NOTE TO USERS

This reproduction is the best copy available.

AN EVOLVING NEURAL FUZZY CLASSIFIER FOR MACHINERY DIAGNOSTICS

By

Ofelia Antonia Jianu

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF MASTER OF SCIENCE

IN

CONTROL ENGINEERING

FACULTY OF ENGINEERING

LAKEHEAD UNIVERSITY

THUNDER BAY, ONTARIO, CANADA, 2010

Your file Votre référence
ISBN: 978-0-494-71956-5
Our file Notre référence
ISBN: 978-0-494-71956-5

Abstract

The classical techniques for fault diagnosis require periodic shut down of machines for manual inspection. Although these techniques can be used for fault diagnosis in simple machines, they can rarely be used effectively for complex ones. Due to the rapid growing market competitiveness, more reliable and robust condition monitoring systems are critically needed in a wide array of industries to improve production quality and reduce cost. As a result, in recent years more efforts have been taken to develop intelligent techniques for online condition monitoring in machinery systems. Several neural fuzzy classification schemes have been proposed in literature for fault detection. However, the reasoning architecture of the classical neural fuzzy classifiers remains fixed, allowing only the system parameters to be updated in pattern classification operations. To improve the reliability of machinery fault diagnostics, an evolving fuzzy classifier is developed in this work for gear system condition monitoring. The evolution is performed based on the comparison of the potential of the incoming data set and the existing cluster centers. One key feature of the developed evolving fuzzy classifier is that it has the ability of developing continuously - by adding or subtracting rules and by modifying existing rules and parameters. In performance evaluation, the proposed evolving classifier is firstly tested with the use of benchmark data sets, such as Iris data, Wisconsin breast cancer data and wine data. Then the adopted evolving classifier is implemented for gear fault diagnosis. A distinguishable pattern is determined between the input data and the output patterns to evaluate the data sets. Several signal processing techniques are utilized to generate representative features to train the proposed evolving fuzzy classifier. Simulation test

results show that the proposed classifier can effectively identify the condition of a gear, both spur and helical types, and it outperforms provide other related methods.

Acknowledgment

Walking the road to success alone is always hard, if not impossible. For this reason I would like to express my gratitude to the many people that stood by me during my walk to completing my M.Sc. in Engineering.

First and foremost, I am thankful to my supervisor, Dr. Wilson Wang, whose guidance and support from the initial to the final level enabled me to develop an understanding of the subject. Also, a sincere thank you to Dr. M. Uddin, my co-supervisor.

Second, I would like to thank Dr. Laurie Garred, Dr. K. Liu, Dr. Meilan Liu and Ms. Heather Moynihan for their mentoring and kindness throughout these years.

Third, a special thanks to my parents, Angela and Radu, as well as my brother, Ovidiu, for standing by my every step of the way. In my upbringing, they always emphasized on the importance of education therefore my success is partially dedicated to them.

Forth, a heartfelt mention goes to Gianmarc. His professional and emotional encouragements made me stronger and more determined in reaching my goal.

Lastly, I offer my regards to all of those who supported me in any respect during the completion of the thesis.

Table of Contents

List of Figures

List of Tables

List of Acronyms

ANFIS:	adaptive neuro-fuzzy inference system
CF:	crest factor
DEKF:	decoupled-extended Kalman filter
e^2TS:	enhanced-evolving Takagi-Sugeno
EFS:	evolving fuzzy system
EO:	energy operator
eTS:	evolving Takagi-Sugeno
ie2TS:	improved-enhanced-evolving Takagi-Sugeno
Kurt:	kurtosis
MF(s):	membership function(s)
NF:	neuro-fuzzy
NN:	neural network
RLSE:	recursive least squares estimate
RMSE:	root mean square error
RMSV:	root mean square value
sDEKF:	scaled decoupled extended Kalman filter
TS:	Takagi-Sugeno
TSA:	time synchronous average

Chapter 1

1.0 Introduction

"There is a great satisfaction in building good tools for other people to use."

FREEMAN DYSON

1.1 Motivation of the Research

Gear trains are widely used in various mechanism and machines, from watches to helicopters. Their malfunction can lead to costly shutdowns, lapses in production, and even human casualties. The detection of faults in mechanical systems, more precisely in gears, is of great interest nowadays. Many accidents can be prevented when a fault is detected at the right time, costs associated with repairs of machinery can be reduced and inspection time of gear trains to find the faulty component is greatly reduced when an efficient gear fault detection technique is implemented.

There are two different classes of gear defects known in literature: distributed and localized defects [1]. Two forms of distributed faults are known: the adhesive wear and the abrasive wear [1]. The transfer of metal particles from one tooth to a mating tooth by a welding action characterizes the adhesive wear whereas the abrasive wear is induced by the presence of abrasive particles in the meshing process. A localized defect or rolling contact fatigue faults is a critical fault such as pitting, cracking and scoring [2].

Although both classes induce transmission errors and increase the noise and vibration in gearboxes, researchers give more attention to the latter of the two. That is because localized gear defects can cause sudden failure of the mechanical system, whereas the distributed faults are usually initiated with a localized fault. As a result, this work will focus on localized faults, as they can cause catastrophic failures in machines such as helicopters and airplanes.

1.1.1 Pitting

Gears are one example of solids bodies in relative motion coming into contact. The resulting normal and tangential forces at the contact region are transmitted from a surface to its pair [2]. When the contact stress is higher than the endurance limit of the gear material, a type of fault referred to as pitting occurs. Pitting is a surface fault characterized by pieces of material detaching at the contact surface of the tooth due to fatigue. In their research [2], Aslants and Tasgetiren have found that for an as-cast specimen, a pitting size of $500\,\mu m$ can develop after 1.850×10^6 cycles. However, gear materials are generally subjected to heat treatment to prevent surface fatigue failures such as pitting.

1.1.2 Tooth Breaking

Another case of fatigue failure is a tooth crack of break. A tooth break usually originates as a crack and propagates until the tooth breaks off. Once the tooth breaks off, the subsequent tooth will exhibit higher impact loading therefore it is susceptible to breakage. If undetected, this failure can cause serious damage in the operating machine.

1.1.3 Scoring

Although scoring is another example of contact surface fault, it is primarily related to incorrect mounting, improper lubrication and overloading. Scoring arises when the lubrication film breaks down and the gear teeth are in direct metal-to-metal contact. Due to the high temperatures experienced by the teeth surfaces, they exhibit welding. As the gear rotates, this welded spots on the teeth surfaces break off. This type of fault can also cause severe damage to the gearbox and consequently to the operating machine.

As a result, the focus of this work is to develop an intelligent condition monitoring scheme for gearbox diagnosis. Before introducing the proposed scheme, an understanding of classical gear condition monitoring techniques is necessary.

1.2 Gear Condition Monitoring Techniques

The aforementioned faults arise during the operation of a gear train. It is crucial that these defects are detected at an early stage without machinery disassembly in order to reduce downtime costs and prevent catastrophic failures. Several methods are used for the diagnosis of gear trains. Based on signal properties, these methods are broadly classified as acoustic measurements, temperature monitoring, wear debris detection, and vibration analysis [3].

1.2.1 Acoustic Measurement

One of the most effective acoustic-based health monitoring in rotating machinery, including gearboxes, is acoustic emission. The detection of cracks is the prime application of acoustic emission. The measurement of a machine's sound can also be employed for

detecting defects in gearboxes. Typically, the accuracy of these methods depends on sound pressure and sound intensity data [3].

1.2.2 Temperature Monitoring

Frictional heating from contacts of gear teeth is of extreme importance for monitoring the condition of a gear transmission under its continuing operation. The surface temperature holds the critical information about the condition of a gear [3].

1.2.3 Wear Debris Analysis

In this method, the presence of any metallic particles in the lubricant is detected with the aid of sensors [3]. Wear particles are considered to be a critical alarm indicating the need to change the gear before a forced outage occurs.

1.2.4 Vibration Measurement

Vibration is a symptom of an internal defect. In fact, it is a very sensitive and early predictor of a developing fault. A vibration signal indicating a fault is generated by the interaction between gears regardless of the defect type. Consequently, a vibration analysis can be employed for the diagnosis of all types of faults, either localized or distributed. Furthermore, low-cost sensors, accurate results, simple setups, specific information on the damage location, and comparable rates of damage are other benefits of the vibration measurement method [3]. It is for this reason that vibration analysis is widely employed in the industry, which also will be used in this work.

1.3 Vibration Based Condition Monitoring

Signal processing is the process in which representative features are extracted from the collected vibration signals. There are many signal processing techniques proposed in

literature for gear fault detection. Based on processing tools, these techniques can be classified into three categories, that is, methods in the time-domain, frequency domain, and time-frequency domain, respectively.

Condition monitoring based on vibration analysis can monitor all parts of a gearbox (i.e. bearings, shafts and gears); however, the focus of this thesis is directly related to gears. In order to focus only on the important part of the vibration signal, the time domain signal is synchronously averaged. This signal average is then used for advanced analysis to compute condition indicators (CIs), which are compared in the decision-making unit.

Condition indicators have been introduced in 1977 when Stewart developed FM0, FM4 and some other techniques [4]. Since then, research had progressed and Zakrejsek [5] introduced NA4. NA4 has further been improved by Decker [6] and then by Demsey [7] for a better performance related to torque changes. One of the many techniques available for detecting gear faults such as cracks, files and chips is the order cepstrum analysis. In order to process signals using this procedure, vibration signals are collected at constant time increments in time domain and then data is resampled in angle-domain such that the non-stationary signals are changed into stationary signals. Order cepstrum is then implemented for accurate diagnosis [8]. Suppose data is collected at variable rotational speeds in time-domain. As the rotational speed changes, the sampling dots change generating a "frequency ambiguity". This ambiguity does not occur when data is sampled in angle-domain. The sampling in the angle-domain is called order tracking and it has advantages over the traditional spectral analysis since it samples signals at constant angle generating constant sampling dots and thus changing the non-stable signal to a stable one. The necessary assumption for this procedure is that the acceleration is constant. In

literature, autoregressive (AR) models have been established for signal analysis from a monitored gearbox under healthy conditions [9]. This model was used as a prediction filter that filtered future signals of the same gearbox. A discrete time signal $x(n)$ can be represented as a regression on itself plus an approximation error; this process is called autoregression [10]. This model is used as a linear predictor, generating AR residual signals when the filtered signals are subtracted from the original signal. In the case in which the gears in the gearbox are healthy, the residual signal is the prediction error of the AR model, generally having a random distribution. On the other hand, when the gears become unhealthy, the AR model cannot predict the vibration signals therefore the residual signal deviates from being randomly distributed at the location of the fault [11].

The wavelet analysis is one of the time-frequency methods for detecting faults in gears by detecting sudden changes in non-stationary signals. The advantage of the continuous wavelet transform is that it has a constant relative resolution which means that it has good time resolution at high frequencies and good frequency resolution at low frequencies [11]. Another time-frequency method for detecting gear faults is the ensemble empirical mode decomposition followed by Hilbert-Huang transform. This method introduced by Ai and Li involves the decomposition of the vibration signal using ensemble empirical mode decomposition followed by calculation of the Hilbert-Huang transform (HHT) and finishing with a diagnostic conclusion according to the HHT spectrum [10], [11]. More popular methods in literature, which will also be utilized throughout this thesis, are presented in Chapter 6. Through the use of the aforementioned signal processing techniques, it is possible to obtain vital diagnostic information from the vibration signals. However, these techniques require a good deal of expertise to apply them successfully. For

this reason, simpler approaches are sought which can generate decision options on gear conditions automatically and reliably.

1.4 Literature Review

In order to prevent this several neural fuzzy classification schemes have been proposed in literature for fault detection. To better understand neural fuzzy schemes, a review of neuro-fuzzy (NF) and soft computing techniques [13,14] was necessary. Some of the NNs reviewed include adaptive networks with emphasis on feedforward networks (FFNN) and supervised-learning neural networks emphasizing on radial basis function networks (RBFN).

Some optimization techniques implemented for training of the linear and non-linear parameters are least-squares estimate (LSE), and Lavenburg-Marquardt, and gradient techniques. While these methods have shown promising, they do tend to converge to poor local optima. For this reason, other training techniques, not introduced in [13], namely Kalman filter methods have been investigated. It was demonstrated that the EKF algorithm exhibits faster training, both in terms of the number of presentations of training data and in total training time on a sequential computer, than a standard implementation of backpropagation for problems in pattern classification and function approximation [15]. The node decoupled extended Kalman filter algorithm for NNs was studied [16], [17]. Studies have shown that the NDEKF algorithm significantly reduces the complexity and memory requirements of the extended Kalman filter (EKF). Also, it was shown in studies that the covariance matrices have a significant impact on the Kalman filtering performance [18]. Hence, techniques to improve the performance of the Kalman filter by updating the

covariance matrices have been investigated [19-22] in order to develop a better training technique.

The abovementioned methods were implemented in training of the premise and consequent parameters. Adaptive neuro-fuzzy inference system (ANFIS) was also introduced by Jang as a universal approximator for NF modeling [13]. Since ANFIS does not have the ability to identify the structure, several data clustering techniques were studied, such as K-means clustering [23], which is the simplest unsupervised learning clustering algorithm and the fuzzy C-means clustering developed by Dunn [24] and improved by Bezdek [25].

Due to ANFIS's limitation, a neuro-fuzzy system, which can adapt not only the parameters but also the structure itself, has been investigated. Dr. Angelov' work, the pioneer of evolving fuzzy systems, was given special attention [26-30]. In his work, he introduces the concept of rule base evolution over time such that the dependence on computationally expensive techniques is minimized. Conversely, there exist other interesting evolving clustering methods. The method proposed in [31] performs a one-pass, maximum distance-based clustering without any optimization. Unlike Angelov's clustering method, which does not allow for center update, the method in [31] updates both the centers and the radiuses of the cluster. Other methods, developed by the same author as in [31], propose transductive NF inference systems with weighted data normalization (TWNFI) [32] and NF inference method (NFI) [33] for transductive reasoning systems. The latter is a continuation Kasabov's dynamic evolving NF inference system (DENFIS) [34]. DENFIS evolves through incremental supervised or unsupervised learning accommodating new data inputs and new features and classes. All these methods are based

on the Takgi-Sugeno (TS) Type I fuzzy model and can be implemented online as well as offline. On the other hand, these methods have not been applied on systems monitoring gear trains. Research related to gear trains has been done by Dr. Wang and his research group, in which they developed several types of NF classifiers for machinery condition diagnostics with applications to gears [35-40]. In their work, several condition indicators have also been developed. However, the previous work was mainly for forecasting applications. On the other hand, this work focuses on developing an intelligent classifier for gearbox diagnosis.

1.5 Objectives and Contributions

The objective of this research is to develop an advanced evolving NF scheme, called ie^{2}TS-sDEKF, for diagnostic classification of gear systems.

The following contributions have been made during the course of this research work:

1. A novel ie^{2}TS-sDEKF system has been proposed to improve the classification efficiency. A decision-making component has been introduced to determine whether or not the rules should be updated as well as the method in which they are updated. By introducing the design parameter, the proposed clustering technique, ie^{2}TS, became more flexible in regards to the overlap of the clusters.

2. An improvement has been made to the network training technique. Since the covariance matrices in a Kalman filter have a great impact on the performance of the filter, a scaling

method has been proposed so that the covariance matrices are not fixed but update after each epoch. This can result in an overall improvement to the stability and performance of the network.

3. Systematic tests have been taken to verify the effectiveness of the proposed evolving classifier and the related techniques. Three robust signal indices were used for classification operations. Test results have shown that the proposed classifier outperforms other related classifiers.

1.6 Outline of Thesis

In Chapter 2, a review of Intelligent Tools is presented. First, the concept of Fuzzy Logic is introduced and a description of its uses is presented. Second, NNs are also introduced and its advantages and disadvantages are presented. Third, Adaptive NF Inference Systems as well as Evolving Networks are presented with focus on evolving TS networks; these are commonly used for classification.

The aim of Chapter 3 is to introduce the proposed clustering technique. The process of clustering is explained in detail and its advantages and disadvantages are discussed.

Chapter 4 deals with the training of the fuzzy classifier. Three steps are involved in training the fuzzy classifier. In the first step the network is identified using the clustering technique presented in Chapter 3. Next, during the forward pass the consequent parameters are updated. Then, the premise parameters are updated in the backward pass. Notably, in

this chapter the proposed scaling of the covariance matrices technique will also be presented. This technique is implemented in the backward pass.

Once the network identification and the training of the network are presented, the performance of fuzzy classifier is examined. Three sets of benchmark data have been implemented in order to assess the performance of the proposed classifier. First, the Iris Benchmark data was utilized. Second, wine has been classified using the Wine Benchmark Data. Third, the Wisconsin Breast Cancer Data has been used. The performance results are presented in Chapter 5.

Chapter 6 primarily dealt with applications. The experimental setup is introduced, followed by a discussion of the classification method. Helical gears as well as spur gears with various faults have been used. With the use of the proposed fuzzy classifier, the gear trains were successfully classified into two classes: healthy or damaged. Results for the proposed and other related schemes are presented herein.

In Chapter 7 the concluding remarks and future work are discussed.

Chapter 2

2.0 Intelligent Tools

"Intelligence is the capacity to receive,
decode and transmit information efficiently."
ROBERT ANTON WILSON

Real-world problems, for which a mathematical model is sometimes impossible to obtain, depend on intelligent systems for analysis. Intelligent schemes adapt themselves to a variety of environments by learning, evolving and making decisions. The foundations of these systems are NNs, fuzzy logic (FL), and neural fuzzy synergistic schemes (e.g. ANFIS), which will be discussed briefly in this chapter.

2.1 Fuzzy Logic

Dr. Zadeh, a professor at the University of California at Berkley, firstly conceived the concept of FL [41]. Such concept allowed the processing of data by introducing partial set membership rather than crisp set membership or non-membership. FL has been conceived as a better method for sorting and handling data since it mimics human control logic. It uses an imprecise but very descriptive language to deal with input data more like a human operator.

There are three types of fuzzy reasoning models described as follows:

Type I: Mamdani fuzzy model

In a Mamdani fuzzy model the inputs are crisp values whereas the output is a fuzzy set. A defuzzification process is necessary in a Mamdani model in order to extract a crisp value from the fuzzy sat. There are five defuzzification methods such as centroid of area, bisector of area, mean of maximum, smallest of maximum and largest of maximum [13].

A typical fuzzy rule in a Mamdani model is of the following form:

IF x is A and y is B THEN z is C

where A, B and C are fuzzy sets, x and y are inputs whereas z is the output.

Type II: Sugeno fuzzy model

Also known as a TSK fuzzy model, this model was proposed by Takagi, Sugeno and Kank in order to generate fuzzy rules from a given input-output data set [13]. The fuzzy rules can be represented as follows:

IF x is A and y is B THEN $z = f(x,y)$

where A and B are fuzzy sets in the antecedent and z is a crisp value in the consequent. When the output is a first order function, the model is called a first-order Sugeno fuzzy model whereas when the function is a constant, it is called a zero-order Sugeno fuzzy model [13].

Type III: Tsukamoto fuzzy model

In this model, the consequent of each rule is represented by a fuzzy set with a monotonical membership function (MF). Hence, the output of each rule is defined as a crisp value. The rules can be represented as follows:

IF X is small then Y is C

To illustrate the performance of fuzzy logic, consider the following example:

The aim is to control the speed of a motor by changing the input voltage. Suppose a set point is defined and in the case in which the motor runs faster, we need to slow it down by reducing the input voltage. On the other hand, if the motor slows below the set point, the input voltage must be increased so that the motor speed reaches the set point.

Let the input status words be: *Too slow, just right* and *too fast*

Let the output sets be: *Less voltage, not much change* and *more voltage.*

The rule-base is defined as follows:

$\Re 1$. IF the motor is running too slow, THEN apply more voltage.

$\Re 2$. IF the motor speed is about right, THEN not much change.

$\Re 3$. IF the motor speed is to fast, THEN apply less voltage.

The corresponding membership functions for inputs and output variable are shown in Figure 2.1.

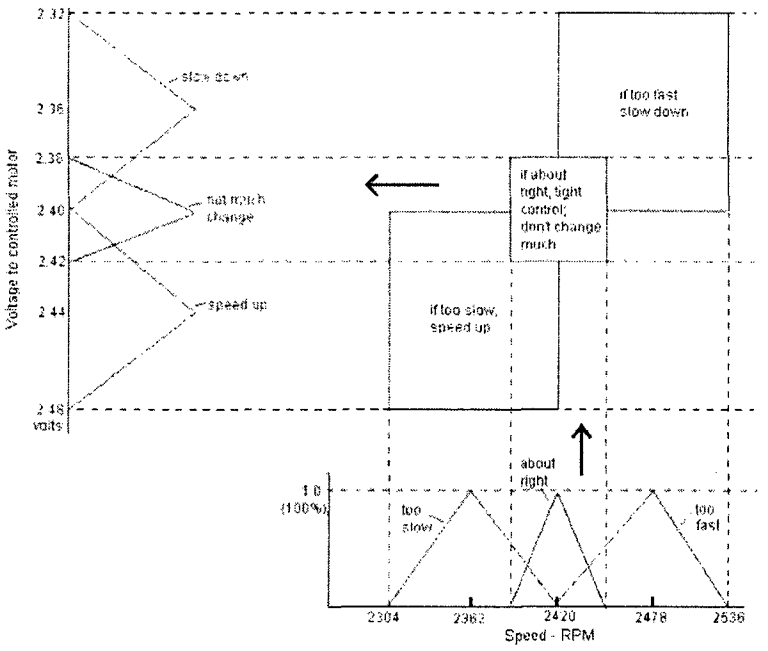

Figure 2.1 - Cause-Effect

The case when the motor's speed increases from 2420 RPM to 2437.4 RPM is depicted on the membership functions (MFs) as shown in Figure 2.2

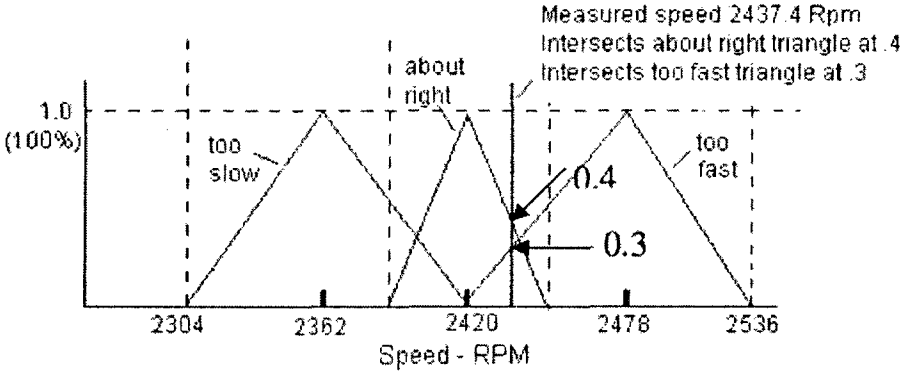

Figure 2.2 - Speed above set point

The intersection points with the second MF and the third MF are 0.4 and 0.3, respectively. This is depicted in Figure 2.2. The corresponding changes for the input voltage are represented in Figure 3.3.

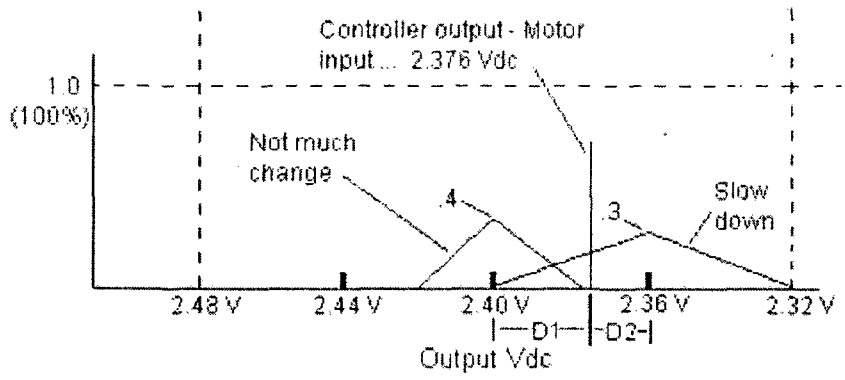

Figure 2.3 - Motor voltage

Through simple math, the area of the "Not much change" triangle and "Slow down" triangle can be determined to be 0.008 and 0.012, respectively. The output is determined by obtaining the point at which two triangles are balanced.

Thus,

$$0.008D_1 = 0.012D_2 \tag{2.1}$$

$$D_1 + D_2 = 0.04 \tag{2.2}$$

From Eq. (2.1) and Eq. (2.2) we obtain $D_1 = 0.024$ and $D_2 = 0.016$. Thus the voltage required would be 2.40-0.024=2.376 V

From the above example, it is obvious that fuzzy logic has advantages and disadvantages. The advantages are that it allows the use of linguistic terms in the rules and it reasons similar to the human brain. However, it is difficult to estimate the MFs requiring experience or trial and error.

2.2 Neural Networks

A NN is a parallel system, capable of resolving paradigms that linear computing cannot. It is composed of artificial neurons or nodes connected through directional links. Typically, adaptive networks are classified into two classes: feedforward NNs and recurrent NNs; however, throughout this work feedforward NNs have been employed. Figure 2.4 presents an example of such a network.

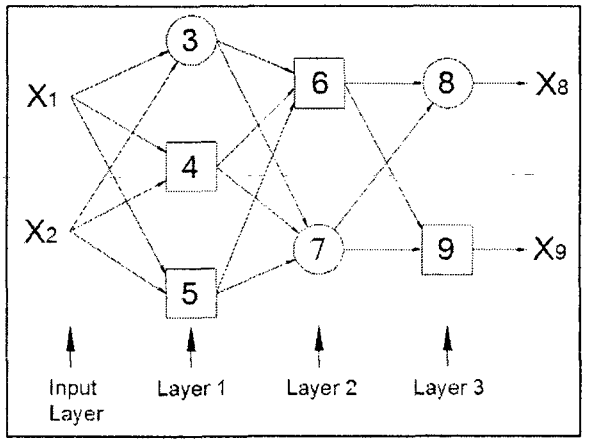

Figure 2.4 - A feedforward network structure

Feedforward networks propagate from the input to the output through the directional links and nodes. Each node has a node function whereas the links connecting them represent the causal relationship between the connected nodes [13]. Some of the nodes in a neural network are adaptive, meaning that the outputs of these nodes are subject to change as the parameters pertaining to these nodes update. There are many learning paradigms proposed in literature, some derivative based such as steepest descent and others derivative-free such as sequential simplex. For this work, the recursive least square estimator is employed in order to estimate the consequent parameters, and the premise parameters are updated using the node decoupled extended Kalman filter (DEKF). This

update is performed in two passes: forward pass and backward pass with the use of a data set (i.e. training data set). This training data set consisted of desired input/output data pairs of the system to be modeled. At the end of the forward pass, the network calculates the error between the actual output and desired output such that

$$E_p = \sum_{k=1}^{N(L)} \left(y_k - \hat{y}_{L,k} \right)^2 \tag{2.3}$$

where y_k k^{th} component of the p^{th} desired output vector and $\hat{y}_{L,k}$ is the k^{th} component of the actual output vector produced by presenting the p^{th} input vector. The purpose of the training is to minimize the error. Obviously, the error is minimized when the actual and target outputs are identical.

Based on this error, the network proceeds to perform the backward pass. During the backward pass the nonlinear parameters are updated.

There are two learning paradigms found in literature to suit the need of different applications. One paradigm refers to the off-line training in which the update of the parameters is performed once the whole training data set has been presented. That is, at the end of the epoch. The second paradigm refers to an on-line training method where the parameters are updated immediately after the input/output data pair is presented. When modeling is performed based on given input/output data sets, the networks are referred to as supervised learning neural networks.

Similar to any other system, NN have advantages and disadvantages. The advantages are that a NN can perform tasks that a linear program cannot, it has the ability to learn and it can be implemented in any application. However, a NN is a "black box", meaning the user

can take no approach to establish what happens between the input into the NN and the output. Although it has the ability to learn, it needs to be trained through a process which can be time consuming depending on the size of the NN.

Since both FL and NN have their advantages and disadvantages, alternative is to use their synergistic paradigm, that is, neural fuzzy (NF) method.

2.3 Adaptive Neuro-Fuzzy Systems

An NF scheme (e.g., ANFIS) is an intelligent system that combines the human-like reasoning style of fuzzy systems with the learning and connectionist structure of neural networks. In literature, such schemes are referred to as Neuro-Fuzzy Systems (NFS) or Fuzzy Neural Networks (FNN) [14]. One example of such a NFS would be Adaptive Neuro-Fuzzy Inference Systems (ANFIS), which will be employed throughout this work. ANFIS is a class of adaptive networks that are functionally equivalent to fuzzy inference systems. The structure is presented in Figure 2.5.

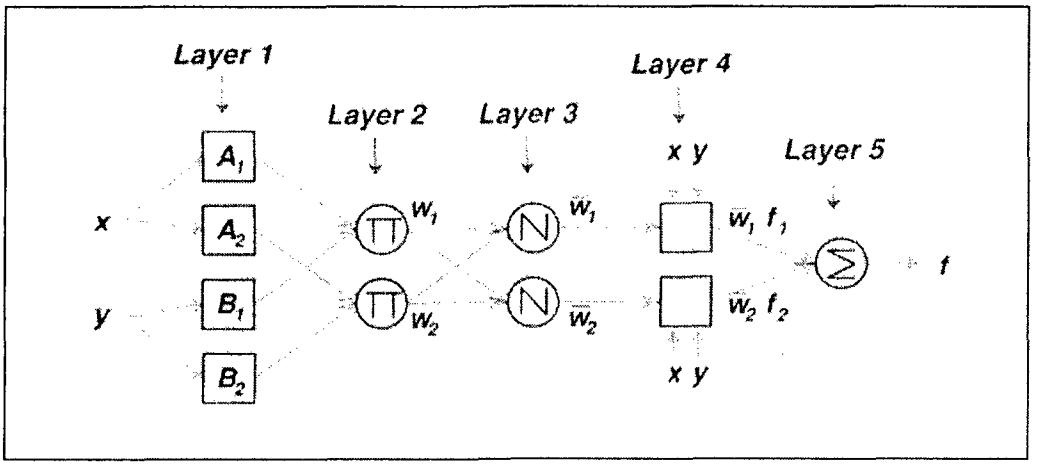

Figure 2.5 - ANFIS structure [4]

There are five layers numbered from Layer 1 to Layer 5, and each layer has nodes with node function described below.

Layer 1: Nodes are adaptive with a node function

$$\begin{aligned} O_{1,i} &= \mu_{Ai}(x), \quad \text{for } i = 1,2 \\ O_{1,i} &= \mu_{Bi-2}(y), \quad \text{for } i = 3,4 \end{aligned} \tag{2.4}$$

where $\mu_{Ai}(x)$ and $\mu_{Bi-2}(y)$ are MFs specifying the degree to which the given inputs satisfy the quantifier.

Layer 2: Nodes are fixed and the output is the product of all incoming signals:

$$O_{2,i} = w_i = \mu_{Ai}(x)\mu_{Bi}(y), i = 1,2 \tag{2.5}$$

Layer 3 Nodes are fixed and the output is the normalized firing strength

$$O_{3,i} = \bar{w}_i = \frac{w_i}{w_1 + w_2}, i = 1,2 \tag{2.6}$$

Layer 4 Nodes are adaptive with output

$$O_{4,i} = \bar{w}_i f_i \tag{2.7}$$

where $\begin{aligned} f_1 &= \theta_1 x_1 + \theta_2 x_2 + \theta_3 \\ f_2 &= \theta_4 x_1 + \theta_5 x_2 + \theta_6 \end{aligned}$

and $\theta_i, i = 1,2...,6$ is the linear parameter set to be updated

Layer 5 Single fixed node with output as the summation of all incoming signals

$$O_{5,1} = \sum_i \bar{w}_i f_i, \quad i = 1,2 \tag{2.8}$$

One of NF's advantages is that it uses a hybrid learning procedure for estimation of the premise and consequent parameters [13]. Different optimization techniques such as gradient descent (GD), least square estimate (LSE) could be implemented in order to

optimize the structure's linear (i.e. consequent) and nonlinear (i.e. premise) parameters. The disadvantage of ANFIS is that the structure remains fixed allowing only the parameters to update. To overcome this problem, evolving fuzzy systems were developed.

2.4 Evolving Neural/Fuzzy Systems

Evolving fuzzy systems (EFS) can be defined as self-developing, self-learning fuzzy rule-based or neuro-fuzzy systems that have both their parameters and their structure self-adapting [29]. The fuzzy systems developed by Dr. Angelov gradually develop mimicking the evolutionary process that takes place in populations of individuals. Meaning that EFS are mathematical paradigms that can approximate the human-like reasoning by representing it with dynamically evolving fuzzy rule-based structure. Similar to ANFIS, EFS structural framework is utilizing the Takagi-Sugeno (TS) [Takagi-Sugeno] fuzzy rule-based system. This system is of the following form:

$$\Re_j : \text{IF } (x_1 \text{ is } A_1^j) \text{ AND } (x_2 \text{ is } A_2^j) \text{ AND } \ldots (x_n \text{ is } A_n^j) \text{ THEN } (y_j = f_l^j) \tag{2.9}$$

where $\Re_j$ denotes the jth fuzzy rule, $j \in [1, Nr]$; N is the total number of fuzzy rules (clusters); A_i^j is the jth fuzzy set for x_i, $i \in [1, n]$; $y_j = [y_{j1}, y_{j2}, \cdots, y_{jM}]$ is an M-dimensional consequent (output) structure [42].

The evolving TS (eTS) fuzzy system has the ability to be represented as a neural network therefore, it can also be considered a neural-fuzzy system [43]. It is this structural framework that can be used to solve a range of problems offering flexibility, adaptation, robustness, and improved precision with small computational efforts. Some of the

problems that can be solved using this paradigm are related to clustering (i.e. network identification), time-series prediction or filtering and classification.

Although this method works particularly well, it also has its shortcomings in that the cluster centers and radiuses stay fixed. A method to alleviate this problem was introduced in [28] however, it solves the problem of the radiuses only. Hence, this thesis introduces a technique to improve the performance of eTS.

Chapter 3

3.0 Network Identification Techniques

"Not everything that counts can be counted,
and not everything that can be counted counts."

ALBERT EINSTEIN

Clustering of numerical data forms the basis of many modeling and pattern classification algorithms. The purpose of clustering is to find natural groupings of data in a large data set, thus revealing patterns in the data that can provide a concise representation of the data behavior. As stated in Chapter 1, several NF schemes have been proposed in literature for pattern classification applications. Most of the current NF classifiers, however, only deal with parameter identification whereas system structure is determined based on expertise and remains unchanged in operations. The alternative is to use cluster-based evolving methods [29, 42]. The proposed evolving fuzzy (EF) classifier will be discussed in this chapter.

3.1 Proposed Clustering Technique

The proposed EF classifier is a data-driven, non-iterative, one-pass technique modeled off of other potential based algorithms. Different from previous potential based models [29, 42], this technique performs a feasibility check on the incoming data sample. If the data point is located within the influence range of an existing cluster, then the new

data point is not treated as a new cluster center. Otherwise, if the data point is beyond a specified distance from the cluster center, then calculate its potential as well as its potential to be a cluster center. Whether the incoming data point is selected as a cluster center depends on its potential. The proposed cluster center as well as the radius is chosen based on standard deviation. Before a decision is made, the potential of the proposed cluster center is being calculated. A decision is made whether the proposed cluster center becomes a cluster center or the incoming data point is a better candidate. This procedure is presented in Figure 3.1. Subsequently, a step-by-step explanation of the process is also given.

The performance to the proposed clustering technique, named ie^{2}TS, will be tested using three sets of benchmark data. The number of identification errors (i.e. misclassified data during the identification process) generated during the identification process determines the performance of the proposed clustering technique. The results pertaining to this technique are presented in Chapter 5.

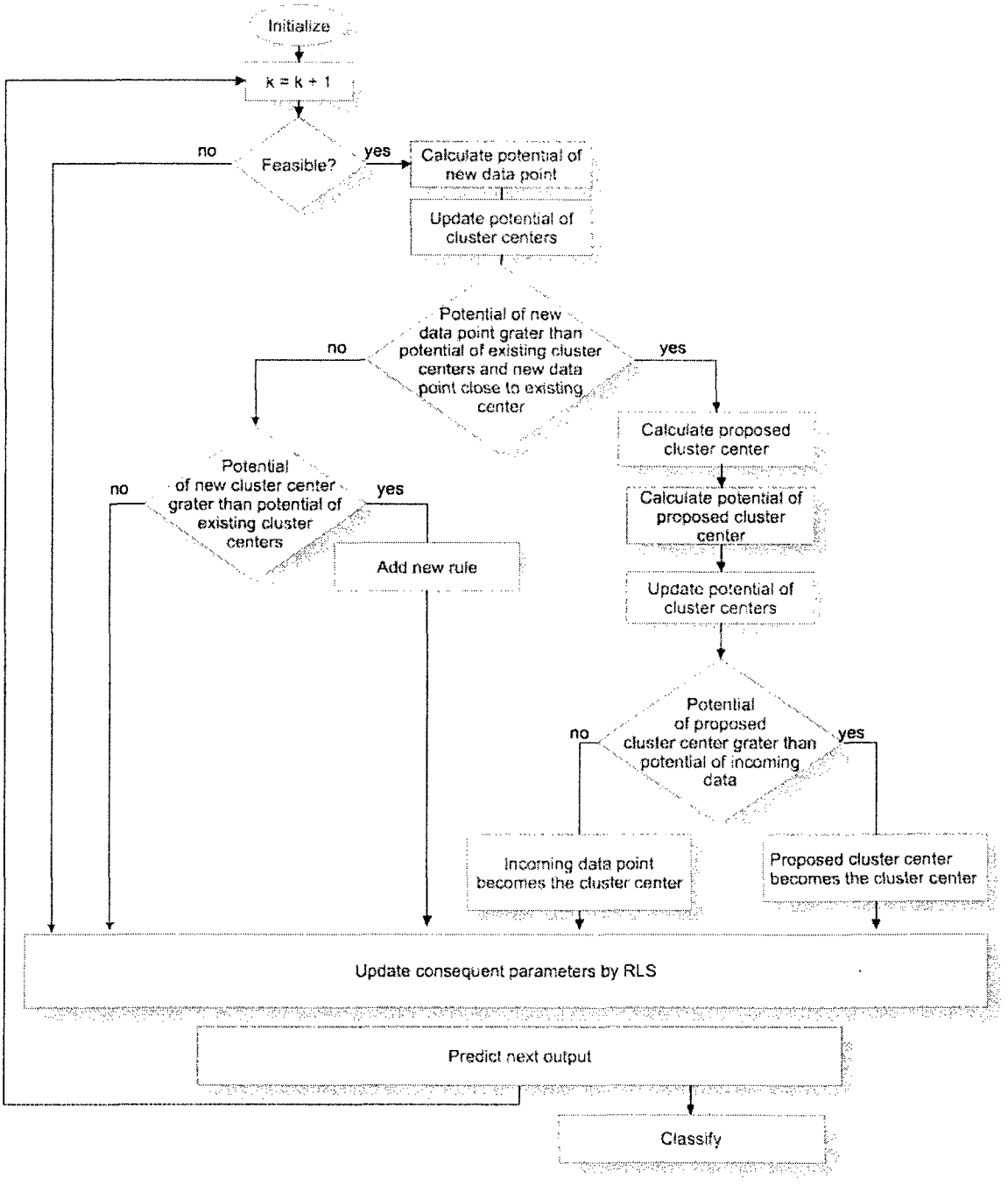

Figure 3.1 - Network identification flowchart

Step 1: The structure is initialized with one cluster, $Nr := 1$. The first data sample is the center of the cluster and the spread of the cluster is predefined such that:

$$\begin{aligned} k &:= 1 \qquad & \mathrm{P}\left(z_1^I\right) &:= 1 \\ m_1^I &:= x_1 \qquad & \sigma_1^I &:= spread \\ m_1^O &:= y_1 \qquad & \sigma_1^O &:= spread \end{aligned} \tag{3.1}$$

where $m_k^I, m_k^O, \sigma_k^I, \sigma_k^O$ are the cluster centers and user defined spreads in the input and output space, respectively, at the k^{th} time step and $P\left(z_1^I\right)$ is the potential of the first data point, $z_1^I = [x_1 \ y_1]$.

Step 2: Next data sample is being presented, $z_k^I = [x_k \ y_k]$, where $k = k + 1$.

Step 3: The feasibility of the point to become a cluster center is calculated. The Euclidian distance from the incoming point to all existing cluster centers is determined and compared to the radiuses of the clusters, as described in Eq. 3.2.

$$s_k = \left\| z_k^I - m_i^I \right\| \tag{3.2}$$

where $i = 1,2,...,Nr$ and Nr is the number of clusters generated.

If $s_k < \varsigma$ where ς is a user defined distance, then go to Step 7.

Step 4: The potential [29] of the new data sample is calculated by:

$$P_k(z_k) = \frac{k-1}{(k-1)(\vartheta_k + 1) + \gamma_k - 2\upsilon_k} \tag{3.2}$$

where $\vartheta_k := \sum_{j=1}^{n+1} \left(z_k^j\right)^2$; $\gamma_k := \gamma_{k-1} + \sum_{j=1}^{n+1} \left(z_{k-1}^j\right)^2$; $\upsilon_k := \sum_{j=1}^{n+1} z_k^j \beta_k^j$ $\beta_k^j := \beta_{k-1}^j + z_{k-1}^j$.

Step 5: The potential of the existing clusters are being updated by:

$$P_k\left(z_l^*\right) := \frac{(k-1)P_{k-1}\left(z_l^*\right)}{k-2+P_{k-1}\left(z_l^*\right)\sum_{j=1}^{n+1}\left(d_{k(k-1)}^j\right)^2} \tag{3.3}$$

where $d_{k(k-1)}^j := z_k^j - z_{k-1}^j$ and $l = [1, Nr]$, Nr is the number of clusters.

Step 6: The potential of the new data sample is compared to the potential of all existing cluster centers and a decision is made as follows:

a. IF potential of new data point is higher than the potential of all existing cluster centers

AND new data point is close to an existing center

THEN the new cluster center is [42]

$$\begin{aligned} m_n^I &= m_j^I + \frac{\sum\left(x_k - m_j^I\right)^2}{N} \\ m_n^O &= m_j^I + \frac{\sum\left(y_k - m_j^O\right)^2}{N} \end{aligned} \tag{3.4}$$

where n denotes a new temporary point , j is an index such that $z_j^* = \arg\min_{i=1}^{Nr}\left(\left\|z_k - m_i\right\|\right)$,

$m_n = \left[m_n^I, m_n^O\right]$ and N is the number of samples in the cluster .

Calculate the potential of m_n by Eq. 3.2.

Update the potential of existing clusters by Eq. 3.3.

IF potential of $P_k(m_n)$ is greater than potential of existing cluster centers

THEN the newly assigned point replaces the cluster center, cluster radius and potential are updated by

$$z_j^* = m_n;\ \sigma_j^* = \frac{\sum(\sigma_i^* - m_n)^2}{Nr};\ P_k(z_j^*) = P_k(z_n)$$

ELSE the data point replaces the cluster center and potential is updated

$$z_j^* = z_k;\ P_k(z_j^*) = P_k(z_k)$$

END IF

b. ELSE IF potential of new data point is higher than the potential of all existing cluster centers

THEN a new cluster is formed:

$$Nr = Nr+1;\ z_{Nr}^* = z_k;\ P_k(z_{Nr}^*) = P_k(z_k)$$

END IF

Step 7: Consequent parameters are updated by

Step 8: Output at the next time step is predicted by

$$\hat{y}_{k+1} = \psi_k^T \hat{\theta}_k,\ \ k = 2,3,...$$

Step 9: Output is classified

By introducing Step 3 in the identification procedure, the user is able to define the amount of cluster overlapping. In other words, it ensures that a new cluster is not being created if the incoming point is within a cluster. If the incoming data set is not feasible, calculating its potential would be unnecessary. This computation redundancy is being eliminated as a result of introducing the feasibility check.

Different than the method in [16], in this method a new decision is made before accepting the proposed data set as a cluster center. This step is ensuring that the proposed data set is indeed a better candidate to become a cluster center before accepting it. If the proposed data set is not a better candidate to become a cluster center, then the proposed data set is being disregarded and the incoming data set become the cluster center.

Each cluster center found is in essence a prototypical data point that exemplifies a characteristic input/output behavior of the system. Also, each cluster center represents a fuzzy rule that describes the system performance [49]. Hence, we can translate each cluster into a fuzzy rule describing the network. For example, suppose after clustering was applied to the group of data, cluster center m_{ij}^I with a radius σ_{ij}^I was found. This cluster center and radius provide the rule:

$\Re_j$. IF input is near m_{ij}^I then $y_j = f_l^j$

This can also be represented in the TS-1 form as per Eq. 2.9:

$\Re_j$. IF (x_1 is A_1^j) AND (x_2 is A_2^j) AND ... (x_n is A_n^j) THEN ($y_j = f_l^j$)

where

$$A_i^j = e^{-\frac{1}{2}\left(\frac{x_i - m_{ij}^I}{\sigma_{ij}^I}\right)^2} \tag{3.5}$$

Remark A: Cluster compatibility is achieved when $W_t = W_o$ where $W_I = \arg\min_{k=1}^{K} \left\| \boldsymbol{m}_k^I - \boldsymbol{x}_t \right\|$ and $W_O = \arg\min_{k=1}^{K} \left\| \boldsymbol{m}_k^O - \boldsymbol{y}_t \right\|$. This ensures that noise affected data is excluded and more meaningful clusters are created.

Chapter 4

4.0 Training of the Evolving Fuzzy Classifier

"AI is the science of making machines do tasks that humans can do or try to do."

JAMES F. ALLEN

Once the structure of the EF classifier is created based on the proposed clustering technique as discussed in Chapter 3, its parameters should be trained properly to provide optimal classification operations. In this chapter, the training technique implemented throughout this work is presented. Section 4.1 presents the training of the consequent parameters whereas Section 4.2 presents the training of the premise parameters. In Section 4.2.2 an improvement to the classical Kalman filter training technique is presented.

4.1 Offline Training based on R-LSE

Four steps are involved in the fuzzy classifier. First, the structure is identified using the procedure. Second, the linear consequent parameters are estimated in the forward pass using R-LSE algorithm [44, 45]. Third, the output is classified into different categories. Fourth, the nonlinear premise parameters are optimized in the backward pass using a new method, scaled Node Decoupled Kalman Filter (sDEKF) presented in the next section. The learning paradigm employed throughout this work is supervised learning meaning that the input/output data pairs are given and the scope is to find a function that best matches the

input/output pair. In other words, the goal is to infer how the mapping implied by the data and the cost function is related to the mismatch between the mapping and the data.

In this work, the consequent parameters (m_j^l and σ_j^l) of the TS-1 fuzzy model are estimated using R-LSE. The objective function at time instant t is given by

$$E(\theta_t) = \frac{1}{2}\sum_{t=1}^{P}\left(y_t - \psi_t^T\theta_t\right)^2 \tag{4.1}$$

where ψ_t is vector of inputs weighted by the firing levels of the rules. The update of the consequent parameters is performed by:

$$\theta_{t+1} = \theta_t + \chi_{t+1}\psi_t\left(y_t - \psi_t^T\theta_t\right) \tag{4.2}$$

$$\chi_{t+1} = \alpha_t\left(\chi_t - \frac{\chi_t\psi_t\psi_t^T\chi_t}{\alpha_t + \psi_t^T\chi_t\psi_t}\right) \tag{4.3}$$

where $t = 0,1,...,(P-1)$, P is the number of observation, α_t is a forgetting factor and χ_t is the covariance matrix. At the initial time, the covariance matrix is initiated as $\chi_0 = \rho I$ where ρ is a constant. Since the performance of the algorithm is dependent on the initial values, many trials were performed in order to find the initial covariance matrix as well as the initial vector of parameters that give the best performance. Therefore, in this work ρ was selected as 10^3 whereas the vector of parameters has been initiated to $\theta_0 = 0$.

4.2 Training Based on DEKF

It is known that the process noise covariance matrix (Q) and observation error covariance matrix (R) have a considerable impact on the performance of Kalman filter since they are dependent on the application environment and process dynamics [18]. Q and R are responsible for the weight that the system applies between measurements. Consequently, the filter may diverge or never achieve optimal results given Q and R exhibit errors.

Generally, the covariance matrices are fixed to a value determined from intensive empirical analysis. However, for a complex dynamic system in a noisy environment, determination of the covariance matrices in advance is sometimes unachievable. Accordingly, a covariance matrix updating technique is proposed in this thesis. The motivation behind this comes from the high possibility for improved performance of DEKF. Before explaining the proposed sDEKF technique, a brief introduction to Kalman filter is given.

4.2.1 Conventional Kalman Filter

Updating and predicting are the two distinct phases of Kalman filter. During the prediction phase, the posteriori state estimate from the previous time step is used to produce an estimate of the state at the current time step. The In the update phase, the priori prediction is combined with the current information to refine the state estimate as well as the posteriori error covariance matrix.

Consider a multivariable system of the form:

$$x_k = F_k x_{k-1} + w_{k-1} \tag{4.4}$$

$$z_k = H_k x_k + v_k \tag{4.5}$$

where x_k is an $(\mathrm{n}\times 1)$ state vector, F_k is an $(n\times n)$ transition matrix, z_k is an $(r\times 1)$ observation vector and H_k is an $(r\times n)$ observation matrix. w_k is the process noise and v_k is the observation noise. Both are considered white Gaussian noise with zero means and covariances given by:

$$\begin{aligned} &E(w_k) = E(v_k) = 0 \\ &E\left(w_k w_i^T\right) = \begin{cases} Q_k, & i = k \\ 0, & i \neq k \end{cases} \\ &E\left(v_k v_i^T\right) = \begin{cases} R_k, & i = k \\ 0, & i \neq k \end{cases} \end{aligned} \tag{4.6}$$

where $E(\bullet)$ denotes the expectation, Q_k and R_k are the process noise and observation noise covariance matrices, respectively.

1) Prediction Phase

Predicted state:

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} \tag{4.7}$$

Predicted estimate covariance matrix (a measure of the estimated accuracy of the state estimate):

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k \tag{4.8}$$

where $P_{k|k-1} = \mathrm{cov}\left(x_k - \hat{x}_{k|k}\right)$

2) Updating Phase

Measurement residual:

$$\tilde{y}_k = z_k - H_k \hat{x}_{k|k-1} \tag{4.9}$$

Optimal Kalman gain:

$$K_k = P_{k|k-1} H_k^T \left(H_k P_{k|k-1} H_k^T + R_k \right)^{-1} \tag{4.10}$$

State estimate update:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k \tag{4.11}$$

Estimate covariance matrix update:

$$P_{k|k} = P_{k|k-1} + K_k H_k P_{k|k-1} \tag{4.12}$$

where $x_{k|k-1}$ represents the estimate of x at time instant k given observations up to and including $k-1$.

4.2.2 Proposed Updating of Process Noise and Observation Error Covariance Matrices

Covariance provides a measure of the strength of the correlation between two or more sets of random variables. The process noise covariance matrix and the observation noise covariance matrix are responsible for the performance of the Kalman filter [18]. In general, when a mathematical model of the system can be obtained, the covariance matrices are chosen based on experience or through experiments. However, this can be a daunting

process. It is very difficult (if not impossible) to derive accurate mathematical models for many complex systems. This leaves the system to be approximated by a Kalman filter. As a consequence, the covariance matrices must also be approximated. This process leaves room for a significant amount of error making the training technique somewhat unreliable. Therefore, a new method to update process noise and observation error covariance matrices is proposed in this section to improve robustness of the training technique. From Eq. 4.10 the innovation covariance matrix is defined as:

$$S_k = \left(H_k P_{k|k-1} H_k^T + R_k\right)^{-1} \tag{4.13}$$

and from Eq. 4.8, $P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k$.

Intuitively, the updates of the process noise and observation error are performed with the help of the innovation covariance matrix such that:

$$R_{k|k} = R_{k|k-1} + R_{k|k-1}\left(S_k\right)^{\beta} \tag{4.14}$$

$$Q_{k|k} = Q_{k|k-1} - Q_{k|k-1}\left(S_k\right)^{\beta} \tag{4.15}$$

where β is a design parameter between 0 and 1.

The process noise covariance matrix and observation error covariance matrix are diagonal matrices initialized at 0.01 and 0.85, respectively. Different tests were performed with initialization values ranging between 0.0001 and 1. After each epoch, the update of the noise covariance matrix and observation error covariance matrix is performed utilizing Eq. 4.14 and Eq. 4.15, respectively. The motivation for adopting this direct modification of

the algorithm lies in the fact that the performance of the Kalman filter is dependently related on the covariance matrices. Since there exist no direct method to determine the parametric values of these error matrices, the algorithm is modified to account for this task.

During the training with sDEKF all covariance matrices update to achieve more robust Kalman filter. It has been noted that with the introduction of the scaling factor, the predicted estimate covariance matrix, P, changes at a slower rate whereas the process noise covariance matrix and observation error covariance matrix change at a faster rate. This can only imply that the observed error and process noise covariance matrices are poorly approximated. Since all covariance matrices are being updated the state estimate update is more robust making the training technique more reliable, as it can be noted in Chapter 5.

Many trials have been conducted in order to find the optimal value of β. In this work, a value of $\beta = \frac{1}{2Nr}$ will be utilized. Details of implementation and system training based on the suggested sDEKF will be discussed in the following chapter.

Chapter 5

5.0 Performance Evaluation

"Errors using inadequate data are
much less than those using no data at all."
CHARLES BABBAGE (1791 - 1871)

The methods outlined in the previous chapters, which were utilized to identify the networks (i.e. ie^{2}TS) as well as to train the networks (i.e. sDEKF), are tested in this chapter. The results pertaining to the clustering and training technique are presented and discussed in the following sections. At this point it is important to note that the proposed network is a combination of the identification technique ie^{2}TS and training technique sDEKF, denoted as ie^{2}TS-sDEKF. As a comparison, ie^{2}TS-DEKF is the paradigm employing the proposed clustering technique without the scale modification to the Kalman filter. The network denoted e^{2}TS-DEKF is a related method proposed by the author's research group in [45], whereas eTS-DEKF is the algorithm developed by Angelov and Filev in [29]. The rest of the chapter is organized as follows: Section 5.1 illustrates the classification process for the three benchmark data sets, Section 5.2 displays the performance results. Section 5.2 illustrates the processing results using the Iris benchmark data [46], the Wisconsin breast cancer data [47] and the Wine benchmark data [48], respectively.

5.1 Classification Process

As stated previously, classification is the process by which large amounts of data are divided into different categories corresponding to different states. The input/output data pairs are presented to the network, the consequent parameters are optimized and the output is computed. Before the network back propagates, the output is classified. The classification rules are presented in the performance evaluation section.

5.2 Performance Evaluation

5.2.1 Iris Benchmark Data

Since Iris Benchmark Data [46] is classified into three classes (i.e. Setosa, Versicolor, Virginica), the following rules are employed:

$\Re 1$: **IF** Output < 0.33 **THEN** Iris *is* Setosa.

$\Re 2$: **IF** Output ≥ 0.33 **AND** Output < 0.67 **THEN** Iris *is* Versicolor.

$\Re 3$: **IF** Output ≥ 0.67 **THEN** Iris *is* Virginica.

As discussed previously, the network structure is identified prior to the training of the structure. The Iris benchmark data consisted of four inputs and one output, which is to be further classified into three classes. Seventy-five input/output data pairs are utilized to identify the network, train the network and test the performance of the network. For comparison purposes, identical design parameters are given to all four techniques, $\delta_{\min} = 0.01$ and $\sigma_1^I = 0.12$. The networks are tested under different design parameters, however, the ones indicated returned the most promising results.

The generated clusters and their corresponding membership functions (MFs) are illustrated in Figures 5.1 and 5.2 for ie^{2}TS-sDEKF. The clusters were formed such that only relevant data was included. With the help of the introduced design parameter in ie^{2}TS, redundant overlapping of the clusters was prevented.

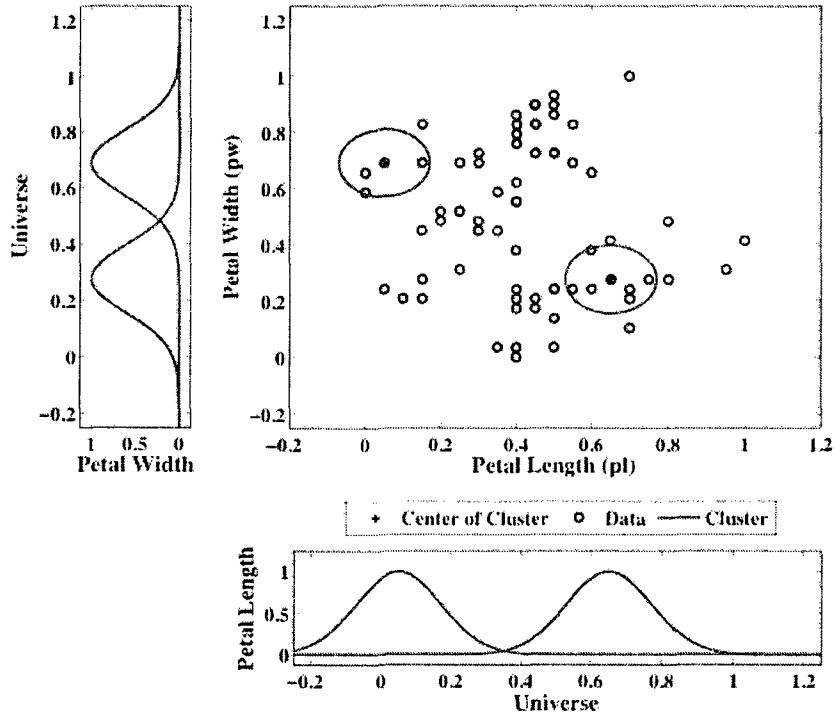

Figure 5.1 - Generated clusters and corresponding MFs for the petal using ie^{2}TS-sDEKF - Iris

Particularly, the radius of the ellipse gives the MFs spread whereas the center of MFs is the origin of the ellipse. Two clusters were generated which represent the two rules in the network. The evolution of the cluster generation is represented in Figure 5.3. It can be observed that after about 15 samples the number of rules converged to 2.

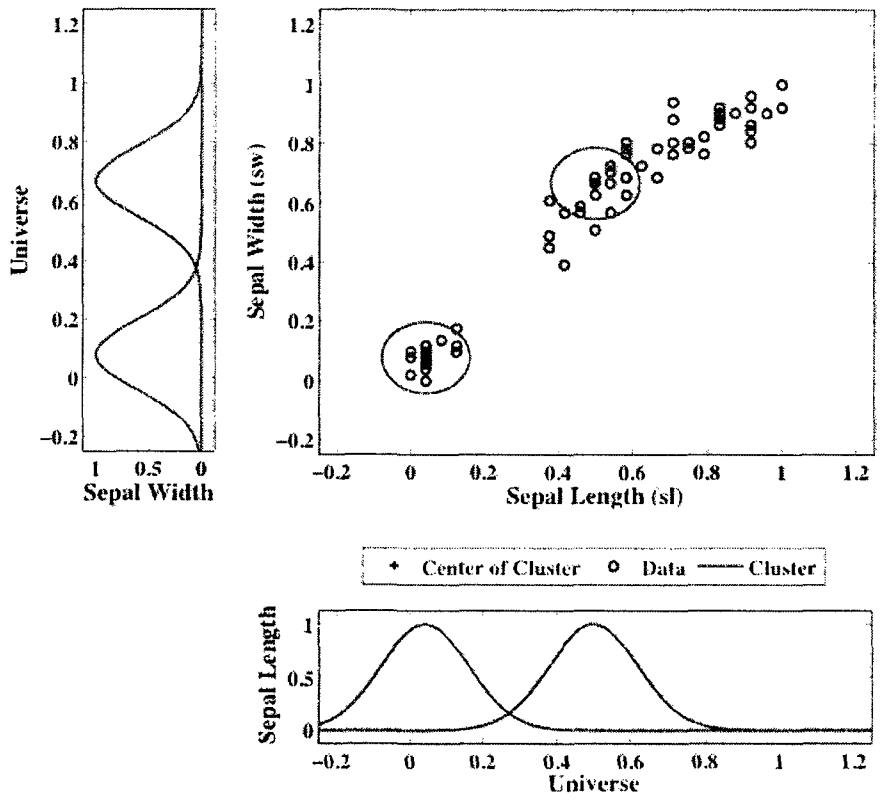

Figure 5.2 - Generated clusters and corresponding MFs for the sepal using ie^{2}TS-sDEKF - Iris

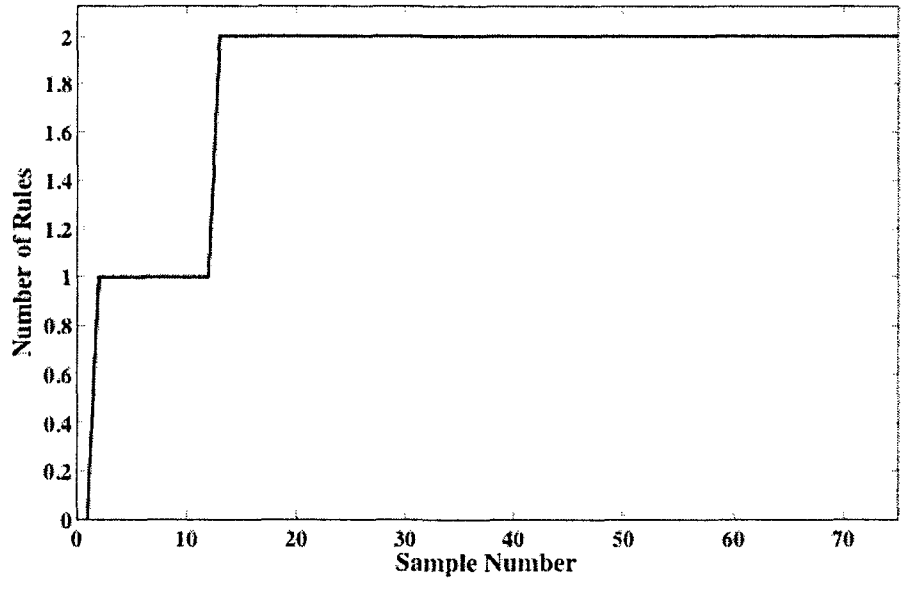

Figure 5.3 - Generated number of rules using ie^{2}TS-sDEKF - Iris

Figure 5.4 represents the identified network output and classification error. As it can be observed, there is a low discrepancy between the actual output and the target output of the network. Since the error between the actual and target output is not significant, the chosen classification scheme places the data in the proper class. In this case there existed only one misclassification.

Table 5.1 demonstrates the comparison of the related techniques. Evidently, the proposed technique provides an improvement over the other techniques. Starting with the identification process, the proposed scheme, ie^{2}TS, generated only one misclassification error whereas e^{2}TS produced 49 and eTS produced 25. The number of rules generated by the identification procedure is also an indication of an improvement. The proposed scheme, ie^{2}TS, generated half the rules generated by the other schemes. In training, the results also show improvement. The proposed technique, ie^{2}TS-sDEKF, obtained 3.993 average training errors whereas the other schemes obtained values above 5. Since the average RMSE for the proposed scheme ie^{2}TS-sDEKF, in Table 5.1 is lower than the other schemes, it can be stated that the training technique is improved. In testing, the proposed scheme, ie^{2}TS, also generated less errors thus, obtaining a recognition rate of 96% whereas the schemes used for comparison obtained a recognition rate of 93.333%. In terms of efficiency, it can also be stated that the proposed scheme is more efficient than the schemes utilized for comparison. Observing the average training time per epoch, it is obvious that the proposed scheme was trained faster. This is the result of the identification procedure generating less rules.

Iris Benchmark Data								
Method	Number of Clusters	Epochs	Avg. Training Time per Epoch (s)	Identification Errors	Average Training Errors	Testing Errors	Rate	Avg. RMSE
eTS-DEKF	4	150	0.09905	25	5.253	5	93.333	0.1033
e²TS-DEKF	4	150	0.11758	49	5.913	5	93.333	0.1034
ANFIS-DEKF	4	150	0.09997	n/a	5.940	5	93.333	0.1037
ie²TS-DEKF	2	150	0.05709	1	5.593	5	93.333	0.1011
ie²TS-sDEKF	2	150	0.05500	1	3.993	3	96.000	0.0884

Table 5.1 - Iris benchmark data results

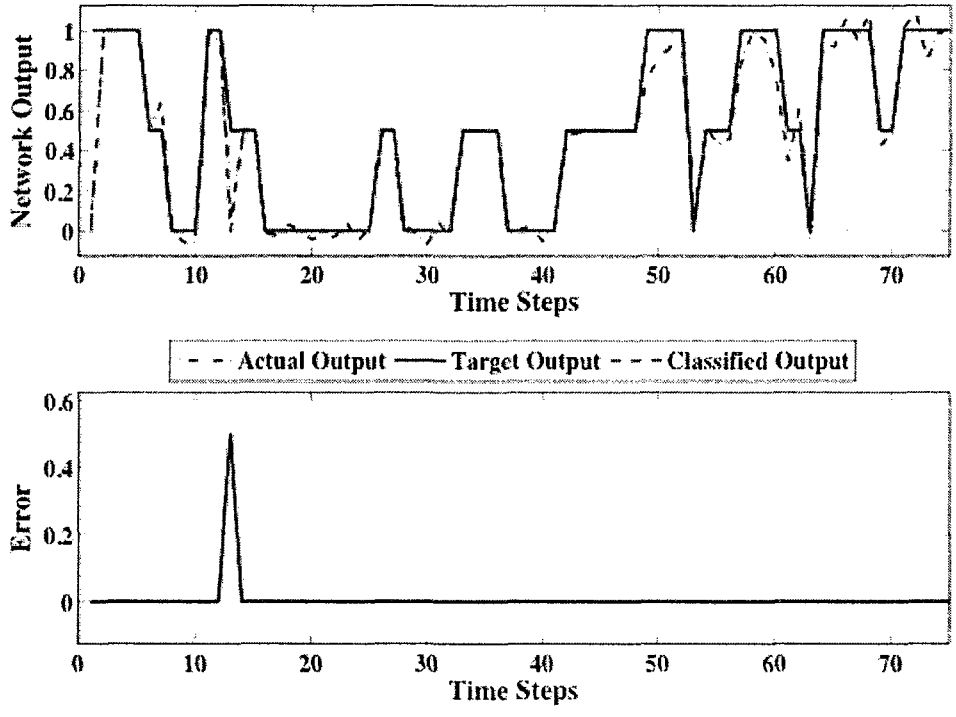

Figure 5.4 - Identified network output and classification error using ie²TS-sDEKF - Iris

Figure 5.5 and 5.6 illustrate the clusters as well as the MFs at the end of the training technique, and Figure 5.7 illustrates the network output and classification error. The robustness of the identification procedure can be realized in the error produced at the identification process (i.e. one error) as opposed to the error at the end of the training process (i.e. four errors).

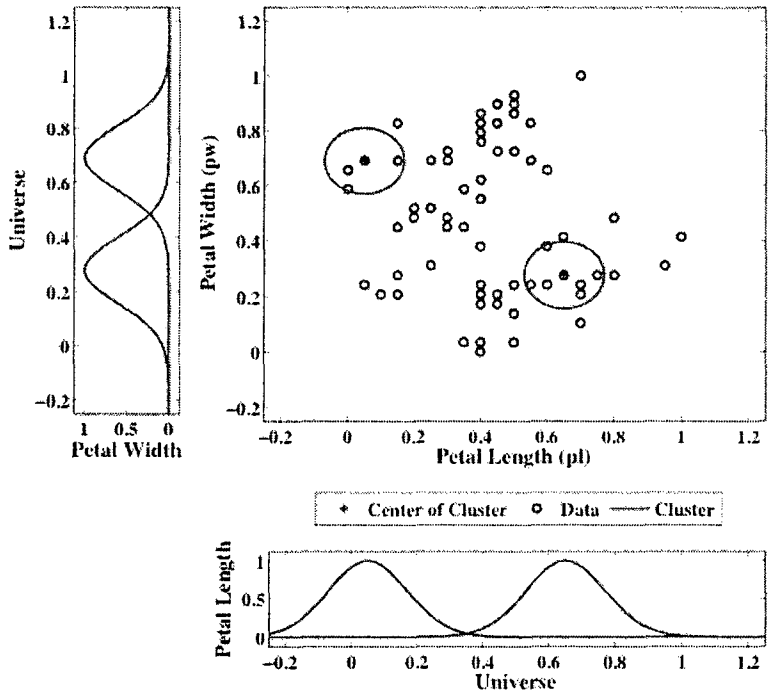

Figure 5.5 - Updated clusters and corresponding MFs for the petal using ie^{2}TS-sDEKF - Iris

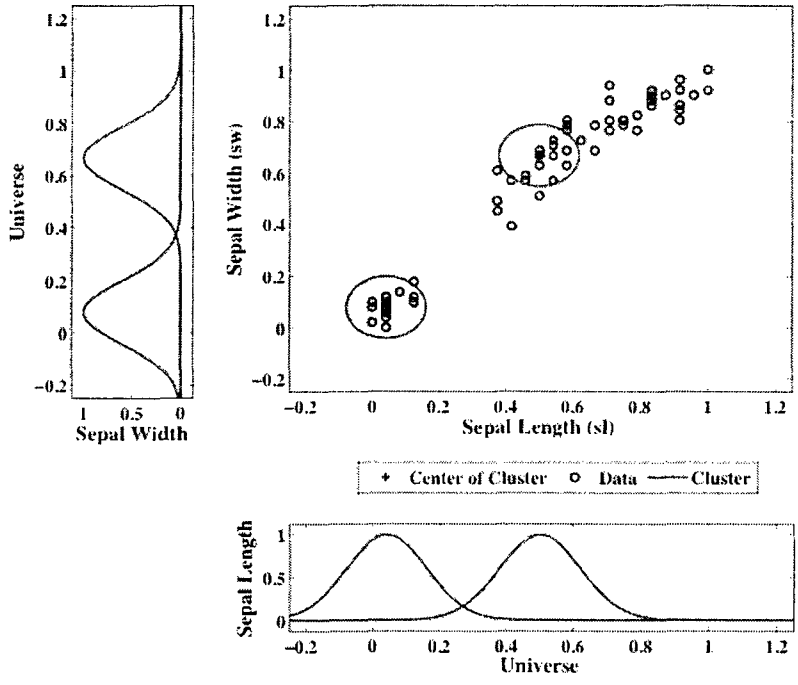

Figure 5.6 - Updated clusters and corresponding MFs for the sepal using ie^{2}TS-sDEKF - Iris

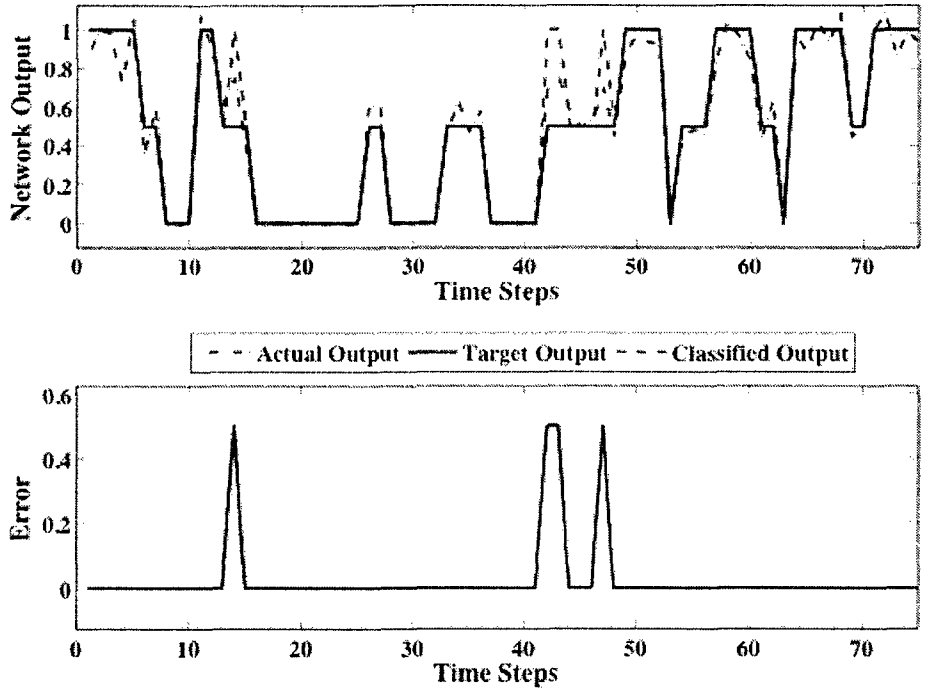

Figure 5.7 - Trained network output and classification error using ie^{2}TS-sDEKF - Iris

The network misclassified only four of the instances at the end of the training process; the performance of the proposed method, sDEKF, can also be evaluated using the root mean square error (RMSE). Figure 5.8 clearly indicates the proposed technique generates the lowest RMSE compared with other related methods. In addition, the RMSE of sDEKF quickly stabilizes when compared to the other methods.

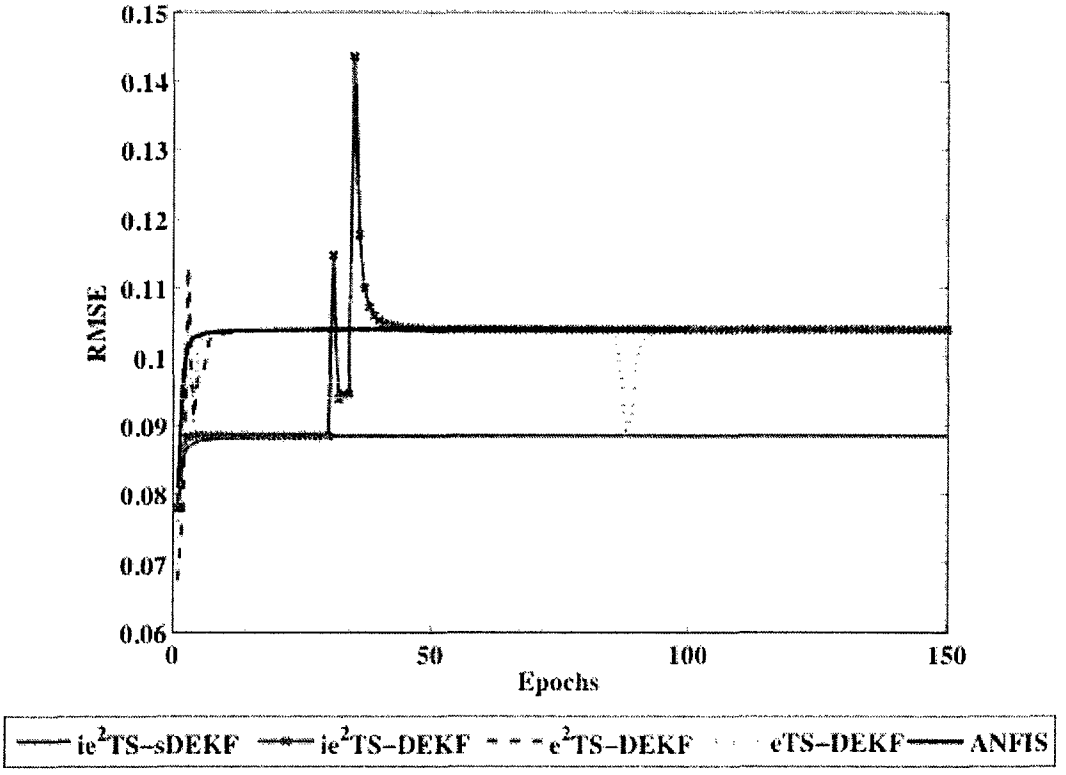

Figure 5.8 - Root mean squared error - Iris Benchmark Data

As shown in Figure 5.9, during testing, the network performed well, producing 3 classification errors and a recognition rate of 96%. All other methods produced a recognition rate of 93.33%, each having 5 testing errors in this case. These results are summarized in Table 5.1.

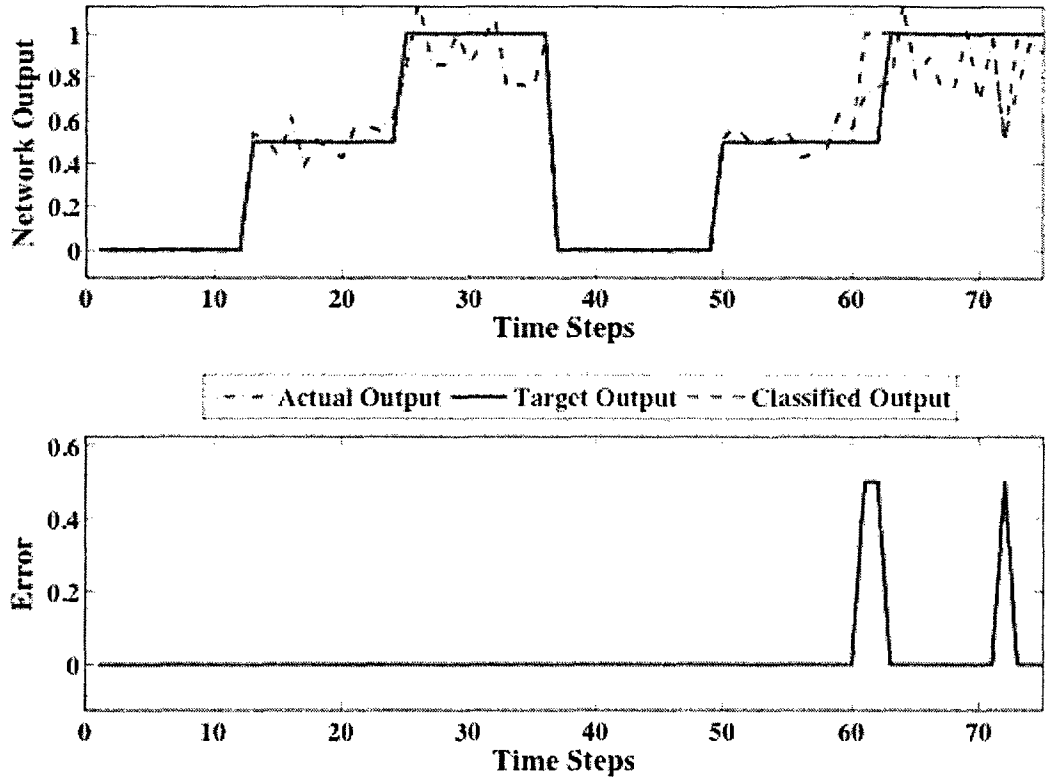

Figure 5.9 - Network verification output and classification error ie^{2}TS-sDEKF - Iris

5.2.2 Wisconsin Breast Cancer Benchmark Data

In this section, the Wisconsin breast cancer data [47] is used for analysis. The Wisconsin breast cancer data consisted of nine inputs, namely nine attributes corresponding to the shape and sizes of the cells, and one output. This output can be further classified into two classes: malignant or benign.

The rules employed are as follows:

$\Re 1$: **IF** Output < 0.5 **THEN** Breast Cancer *is* Benign.

$\Re 2$: **IF** Output ≥ 0.5 **THEN** Breast Cancer *is* Malignant.

Two hundred and fifty input/output data pairs have been utilized to identify the network structure. For comparison, all networks are initialized under the same conditions for $\delta_{min} = 0.025$ and $\sigma_1^I = 0.25$.

Wisconsin Breast Cancer Benchmark Data								
Method	**Number of Clusters**	**Epochs**	**Avg. Training Time per Epoch (s)**	**Identification Errors**	**Average Training Errors**	**Testing Errors**	**Rate**	**Avg. RMSE**
eTS-DEKF	4	150	7.27950	3	0.06	2	99	0.1862
e²TS-DEKF	2	150	0.65595	10	0.06	2	99	0.1864
ANFIS-DEKF	4	150	7.10940	n/a	0.60	2	99	0.1858
ie²TS-DEKF	2	150	0.60573	8	0.06	2	99	0.1677
ie²TS-sDEKF	2	150	0.60147	8	0.05	0	100	0.1682

Table 5.2 - Wisconsin breast cancer benchmark data results

The results are tabulated in Table 5.2. The ie^{2}TS technique identified two rules (i.e. large or small) corresponding to MFs as illustrated in Figure 5.10. The evolution of the two rules generated is illustrated in Figure 5.11. The identification procedure misclassified eight instances out of two hundred and fifty. e^{2}TS method misclassified ten instances and eTS misclassified three instances. Although the eTS only produced three errors, it generated double the number of clusters than e^{2}TS and ie^{2}TS. Compared with ie^{2}TS-sDEKF and ie^{2}TS-DEKF in Table 5.2, the training method sDEKF generated lower training errors and RMSE than the classical DEKF; it can effectively optimize the performance of the overall networks. The RMSE of sDEKF network is also smoother between epochs indicating a more stable operation when compared to the other networks.

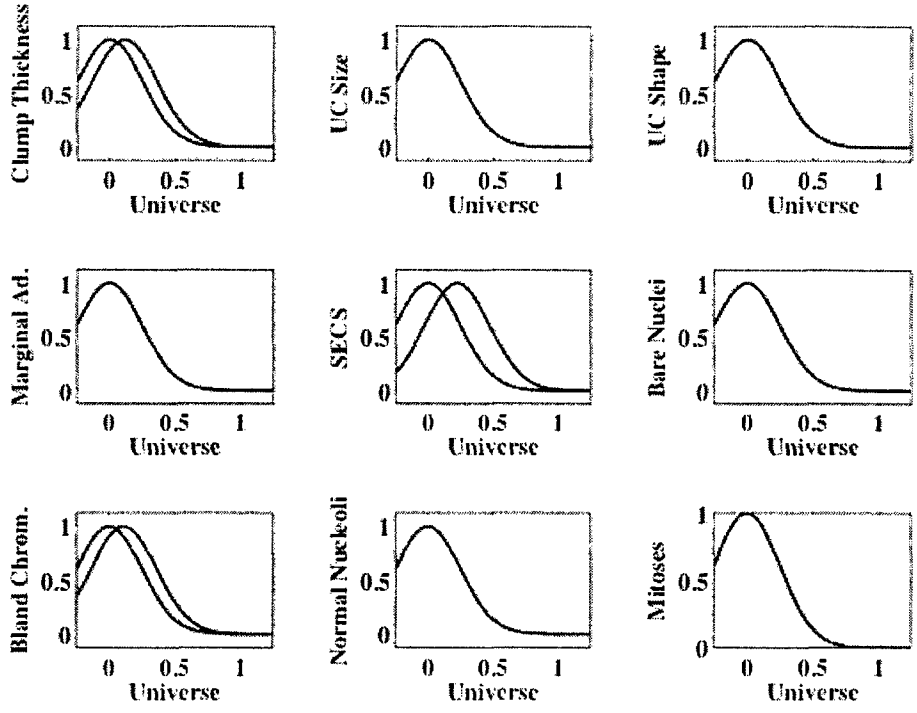

Figure 5.10 - Identified MFs using ie^{2}TS-sDEKF - Wisconsin Breast Cancer

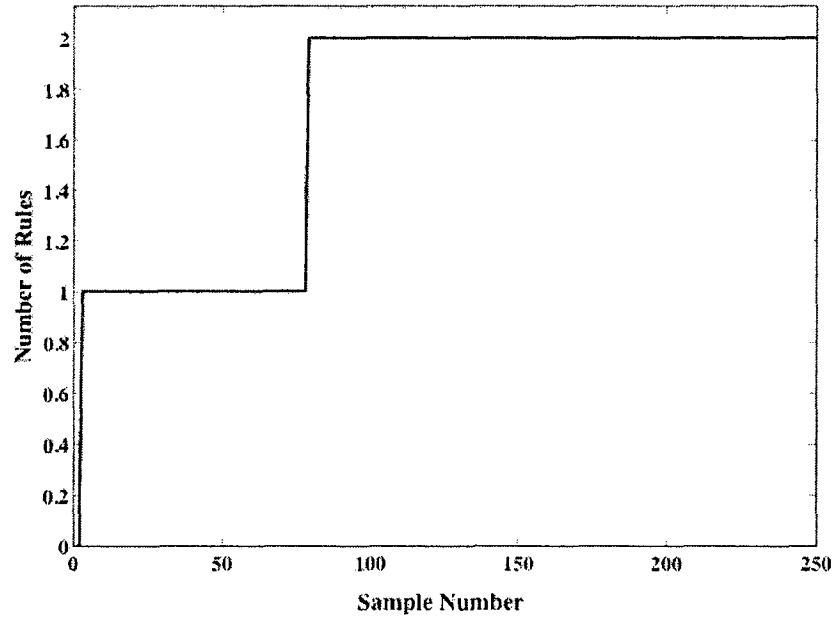

Figure 5.11 - Identified number of rules using ie^{2}TS-sDEKF - Wisconsin Breast Cancer

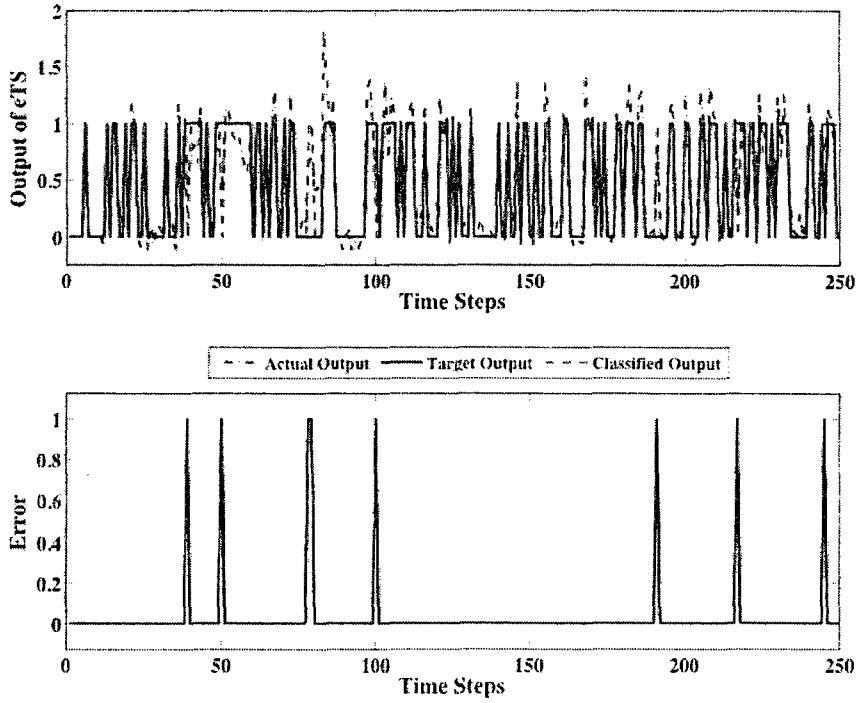

Figure 5.12 - Identified network output and classification error using ie^{2}TS-sDEKF - Wisconsin Breast Cancer

Figure 5.13 shows the updated MFs whereas Figure 5.14 illustrates the network output and classification error. Two hundred data sets were utilized for testing. It is seen that the proposed network (ie^{2}TS-sDEKF) produced zero testing errors over the two hundred input/output instances. Consequently, the recognition rate is 100%, which validates the robustness of the proposed network.

In terms of efficiency, once again the proposed scheme proves more efficient. It is faster that eTS-DEKF since it produced half the rules as noted in Table 5.2. When comparing the proposed scheme, ie^{2}TS-sDEKF, to e^{2}TS-DEKF a slight improvement in the training time can be observed. The reason for which e^{2}TS is slower is due to the introduced

constraints which add four if statements in the training algorithm, thus decreasing its efficiency.

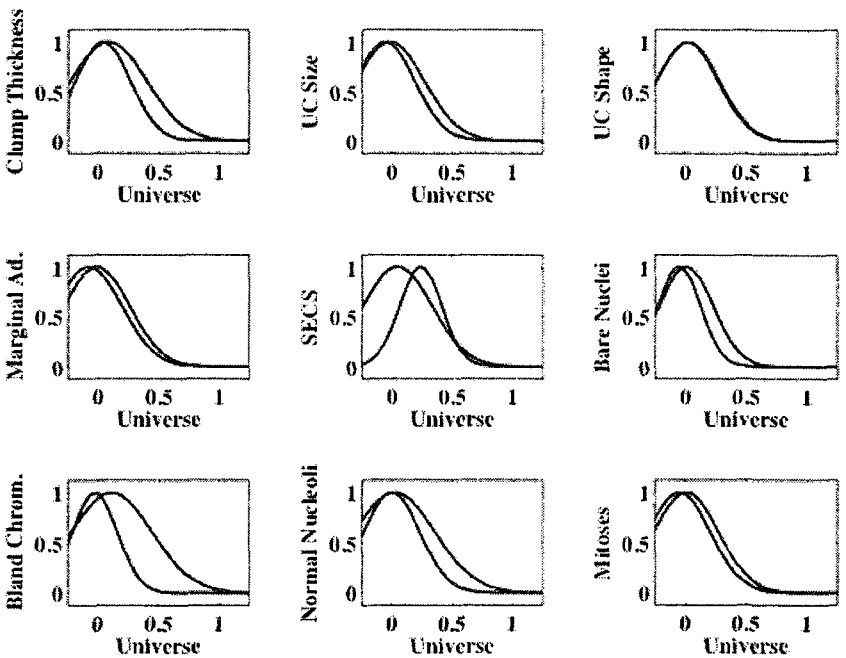

Figure 5.13 - Updated MFs using ie^{2}TS-sDEKF - Wisconsin Breast Cancer

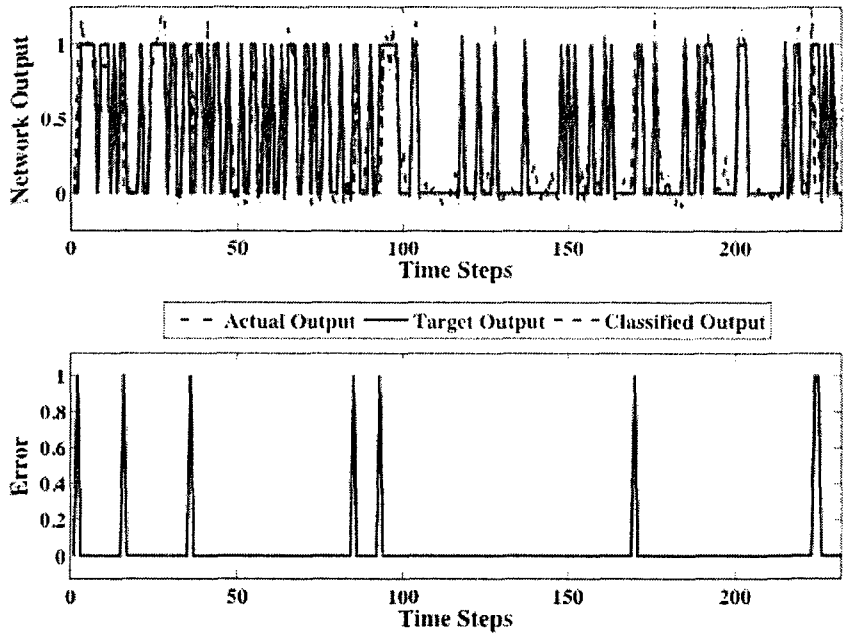

Figure 5.14 - Network training output and classification error using ie^{2}TS-sDEKF - Wisconsin Breast Cancer

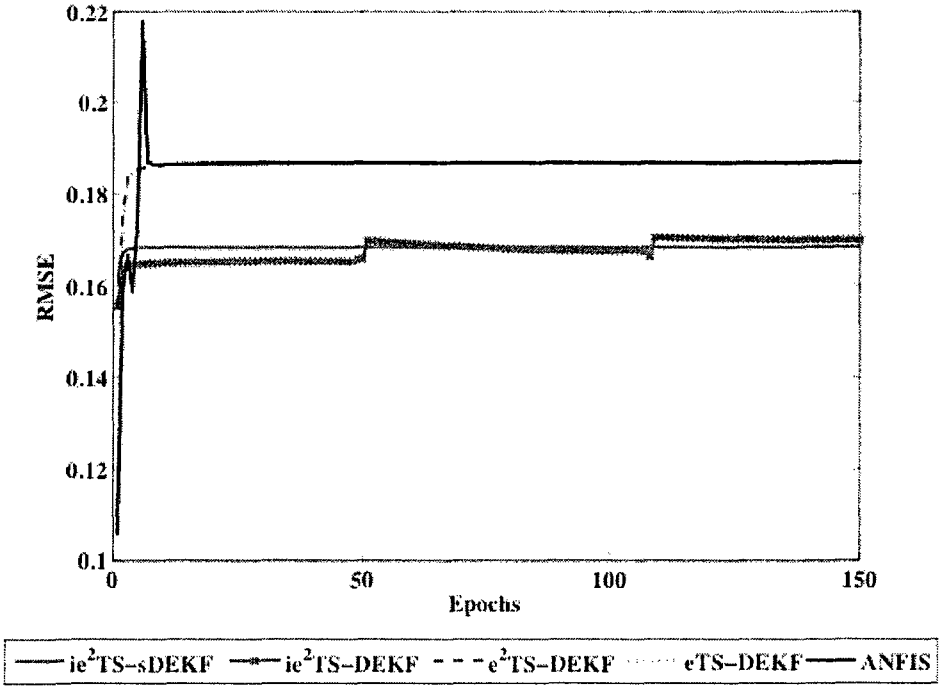

Figure 5.15 - Root mean square error of networks using ie^{2}TS-sDEKF - Wisconsin Breast Cancer

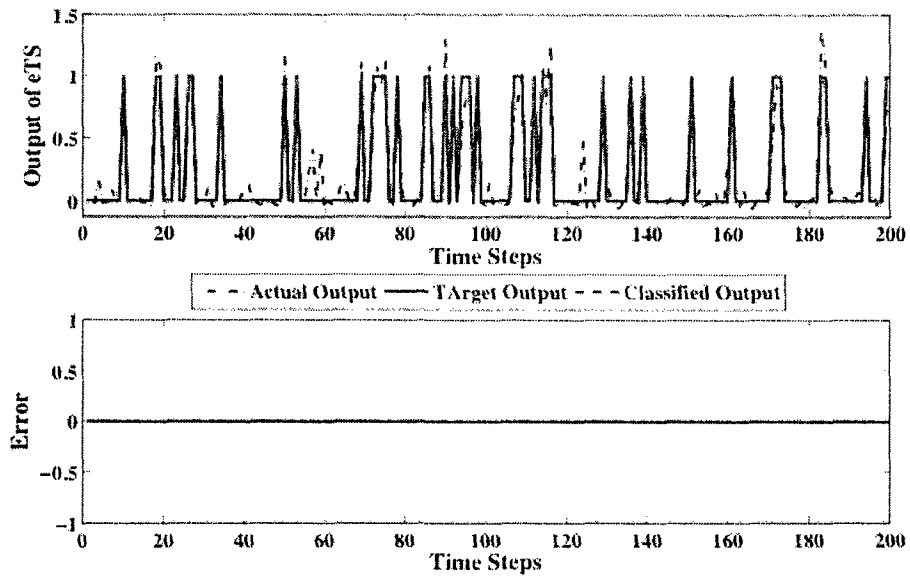

Figure 5.16 - Network verification output and classification error using ie^{2}TS-sDEKF - Wisconsin Breast Cancer

5.2.3 Wine Quality Benchmark Data

Another commonly used benchmark data set in the research of pattern classification is Wine benchmark data [48].

The Wine Benchmark Data set consists of three classes. The rules utilized are as follows:

$\Re 1$: **IF** Output < 0.33 **THEN** Wine *is* Class I.

$\Re 2$: **IF** Output ≥ 0.33 **AND** Output < 0.67 **THEN** Wine *is* Class II.

$\Re 3$: **IF** Output ≤ 0.67 **THEN** Wine *is* Class III.

This data set has eleven inputs and one output pertaining to three classes as shown above.. Some of the inputs consisted of: alcohol, malic acid, ash, etc. Four rules were identified by using ie^{2}TS. The identified MFs are illustrated in Figure 5.17. The rule generation process is illustrated in Figure 5.18. It is clear that after about 80 samples it converged to 4 rules. During the identification procedure ie^{2}TS and eTS misclassified 1 instance, whereas e^{2}TS misclassified 73 instances.

Remark B: Due to the nature of e^{2}TS, in which cluster centers are chosen without verifying the potential of the cluster center, it is possible for e^{2}TS to generate significantly more identification errors than the other techniques. This is evident from the results as illustrated in Table 5.3.

Wine Benchmark Data								
Method	**Number of Clusters**	**Epochs**	**Avg. Training Time per Epoch (s)**	**Identification Errors**	**Average Training Errors**	**Testing Errors**	**Rate**	**Avg. RMSE**
eTS-DEKF	7	100	12.6015	1	18.06	11	81.667	0.1438
e^{2}TS-DEKF	4	100	4.0848	73	150.86	39	33.146	0.9732
ANFIS-DEKF	7	100	12.1173	n/a	71.00	35	28.090	0.5448
ie^{2}TS-DEKF	4	100	4.2718	1	18.52	12	80.000	0.1458
ie^{2}TS-sDEKF	4	100	4.2252	1	6.48	8	86.670	0.1187

Table 5.3 - Wine benchmark data results

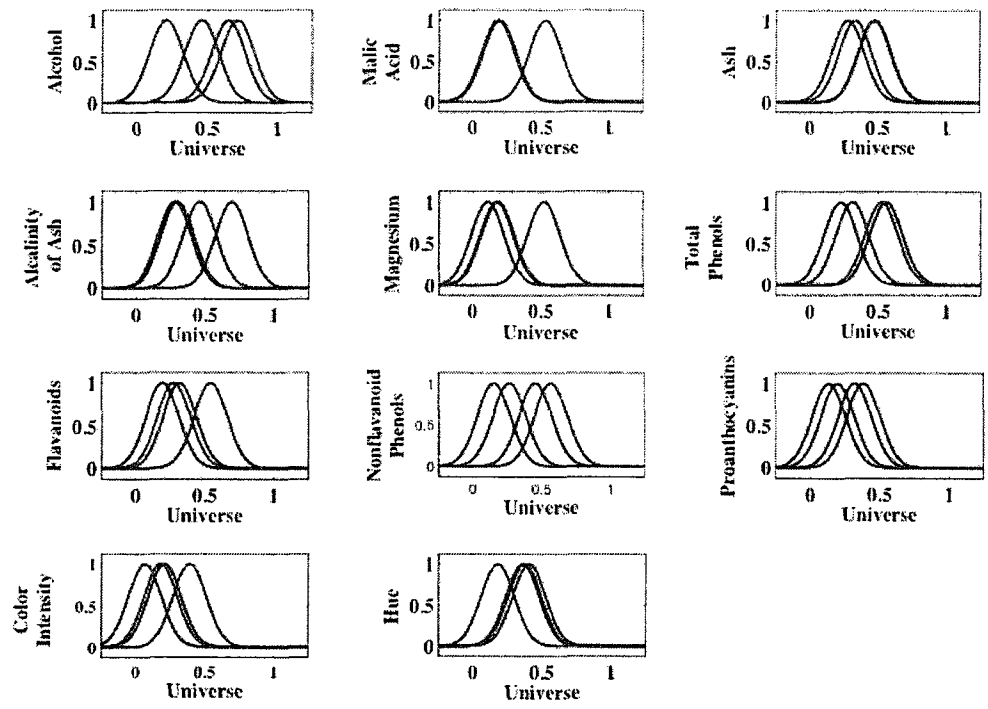

Figure 5.17 - Identified MFs using ie^{2}TS-sDEKF - Wine

Remark C: It can be noted from comparing the identification errors and the testing errors that it is sometimes possible to obtain a larger amount of errors in testing the network than was found in the initial identification. One of reasons for this problem is that the data used in identification is different from that employed in testing the network.

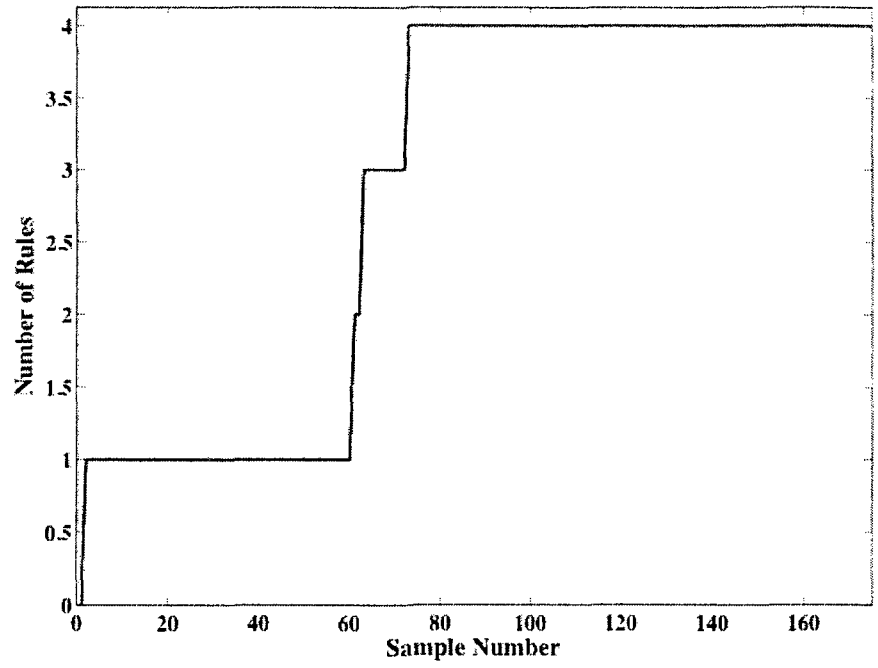

Figure 5.18 - Generation of rules using ie^{2}TS-sDEKF - Wine

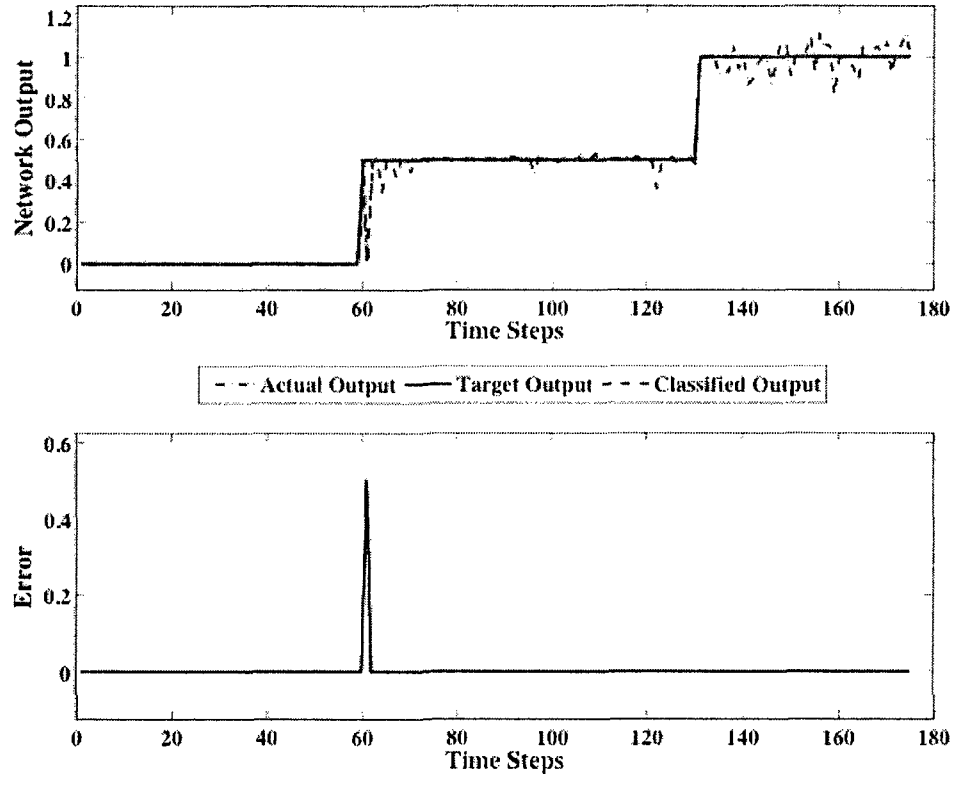

Figure 5.19 - Identification network output and classification error using ie^{2}TS-sDEKF - Wine

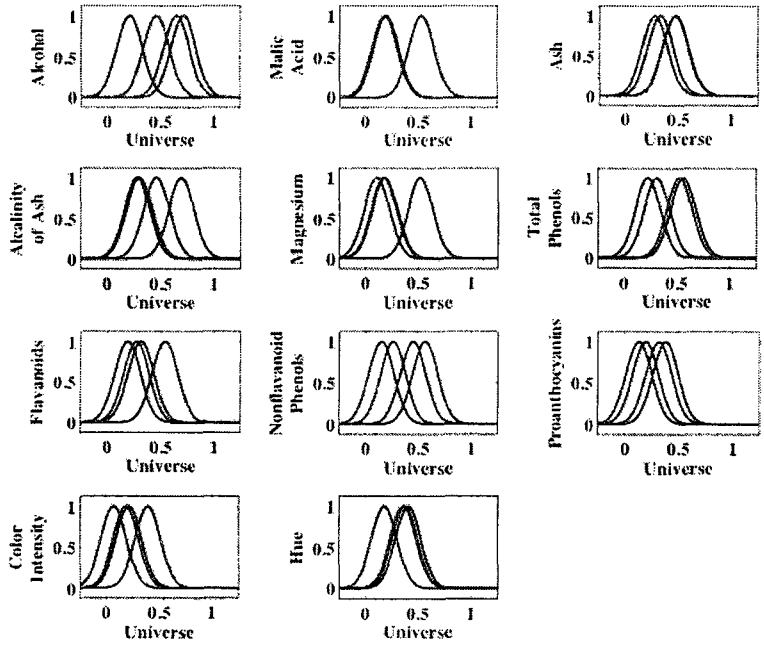

Figure 5.20 - Updated membership functions using ie^{2}TS-sDEKF - Wine

In training, it can be observed from Figures 5.21 and 5.22 that ie^{2}TS-sDEFK converged to an RMSE of about 0.12 after 10 epochs. This corresponds to 8 testing errors and 6.48 training errors. The other techniques using DEFK generated more errors, which can be observed in Table 5.3 and in Figures 5.23 and 5.24 where the RMSE convergence is shown per epoch.

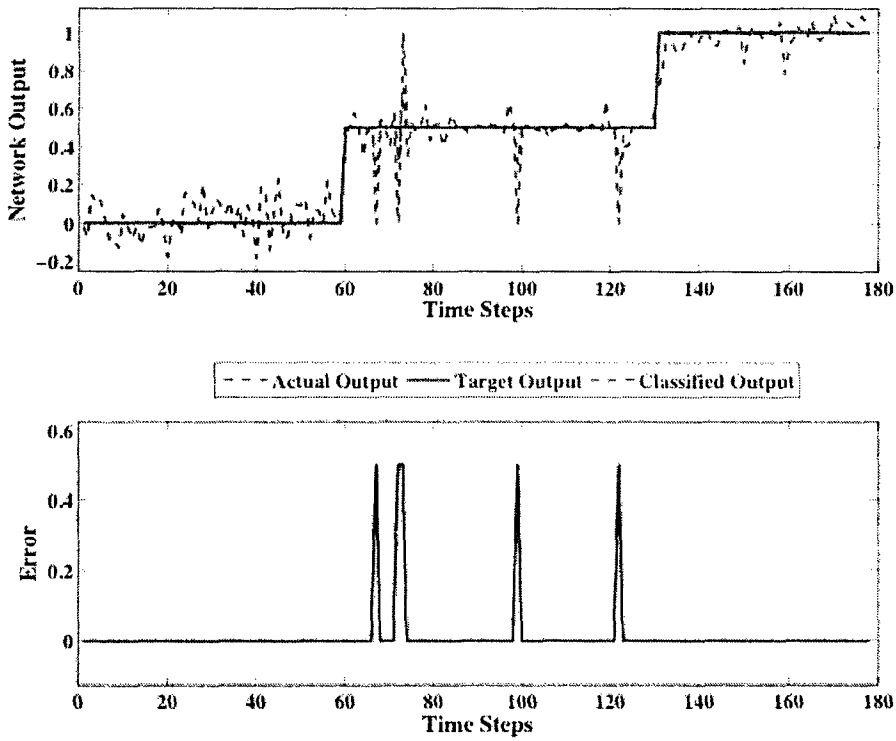

Figure 5.21 - Trained network output and classification error - Wine

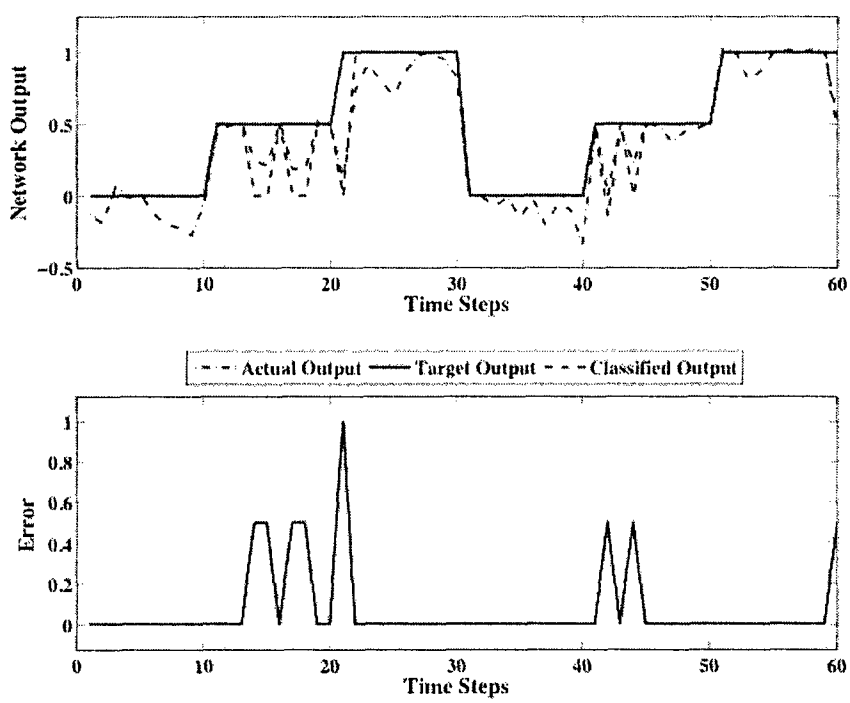

Figure 5.22 - Verification network output and classified error using ie^{2}TS-sDEKF - Wine

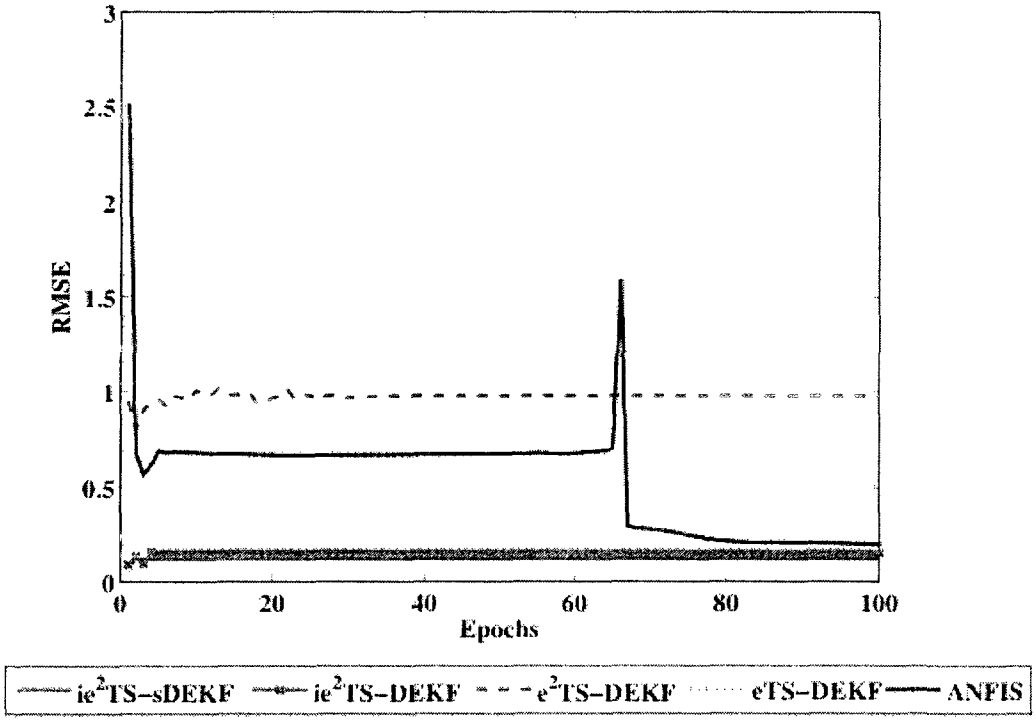

Figure 5.23 - Root mean squared error - Wine

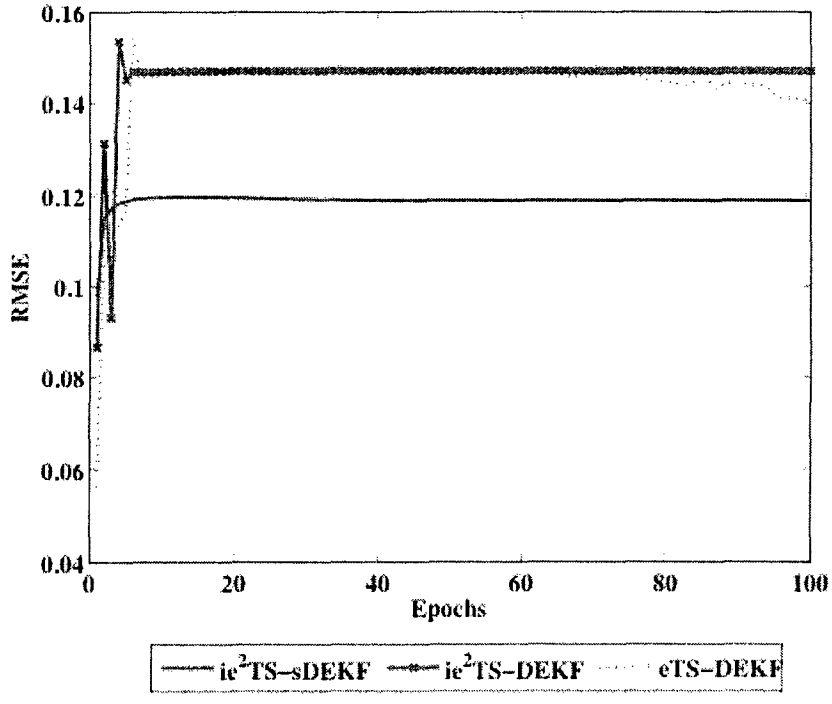

Figure 5.24 - Root mean square error - close up - Wine

From the results, one can state with confidence that the proposed technique ie^{2}TS-sDEKF is an efficient evolving classifier. In all three test instances, the proposed technique generated the least clusters/rules. This is because the introduced design parameter, ς , can make the identification technique more flexible. This was explained in detail in Chapter 3. The efficient classification performance of the proposed ie2TS-sDEKF technique can also be recognized by the number of errors in identification, training and testing. Although the proposed classifier technique requires slightly more computational effort than the existing ones, the amount of additional effort is negligible particularly offline training operations.

Chapter 6

6.0 Gear Fault Diagnosis

"Machines will be capable, within twenty years,
of doing any work that a man can do."
HERBERT SIMON (1916-2001)

The primary objective of this chapter of this thesis is to implement the developed techniques for gear fault detection. The main results of this section utilize the proposed clustering technique (i.e. ie^{2}TS) for identifying the network and the training technique (i.e. sDEKF) to train the network. To make a comparison, test results from other related networks will also be discussed.

This chapter is organized as follows: Section 6.1 gives a description of the experimental apparatus, Section 6.2 introduces the employed condition indicators whereas Section 6.3 describes the classification procedure and test results.

6.1 Experimental Apparatus

To study the signatures of the gearbox faults, the SpectraQuest's Machinery Dynamics Simulator has been used, as shown in Figure 6.1. This system is set up in the

Laboratory for Intelligent Mechatronic Systems (LIMS) at Lakehead University. It reflects a modular design that provides versatility, operational simplicity and robustness.

The experimental setup is driven by a 3Hp DC motor with a speed range from 20 to 4200 RPM. The shaft's rotational speed is controlled by a speed controller (Delta VFD-PU01).

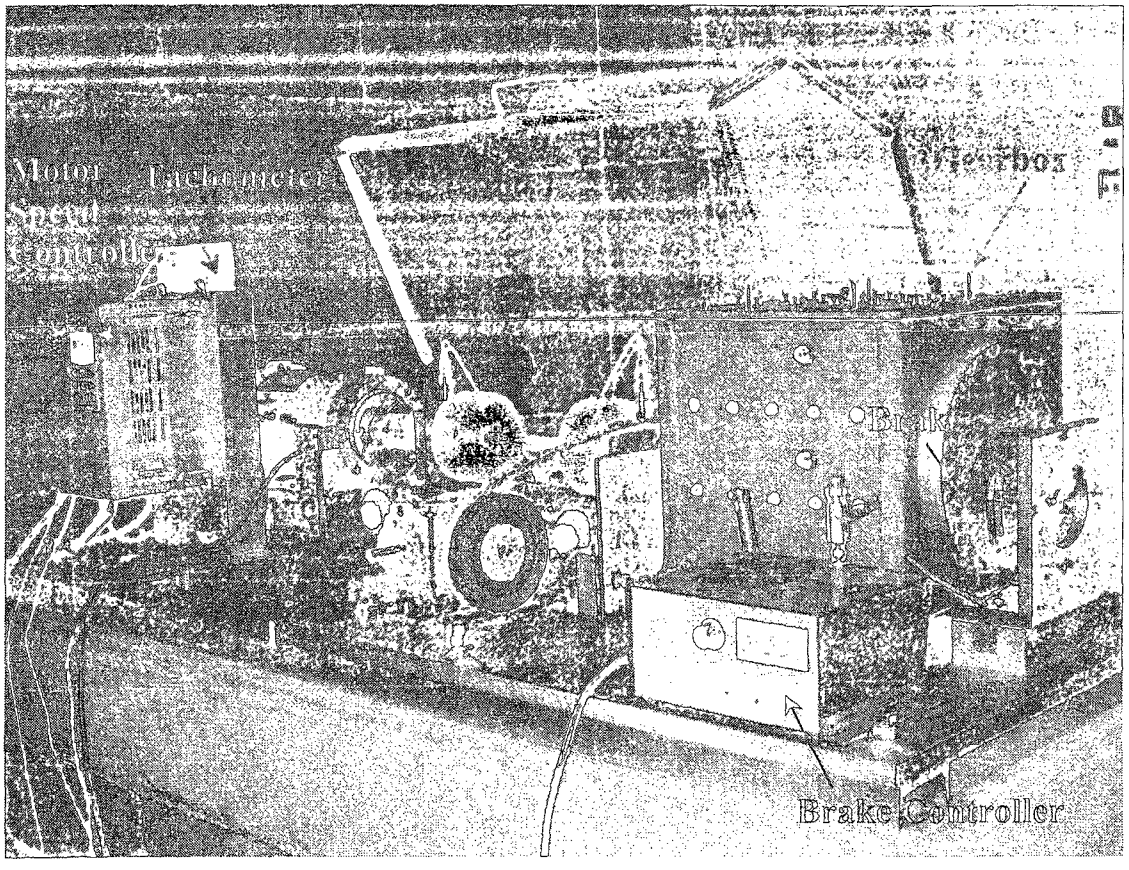

Figure 6.1 - Machinery fault/gearbox dynamics simulator

The collected signals are fed to the computer through a data acquisition board (NI-DAQ PCI-4472) which has built-in A/D converters and antialiasing filters. Vibrations induced by the gears were collected using industrial, ceramic shear ICP accelerometers (ICP-IMI, SN98697) with a sensitivity of 100mV/g. The accelerometers are secured on the gearbox as shown in Figure 6.2. The reference signals are collected using a mini-bean high-speed retro optical sensor (SM312LVMHS). Two disks apply a static load whereas the variable load is applied by a magnetic brake system (Placid Industries, B150-24-H) through a bevel gearbox and a belt drive.

Real-time Matlab code has been developed to control the data acquisition processes. Signal analysis and classification operations are also performed in MATLAB. Signals are collected using a sampling frequency $F_s = 10^5 Hz$ in order to attenuate the noise.

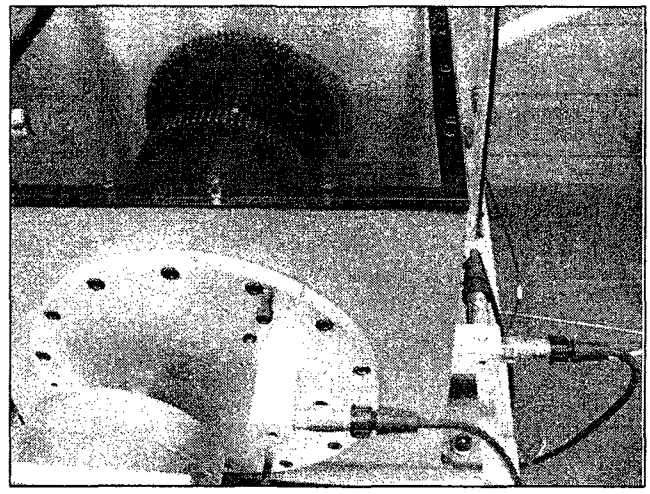

Figure 6.2 - Gearbox and accelerometers

There are three shafts in the gearbox: the input shaft, the intermediate shaft and the output shaft. The configuration of the spur gears in the gearbox is presented in Figure 6.3

[50] whereas the configuration of the helical gears can be seen in Figure 6.4. The results presented in this work are pertaining to the gear on the input shaft.

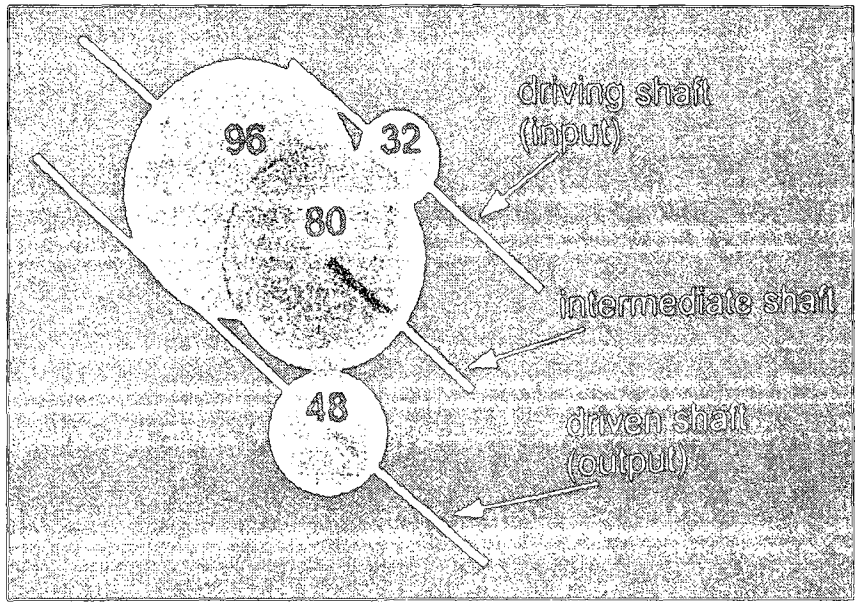

Figure 6.3 - Spur gears configuration

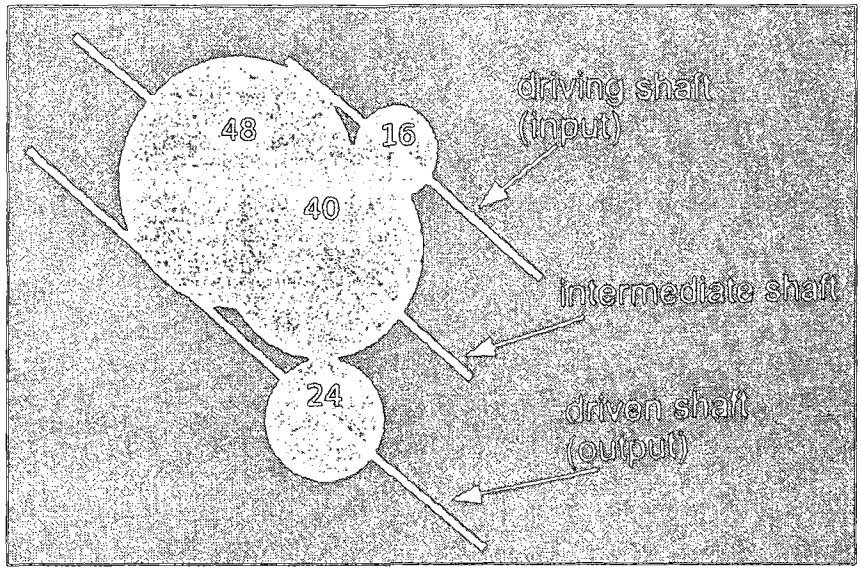

Figure 6.4 - Helical gears configuration

Since the scope of this research is to detect faults in gearboxes, various faulted gears have been introduced: Figure 6.5 shows a helical gear with a 40% chipped tooth and Figure 6.5 illustrates a spur gear with a 90% missing tooth. Other gear conditions used for testing were severe cracked gears, minor cracked gears, and of course, healthy gears.

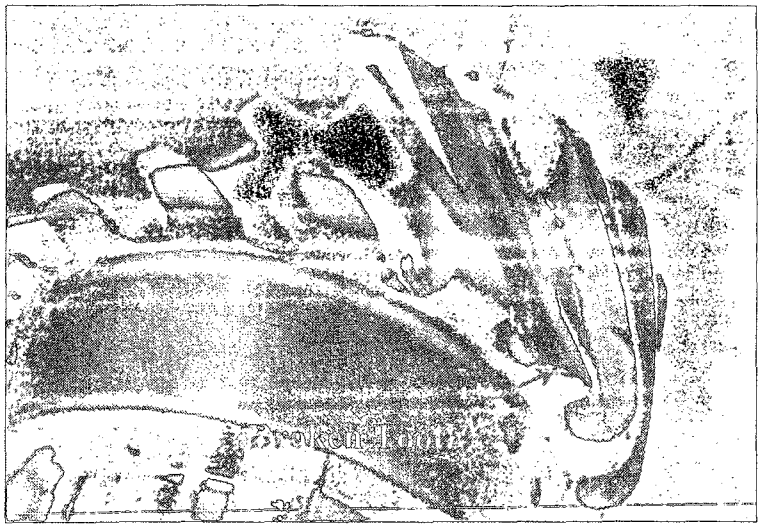

Figure 6.5 - Broken tooth on a helical gear

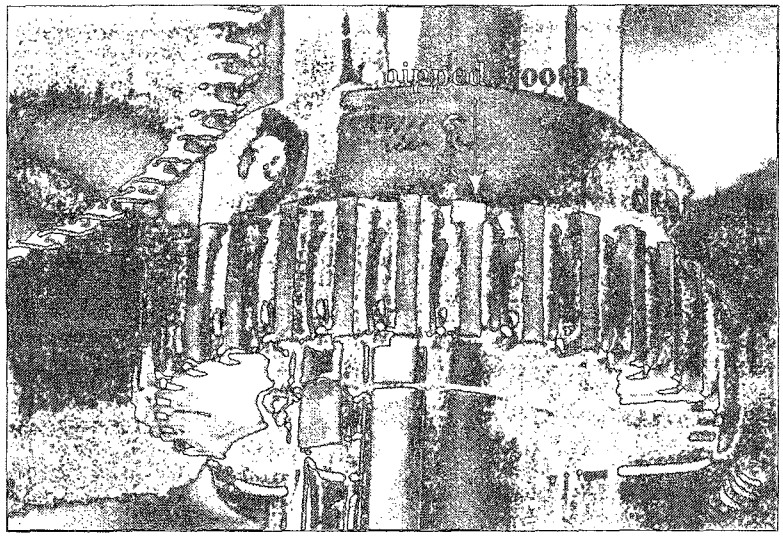

Figure 6.6 - Chipped tooth on a spur gear

The collected signals under different healthy and faulty gear conditions will be analyzed for automatic fault detection as discussed in the following sections.

6.2 Condition Indicators

As mention in Chapter 1, one of the most popular methods for gearbox diagnosis used in literature is the time domain analysis. First, the time synchronous average is performed such that any noises non-synchronous with the gear being inspected are removed. Although this process has been proven to be efficient in filtering out noise and other such events, it has a disadvantage when it comes to detecting slight damages since very small variations in the synchronous time average may occur. These variations may be so small that any change in tooth condition would be hard to detect against the dominated pattern of the tooth meshing vibration. Condition indicators are then calculated using different methods. Although the root mean square value (RMSV) is not sensitive to early stage defects [3], it is the simplest method employed to determine the overall health of a gear. Crest Factor is another indicator used in literature [51], [3]. Authors have proven that the crest factor is a good indicator of small size defects; although, when localized damage propagates, the value of the crest factor decreases significantly due to the increasing RMSV. Kurtosis is also a well-used condition indicator as it gives valuable information regarding the peakedeness of the signal. Although, researchers [3] have found the Kurtosis value to be more useful, when it is compared with the RMS and crest factor when monitoring bearings, these methods were employed for monitoring gears as well [51]. Another condition indicator utilized for classification of gears in this study is energy

operator (EO). Computed similar to Kurtosis, energy operator also provides valuable information regarding the state of the gears being monitored.

6.2.1 Time Synchronous Average (TSA)

Since gear signals are periodic, it is possible to extract the signature corresponding to each gear in the gearbox by using the TSA technique. TSA was firstly proposed by McFadden [52]. It is an averaging process over a large number of cycles, synchronous with the running speed of a specific shaft in the gearbox. As a result, signatures that are not synchronous with the rotation of the gear being monitored will be filtered out over sufficient rotation cycles, and the resulting feature specific to this gear is represented only over one complete revolution in the time domain [53]. In the case of a fault being present in the monitored gear, an impulse signal would be present in the signal average, which will produce additional amplitudes and phase modulations of the vibration signal.

Assuming the collected vibration signal $y(t)$ consists of R revolutions, each revolution has L data samples and the time interval between two samples is $\Delta t = 1/F_s$ where F_s is the sampling frequency, then the TSA signal can be represented as follows:

$$x(t)=\frac{1}{R}\sum_{r=0}^{R-1} y(t+rL\Delta_t) \tag{6.1}$$

where $t = k\Delta_t$, $k = 0,1,...,(L-1)$, $\Delta_t = 1/f_s$ is the time interval between two samples with f_s being the sampling frequency, L is the number of data samples and R is the number of revolutions obtained from the vibration signal $y(t)$. R should be taken as large as possible

in order to reduce the noise efficiently. The obtained signal is referred to as signal average. Figure 6.7 illustrates the TSA processes.

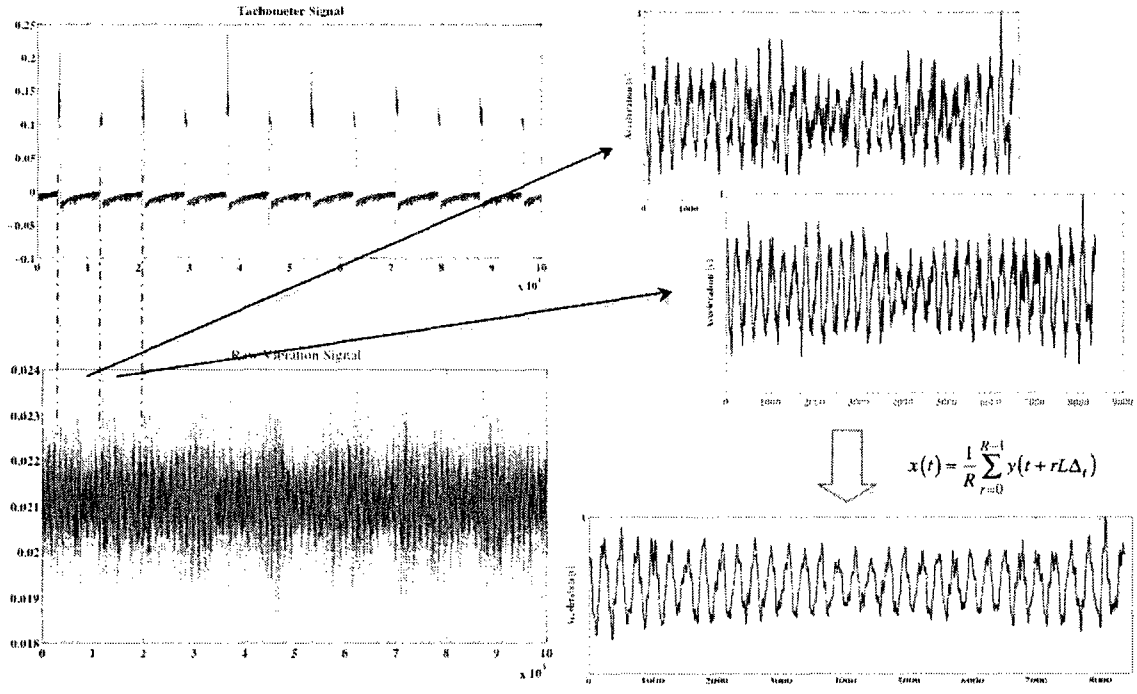

Figure 6.7 - Extraction of gear signature via time synchronous average

Several revised TSA techniques have been proposed in literature. For example, in [54], Vachtsevanos introduced the frequency domain TSA in which the discrete Fourier Transform is taken between each tachometer zero crossings. Combet [55] suggested a tachometer less TSA to reduce influence of noise in tachometer readings. All these techniques were evaluated in [56] by Bechhoefer et al.

6.2.2 Kurtosis

In mathematical terms, kurtosis is the forth moment of the signal normalized by the square of the variance. Consider a time signal s, the kurtosis is defined as:

$$Kurt = \frac{N\sum_{i=1}^{N}(s_i - \bar{s})^4}{\left(\sum_{i=1}^{N}(s_i - \bar{s})^2\right)^2} \tag{6.2}$$

where $Kurt$ is kurtosis, N is the number of total sample points in time signal s, s_i is the i^{th} point in time signal s and $\bar{s}$ is the mean value of the time signal s.

Kurtosis can describe the shape of the signal [51, 57]. More specifically, kurtosis is a measure of how peaky or flat a signal is. A signal containing sharp peaks with high amplitude would return a higher kurtosis value, which, in general, is attributed to a faulty gear [51].

6.2.3 Energy Operator

Consider a signal x, the energy operator (EO) is defined as:

$$EO = \frac{N^2\sum_{i=1}^{N}(\Delta x_i - \Delta\bar{x})^4}{\left(\sum_{i=1}^{N}(\Delta x_i - \Delta\bar{x})^2\right)^2} \tag{6.3}$$

where $\Delta\bar{x}$ is the mean value of signal Δx, N is the number of points in dataset x and $\Delta x_i = s_{i+1}^2 - s_i^2$ where s_{i+1}^2 and s_i^2 are the $(i+1)^{\text{th}}$ and i$^{\text{th}}$ data points in the signal average, respectively.

The energy operator is also sensitive to fluctuations in the signal. A signal with high amplitude peaks would have a higher energy operator therefore a indicating a faulty gear.

6.2.4 Crest Factor

The crest factor (CF) is defined by ratio of the peak-to-peak value of the signal average to the RMSV of the signal average as follows:

$$CF = \frac{s_{peak-peak}}{s_{rmsv}} \tag{6.4}$$

where CF is the crest factor, $s_{peak-peak}$ is the peak-to-peak value of the residual signal s, and s_{rmsv} is root mean square value given by:

$$s_{rmsv} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} s_i^2}$$

N is the number of points in the signal average s, and s_i is the i^{th} component of the signal average, s.

The purpose of the crest factor calculation is to give an analyst a quick idea of how much impacting is occurring in a waveform. Impacting is often associated with roller bearing wear, cavitation and gear tooth wear. This parameter enables very tiny damages to be detected at an early stage [51].

6.3 Gear Fault Diagnosis Using the Proposed EF Classifier

In order to make a decision related to the condition of gears in the gearbox, which are inaccessible without dismantling the machine, it is important to analyze external relevant information. Typically, the most relevant information comes from different types of vibration signals. Running speeds and loading conditions are two important parameters affecting the vibration levels in gearboxes [57]. Since these parameters are usually held constant in condition monitoring, changes in vibration signals are mostly attributed to faulty gears such as, tooth breakage, tooth fracture and pitting.

The condition monitoring work presented here has been conducted in spur gears as well as helical gears. Although sometimes it is possible to recognize the fault condition in spur gears directly from the vibration signature, this is not always the case, especially in real industrial situations where noise signals disrupt the primary signature. In the case in which helical gears are used, classification by signal signature alone is almost impossible in real applications [53]. As a result, tools for automatic fault detection tools using decision-making intelligent schemes are sought. Therefore, the developed ie^{2}TS-sDEKF technique will be implemented for gear fault diagnosis.

The condition monitoring scheme starts with processing the vibration signals collected with the three accelerometers shown in Figure 6.2. However, the results shown in this work are pertaining to the accelerometer closer to the gear being monitored and perpendicular to the shaft. Although many tests were performed under different shaft speeds and different loading conditions, some results, obtained from the operating

conditions of shaft speed of 15Hz with the loading levels of no loading, 100mA, 150mA and 250mA will be used to demonstrate the effectiveness of the proposed techniques. The signals are time synchronous averaged in order to extract the vibration signature of the gear to be analyzed from the total vibration of the gearbox. Figures 6.8(a)-6.8(c) illustrate an example of TSA signal for a healthy helical gear whereas Figures 6.9(a)-6.9(c) illustrates a faulty helical gear. A change in the vibration signal is observable in the three graphs of both figures representing different loading conditions. However, when comparing Figure 6.8 with Figure 6.9 (i.e. faulty gear signal with healthy helical gear) under same loads (e.g. Graph (a) - no load) a conclusion regarding the condition of the two gears is unattainable. This was the case for all experiments. A simple investigation of the averaged signals could not draw any conclusive results with confidence. In fact, it is suggested that this would be the case in real industrial applications.

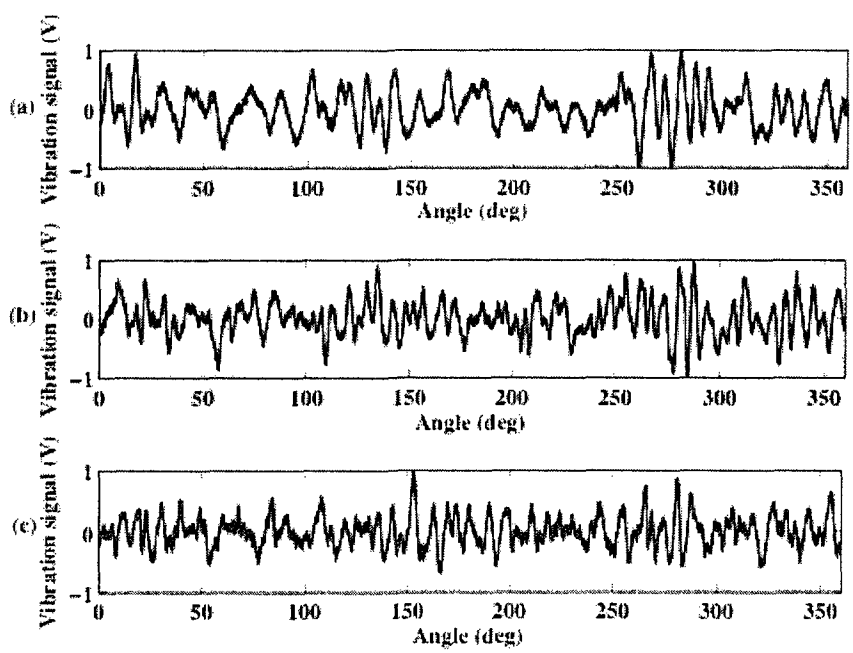

Figure 6.8 - Time synchronous average for a healthy helical gear: (a) – no load, (b) – load level of 150mA, (c) – load level of 250mA

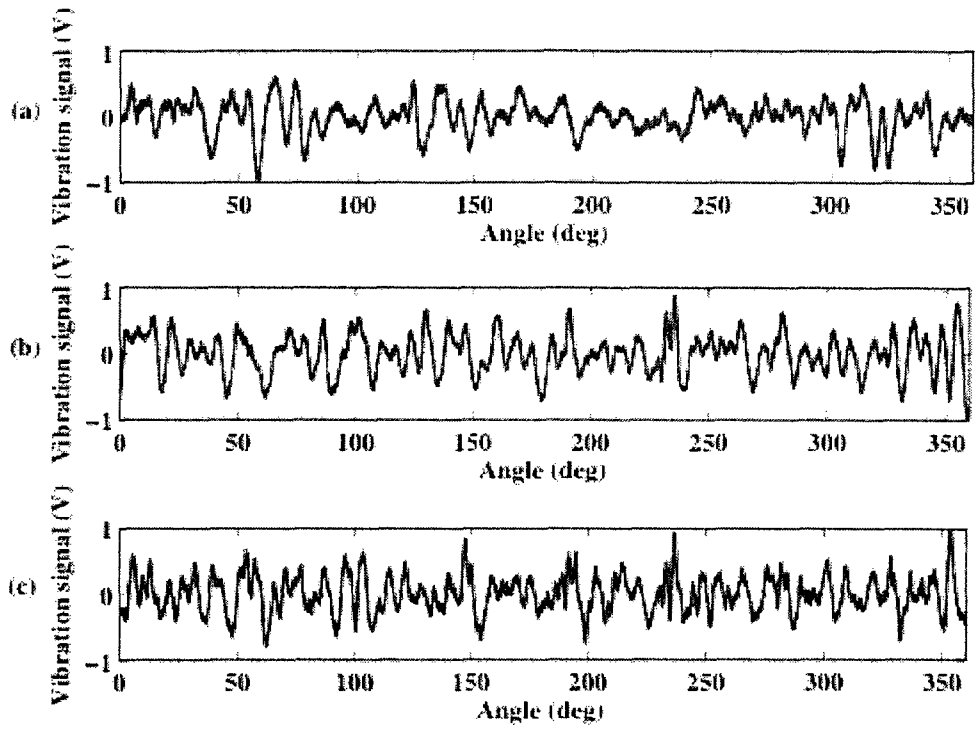

Figure 6.9 - Time synchronous average for a faulty helical gear: (a) – no load, (b) – load level of 150mA, (c) – load level of 250mA

One of the primary reasons why it is more difficult to detect fault in helical gears directly from the averaged time signal is related to the meshing properties. In a helical gear the load is shared among more pairs of mating teeth than in spur gears; the changes in vibration magnitude become less significant when a tooth enters or leaves the meshing region under load. As a result, it becomes more difficult to examine feature modulations due to defects in helical gears.

Figure 6.10(a)-6.10(c) and 6.11(a)-6.11(c) are examples of TSA signals for a healthy spur gear and a faulty spur gear, respectively. When inspecting Figures 6.10(a)-6.10(c), it is clear that the load conditions have an impact on the resulting averaged time signal. To some extent, when the load increases in the gear train the signature modulation

due to gear defect will become more prominent. This can be attributed to changes in the signal to noise ratio. Consequently, it is difficult to recognize difference between the gear signatures with no load (i.e. Graph (a)) and with 150 mA load level (i.e. Graph (b)); however, the signature of the gear is recognizable in Graph (c) when a load level of 250 mA was applied.

From visual inspection of Figures 6.8 to 6.11, changes in vibrations can be seen in the signatures of the spur gears under different loading conditions, as well as those of helical gears. Since it is difficult to identify clear signature modulations that may be related to faulty gear conditions, advanced analysis using more features is required, as discussed next.

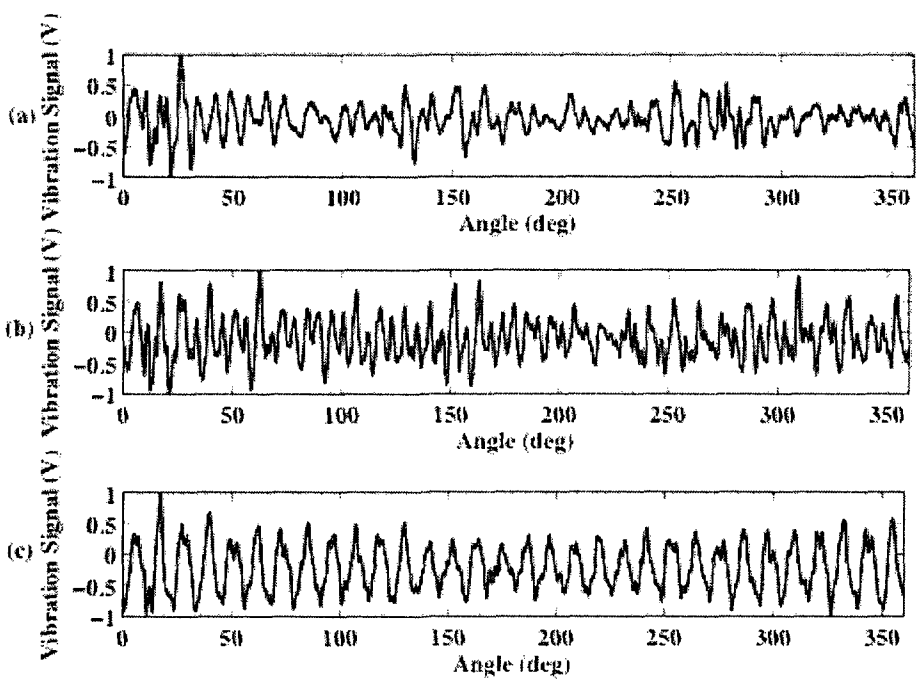

Figure 6.10 - Time synchronous average for a healthy spur gear: (a) – no load, (b) – load level of 150mA, (c) – load level of 250mA

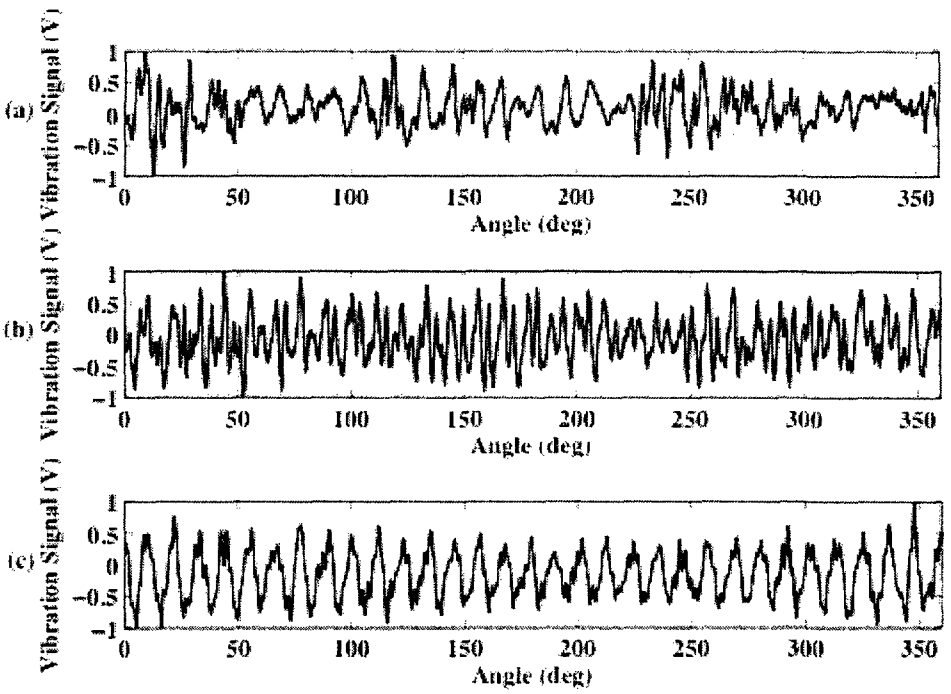

Figure 6.11 - Time synchronous average for a faulty spur gear: (a) – no load, (b) – load level of 150mA, (c) – load level of 250mA

Condition monitoring indicators have been computed for different fault conditions and under different loading conditions for both spur and helical gears. These indicators are computed from features of the related signal processing techniques. Although many real life applications such as in marine and aeronautics require condition analysis of the gears per tooth, this study focused on individual gears on the gear train with the scope of deciding the condition of the entire gear without focusing on the type of fault.

Tables 6.1 and 6.2 display some values of the condition indicators of the related the aforementioned techniques. By examining the condition indices, one conclusion can be drawn: the condition indicators are not robust for gear fault detection. For example, there is no expected significant change in the values of the kurtosis indicator for a healthy and

damaged spur gear (refer to Table 6.1) under a load level of 250mA. The situation is even worse as the faulty gear indicators are even lower than the ones for the healthy gear for 0mA and 150mA load levels. Similar conclusions can be drawn in examining the processing results using the crest factor indicator and the energy operator indicator, respectively. One reason for the discrepancy between CI's of faulty and healthy gears can be attributed to the averaged time signal being corrupted by noise. Notably, the signal to noise ratio is lower for the cases with lesser load.

Condition Indicators for Spur Gears			
Loading conditions: 0			
Kurtosis	Crest Factor	Energy Operator	Condition
3.909	0.23646	6.1531	healthy gear
4.8036	0.2203	6.4672	
4.7033	0.229	5.8284	
4.6247	0.23571	5.6874	
4.9348	0.2272	7.118	
3.3718	0.25725	6.673	faulty gear
3.8774	0.23451	6.1721	
3.4649	0.23928	5.9235	
3.3108	0.24717	5.7699	
4.0645	0.23611	7.716	
Loading conditions: 150			
2.8831	0.30385	5.7217	healthy gear
2.7186	0.32037	5.2966	
2.6828	0.32855	5.5347	
2.6856	0.32268	5.5027	
2.7255	0.33142	5.2165	
2.4183	0.34603	5.9678	faulty gear
2.4059	0.32035	4.7182	
2.3753	0.35584	4.9798	
2.5332	0.35116	6.1804	
2.5074	0.33015	4.6591	
Loading conditions: 250			
2.0015	0.34678	5.4112	healthy gear
2.1035	0.33984	6.2695	
2.0786	0.32725	5.1522	
1.8998	0.42837	6.1234	
2.0431	0.35836	5.2894	
2.1412	0.31985	5.3088	faulty gear
2.0494	0.35569	6.1688	
1.9612	0.34872	6.0344	
1.8543	0.38043	5.8159	
2.0729	0.34559	5.486	

Table 6.1 - Spur gear condition indicators using different techniques

Condition Indicators for Helical Gears			
Loading conditions: 0			
Kurtosis	Crest Factor	Energy Operator	Condition
2.9794	0.32458	5.2458	healthy gear
2.9788	0.31438	6.7411	
2.9815	0.2894	5.0539	
2.733	0.32193	4.8636	
2.7115	0.34152	5.6053	
6.3388	0.18785	5.0913	faulty gear
5.2069	0.20335	4.6823	
4.9681	0.19967	5.5381	
6.4048	0.18353	4.8808	
5.2863	0.20632	5.0246	
Loading conditions: 100			
2.9542	0.33105	5.3425	healthy gear
2.9744	0.32405	5.5813	
2.7816	0.32674	5.3804	
2.9433	0.31456	6.3746	
2.9537	0.33572	5.5516	
3.0021	0.29318	5.4503	faulty gear
2.9912	0.295	7.3838	
2.9304	0.30131	6.4985	
3.0043	0.28806	5.9877	
3.142	0.2895	5.7547	
Loading conditions: 250			
3.2869	0.25728	5.4399	healthy gear
3.2997	0.25429	5.6884	
3.149	0.2639	4.9158	
3.2954	0.26282	4.8596	
3.5134	0.25292	4.4767	
3.1675	0.26167	5.3027	faulty gear
3.1324	0.25036	5.4033	
3.3559	0.24758	4.6343	
3.2061	0.25419	4.7488	
3.2725	0.25369	4.8626	

Table 6.2 - Condition indicators for helical gears

It can be concluded that that the selected indicators are not very robust, but each has its own advantages and limitations. It is difficult to achieve reliable gear fault diagnosis if only one indicator is used in this case. Corresponding, the strategy in this work is to use the proposed EF classifier to integrate the strengths of the selected indicators so as to provide a more positive assessment of the gear health conditions.

In classification analysis the same condition indices computed above are used as inputs intro the proposed classifier. The network has one output, further classified into two classes as follows:

$\Re_1$: **IF** Output < 0.5 **THEN** Gear is Faulty

$\Re_2$: **IF** Output ≥ 0.5 **THEN** Gear is Healthy

An output of one represents 100% healthy state whereas an output of zero indicates the 100% faulty condition. The appropriate number of rules has been created using the clustering techniques mentioned in Chapter 3. The classifiers are trained over 50 epochs using the data sets corresponding to the spur gears and helical gears, respectively. Both classifiers are trained under different values for η_r and η_q ranging from $\eta_r = 0.5 \sim 1.5$ and $\eta_q = 1.0 \times 10^{-5} \sim 1.0$ where $R = \frac{1}{\eta_r} I$, $Q = \frac{1}{\eta_q} I$ and I is the identity matrix. The results are discussed in the following subsections. Even though all classifiers were tested, the primary scope was to test the proposed classifier.

6.3.1 Spur Gear

The results obtained for spur gears have been obtained using $\eta_r = 0.85$ and $\eta_q = 1.0 \times 10^{-3}$. A total of eighty-five input/output data pairs for three loading conditions have been computed using the techniques presented previously (i.e. ie2TS-sDEKF). Fifty data pairs have been used to generate the necessary amount of clusters using the aforementioned clustering technique. The results of the clustering technique as well as the MFs and evolving rules are presented in Figures 6.12 and 6.13, respectively.

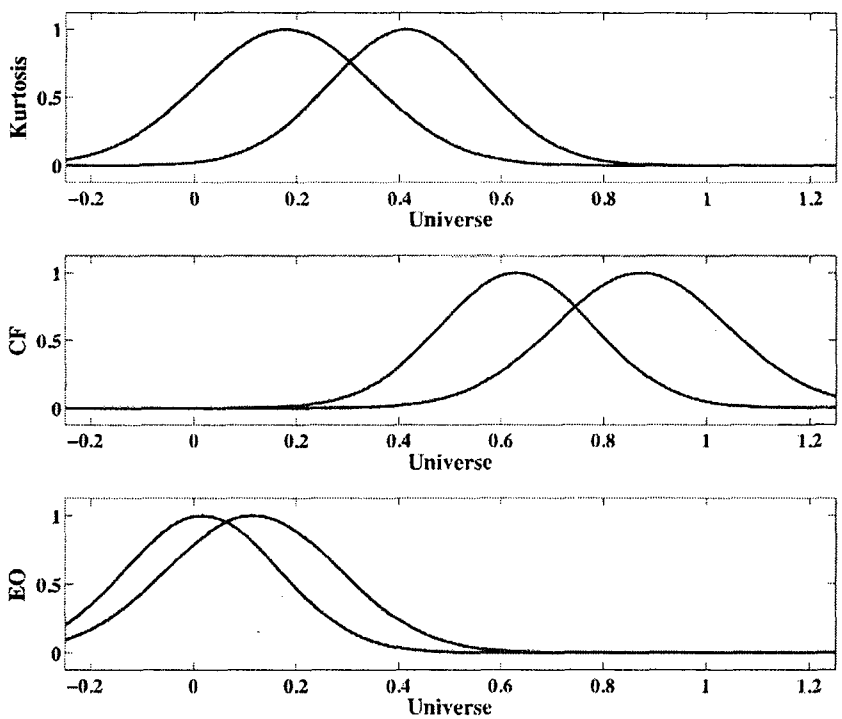

Figure 6.12 - Identified membership functions using ie^{2}TS-sDEKF - Spur gears

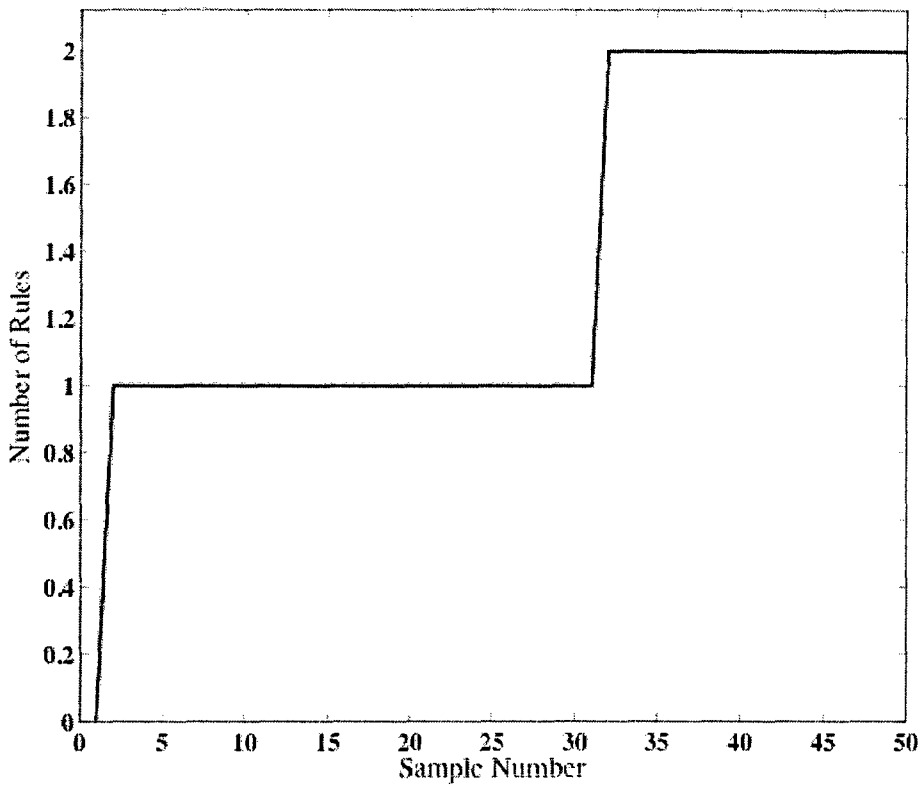

Figure 6.13 - Evolution of rules using ie^{2}TS-sDEKF - Spur gears

Figure 6.14 displays the output, where the demonstrated error is that produced after the identification of the EF classifier. It is clear that once the EF architecture is identified using the first fifty data sets, no sets have been misclassified in the following test process.

Other classifiers used for comparison produced one misclassified data set for the loading conditions however, the networks have done so by producing more clusters (i.e. fuzzy rules). Table 6.3 displays the results generated for different loading conditions.

Spur Gear Results								
Load =250								
Method	**Number of Clusters**	**Epochs**	**Avg. Training Time per Epoch (s)**	**Identification Errors**	**Average Training Errors**	**Testing Errors**	**Rate**	**Avg. RMSE**
eTS-DEKF	4	50	0.067462	2	4	4	92	0.3155
e^{2}TS-DEKF	4	50	0.066015	2	4.52	4	92	0.3112
ie^{2}TS-DEKF	2	50	0.041957	5	4.06	4	92	0.3166
ie^{2}TS-sDEKF	2	50	0.042156	5	3.86	4	92	0.3137
Load = 150								
Method	**Number of Clusters**	**Epochs**	**Avg. Training Time per Epoch (s)**	**Identification Errors**	**Average Training Errors**	**Testing Errors**	**Rate**	**Avg. RMSE**
eTS-DEKF	3	50	0.048902	1	1	1	98	0.1583
e^{2}TS-DEKF	3	50	0.050405	1	1	1	98	0.1577
ie^{2}TS-DEKF	2	50	0.03688	1	1	1	98	0.159
ie^{2}TS-sDEKF	2	50	0.037358	1	0	0	100	0.0956
Load = 0								
Method	**Number of Clusters**	**Epochs**	**Avg. Training Time per Epoch (s)**	**Identification Errors**	**Average Training Errors**	**Testing Errors**	**Rate**	**Avg. RMSE**
eTS-DEKF	3	50	0.04281	1	1.02	1	98	0.2271
e^{2}TS-DEKF	3	50	0.056417	1	1	1	98	0.2267
ie^{2}TS-DEKF	2	50	0.041573	0	1	1	98	0.2274
ie^{2}TS-sDEKF	2	50	0.041972	0	1	1	98	0.2108

Table 6.3 - Spur gear results

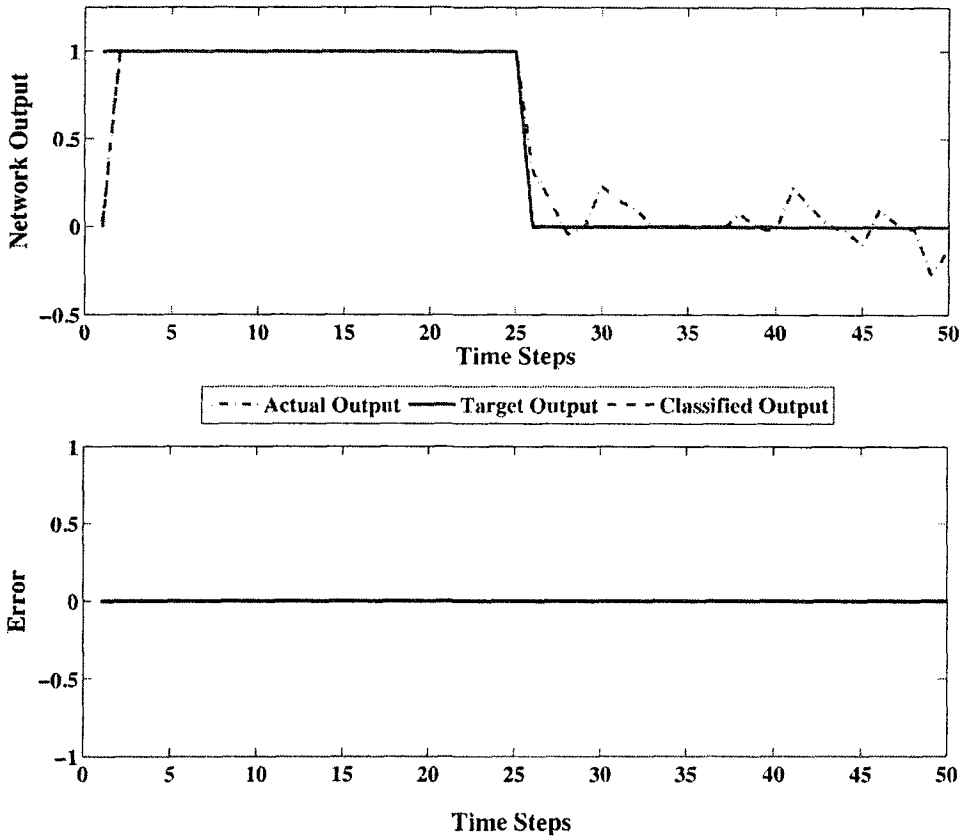

Figure 6.14 - Network identification output and classification error using ie^{2}TS-sDEKF - Spur gear

The MFs at the end of the training procedure using the proposed classifier, ie^{2}TS-sDEKF, are illustrated in Figure 6.15.

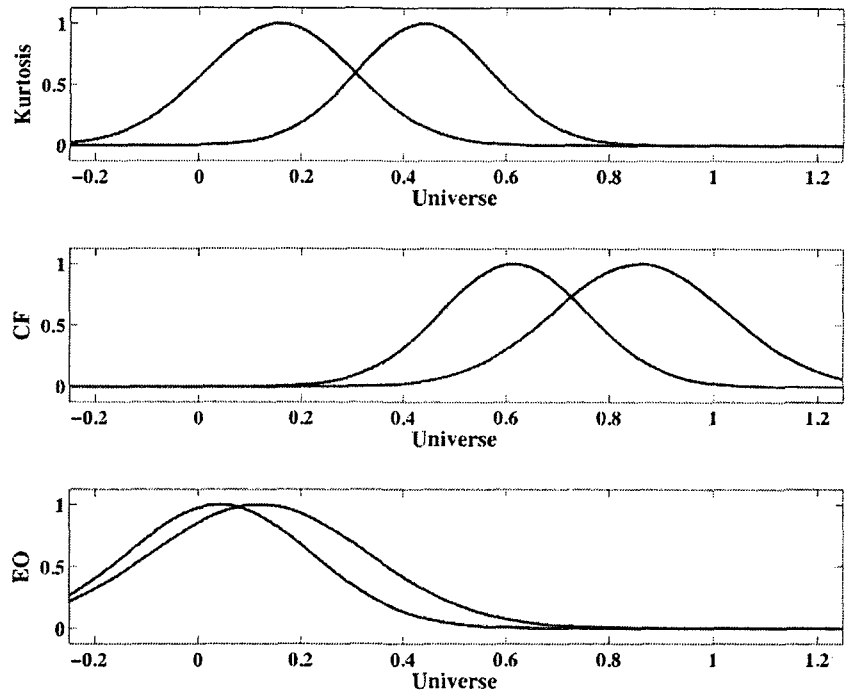

Figure 6.15 - Membership functions after training using ie^{2}TS-sDEKF - Spur gears

Figure 6.16 displays the output of the proposed classifier, ie^{2}TS-sDEKF, as well as the classification error. It can be seen that at the end of the training procedure, one instance has been misclassified. Figure 6.17 shows the testing results using 50 data sets. It is clear that the proposed classifier, ie^{2}TS-sDEKF, can effectively assess the gears' health conditions during the testing process, with only one instance misclassified. Although the classifiers used for comparison generated the same amount of errors, they did so at the expense of more clusters. Consequently, the efficiency of the classifiers used for comparison is degraded, as it can be observed in Table 6.3.

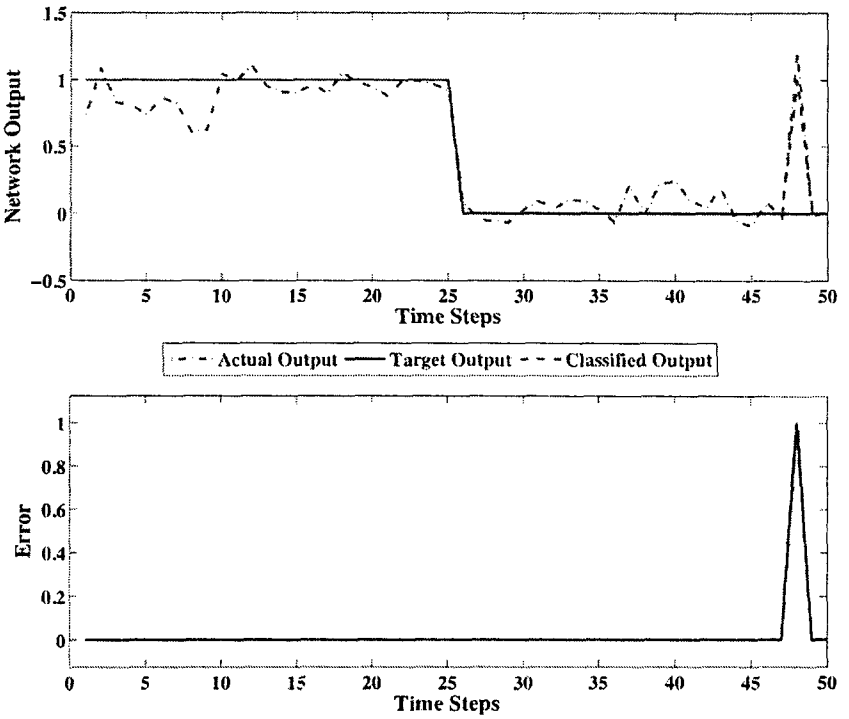

Figure 6.16 - Trained network output using ie^{2}TS-sDEKF - Spur gears

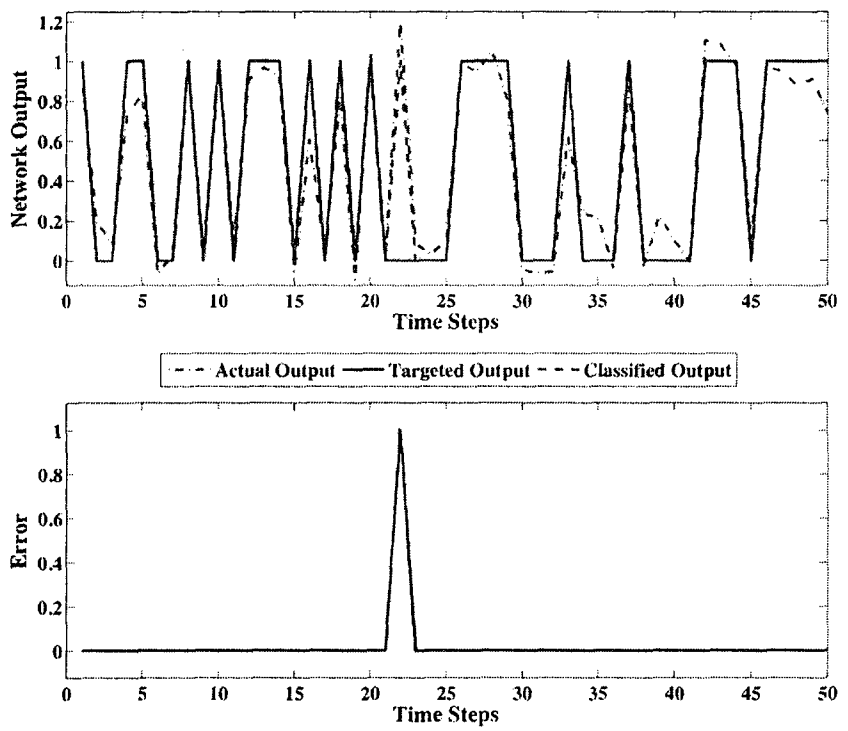

Figure 6.17 - Verification output and classification error using ie^{2}TS-sDEKF - Spur gears

When comparing the root mean square error (RMSE) produced during the training of all classifiers utilized for fault condition monitoring, the proposed training technique (i.e. ie^{2}TS-sDEKF) can effectively converge to a lower RMSE, as demonstrated in Figures 6.18 to 6.20. Although the RMSE is not significantly lower for the 250mA loading levels, the proposed classifiers is able to generate fewer clusters than the other classifiers. Also, the RMSE of the ie^{2}TS-sDEKF converged quickly to a desired value whereas the RMSE for the other classifiers stayed unstable for the entire training process. Evidently, the proposed ie^{2}TS-sDEKF classifier can effectively determine the states of the gear trains with a higher confidence level than other related classification schemes.

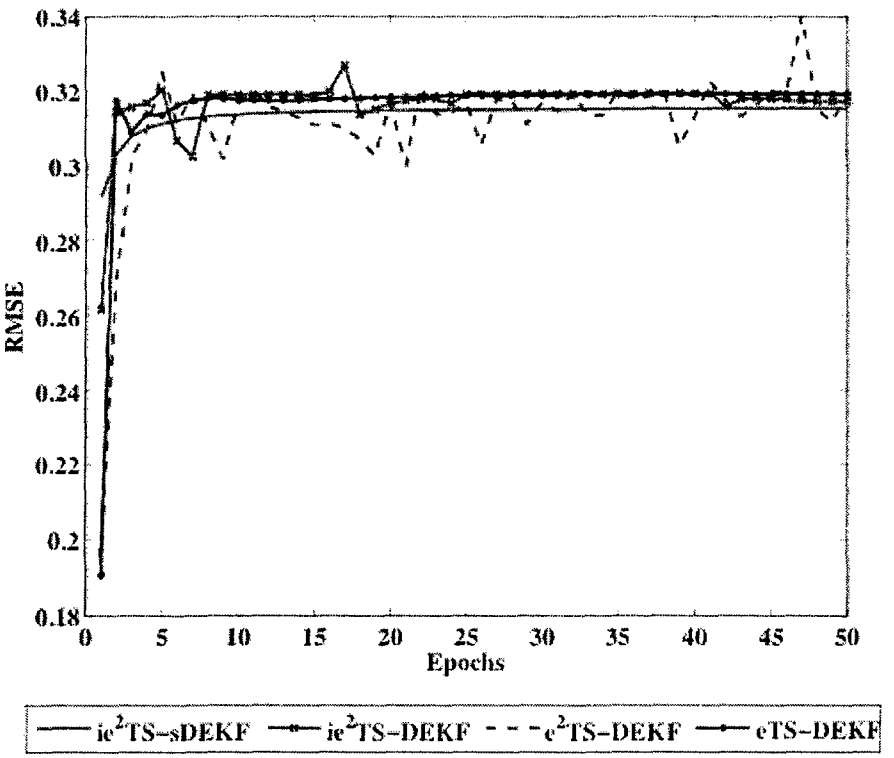

Figure 6.18 - Root mean square error for spur gears - 250mA load level

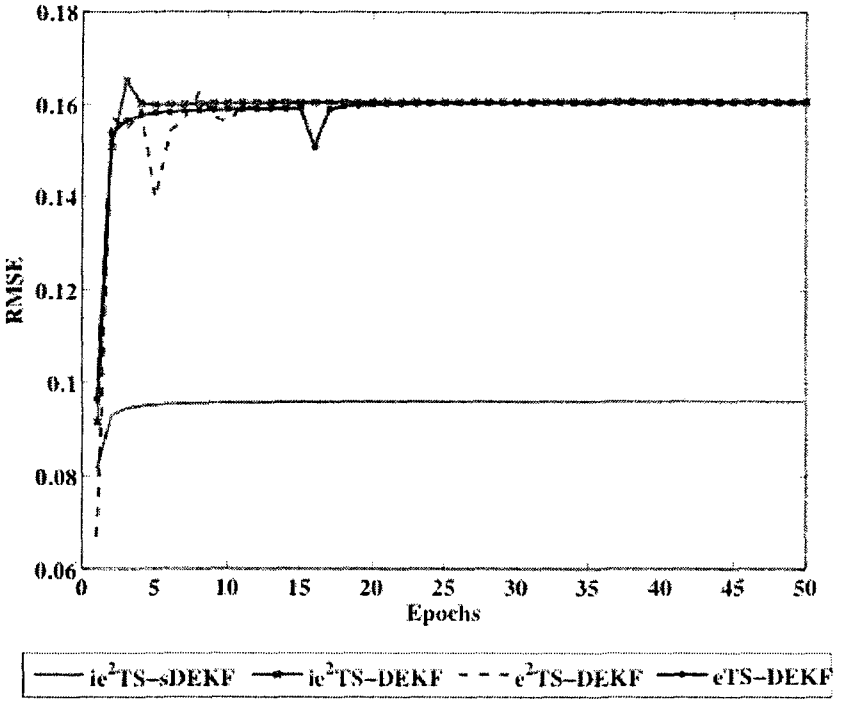

Figure 6.19 - Root mean square error for spur gear - 150mA load level

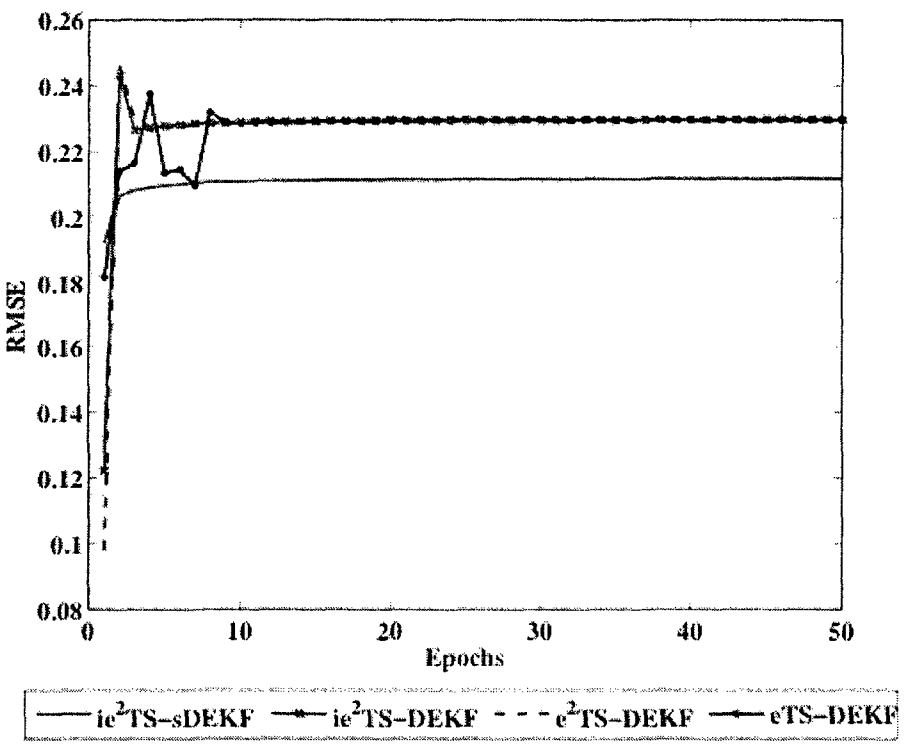

Figure 6.20 - Root mean square error - 0 load

Figure 6.21 displays the identified structure for spur gears classification.

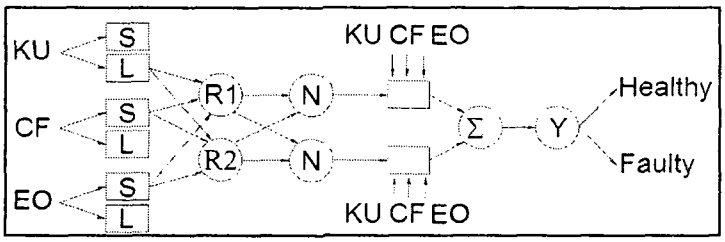

Figure 6.21 - Identified structure for spur gears using ie^{2}TS-sDEKF

6.3.2. Helical Gears

In this section, the proposed classification technique is implemented for fault detection of helical gears. Similar to spur gears, the results obtained for helical gears have been obtained using $\eta_r = 0.85$ and $\eta_q = 1.0 \times 10^{-3}$. 60 data pairs are used for clustering and training of the network where another 60 data pairs will be employed for verification. The developed ie^{2}TS-sDEKF classifier has generated two clusters (i.e., two rules), where the related MFs are shown in Figure 6.22.

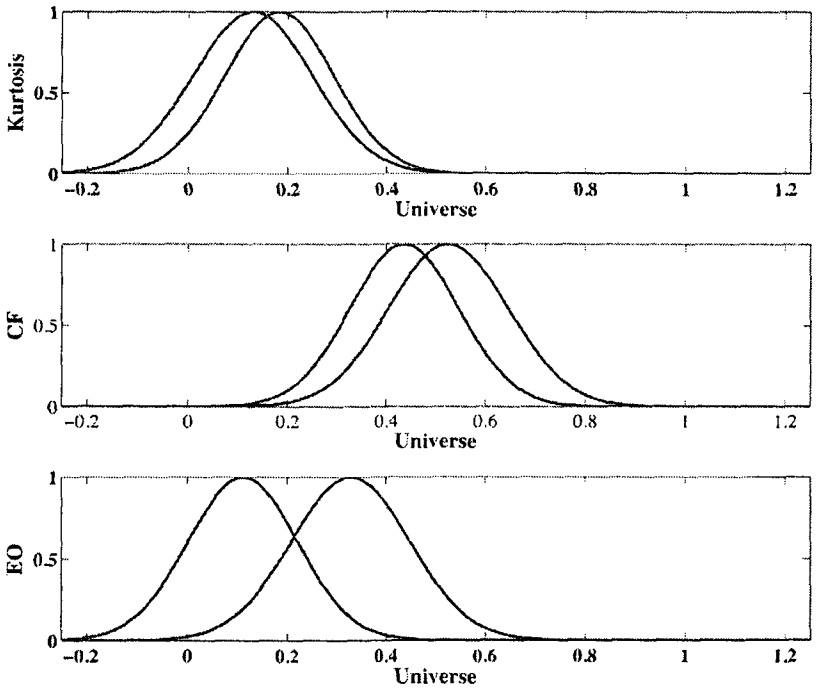

Figure 6.22 - Identified membership functions using ie^{2}TS-sDEKF - Helical gears

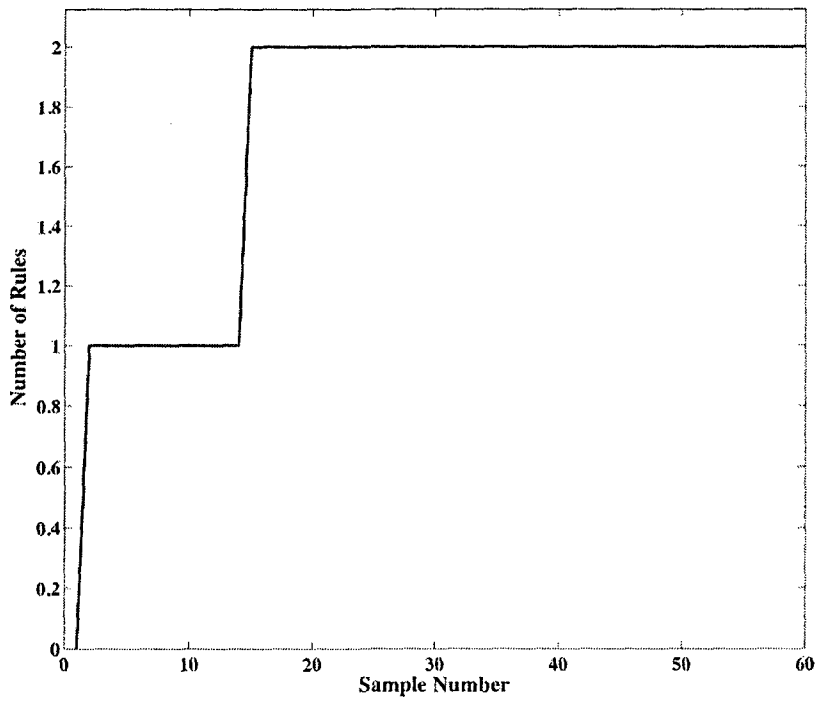

Figure 6.23 - Evolution of rules generation using ie^{2}TS-sDEKF- Helical gears

The proposed ie^{2}TS-sDEKF outperforms other related classifiers except in one identification case when load level is 250mA as depicted in Table 6.4. Figure 6.24 illustrated the identification results for the case when no external loading conditions were applied.

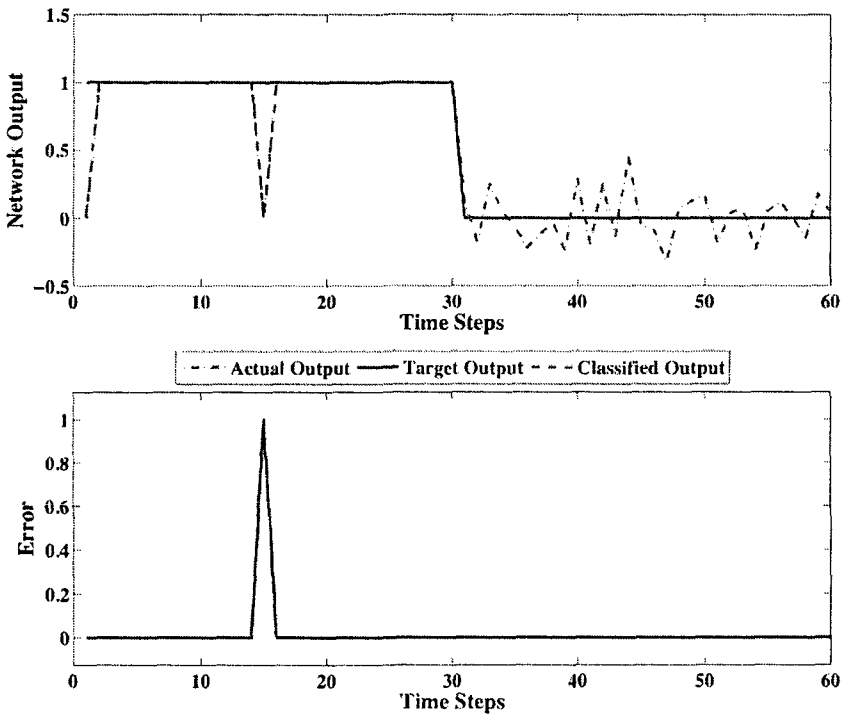

Figure 6.24 - Identified network output and classification error using ie^{2}TS-sDEKF- Helical gear

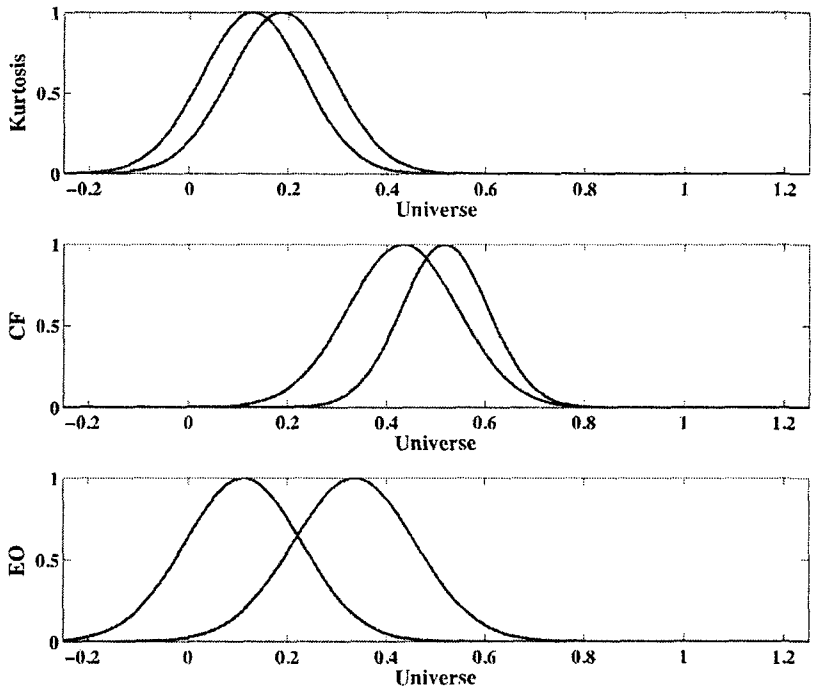

Figure 6.25 - Membership functions after training using ie^{2}TS-sDEKF - Helical gears

The proposed classifier, however, indicates a better performance being able to produce no classification errors at the end of the training process. This effect is demonstrated in Figure 6.26.

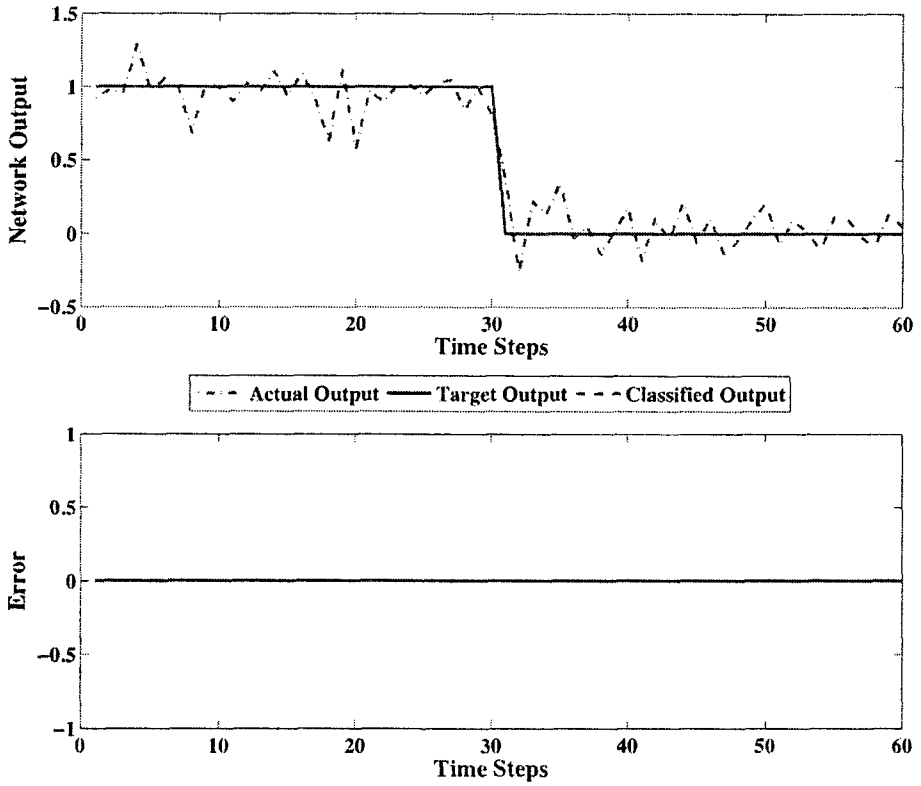

Figure 6.26 - Trained network output using ie^{2}TS-sDEKF - Helical gears

In the verification process, once again, the proposed classifier was able to classify all gear conditions accurately without classification errors. This is an indication of an effective training procedure, which can be demonstrated in Figure 6.27 based on the network output and the classification error.

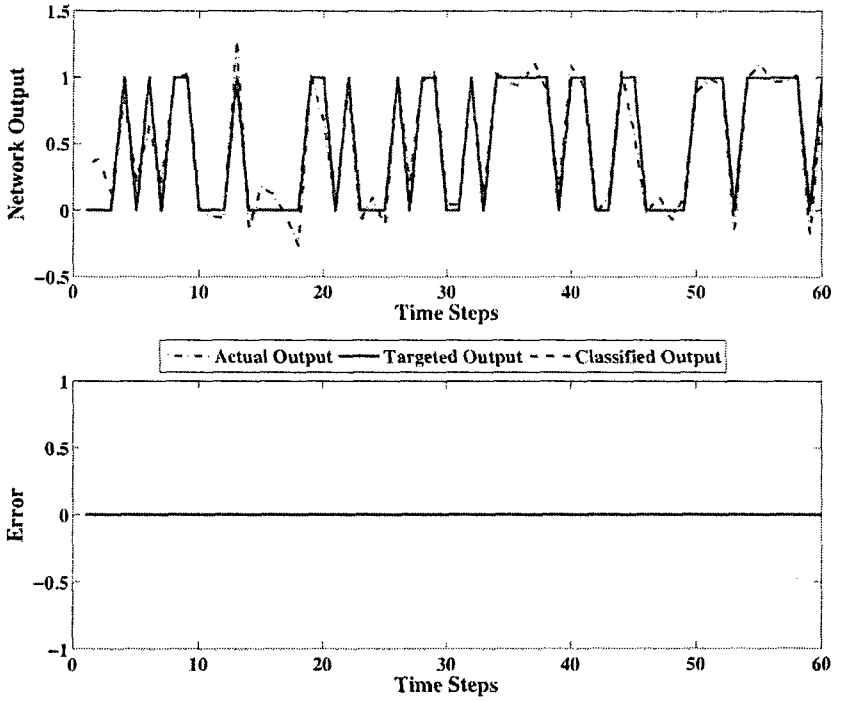

Figure 6.27 - Verification output and classification error using ie^{2}TS-sDEKF - Helical gear

The classification performance of all other related classifiers is summarized in Table 6.4. Some interesting aspects can be noted in Table 6.4 mainly in the identification errors for no loading levels. That is, the number of errors generated by the classifier e^{2}TS-DEKF significantly higher than all the other networks. This is thought to be due to the classifiers' identification technique in which cluster centers and radiuses are being assigned without verifying their potential. Hence, the numbers of errors indicates that the clusters generated are not clusters with the highest potential. Another aspect worth mentioning is pertaining to the loading levels of 100mA. Although all classifiers identified the same number of clusters, the proposed classified had the ability to generate fewer errors in a slightly faster time. This validates the proposed classifier's robustness.

Helical Gears Results								
Load = 250								
Method	**Number of Clusters**	**Epochs**	**Avg. Training Time per Epoch (s)**	**Identification Errors**	**Average Training Errors**	**Average Testing Errors**	**Rate**	**Avg. RMSE**
eTS-DEKF	3	50	0.061558	6	5.9	6	90	0.306
e²TS-DEKF	3	50	0.062941	6	6	6	90	0.2812
ie²TS-DEKF	2	50	0.047287	7	6.04	6	90	0.301
ie²TS-sDEKF	2	50	0.047375	7	6	6	90	0.2746
Load = 100								
Method	**Number of Clusters**	**Epochs**	**Avg. Training Time per Epoch (s)**	**Identification Errors**	**Average Training Errors**	**Average Testing Errors**	**Rate**	**Avg. RMSE**
eTS-DEKF	2	50	0.049132	5	5.96	6	90	0.2999
e²TS-DEKF	2	50	0.048437	6	6.58	7	88.33	0.2999
ie²TS-DEKF	2	50	0.047476	4	6	6	90	0.3001
ie²TS-sDEKF	2	50	0.047618	4	3.4	4	93.33	0.2499
Load = 0								
Method	**Number of Clusters**	**Epochs**	**Avg. Training Time per Epoch (s)**	**Identification Errors**	**Average Training Errors**	**Average Testing Errors**	**Rate**	**Avg. RMSE**
eTS-DEKF	4	50	0.075822	1	0.04	1	98.33	0.1733
e²TS-DEKF	4	50	0.075473	58	0.12	1	98.33	0.17
ie²TS-DEKF	2	50	0.047102	1	0	0	100	0.1756
ie²TS-sDEKF	2	50	0.047102	1	0	0	100	0.1488

Table 6.4 - Helical gears results

Although ie^{2}TS-DEKF classifier was also able to classify most instances correctly, its classification accuracy is still lower than ie^{2}TS-sDEKF, indicating the effective convergence of the proposed sDEKF technique. This can be observed in Figures 6.28-6.30 as well as Table 6.4.

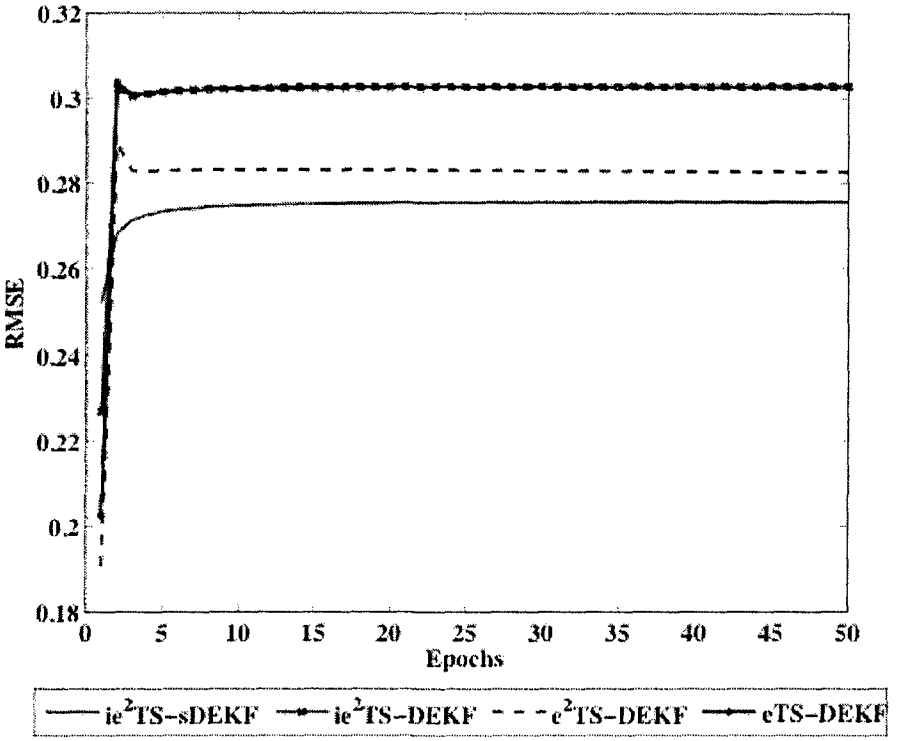

Figure 6.28 - Root mean square error for helical gears - 250mA load level

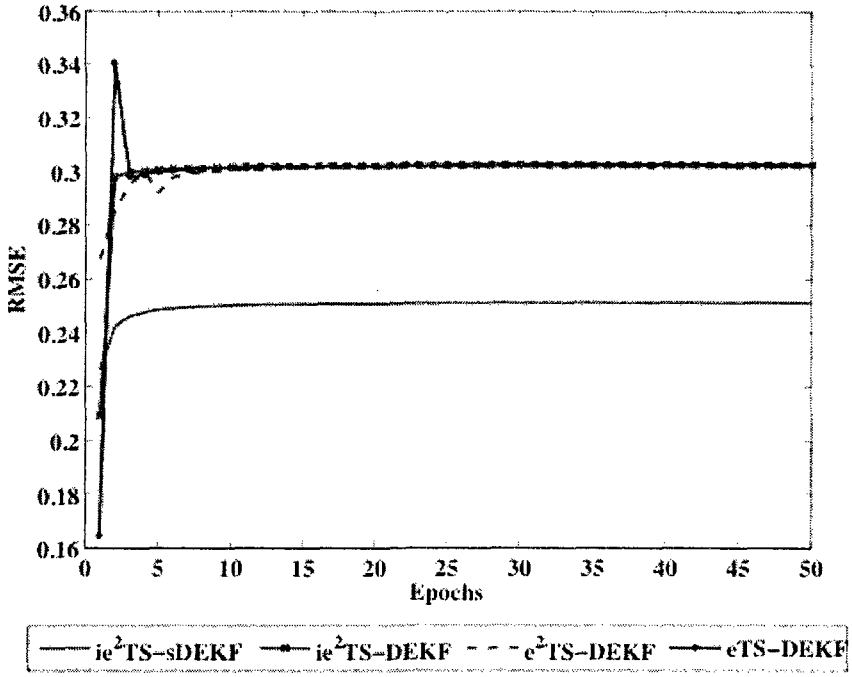

Figure 6.29 - Root mean square error for helical gears - 100mA load

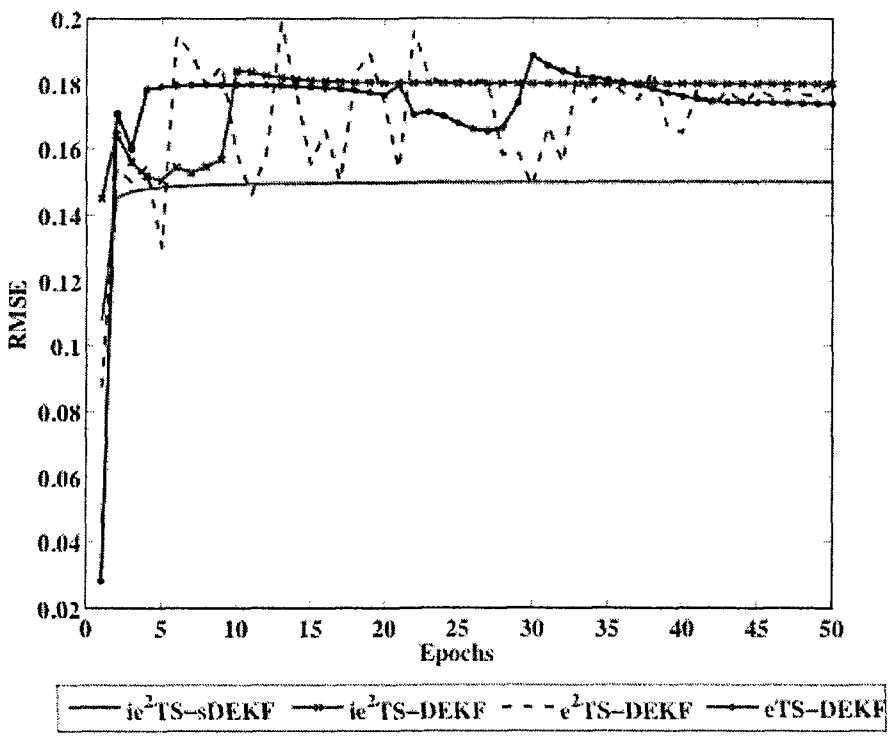

Figure 6.30 - Root mean square error for helical gears - 0mA load

Figure 6.31 displays the identified structure for classifying helical gears.

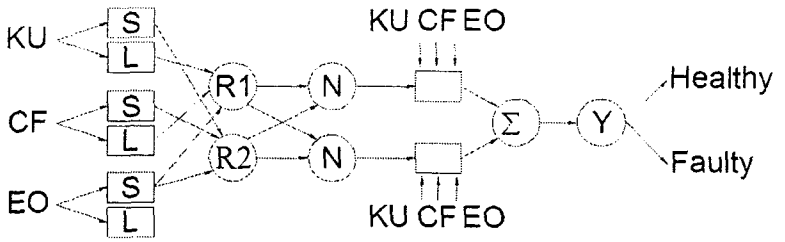

Figure 6.31 - Identified structure for helical gears

Remark D: The EF classifier was able to accurately classify both spur and helical gears despite the CIs inconsistency. This is the result of an efficient training technique.

Chapter 7

7.0 Conclusion and Future Work

"Reasoning draws a conclusion,
but does not make the conclusion certain,
unless the mind discovers it by the path of experience."

ROGER BACON

The aim of this research was to develop an improved intelligent classifier that can be implemented for machinery condition monitoring. Notably, a few direct improvements have been made and the proposed classifier has been shown to have a more flexible clustering initiation and better performing training technique.

Firstly, an improvement was made to the network identification technique. A feasibility check is performed once the input/output data pair is presented. By introducing the feasibility check one can ensure that the incoming data point considered for a cluster center is not overlapping another cluster. Once the input/output data set passes the feasibility check its potential to become a cluster center is calculated. If the data point is deemed a good cluster center candidate, a new point calculated using standard deviation is also proposed as cluster center. Different from other techniques, in this work a new step is introduced. Before accepting the proposed point as cluster center, its potential to be a

cluster center is calculated. The point is accepted as a cluster center if its potential is higher than the potential of the incoming point. Otherwise, the incoming point is chosen as a cluster center. These changes have been shown to make the clustering technique more robust and more flexible, allowing the user to define how far apart the clusters could be.

Second, an improvement has been made to the training technique based on the node decoupled Kalman filter. The motivation of this improvement lies in known inherent attributes of the Kalman filter. The main attribute is the fact that the performance of the Kalman filter is directly dependent on the covariance matrices. For this reason, an updating of the covariance matrices during the training technique has been proposed. It has been found that the performance of the network is improved. This was validated in testing and in training by analyzing the root mean square error produced. The proposed classifier has been compared with other classification methods to validate the findings.

Industries utilizing machinery are in great need of reliable online monitoring systems to detect the occurrence of defects in machines and therefore prevent their performance degradation, malfunction, and even catastrophic failures. As a result, in this research a more reliable condition-monitoring scheme has been used for the classification of spur and helical gears in an experimental system. Preprocessing of signals using signal processing techniques such as time synchronous average, and condition monitoring indicators computed using methods such as kurtosis, root mean square value, crest factor and energy operator has been performed and discussed. Classification in this work mainly focuses on neural fuzzy based classifiers. These classifiers have the ability to adapt the rules keeping the structure fixed as shown in Chapter 6. The proposed classifier had the ability to successfully classify the gears into two categories, namely, healthy or damaged,

based on fifty input/output pairs. Such a classification would not have been possible based on inspection of the time synchronous averaged signals, or further based on the condition indicators computed using the aforementioned methods alone, thus illustrating the effectiveness of utilizing intelligent tools for machinery condition monitoring.

Future works pertaining to the applications consist of the following:

- implementation of the intelligent system for online condition monitoring. In case of machinery performance degradation, the intelligent system can identify the faulty components such that repair personnel can efficiently act upon it.
- computation of condition indices pert tooth rather than per gear as well as detection of type of fault (i.e. pitting, cracking, tooth breaking)
- developing a sensor such as the one presented in [59], in which the classification paradigm could be programmed in the sensor and the decision would be displayed on line. Such a sensor is connected to a laptop by means of a USB cable therefore it would be very useful in industry where space is limited and a large condition monitoring setup would not be advantageous. Another advantage would be the price; production of such a sensor would cost about $500 whereas other condition monitoring setups would cost at least $5000.

In the future, the following endeavors pertaining to intelligent tools will be carried out:

- Implement the developed ie2TS-sDEKF for a wide range of real applications such as bearing classification, shaft misalignment
- Develop online training strategies for classification

- Develop more effective training algorithms to further improve training performance and prevent local minima
    - One way of achieving this would be to change the optimization process to ensure that a convergence to a global minimum is achieved. Optimizations processes such as genetic algorithm and particle swarm optimization are some examples.
- Test the performance of the classifier using data from a different set-up in order to ensure that testing and training data independence.
- Test the performance of the classifier in real applications to ensure its robustability.

References

[1] Wang, W. (2009). Signal Processing course Notes. Lakehead University, Thunder Bay, Canada.

[2] Aslantas, K. & Tasgetiren, S. (2004). A Study of Spur Gears Pitting Formation and Life Prediction. Wear 257, 1167-1175.

[3] Ghafari, S. H. (2007). A Fault Diagnosis System for Rotary Machinery Supported by Rolling Element Bearing. Ph.D. Thesis. University of Waterloo, Waterloo, Canada.

[4] Stewart, R. M. (1977). Some Useful Data Analysis Techniques for Gearbox Diagnostics. Report MHM/R/77, Machine Health Monitoring Group, Institute of Sound and Vibrations Research, University of Southampton.

[5] Zakrajsek, J. J., Townsend, D.P. & Decker, H. J. (1993). An Analysis of Gear Fault Detection Methods as Applied to Pitting Fatigue Failure Data (NASA). NASA TM 105950, 47th Meeting of the Society for Machinery Failure Prevention Technology.

[6] Decker, H. J., Handschuh, R. F. & Zakrajsek, J. J. (1994). An Enhancement to the NA4 Gear Vibration Diagnostic Parameter. 18th Annual Meeting Vibration Institute, Hershey, PA.

[7] Dempsey, P. J., Zakrajsek, J. J. (2001) Minimizing Load Effects on NA4 Gear Vibration Diagnostic Parameter. NASA TM-2001-210671. Glenn Research Center, Clevland, OH.

[8] Junying, L., Yanjie, Q., Haiying, K., & Haiqi, Z. (2007). Application of Order Cepstrum in Gear Wearing. The Eights International Conference on Electronic Measurement and Instruments, 643-646. China.

[9] Fyfe, K. R., & Munck, D. S. (1997). Analysis of Computed Order Tracking. Mechanical Systems and Signal Processing , 187-205.

[10] Ai, S., & Li, H. (2006). Application of Order Cepstrum and Neural Network to Gear Fault Detection. Computational Engineering in Systems Applications, 1822-1827. Beijing.

[11] Wang, W., & Wong, A. K. (1999). Some New Signal Processing Approaches for Gear Fault Diagnosis. Fifth International Symposium on Signal Processing and its Applications, Brisbane, 587-590 .

[12] Liu, Z. (2006). A Novel Time-Frequency Analysis Method for Mechanical Failure Detection. International Conference on Hybrid Information Technology. IEEE Computer Society.

[13] Jang, J. S. R., Sun, C. T. & Mizutani, E. (1997). Neuro-Fuzzy and Soft Computing. Prentice Hall.

[14] Takagi, T. and Sugeno, M. Fuzzy identification of systems and its application to modeling and control. IEEE Trans. on Systems, Man & Cybernetics 15, 116-132, 1985.

[15] Li, H. Wunsch, D. C., O'Hair, E. Giesselmann, M. G. (2002). Extended Kalman Filter Training of Neural Networks on a SIMD Parallel Machine. Jurnal of Parallel and Distributed Computing. Vol. 62, No. 4, 544-562.

[16] Murtuza, S & Chorian, S.F.(1994). Node Decoupled Extended Kalman Filter Based Learning Algorithm for Neural Networks. IEEE International Symposium on Intelligent Control, 364-368. Columbus.

[17] Wei, G., Wang, X., & Sun, J. (2009). Extended Kalman Filter Training T-S Fuzzy Model for Signal Reconstruction of Multifunctional Sensor. International Instrumentation and Measurement Technology Conference.

[18] Ding, W., Wang, J. & Rizos, C. (2007). Improving covariance based adaptive estimation for GPS/INS integration. The Journal of Navigation, Vol. 60, No. 3, 517-529.

[19] Jackson, Q. & Landgrebe, D. (2002). An Adaptive Method for Combined Covariance Estimation and Classification. IEEE Transactions on Geoscience and Remote Sensing, Vol. 40, No. 5, 1082-1087.

[20] Yang, Y. & Gao, W. (2006). An Optimal Adaptive Kalman Filter. Journal of Geodesy, Vol. 80, No. 4, 177-183.

[21] Yang, Y. & Gao, W. (2005). Comparison of Adaptive Factors in Kalman Filters on

Navigation Results. The Journal of Navigation. Vol. 58, 471-478.

[22] Hu, C., Chen, W., Chen, Y. & Liu, D. (2003). Adaptive Kalman Filtering for Vehicle Navigation. Journal of Global Positioning Systems, Vol. 2 No. 1, 42-47.

[23] MacQueen, J. B. (1967). Some Methods for classification and Analysis of Multivariate Observations. Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press, 1:281-29

[24] Dunn, J. C. (1973). A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters, Journal of Cybernetics Vol. 3, 32-57.

[25] Bezdek, J. C. (1981). Pattern Recognition with Fuzzy Objective Function Algoritms, Plenum Press, New York.

[26] Angelov, P. and Buswell, R. (2001). Evolving rule-based models: a tool for intelligent adaptation. 9th IFSA World Congress, 25-28. Vancouver, BC, Canada.

[27] Angelov, P. & Buswell, R. (2002). Identification of Evolving Fuzzy Rule-Based Models. IEEE Transactions on Fuzzy Systems, Vol. 10, No. 5, 667-677.

[28] Angelov, P., Zhou, X., & Klawonn, F. (2007). Evolving Fuzzy Rule-based Classifiers. IEEE Symposium on CIISP, 220-224.

[29] Angelov, P. & Filev, D. (2004). An Approach to Online Identification of Takagi-Sugeno Fuzzy Models. IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics, Vol. 34, No. 1, 484-498.

[30] Angelov, P. & Zhou, X. (2008). Evolving Fuzzy-Rule-Based Classifiers From Data Streams. IEEE Transactions on Fuzzy Systems, Vol. 16, No. 6, 1462-1474.

[31] Song, Q., & Kasabov, N. (2001). ECM — A novel on-line, evolving clustering

method and its applications. In Proceedings of the fifth biannual conference on artificial neural networks and expert systems (pp. 87–92).

[32] Song, Q., & Kasabov, N. (2006). TWNFI - A Transductive Neuro-Fuzzy Inference System with Weighted Data Normalization for Personalized Modeling. Neural Networks, Vol. 19, 1591-1596.

[33] Song, Q. and Kasabov, N. (2005). NFI - Neuro-Fuzzy Inference Method for Transductive Reasoning and Applications for Prognostic Systems. IEEE Trans. Fuzzy Systems. Vol.13, No.6. 799-808.

[34] Kasabov, N. and Song, Q. (2002). DENFIS: Dynamic, Evolving Neural-Fuzzy Inference Systems and Its Application for Time-Series Prediction, IEEE Transactions on Fuzzy Systems Vol. 10, 144–154.

[35] Wang, W. & Vrbanek, J. (2008). An Evolving Fuzzy Predictor for Industrial Applications. IEEE Transactions on Fuzzy Systems, Vol. 16, No.6, 1439-1449.

[36] Wang W., Ismail F. and F. Golnaraghi. (2004). A neuro-fuzzy approach for gear system monitoring. IEEE Transactions on Fuzzy Systems, 12: 710-723.

[37] Wang, W. (2008). An enhanced diagnostic system for gear system monitoring, IEEE Transactions on Systems, Man, Cybernetics, Part B, Vol. 34, pp. 102-112.

[38] Wang, W. (2008). An intelligent system for machinery condition monitoring, IEEE Transactions on Fuzzy Systems, Vol. 16, No. 1, pp. 110-122.

[39] Wang, W. (2007). An adaptive predictor for dynamic system forecasting. Mechanical Systems and Signal Processing, 21: 809-823.

[40] Liu, J. (2008). An Intelligent System For Bearing Condition Monitoring. Ph.D. Thesis, Waterloo University, Waterloo, Canada.

[41] Zadeh, L.A. (1965). Fuzzy sets. Information and Control Vol. 8, No.3, 338-353.

[42] Vrbanek, J. (2008). Advanced Intelligent Tools for System Classification and Prediction. M.Sc. Thesis, Lakehead University, Thunder Bay, Canada.

[43] Angelov P and Filev D (2002) Flexible Models with Evolving Structure , In Proc. IEEE Symposium on Intelligent Systems, Varna, Bulgaria, v.2, pp.28-33.

[44] Kasabov, N. & Filev. D. (2006). Evolving Intelligent Systems: Methods, Learning and Applications. Proc. 2nd Int. Symp. Evolving Fuzzy Systems, London, U.K., IEEE Press, 8-18.

[45] Wang, W. & Vrbanek, J. (2008). An Evolving Fuzzy Predictor for Industrial Applications. IEEE Transactions on Fuzzy Systems, Vol. 16, No.6, 1439-1449.

[46] Frank, A. & Asuncion, A. (2010). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[47] Street, W.H., Wolberg, W.H. & Mangasarian, O.L. (1993). Nuclear feature extraction for breast tumor diagnosis. IS&T/SPIE 1993 International Symposium on Electronic Imaging: Science and Technology, Vol. 1905, 861-870, San Jose, CA.

[48] Forina, M. et al. (1991). An Extendible Package for Data Exploration, Classification and Correlation. Institute of Pharmaceutical and Food Analysis and Technologies, Via Brigata Salerno, 16147 Genoa, Italy.

[49] Chiu, S. L. (1997). Extracting Fuzzy Rules from Data for Function Approximation and Pattern Classification. Chapter 9 in Fuzzy Information Engineering: A Guided Tour of Applications, ed. D. Dubois, H. Prade, and R. Yager, John Wiley & Sons.

[50] User Operating Manual for Machinery Fault/Gearbox Dynamics Simulator. (2006). Spectra Quest, Inc., 67.

[51] Vecer, P. Kreidl, M., Smid, R. (2005). Condition Indicators for Gearbox Condition Monitoring Systems. Acta Polytechnica Vol. 45, No. 6.

[52] Wang, W. Q., Ismail, F., & Golnaraghi, F. (2001). Assesment of Gear Damage Monitoring Techniques Using Vibration Measurements. Mechanical Sistems and Signal Processing , 905-922.

[53] McFadden, P. D. (2000). Detection of Gear Faults by Decomposition of Matched Differences of Vibration Signals. Mechanical Systems and Signal Processing , 805-817.

[54] Vachtsevanos, G., Lewis, F. L. (2006). Intelligent Fault Diagnosis and Fault Prognosis for Engineering Systems. John Wiley & Sons, Inc. 418-419.

[55] Combet, L., Gelman, L. (2007). An Automated Methodology for Permorming Time Synchronous Averaging of a Gearbox Signal Without Speed Sensor. Mechanical Systems and Signal Processing, Vol. 21, No. 6, 2590-2606.

[56] Bechhoefer, E., Kingsley, M. (2009). A Review of Time Synchronous Average Algorithms. Annual Conference of the Prognostics and Health Management Society.

[57] Dron, J.P., Bolaers, F, Rasolofondraibe, L. (2004). Improvement of the sensitivity of the scalar indicators (crest factor, kurtosis) using a de-noising method by spectral subtraction: application to the detection of defects in ball bearings. Journal of Sound and Vibration, 270 (1-2), 61-73

[58] Baydar, N., Chan, Q., Ball, A., Kruger, U. (2001). Detection of Incipient Tooth Defect in Helical Gears Using Multivariate Statistics. Mechanical Systems and Signal Processing. Vol. 15, No. 2, 303-321.

[59] Wang, W. & Jianu, O. (2010). "A smart monitor for machinery fault detection," IEEE/ASME Transactions on Mechatronics, Vol. 15, No. 1, 70-78.