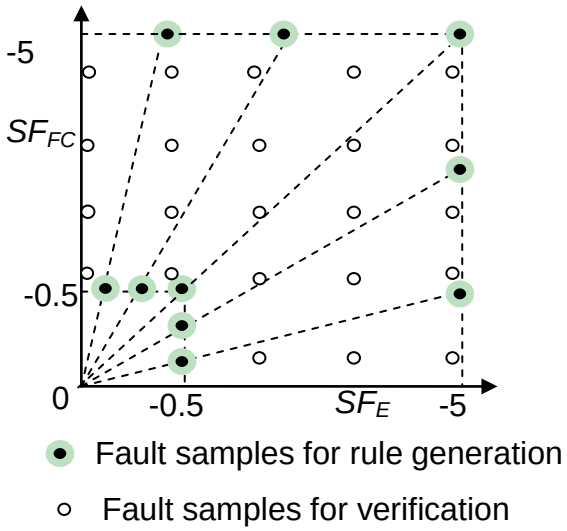

Figure 4-14 Fault sample selection for a compressor

In this study, fault samples for constructing decision tables are selected at intersection points between constant SF_{FC}/SF_E lines and the maximum and

minimum boundaries of the health parameters being considered. The reason for do so is due to the understanding that the measurement deviations may vary linearly with the health parameters if the level of the degradation is low. Therefore the two intersection points on each constant SF_{FC}/SF_E line should be sufficient to represent the range of deviation of the measurements along this line within the considered degradation ranges. Thus, selecting points at the boundaries in the considered degradation area may be able to represent the range of the variation of measurements corresponding to the considered degradation area in the space.

To test the effectiveness of the developed fault isolation approach, fault samples for verification are simulated at discrete points with constant intervals in between within the range of degradation in concern, see Figure 4-14. Unlike the fault samples for decision table constructions, the verification samples are simulated with implanted measurement noises whose assumed ranges are shown in Table 4-14. These noises are assumed to be Gaussian distributed.

Table 4-14 Measurement noise ranges [74]

Measurement	Measurement noise range	
	Minimum (%)	Maximum (%)
T3	-0.1	0.1
P3	-0.1	0.1
T5	-0.2	0.2
P5	-0.1	0.1
T6	-0.4	0.4
P6	-0.1	0.1
T8	-0.4	0.4
P8	-0.1	0.1
T9	-0.4	0.4
P9	-0.1	0.1
FF	-1	1
N2	-0.02	0.02

The constructed engine performance model was used to generate a total of 3,612 fault samples for building decision tables. These 3612 samples cover all potential single and dual-component faults (5 single faults and 10 dual faults) and some different levels of degradation for each fault. Notes that all measurement deviation values in each sample are rounded down to decimal point. All the samples were generated at full power setting and ISA sea level ambient condition.

Table 4-15 Generated fault samples

Fault type		No. of samples			
		for constructing decision tables	for verification		
			Half assumed noise ranges	Normal assumed noise ranges	Double assumed noise ranges
Single Faults	FAN	40	100	100	100
	IPC	40	100	100	100
	HPC	40	100	100	100
	HPT	80	200	200	200
	LPT	80	200	200	200
Dual-Faults	FAN&IPC	196	625	625	625
	FAN&HPC	196	625	625	625
	IPC&HPC	196	625	625	625
	HPT&LPT	392	1250	1250	1250
	FAN&HPT	392	1250	1250	1250
	FAN&LPT	392	1250	1250	1250
	IPC&HPT	392	1250	1250	1250
	IPC&LPT	392	1250	1250	1250
	HPC&HPT	392	1250	1250	1250
	HPC&LPT	392	1250	1250	1250
	Total	3612	11325	11325	11325

To investigate the capability of the fault isolation approaches to deal with different levels of measurement noises, for each selected verification point, see

Figure 4-14, three fault samples were generated with half, normal and double the assumed noise ranges respectively. Different amounts of verification fault samples were generated for different types of faults as shown in Table 4-15. It is worth mention that the number of samples generated for each type of fault is depends on its complexity. For example, each single fault has fewer samples for either rule generation or verification since each of them only involves with two health parameters, while each dual-fault involving with four health parameters has more samples. In addition, the degradation range of each turbine are wider than that of each compressor, hence faults involving with turbines have more samples compared with those which have the same number of health parameter but don't involve with any turbine(s).

4.3.1.3 Discretization and attribute reduction

Using the generated fault samples, decision table(s) was then constructed and import to RSES 2.2. Consequently discretization and attribute reduction were carried out to the decision table(s) in each case:

- in cases 1, following the fault isolation process described in Section 4.2.2, firstly, 280 single fault samples were used for constructing a decision table which use 280 single fault samples as objects, conventional fault signatures (i.e. 12 deviations of the measurements shown in Table 4-12) as condition attributes and the fault type as a decision attribute, secondly, discretization and attribute reduction were carried out to the decision table to discretized fault signatures and select appropriate measurements for fault isolation, and finally a discretized and reduced decision table was formed.
- In case 2, following the fault isolation process described in Section 4.2.3, firstly, construct the same decision table as described in last paragraph and carry out discretization and attribute reduction to select appropriate measurements for fault isolation, secondly, construct another decision table, which use the same 280 single fault samples as objects, derived enhanced fault signatures (i.e. the measurement deviations and ranking

parameters derived from the select measurements according to Equation (4-6)) as condition attributes and the fault type as a decision attribute, thirdly, discretization and attribute reduction were carried out to this new decision table to form a discretized and reduced decision table.

- In case 3, the discretization and attribute reduction process is same as in case 1, however both the decision table used in case 3 uses all 3612 fault samples (i.e. both single and dual-fault samples) as objects.
- In case 4, the discretization and attribute reduction process is same as in case 2, however both the two decision tables (one for measurement selection and one for rule generation) used in case 4 use all 3612 fault samples as objects.
- In case 5, the discretization and attribute reduction were carried out separately. Firstly, the same decision table as the one constructed in case 3 was built, and the manual discretization algorithm mentioned in Section 4.2.1 (the one used for creating Table 4-2) was used for discretization, then attribute reduction was carried out to the discretized decision table to select measurements for fault isolation by the Genetic Algorithm [37] based attribute reduction approach introduced in Section 4.2.1, and finally a discretized and reduced decision table was formed.
- In case 6, the discretization and attribute reduction were carried out separately using the same methods used in case 5, the decision table used in this case is similar to the one used in case 5, the only different between them is that in this case the decision table used enhanced fault signatures derived from the measurements selected for fault isolation in case 5 as condition attributes (in this way, both cases 5 and 6 used the same measurements for fault isolation and the only different between them is that they used different type of fault signatures).

The selected measurements for fault isolation in each case are shown in Table 4-16.

Table 4-16 Selected measurements in each case

	selected measurements
Cases 1 & 2	T3,P3,T5,P5,T6,P6,P8,FF,N2 (9 measurements)
Cases 3 & 4	T3,P3,T5,P 5,T6,P6,P8,T9,P9,FF,N2 (11 measurements)
Cases 5 & 6	T3,P3,T5,P5,T6,P6,T8,P8,T9,P9,FF,N2 (12 measurements)

Form Table 4-16 the following can be observed:

- Firstly for single fault isolation using the fault isolation Approach 1, T8, T9 and P9 are not needed, and this may due to that the similarity of the deviations of these three measurements caused by different faults is high, and hence it is difficult to use them to distinguish faults.
- Secondly, for fault isolation of single and dual faults using the Approach 1 only T8 can be excluded as one the complexity and number of faults needed to deal with are much higher and hence more measurements need to distinguish them.
- Thirdly, the Approach 2 requires all the 12 measurements for fault isolation. The reason the Approach 2 requires more measurements than the Approach 1 is that it uses an inefficient discretization which causes large information loss and therefore in order to distinguish all faults in the decision table the Approach needs to discover knowledge from more measurements to compensate the information loss.

The numbers of enhanced fault signatures remaining after the attribute reduction in each of the cases using enhanced fault signatures are shown in Table 4-17. In such a case, enhanced fault signatures are derived from the

selected measurements shown in Table 4-16. For example, Table 4-17 shows that 45 enhanced fault signatures (9 measurement deviations and 36 ranking parameters) were derived from the selected 9 measurements in case 2.

Table 4-17 Attribute reduction results for enhanced fault signatures

	No. of enhanced fault signatures	
	before attribute reduction	after attribute reduction
Case 2	45	7
Case 4	66	33
Case 6	78	64

From Table 4-17, several conclusions can be made:

- Firstly, in each case in this table the number of enhanced fault signatures is reduced by attribute reduction, (especially in case 2, only 7 enhanced fault signatures remained after the attribute reduction).
- Secondly, the number of enhanced fault signatures required for fault isolation increases as the complexity and number of potential faults increases, which can be observed by comparing the remaining numbers of signatures in case 2 and 4.
- Thirdly the Approach 2 needs much more enhanced fault signatures for fault isolation than the Approach 1 does (the former requires 64 signatures and the latter 33 signatures), the reason for this is that the discretization method of the former is not effective as that of the latter and the faults needed to be distinguish are the same for them and hence the former requires more enhanced fault signatures to distinguish potential faults.

In addition, statistical information about the cuts of the condition attributes (fault signatures) being taken in the discretization processes in the cases using the Approach 1 is shown in Table 4-18.

Table 4-18 Statistical information about the condition attribute cuts in the discretized and reduced decision tables used for rule generation

	No. of condition attributes having more than 5 cuts	No. of condition attributes having more than 2 cuts
Case1	0	0
Case2	0	0
Case3	6	10
Case4	1	5

Form Table 4-18 some observations can be obtained:

- In cases 1 and 2, all the condition attributes have no more than two cuts and hence they all only have one cut because the similarity between five single faults is very low and hence less cuts need to be taken to distinguish them.
- As the complexity and number of potential faults increase the number of condition attributes having a large number of cuts increases as well. This can be proved by comparing the statistical information about the cuts in case 1 and 3 or in case 2 and 4.
- Apart from that, it also can be seen from this table that: when dealing with complex faults the introduction of enhanced fault signatures largely reduced the number of the condition attributes having many cuts after discretization. This can be observed in the comparison between case 3 and case 4 in this table where the number of condition attributes having more than either 5 or 2 cuts in Case 4 is much lower than in Case 3.

4.3.1.4 Rule generation

Table 4-19 Number of rules generated in each case

	Number of generated rules	Time spent on the rule generation
Case1	140	0.2 minutes
Case2	106	0.1 minutes
Case3	17472	19 minutes
Case4	32925	110 minutes
Case5	671	1.5 minutes
Case6	34444	82 minutes

Once a discretized and reduced decision table for rule generation had been constructed in each case, a set of fault isolation rules then were generated from each discretized and reduced decision table by the method mentioned in Section 4.2.2.4. The number of rules generated and the corresponding computational time spent using a standard desktop computer with Intel Core 2 Quad 2.4GHz /1.98GB RAM CPU in each case are shown in Table 4-19. Examples of the generated rules in cases 1 to 4 are shown in 6Appendix D.

By analyzing the results shown in Table 4-19, the following can be observed:

- By comparing the numbers of rules in case 1 and case 2 with the numbers in case 3 and case 4 respectively, it can be seen that when the potential faults become more complex, the number of rules are required to distinguish them increases largely. Another reason causing this phenomenon is that the number of single fault samples used in this study is much lower than that of dual-fault samples and hence fewer rules are required to distinguish the faults in the single fault samples.
- In addition, by comparing the numbers of rules in case 3 and case 5 with the numbers in case 4 and case 6 respectively, it shows that the introduction of enhanced fault signatures in either the Approach 1 or

Approach 2 leads to large rise of the number of generated rules. Especially for the Approach 2, with conventional fault signatures (case 5) only 671 rules were generated due to that the simple manual discretization method results in large information loss in the decision table and hence less knowledge can be discovered and consequently fewer rules can be generated, however with enhanced fault signatures (case 6) the characteristics of faults can be enhanced and hence more knowledge can be discovered, which leads to that large amount of rules were generated in this case.

- The results also indicate that a large number of rules can be generated in a short period of time (the longest period used for rule generation in the cases is only 110 minutes), hence rough set based fault isolation approaches are able to offer an efficient way to generate rules.

4.3.1.5 Fault isolation

In each case, the corresponding generated rules were used for isolating faults in the corresponding verification samples by the reasoning algorithm presented in Section 4.2.2.5. In cases 1 and 2, their rules were used to isolate faults in each of the three sets of 700 single fault verification samples shown in Table 4-15 separately. In cases 3, 4, 5, and 6, their rules were used for fault isolation of each of the three sets of 11325 single and dual-fault verification samples. In this study, the accuracy of fault isolation in each case is measured by a parameter called success rate which is the ratio between the total number of the fault samples being carried out fault isolation successfully and the total number of fault samples used for verification. The success rates obtained in each case are shown in Table 4-20.

Table 4-20 Fault isolation success rates

	Number of verification samples	Total success rate		
		under half assumed noise ranges	under normal assumed noise ranges	under double assumed noise ranges
Case1	700	0.94	0.92	0.85
Case2	700	0.94	0.89	0.82
Case3	11325	0.62	0.59	0.53
Case4	11325	0.88	0.82	0.76
Case5	11325	0.40	0.38	0.37
Case6	11325	0.76	0.73	0.69

The fault isolation results in Table 4-20 show the impact of the fault complexity on the fault isolation accuracy and the effectiveness of using the enhanced fault signatures to improve fault isolation accuracy. The following can be observed from this table:

Firstly, the single fault isolation under different levels of measurement noises either in case 1 or case 2 is very successful. The fault isolations of single and dual-faults in both cases 3 and 5 (both using conventional fault signatures) are not successful due the extremely low success rates they have, while the same fault isolation for single and dual-faults in both case 4 and case 5 (both using enhanced fault signatures) are much more successful.

Secondly, the use of the enhanced fault signatures does not contribute to any improvement in the accuracy of single fault isolation. On the contrary, the success rate in Case 2 is slightly worse than that in Case 1 under normal and double assumed noise ranges. This is due to the following reasons:

- For single fault isolation, conventional fault signatures can provide enough information to distinguish faults with high accuracy since the complexity of the faults is relatively low. Therefore the use of enhanced fault signatures does not provide any advantages in this situation.

- The number of condition attributes in Case 1 (9 condition attributes, refer to Table 4-17) in the discretized and reduced decision table is larger than that in Case 2 (7 condition attributes, refer to Table 4-18). This leads to more rules being generated in Case 1 (refer to Table 4-19). In the fault isolation, more rules may lead to a rise in the total number of “votes” for the correct type of faults during reasoning due to the statistical nature of the reasoning method, and consequently may lead to a high possibility of successful isolation.

Thirdly, when using either the conventional or enhanced fault signatures in rough set diagnostics, the fault isolation become less accurate when the potential faults varies from single faults to single and dual-faults. This can be seen by the comparison of success rates in cases 1 and 3 or in case 2 and 4.

Fourthly, when the complexity of the faults increases, fault isolation using the enhanced fault signatures shows significant advantages in accuracy compared with that using conventional fault signatures. This applies to both the rough set based fault isolation approaches (the Approach 1 and Approach 2) studied in the research. This can be seen by comparing the success rates in cases 3 and 4 or cases 5 and 6. Especially for the Approach 2, the employment of enhanced fault signatures (case 6) gives much better success rates under all different levels of measurement noises than the case using conventional fault signatures (case 5). Therefore it can be concluded that the use of enhanced fault signatures in rough set based fault isolation approaches can largely improve the isolation accuracy of complex component faults. The main reasons that the use of enhanced fault signatures can lead to a significant improvement in accuracy for complex fault isolation are:

- The Introduction of enhanced fault signatures gives a reduction in the number of the condition attributes having a large number of cuts (this only applies to the Approach 1), see the statistical information about cuts shown in Table 4-18. The numbers of cuts of condition attributes directly influences the tolerance of the rules measurement noises. The fewer cuts condition attributes have, the larger the intervals will be included in

the rules and consequently the rules will have better tolerance to the uncertainty caused by measurement noises. The use of enhanced fault signatures provides the following advantages: (1) it offers more condition attributes and therefore is able to reduce the number of cuts needed on each condition attribute; (2) the ranking patterns among some measurement deviations may be consistent for the same type of faults but different for different types of faults, which make the ranking parameters effective condition attributes to distinguish faults and hence less cuts will be required.

- The use of enhanced fault signatures leads to that more rules can be generated from the same amount fault samples as enhanced fault signatures are able to enhance the characteristics of faults and more knowledge can be discovered and consequently more rules can be generated (This applied to both the Approach 1 and Approach 2). As mentioned earlier in this section, such an increase could lead to a rise in the total "votes" for the correct type of faults during reasoning, and therefore result in a higher possibility of successful isolation.

Table 4-21 Computational time spent in fault isolation

	No. of verification samples	Computational time spent	
		Total	average
Case1	700	0.1 minutes	0.01 seconds
Case2	700	0.1 minutes	0.01 seconds
Case3	11325	4 minutes	0.02 seconds
Case4	11325	19 minutes	0.10 seconds
Case5	11325	0.6 minutes	0.003 seconds
Case6	11325	7 minutes	0.04 seconds

The total computational time used in fault isolation in each case and the average time used for each verification sample in each is shown in Table 4-21

(the same computer used for rule generation in Section 4.3.1.4 is used for fault isolation as well)

The difference in the total computational time spent on fault isolation between different is mainly due to the difference in the number of rules involved in the reasoning process, the complexity of the rules and, the total number of verification samples. Basically, from Table 4-21, it can be seen that the average computational time spent on each sample in each case is no more than 0.1 seconds; hence the developed fault isolation approach has the potential for on-line applications.

4.3.2 Fault isolation with limited measurements

This section presents the application of the version of the developed fault isolation approach with limited measurements as described in Section 4.2.4. In this application, 11 fault isolation test cases, as shown in Table 4-22, were studied and each of them only allows a certain number of measurements for fault isolation.

In cases 1 to 5, the measurement selection method, mentioned in Section 4.2.4.1, based on the fault discernibility of measurements, was used for selecting a certain number of measurements having a higher discernibility for fault isolation in each case. The selected measurements were then used for rule generation by the method presented in Section 4.2.3.3 and consequently the generated rules were employed for single and dual faults isolation.

In cases 6 to 10, the way to select measurements was also based on their discernibility but in these cases the measurements having lower discernibility were chosen, and in each case rule generation (also by the method presented in Section 4.2.3.3) and fault isolation were also carried out using the selected measurements.

Table 4-22 Fault isolation test cases with different numbers of measurements

Case	Number of measurements allowed for fault isolation	The way to select measurement
1	6	select measurement with **higher** discernibility
2	7	select measurement with **higher** discernibility
3	8	select measurement with **higher** discernibility
4	9	select measurement with **higher** discernibility
5	10	select measurement with **higher** discernibility
6	6	select measurement with **lower** discernibility
7	7	select measurement with **lower** discernibility
8	8	select measurement with **lower** discernibility
9	9	select measurement with **lower** discernibility
10	10	select measurement with **lower** discernibility
11	11	select measurement with **higher** discernibility

In cases 1 to 5, the number of measurements allowed for fault isolation are 6, 7, 8, 9, and 10 in cases 6 to 10 are also 6, 7, 8, 9, and 10. The intention in studying the first 10 cases is to compare cases 1, 2, 3, 4, and 5 with 6, 7, 8, 9, and 10 respectively to see whether choosing measurements having discernibility can lead to higher fault isolation accuracy when the number of measurements used for isolation is the same.

The last case is for comparing this version with the one using enhanced fault signatures in terms of fault isolation accuracy. This was done by using the former to select the same number of measurements selected by the latter in

Section 4.3.1 (which is 11 measurements, see Table 4-16) and using them for rule generation and fault isolation and consequently comparing the fault isolation results resulting from both of them.

The fault isolation process in each case, which follows the process presented in Figure 4-8 in Section 4.2.4, is described in Sections 4.3.2.1, 4.3.2.2, 4.3.2.3, and 4.3.2.4.

4.3.2.1 Fault samples

The same gas turbine engine used in Section 4.3.1.1 is employed in this Section, and hence the same fault samples for rule generation and the three sets of fault samples for verification shown in Table 4-15 in Section 4.3.1.2 are also used in this application.

4.3.2.2 Measurement selection

Using the generated fault samples, a decision table was constructed for each of the 12 potential measurements; the decision table uses all the 3125 fault samples as objects, fault type as a decision attribute and only the deviation of this measurement as the condition attribute. For each decision table, the low approximation of W^D with respect to the condition attribute (refer to Section 4.2.1), which indicates the discernibility of this condition attribute to distinguish faults in the decision table, was calculated by RSES 2.2. In order to normalise the low approximation, the ratio of the number of fault samples in the low approximation and the total number of the fault samples (3125) is calculated and used to represent the low approximation. The normalised low approximation given by each measurement is shown in Table 4-23.

Table 4-23 Measurement discernibility table

Measurement	Fault discernibility Ranking	Low approximation region (%)
P5	1	4.6
T5	2	2.0
P8	3	1.9
P3	4	1.7
P9	5	1.7
N2	6	1.5
T9	7	1.4
P6	8	1.1
FF	9	1.1
T6	10	0.9
T8	11	0.4
T3	12	0.3

This discernibility table indicates that most of the pressure measurements have higher discernibility than the temperature measurements, and P5 seems to be the most effective measurement to distinguish faults. Therefore, the similarity of pressure measurement deviations caused by different faults is lower than that of temperature measurement deviations.

According to the ranking Table 4-23, measurements having higher ranks were chosen in cases 1 to 5 (the number of selected measurements in each case equals the number allowed in this case), while those having lower ranks were chosen in cases 6 to 10. In addition, 11 measurements having higher ranks were selected in case 11. The selection results are shown in Table 4-24.

Table 4-24 Measurement selection results

Case	selected measurements
1	P5,T5,P8,P3,P9,N2
2	P5,T5,P8,P3,P9,N2,T9
3	P5,T5,P8,P3,P9,N2,T9,P6
4	P5,T5,P8,P3,P9,N2,T9,P6,FF
5	P5,T5,P8,P3,P9,N2,T9,P6,FF,T6
6	T3,T8,T6,FF,P6,T9
7	T3,T8,T6,FF,P6,T9,N2
8	T3,T8,T6,FF,P6,T9,N2,P9,
9	T3,T8,T6,FF,P6,T9,N2,P9,P3
10	T3,T8,T6,FF,P6,T9,N2,P9,P3,P8
11	P5,T5,P8,P3,P9,N2,T9,P6,FF,T6,T8

4.3.2.3 Rule generation

In each case, the generated 3125 samples with the selected measurements were utilised to construct a decision table for rule generation. Such a decision table considers the fault samples as objects, the enhanced fault signatures derived from the selected measurements as condition attributes, and fault type as a decision attribute. Discretization and attribute reduction were then carried to each decision table to form a discretization and reduced decision table by the algorithm presented in Section 4.2.2.3, and rules were then generated by the rule generation method mentioned in Section 4.2.2.4.

4.3.2.4 Fault isolation

Finally, in each case, the generated rules were employed for fault isolation of the three sets of fault samples (each set includes 11325 samples) generated for verification by the reasoning algorithm introduced in Section 4.2.2.5. The fault isolation results in cases 1 to 10 are shown in Table 4-25.

Table 4-25 Fault isolation results with different measurements

Case	Fault isolation success rate		
	under half assumed noise ranges	under normal assumed noise ranges	under double assumed noise ranges
1	0.68	0.67	0.65
2	0.68	0.65	0.62
3	0.78	0.75	0.72
4	0.77	0.74	0.71
5	0.82	0.78	0.71
6	0.59	0.53	0.47
7	0.62	0.55	0.49
8	0.66	0.59	0.51
9	0.75	0.67	0.59
10	0.85	0.80	0.72

The fault isolation results in cases 1 to 10 are also presented in Figure 4-15 in order to compare them more conveniently.

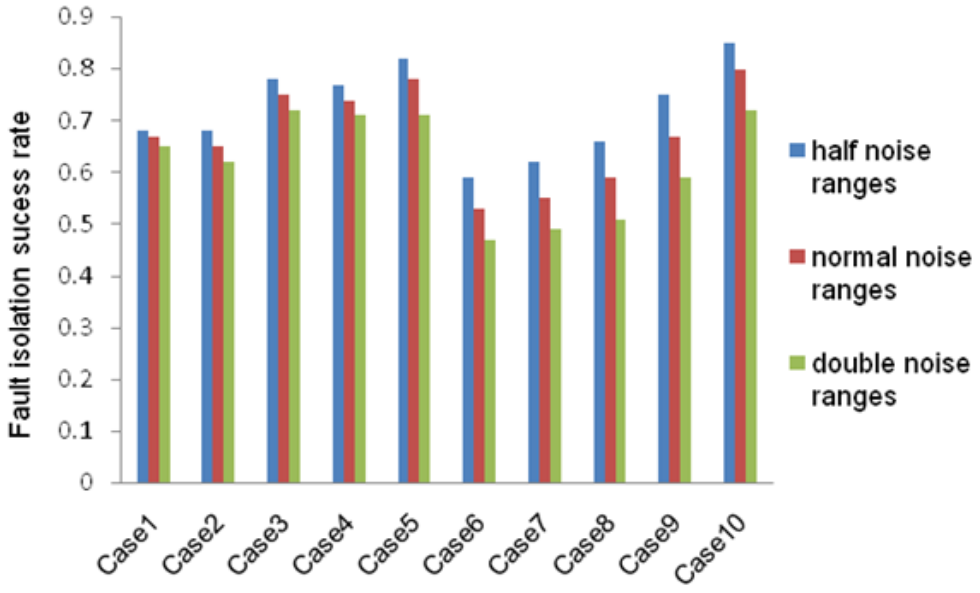

Figure 4-15 Fault isolation results with different measurements

By comparing the fault isolation results from cases 1, 2, 3, 4 and 5 with the those from 6, 7, 8, 9 and 10 respectively (see Figure 4-15), it can be seen that: Generally when the number of measurements for fault isolation is the same, selecting measurements with higher fault discernibility ranks results in better fault isolation accuracy under different noise levels. In this study, this applies to the circumstances when the numbers of measurements allowed for fault isolation are 6, 7, 8, and 9. However, this does not apply to those where the number of selected measurements is 10. This indicates that the effectiveness of this measurement selection approach may reduce when the number of measurements allowed for isolation is high. The reasons causing this may be:

- Firstly, when the number of measurements allowed to be used for fault isolation increases, the similarity between the selected measurements given by the two different measurement selection methods (the one selects measurements having higher ranks and the other chooses the measurements having lower ranks) increases. Therefore, the fault isolation accuracies resulting from the two measurement selection methods may be similar as well.
- Secondly, the ranking in the discernibility table only considers the individual measurement's discernibility and the correlation between measurements is not taken into account. Thus, when there are many measurements selected for isolation, the faults distinguished by a measurement may be similar to those given by other measurement(s) in the selected measurements and hence make this measurement selection method less effective in these circumstances.

The results in case 11, along with the fault isolation results given by the version using enhanced fault signatures, (i.e. the results of case 4 shown in Table 4-20 in Section 4.3.1) are illustrated in Table 4-26.

Table 4-26 Comparison of fault isolation results given by two different versions of the developed approach

	Fault isolation success rate		
	under half assumed noise ranges	under normal assumed noise ranges	under double assumed noise ranges
the version with limited measurements (case 11 in this application)	0.82	0.80	0.73
the version using enhanced fault signatures	0.88	0.82	0.76

The comparison illustrated in Table 4-26 indicates that if the number of measurements used is the same, the new measurement selection is less effective than the one employed by the version using enhanced fault signatures as the latter is able to lead to a better fault isolation success rate.

Therefore, it can be concluded that the new measurement selection method employed in this version of the developed fault isolation approach is able to provide good measurement selection solutions for rough set based fault isolation when the number of measurements allowed for fault isolation is not very high. This make this version a suitable replacement for the version using enhanced fault signatures which may require many measurements for fault isolation under the circumstance when the number of measurements required by the latter cannot be satisfied. However, the version using enhanced fault signatures still has advantages when the number of measurements required by it can be satisfied.

4.3.3 Fault isolation using different frameworks

The application of the three different frameworks (shown in Figure 4-10, Figure 4-11 and Figure 4-12) integrated with the developed fault isolation approach using enhanced fault signatures is presented in this section. In this application, the three different fault isolation frameworks were studied by measuring their average and individual fault isolation accuracies for different types of component faults under different levels of measurement noises. In addition, based on the study of the three different frameworks, a new framework is proposed in this section which is a combination of all three frameworks, and is more robust and flexible than each of the individual frameworks as it is able to provide a useful decision tree for users in order ensure the accuracy of fault isolation. Note; the first framework (framework 1 shown in Figure 4-10) actually have been tested in Section 4.3.1; in this section it will be compared with the other two frameworks.

Sections 4.3.3.1, 4.3.3.2, 4.3.3.3, and 4.3.3.4 describe the application of fault isolation based on the three different frameworks.

4.3.3.1 Fault samples

The same gas turbine engine used in Section 4.3.1.1 is employed in this section, and hence the same fault samples for rule generation and the same three sets of fault samples, shown in Table 4-15 in Section 4.3.1.2, are also used in this application.

4.3.3.2 Measurement selection

One decision table, using 12 measurement deviations as condition attributes, the fault type as a decision attribute, faulty component(s)'s names as the decision attribute values, and all 3612 fault samples generated for decision table construction as objects, was constructed for the measurement selection for all three frameworks. This decision table is exactly the same as the one used for measurement selection in case 4 in Section 4.3.1 (i.e. fault isolation based on the developed approach using enhanced fault signatures), hence the

same selection result was obtained in each case which is a set of measurements (11 measurements) including all the potential measurements shown in Table 4-12 in Section 4.3.1.1 except T8.

4.3.3.3 Rule generation

Since the isolation purposes at different steps in different frameworks are not the same, a decision table needed to be constructed to generate a unique set of rules for each step. The generated fault samples with the selected 11 measurements were then used to construct 8 decision tables for different steps in different frameworks according to the description in Table 4-9 in Section 4.2.5.3. It is worth mentioning that all the 8 decision tables using enhanced fault signatures derived from the 11 selected measurements as condition attributes (i.e. 11 measurement deviations and 55 ranking-parameters derived from the 11 measurement deviations).

Table 4-27 Statistics of enhanced fault signatures and generated rules

Framework		No. of enhanced fault signatures		No. of generated rules
		before attribute reduction	after attribute reduction	
1		66	33	32925
2	Step1	66	31	13623
	Step2a	66	7	140
	Step2b	66	25	26533
3	Step1	66	27	18739
	Step2a	66	17	1803
	Step2b	66	12	322
	Step2c	66	19	8934

After building the decision tables, the tables were discretized and reduced using the same discretization and attribute reduction algorithm described in Section 4.2.2.3, and hence each of them resulted in a new discretized and reduced decision and the 66 enhanced fault signatures were then reduced. The reduction results of enhanced fault signatures for each step in each framework are shown in Table 4-27. By applying the same rule generation principle introduced in 4.2.2.4, once discretization and attribute reduction have been carried out to the 8 decision tables, a set of decision rules can then be generated from each of the decision tables. The number of rules generated for each step is also shown in Table 4-27.

From Table 4-27, it can be seen that the total number of reduced enhanced fault signatures increases as the number or the complexity of the potential faults rises. Framework 1 has the largest number of reduced fault signatures due to the fact that it needs to deal with more faults (15 different faults-5 single faults and 10 dual-faults) compared with the steps in the other frameworks.

Apart from that, it also can be seen from this table that Framework 1 has the largest number of rules generated (32925) because it involves the highest numbers of objects, condition attributes and potential faults. Similarly, such a number for step2a in framework 2 is the lowest due to the extremely low numbers of objects, condition attributes and complexity of faults included in the table. Note that in Framework 2, compared with step1, step2b has many more generated rules, although the former has more objects and condition attributes; this is because the complexity of the faults needed to be dealt with at step2b is much higher than at step1.

4.3.3.4 Fault isolation

The accuracies of the fault isolation at each individual step and the overall framework are assessed using the success rate and the average success rate respectively. The average success rate represents the ratio between the total numbers of the verification fault samples being successfully isolated at both

steps 1 and 2 and the total fault verification samples. The fault isolation results obtained in the three different frameworks are shown in Table 4-28.

Table 4-28 Fault isolation results

Framework		No. of verification samples	success rate		
			under half noise ranges	under normal noise ranges	under double noise ranges
1		11325	0.88	0.82	0.76
2	Step1	11325	0.96	0.95	0.94
	Step2a	700	0.94	0.89	0.82
	Step2b	10625	0.84	0.8	0.71
	average		0.81	0.77	0.67
3	Step1	11325	0.96	0.92	0.9
	Step2a	2175	0.88	0.83	0.77
	Step2b	552	0.79	0.78	0.77
	Step2c	8598	0.8	0.75	0.7
	average		0.78	0.71	0.65

From the fault isolation results shown in Table 4-28several observations can be drawn:

Firstly, the results in the table show that Framework 1 has a higher average success rate than Frameworks 2 and 3. The intention of the introduction of Frameworks 2 and 3 is to improve the quality of fault isolation, based on rough set, by dividing the isolation process into steps, and at each step the complexity of potential faults is lower, which may improve the overall isolation accuracy. However neither of the obtained average success rates in Frameworks 2 or 3 is higher than Framework 1. The main reason causing this phenomenon is because the utilization of the enhanced fault signatures in Frameworks 2 or 3 is not as successful as in Framework 1. One of the features of the employed discretization algorithm is that it carries out discretization and attributes

reduction simultaneously. Although the attribute reduction can be utilised for measurement selection, it also limits the use of enhanced fault signatures. From Table 4-28, it can be seen that many enhanced fault signatures were removed from the decision tables constructed for rule generation after attribute reductions and which directly leads to fewer generated rules from each decision tables. The drop in number of rules may lead to lower success rates due to the statistical nature of the reasoning method. Table 4-28 also shows that the total number of enhanced fault signatures remaining after the attribute reduction in Framework 1, is the largest one. Consequently, the effect of the drop on generated rules, which is caused by the reduction of the enhanced fault signatures number during the discretization process, has the smallest impact on Framework 1. The benefits, given by reducing the complexity of potential faults at each step in frameworks 2 and 3, does not compromise the effect of the drop on generated rules. This means that Framework 1 has the highest average success rate.

Secondly, although Framework 1 has the highest overall isolation accuracy, the other two still have advantages in isolation accuracy under certain circumstances which are likely to occur in practice. This can be seen clearly from Table 4-28 where the success rates at some steps in both Frameworks 2 and 3 are quite high (i.e. step 1 and step 2a in Framework 2, and step 1 in Framework 3). The reason for this is because the low complexity of potential faults needed to be dealt with at each step compensates for the effect caused by the drop of generated rules. Therefore, Framework 2 is actually very effective when users just want to find out whether the fault causes rapid (short-term) deterioration (usually caused by single faults) or gradual (long-term) deterioration (usually caused by multiple faults), or the fault has been identified as a single fault by monitoring rapid trend variation of measurements and the faulty component needs to be identified. In addition, Framework 3 becomes more useful if it is desired to determine which type(s) of components (i.e. compressors or turbines) are involved with the fault, since it has a suitable option (step 1) to do so and, as shown in Table 4-28, very high isolation accuracy can be achieved in this instance.

Thirdly, frameworks 2 and 3 seem to have better tolerances to measurement noises under some of the circumstances mentioned in the last paragraph, although framework 1 has a slightly better overall tolerance. To better represent the tolerance of each set of rules to measurement noises, a parameter called the accuracy drop rate, which calculated by Equation (4-8) and represents how much success rates will drop if baseline noise ranges are increased by 100% (the half assumed noise ranges are considered as baseline noise ranges), is calculated based on the results shown in Table 4-28. The accuracy drop rate for each step in each framework is shown in Figure 4-16.

$$\text{Accuracy drop rate} = (SR_{half} - SR_{double})/1.5 \quad (4\text{-}8)$$

where SR_{half} *and* SR_{double} denote success rates under half and double assumed ranges respectively.

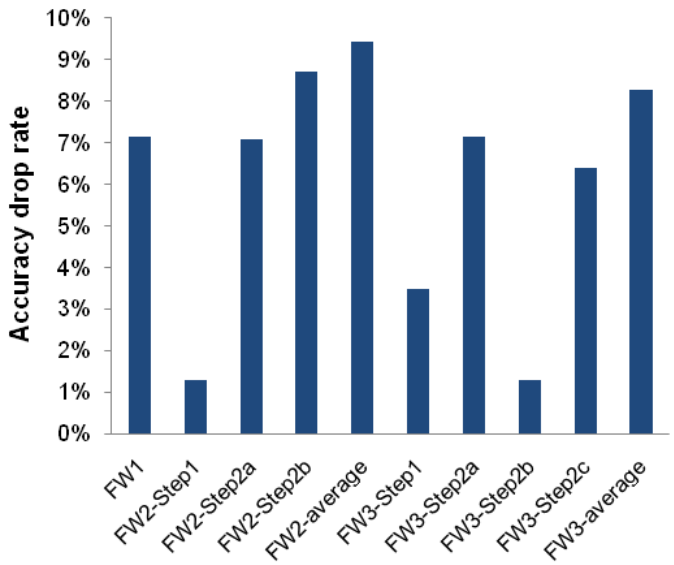

Figure 4-16 Accuracy drop rates

Figure 4-16 provides the evidence that the increase of measurement noise ranges has a direct impact on the effectiveness of each framework. Apart from that, from this figure it can also be seen that rules generated from some steps have a much better tolerance to measurement noises than the rules from Framework 1, although this framework has a better overall tolerance (lower

accuracy drop rate) than the other two frameworks. For example, the accuracy drop rates at step 1 in Framework 2, and at step 1 and step 2b in Framework 3 are higher than that of Framework 1. This further enhances the advantages of Frameworks 2 and 3 under some of the circumstances mentioned in the previous paragraph (i.e. when only step 1 in Framework 2 or in Framework 3 needs to be carried out).

Therefore it can be concluded that although Framework 1 shows the best overall performance in the isolation of single and dual-faults of the turbofan engine being studied, the other two frameworks have their advantages in isolation accuracy under the following circumstances:

(a) When users just want to know whether the real fault is a single or multiple-fault
(b) When the users have already known that the real fault is a single fault and want to identify the faulty component
(c) When the users want to find out whether it is compressor deterioration, turbine deterioration or a combination of compressor and turbine deterioration

In addition, the other two frameworks also have better tolerance to measurement noises under two of the above-mentioned circumstances (i.e. (a) and (c)).

4.3.3.5 Proposed framework

According to the discussion in the previous section, each of the three frameworks has it own advantages under different circumstances. Thus in order to achieve fault isolation with the highest accuracies under any circumstances, all three frameworks should be integrated to form a new framework, as illustrated in Figure 4-17, where Ca, Cb and Cc stand for the circumstance (a), (b) and (c) mentioned in last section and *Else* denotes any other circumstances.

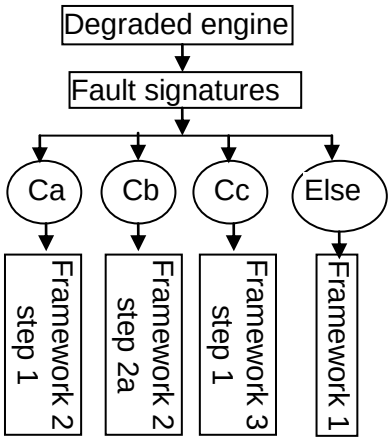

Figure 4-17 Proposed framework

This proposed framework is a combination of the three frameworks. Frameworks 2 or 3 will be used for fault isolation under the circumstances (a), (b), and (c), and framework 1 will be employed otherwise.

4.4 Chapter conclusions

In this chapter, a new gas turbine fault isolation approach based on rough set, which discovers knowledge hidden in engine fault samples, and transfers the knowledge into rules and then uses those rules for fault isolation, as well as three different versions of this fault isolation approach are presented. In addition, a suitable framework was proposed for the developed fault isolation approach by studying three different fault isolation frameworks. On the basis of the research presented in Section 4, several conclusions can be made:

- The first version, using convention fault signatures, is the most effective for single fault isolation in the presence of measurement noises but not effective for the isolation of single and dual faults.
- The second version, using enhanced fault signatures, is able to carry out the isolation of single and dual faults effectively in the presence of measurement noises. However, the number of measurements required by this version for fault isolation can be high.
- The third version is able to provide more flexible measurement selection for fault isolation and hence it is a good replacement for the second version when the measurements required by the latter cannot be satisfied.
- The most suitable framework for the developed fault isolation approach is a combination of all three frameworks being studied; it is more robust and flexible than each of the individual framework as it is able to provide a useful decision tree to users to ensure accuracy of fault isolation.
- The employment of enhanced fault signatures can enhance the characteristics of faults, and make them more discernible, and consequently can: firstly make rough set based fault isolation approaches much more robust in fault isolation when dealing with complex faults, and secondly lead to more fault isolation rules being generated.

It has also been proved that this developed approach has the following advantages:

- It is able to carry out measurement selection for fault isolation, even in the circumstance when the number of measurements allowed to be used for fault isolation is fixed.
- It is capable of generating fault isolation rules very effectively within a short period. This indicates that it can provide expert knowledge to users to understand and solve diagnostics problems. In addition, by employing enhanced fault signatures, more rules can be generated from a certain amount of fault samples when conventional fault signatures are not sufficient to distinguish potential faults.
- It has the potential to be used for on-wing applications since the computational time it requires for fault isolation is very low (average of more than 0.1 seconds is required to carry out fault isolation for each verification sample).
- It is equipped with a good capability to deal with measurement noises; it can achieve high fault isolation accuracy in the presence of measurement noises
- It has low complexity as it does not need any engine performance model in the fault isolation process.

However it also has several shortcomings:

- When the complexity or the number of potential faults increases, its effectiveness in fault isolation reduces.
- Since it discovers fault isolation knowledge from fault samples provided, It can only isolate the faults that are covered by those fault samples.
- Large amounts of fault samples are required to generate fault isolation rules.

Based on the advantages and the shortcomings of the developed rough set based fault isolation approach listed in this section, comparisons between this developed approach and the other fault isolation approaches can be made:

- Among the diagnostics approaches introduced in Section 2.1, ANN, ES, and fuzzy logic are the ones which are able to carry out fault isolation separately (i.e. not together with fault quantification). Compared with these three approaches, the developed approach has two unique features: firstly it can carry out measurement selection for fault isolation and unlike other measurement selection methods which focus more on selecting appropriate measurement for achieving optimal fault quantification the measurement selection provided by this approach is specifically designed for fault isolation; secondly, it can use enhanced fault signatures to generate more rules and hence more expert knowledge.

- On the other hand, all three existing approaches (ANN, ES, and fuzzy logic) and the developed approach share some shortcomings which are the last two shortcomings listed in this section.

5 Conclusions

The objective of this research is to develop one effective gas turbine off-design performance model adaptation approach and also one fault isolation approach which is able to solve all the fault isolation problems listed in Section 1.4. On the basis of the work presented in this thesis, it can be concluded that this research objective has been successfully accomplished. The two developed approaches have a great potential to be used to enhance Engine Health Management systems. In addition, contributions made by the presented research are:

- A new gas turbine off-design performance model adaptation approach was developed. The novelty of this approach is that it uses multiple scaling factors to scale component characteristic maps and the non-linear map scaling can therefore be achieved and hence more effective adaptations can be accomplished.

- The novel development of the rough set based fault isolation approach, which is able to perform measurement selection, rule generation and fault isolation, was accomplished. This approach introduces an algorithm based on a combination of rough set and Boolean reasoning for discretization and attribute reduction. This algorithm is able to carry out discretization with minimal information loss.

- A novel rough set based measurement selection method, which selects measurements based on their fault discernibility and can carry out measurement selection according to the number of measurements users want to use for fault isolation, was created.

- Based on the study of three different frameworks, a new framework for the developed fault isolation approach was proposed. The new framework is a combination of all three frameworks and is more robust and flexible than each of the individual frameworks as it is able to provide a useful decision tree to users to ensure the accuracy of fault isolation.

- Novel fault signatures, called enhanced fault signatures, were created, which enhances the characteristics of component faults and makes them easier to distinguish. The employment of enhanced fault signatures in rough set based fault isolation approaches can result in a significant improvement of isolation accuracy and also a sharp increase in fault isolation rules being generated.

6 Future works

The following recommendations are for future studies:

- Use the developed rough set based fault isolation approach for generating fault isolation rules for fuzzy-logic or expert system based diagnostic approaches, since these diagnostic approaches require fault isolation rules for diagnostics and the developed approach offers a very effective way to generate rules.
- Use the developed fault isolation approach to select appropriate measurements for fault isolation based on fuzzy-logic, ES or ANN.
- Take measurement noises into account while carrying out rule generation; using noisy fault samples to construct decision tables for rule generation, and therefore fault isolation using the generated rules may have higher accuracy in the presence of measurement noises.
- Fully utilise all available enhanced fault signatures to generate more rules as it has been proved that more rules can lead to higher fault isolation accuracy. In this research, if enhanced fault signatures were used for fault isolation, attribute reduction was carried out to select appropriate enhanced fault signatures for isolation and the unselected signatures were not used for rule generation. Hence it would be worth trying to use the unselected signatures to generate extra rules.
- Set benchmarks for good fault isolation and compare the developed rough set based fault isolation approach with other similar approaches based on the benchmarks.

REFERENCES

[1] Volponi, A. and Wood, B., "Engine health management for aircraft propulsion systems" Integrated System Health Engineering and Management (ISHEM) 2005 Conference Proceedings, November 7-10, 2005.

[2] Li, Y. G., “Gas turbine diagnostics”, Lecture note, Department of Power and Propulsion, School of Engineering, Cranfield University, Oct, 2006.

[3] Urban, L.A., “Gas turbine engine parameter interrelationships”, Hamilton Standard Division of United Aircraft Windsor Locks, USA.

[4] Crosby, J.K., “Factors relating to deterioration based on Rolls-Royce RB211 in-service performance”, Turbomachinery Performance Deterioration, vol. 37, pp. 41-47, 1986.

[5] Li, Y.G, “Gas turbine performance and health status estimation using adaptive gas path analysis”, ASME GT2009-59168, also ASME Journal of Engineering for Gas Turbines and Power, Vol. 132, 041701, April 2010.

[6] Dorronsoro, J. R., López, J., Cruz, C. S. and Sigüenza, J. A., “Autoassociative neural networks and noise filtering”, IEEE Transactions on Signal Processing, vol. 51, no. 5, pp. 1431-1438, May, 2003.

[7] Stamatis, A., Mathioudakis, K. and Papailiou, K., "Optimal Measurement and Health Index Selection for Gas Turbine Performance Status and Fault Diagnosis", Journal of engineering for gas turbines and power, vol. 114, no. 2, pp. 209-216, 1992.

[8] Provost, M.J., “The use of optimal estimation techniques in the analysis of gas turbine”, PhD Thesis, School of Engineering, Cranfield University.

[9] Ogaji, S. O. T., Sampath, S., Singh, R. and Probert, S.D., "Parameter Selection for Diagnosing a Gas Turbine Performance-Deterioration", Journal of Applied Energy, Elsevier, vol. 73, pp. 25-46, 2002.

[10] Sowers, T. S., Kopasakis, G. and Simon, D. L., "Application of the Systematic Sensor Selection Strategy for Turbofan Engine Diagnostics", ASME GT2008-50525 ASME Turbo Expo, 2008.

[11] Jasmani, m. s., Li, y. g. and Ariffin, z., “Measurement selections for multi-component gas path diagnostics using analytical approach and measurement subset concept”, ASME GT2010-22402, ASME Turbo Expo, 2010.

[12] Doel, D.L., “An assessment of weighted-least-squares-based gas path analysis”, ASME Journal of Engineering for Gas Turbines and Power, vol. 116, no.1, pp. 366-373, 1994

[13] Barwell, M.J., “Compass-Ground based engine monitoring program for general application”, SAE technical paper series, 871734, Aerospace Technology Conference and Exposition, Long Beach, California, October 1987.

[14] Doel, D.L., “TEMPER-a gas-path analysis tool for commercial jet-engines”, ASME Journal of Engineering for Gas Turbines and Power, vol. 116, pp. 82-89, 1994.

[15] Stamtic, A., Mathioudakis, K., Smith, M. and Papiliou, K., “Gas turbine component fault identification by means of adaptive performance modelling”, ASME paper 90-GT-376, 1990.

[16] House, P., “Gas path analysis techniques applied to turboshaft”, MSc thesis, Cranfield University, England, 1992.

[17] Escher, P.C, “Pythia: An object-orientated gas path analysis computer program for general applications”, PhD thesis, Cranfield University, 1995

[18] Zedda, M., “Gas turbine engine and sensor fault diagnosis”, PhD Thesis, School of engineering, Cranfield University.

[19] Ogaji, S.O.T. and Singh, R., “Gas path fault diagnosis framework for a three-shaft gas turbine”, IMechE Journal of Power and Energy, vol. 217, pp.149-157, 2003.

[20] Marinai, L.M. and Singh, R., “Fuzzy-logic-based diagnostic process for turbofan engines”, ISABE-2003-1149, XVI International Symposium on air breathing engines, August 2003.

[21] Romessis, C., Stamatis, A. and Mathioudakis, K., “Setting-up a belief network for turbofan diagnosis with the aid of an engine-performance model”. ISABE-2001-1032, 15th International Symposium on air breathing engines, September 2001.

[22] Torella, G. and Torella, R., Probabilistic expert systems for the diagnostics and trouble-shooting of gas turbine apparatuses. In: AIAA 99-2942, 1999.

[23] Urban, L.A., “Gas path analysis applied to turbine engine condition monitoring”, AIAA-72-1082, 1972.

[24] Kalman, R. E., “A new approach to linear filtering and prediction problems”, Transaction of the ASME Journal of Basic Engineering, no. 82, series D, pp. 35-45, March, 1960.

[25] Marinai, L., Probert, D., and Chrunock, B., “Prospects for aero gas turbine diagnostics: a review”, Applied Energy, vol. 79, issue 1, pp. 109-126, September 2004

[26] Doel, D.L., “Interpretation of Weighted-Leas-Squares gas path analysis results”, ASME Journal of Engineering for Gas Turbines and Power, Vol. 125, pp. 624-633, July 2003

[27] Zedda, M. and Singh, R., “Gas turbine engine and sensor fault diagnosis”, ISABE 99-7238, 13th International Symposium on air breathing engines, 1999.

[28] Gulati, A., Taylor, D. and Singh, R., “Multiple operating point analysis using genetic algorithm optimization for gas turbine diagnostics”, ISABE-2001-1139, 15th International Symposium on air breathing engines, September, 2001.

[29] Doel, D.L., “The role for expert systems in commercial gas turbine engine monitoring”The Gas Turbine and Aero-engine Congress and Exposition, ASME 90-GT-374, June 1990.

[30] Torella, G., “Expert systems for the simulation of turbofan engines”, ISABE 93-7133, 11th International Society for Air Breathing Engines, 1993.

[31] Winston, H., Sirag, D., Hamilton, T., Smith, H., Simmons, D. and Ma, P., “Integrating numeric and symbolic processing for gas path maintenance”, AIAA-91-0501, 1991.

[32] Dundas, R.E., Sullivan, D.A., and Abegg, F., “Performance monitoring of gas turbines for failure prevention”, ASME 92-GT-267, 1992.

[33] Li, Y.G., “Performance analysis based gas turbine diagnostics: a review”, ASME Journal of Engineering for Gas Turbines and Power, vol. 216, no. 5, pp. 363-377, 2002.

[34] Pawlak, Z. and Skowron, A., “Rudiments of rough sets”, Information Sciences, Vol. 177, No. 1, Jan, 2007, pp.3-27.

[35] Jensen, R., Shen, Q. and Tuson, A., "Finding Rough Set Reducts with SAT," Proceedings of the 10th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing, LNAI 3641, pp. 194-203, 2005.

[36] Chen, T. and Sun, J-G., “Rough set and neural network based fault diagnosis for aeroengine gas path”, ASME GT2005-68192, ASME Turbo Expo 2008, June 6-9, 2004.

[37] Szczuk, M., “Rough set and artificial neural networks”, In: rough sets in knowledge discovery 2: Applications, Case studies and software systems, Physica-Verlag, Heidelberg, pp. 449-470, 1998.

[38] Wroblewski, J., “Finding minimal reducts using genetic algorithm (extended version)”, Warsaw university of Technology, Institute of Computer Science, Reports-16-95.

[39] Dietz, W. E., Kiech, E. L. and Ali, M., " Jet and rocket engine fault diagnosis in real time", Journal of Neural Network Computing, vol.1, no.1, pp. 5-18, 1989.

[40] Denny, G., "F-16 Jet engine trending and diagnostics with neural networks", Applications of Neural networks, vo. 4, pp419-422, Sep 1993.

[41] Torella, G. and Lombardo, G., "Utilization of neural networks for gas turbine engines", ISABE 95-7032, 12th International Symposium on air breathing engines, 1995

[42] Eustace, R., "Neural network fault diagnosis of a turbine engine", ISABE 93-7091, 11th International Symposium on air breathing engines, Sept. 19993.

[43] Eustace, R. and Frith, P.C. W., "Utilising repair and overhaul experience in a probabilistic neural network for diagnosing gas-path faults", ISABE-2001-1050, 15th International Symposium on air breathing engines, Sept. 2001.

[44] Roemer, M. J., "Testing of a real-time health monitoring and diagnostic system for gas turbine engines", AIAA-98-3603, 1998.

[45] Marinai, L., "Gas-path diagnostics and prognostics for aero-engines using fuzy logic and time series analysis", PhD Thesis, Department of Power and Propulsion, Cranfield University, 2004.

[46] Tang, G., Yates, G. L. and Zhang, J. and Chen, D., "A practical intelligent system for condition monitoring and fault diagnosis of jet engines", AIAA 99-2533, 1999.

[47] Siu, C., Shen, Q., and Milne, R., "TMDOCTOR: a fuzzy rule-and case-based expert system for turbomachinery diagnosis", Proceedings of IFAC Fault Detection, Supervision and safety for technical processes, vol. 2, Aug. 1997.

[48] Ganguli, R., "Application of fuzzy logic for fault isolation of jet engines", ASME 2001-GT-0013, ASME Turbo Expo 2001, June, 2001.

[49] Karvounis, G., and Frith, O., “Automated detection of engine health using huybrid model-based and fuzzy-logic approaches”, ISABE-2003-1232, XVI International Symposium on air-breathing engines, Aug, 2003.

[50] Volponi, A., DePold, H., Ganguli, R. and Daguang, C., “The use of Kalmn-filter and neural-network methodologies in gas-turbine diagnostics: a comparative study”, ASME 2000-GT-547, ASME Turbo Expo 2000, May, 2000.

[51] Sun, B., Zhang, J. and Zhang, S., “An investigation of artificial neural network (ANN) in quantitative fault diagnosis for turbofan engine”, ASME 2000-FT-32, ASME Turbo Expo 2000, May, 2000.

[52] Kobayashi, T. and Simon, D. L., “A hybrid neural network-genetic algorithm technique for aircraft engine performance diagnostics”, AIAA-2001-3763, 2001.

[53] Green, A., “Artificial-intelligence for real-time diagnostics and prognostics of gas-turbine engines”, AIAA 97-2899, July, 1997.

[54] Breese, J. S., Gay, R. and Quentin, G. H., “Automated decision0analysitc diagnosis of thermal performance in gas turbines”, ASME 94-GT-317, 1994.

[55] Pear, J., “Probabilistic reasoning in intelligence systems”, Morgan Kaufman Publishers Inc, San Francisco, CA, 1988.

[56] Torella, G., “Expert systems and neural networks for fault isolation in gas turbines”, ISABE 97-7148, 13th International Symposium on air breathing engines, Sept, 1997.

[57] Stamatis, A., Mathioudakis, K. and Papailiou, K. D., “Adaptive Simulation of Gas Turbine Performance”, ASME Paper 90-GT-205, Journal of Gas Turbine and Power, Vol. 112, No. 2, pp.168-175, 1990.

[58] Lambiris, B., Mathioudakis, K. and Papailiou, K. D., “Adaptive modeling of jet engine performance with application to condition monitoring”, ISABE 91-7058, 1991.

[59] Kong, C. D., and Ki, J. Y., 2003, “A new scaling method for component maps of gas turbine using system identification,” ASME J. Eng. Gas Turbines Power, 125_4_, pp. 979–985.

[60] Kong, C, Kho, S. and Ki, J., “Component map generation for a gas turbine using genetic algorithms”, ASME GT2004-53736, ASME TURBO EXPO 2004, June 2004.

[61] Li, Y. G., Pilidis, P. and Newby, M., “An adaptation approach for gas turbine design-point performance simulation”, ASME Transactions: Journal of Engineering for Gas Turbine and Power, Vol. 128, pp.789-795, October 2006.

[62] Roth, B. R., Mavris, D., Doel, D. L. and Beeson, D., “High-accuracy matching of engine performance models to test data”, ASME GT2003-38784, ASME TURBO EXPO 2004, June 2004.

[63] Roth, B. A., Doel, D. L. and Cissell, J. J., “Probabilistic matching of turbofan engine performance models to test data”, ASME Turbo Expo 2005, GT2005-68201, 2005.

[64] Lo Gatto, E., Li, Y.G., Pilidis, P., “Gas turbine off-design performance adaptation using genetic algorithm”, ASME GT2006-90299, TURBO EXPO May 2006

[65] Marinai, L., Li, Y. G. and E., Pachidis, V, “Multiple point adaptive performance simulation tuned to aero-engine test bed data”, ISABE-2007-1117, XVI International Symposium on Air Breathing Engines, 2005

[66] Li, Y. G., Marinai, L., Lo Gatto, E., Pachidis, V. and Pilidis, P., “Multiple point adaptive performance simulation tuned to aerospace test-bed data”, Transactions of the AIAA, Journal of Propulsion and Power, Vol. 25, No. 3, p. 635, May-June, 2009

[67] Li, Y. G. and Singh, R., “An advanced gas turbine gas path diagnostic system – PYTHIA”, ISABE-2005-1284, XVII International Symposium on Air Breathing Engines, Munich, Germany, 2005.

[68] Øhrn, A., Komorowski, J., Skowron, A. and Synak, P., “The design and implementation of a knowledge discovery toolkit based on rough sets: The ROSETTA System”, In Rough Sets in Knowledge Discovery 1: Methodology and Applications, L. Polkowski and A. Skowron (eds.), Studies in Fuzziness and Soft Computing, Vol. 18, Chapter 19, pp. 376-399, Physica-Verlag. ISBN 3-7908-1119-X.

[69] Øhrn, A., “ROSETTA technical reference manual”, Department of Computer and Information Science, Norwegian University of Science and Technology (NTNU), Trondheim, Norway, 2000.

[70] Pawlak, Z., Skowron, A., “Rough sets and Boolean reasoning”, Information Sciences, Vol.177, No. 1, Jan, 2007, pp. 41-73.

[71] Bazan, J., Nguyen, H. S. Nguyen, S. H., Synak, P. and Wróblewski, J., “Rough set algorithms in classification problem”, Rough Set Methods and Applications, Physica-Verlag, Heidelberg New York, pp. 49–88, 2000.

[72] Bazan, J., Szczuka, M., “RSES and RSESlib – a collection of tools for rough set computations”, Lecture Notes in Artificial Intelligence 3066, 592–601, Berlin, Heidelberg: Springer-Verlag, 2000.

[73] http://logic.mimuw.edu.pl/∼rses

[74] Ogaji, S., Li, Y.G., Sampath, S. and Singh, R., “Gas-path fault diagnosis of a turbofan engine from transient data using artificial neural networks”, ASME GT2003-38423, ASME Turbo Expo, Atlanta, Georgia, USA,16–19 June 2003.

APPENDICES

Appendix A Compressor maps

A.1 Map configuration

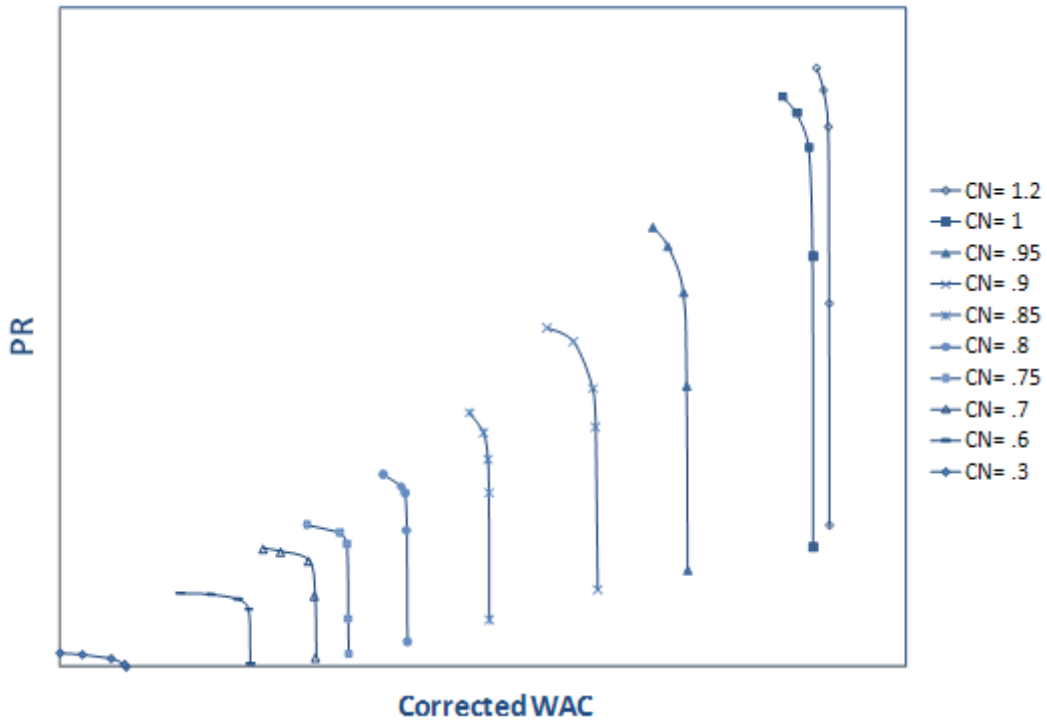

Figure A -6-1 “Inaccurate” compressor map-part 1

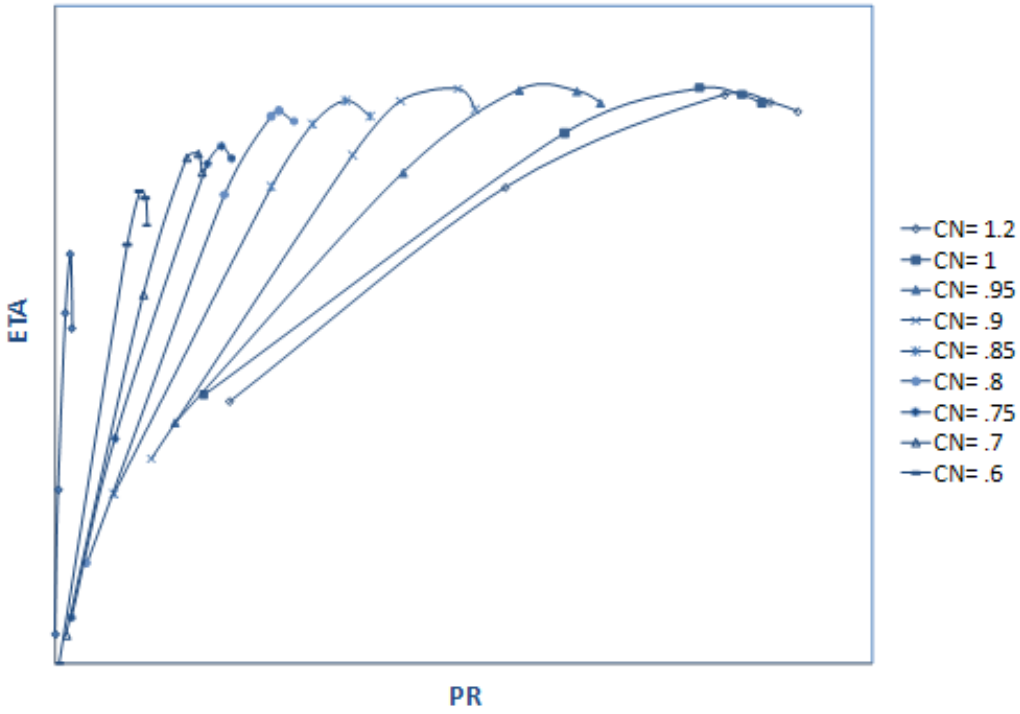

Figure A -6-2 “Inaccurate” compressor map-part 2

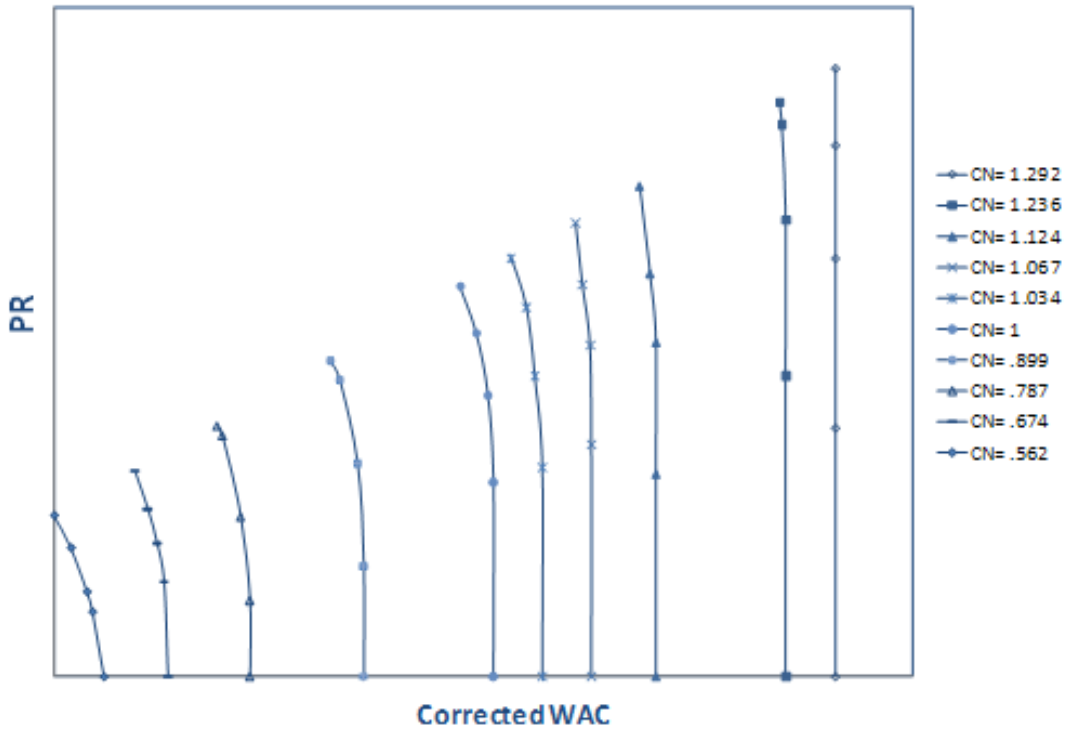

Figure A -6-3 “Accurate” compressor map-part 1

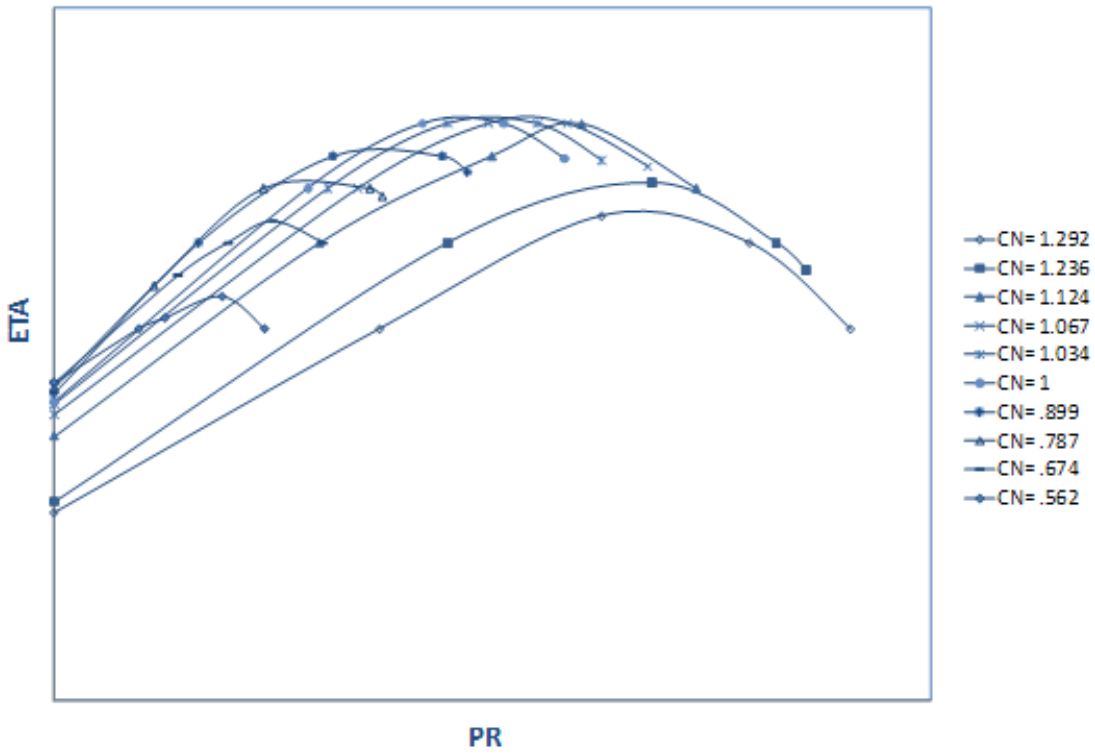

Figure A -6-4 “Accurate” compressor map-part 2

Appendix B Adaptation results

B.1 Searching results from the GA

Table A -6-1 Searching results for speed line with CN 0.90

		Scaling factor searching range		Searching result	Fitness
		minimum	maximum		
Try1	ETA	0.990	1.010	0.990	0.079
	WAC	0.990	1.010	1.010	
	PR	0.990	1.010	1.010	
Try2	ETA	0.951	1.000	0.951	0.122
	WAC	1.000	1.049	1.049	
	PR	1.000	1.049	1.049	
Try3	ETA	0.921	0.971	0.922	0.198
	WAC	1.029	1.079	1.078	
	PR	1.029	1.079	1.079	
Try4	ETA	0.902	0.951	0.902	0.307
	WAC	1.049	1.098	1.098	
	PR	1.049	1.098	1.098	
Try5	ETA	0.882	0.931	0.882	0.517
	WAC	1.069	1.118	1.117	
	PR	1.069	1.118	1.115	
Try6	ETA	0.862	0.912	0.903	0.780
	WAC	1.088	1.138	1.136	
	PR	1.088	1.138	1.131	
Try7	ETA	0.872	0.921	0.908	0.978
	WAC	1.098	1.147	1.146	
	PR	1.098	1.147	1.127	
Try8	ETA	0.872	0.921	0.908	0.979
	WAC	1.118	1.167	1.145	
	PR	1.098	1.147	1.126	

Table A -6-2 Searching results for speed line with CN with 0.95

		Scaling factor searching range		Searching result	Fitness
		minimum	maximum		
Try1	ETA	0.995	1.005	0.995	0.132
	WAC	0.995	1.005	1.005	
	PR	0.995	1.005	1.005	
Try2	ETA	0.976	1.000	0.976	0.181
	WAC	1.000	1.024	1.024	
	PR	1.000	1.024	1.024	
Try3	ETA	0.961	0.986	0.962	0.245
	WAC	1.014	1.039	1.038	
	PR	1.014	1.039	1.038	
Try4	ETA	0.952	0.976	0.952	0.323
	WAC	1.024	1.048	1.048	
	PR	1.024	1.048	1.048	
Try5	ETA	0.942	0.966	0.942	0.420
	WAC	1.034	1.058	1.058	
	PR	1.034	1.058	1.058	
Try6	ETA	0.933	0.957	0.934	0.557
	WAC	1.043	1.067	1.067	
	PR	1.043	1.067	1.062	
Try7	ETA	0.928	0.952	0.935	0.624
	WAC	1.048	1.072	1.072	
	PR	1.048	1.072	1.054	
Try8	ETA	0.928	0.952	0.952	0.815
	WAC	1.058	1.082	1.082	
	PR	1.048	1.072	1.070	
Try9	ETA	0.928	0.952	0.951	0.957
	WAC	1.063	1.087	1.087	
	PR	1.053	1.077	1.061	
Try10	ETA	0.937	0.961	0.951	0.979
	WAC	1.072	1.096	1.088	
	PR	1.053	1.077	1.059	

Table A -6-3 Searching results for speed line with CN with 1.2

		Scaling factor search range		Searching result	Fitness
		minimum	maximum		
Try1	ETA	0.980	1.020	0.980	0.046
	WAC	0.980	1.020	1.020	
	PR	0.980	1.020	1.020	
Try2	ETA	0.899	1.000	0.901	X
	WAC	1.000	1.101	X	
	PR	1.000	1.101	X	
Try3	ETA	0.980	0.899	X	0.223
	WAC	1.101	1.202	1.202	
	PR	1.101	1.202	1.137	
Try4	ETA	0.737	0.838	X	X
	WAC	1.162	1.263	X	
	PR	1.101	1.202	X	
Try5	ETA	0.798	0.899	0.867	0.468
	WAC	1.162	1.263	1.263	
	PR	1.101	1.202	1.182	
Try6	ETA	0.798	0.899	X	X
	WAC	1.202	1.303	X	
	PR	1.162	1.263	X	
Try7	ETA	0.798	0.899	X	X
	WAC	1.182	1.283	X	
	PR	1.121	1.223	X	
Try8	ETA	0.798	0.899	X	X
	WAC	1.182	1.283	X	
	PR	1.101	1.202	X	
Try9	ETA	0.838	0.939	X	X
	WAC	1.182	1.283	X	
	PR	1.101	1.202	X	
Try10	ETA	0.798	0.899	0.870	0.563
	WAC	1.172	1.273	1.273	
	PR	1.101	1.202	1.175	
Try11	ETA	0.838	0.899	X	X
	WAC	1.182	1.283	X	
	PR	1.121	1.182	X	

Appendix C Map scaling

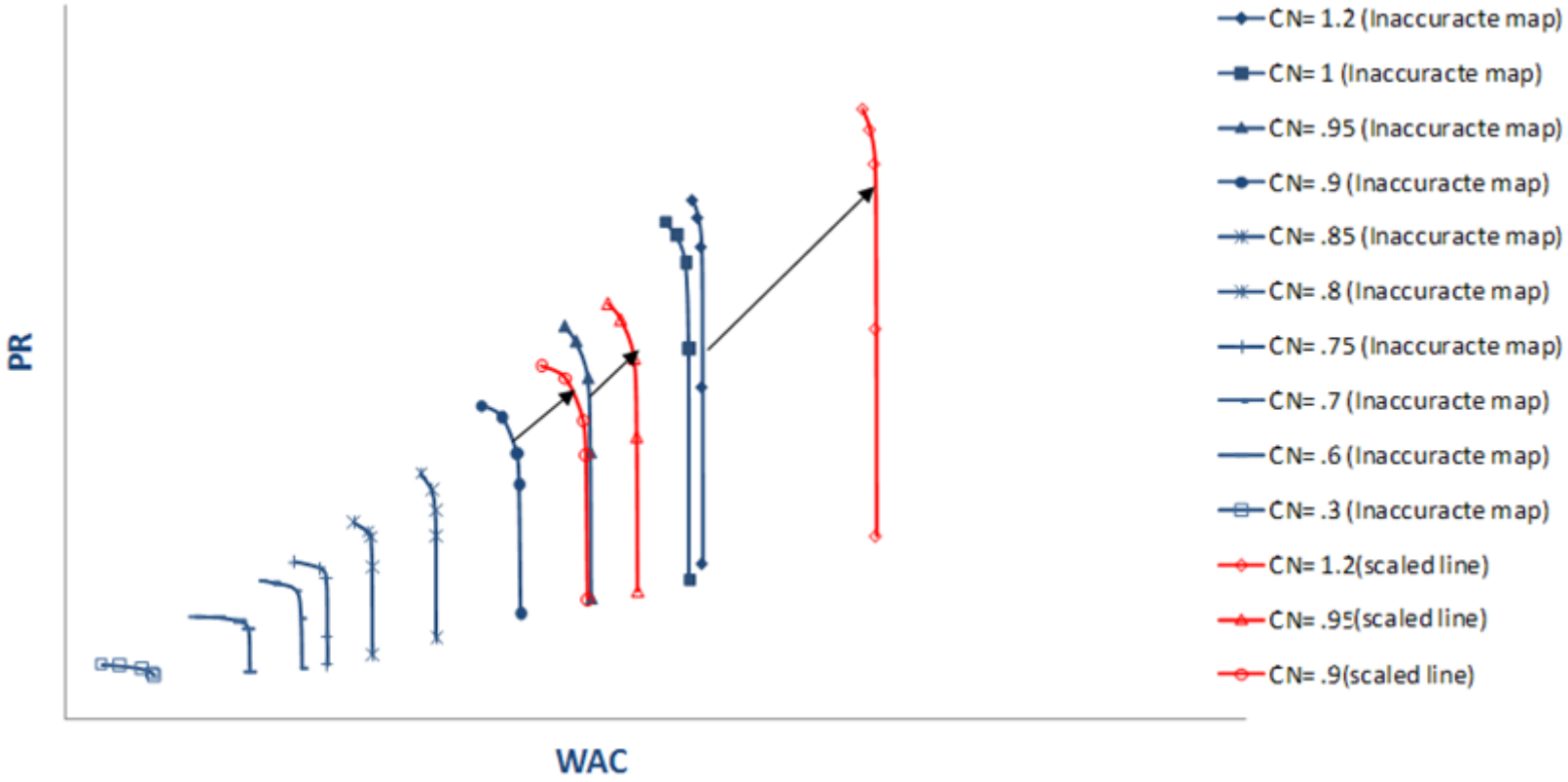

Figure A 6-5 Scaling of speed lines with CNs 1.2, 0.95, and 0.9 –part 1

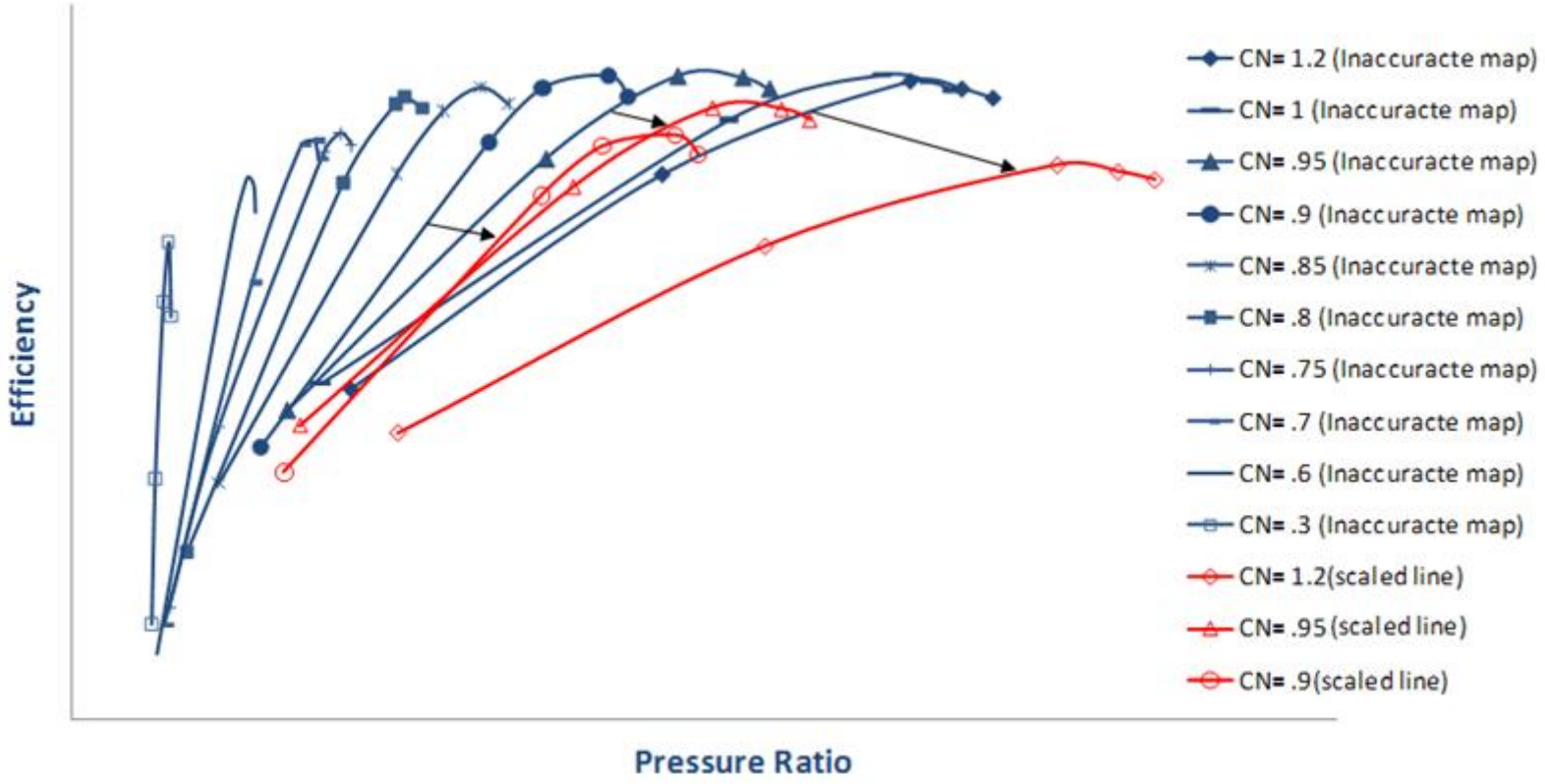

Figure A 6-6 Scaling of speed lines with CNs 1.2, 0.95, and 0.9 –part 2

Appendix D Examples of fault isolation rules

Note each of the fault isolation rules listed below does not contain any discrete values as all the discrete values have been replaced by their corresponding intervals.

Case1

If $\triangle T3 \in (0.05,\infty)$, and $\triangle N2 \in (-\infty,0.05)$

→ Fan Fault

If$\triangle T5 \in (-\infty,-0.25)$, and $\triangle P8 \in (-\infty,-0.15)$., and $\triangle N2 \in (-\infty,-0.05)$

→LPC Fault

If$\triangle P5 \in (0.25, \infty)$, and $\triangle T6 \in (0.05, \infty)$, and $\triangle FF \in (-\infty,-1.35)$

→ HPC Fault

If$\triangle P5 \in (0.25, \infty)$, and $\triangle T6 \in (-\infty,-0.05)$, and $\triangle P8 \in (-\infty,-0.15)$

→ HPT Fault

If$\triangle T3 \in (-\infty,-0.05)$, and $\triangle P5 \in (-\infty,0.25)$, and $\triangle T6 \in (0.06, \infty)$, and $\triangle N2 \in (-0.05, \infty)$ →LPT Fault

Case2

If$\triangle T6 \in (0.05,\infty)$, and R(P3P9) $\in (-0.05,\infty)$, and R(P5P6) $\in (-0.05, \infty)$, and R(P8T5) $\in (-0.45, \infty)$ → Fan Fault

If$\triangle T6 \in (-\infty,-0.15)$, and R(P3P9) $\in (-0.05,\infty)$, and R(P5P6) $\in$ $(-\infty,-0.05)$

→ LPC Fault

If$\triangle T6 \in (-\infty,-0.15)$, and R(P3P9) $\in (-\infty,-0.05)$, and R(P5P6) $\in$ $(-0.05,\infty)$

→ HPC Fault

If R(P3P9) ∈ (-0.05,∞), and R(P5P6) ∈ (-∞,-0.05), and R(P3FF) ∈ (-∞,0.15)

→ HPT Fault

If R(P3P9) ∈ (-∞,-0.05), and R(P5P8) ∈ (-∞,-0.15), and R(P8T5) ∈ (-0.45,∞) →

LPT Fault

Case3

If△P5 ∈ (11.75,∞), and △T6 ∈(0.65,2.25), and △P9 ∈(-1.05,∞)

→ Fan & HPC Fault

If△T3 ∈ (-1.15,-0.85), and △P5 ∈(11.75,∞), and △T8 ∈(-0.05,∞), and △T9 ∈(-

∞,0.05) → HPC & HPT Fault

If△T6 ∈ (-1.55,-1.25), and △T9 ∈(0.05,∞), and △FF ∈(2.75,0.45)

→ HPT & LPT Fault

Case4

If△T3 ∈ (-0.05,∞), and R(P3P5) ∈(-0.45,∞), and R(P6P9) ∈ (-∞, -0.05), and

R(P9T9) ∈ (-2.95,-∞) →Fan & LPT Fault

If△T6 ∈ (0.45,∞), and △P6 ∈(-∞,0.05), and R(P3P5) ∈ (-∞, -0.45), and R(T5T9) ∈

(-∞,0.05) →HPC & LPT Fault

If△T6 ∈ (-∞,0.25), and R(P6P8) ∈ (0.05,∞), and R(FFN2) ∈ (0.65,∞), and

R(T5T9) ∈ (-∞,0.05) →HPT & LPT Fault
