FAULT DETECTION AND DIAGNOSIS IN NONLINEAR DYNAMICAL SYSTEMS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ERDAL KILIÇ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

AUGUST 2005

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Canan Özgen
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. İsmet Erkmen
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. Kemal Leblebicioğlu
Supervisor

Examining Committee Members

Prof. Dr. Mübeccel Demirekler (METU, EE)

Prof. Dr. Kemal Leblebicioğlu (METU, EE)

Prof. Dr. Uğur Halıcı (METU, EE)

Prof. Dr. Turhan Çiftçibaşı (Başkent Unv., EE)

Asst. Prof. Dr. Halit Oğuztüzün (METU-CENG)

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Erdal KILIÇ

ABSTRACT

FAULT DETECTION AND DIAGNOSIS IN NONLINEAR DYNAMICAL SYSTEMS

Kılıç, Erdal

Ph.D., Department of Electrical and Electronics Engineering

Supervisor: Prof. Dr. Kemal Leblebicioğlu

August 2005, 171 Pages

The aim of this study is to solve Fault Detection and Diagnosis (FDD) problems occurring in nonlinear dynamical systems by using model and knowledge-based FDD methods and to give a priority and a degree about faults. For this purpose, three model-based FDD approaches, called FDD by utilizing principal component analysis (PCA), system identification based FDD and inverse model based FDD are introduced. Performances of these approaches are tested on different nonlinear dynamical systems starting from simple to more complex. New fuzzy discrete event system (FDES) and fuzzy discrete event dynamical system (FDEDS) concepts are introduced and their applicability to an FDD problem is investigated. Two knowledge-based FDD methods based on FDES and FDEDS structures using a fuzzy rule-base are introduced and they are tested on nonlinear dynamical systems. New properties related to FDES and FDEDS such as fuzzy observability and diagnosibility concepts and a relation between them are illustrated. A dynamical rule-base extraction method with classification techniques and a dynamical and a static diagnoser design methods are also introduced. A nonlinear and event based extension of the Luenberger observer and its application as a diagnoser to isolate faults are illustrated. Finally, comparisons between the proposed model and knowledge-based FDD methods are made.

Keywords: Fault, diagnosis, fuzzy discrete event systems, fuzzy observability, fuzzy diagnosibility.

ÖZ

DOĞRUSAL OLMAYAN DİNAMİK SİSTEMLERDE HATA TESBİT VE TANILAMA

Kılıç, Erdal

Doktora, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Kemal Leblebicioğlu

Ağustos 2005, 171 Sayfa

Bu çalışmadaki amaç, doğrusal olmayan dinamik sistemlerdeki hata tesbit ve tanılama problemlerini model ve bilgi tabanlı yöntemler kullanarak çözmek ve hatalarla ilgili öncelik ve miktar vermektir. Bunun için temel elemanlar analizi, model belirleme ve ters modele dayanan üç farklı matematiksel model kullanan hata tanılama ve algılama yöntemleri tanıtılmıştır. Bu yöntemlerin performansı farklı, doğrusal olmayan, basit ve karmaşık sistemler üzerinde denenmiştir. Yeni bir bulanık ayrık olaylı sistem yapısı ve dinamik bulanık ayrık olaylı sistem yapısı geliştirilmiştir. Daha sonra bu geliştirilen yapıların hata tanılama problemine uyarlanabilirliği araştırılmıştır. Geliştirilen bu yapılara dayanan uzman kurallar içeren iki farklı bilgi tabanlı hata algılama ve tanılama yöntemi tanıtılmış ve bunların performansı doğrusal olmayan sistemler üzerinde denenmiştir. Bu yeni yapılarla ilgili olarak gözlenebilme ve tanılanabilme kavramları ve bunlar arasındaki ilişki ortaya konmuştur. Sınıflama yöntemleri kullanılarak dinamik kural tabanının nasıl oluşturulacağı gösterilmiştir. Ayrıca dinamik ve statik tanılayıcı tasarımı verilmiş, Luenberger tipi gözlemleyicinin doğrusal olmayan uzantısı ile olay tabanlı uzantısı tanıtılmıştır. Son olarak bu çalışmada yer alan model ve bilgi tabanlı hata algılama ve tanılama sistemlerinin başarım karşılaştırması yapılmıştır.

Anahtar Kelimeler: Hata, tanılama, bulanık ayrık olaylı sistemler, bulanık gözlenebilme, bulanık tanılanabilme.

AKNOWLEDGMENTS

I would like to express my appreciation to my supervisor Prof. Dr. Kemal Leblebicioğlu for his support and advices during my research.

I have worked with the DCSC group in the Delft Technical University in the Netherlands as a visiting researcher for nine months. I would like to thank to Prof. Dr. Robert Babuska and Dr. Vincent Verdult and other members of the group for their collaboration and hospitality.

I would like to acknowledge TUBİTAK-BAYG and ODTÜ-ÖYP for the scholarship that covered my expenses in the Netherlands.

Finally, I want to thank my family and my son for their consistent support.

TABLE OF CONTENTS

LIST OF TABLES

TABLE

LIST OF FIGURES

FIGURE

CHAPTER 1

INTRODUCTION

1.1 Importance of Fault Detection and Diagnosis (FDD)

Today, nonlinear, complex systems are everywhere in our life. They are in computers, aircrafts, ships and cars. Complex systems are seen in every industry such as chemical reactors, nuclear power reactors and distillation columns. They are constantly working, making our life more comfortable and more pleasant till these systems fail.

Faults in complex systems are events that happen rarely at unexpected moments of time. Fault is defined by Isermann and Balle [21] as:

> **Fault** is an unpermitted deviation of at least one characteristic property or parameter of the system from the usual or standard condition.

It is difficult to predict and prevent faults in dynamical systems. Faults may lead to economical and human loses via incidents in safety–critical systems. Several examples are: the explosion at the nuclear power plant at Chernobyl, Ukraine, on 26 April 1986. About 30 people were killed immediately, while another 15,000 were killed and 50,000 left handicapped in the emergency clean-up after the accident and the explosion of the Ariane 5 rocket on 4 June 1996, where the reason was a fault in the internal reference unit that has the task to provide the control system with altitude and trajectory information.

The question is "Could something have been done to prevent these disasters?" While in most situations the occurrences of faults in the complex systems cannot be prevented, the consequences of the faults could be avoided, or at least their severity could be minimized. In order to minimize the possibility of occurrences of catastrophic events, the most important step is the utilization of the means of FDD methods. They are designed to increase nonlinear, complex systems' reliability and safety in theory [1], [2], [3], [21], [22], [23] and in practice [66]. All the examples above including those in the references show the need for FDD systems in order to improve the reliability and safety in nonlinear, complex systems.

1.2 Nomenclature

The terminology in this field is not consistent. This makes it difficult to understand the goals of contributions and to compare different approaches. For example, what the differences between fault or failure detection, isolation, identification and diagnosis are is not very clear. Hence, the SAFEPROCESS Technical Committee discussed this matter and tried to find commonly accepted definitions [21].

Fault: An unpermitted deviation of at least one characteristic property or parameter of the system from the acceptable / usual / standard condition.

Failure: A permanent interruption of a system's ability to perform a required function under specified operating conditions.

Malfunction: An intermittent irregularity in the fulfillment of a system's desired function.

Error: A deviation between a measured or computed value (of an output variable) and the true, specified or theoretically correct value.

Disturbance: An unknown and uncontrolled input acting on a system.

Residual: A fault indicator, based on deviation between measurements and model-equation-based computations.

Symptom: A change of an observable quantity from normal behavior.

Fault detection: Determination of faults present in a system and the time of detection.

Fault isolation: Determination of the kind, location and time of detection of a fault. Follows fault detection.

Fault identification: Determination of the size and time-variant behavior of a fault. Follows fault isolation.

Monitoring: A continuous real time task of determining the conditions of a physical system, by recording information, recognizing and indicating anomalies in the behavior.

Supervision: Monitoring a physical system and taking appropriate action to maintain the operation in the case of faults.

Reliability: Ability of a system to perform a required function under stated conditions, with in a given scope, during a given period of time.

Safety: Ability of a system not to cause danger to persons or equipment or the environment.

Availability: Probability that a system or equipment will operate satisfactorily and effectively at any point of time

Quantative model: Use of static and dynamic relations among system variables and parameters in order to describe a system's behavior in quantative mathematical terms.

Qualitative model: Use of static and dynamic relations among system variables and parameters in order to describe a system's behavior in qualitative terms such as causalities and if-then rules.

Diagnostic model: A set of static or dynamic relations which link specific input variables-the symptoms-to specific output variables-the faults.

Analytical redundancy: Use of two or more (but not necessary identical) ways to determine a variable, where one way uses a mathematical process model in analytical form.

1.3 Fault Classification

Faults are events that can take place in different parts of a nonlinear complex system. In literature, faults are classified as actuator faults, sensor faults and component faults [102].

Actuator faults represent partial or complete loss of control action. Total actuator fault can occur, for instance, as a result of a breakage, cut or burned wiring, shortcuts, or the presence of outer body in the actuator. Despite of the input applied to an actuator, it produces no actuation. This is an example of a completely lost actuator (**stuck** actuator). Partially failed actuator produces only a part of the normal (i.e., under nominal operating condition) actuation. It can result from, e.g., hydraulic or pneumatic leakage, increased resistance or fall in the supply voltage**.**

Sensor faults represent incorrect reading from the sensors. They also are subdivided into partial and total. Produced information is not related to value of the measured physical parameter in case of the total actuator fault. They can be due to broken wires, lost contact with the surface, etc. The output containing useful information could still be retrieved. This can, for instance, be a gain reduction, a biased measurement or increased noise.

Component faults are faults in the components of a complex system, i.e., all faults that cannot be categorized as sensor or actuator faults. These faults represent changing in the damping constant, etc., that are often due to structural damages. They often result in a change in the dynamical behavior of a nonlinear complex system. They are the most frequently encountered types in fault family to deal with.

Further, faults are classified as *additive* and *multiplicative*. Additive faults are suitable to represent component faults in the system, while sensor and actuator faults are in practice most often multiplicative by nature. Faults are also classified according to their time characteristic as *abrupt, incipient* and *intermittent* as shown in figure 1.1.

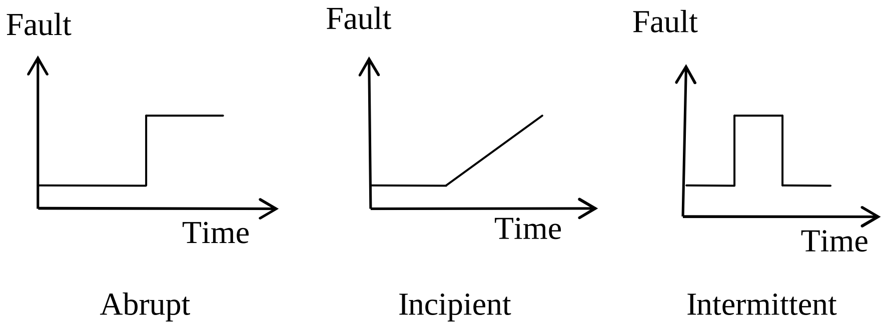

Figure 1.1: Fault time characteristics

Abrupt faults occur instantaneously often as a result of hardware damage. Usually they are very severe as they affect the performance or the stability of the system. Incipient faults represent slow in time parametric changes, often as a result of aging. They are more difficult to detect. Finally, intermittent faults are faults that appear and disappear repeatedly; for instance, due to a partially damaged wiring.

1.4 Modeling Faults

In this section we concentrate on the mathematical representation of fault types. Although we are interested in FDD problems in complex, nonlinear dynamical systems, for the sake of simplicity, the linear time invariant dynamical system representation in state space form is used to discuss the effect of the faults on the system's dynamics. The dynamics of the system can be described by the following discrete-time, time-invariant, linear dynamical system representation in state-space form [102] as:

$$\begin{aligned} x(k+1) &= A.x(k) + B.u(k) \\ y(k) &= C.x(k) + D.u(k) \end{aligned} \tag{1-1}$$

where $u \in R^m$, $y \in R^l$, $x \in R^n$ denote system inputs, outputs and the state of the system, respectively.

1.4.1 Multiplicative Faults

Multiplicative fault modeling is mostly used to represent sensor and actuator faults [102]. Actuator faults show the malfunction of the actuators and can be modeled as an abrupt change of the system input u_k to

$$u_k^f = u_k + (I - \Sigma_\circ)(\bar{u} - u_k) \tag{1-2}$$

where $\bar{u} \in R^m$ is a (not necessarily constant) vector that cannot be manipulated and where

$$\Sigma_\circ = diag\{[\sigma_1^\circ, \quad \sigma_2^\circ, \quad ..., \quad \sigma_m^\circ]\}, \sigma_i^\circ \in R. \tag{1-3}$$

In this way $\sigma_i^\circ = 0$ represents a total fault (complete failure) of *i*th actuator of the system so that the control action coming from this *i*th actuator becomes equal to *i*th element of the uncontrollable offset vector $\bar{u}$, i.e., $u_k^f(i) = u(i)$. On the other hand, $\sigma_i^\circ = 1$ implies that the *i*th actuator operates normally $(u_k^f(i) = \bar{u}(i))$. The quantities, σ_i° , *i* = 1, 2*, ... , m* can also take values in between 0 and 1. Possible partial actuator faults can be represented in this way. Substituting the input u_k in equation (1-1) with the faulty u_k^f results in the following state-space model [102]

$$\begin{aligned} x(k+1) &= A\,x(k) + B\Sigma_\circ\, u(k) + B(I - \Sigma_\circ)\bar{u} \\ y(k) &= C\,x(k) + D\Sigma_\circ\, u(k) + D(I - \Sigma_\circ)\bar{u}. \end{aligned} \tag{1-4}$$

Models in the form (1-4) are referred to as multiplicative fault models and have been used in the literature [67]. It needs to be noted that multiplicative actuator faults affect the dynamics of the closed-loop system, and may even affect the controllability of the system. Similarly, sensor faults occurring in the system (1-1) represent incorrect reading from the sensors. As a result, the real output of the system differs from the variable being measured. Multiplicative sensor faults can be modeled [102] as:

$$\begin{aligned} y_k^f &= y_k + (I - \Sigma_S)(\bar{y} - y_k), \\ &= (I - \Sigma_S) y_k + (I - \Sigma_S)\bar{y} \end{aligned} \tag{1-5}$$

where $\bar{y} \in R^l$ is an offset vector, and

$$\Sigma_S = diag\{[\sigma_1^s, \quad \sigma_2^s, \quad ..., \quad \sigma_m^s]\}, \sigma_j^s \in R. \tag{1-6}$$

so that $\sigma_j^s = 0$ represents a total fault (complete failure) of *j*-th sensor, and $\sigma_j^s = 1$ models the normal mode of operation of the *j*-th sensor. Partial sensor faults are then modeled by taking $\sigma_j^s \in (0, 1)$. Substituting the nominal measurement y_k in equation (1-1) with its faulty counterpart y_k^f results in the following state-space model [102] that represents multiplicative sensor faults

$$\begin{aligned} x(k+1) &= A\,x(k) + Bu(k) \\ y(k) &= \Sigma_S\, C\, x(k) + \Sigma_S\, Du(k) + (I - \Sigma_S)\bar{y}. \end{aligned} \tag{1-7}$$

In this way, combinations of multiplicative sensor and actuator faults are represented by

$$\begin{aligned} x(k+1) &= A\,x(k) + B\,\Sigma_\circ\, u(k) + b(\Sigma_\circ, \bar{u}) \\ y(k) &= \Sigma_S\, C\, x(k) + \Sigma_S\, D\,\Sigma_\circ\, u(k) + d(\Sigma_\circ,\ \Sigma_S,\ \bar{u},\ \bar{y}), \end{aligned} \tag{1-8}$$

with

$$\begin{aligned} b(\Sigma_\circ, \bar{u}) &= B(I - \Sigma_\circ)\bar{u}, \\ d(\Sigma_\circ,\ \Sigma_S,\ \bar{u},\ \bar{y}) &= \Sigma_S\, D(I - \Sigma_\circ)\bar{u} + (I - \Sigma_S)\bar{y}. \end{aligned}$$

The multiplicative model is a natural way to model a wide variety of sensor and actuator faults, but cannot be used to represent more general component faults. To use these models, the state-space matrices of the faulty system are needed. In nonlinear case, the linearized model around equilibrium points can be used to construct the system's state-space model.

1.4.2 Additive Faults

The additive fault representation is more general than the multiplicative one and can be used to model a more general class of faults. It is more suitable for the design of FDD schemes because the faults are represented by one signal rather than by changes in the state-space matrices of the system. For that reason the majority of FDD methods are focused on additive faults [68], [102]. A state-space model with additive faults has the following form

$$\begin{aligned} x(k+1) &= A.x(k) + B.u(k) + Ff_k \\ y(k) &= C.x(k) + D.u(k) + Ef_k, \end{aligned} \tag{1-9}$$

where $f_k \in R^{n_f}$ is a signal describing the faults. This representation may, in principle, be used to model a wide class of faults, including sensor, actuator and component faults. Using model (1-9), however, often results f_k becoming related to one or more of the signals u_k, y_k and x_k. For example, if one would use this additive fault representation to model a total fault in all actuators (set $\sum_\circ = 0$ and $\bar{u} = 0$ in equation (1-2)), then in order to make model (1-9) equivalent to model (1-4) one needs to take a signal f_k such that $\begin{bmatrix} F \\ E \end{bmatrix} f_k = -\begin{bmatrix} B \\ D \end{bmatrix} u_k$ holds, making f_k dependent on u_k. It is clearly concluded that the fault signal depends on u_k. However, this conclusion is not valid for the multiplicative case. When used to represent sensor and actuator fault in terms of input-output relations, these two faults become difficult to distinguish [102]. Indeed, suppose that the model

$$\begin{aligned} x(k+1) &= A.x(k) + B.u(k) + f_k^a \\ y(k) &= C.x(k) + D.u(k) + f_k^s, \end{aligned}$$

is used to represent faults in the sensors and actuators. By writing the corresponding transfer function,

$$Y(z) = (C(zI - A)^{-1}B + D)u_k + C(zI - A)^{-1} f_k^a + f_k^s,$$

It becomes indeed clear, that the effect of an actuator fault on the output of the system can be modeled not only by the signal f_k^a, but also by f_k^s.

1.4.3 Component Faults

The component faults may bring changes in practically any element of the complex system. These kinds of faults cannot be classified as sensor or actuator faults. Component faults lead to changes in each matrix of the state-space representation of the system due to physical parameter that undergoes a change. In literature, these faults are often modeled [102] as

$$\begin{aligned} x(k+1) &= A(f).x(k) + B(f).u(k) \\ y(k) &= C(f).x(k) + D(f).u(k), \end{aligned} \tag{1-10}$$

with

$$A(f) = A + \Delta A,\ B(f) = B + \Delta B,\ C(f) = C + \Delta C,\ D(f) = D + \Delta D$$

where $f \in R^{nf}$ is a parameter vector representing the component faults and Δ denotes the changes in the state matrices due to the parameter faults. This model might also be used for modeling sensor and actuator faults. Since the matrices may depend in a general nonlinear way on the fault signal f_k this model is less suitable for FDD. Hence, in this thesis, component faults are considered as additive faults given by the model (1-9).

1.5 Fault Detection and Diagnosis

A successful detection of a fault begins with obtaining residuals or symptoms having the maximal sensitivity to its occurrence. Such a stage is followed by the fault diagnosis procedure, which allows distinguishing a particular fault from others. Faults are distinguishable or isolatable using a residual or symptom set if each residual or symptom is sensitive to a subset of faults. In literature, one can find many fault detection and diagnosis methods [23], but these methods are based on systems' mathematical models and knowledge obtained from the systems.

1.5.1 Model-Based Fault detection

There exist a wide variety of model based fault detection methods. Basic model-based fault detection methods use system input and output measurements, i.e., dynamic observers (i.e., dedicated observers, fault detection filters and output observers), parity equations and identification and parameter estimation techniques. They generate residuals for output variables with parametric and non-parametric models. If only the system output signal can be measured, signal model based methods such as band-pass filters, spectral analysis and maximum-entropy estimation techniques [2], [21], [97], [98], [99] can be used for FDD purpose. The characteristic features of fault detection methods show stochastic behavior with mean values and variances. Deviations from the normal behavior have then to be detected by methods of change detection like mean and variance estimation, likelihood ratio test, Bayes decision, run-sum test and two-probe t-test [2], [21]. It can be stated that parameter estimation and observer based methods are the most frequently used techniques for fault detection.

1.5.2 Knowledge-Based Fault Detection

Knowledge-based fault detection is achieved by analytical and heuristic symptom generation. The features from system characteristic values (variances, amplitude, frequency, model parameters, state variables, transformed residuals, special noise, color, smell, vibration) are extracted, while system working normally and under faulty conditions by using analytic and heuristic knowledge. Then the features of the faulty system are compared with the normal features of the non-faulty process and methods of change detection are applied. Fuzzy logic based, artificial neural network and neuro-fuzzy approaches can be considered as knowledge-based methods [94], [95], [96], [100], and [101].

1.5.3 Fault Diagnosis Methods

Determination of the kind, size, location and time of detection of a fault is called fault diagnosis including fault isolation and identification. A way of determining faults from symptoms is to use pattern based approaches as classification methods. Some classification methods are K-means clustering, fuzzy clustering, artificial neural network and geometrical distance and probabilistic methods [23]. When more information about the relations between symptoms and faults is available methods of reasoning can be applied. In such kind of methods, the diagnostic model can first be constructed by using symptom-fault causalities expressed with IF-THEN rules, e.g., in the form of symptom fault tree. Then, analytic as well as heuristic symptoms are evaluated [23]. By forward and backward reasoning, probabilities of faults are obtained as a result of diagnosis. Typical approximate reasoning methods are probabilistic reasoning, possibilistic reasoning with fuzzy logic and reasoning with artificial neural networks. Neural networks are widely used for classification purposes.

1.6 Overview of Previous and Related Works

The developments in fault detection and diagnosis began in the early 1970s and different approaches have been proposed in the last 30 years [1], [2], [3], [4], [21], and [53]. In the start, the research was mainly concentrated on the area of aeronautics and aviation. Different research groups proposed FDD approaches based on their own field. The high diversity of solutions has increased by the growing interest from industry in FDD. This was mainly due to the hope of improving efficiency, safety and reliability. Most methods are covered by the term model-based FDD. The idea is to use analytical redundancy given by a model of a system. The methods use a system model and observables of the systems (control and measurement signals) to generate residuals. Residuals are measures for the discrepancy between expected and measured system behavior. Their analysis led to model based FDD. Model-based FDD methods are categorized as process-model-based FDD and signal-model-based FDD. The process-model-based FDD using system input-output measurements are constructed by observers (i.e., dedicated observers, Kalman filters, bank of observers, output observers and fault-sensitive filters), parity equations, identification and parameter estimation techniques (i.e., equation error methods, output error methods) [2], [21]. These methods use parametric and non-parametric models. If the output signal is measured only, signal-model-based-methods can be applied. These methods are constructed by band-pass filters, spectral analysis and maximum-entropy estimation techniques [2], [21]. The quantities obtained from fault detection methods show stochastic behavior with mean values and variances. Deviations from the normal behavior are detected by the methods like mean and variance estimation, likelihood ratio test, Bayes decision, run-sum test and two-probe t-test [2], [21]. It seems that analytical redundancy-based methods have their best application areas in mechanical systems where the models of the process are relatively precise. Linear or multi linear models of the system are used in most of the model-based FDD. Principally, these approaches count on linearized models of the system and try to design an FDD scheme. Some other solutions can be proposed for particular nonlinearities, but this reduces the number of potential applications [64]. The trend of using linearized models is decreasing. The favorite linear process under investigation is the DC motor. In real life, the methods using linearized models may undergo drawbacks because of the complex nonlinearities occurring in a complex nonlinear dynamical system. Most FDD design methods lead to false alarms because they rely on linear methods. Therefore, demand for a reliable FDD is increasing. Naturally, the use of nonlinear models may ease some of these problems by allowing greater model accuracy. An alternative to nonlinear modeling is system identification. This is the name given to a collection of methods based on developing dynamical models by using observed input-output data. In recent years, several FDD approaches have been improved to handle nonlinear systems, e.g., observer-based approaches, the parity space approach, and parameter estimation approaches. Also fuzzy observers and artificial neural networks were considered for nonlinear systems [46], [50], [58], and [63]. There have been other works incorporated with neural networks [44], [45], [51]. Although neural network or other adaptive function based methods have received a lot of intention, these approaches include some problems such as training, fault isolation capability, optimality of the constructed neural network and parameter error convergence. Observer design for fault detection has often been derived from classical state space theory [43]. However, the detailed knowledge of state space is not easily applicable in this case. The existing non-linear observer approaches are mostly limited to a small class of non-linear systems having correct prediction problems [43], [47]. There are some other methods based on systems mathematical models, in which the processes dynamics are used [44], [48], [52] and [49]. A model is directly identified from empirical data and a suitable optimization algorithm is applied to increase model accuracy. This allows one to introduce constraints into the problem that can improve the power of the fault detection algorithm. The number of applications using nonlinear models has been grown. Many nonlinear processes under investigation belong to the group of thermal and fluid dynamic processes.

Linear or non-linear model-based [60] approaches to fault diagnosis problem may not provide accurate results to isolate faults, since it is very difficult to build accurate mathematical models of the systems. The system information is incomplete or uncertain [27], [28]. Hence, it is essential to deal with the incomplete knowledge in an efficient way. Moreover, due to the complexity of nonlinear systems, model-based methods deal with the linearized system models, that are subjected to simple single and multiple faults. Hence, this assumption has limited the success in practical applications. To overcome these difficulties, a more suitable solution may be the utilization of knowledge-based techniques (i.e., a fault diagnosis problem is solved using the knowledge of cause-effect relations). The knowledge-based FDD is achieved by using analytical and heuristic knowledge. The features from system characteristic values are extracted, while system working normal and under faulty conditions. If no information is available on the fault-event relations, classification methods (i.e., fuzzy clustering, artificial neural network and probabilistic methods) can be used. If more information about events and faults is available, different methods of reasoning (i.e., probabilistic reasoning, probabilistic reasoning with fuzzy logic and reasoning with artificial intelligence [25], [27], [29], [32], and [54]) can be applied. The main benefit of the probabilistic reasoning approach is to provide a treatment of uncertainty, but in order to reduce the computational effort; independent events have to be assumed [33]. Fuzzy sets and fuzzy logic are used to deal with uncertainty [10], [24], [25], [26]. Such representations and calculations are mathematically precise. Hence, the fuzzy logic reasoning (IF-THEN-rule system) can be quite appropriate if there is uncertainty [25], [27], and [55]. When fuzzy reasoning is utilized, it is possible to present the results in the form of possibility of faults and their sizes [33]. The adaptive neuro-fuzzy systems can be used in order to improve a rule-base further [3], [22]. The extension of knowledge-based approaches (i.e., neural networks, adaptive neural networks, neuro-fuzzy systems and hybrid neuro-fuzzy systems) and design methodologies can be seen in [3], [22], [23] and [59].

Recently, the failure diagnosis problem has been investigated via discrete event system (DES) approach [8] and [20]. Although, conventional DES has been applied in many engineering fields, they are not adequate for some other fields. This is especially true when we consider fault diagnosis applications, in which the states (e.g., a component health status) are somewhat uncertain (e.g., degree of fault) and vague in a deterministic sense [22], [23]. Sometimes one may need to model systems that cannot be modeled by the current DES modeling methods due to the vagueness in the definitions of the states and/or events. In order to overcome these difficulties, the concepts of fuzzy discrete event system (FDES) can be used. In this structure states and events are fuzzy valued [11], [10]. Lin and Ying [13], [14] initiated the study of FDES, and then applied their results about FDES to HIV/AIDS treatment planning problem [31]. One of the few studies on FDES is given in [30] where supervisory control of FDES systems has been dealt with. In order to solve the fault diagnosis problem, an FDES or a dynamic FDES (i.e., FDEDS) approach based on fuzzy rule-base can be used. The construction of the FDES / FDEDS framework is so simple that it allows one first to build each component model separately. The FDES / FDEDS concept is more convenient in the investigation of multiple failures occurring at the same time. In the literature, there are few FDD applications employing DES [8] but no FDES based applications, so far.

In this thesis, we generalized some properties (i.e., observability and diagnosibility) of the DES to the FDES / FDEDS. New fuzzy-observability and diagnosibility concepts [13], [14], [15], [16], [17], [18] are presented. One can give a degree of observability and diagnosibility about nonlinear systems using these newly proposed definitions. The proposed FDES and FDEDS can be considered as continuous time, nonlinear systems. We have found the observability definition of the DES proposed by Özveren and Willky [16] convenient to handle nonlinear systems' observability. This definition allows us to generalize the DES observability concept to FDES / FDEDS observability concepts. However, it is not easy to measure observability degree of a system using these observability definitions originated from Özveren and Willky's DES observability definition. Hence a new definition is needed. To measure information related to FDES, first Lin and Ying have generalized crisp observability to fuzzy observability. This new fuzzy-observability concept, which is different from other uncertainty measurement methods [24], [25], [26], [27], is presented to handle uncertainty in the systems. It is easy to check systems' observability degree using this definition, but there are some difficulties in the application of this definition. In the Lin and Ying approach [13], [14], the crisp and fuzzy observability are based on a similarity matrix based on consistency of decisions at different states (a particular decision under consideration). This observability depends not only on the DES and the set of observable events (all the events occur at the same time with different membership degrees in our approach), but also on a particular decision to be made, which is represented by the similarity matrix. It is not easy to construct the similarity matrix. To overcome this difficulty, in our approach to define observability, we have used a special relation among states. This is constructed by using a "dissimilarity" (relation) matrix. Dissimilarity matrix construction is easier than the similarity matrix construction.

1.7 Objectives and Contributions

This thesis focuses on model and knowledge-based fault detection and diagnosis methods for nonlinear dynamic systems. The objectives are to

- Propose three model-based FDD approaches by utilizing principal component analysis (PCA), system identification based FDD and inverse model based FDD and apply them to different nonlinear systems starting from simple to more complex. The latter is done to investigate their applicability.
- Propose a priority and a degree to faults.
- Propose new FDES and FDEDS structures and investigate their applicability to an FDD problem.
- Propose two knowledge-based FDD methods based on FDES and FDEDS structures using a fuzzy rule-base and apply them different nonlinear dynamical systems.
- Handle the diagnosibility problem as an unknown input observability problem.

- Derive new properties related to the FDES and FDEDS such as fuzzy observability and diagnosibility concepts and give a relation between them.
- Propose a dynamic rule-base extraction method by using analytic and heuristic knowledge by using classification techniques based on the information coming from residuals.
- Propose a dynamic and a static expert diagnoser design method.
- Construct a nonlinear and events based extension of the Luenberger observer and use them as a diagnoser to isolate faults.
- Make comparisons between the model and knowledge-based FDD methods proposed.

In order to address these objectives the thesis contributes in the following way:

- The details of the FDD method utilizing PCA, identification and inverse model based FDD are addressed. Several simulations are done starting from a simple nonlinear system up to a complex nonlinear system to analyze the performance of the methods.
- New FDES and FDEDS structures are illustrated and fuzzy event and state concepts are introduced.
- New fuzzy observability and diagnosibility definitions and relations between these concepts have also been given. It is possible to give observability and diagnosibility degrees by using these new definitions. It is not easy to check a system's observability by using the observability definition. To overcome this difficulty a simple observability checking method is proposed. By using this new observability checking method one can easily check a system's observability.
- The diagnosibility problem is handled as an observability problem Faults are considered as system inputs. By this way, they can be determined by using an unknown input observer.

- A nonlinear and event-base extension of the linear Luenberger observer is illustrated and how to use it as an observer/diagnoser to produce residuals and isolate faults is explained.
- The utilization of the new FDES and FDEDS structures to solve FDD problems in nonlinear dynamic systems are presented. A dynamic and static expert diagnoser design procedure is introduced to isolate faults. The FDES and FDEDS based FDD techniques are applied to detect and diagnose component faults in an induction motor and an unmanned small helicopter. It is possible to give a priority and degree about faults by this new knowledge-based FDD techniques
- It is also shown how to design a dynamic rule-base by using a system' states and/or events. Based on the analytical and heuristic knowledge, how to generate events and construct a rule-base using k-means clustering technique have been presented.
- The proposed knowledge-based FDD methods' performances are compared with the model-based FDD methods performances.
- Based on the simulation results the advantages and disadvantages of the methods proposed are discussed.

1.8 Thesis Outline

The thesis is organized as follows:

Chapter 2 gives a brief introduction into the field of model-based fault detection and diagnosis. The idea of PCA, identification and inverse model based FDD are briefly addressed.

Chapter 3 represents the application of the FDD utilizing PCA approach to an induction motor, and the application of identification and inverse model based FDD approach to a gas pipeline system, a four-tank system and an induction motor. The simulation results show the proposed FDD methods performance. Finally, the results are discussed to evaluate the performance of the model-based approaches.

Chapter 4 introduces the concepts of FDES and FDEDS structures. In this chapter the concept of fuzzy observability is illustrated. A simple fuzzy observability checking method is proposed.

Chapter 5 gives a brief introduction into the field of knowledge-based fault detection and diagnosis techniques, and it is shown how to apply the proposed FDES and FDEDS structures in a FDD problem. A new fuzzy diagnosibility concept and a relationship between diagnosibility and observability concepts are given. Then, the Luenberger observer design and expert diagnoser design procedures for FDES and FDEDS, event generation techniques using analytic and heuristic knowledge and rule extraction methods by k-means clustering are clearly addressed. Lastly, conclusions are given.

Chapter 6 illustrates the application of the knowledge-based FDD approach using FDES and FDEDS structures to an induction motor, and an unmanned small size helicopter. Finally, the results are discussed to evaluate the performance of the proposed model-based approaches.

Chapter 7 gives a comparison between model and new knowledge based methods and summarizes the contributions and achievements of the thesis providing some suggestions for possible further research topics as an extension of these works.

1.9 Conclusions

The first chapter of the thesis tried to suggest a common terminology in the fault detection and diagnosis framework in order to comment on some developments in the field of FDD based on papers given in the references. The contents of 6 chapters composing this thesis and the main contributions were presented.

CHAPTER 2

MODEL BASED FDD

2.1 Introduction

Model-based approaches to fault detection in dynamic systems have been received much attention over the last decades, both in research context and in the domain of application studies on real plants. The most important issue in model-based FDD concerns the accuracy of the model describing the behavior of the monitored system. This issue has become a central research theme over recent years. The field of model-based FDD is well studied. As mentioned in the introduction there exist a wide variety of model-based FDD approaches for linear systems, e.g., observer-based approach, the parity space approach, and the system identification approach. Key references can be found in [1], [23], [24] [62] and [66]. Also for nonlinear systems, there exist several model-based FDD methods such as [63] and [69]. However, most of the approaches handle only a specific class of nonlinear FDD problem. This is mainly due to the fact that there exists different classes of nonlinear systems also including phenomena like saturation effects or non-analytical behavior.

2.2 Model Based FDD Techniques

Model-based FDD can be defined as the detection and diagnosis of faults on a system by means of methods, which extract features from available signals (i.e., known inputs and measurements) and process's mathematical model. Model based FDD is also called analytical redundancy. Faults are detected by setting fixed or variable thresholds on residuals generated from the difference between actual measurements and their estimates obtained by using the process model. A number of residuals can be created each being sensitive to individual faults occurring in different locations of the system. The analysis of each residual, once the threshold is exceeded, leads to fault diagnosis.

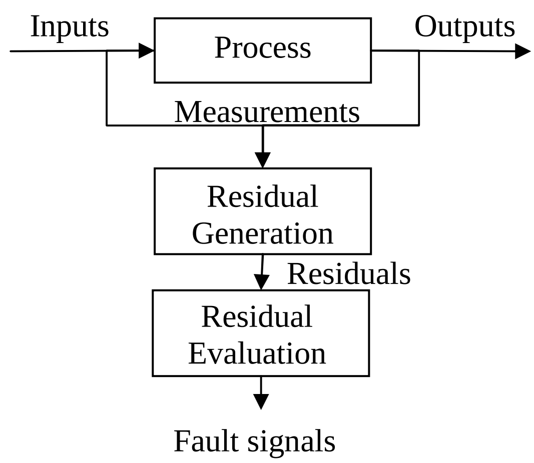

Figure 2.1: Structure of a model-based FDD system

Figure 2.1 shows the general block diagram of a model-based FDD, as generally accepted by the fault diagnosis community. The two main blocks are described as residual generation and residual evaluation blocks. **Residual generation** block generates residual signals using available inputs and outputs from the monitored system. Its output should be normally zero or close to zero under no fault condition. The procedure used to compute residuals is called *residual generation.* Such a procedure is used to extract fault symptoms from the system, with the fault symptom represented by the residual signal. Most of the contribution in the field of model based FDD focuses on the residual generation problem, since the decision-making becomes relatively easy if residuals are well designed especially in the multiple fault case. The difficulty of handling multiple faults lies in the fact that resulting fault effects caused by single faults occur at the same time. Hence, they might compensate each other or they might add up in a way that either only one of them or a completely other fault is detected and diagnosed. Therefore, it is important to obtain the correct residual structure for correct residual evaluation. **Residual evaluation** stage examines residuals for the faults and a decision rule-base is then utilized to determine if any faults have occurred. The residual evaluation stage may perform a simple threshold test on the instantaneous values or moving averages of the residuals, or it may utilize statistical methods such as generalized likelihood ratio testing. The model-based FDD techniques are restricted because they require a precise model to obtain sufficiently high FDD performance.

2.2.1 Residual Generation Techniques

The generation of symptoms is the main issue in the model-based FDD. A variety of methods are available in literature for residual generation. The residual generation for model-based FDD is based on the available analytical redundancy. In most approaches the analytical redundancy is represented by a set of differential equations. The aim is to generate structured residuals to obtained sufficiently high FDD performance. A common way to generate residuals is to estimate system output vector y or the system parameter vector Ω. Then the estimates $\hat{y}$ and $\hat{\Omega}$ are subtracted from the real measurement y and the nominal value of the parameter Ω_{nom}. This leads to the following residual vectors:

$$r_y = y - \hat{y} \quad \text{and} \quad r_\Omega = \Omega_{nom} - \hat{\Omega}$$

The residual vector r_Ω recalls the parameter estimation approach. The residual vector r_y is typically encountered in the dedicated observers (i.e., observer excited by one system output, Kalman filter excited by all system outputs, bank of observers excited by all system outputs, bank of observers excited by a single system output, bank of observers excited by all system outputs except one), output observers (i.e., used to construct system output signals) and fault detection filters (i.e., a class of Luenberger observers with a specially designed feedback gain matrix), but it is also used by the so called parity relation approach [63], [66]. Residuals can also be produced by the methods of change detection, since the presence of noise, disturbances and other unknown signals, the measured and estimated signals, parameters, state variables are usually stochastic variables with some mean value and variance.

2.3 Robustness

Model-based FDD methods are based on mathematical models; however, a precise and accurate model of a real system might not be easy to obtain. There are some obvious reasons; e.g., unknown structure of disturbances, different noise effects, and uncertain or time varying (due to aging) system parameters. FDD methods that are able to handle this kind of model uncertainty are referred to as robust. Model uncertainty can cause false and missed alarms; hence, it needs to be considered when implementing FDD systems. If it is not handled properly, it can have strong impact on FDD performance. There exist several approaches to handle the robustness issue. They are divided into two groups as active and passive robustness approaches. The active robustness approach deals with the model uncertainty in the residual generation phase. The aim is to avoid model uncertainty effects on the residuals. The passive robustness approaches are implemented in the residual evaluation phase, e.g., by using time varying thresholds, also known as adaptive thresholds. For further details about robust FDD, the papers [60], [62] and [63] can be seen.

In the next section the principal component analysis (PCA) and the system identification approaches will be addressed.

2.4 PCA Based FDD

PCA is one of the multivariate statistical techniques, which can reduce the dimensionality. It was fist introduced by Karl Pearson in 1901 and developed by Hotelling [75]. The ideas were applied to solve FDD problems. The FDD can be implemented in a low dimensional space by monitoring square prediction error (SPE) and principal score charts using PCA. The methods are based on malfunction detection, but they might lead to difficulties in the diagnosis process. There are several existing diagnostic tools based on PCA. These tools use score plot, SPE plot, contribution plot or loading plot. Process monitoring and fault diagnosis using PCA were studied intensively and applied to industrial processes. In the literature, linear PCA and its various extensions (like multi-scale PCA, neural PCA, model based PCA or multiple local PCA) were applied to a variety of dynamic and static systems to diagnose system faults [71], [72], [73], [77], [78], [79], [82] and [83].

Some researchers propose modifications to linear PCA, which renders it more suitable to apply to the data collected from industrial processes. One of the first approaches, introduced by Kramer in 1991, uses an auto associative neural network trained using back propagation. Despite its successful applications, the non-linear representations generated by this technique are not true principal components since; they can not be guaranteed to be orthogonal to each other, [70], [79] and [84].

Many other approaches also have been suggested to extend the monitoring capabilities of PCA using different methods such as support vector machines [81] and genetic programming [74]. Recently, another linear transformation method called independent component analysis (ICA) has been studied in data analysis [76] and used FDI purposes [80]. It includes higher-order statistics, rather than a second-order one, to extract the independent hidden factors from the observation data. It uses information on the distribution of the data matrix that is not contained in the covariance matrix. In order for this to be meaningful, the distribution of the data matrix must not be Gaussian, since all the information of Gaussian variables is contained in the covariance matrix.

2.4.1 System

The discrete time linear system is assumed to be of the form

$$
\begin{aligned}
x_{k+1} &= Ax_k + Bu_k + B_f f_k \\
y_k &= Cx_k + Du_k + D_f f_k
\end{aligned}
\tag{2-1}
$$

where x_k, u_k and y_k denote the state, the known input applied to the system and the system output at time *k,* respectively. f_k is the unknown fault input and the matrices B_f and D_f determine which part of the system (i.e., components, actuators or sensors faults) will be affected by different faults.

If the data is collected dynamically, i.e., $Y_k = (y_{k-l+1}^T ... y_k^T)^T$ the output signal Y_k can be formulated as:

$$
Y_{k,s,N} = \Gamma X_{s,N} + HU_{k,s,N} + GF_{k,s,N} \qquad s \le n \tag{2-2}
$$

where, n, k and N denote the system order, time index and the number of observations, respectively.

$$
\Gamma = \begin{bmatrix} C \\ CA \\ \cdot \\ \cdot \\ \cdot \\ CA^{s-1} \end{bmatrix} \quad X_{s,N} = \left[x_s, x_{s+1}, ..., x_{s+N-1} \right]
$$

$$
H = \begin{bmatrix} 0 & 0 & 0 & ... & 0 \\ CB & 0 & 0 & ... & 0 \\ CAB & CB & 0 & ... & 0 \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ CA^{s-1}B & CA^{s-2}B & ... & CB & 0 \end{bmatrix}
$$

$$G = \begin{bmatrix} D_f & 0 & 0 & ... & 0 \\ CB_f & D_f & 0 & ... & 0 \\ CAB_f & CB & D_f & ... & 0 \\ . & . & . & ... & . \\ . & . & . & ... & . \\ . & . & . & ... & . \\ CA^{s-1}B_f & CA^{s-2}B & ... & CB_f & D_f \end{bmatrix}$$

The sequences u_k and f_k are stored in a similar way in the matrices $U_{k,s,N}$ and $F_{k,s,N}$. This representation is called the parity space model. After obtaining the parity space model, we can define a residual (r) as:

$$r = W^T (Y_{k,s,N} - HU_{k,s,N}) = W^T GF_{k,s,N} \qquad (2\text{-}3)$$

where the W is selected as a basis for the null space of Γ, i.e., $W^T\Gamma = 0$. The equation shows that the system faults can be found by using measured system outputs and system matrices. However, this approach is used if a nonlinear system's linearized model is utilized. Hence, PCA based FDD approach can be a solution to determine faults in nonlinear complex systems as long as their mathematical models are available.

2.4.2 PCA Background

A principal component is defined as a linear transformation of the original variables, which are normally correlated, into a new set of variables that are orthogonal to each other. The basic goal in PCA is to reduce the dimension of the data. This is done in the mean square sense. Such a reduction in dimension decreases the computation time and removes the effects of the noise. We know that in PCA the data matrix Y may be decomposed as:

$$Y = W\Sigma^{1/2}V^T \qquad (2\text{-}4)$$

where W is an m×m matrix of eigenvectors of YY^T, V is an n×n matrix of eigenvectors of Y^TY, the data correlation matrix. The elements of diagonal matrix $\Sigma^{1/2}$ are the positive square roots of the eigenvalues $\lambda_i \, (i = 1,...,n)$ of $Y^T Y$ and are called the singular values. The principal components of *Y* are the columns of the scores matrix *T* defined as

$$T = YV = W\Sigma^{1/2} \tag{2-5}$$

The i^th principal component is essentially a weighted sum of the standardized variables where the weights are defined by the elements of the i^th eigenvector, v_i, i.e.

$$t_i = Yv_i \quad \text{(i =1…n)} \tag{2-6}$$

The variance of t_i is given by

$$\mathrm{var}(t_i) = v_i Y^T Y v_i = \lambda_i \tag{2-7}$$

The variance of each component is given by the corresponding eigenvalue, λ_i of the data correlation matrix, since the correlation matrix is at least positive semidefinite all its eigenvalues have magnitudes grater than or equal to zero. In SVD they are usually arranged in descending order. By discarding those principal components that do not contribute to overall variation the dimension of the problem is reduced.

$$\begin{aligned} Y &= \begin{bmatrix} W_1 & W_2 \end{bmatrix} \begin{bmatrix} \Sigma_1^{1/2} & 0 \\ 0 & \Sigma_2^{1/2} \end{bmatrix} \begin{bmatrix} V_r^T \\ \hat{V}_r^T \end{bmatrix} \\ &= \begin{bmatrix} W_1\Sigma_1^{1/2} & W_2\Sigma_2^{1/2} \end{bmatrix} \begin{bmatrix} V_r^T \\ \hat{V}_r^T \end{bmatrix} = \begin{bmatrix} T_r & \hat{T}_r \end{bmatrix} \begin{bmatrix} V_r^T \\ \hat{V}_r^T \end{bmatrix} \\ &= T_r V_r^T + \hat{T}_r \hat{V}_r^T \end{aligned} \tag{2-8}$$

By using equations (2-5) and (2-8), it becomes

$$Y = YV_r V_r^T + Y\hat{V}_r \hat{V}_r^T = \hat{Y} + \overset{\circ}{Y} \tag{2-9}$$

In equation 2-8, W_1 and W_2 are matrices containing eigenvectors related to the system model and error. V_r and T_r are called reduced loading and scores matrix, respectively. Making the same derivation as above, the mathematical form of the PCA can also be written by using the matrices W_1 and W_2 as:

$$Y = W_1 W_1^T Y + W_2 W_2^T Y = \hat{Y} + \mathring{Y} \tag{2-10}$$

Looking at the equations 2-9 and 2-10 it can be said that PCA splits the data into two parts, model ($\hat{Y}$) and error ($\mathring{Y}$). This equation is similar to the equation 2-1. Hence, this approach can also be considered as a model-based approach.

2.4.3 FDD with PCA

In this sub section, a PCA based fault detection (FD) algorithm is implemented to determine parameter, and actuator faults. The difference between conventional model based FD and PCA based FD can be seen Figure 2.2.

There are three steps in the PCA based FD approach (i.e., data manipulation, off-line procedure, on-line fault monitoring). The first step is data manipulation stage. The data matrix Y can be constructed in two ways (i.e., static and dynamic). These matrices are constructed under normal operating conditions from the samples of the system inputs ($u(k)$) and/or outputs ($y(k)$) like $Y = [u_k \; y_k]$ and $Y = y_k$.

$$Y_{static} = Y_k^T, \quad Y_{dynamic} = [Y_{k-l+1}^T \quad Y_{k-l+2}^T, \; \ldots, \; Y_k^T]^T \tag{2-11}$$

where l denotes system order and Y contains input/output data of length k. Since different variables in engineering systems usually use different units, the columns of Y usually need to be scaled; so that they have zero mean and unity variance.

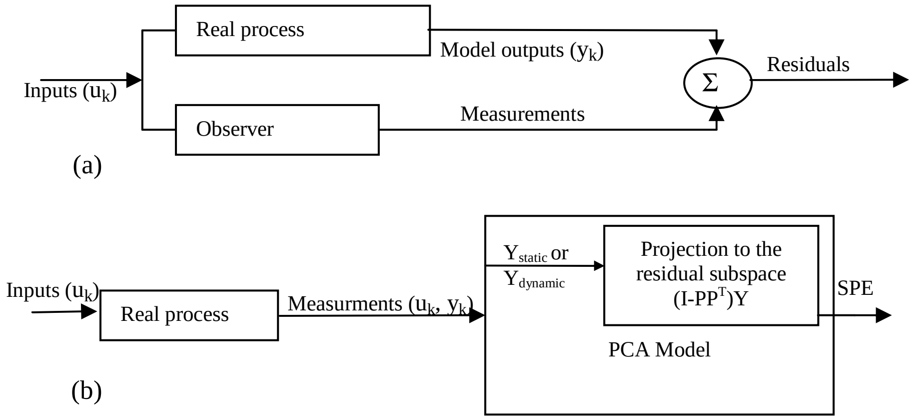

Figure 2.2: Comparison of conventional model based FDD method with PCA based FDD method: (a) scheme of conventional model based FDD method; (b) scheme of PCA based FDD method.

2.4.3.1 Off Line Procedure

The off-line calculation procedure is used to calculate mean, variance and principal components (PCs) of the data matrix. The data matrix is also auto scaled (zero mean, unity variance) using calculated mean and variances before constructing the correlation matrix (covariance) in this stage. Then the covariance matrix is constructed by using this auto scaled matrix. The covariance matrix is calculated as:

$$Cov = \frac{Y^T Y}{n-1} \qquad (2\text{-}12)$$

where Y shows auto scaled data matrix. To calculate PCs, the eigenvectors and eigenvalues of the covariance matrix were computed and arranged in decreasing order of eigenvalues. The eigenvectors of the auto scaled covariance matrix are called PCs.

2.4.3.2 On Line Fault Monitoring

In the on-line fault monitoring stage, each new observation vector is auto scaled using the means and variances obtained in the off-line stage and projected onto the principal component sub-space. Then the residual (*R*) is calculated, using a few principal components (PCs) related to the system.

$$R = \left\| Y - \hat{Y} \right\|^2 = \left\| (I - W_1 W_1^T) Y \right\|^2 \tag{2-13}$$

In a different way, the residual (*R*) is calculated using a few PCs related to the error matrix W_2 as:

$$R = \left\| W_2 W_2^T Y \right\|^2 \tag{2-14}$$

After scaling, R becomes:

$$\overline{R} = \left\| W_2 \Sigma_2^{-1/2} W_2^T Y \right\|^2 \tag{2-15}$$

where $\hat{Y}$ is called prediction of the measurement vector. If residual exceeds a predefined threshold value, it is said that faults occurred in the system. After detecting faults, a fault isolation technique is needed. Threshold based fault isolation techniques may not work well in the PCA based FDD. Hence, classification techniques or reasoning based fault isolation methods should be used.

2.5 System Identification Based FDD

The problem of identifying an unknown system given samples of its behavior is well known [59], [61] and [85]. When a priori knowledge on the characteristics of the unknown system is available, the identification procedure can be applied. This knowledge may act as a set of constraints shaping the space of models so that identification in this new space is an easier problem. As an example, the regularity of the unknown system can be converted into smoothness constraints of some kind, transforming the identification problem into a minimization problem. In recent years, system identification approaches for nonlinear systems have been applied successfully. This part of the thesis focuses on a general, practical method of nonlinear FDD method, based on system identification. The proposed FDD method can be applied to a wide range of nonlinear systems.

2.5.1 Description of the Method

The proposed identification based FDD approach is mainly based on system fault model. Hence we will first begin to define what a fault model is.

2.5.1.1 Fault Model

A nonlinear system model can be given as:

$$\begin{aligned} \dot{x}(t) &= f(x(t),\ u(t)) \\ y(t) &= g(x(t),\ u(t)) \end{aligned} \tag{2-16}$$

where $x \in R^n$ is the state vector, $u \in R^m$ is a known input, and $y \in R^p$ are the measured outputs and t is the time index. The proper starting point to construct identification based FDD is first to establish a system fault model; because the proposed identification method is based on the system fault model [56], [57]. To construct this model, all possible common faults related to system components, sensors and actuators are inserted in the system's mathematical model. In this thesis we are mostly interested in additive component faults. Hence, a nonlinear system model including all possible additive component faults can be given as:

$$\begin{aligned} \dot{x}(t) &= f_1(x(t),\ \bar{u}(t)) \\ &= f(x(t), u(t)) + f_c(t) \\ y(t) &= g(x(t),\ u(t)) \end{aligned} \tag{2-17}$$

where $x \in R^n$ is the state vector, $\bar{u}(t) = (u(t),\ f_c(t)) \in R^m \times R^q$, $u(t) \in R^m$ is a known input, $f_c(t) \in R^q$ is an unknown fault vector, and $y \in R^p$ is the measured output and t is the time index. The fault model will be used in the optimization based FDD algorithm.

2.5.1.2 Identification Based FDD Method

The system dynamics is given in equation 2-16. The problem is to determine the unknown fault vector $f_c(t)$ in equation 2-17. Suppose that the system states are directly measurable between an initial time (i.e., the time at which residuals exceed a predefined threshold value) t_{in} and a final time t_f. Then the measurement data is collected between an initial and final time. The initial state vector is also known. The unknown fault vector $f_c(t)$ can be determined as a solution of a nonlinear optimization problem as:

$$\begin{aligned} &\min \int_{t_{in}}^{t_f} (x(t) - x_m(t))^T (x(t) - x_m(t))dt \\ &subject\ to\ \dot{x}(t) = f(x(t),\ u(t)) + f_c(t) \\ &\qquad\qquad f_c(t) \geq 0 \end{aligned} \tag{2-18}$$

where $x(t)$ and $x_m(t)$ shows calculated and measured state values at time instant *t*, respectively.

2.6 FDD with Inverse Model

This is a special case of the system identification based FDD method. A static system model can be given as:

$$y(t) = g(u,t) \tag{2-19}$$

where $u \in R^m$ is a known input, and $y \in R^p$ are the measured outputs and t is the time index. All possible common faults related to system components, sensors and actuators can be inserted in the system's mathematical model as.

$$\begin{aligned} y(t) &= g_1(u,\ f_c,t) \\ &= g(u,t) + f_c(t) \end{aligned} \tag{2-20}$$

where $f_c(t) \in R^q$ is an unknown fault vector. The problem is to determine the unknown fault vector $f_c(t)$ in equation 2-20. If the system outputs are measurable and the system inputs are known, the unknown fault vector $f_c(t)$ can be determined by using an algebraic calculation as

$$f_c(t) = g_2(y, u, t) \tag{2-21}$$

2.7 Conclusions

In this section, two kinds of model-based approaches are introduced. For some systems, the system dynamics and hence any changes thereof can be identified to a considerable extent by analyzing only the correlations of the output data. To isolate faults in single fault case an adaptive threshold may be enough. However in multiple fault cases threshold based fault isolation techniques are not enough. Hence, intelligent residual classification techniques should be used.

The identification based FDD presented here is flexible, easy to implement and can be applied to a wide range of nonlinear dynamical system as long as the system states are available. Applying constraints to the identification problem also allows for additional information to be incorporated in the fault detection scheme, and can increase the power of the fault detection process. It is also possible to give priority and degrees about faults using the identification based FDD approach.

Main difficulty with the inverse model based FDD approach is the number of unknowns. Sometimes, it may not possible to determine faults if the numbers of unknown variables are more than the number of equations in the model.

CHAPTER 3

MODEL BASED FDD APPLICATIONS

3.1 Introduction

In the following section, three examples are presented in order to test the FDD techniques studied in Chapter 2. The FDD is performed by using PCA and identification based approaches. In this thesis, we are mostly interested in component faults since it is a more challenging task to detect such kinds of faults than the actuator faults. Hence, single and multiple additives, abrupt and incipient component faults with input and system disturbances are considered and simulated. The following processes are described.

1. MIMO simulink model of the quadruple-tank process with two PID controllers.
2. MIMO simulink model of an induction motor working with electrical mains.
3. MIMO MATLAB model of a gas pipeline system.

3.2 FDD in Quadruple Tank Process with System Identification

The quadruple tank system as shown in Figure 3.1, originated by Johansson [86] was designed and built at the University of Delaware. The system consists of four interconnected water tanks, two pumps and associated valves. The system inputs are the voltages supplied to the pumps v_1 and v_2 and the outputs are the water levels in the tanks, *h1 … h4*. The flow into the each tank is adjusted using the associated valves γ_1 and γ_2.

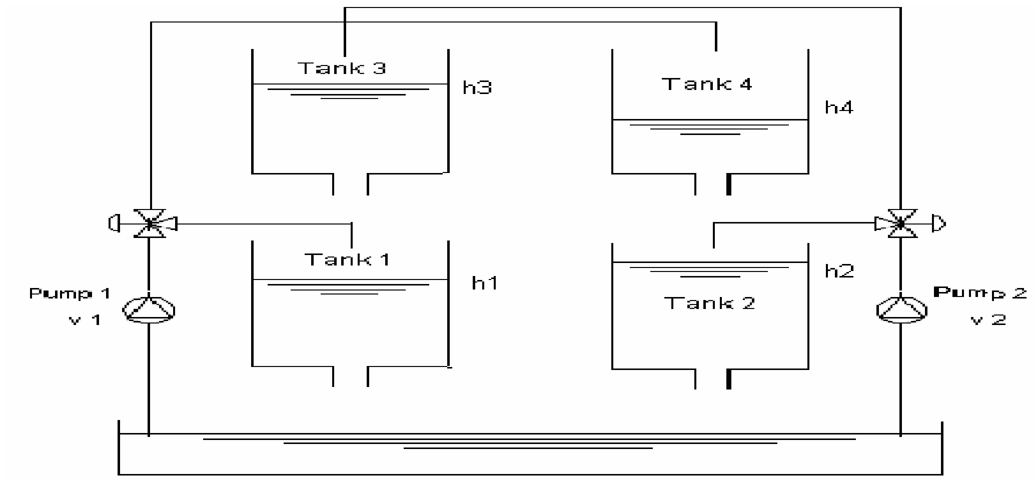

Figure 3.1: Quadruple tank system.

A nonlinear mathematical model of the four-tank model is derived based on mass balances and Bernoulli's law. Mass balance for one of the tanks is

$$A\frac{dh}{dt} = -q_{out} - q_{in} \tag{3-1}$$

where *A* denotes the cross section of the tank, *h*, q_{in} and q_{out} denote the water level, the inflow and outflow of the tank, respectively. In order to establish a relationship between output and height Bernoulli's law is used. It states that

$$q_{out} = a\sqrt{2gh} \tag{3-2}$$

where *a* is the cross section of the outlet hole (cm^2) and *g* is the acceleration due to gravity. This relationship is roughly the expected output due to height relationship although it does not take into account any flow dynamics of the orifice. A common multiplying factor for an orifice of the type being used in this system is coefficient of discharge *k.* We can therefore rewrite Bernoulli's equation as:

$$q_{out} = ak\sqrt{2gh} \tag{3-3}$$

The flow through each pump is split so that a proportion of the total flow travels to each corresponding tank. This can be adjusted via one of the two valves shown in Figure 3.1. Assuming that the flow generated is proportional to the voltage applied to each pump, (change) v, and that q_T and q_B are the flows going to the top and bottom tanks, respectively, we are able to come up with the following relationship.

$$q_B = \gamma kv \qquad q_T = (1-\gamma)kv \qquad \gamma \in [0, \quad 1] \tag{3-4}$$

Combining all the equations for the interconnected four-tank system we obtain the following equations, which represent the physical system.

$$\begin{aligned}
\frac{dh_1}{dt} &= \frac{\gamma_1 k_1}{A_1} v_1 - \frac{a_1}{A_1}\sqrt{2gh_1} + \frac{a_3}{A_1}\sqrt{2gh_3} \\
\frac{dh_2}{dt} &= \frac{\gamma_2 k_2}{A_2} v_2 - \frac{a_2}{A_2}\sqrt{2gh_2} + \frac{a_4}{A_2}\sqrt{2gh_4} \\
\frac{dh_3}{dt} &= \frac{(1-\gamma_2)k_2}{A_3} v_2 - \frac{a_3}{A_3}\sqrt{2gh_3} \\
\frac{dh_4}{dt} &= \frac{(1-\gamma_1)k_1}{A_4} v_1 - \frac{a_4}{A_4}\sqrt{2gh_4}
\end{aligned} \tag{3-5}$$

For tank i, A_i is the cross-section of the tank, a_i the cross-section of the outlet hole, and h_i is the water level. The voltage applied to pump i is v_i and the corresponding flow is $k_i v_i$.

Tank leakage faults are studied in this part. To check robustness, measured h_i and system inputs are corrupted by Gaussian noise. The tank levels are controlled with two PID controllers. Figure 3.2 shows phase characteristics of the system as a function of valve positions. This figure is important since it shows us how to control tank water level. If the majority of the flow is sent to the top tanks, the system will become non-minimum phase resulting in inverse response behavior. The control of tank level is achieved if the system is in non-minimum phase region. It is easier to control h_1 with v_1 and h_2 with v_2 in this region. The control of water levels of tanks is particularly difficult if system is minimum phase.

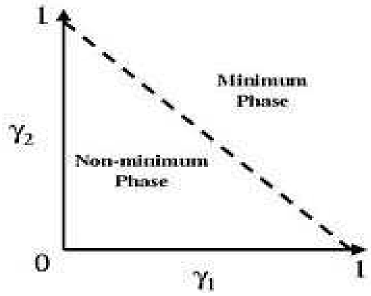

Figure 3.2: Phase characteristics of the system as a function of valve positions

3.2.1 Fault Model for Quadruple Tank

Fault model for the quadruple tank system is given in Figure 3.3. To construct this model extra holes are added to each tank.

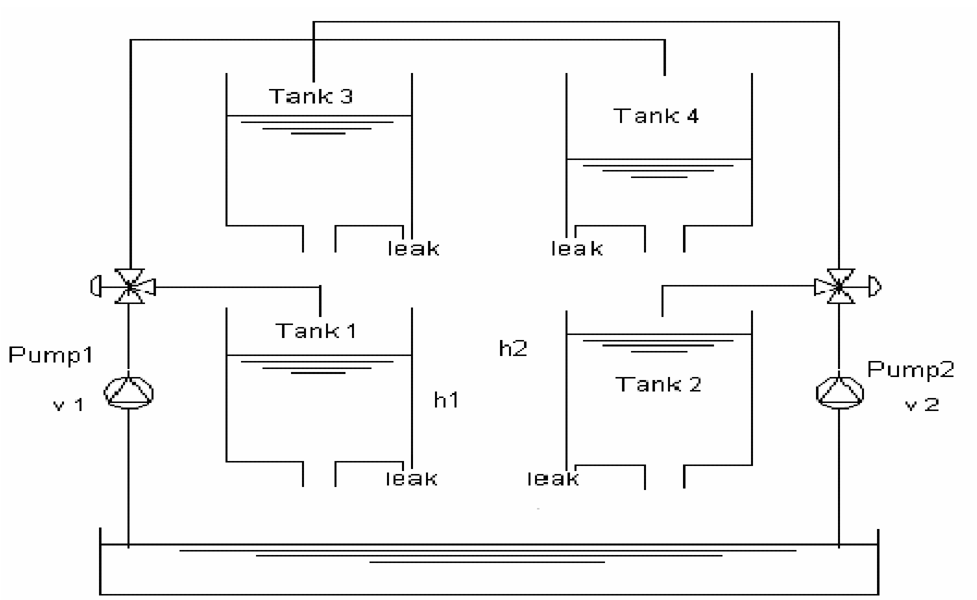

Figure 3.3: Fault model for the quadruple tank system

The mathematical model of the faulty quadruple tank system can be given as:

$$
\begin{aligned}
\frac{dh_1}{dt} &= \frac{\gamma_1 k_1}{A_1} v_1 - \frac{a_1}{A_1}\sqrt{2gh_1} + \frac{a_3}{A_1}\sqrt{2gh_3} - \frac{a_{leak1}}{A_1}\sqrt{2gh_1} \\
\frac{dh_2}{dt} &= \frac{\gamma_2 k_2}{A_2} v_2 - \frac{a_2}{A_2}\sqrt{2gh_2} + \frac{a_4}{A_2}\sqrt{2gh_4} - \frac{a_{leak2}}{A_2}\sqrt{2gh_2} \\
\frac{dh_3}{dt} &= \frac{(1-\gamma_2) k_2}{A_3} v_2 - \frac{a_3}{A_3}\sqrt{2gh_3} - \frac{a_{leak3}}{A_3}\sqrt{2gh_3} \\
\frac{dh_4}{dt} &= \frac{(1-\gamma_1) k_1}{A_4} v_1 - \frac{a_4}{A_4}\sqrt{2gh_4} - \frac{a_{leak4}}{A_4}\sqrt{2gh_4}
\end{aligned} \tag{3-6}
$$

3.2.2 Simulation Results

In the quadruple tank system, the system states are tank liquid levels, which are measurable. The problem is to determine the unknown fault vector $f_c(t) = \left[a_{leak1}(t) \quad a_{leak2}(t) \quad a_{leak3}(t) \quad a_{leak4}(t)\right]^T$ using the system's fault model. Since the system states are directly measurable, they are recorded between an initial time (i.e., residuals that exceed a predefined threshold value at that time) t_{in} and final time t_f. Then, to determine the unknown fault vector f_c(t), the nonlinear optimization problem given in equation 2-18 is solved by using Matlab optimization and simulink toolboxes.

3.2.2.1 Fault Scenarios

Two fault scenarios are created by using the quadruple tank system in the simulation program. In these scenarios incipient single and multiple tank faults (i.e., leakages) are created by changing some system parameters manually during the simulation at certain times. The system inputs, outputs and/or some states are corrupted by Gaussian noise with zero mean and standard deviation of 0.1.

Scenario I

In this scenario, while the system is working in real time, single incipient fault (i.e., tank1 leakage percentage), is created by changing the parameter a_{leak1} to 0.81 cm^2 (i.e., the value 0.81 is 30 percent of the cross-section of the outlet hole of the tank1) in the quadruple tank at 350 seconds. Results obtained are given in Figure 3.4.

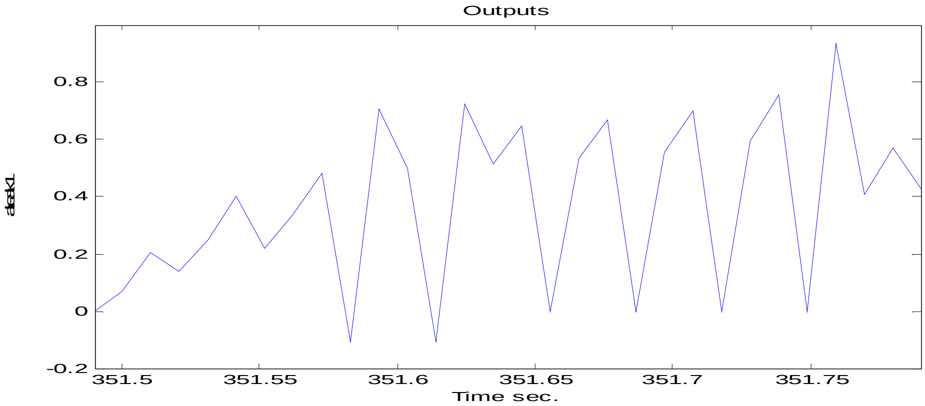

Figure 3.4: Single fault case

Scenario II

In this scenario, while the system is working in real time, multiple incipient faults (i.e., tank2 and 3 leakage percentages) are created by changing the parameter a_{leak2} to 1.62 cm^2, a_{leak3} to 0.54 cm^2 (i.e., the value 1.62 is 60 percent of the cross-section of the outlet holes of the tank2, and 0.54 is 20 percent of the cross-section of the outlet holes of the tank3) in the quadruple tank at 350 seconds. Results obtained are shown in Table 3.5.

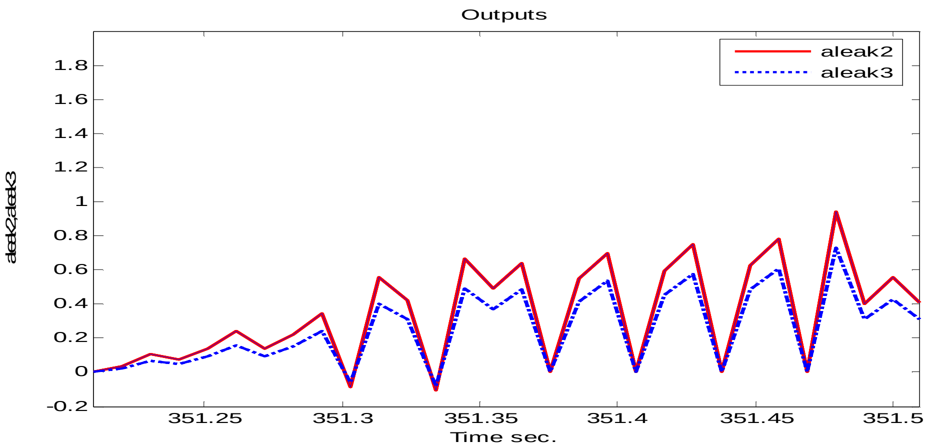

Figure 3.5: Multiple fault case

The results obtained in our various application examples show that, the created fault priorities and calculated fault priorities are exactly same. The proposed identification based FDD approach is robust but there is the problem of convergence of the optimization algorithm.

3.3 PCA Based FDD in an Induction Motor

3.3.1 Induction Motor

An induction motor made by three stator windings and three rotor windings is investigated for FDD [19]. In this thesis, we used the two-phase equivalent machine representation as the mathematical model of the motor, which is also used as the observer. The fifth-order model below gives the overall dynamics of an induction motor under the assumptions of equal mutual inductances and linear magnetic circuit:

$$\frac{dw}{dt} = \frac{n_p M}{J L_r}(\Psi_{ra} i_{sb} - \Psi_{rb} i_{sa}) - \frac{T_L}{J} - \frac{T_{damp}}{J} \tag{3-7}$$

$$\frac{d\Psi_{ra}}{dt} = -\frac{R_r}{L_r}\Psi_{ra} - n_p w \Psi_{rb} + \frac{R_r}{L_r} M i_{sa} \tag{3-8}$$

$$\frac{d\Psi_{rb}}{dt} = -\frac{R_r}{L_r}\Psi_{rb} + n_p w \Psi_{ra} + \frac{R_r}{L_r} M i_{sb} \tag{3-9}$$

$$\frac{di_{sa}}{dt} = \frac{M R_r}{\sigma L_s L_r^2}\Psi_{ra} + \frac{n_p M}{\sigma L_s L_r} w \Psi_{rb} - \left(\frac{M^2 R_r + L_r^2 R_s}{\sigma L_s L_r^2}\right) i_{sa} + \frac{1}{\sigma L_s} u_{sa} \tag{3-10}$$

$$\frac{di_{sb}}{dt} = \frac{M R_r}{\sigma L_s L_r^2}\Psi_{rb} - \frac{n_p M}{\sigma L_s L_r} w \Psi_{ra} - \left(\frac{M^2 R_r + L_r^2 R_s}{\sigma L_s L_r^2}\right) i_{sb} + \frac{1}{\sigma L_s} u_{sb} \tag{3-11}$$

where R, i, ψ, n_p, u_s denote resistance, current, flux linkage, the number of pole pairs and stator voltage input to the machine. The subscripts *s* and *r* stand for stator and rotor, (a, b) denote the components of a vector with respect to a fixed stator reference frame and $\sigma = 1 - (M^2 / L_s L_r)$. L_s, L_r are inductances and M is the mutual inductance, J is the moment of inertia of the rotor and T_L, T_{damp} denote load torque and damping factor related to friction, respectively. w is the rotor speed.

3.3.2 Data Collection Procedure

There are 8 variables measured in the simulation model. These variables are the three phase motor input voltages, three phase motor stator currents, motor speed and motor torque. To collect data, the simulation program was executed from time zero to time 5 seconds. After that, the static and dynamic data matrices were constructed using the measured variables. The dynamic data set is needed since the static data set may not represent very well a nonlinear dynamical system behavior. After collecting the data, the columns of the data matrix were scaled; that is, from the entries in each column their mean values were subtracted and divided by their standard deviations.

3.3.2.1 Construction of the Static Data Matrix

To construct the static data matrix *Y*, the systems' measurable inputs and outputs are used. The *k*×*t* static data set (matrix) can be constructed under normal operating condition from the samples of the measured variables (k: number of samples, t: total number of the systems inputs and outputs) as follows:

$$Y = \begin{bmatrix} U_0(0) & U_1(0) & \ldots & U_n(0) & y(0) & y_1(0) & \ldots & y_m(0) \\ U_0(1) & U_1(1) & \ldots & U_n(1) & y(1) & y_1(1) & \ldots & y_m(1) \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ U_0(k) & U_1(k) & \ldots & U_n(k) & y(k) & y_1(k) & \ldots & y_m(k) \end{bmatrix}$$

where *U* and *y* show induction motor's inputs and outputs and *n* and *m* show the numbers of the inputs and outputs, respectively. Each row of this matrix is also called a measurement vector.

3.3.2.2 Construction of the Dynamic Data Matrix

We know that a static data matrix may not represent system dynamic behavior very well. Hence, a dynamical representation form is needed. The dynamic data matrix is constructed as:

$$Y_d = \left[Y_{k-l+1}{}^T \quad Y_{k-l+2}{}^T , \; \ldots \; , \; Y_k{}^T \right]^T$$

where $l \geq n$, l is called lag and *n* shows order of the system.

3.4 Simulation Results

In this part, PCA based FDD is applied to static and dynamic data sets to detect induction motor bearing and stator winding faults. For this purpose, several scenarios are investigated, but five of them are presented in the thesis. Two different static and dynamic data matrices are constructed. In the first case, to construct the static and dynamic data matrices whose sizes are 3000×7 and 3000×42, seven measurement variables are used. These variables are the induction motor’s three phase input voltages and stator currents and speed. The motor torque measurement is not included in data matrices because torque measurement is expensive. By applying off-line procedure using the static and dynamic data matrices, means, variances and PCs are calculated. 7 and 42 PCs are obtained by using the static and dynamic matrices, respectively. The results can be seen in Table 3.1. If we look at the variances, sum of the three principal components exceeds 80 % of the total (cumulative) variance of the original data when either the static or dynamic data set is used. In application of the PCA method, eigenvectors associated with only the first *n* largest eigenvalues are used in the PCA model. Therefore only *n* principal components need to be analyzed. There is no criterion to choose these *n* principal components, but it is commonplace that it is acceptable to represent data if the sum of the variances of the first few principal components exceeds 80 % of the total (cumulative) variance of the original data. Hence, we used the first tree PCs for on line fault monitoring stage. Then, we applied online fault monitoring procedure as mentioned in chapter 2 to each of the new observation vectors. The lag value is chosen 6 in this case.

Table 3.1: Variances of the data matrixes a) Static case, b) Dynamic case

a)

PCs	**Eigenvalues**	**Variances %**	**Cumulative Variances %**
1	2.5028	35.754	35.7470
2	2.4724	35.32	71.0631
3	1.0022	14.317	85.3806
4	0.51344	7.3349	92.7206
5	0.50919	7.2741	100
6	1.1863e.027	1.6947e.026	100
7	5.343e.030	7.6328e.029	100

b)

PCs	**Eigenvalues**	**Variances %**	**Cumulative Variances %**
1	14.923	35.531	35.531
2	14.788	35.21	70.741
3	6.0268	14.349	85.09
4	3.0536	7.2704	92.360
5	3.0416	7.2419	99.60
6	0.099618	0.23719	99.83
.	.	.	.
.	.	.	.
42	0	0	100

3.4.1 Fault Scenarios

In the fault scenarios, results are obtained by using dynamic and static measurement data vectors. To obtain the residual plots abrupt and incipient faults are created in some system parameters at a certain time. One can also find some other residual plots, which are obtained by decreasing the inputs applied to the system. We did not present the static residual plots here, since all of them show zero residual for all scenarios. It is not possible to detect faults by using static data since the system considered has a dynamical structure.

Scenario I

a) In this scenario, while the system is working in real time, abrupt bearing fault is created by changing the friction coefficient from zero to 1 at time 1.8 seconds. Figure 3.6 shows the dynamic residual plot.

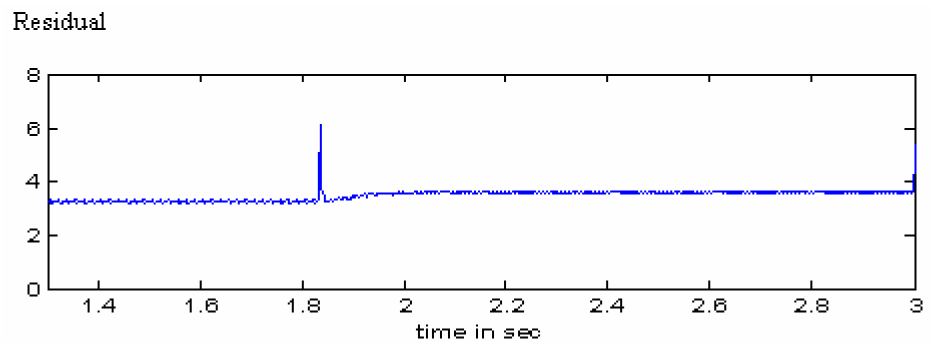

Figure 3.6: Dynamic residual obtained by using input and output measurement variables.

b) Applying the scenario above, the motor friction coefficient is increased gradually from 0 to 1 between 1.5 and 1.8 seconds. The plot is in Figure 3.7.

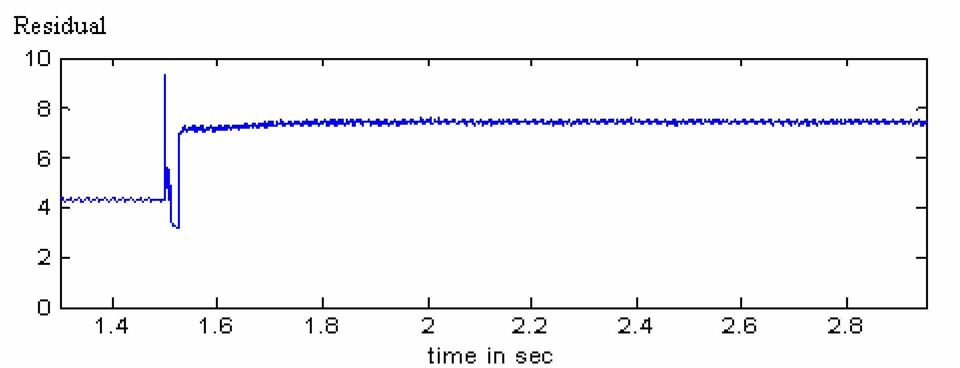

Figure 3.7: Dynamic residual obtained after creating incipient bearing fault.

c) In this part, the system's input values are changed in an acceptable range to check fault monitoring system performance corresponding to an abrupt change (i.e., robustness).

- First, before any fault occurred, the motor rated voltage is decreased from 200 to 160 volts. The result obtained is in Figure 3.8.

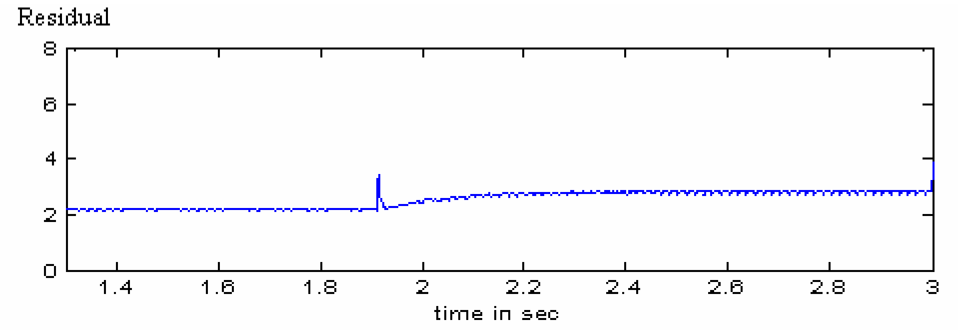

Figure 3.8: Dynamic residual obtained after decreasing the motor's rated voltage.

- Before any fault occurred, the frequency applied to the motor is decreased from 60 to 50 Hz. The result obtained is in Figure 3.9.

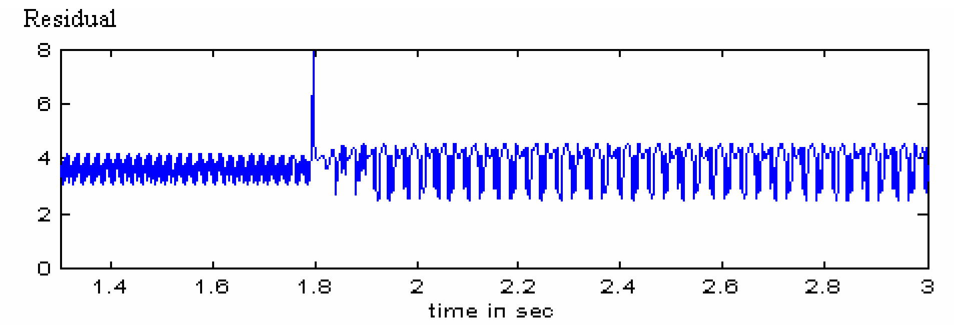

Figure 3.9: Dynamic residual obtained after changing the motor's frequency.

- In this case, before any fault occurred, the frequency applied to the motor is changed from 60 to 50 Hz and the motor rated voltage from 200 to 160 volts. The result obtained is in Figure 3.10.

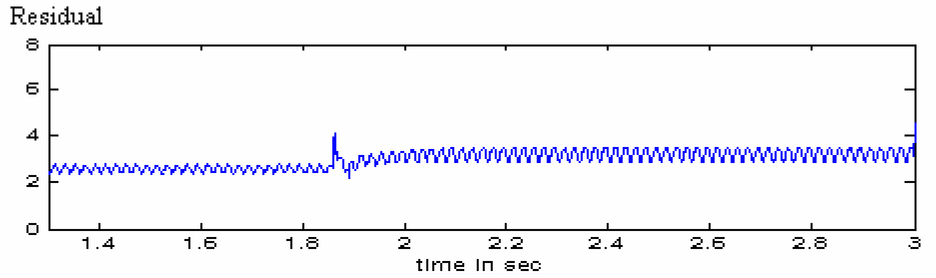

Figure 3.10: Dynamic residual obtained after decreasing the motor's frequency and rated voltage.

Scenario II

a) In this scenario, while the system is working in real time, abrupt winding fault is created by decreasing the stator winding by an amount of 3 % at time 1.8 seconds. Figure 3.11 shows the dynamic residual plot.

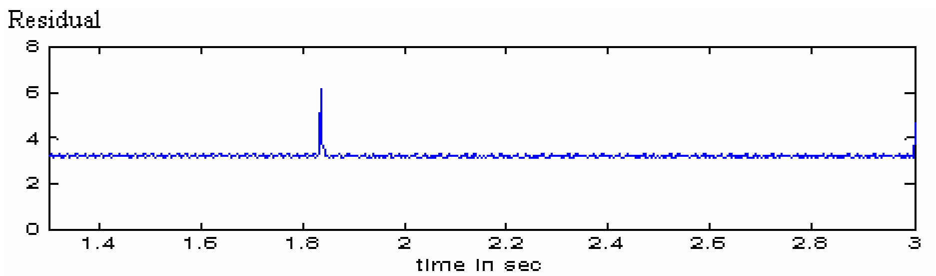

Figure 3.11: Dynamic residual obtained by using input and output measurement values.

b) In this part, again, the system inputs are changed in an acceptable range to see the fault monitoring system performance using scenario 2.

- First, before any fault occurred, the motor rated voltage value is decreased from 200 to 160 volts. The result obtained is in Figure 3.12.

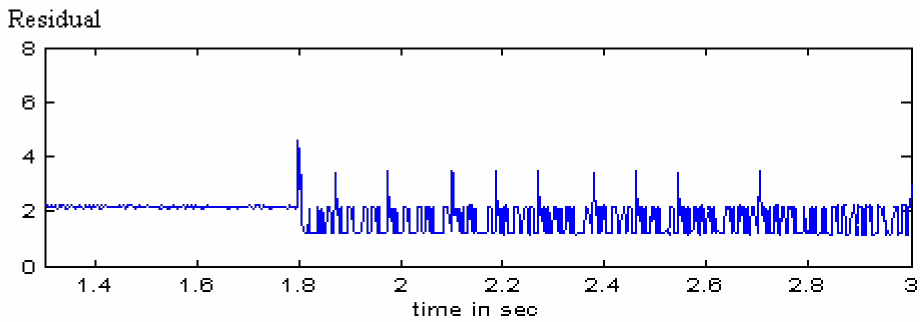

Figure 3.12: Dynamic residual obtained after decreasing the motor's rated voltage.

- Before any fault occurred, the frequency applied to the motor is decreased from 60 to 50 Hz. The result obtained is in Figure 3.13.

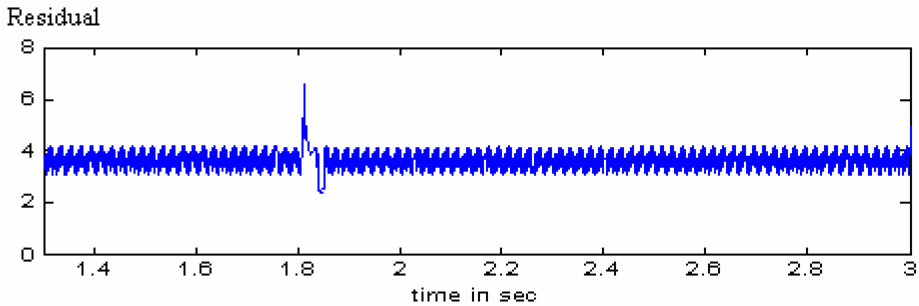

Figure 3.13: Dynamic residual obtained after decreasing the motor's frequency.

- In this case, before any fault occurred, the frequency applied to the motor is decreased from 60 to 50 Hz and rated motor voltage from 200 to 160 volts. The result obtained is in Figure 3.14.

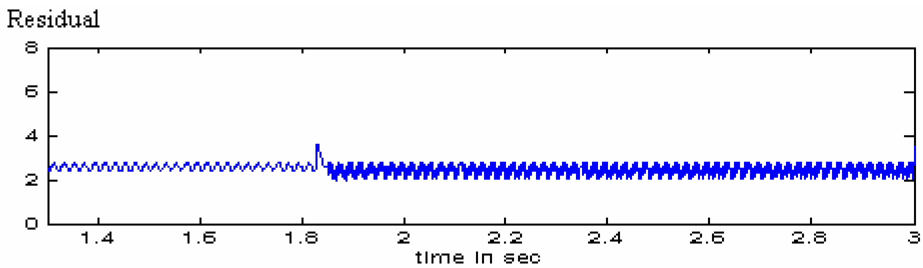

Figure 3.14: Dynamic residual obtained after decreasing the motor's frequency and rated voltage.

Scenario III

In this scenario, while the system is working in real time, first, abrupt winding fault is created at time 1.5 seconds by decreasing the stator winding by an amount of 3 % and then bearing fault is created at time 2.5 seconds by changing the friction coefficient from zero to 3. Figure 3.15 shows the dynamic residual plot.

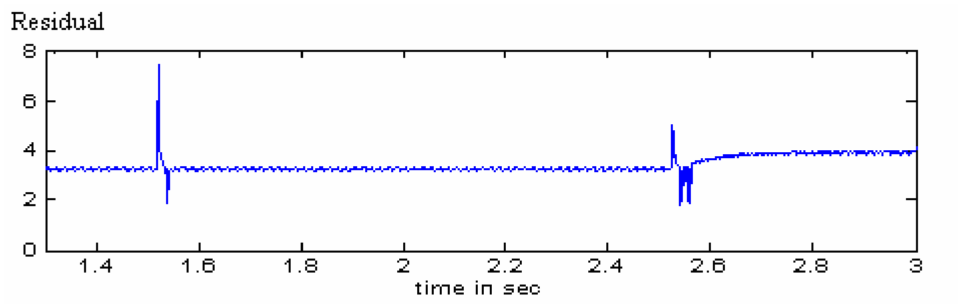

Figure 3.15: Dynamic residual obtained by using input and output measurement values.

Scenario IV

a) In this scenario, while the system is working in real time, abrupt winding and bearing faults are created at the same time (1.5 seconds) by decreasing the stator winding by an amount of 3 % and changing the friction coefficient from zero to 3 (i.e., multiple fault case). Figure 3.16 shows the residual plot for the dynamic case.

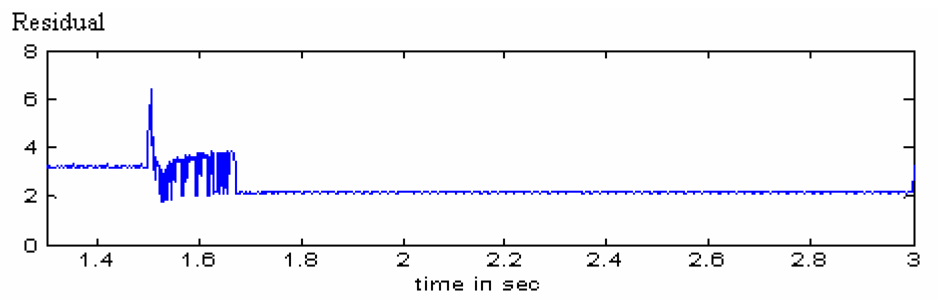

Figure 3.16: Dynamic residual obtained by using input and output measurement values.

b) In this part, again the system's inputs are changed in an acceptable range to see whether the fault monitoring system performance is robust using scenario IV.

- First, before any fault occurred, the motor rated voltage is decreased from 200 to 160 volts. The result obtained is in Figure 3.17.

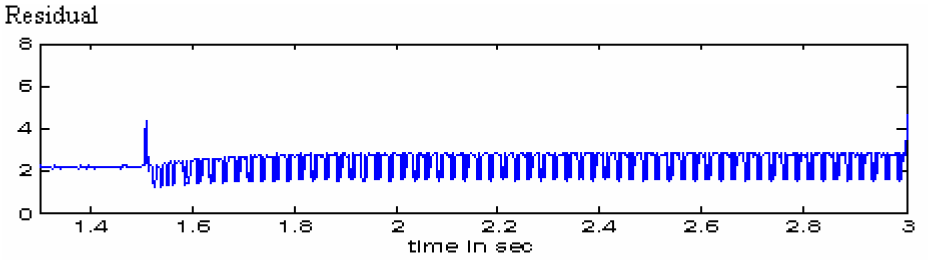

Figure 3.17: Dynamic residual obtained after decreasing the motor's rated voltage.

- Before any fault occurred, the frequency applied to the motor is decreased from 60 to 50 Hz. The result obtained is in Figure 3.18.

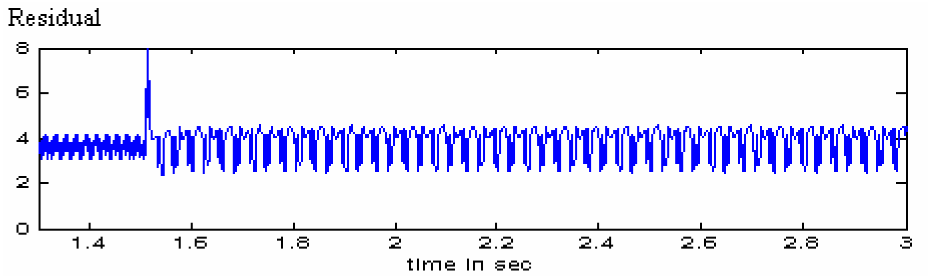

Figure 3.18: Dynamic residual obtained after decreasing the motor's frequency.

- In this case, before any fault occurred, the frequency applied to the motor is decreased from 60 to 50 Hz and the motor rated voltage from 200 to 160 volts. The result obtained is in Figure 3.19.

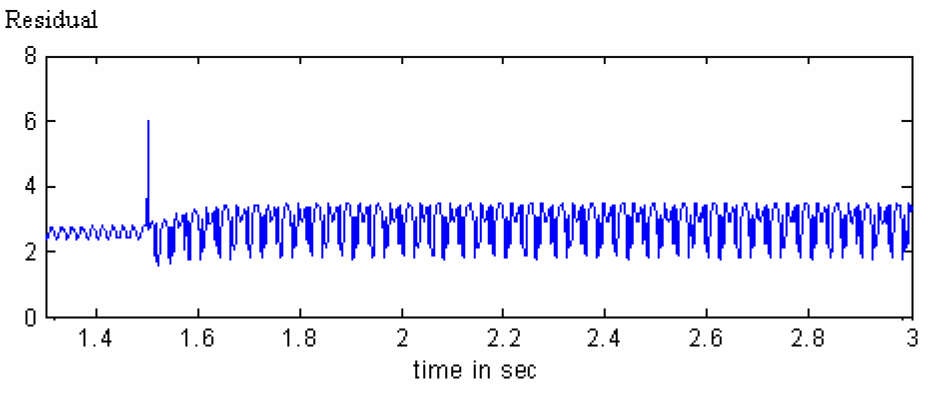

Figure 3.19: Dynamic residual obtained after decreasing the motor's frequency and rated voltage.

Scenario V

a) In this scenario, while the system is working in real time, first, abrupt bearing fault is created at time 1.5 seconds by changing the friction coefficient from zero to 2 and then abrupt winding fault is created at time 2.5 seconds by decreasing the stator winding by an amount of 3 %. Figure 3.20 shows the residual plot for the dynamic case.

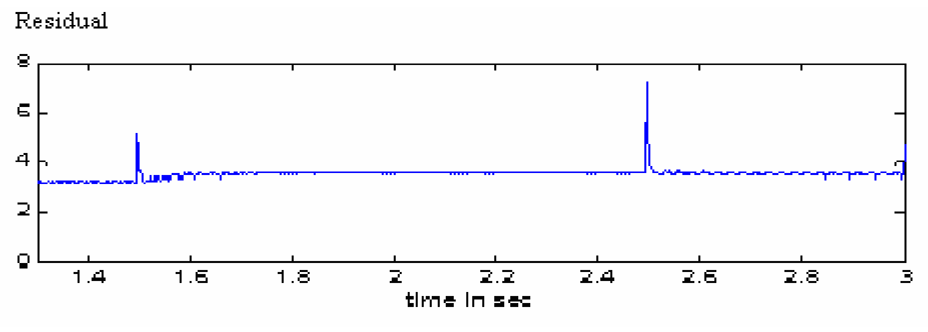

Figure 3.20: Dynamic residual obtained by using input and output measurement values.

b) Before any fault occurred, the frequency applied to the motor is changed from 60 to 58 Hz and the rated motor voltage is changed from 200 to 185 volts. The result obtained is in Figure 3.21.

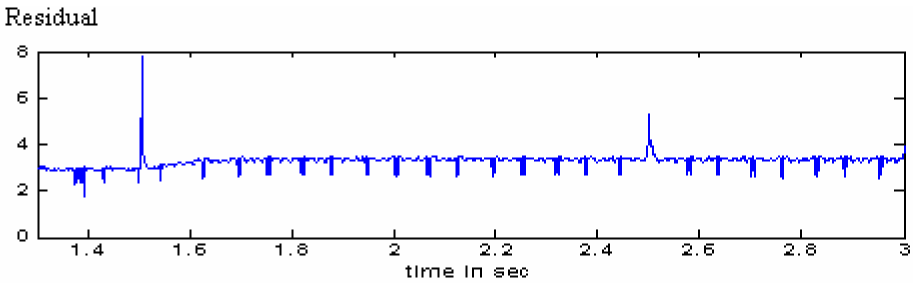

Figure 3.21: Dynamic residual obtained after decreasing the motor's frequency and rated voltage.

In the second case, to construct static and dynamic data matrices, the induction motor inputs are omitted from the previously obtained static and dynamic data matrices. The aim is to see the effect of the input variables on the performance of the FDD algorithm. If the fault monitoring system works well only using output measurement parameters, the cost of the fault monitoring system can be reduced. This means fewer sensors. In this case, to construct static and dynamic data matrices whose sizes are 3000×4 and 3000×24, four measurement variables are used. These variables are three phase input currents and the motor speed.

Table 3.2: Variances of the data matrices a) Static case b) Dynamic case

	PCs	**Eigenvalues**	**Variances %**	**Cumulative Variances %**
a)	1	1.5263	38.158	38.1573
	2	1.4861	37.152	75.3101
	3	0.98759	24.69	100
	4	5.0956e.030	1.2739e.028	100
	PCs	**Eigenvalues**	**Variances %**	**Cumulative Variances %**
b)	1	9.0524	37.718	37.718
	2	8.8719	36.966	74.684
	3	5.9611	24.838	99.522
	4	0.086729	0.36137	99.883
	5	0.024018	0.10008	99.983
	6	0.0018705	0.0077936	99.991
	.	.	.	.
	.	.	.	.
	24	0	0	100

Again, in this case, the motor torque measurement was not included in the data matrices. By applying off-line procedure to the static and dynamic data matrices; means, variances and PCs are calculated. 4 and 24 PCs are obtained by using static and dynamic data matrices, respectively. The results obtained can be seen in Table 3.2. The first three principal components are used again to on-line fault monitoring stage. The same 5 scenarios as mentioned before are implemented.

3.4.2 Fault Scenarios

Scenario I

a) In this scenario, while the system was working in real time, abrupt bearing fault is created by changing the friction coefficient from zero to 1 at time 1.9 seconds. Figure 3.22 shows the dynamic residual plot.

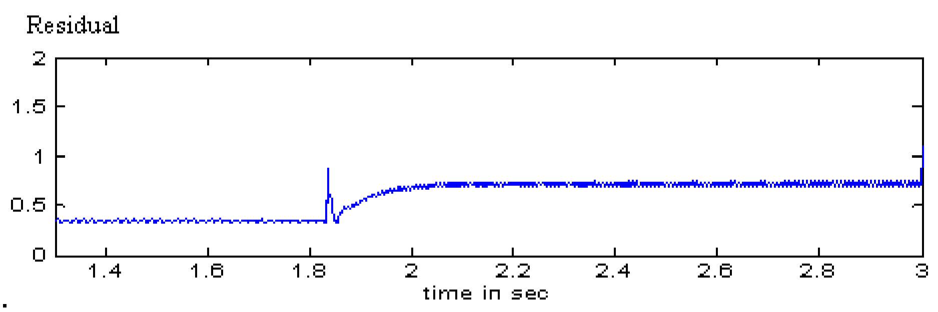

Figure 3.22: Dynamic residual obtained by using output measurement values

b) Applying the scenario above, the motor friction coefficient is increased gradually from 0 to 1 between 1.5 and 1.8 second. The result is in Figure 3.21.

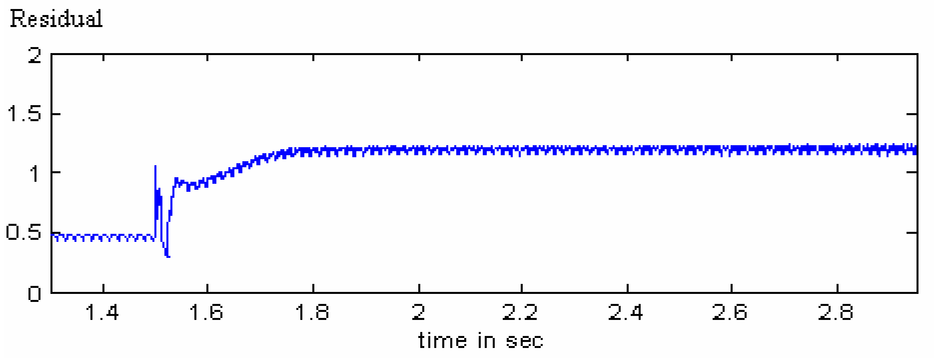

Figure 3.23: Dynamic residual obtained after increasing the motor friction coefficient gradually.

c) In this part, the systems input values are changed in an acceptable range to check the robustness of the fault monitoring system corresponding to abrupt changes.

- First, before any fault occurred, the motor rated voltage is decreased from 200 volts to 160 volts. The result obtained is in Figure 3.24.

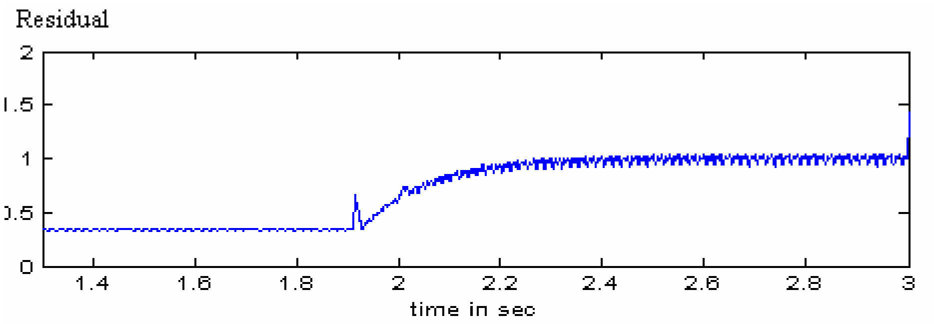

Figure 3.24: Dynamic residual obtained after decreasing the motor's rated voltage.

- Before any fault occurred, the frequency applied to the motor is decreased from 60 to 50 Hz. The result obtained is in Figure 3.25.

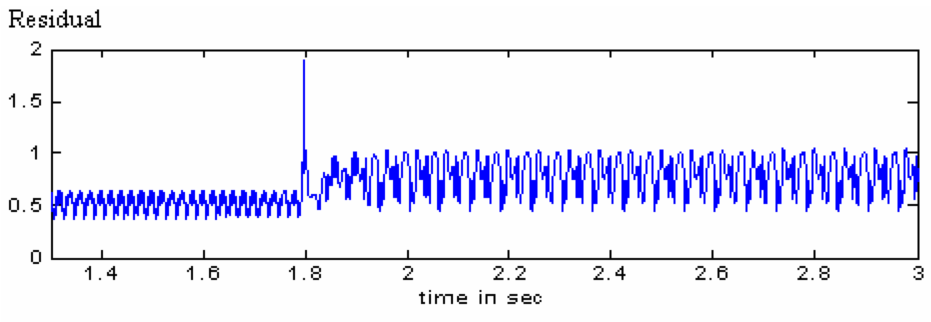

Figure 3.25: Dynamic residual obtained after decreasing the motor's frequency.

- In this case, before any fault occurred, the frequency applied to the motor is decreased from 60 to 50 Hz and the rated motor's voltage from 200 to 160 volts. The result obtained is in Figure 3.26.

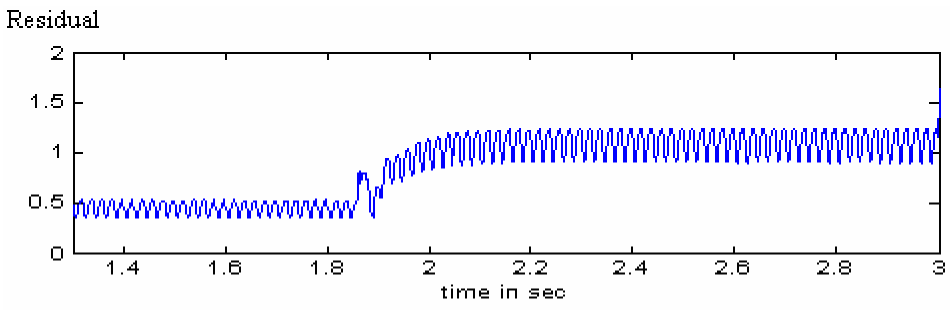

Figure 3.26: Dynamic residual obtained after decreasing the motor's frequency and rated voltage.

Scenario II

a) In this scenario, while the system is working in real time, abrupt winding fault is created by decreasing the stator winding by an amount of 3 %, at time 1.8 seconds. Figure 3.27 shows the residual plot for the dynamic case.

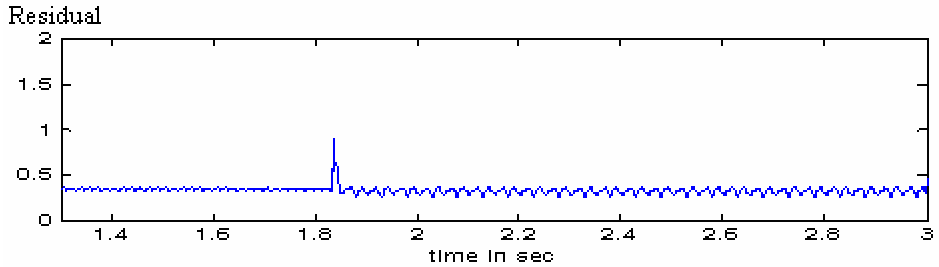

Figure 3.27: Dynamic residual obtained by using output measurement values.

b) Applying the scenario above, the number of motor winding turns is decreased gradually from 100 % to 97 % between 1.2 and 1.5 seconds. The plot is in Figure 3.28.

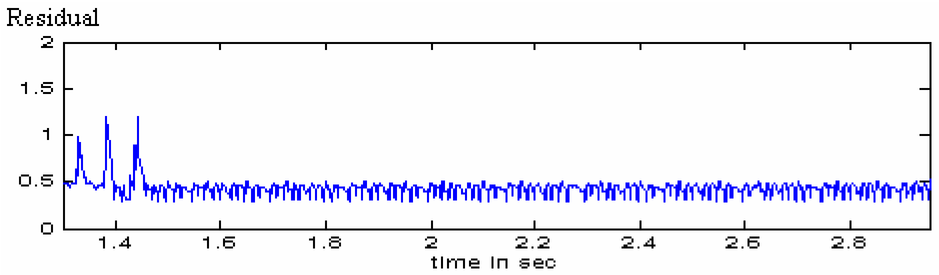

Figure 3.28: Dynamic residual obtained after decreasing the motor's windings turns gradually.

c) In this part, the system input values are changed in an acceptable range to check the robustness of the fault monitoring system corresponding to abrupt changes.

- First, before any fault occurred, the motor's rated voltage is decreased from 200 volts to 160 volts. The result obtained is in Figure 3.29.

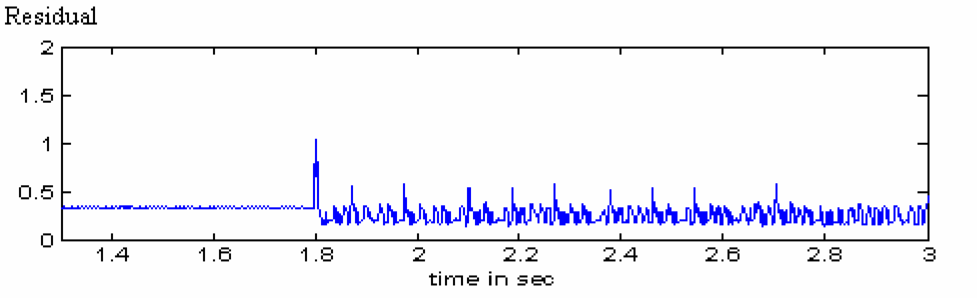

Figure 3.29: Dynamic residual obtained after decreasing the motor's rated voltage.

- Before any fault occurred, the frequency applied to the motor is decreased from 60 Hz to 50 Hz. The result obtained is in Figure 3.30.

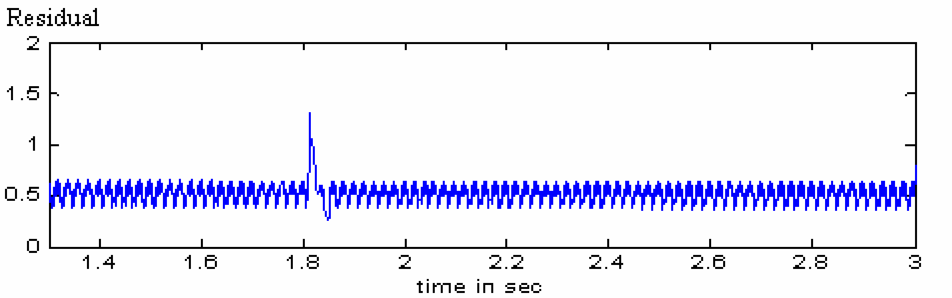

Figure 3.30: Dynamic residual obtained after decreasing the motor's frequency.

- In this case, before any fault occurred, the frequency applied to the motor is decreased from 60 to 50 Hz and rated motor voltage from 200 to 160 volts. The result obtained is in Figure 3.31.

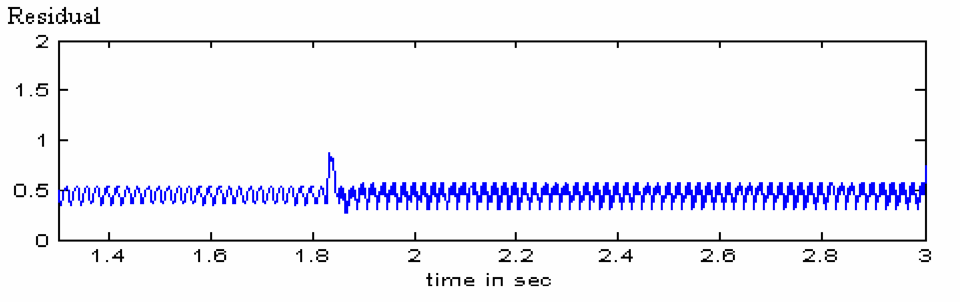

Figure 3.31: Dynamic residual obtained after decreasing the motor's frequency and rated voltage.

Scenario III

In this scenario, while the system is working in real time, first abrupt winding fault is created at time 1.5 seconds by decreasing the stator winding by an amount of 3 % and then abrupt bearing fault is created at time 2.5 seconds by changing the friction coefficient from zero to 3. Figure 3.32 shows the residual plot for the dynamic case.

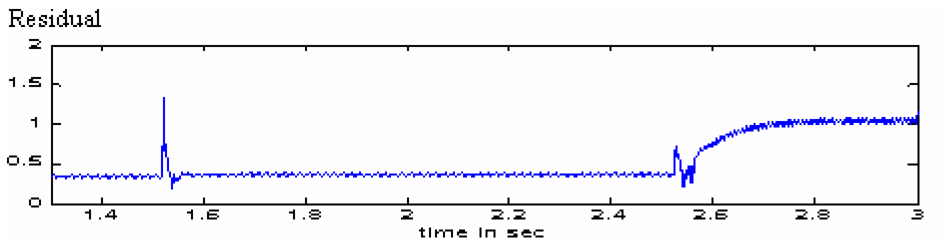

Figure 3.32: Dynamic residual obtained by using output measurement values.

Scenario IV

a) In this scenario, while the system is working in real time, abrupt winding and bearing faults are created at the same time (1.5 seconds) by decreasing the stator winding by an amount of 3 % and changing the friction coefficient from zero to 1. Figure 3.33 shows the residual plot for the dynamic case.

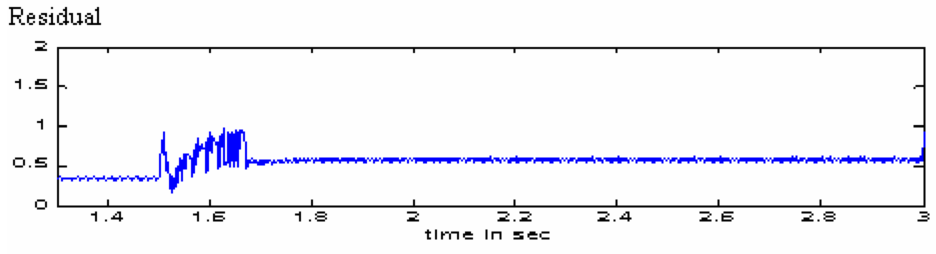

Figure 3.33: Dynamic residual obtained by using output measurement values.

b) In this part, again, the motor inputs are changed in an acceptable range to see the robustness performance of the fault monitoring system on scenario IV.

- First, before any fault occurred, the motor's rated voltage is decreased from 200 to 160 volts. The result obtained is in Figure 3.34.

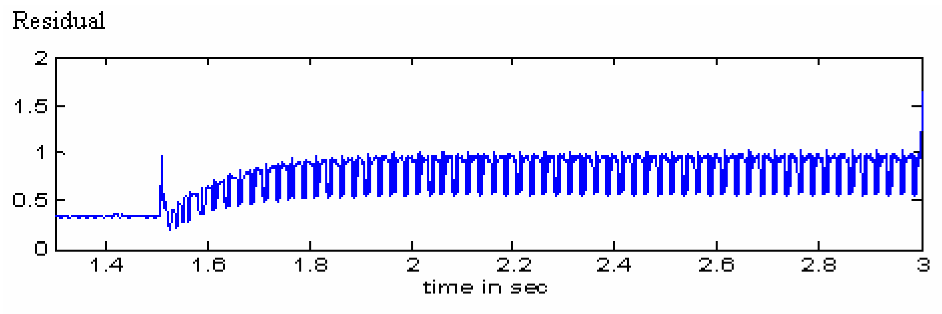

Figure 3.34: Dynamic residual obtained after decreasing the motor's rated voltage.

- In this case, before any fault occurred, the frequency applied to the motor is decreased from 60 to 50 Hz and the motor rated voltage is decreased from 200 to 160 volts. The result obtained is in Figure 3.35.

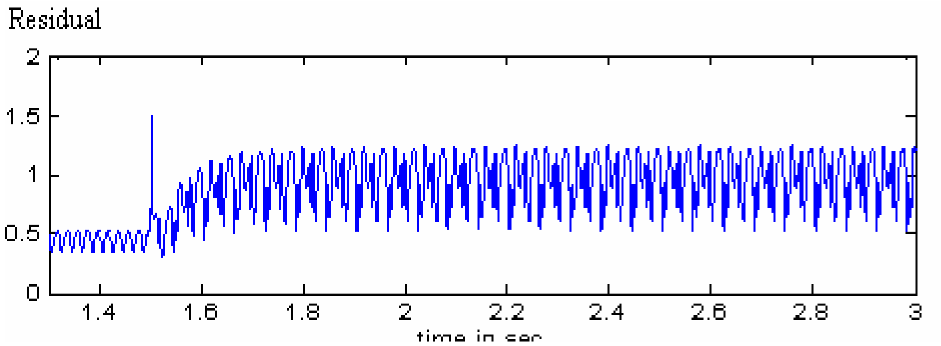

Figure 3.35: Dynamic residual obtained after decreasing the motor's frequency and rated voltage.

Scenario V

a) In this scenario while the system is working in real time, first, abrupt bearing fault is created at time 1.5 seconds by changing the friction coefficient from zero to 2 and then winding fault is created at time 2.5 seconds by decreasing the stator winding by an amount of 3 %. Figure 3.36 shows the residual plot for the dynamic case.

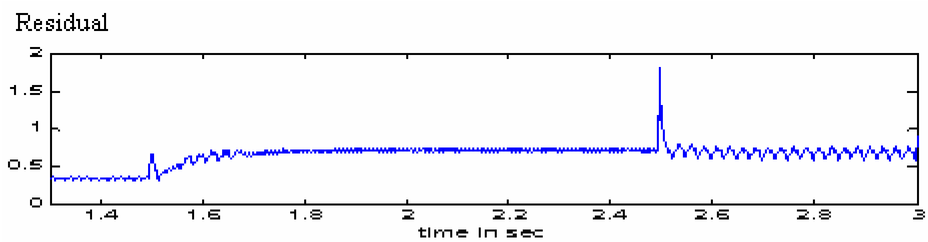

Figure 3.36: Dynamic residual obtained by using output measurement values.

b) In this case, the motor input parameters are changed in an acceptable range. Before any fault occurred, the frequency applied to the motor is decreased from 60 to 58 Hz and rated motor voltage is decreased from 200 to 185 volts. The result obtained is in Figure 3.37.

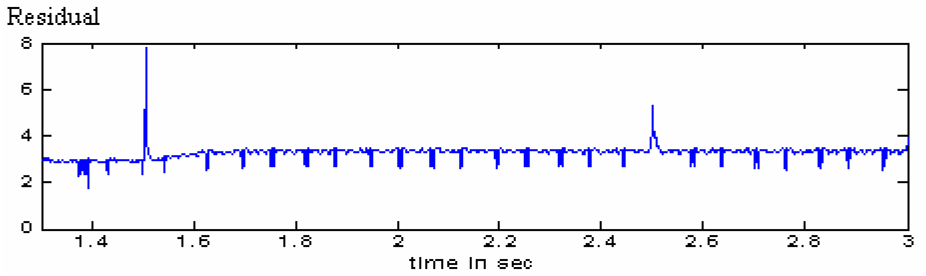

Figure 3.37: Dynamic residual obtained after decreasing the motor's frequency and rated voltage.

3.5 System Identification Based FDD in an Induction Motor

The induction motor model and parameter names used were given in chapter 3.3.1.

3.5.1 Simulation Results

In the induction motor, the system states are motor speed, rotor currents and fluxes, which are measurable. The problem is to determine the unknown fault vector $f_c(t) = [xM(t) \quad R_s(t) \quad T_{damp}(t) \quad x_{LS}(t)]^T$ using system fault model. Since the system states are directly measurable, the system states measured are recorded between an initial time (i.e., residuals exceed a predefined threshold value at that time) t_{in} and final time t_f. Then, to determine the unknown fault vector $f_c(t)$, the nonlinear optimization problem given in equation 2.18 was solved between time t_{in} and t_f by using Matlab optimization and simulink toolboxes.

3.5.2 Fault Scenarios

The overall FDD architecture is tested based on three different (i.e., single and multiple faults) scenarios by starting the simulation program from different initial points. These points are generated by a random uniform distribution with a zero mean and unity variance. An initial point can be chosen as $f_c(t_{in}) = [0.88 \quad 0.92 \quad 0.52 \quad 0.98]^T$ as an example.

Scenario I

In this scenario, while the system is working in real time, 10 % incipient bearing fault is created by changing the motor friction coefficient gradually from 0 to 1 in 2.35 seconds. The unknown fault vector $f_c(t) = [xM(t) \quad R_s(t) \quad T_{damp}(t) \quad x_{LS}(t)]^T$ is calculated where *t* is in between 2.38 and 2.58 seconds. Second row of the Table 3.3 denotes the fault vector variation in the selected time range i.e. 2.38 < t < 2.58.

Table 3.3: Single fault case

Faults	xM	R_s	$Tdamp$	x_{LS}
Created	0%	0%	10%	0%
Calculated	0-0.03%	0-0.02%	8-12%	0-0.001%

Scenario II

In this scenario, while the system was working in real time, abrupt winding fault is created by decreasing the stator winding by an amount of 10 % in 2.4 seconds. The unknown fault vector $f_c(t) = [xM(t) \quad R_s(t) \quad T_{damp}(t) \quad x_{LS}(t)]^T$ is calculated where *t* is in between 2.38 and 2.58 seconds. Second row of the Table 3.4 denotes the fault vector variation in the selected time range i.e. 2.45 < t < 2.65.

Table 3.4: Single fault case

Faults	xM	R_s	$Tdamp$	x_{LS}
Created	10%	10%	0%	10%
Calculated	6-11.1%	7-12%	0.0%	7-12%

Scenario III

In this scenario, while the system is working in real time, abrupt winding and bearing faults are created at the same time by decreasing the stator winding by an amount of 5 % and changing the friction coefficient from zero to 5 (i.e., multiple fault case). The unknown fault vector $f_c(t) = [xM(t) \;\; R_s(t) \;\; T_{damp}(t) \;\; x_{LS}(t)]^T$ is calculated where *t* is in between 2.38 and 2.58 seconds. Second row of the Table 3.5 denotes the fault vector variation in the selected time range i.e. 2.45 < t < 2.65.

Table 3.5: Multiple fault case

Faults	xM	R_s	$Tdamp$	x_{LS}
Created	5%	5%	50%	5%
Calculated	2.8-8%	3.1-6%	35-53.86%	3-8.86%

The simulation results showed that the fault priorities calculated are consistent with the ones created for all possible single and multiple fault cases. This is also observed in various other applications that are not shown here.

3.6 Inverse Model Based FDD in a Gas Pipeline System

The idea of determining failure possibility using inverse model was applied to a simple gas pipeline system given Figure 3.38. In this model, there are 5 nodes and 6 vertices. Here, Q indicates consumer nodes.

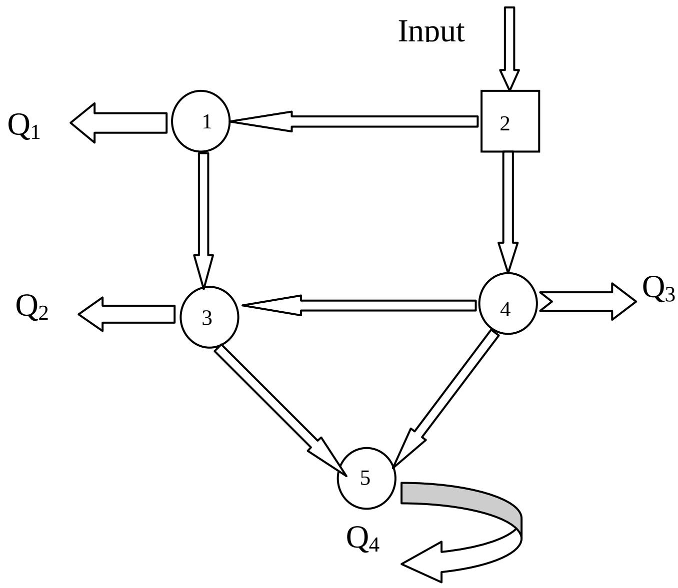

Figure 3.38: Gas pipeline system

The input-output relationship is as.

$$q = a\sqrt{\frac{(P_i^2 - P_j^2)}{L}} \qquad (3\text{-}12)$$

where, P_i (psi), P_j (psi) and *L*(ft) are input-output pressures and line's length, q_{ij} j (MSCF/G) is the flow from *i* to *j* and a is a constant depend on gas' and lines' physical structure (i.e., $a = 3.22\frac{T_b}{P_b}\left[\frac{d^5}{G\,T\;f\;Z}\right]^{0.5}$). T_b and P_b are base temperature and pressure, *d* pipe radius (inches), *G* gas gravity , *T* ambient temperature, *f* friction factor, *Z* compressibility factor.

At the nodes Kirchoff `s node equations are valid.

$$q^i{}_{input} - q^i{}_{output} = 0 \qquad i=1,\dots,5 \qquad (3\text{-}13)$$

where q_{input} and q_{output} denotes input and output flows of the ith node.

3.6.1 Fault Model

In order to be able to determine faults by using the inverse model we need the system's fault model. For this purpose we added hypothetical nodes between any two nodes at every branch and modeled the possible leaks or faults in the associated branch as if there is hypothetical consumption by the so-called hypothetical node. The fault model is in Figure 3.39,

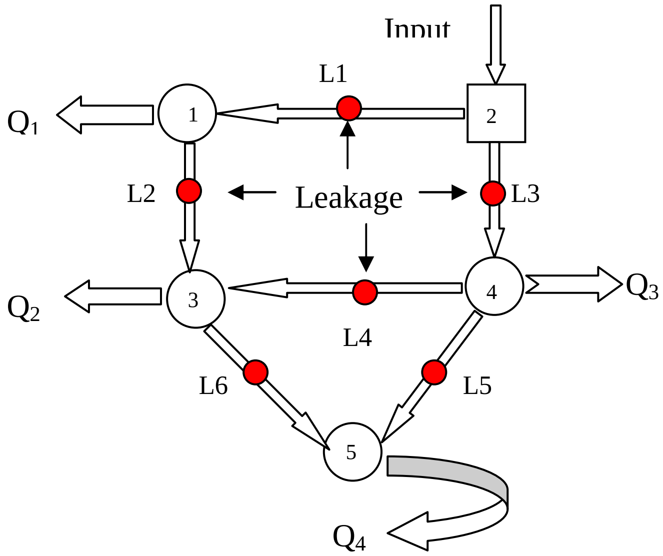

Figure 3.39: Gas pipeline system's fault model

where L_i denotes ith pipe leakage. In the fault model there are 11 node and 12 flow equations given as equations (3-12) and (3-13). If we create 6 leakages at the same time we cannot find faults by using inverse model because we have 23 equations and 24 unknowns. To determine the unknowns we need more than 23 equations. Hence we did not create more than 5 faults at the same time.

3.6.2 Simulation Results

In order to check the performance of the proposed FDD algorithm, some leakages are created at the hypothetical nodes of the gas pipeline system. Then the system outputs and inputs are measured at the static nodes. By using these measurement values unknown pipe leakages are calculated by using the system fault model. To solve the nonlinear equations Newton's method is used. The results obtained are in tables 3.6 and 3.7.

Table 3.6: Single fault cases

	Scenario 1		***Scenario II***		***Scenario III***		***Scenario IV***	
	Fault %		***Fault %***		***Fault %***		***Fault %***	
	Created	***Calculated***	***Created***	***Calculated***	***Created***	***Calculated***	***Created***	***Calculated***
Pipe1	0	0	0	0.16	0	0	0	0
Pipe2	0	0	0	0	0	0	0	0
Pipe3	0	0	0	1.08	30	28.1	0	0
Pipe4	0	0	0	0	0	0	0	0
Pipe5	0	0	10	10.9	0	0	0	0
Pipe6	0	0	0	0.01	0	0	40	39

Table 3.7: Multiple fault cases

	Scenario 1		***Scenario II***		***Scenario III***		***Scenario IV***	
	Fault %		***Fault %***		***Fault %***		***Fault %***	
	Created	***Calculated***	***Created***	***Calculated***	***Created***	***Calculated***	***Created***	***Calculated***
Pipe1	0	0.11	0	0	0	0	0	3.69
Pipe2	0	0	10	11.2	0	0	10	7.9
Pipe3	0	0	30	30.9	10	12.1	20	20.4
Pipe4	0	0	20	10	20	18.3	10	14
Pipe5	50	49.9	0	0	10	19.6	30	25.3
Pipe6	30	30.2	0	0	10	13.3	40	47

If we look at the simulation results, it can be said that the order of magnitude in faults satisfies the norms completely. This is also observed in various other applications that are not shown here. We have to emphasize that the place of the leakages in any pipe affects the results limitedly. The reason for this is that the resistance the pipeline against flow is not at a single node but it is distributed through the branch.

3.7 Conclusions

All results obtained by using system identification and inverse model based FDD are satisfactory for fault priority, but systems' mathematical models must be known well. Besides, all system states should be measurable to use system identification based FDD approach. Moreover, one can face the convergence problem when using this approach because the optimization algorithms converge very slowly. If we have a nonlinear static system, the inverse model can also be used to solve an FDD problem.

When we investigated the results obtained by utilizing PCA, the best ones were obtained in the dynamic cases especially when four measurement variables used. For such systems it won't be necessary to use all of the input and output data in the FDD utilizing PCA; it might be enough to use only output measurements which will provide enough information to detect certain types of faults. Moreover, it can also be argued that the FDD utilizing PCA can also provide satisfactory results for certain nonlinear systems and is robust if the system's inputs variability stay in a feasible range. If the induction motor works in feasible range the inputs and process noises do not produce a fault alarm. To isolate faults in single fault cases an adaptive threshold can be used. However, in multiple fault cases threshold based fault isolation techniques are not successful enough. Hence, intelligent residual classification techniques should be used.

CHAPTER 4

FUZZY DISCRETE EVENT SYSTEMS (FDES)

4.1 Importance of the FDES or FDEDS

Most of the research about the DES approach has focused on some uncertainty problems and complex information and systems in practice [8], [20]. Conventional DES are used to model systems that cannot be described by differential equations or difference equations, but must be described by sequences of events recording significant qualitative changes in the state of the system. Two basic practices for this purpose are automata theory and Petri nets. Some modeling practices using the above theories can be found in [7], [8], [9] and [26]. One can find many different modeling strategies for DES in addition to the above, such as Min-max or dioid algebraic models, communicating sequential process models, queering network models, generalized semi-Markov process models [12], etc. Although conventional DES models have been applied in many engineering fields, they may not be adequate for some other fields. This is especially true when we consider fault diagnosis applications for nonlinear complex systems, in which a component health status is somewhat uncertain (e.g., degree of fault) and vague even in a deterministic sense [22], [23]. Furthermore, the definition of diagnosis as a set of faulty components could be too restrictive since users may want to identify different levels of faults. Usually, the state (healthy or unhealthy) of components, obtained from instrument measurements, an expert experiences, or analysis using probabilistic schemes cannot be determined accurately. The research on the diagnostic problem for such systems with fuzziness is interesting and important. Human observation and judgment play a significant role in describing states, which are usually not crisp. For example, it is vague when an actuator's condition is said to be "good". Furthermore, the transition from one state to another is also vague. It is hard to say exactly how an actuator's condition has changed from "good" to "bad". Sometimes one may need to model systems that cannot be modeled by the current DES modeling methods due to the vagueness in the definitions of the states and/or events. In order to overcome these difficulties, the concepts of fuzzy state and fuzzy event can be used [11], [10]. The construction of the FDES framework is so simple. It also allows one to build each component model separately. Lin and Ying [13], [14] initiated the study of FDES by combining fuzzy set theory with DES to solve problems which are not possible to be solved by conventional DES. They then applied their results about FDES to HIV/AIDS treatment planning problem [31]. In this chapter, a new structure called dynamic FDES (i.e., FDEDS) is first introduced by us.

4.2 Fuzzy Discrete Event Systems

State, event and event transition function values are crisp in a DES. Before an FDES structure is modeled, let's recall a model for a DES structure first.

Definition 4.1**:** Discrete event systems can be modeled by a five-tuple $G = (Q, \Sigma, f, h, q_0)$ [12], where

- Q is the set of states,
- Σ is the set of events containing detectable (i.e., an event is ***detectable*** if it produces a measurable change in the output) and undetectable events, which are generally fired asynchronously,
- $f : Q \text{ x } \Sigma \rightarrow Q$ is the state transition function,
- $h : Q \times \Sigma \rightarrow \hat{\Sigma}$ is the output equation, where $\hat{\Sigma}$ is the set of detectable events, $\hat{\Sigma} \subseteq \Sigma$

- q_0 is a *1Xn* (*n*: the number of states) initial state vector, whose elements are zero or 1.

Figure 4.1: DES model to an actuator condition.

Example 4.1: Let us consider the use of a crisp DES in fault diagnosis applications. For example, if one classifies an actuator's condition as good (G, works properly) and poor (P, does not work properly), then the corresponding crisp DES is shown in Figure 4.1. In the figure α, β and σ denote the events describing whether the actuator is deteriorating and intact, respectively. For the time being, we assume that both of the states and events are crisp. In this model

$$Q = \{G, P\};$$
$$\Sigma = \{\alpha, \beta, \sigma\};$$
$$\hat{\Sigma} = \{\alpha, \beta, \sigma\};$$
$$h = \hat{\Sigma};$$
$$q_0 = [1 \quad 0]$$

There are many methods to construct state transition function. For a special case as in [13], [14], the transition function can be represented by matrices $f(e_i)$, called the state transition matrices, whose elements are either zero or one. The system next state (assuming the event e_i has occurred) can be calculated as:

$$q_{k+1} = q_k \circ f(e_i) \qquad (4\text{-}1)$$

where $\circ$ shows max-product operation and the current states of the system is represented by a vector *q_k=[q_1…q_n]* to indicate that the system is currently in state *k*. For example, the events (in this model only one event occurs at an instant of time) and their transitions are represented by (see figure 4.1).

$$f(\alpha) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad f(\beta) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad f(\sigma) = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

and the next state is calculated (i.e., initially the system is in place *G* and event α occurs at a time *t*) as:

$$q \circ f(\alpha) = [1 \quad 0] \circ \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} = [\max(0,\ 0) \quad \max(1,\ 0)] = [0 \quad 1]$$

Hence we conclude that the crisp DES will be in state P after the occurrence of the event α. Since we are interested in fault diagnosis, the set of states will be restricted to $(Z_2)^n$, where $Z_2 = \{0, \quad 1\}$, in the remaining part of this study.

■

In order to generalize a DES structure into an FDES structure, the concepts of ***fuzzy state*** and ***event*** are proposed in this study. We combine fuzzy set theory with (crisp) DES structure in which events and states have crisp values. In FDES structure, events and state transition functions are fuzzy vectors whose components take values between zero and 1. All events in FDES occur continuously at the same time with different membership degrees (i.e., events firing at the same time with different degrees); hence the system can be in many places (states) at a given instant. This is similar to spanning a vector space by its bases as if among all the unaccountably infinite number of states a countable number of states are selected. By defining that the system may be in more than one state at a time, to be in the states that were not chosen as a basis state is made possible.

Example 4.2: In order to make an analogy, take, for example, the system in Example 4.1. Let the actuator's condition at a particular time *t,* be simultaneously belong to "Good" with a membership 0.8 and "Poor" with a membership 0.2, which is represented by the vector

$$\overline{q} = [0.8 \quad 0.2].$$

Similar to the DES approach, we allow the elements of the transition matrix to take values between zero and 1, so that fuzzy events can be represented. As stated earlier, in the FDES approach all events occur with a membership degree at an instant of time. For example, suppose that the possibility for the system to evolve from state *G* to *P*, *G* to *G* and *P* to *P,* respectively, at the same fixed time *t* are $\alpha_{GP} = 0.6$, $\beta_{GG} = 0.3$ and $\sigma_{PP} = 0.1$. In this structure

$$\begin{aligned}
&\overline{Q}(t) = \left\{\mu_G(t), \mu_P(t)\right\} \Rightarrow \overline{Q} = \bigcup_{t \geq 0} \overline{Q}(t) \\
&\overline{\Sigma}(t) = \left\{\mu_\alpha(t), \mu_\beta(t), \mu_\sigma(t)\right\} \Rightarrow \overline{\Sigma} = \bigcup_{t \geq 0} \overline{\Sigma}(t) \\
&\hat{\Sigma}(t) = \left\{\mu_\alpha(t), \mu_\beta(t), \mu_\sigma(t)\right\}, \ \hat{\Sigma}(t) \subseteq \overline{\Sigma}(t) \\
&\overline{h} : \overline{Q}(t) \times \overline{\Sigma}(t) \to \hat{\Sigma}(t) \\
&\overline{q}_0 = \left[\mu_G(t_{in}), \mu_P(t_{in})\right] \quad t_{in} : \text{initial time}
\end{aligned}$$

where μ shows membership degree of the places. The events (in this model all events occur at the same time with different membership degrees) and their transitions for a special case as in DES (the state transition function is actually a matrix whose components take values between zero and 1) are represented by

$$\bar{f}(\alpha) = \begin{bmatrix} 0 & 0.3 \\ 0 & 0 \end{bmatrix}, \quad \bar{f}(\beta) = \begin{bmatrix} 0.6 & 0 \\ 0 & 0 \end{bmatrix}, \quad \bar{f}(\sigma) = \begin{bmatrix} 0 & 0 \\ 0 & 0.1 \end{bmatrix}$$

■

The states $\bar{q}_i$ have different membership degrees $\mu_{\bar{q}_i}(t)$ at an instant of time t. Thus, the pairs of $(\bar{q}_i, \mu_{\bar{q}_i}(t))$ can be proposed as a generalized states. Furthermore, to understand a system's state directly it is enough to look at the membership degrees of the system's states. In short, the state vector of a system in FDES can be given as $\bar{q}(t) = (\mu_{\bar{q}_1}(t), \ldots, \mu_{\bar{q}_n}(t)) \in \overline{Q}$. where $\mu_{\bar{q}_i}(t)$ is the membership function of the state $\bar{q}_i$ at an instant of time t. In particular we could have defined $\overline{Q}$ and $\overline{\Sigma}$ in the previous example as $\overline{Q} = [0.\ \ 1] \times [0.\ \ 1]$, $\overline{\Sigma} = [0.\ \ 1] \times [0.\ \ 1] \times [0.\ \ 1]$.

If we reformulate the crisp DES into fuzzy DES we may write the definition below for FDES.

***Definition 4.2*:** Fuzzy discrete event systems is modeled by a five-tuple $G = (\overline{Q}, \overline{\Sigma}, \bar{f}, \bar{h}, \bar{q}_0)$ where

- $\overline{Q}$ is the set of state space such that $\bar{q}(t) \in \overline{Q} \subseteq R^n$,where *n* is the number of states,
- $\overline{\Sigma}$ is the set of event space such that $\bar{e}(t) \in \overline{\Sigma} \subseteq R^m$, where *m* is the number of events,
- $\bar{f} : \overline{Q}(t) \text{ x } \overline{\Sigma}(t) \to \overline{Q}(t) \subset \overline{Q}$ is the state transition function ,
- $\bar{h} : \overline{Q}(t) \times \overline{\Sigma}(t) \to \hat{\Sigma}$ is the output equation, where $\hat{\Sigma}$ is a set of detectable events, $\hat{\Sigma} \subseteq \overline{\Sigma}$
- $\bar{q}_0$ is the initial state vector showing the initial membership degrees of a system states.

The FDES can be considered as a fuzzy expert system. The structure of an expert system consists of tree major blocks: a knowledge base, an inference engine and a user interface. The knowledge base is composed of the knowledge represented in the form of rules. IF-THEN rules are the most popular formalism for representing knowledge. An inference engine can use the knowledge in the knowledge base to perform reasoning to obtain answers for user` requests. A user interface provides communication between user and the system. It is important for an expert system to handle inaccuracy in order for it to be a useful tool. Most methods of handling inaccuracy are probability based. This concept has been applied, in several different forms to the handling of inaccuracy in expert systems. These kinds of methods are reasonably effective in specific cases. However, it is interesting that experts often do not think in terms of probability values but in terms of such expressions as “much”, “usually”, “always” and so on. This motivates the use of fuzzy logic in traditional expert systems and forms called fuzzy expert systems. Hence fuzzy set theory may be more natural than probability theory. The application of fuzzy sets and possibility theory to rule-based expert systems has been developed along two lines: (1) generalization of the certainty factor by using linguistic certainty values, (e.g., Possible, ALMOST-Impossible) in addition to the conventional numerical certainty values, and (2) the handling of vague predicates in the expression of expert rules or available information. Fuzzy IF-THEN rules include the preconditions and consequents involve linguistic variables. The general form of the rule in the case of multi-input-single-output system (MISO) is:

$$\mathbf{R^i: IF}\ x \text{ is } A^i_1, \ldots, \mathbf{AND}\ y \text{ is } B^i_1\ \mathbf{THEN}\ z = C^i_1,\ i=1,2,\ldots,n \qquad (4\text{-}2)$$

where $x, \ldots, y$, and z are the input and output variables, respectively, and $A^i_j, \ldots, B^i_j$, and C^i_j are the linguistic labels of fuzzy sets in the universes $U, \ldots, V$, and W $(x \in U,\ y \in V, z \in W)$. The consequent can be represented as a function of the measurable system variables $x, \ldots, y$, that is

$$\mathbf{R^i: IF}\ x \text{ is } A_i, \ldots, \mathbf{AND}\ y \text{ is } B_i\ \mathbf{THEN}\ z = f(x, \ldots, y),\ i=1, 2, \ldots, n \qquad (4\text{-}3)$$

where *f(x,…, y)* is a function of input variables *x,…, y*. The inference, evaluation of fuzzy rules to produce an output for each rule, is the kernel of a fuzzy expert system. A fuzzy rule is a fuzzy relation, which is expressed as a fuzzy implication. According to the compositional rule of inference, conclusion can be obtained by taking the composition of related fuzzy set and fuzzy implication. There are many types of compositional operators but mostly used are: max-min operation, max-product operation, max bounded product operation and max drastic product operation. The compositional operators produce inference results. Max-min and max-product compositional operators are the most commonly and frequently used operators because of their computational simplicity. There are many distinct fuzzy implication functions described in the literature. For example: minimum, product, bounded product, drastic product, arithmetic rule, Zadeh max-min rule etc.

$$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_{\mathbf{B}}(x)) \tag{4-4}$$

$$\mu_{AB}(x) = \mu_A(x).\mu_{\mathbf{B}}(x) \tag{4-5}$$

Among the various fuzzy implications Mamdani's fuzzy implication method (minimum operation) associated with max-min composition is the most frequently used one. It is possible to write infinitely many rules using combination of implication functions.

In the most general case, the fuzzy rule base has the form of a multi-input-multi- output (MIMO) system:

$$R = \left\{ R^1 MIMO, \quad R^2 MIMO \quad ,..., \quad R^n MIMO \right\} \tag{4-6}$$

where $R^i MIMO$ represents the ith rule:

R^i: IF *x* is $A^i_1,\ldots,$ **AND** *y* is B^i_1 **THEN** (z_1 is $C^i_{1},\ldots,z_q$ is C^i_q), $i=1,\ldots,n$ (4-7) where, z_q indicates the q^{th} output variable related to the input variable and C_q^i denotes the linguistic values of the linguistic variable z_q . One can construct MIMO rule base as a combination of MISO rules. The most commonly employed fuzzy expert systems are Mamdani and Sugeno type expert systems. The Sugeno type expert system is computationally effective and works well with the optimization and adaptive techniques. The Mamdani type is widely accepted for capturing expert knowledge. It allows us to describe the expertise in more intuitive, more human-like manner. Hence, the Mamdani type expert system is preferred. In FDES structure, the inputs applied to the fuzzy expert system are the fuzzified states $\overline{q}_i(t)$ and events $\overline{e}_i(t)$ occurring continuously all the time with different membership degrees. The general form of the rule base for FDES is:

R^i **: IF** $\overline{e}_t,...,\overline{e}_{t-N}$ is A^i_1 ,..., A_N^i **AND** $\overline{q}_t,...,\overline{q}_{t-N}$ is B^i_1 ,…, B_N^i **THEN** ($\overline{q}_1$ is C^i_1,…, $\overline{q}_N$ is C_N^i), $i = 1,...,n$ (4-8)

where n denotes the number of rules, N is integer related to the model order and shows how much of the past history is considered and, $e_t…e_{t-N}$ and $q_t,…,q_{t-N}$, denote the past inputs, $\overline{e}_t,...,\overline{e}_{t-N}$ and $\overline{q}_t,...,\overline{q}_{t-N}$ show detectable events and states $(\overline{e}_t,...\overline{e}_{t-N}) \in U,\ U = U_1 \times ... \times U_n \subseteq R^N$, $(\overline{q}_t,...\overline{q}_{t-N}) \in V,\ V = V_1 \times ... \times V_N \subseteq R^N$, $A_1^i,..., A_N^i$, $B_1^i,..., B_N^i$ B and $C_1^i,..., C_N^i$ are linguistic values (labels) represented as fuzzy subsets of the respective universes of discourse U, V and W.

4.3 Fuzzy Discrete Event Dynamical Systems

In the special case, if the events $\overline{e}_i(t)$, states $\overline{q}_i(t)$ and change in states $\Delta\overline{q}_i(t)$ are used as a rule antecedent and consequent parts, a rule is as.

R^i **: IF** $\bar{e}_t, \ldots, \bar{e}_{t-N}$ is $A^i_1, \ldots, A^i_N$ **AND** $\bar{q}_t, \ldots, \bar{q}_{t-N}$ is $B^i_1, \ldots, B^i_N$ **THEN** ($\Delta\bar{q}_1$ is $C^i_1, \ldots, \Delta\bar{q}_N$ is C^i_N), $i = 1, \ldots, n$ (4-9)

If the rules are constructed as in equation (4-9), a dynamical system is obtained as shown below.

$$\dot{q}(t) = \tilde{f}(\bar{q}(t), \bar{e}(t)) \tag{4-10}$$

where $\bar{q}(t)$ and $\bar{e}(t)$ are the state and the event (input) vectors at time t. The fuzzifier performs the function of fuzzification, which is a subjective evaluation to transform incoming crisp input data (q, e) into evaluation of a subjective value. Hence, it can be defined as a mapping from an observed input space to labels of fuzzy sets in a specified input universe of discourse. The implication and compositional operator can be chosen in many different ways in this rule structure. For the special case, if the product implication is used, the p^{th} rule activation degree related to the C^i_l linguistic variable is written as:

$$\alpha_l = C_l^p \left(\prod_{i=1}^{n} \mu_{A_i}^P (\bar{e}_i) \prod_{k=1}^{m} \mu_{B_i}^P (\bar{q}_k) \right), \qquad (l = 1, \ldots, m) \tag{4-11}$$

where $\mu_{A_i}^P (\bar{e}_i)$ and $\mu_{B_i}^P (\bar{q}_k)$ shows the membership function related to the linguistic values $A^i_1, \ldots, A^i_n$ and $B^i_1, \ldots, B^i_m$, respectively. The membership functions of the consequents are clipped at the level of the antecedent truth using the below function,

$$\tilde{\mu}_{C^l}(t) = \begin{cases} \mu_C^l(t) & \text{if} \quad \alpha_l \geq \mu_C^l(t) \\ \alpha_l & \text{if} \quad \alpha_l < \mu_C^l(t) \\ 0 & \text{if } t \text{ is outside of the mebership function domain} \end{cases} \tag{4-12}$$

where $\mu_C{}^l(t)$ shows membership functions related to the output linguistic variable C_l^p. The combined memberships of the output functions can be obtained using maximum composition operation:

$$\mu_{C_l}^P(\Delta u) = (\max \tilde{\mu}_C{}^l((\prod_{i=1}^{n} \mu_{A_i}^P(\bar{e}_i) \prod_{k=1}^{m} \mu_{B_i}^P(q_k)), \ (l = 1,\dots, m) \quad \text{(4-13)}$$

Defuzzification is a conversion of fuzzy set produced by inference engine into a crisp value. This process is necessary because a crisp state value is required. Unfortunately, there is no systematic procedure for choosing a defuzzification strategy. Two commonly used methods of defuzzification are the center of area (COA) and the mean of maximum (MOM) methods. If the universe of discourse is continuous, then the COA strategy generates an output as:

$$\Delta q^l = \frac{\int_{s=0}^{T} \max \tilde{\mu}_C{}^l((\prod_{i=1}^{n} \mu_{A_i}^{P\ s}(\bar{e}_i) \prod_{k=1}^{m} \mu_{B_i}^{P\ s}(\bar{q}_k))}{\int_{s=0}^{T} ((\prod_{i=1}^{n} \mu_{A_i}^{P\ s}(\bar{e}_i) \prod_{k=1}^{m} \mu_{B_i}^{P\ s}(\bar{q}_k))} \quad (l = 1,\dots, m) \quad \text{(4-14)}$$

The explicit nonlinear equation given in 4-14 shows that the fuzzy expert system is actually a nonlinear dynamical system based on events and states. We can define the nonlinear function *f* as:

$$\tilde{f}(\bar{q}, \bar{e}) = (\Delta q_1^l, \dots, \Delta q_m^l)^T \quad \text{(4-15)}$$

where, m denotes number of the outputs. The dynamical system representation can be obtained by the following procedure. Change in *q* is written:

$$\Delta q = \tilde{f}(\bar{q}(t), \bar{e}(t)) \quad \text{(4-16)}$$

This change in *q* is supposed to occur in *Δt* units of time. Hence we can approximate the state at *t+Δt* by,

$$\bar{q}(t+\Delta t) \cong \bar{q}(t) + \tilde{f}(\bar{q}(t), \bar{e}(t))\Delta t \tag{4-17}$$

If we rearrange equation 4-17 and take the limit when ***Δt*** goes to zero, the equation below is obtained:

$$\lim_{\Delta t \to 0} \frac{\bar{q}(t+\Delta t) - \bar{q}(t)}{\Delta t} = \tilde{f}(\bar{q}(t), \bar{e}(t)) \tag{4-18}$$

After rewriting equation 4-18, it becomes

$$\dot{q}(t) = \tilde{f}(\bar{q}(t), \bar{e}(t)) \tag{4-19}$$

Then, the system state at time instant t_1 is calculated as:

$$\bar{q}(t_1) = \int_{t_0}^{t_1} \tilde{f}(\bar{q}(\tau), \bar{e}(\tau))d\tau + \bar{q}(t_0) \tag{4-20}$$

Hence the FDEDS is a nonlinear dynamical system modeled by,

$$\begin{aligned} \dot{q}(t) &= \tilde{f}(\bar{q}(t), \bar{e}(t), t) \\ y(t) &= \tilde{h}(\bar{q}(t), \bar{e}(t), t) \end{aligned} \tag{4-21}$$

Remark: In equation 4-21, the output function $\tilde{h}$ can be defined as $\tilde{h}: \bar{Q}(t) \times \bar{\Sigma}(t) \to \Delta\hat{\Sigma}$. The definition for FDEDS can be given:

***Definition 4.3*:** A fuzzy discrete event dynamical system is modeled by a five-tuple $\tilde{G} = (\bar{Q}, \bar{\Sigma}, \tilde{f}, \tilde{h}, \bar{q}_0)$ where

- $\bar{Q}$ is the set of state space such that $\bar{q}(t) \in \bar{Q} \subseteq R^n$,where *n* is the number of states,
- $\bar{\Sigma}$ is the set of event space such that $\bar{e}(t) \in \bar{\Sigma} \subseteq R^m$, where *m* is the number of events,
- $\tilde{f}: \bar{Q}(t) \text{ x } \bar{\Sigma}(t) \to \bar{Q}(t) \subset \bar{Q}$ is the state transition function ,
- $\tilde{h}: \bar{Q}(t) \times \bar{\Sigma}(t) \to \Delta\hat{\Sigma}$ is the output equation, where $\Delta\hat{\Sigma}$ is a set of detectable events related to changes in states.

- $\bar{q}_0$ is the initial state vector showing the initial membership degrees of a system states.

Can one really say that the constructed model is a dynamical system? More formally the definition below [65] can be used.

***Definition 4.4*:** A dynamical system is a quintuple $\{u,\ y,\ X,\ \Phi,\ r\}$ satisfying the following axioms, for all $u_1, u_2 \in U,\ x_0 \in X, t_0,\ t_1,\ t_2 \in R,\ t_0 \le t_1 \le t_2$

a) State transition axiom: $\Phi(t_1,\ t_0,\ x_0,\ u_1) = \Phi(t_1,\ t_0,\ x_0,\ u_2)$ whenever $u_1(t) = u_2(t)\ \ for\ t_0 \le t \le t_1$.

b) Semi group axiom: $\Phi(t_2,\ t_0,\ x_0,\ u) = \Phi(t_2,\ t_1,\ \Phi(t_1,\ t_0,\ x_0,\ u),\ u)$.

where**,** *u, y* denotes the space of continuous functions defined on $R = (-\infty\ \infty)$ with values in *U, Y* respectively. *u* is termed the input space, *y* the output space, *U* the set of input values, *Y* the set of output values, and *X* the state space. Φ and *r* shows state transition function and read-out map ($T \times X \times U \to Y$), respectively.

It can trivially be shown that the proposed structure satisfies both of the axioms. Hence the proposed structure is really a dynamical system. At this point we will give an example to show the differences among DES, FDES and FDEDS models.

Example 4.3: As a simple example is the two-tank system shown in Figure 4.2. It consists of two tanks (*T*1 and *T*2) and two valves (*V*1, *V*2). Valve operations are binary, i.e., they can be fully open (=1) or fully closed (=0). When *V*1 is open, there is a steady water flow into the system. Water level in each tank is monitored by a level-crossing sensor, which reports only whether water level is above or below some predefined values, e.g., above 0.3 m or below 0.3 m. The state-space equations for the system shown below cover all the operating modes.

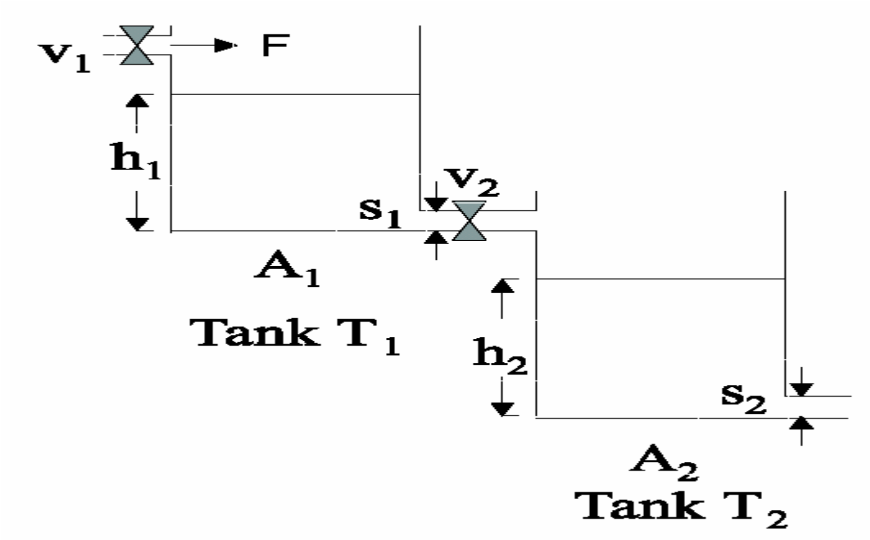

Figure 4.2: Two-tank system

$$f(x,u) = \dot{x} = \begin{bmatrix} \dot{h}_1 \\ \dot{h}_2 \end{bmatrix} = \begin{bmatrix} \dfrac{u_1 F - u_2 s_1 \sqrt{2gh_1}}{A_1} \\ \dfrac{u_2 s_1 \sqrt{2gh_1} - s_2 \sqrt{2gh_2}}{A_2} \end{bmatrix} \quad (4\text{-}22)$$

where $F = 5m^3/s$ is the steady water flow rate, $h_i \in [0, 9)m$, $A_i = 2m^2$, $s_i = 0.3m^2$ is the intersection area of Tank i, $u_i \in \{0: off\ 1: on\}$ is the control action of V_i with $i \in [1, 2]$, and $g = 9{:}8\ Nm/s^2$ is the gravity constant.

After presenting the continuous model, the discrete model is constructed as: Assume that for each i, x_i is partitioned into a set of q_i ($Q = \cup_{i=1}^{m} q_i$). In the two-tank system $X = \{(h_1, h_2) \mid h_1 \in [0 \quad 9] \wedge h_2 \in [0 \quad 9]\}$. If we divide the height of each tank into three equal intervals, the partition of X is obtained as shown in Figure 4.3.

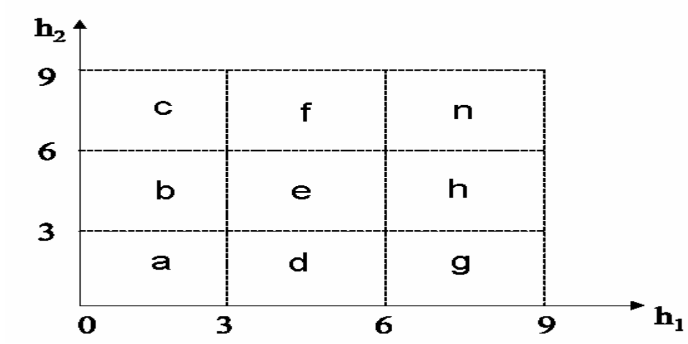

Figure 4.3: Partition on X in two-tank system

Table 4.1: State names for the two-tank system

State	h_1	h_2
a	$[0, \quad 3)$	$[0, \quad 3)$
b	$[0, \quad 3)$	$[3, \quad 6)$
c	$[0, \quad 3)$	$[6, \quad 9)$
d	$[3, \quad 6)$	$[0, \quad 3)$
e	$[3, \quad 6)$	$[3, \quad 6)$
f	$[3, \quad 6)$	$[6, \quad 9)$
g	$[6, \quad 9)$	$[0, \quad 3)$
h	$[6, \quad 9)$	$[3, \quad 6)$
n	$[6, \quad 9)$	$[6, \quad 9)$

In this figure, the names in the rectangular blocks represent the discrete states q_i and they are given in Table 4.1.

$$Q = \{a, b, c, d, e, f, g, h, n\};$$

Table 4.2: Events for the two-tank system

Events	***U₁***	***U₂***
1	0 (V_1 close)	0 (V_2 close)
2	0 (V_1 close)	1 (V_2 open)
3	1 (V_1 open)	0 (V_2 close)
4	1 (V_1 open)	1 (V_2 open)

A *detectable* event creates a measurable change in the output. We have chosen four detectable events shown in Table 4.2 and no undetectable events (Σ_u) in the system.

$$\tilde{\Sigma} = \{1, 2, 3, 4\}; \Sigma_u = \{\ \}$$

Figure 4.4 displays the discrete event model obtained. In this picture, the letter in each state is the state name and the value on each edge is the event.

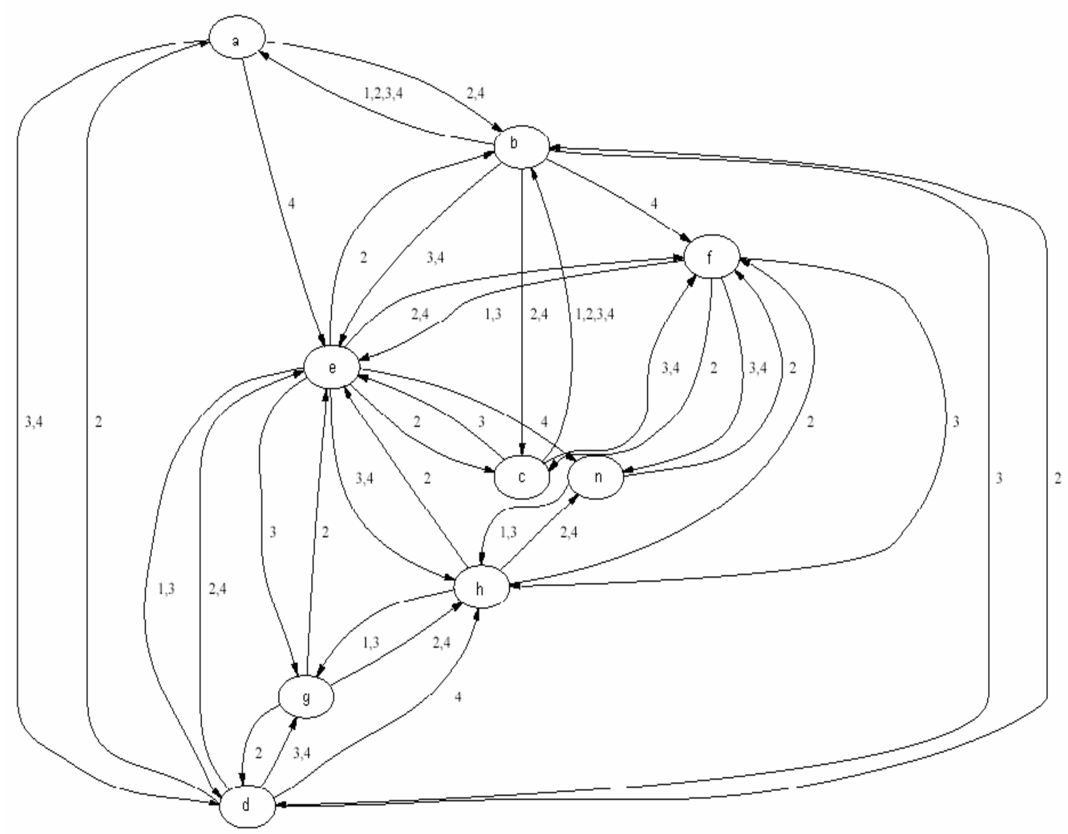

Figure 4.4: DES model for two-tank system.

If the initial state is taken as *a*, then the overall DES model can be given as:

$$Q = \{a, b, c, d, e, f, g, h, n\};$$

$$\Sigma = \{1, 2, 3, 4\}; \ \ \Sigma = \hat{\Sigma} \cup \Sigma_u$$

$$q_0 = [1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$$

$f(q, e)$ has already been given by the rules. The state transition process can be given by the following matrix.

$$\begin{bmatrix} 0 & f(2,4) & 0 & f(3,4) & f(4) & 0 & 0 & 0 & 0 \\ f(1,2,3,4) & 0 & f(2,4) & f(2) & f(3) & f(4) & 0 & 0 & 0 \\ 0 & f(1,2,3,4) & 0 & 0 & f(3) & f(3,4) & 0 & 0 & 0 \\ f(2) & f(2) & 0 & 0 & f(2,4) & 0 & f(3,4) & f(4) & 0 \\ 0 & f(2) & f(2) & f(1,3) & 0 & f(2,4) & f(3) & f(3,4) & f(4) \\ 0 & 0 & f(2) & 0 & f(1,3) & 0 & 0 & f(3) & f(3,4) \\ 0 & 0 & 0 & f(2) & f(2) & 0 & 0 & f(2,4) & 0 \\ 0 & 0 & 0 & 0 & f(2) & f(2) & f(1,3) & 0 & f(2,4) \\ 0 & 0 & 0 & 0 & 0 & f(2) & 0 & f(1,3) & 0 \end{bmatrix}$$

where,

$$f(1,\ldots,i) = f(1)\ V\ f(2)\ V \ldots V\ f(i) \qquad f(j) = \begin{cases} 1 \ \ if\ e_i \text{ occurs } i = 1,\ldots,m \\ 0 \quad \text{otherwise} \end{cases} \qquad (4\text{-}23)$$

$$f(i)\ V\ f(j) = \begin{cases} 1, \ \text{if either } f(i) \text{ or } f(j) \text{ equals } 1 \\ 0 \quad \text{otherwise} \end{cases}$$

where, m and n shows number of events and states, respectively.

The output function is defined by

$$h(q,e) = [g(1) \quad g(2) \quad g(3) \quad g(4)]^T$$

where,

$$g(j) = \begin{cases} 1, \ \text{if } e_i \ \text{ has occured} \\ 0 \quad \text{otherwise} \end{cases} \qquad (4\text{-}24)$$

Secondly, the overall FDES model can be given as:

$$\overline{Q} = \{a, b, c, d, e, f, g, h, n\};$$

The membership degree of the states at time t

$$\{\mu_a(t) \quad \mu_b(t) \quad \mu_c(t) \quad \mu_d(t) \quad \mu_e(t) \quad \mu_f(t) \quad \mu_g(t) \quad \mu_h(t) \quad \mu_n(t)\}$$

$$\bar{\Sigma} = \{1, 2, 3, 4\}; \quad \Sigma = \hat{\Sigma} \cup \Sigma_u$$

The membership degree of the events at time *t*

$$\{\mu_1(t),\ \mu_2(t), \mu_3(t), \mu_4(t)\}$$

$$\bar{q}_0 = \begin{bmatrix} \mu_a(t_0) & \mu_b(t_0) & \mu_c(t_0) & \mu_d(t_0) & \mu_e(t_0) & \mu_f(t_0) & \mu_g(t_0) & \mu_h(t_0) & \mu_n(t_0) \end{bmatrix}$$

t_0 : initial time

The state transition function $\bar{f}(q,e)$ has already been given by the rules. We can represent the state transition process by the following matrix.

$$\begin{bmatrix}
0 & \bar{f}(2,4) & 0 & \bar{f}(3,4) & \bar{f}(4) & 0 & 0 & 0 & 0 \\
\bar{f}(1,2,3,4) & 0 & \bar{f}(2,4) & \bar{f}(2) & \bar{f}(3) & \bar{f}(4) & 0 & 0 & 0 \\
0 & \bar{f}(1,2,3,4) & 0 & 0 & \bar{f}(3) & \bar{f}(3,4) & 0 & 0 & 0 \\
\bar{f}(2) & \bar{f}(2) & 0 & 0 & \bar{f}(2,4) & 0 & \bar{f}(3,4) & \bar{f}(4) & 0 \\
0 & \bar{f}(2) & \bar{f}(2) & \bar{f}(1,3) & 0 & \bar{f}(2,4) & \bar{f}(3) & \bar{f}(3,4) & \bar{f}(4) \\
0 & 0 & \bar{f}(2) & 0 & \bar{f}(1,3) & 0 & 0 & \bar{f}(3) & \bar{f}(3,4) \\
0 & 0 & 0 & \bar{f}(2) & \bar{f}(2) & 0 & 0 & \bar{f}(2,4) & 0 \\
0 & 0 & 0 & 0 & \bar{f}(2) & \bar{f}(2) & \bar{f}(1,3) & 0 & \bar{f}(2,4) \\
0 & 0 & 0 & 0 & 0 & \bar{f}(2) & 0 & \bar{f}(1,3) & 0
\end{bmatrix}$$

where, *m* and *n* shows number of events and states, respectively

$$\bar{f}(1,...,i) = \bar{f}(1) V \bar{f}(2) V ... V \bar{f}(i) \qquad \bar{f}(j) = \begin{cases} \mu_{e_i}(t) & \text{if } e_i \text{ has occured} \quad \mathrm{i} = 1,...,\mathrm{m} \\ 0 & \text{otherwise} \end{cases} \tag{4-25}$$

The output function is defined by

$$\bar{h}(q,e) = [\bar{q}(1) \quad \bar{q}(2) \quad \bar{q}(3) \quad \bar{q}(4)]^T$$

where,

$$\bar{g}(j) = \begin{cases} \mu_{e_i}(t), \ if \ \ e_i & \text{has occured} \\ 0 & \text{otherwise} \end{cases} \tag{4-26}$$

Figure 4.5 and 4.6 denote tanks' levels changes. To obtain these figures, the same event sequence (i.e., crisp or fuzzy event sequence) was applied to the system. In crisp case, an event with the membership degree one applied to the system at an instant of time. However, in fuzzy case all events applied to the system continuously with different membership degrees (i.e., the sum of all events membership degrees is one) at an instant of time. The initial state is chosen as e (i.e., see Figure 4.4).

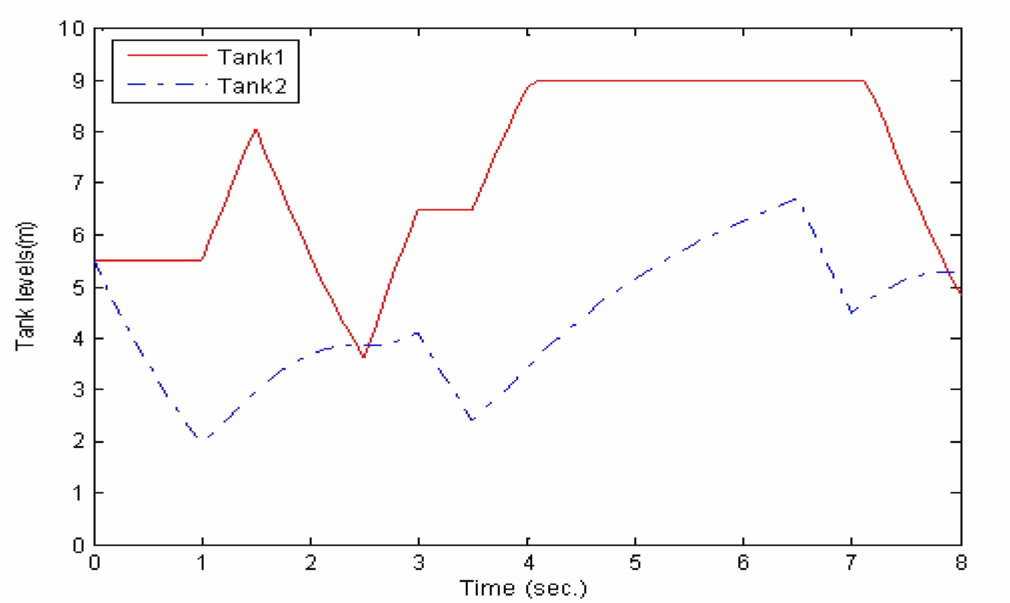

Figure 4.5: Tanks' levels changes for a crisp event sequence.

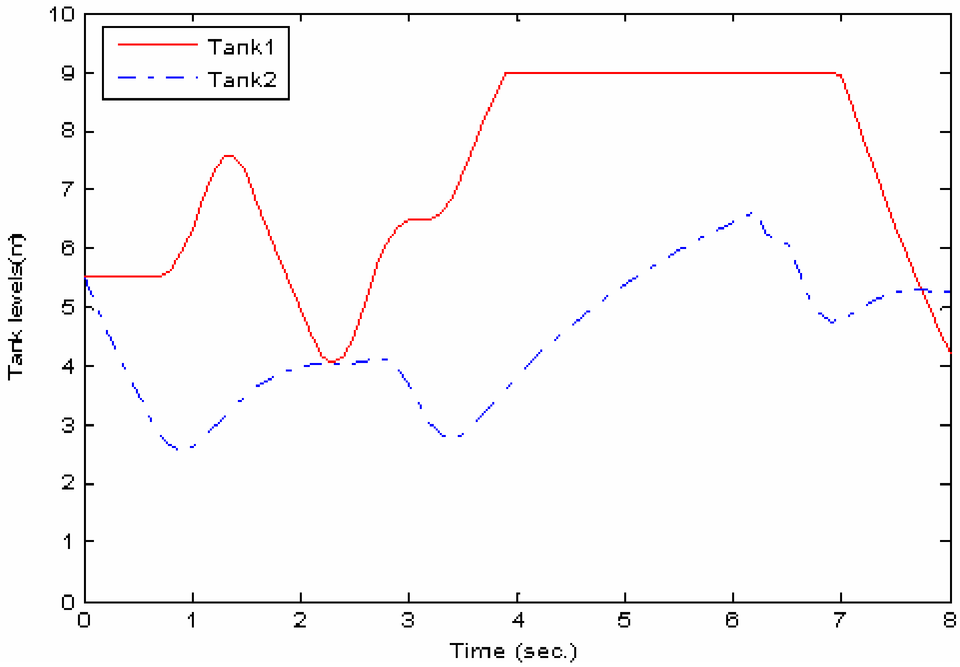

Figure 4.6: Tanks' levels changes for a fuzzy event sequence.

Lastly, the overall FDEDS model can be given by,

$\overline{Q} = \{a, b, c, d, e, f, g, h, n\};$

The membership degree of the states at time t

$\{\mu_a(t) \quad \mu_b(t) \quad \mu_c(t) \quad \mu_d(t) \quad \mu_e(t) \quad \mu_f(t) \quad \mu_g(t) \quad \mu_h(t) \quad \mu_n(t)\}$

$\overline{\Sigma} = \{1, 2, 3, 4\}; \ \Sigma = \hat{\Sigma} \cup \Sigma_u$

The membership degree of the states at time t

$\{\mu_1(t), \ \mu_2(t), \mu_3(t), \mu_4(t)\}$

$$\overline{q}_0 = \begin{bmatrix} \mu_a(t_0) & \mu_b(t_0) & \mu_c(t_0) & \mu_d(t_0) & \mu_e(t_0) & \mu_f(t_0) & \mu_g(t_0) & \mu_h(t_0) & \mu_n(t_0) \end{bmatrix}$$

t_0 : initial time

The state transition function $\tilde{f}(q, e)$ is given by equation 4-15. The output function is defined by

$$\tilde{h}(q,e) = [\tilde{q}(1) \quad \tilde{q}(2) \quad \tilde{q}(3) \quad \tilde{q}(4)]^T$$

where,

$$\tilde{g}(j) = \begin{cases} \mu_{e_i}(t), \; if \;\; e_i \;\; \text{has occured} \\ \quad 0 \qquad \text{otherwise} \end{cases} \tag{4-27}$$

We did not construct the rules for FDEDS, because the rule-base extraction procedure takes too much time. We put the overall FDEDS structure here except the rule-base to show the similarity between FDES and FDEDS structures. It is expected that FDEDS structure will produce the same results obtained when an FDES structure is used.

4.4 Observability

Observability of DES was introduced and used in many studies [15], [16]. The observability of DES is not a single concept in literature [16], [18]. However, the observability concept, which will be introduced in this work is based on the state observability definition introduced by Özveren and Willsky [16]. This definition, based on event sequences, is a generalization of the classical concept of observability. The observability is termed by using an observation sequence to determine the current state exactly at intermittent (but not necessarily fixed) points in time separated by a bounded number of events. It is known that a DES can be modeled by a five-tuple $G = (Q, \Sigma, f, h, q_0)$ as given in definition 4.1. DES observability definition by Özveren and Willsky is based on definition 4.1.

First, before presenting the definition let us introduce the notation.

- Let $L(G,q)$ denote the language generated by *G*, from the state *q,* i.e., $L(G,q)$ is the set of all possible event trajectories of finite length that can be generated if the system is started from the state *q*. Given $s \in L(G,q)$ for some *q*, let s_f denote the final event in *s*. Let

$$L_f(G,q) = \left\{ s \in L(G,q) \quad and \quad s_f \in \hat{\Sigma} \right\}$$

be the set of strings in $L(G,q)$ that have an observable (detectable) event as its final event.

- Given $s \in L(G,q)$ such that $s = pr$, *p* is termed a prefix of *s* and we use *s/p* to denote the corresponding suffix *r*, i.e., the remaining part of *s* after *p* is taken out.

Definition 4.5: G is observable if there exists some integer n_o such that $\forall q \in Q, \forall s \in L(G,q)$ such that $|s| \geq n_o$, there exists a prefix of *s*, $p \in L_f(G,q)$, such that $|s/p| \leq n_o$, $f(q,p)$ is single valued and $\forall y \in Q,\ t \in L_f(G,y) : h(t) = h(p)\ then\ f(y,t) = f(q,p)$.

This definition states the following. Take any sufficiently long string *s* that can be generated from any initial state *q*. For an observable system, we can find a prefix *p* of *s* such that *p* takes *q* to a unique state, and the length of the remaining suffix is bounded n_o. Also, for any other string *t*, from some initial state *y*, such that *t* has the same output string as *p*, we require that *t* takes *y* to the same, unique state to which *p* takes *q*.

4.4.1 Fuzzy Observability

It is known that, an FDES can be modeled by a five-tuple $G = (\overline{Q}, \overline{\Sigma}, \bar{f}, \bar{h}, \bar{q}_0)$ as given in definition 4.2. If the proposed FDES contains asynchronous fuzzy events the same observability definition as in definition 4.5 would have been valid. However, the proposed FDES includes continuously occurring events with different membership degrees at an instant of time; therefore the proposed FDES model is actually a nonlinear system. Hence, the observability definition given in definition 4.5

should be adapted to the newly defined structure of FDES. An extension of the DES observability for FDES is given by.

Definition 4.6: An FDES modeled by $G = (\overline{Q}, \overline{\Sigma}, \bar{f}, \bar{h}, \bar{q}_0)$ is said to be observable at a state *q*, if there exists some $T \rangle 0$, such that $\forall e$, the state trajectory $\Phi(q,e,t)$ creating an output trajectory $\bar{h}(q,e,t)$, and $\forall y$ the state trajectory $\Phi(y,e,t)$ creating an output trajectory $\bar{h}(y,e,t)$ satisfying the property that if $\bar{h}(\Phi(q,e,t),e,t) = \bar{h}(\Phi(y,e,t),e,t)$ *then* $y = q$.

where e denotes an event. In order to give observability degree between zero (i.e., unobservable) and one (i.e., unobservable) for FDES, a new fuzzy observability (FO) definition for FDES, based on definition 4.6, can be given as:

Definition 4.7: An FDES modeled by $G = (\overline{Q}, \overline{\Sigma}, \bar{f}, \bar{h}, \bar{q}_0)$ is said to be $\alpha\beta$ observable at a state q, if there exists $T \rangle 0$, such that $\forall e$ the state trajectory $\Phi(q,e,t)$ creating an output trajectory $\bar{h}(q,e,t)$, and $\forall y$ the state trajectory $\Phi(y,e,t)$ creating an output trajectory $\bar{h}(y,e,t)$ if

$$\max_{[t_0,\ t_0+T]} \left\| \bar{h}(\Phi(q,e,t),e,t) - \bar{h}(\Phi(y,e,t),e,t) \right\| \le 1-\alpha \text{ then } \left\| \mu_y(t) - \mu_q(t) \right\| \le \beta \quad \text{(4-28)}$$

where, t_0 denote initial time and $\|\cdot\|$ shows sup-norm. One can observe different observability degrees at different time instants with respect to this definition. We can use other norm types, but using sup-norm simplifies equation 4-28.

Claim: The expression $\max_{[t_0,\, t_0+T]} \left\| \bar{h}(q,e,t) - \bar{h}(y,e,t) \right\|$ and $\left\| \mu_y(t) - \mu_q(t) \right\| \le \beta$ is less than or equal to one at any time interval $[t_0, t_0+T]$.

Proof: Suppose the maximum value occurs at time t_1 in the *lth* component of the term $\left\|\left(\bar{h}(q,e,t_1)-\bar{h}(y,e,t_1)\right)\right\|$. Then,

$$\left\|(\bar{h}_l(q,e,t_1)-\bar{h}_l(y,e,t_1))\right\| = \left\|(\mu_l(q,e,t_1)-\mu_l(y,e,t_1))\right\|$$

The terms $\mu_l(q,e,t_1)$ and $\mu_l(y,e,t_1)$ denote *lth* event membership degrees at time t_1 starting from states *q* and *y*, respectively. These two terms always stay between zero and one since the membership degree of an event is between zero and one. This fact immediately results in

$$\max_{[t_0,t_0+T]}\left\|\bar{h}(q,e,t)-\bar{h}(y,e,t)\right\| \leq 1.$$

A similar proof can be given for the term $\left\|\mu_y(t)-\mu_q(t)\right\| \leq \beta$.

The FDEDS observability will be the same as the observability concept of the FDES. Hence, definitions 4.6 and 4.7 are valid for FDEDS.

Checking the observability in complex systems based on definitions 4.4, 4.5, 4.6 and 4.7 is not easy. A one-shot simple observability definition may be quite useful. The notion of the observability concept that will be introduced here will be different from the classical observability concept. To make a correct decision about fault types we need sufficient information about the system. Because of multiple faults, a decision maker may not know the states of the system exactly. For example $q=[1 \quad 1 \quad 0 \quad ... \quad 0]$ denotes that the system could be either in state 1 or state 2. This newly proposed form of the observability definition is based on the definition given by Lin and Ying [14] together with some modifications (for certain reasons). In the Lin and Ying approach, the crisp and fuzzy observability are based on a similarity matrix showing consistency of decisions at different states (a particular decision under consideration). This matrix has a static structure; it is not dynamic. This observability depends not only on the DES and the set of observable events but also on the particular decision to make, which is represented by the consistency matrix. In this thesis, the proposed approach to define observability uses the relations among states. This is done by using a dissimilarity (relation) matrix as.

$$w : Q \to Q$$

$$w_{ij} = \begin{cases} 0, & if \quad i = j \\ \max(q_i, q_j), & i \neq j \end{cases} \qquad (4\text{-}29)$$

where *q* denotes a system state. This matrix is state dependent whose size is *n by n (i.e., n*: number of states). The dissimilarity matrix has a dynamic structure based on system's state vector. Hence, the proposed observability definition is state dependent. A systematic way to construct the dissimilarity matrix is also proposed. Under this definition, the ambiguity in state *q* is unimportant to the decision maker, knowing in which state the system is, if $q \circ W \circ q^T = 0$. Here, the $\circ$ is the max-min operation [12].

Claim: Suppose *q* is a state vector in R^n. There is no ambiguity (the system considered could be in only one state, i.e., there is only one nonzero component of *q*) at a given time, if and only if $q \circ W \circ q^T$ equals zero.

Proof: Let us take *q* as: $q = [q_1 \quad q_2 \quad \cdots \quad q_n]$. Then,

$$q \circ W \circ q^T = [q_1 \quad q_2 \quad \cdots \quad q_n] \circ \begin{bmatrix} 0 & \max(q_1, q_2) & \max(q_1, q_3) & \cdots & \max(q_1, q_n) \\ \max(q_2, q_1) & 0 & \max(q_2, q_3) & \cdots & \max(q_2, q_n) \\ . & . & & . & . \\ . & & & . & . \\ & & . & & \\ . & & & . & . \\ \max(q_n, q_1) & \max(q_n, q_2) & \max(q_n, q_3) & \cdots\cdot & 0 \end{bmatrix} \circ \begin{bmatrix} q_1 \\ q_2 \\ . \\ . \\ . \\ q_n \end{bmatrix}$$

$$
= \max \begin{bmatrix}
\min(q_1, \max[\ 0 & \min(q_1, \min(q_2, \max(q_1, q_2))) & \min(q_1, \min(q_3, \max(q_1, q_3))) & \dots & \min(q_1, \min(q_n, \max(q_1, q_n)))]) \\
\min(q_2, \max[\min(q_2, \min(q_1, \max(q_2, q_1))) & 0 & \min(q_2, \min(q_3, \max(q_2, q_3))) & \dots & \min(q_2, \min(q_n, \max(q_2, q_n)))]) \\
. & . & & . & . \\
. & & & . & . \\
& & . & & \\
. & & & . & . \\
\min(q_n, \max[\min(q_n, \min(q_1, \max(q_n, q_1))) & \min(q_n, \min(q_2, \max(q_n, q_2))) & \min(q_n, \min(q_3, \max(q_n, q_3))) & \dots & 0\ ]
\end{bmatrix}
$$

This implies that, if the system is only one state (i.e., there is no ambiguity), then only one component of *q* is unity (and the others being equal to zero), and hence

$$
\begin{aligned}
&\min(q_1, \min(q_2, \max(q_1, q_2))) = 0,\ \min(q_1, \min(q_3, \max(q_1, q_3))) = 0\ , \\
&\dots,\ \min(q_1, \min(q_n, \max(q_1, q_n))) = 0,\ \min(q_2, \min(q_3, \max(q_2, q_3))) = 0, \\
&\dots,\ \min(q_2, \min(q_n, \max(q_2, q_n))) = 0\ , \dots,\ \min(q_{n-1}, \min(q_n, \max(q_{n-1}, q_n))) = 0
\end{aligned} \tag{4-30}
$$

which results in $q \circ W \circ q^T = 0$. On the other hand, if we start with the equality $q \circ W \circ q^T = 0$, equation 4-30 implies that no two component of *q* can simultaneously be zero; otherwise this would imply that $\min(q_i, \min(q_j, \max(q_i, q_j))) = 1$, for some *i* and *j*, which would have resulted in the contradiction that $q \circ W \circ q^T = 1$. Therefore, we have the following definition:

Definition 4.8: A crisp DES modeled by $G = (Q, \Sigma, f, h, q_0)$ is said to be observable with respect to the dissimilarity matrix *W(q)* at a state point *q* if and only if $q \circ W \circ q^T = 0$.

The expression $q \circ W \circ q^T$ is supposed to measure the ambiguity in the states. In the crisp DES observability definition the expression $q \circ W \circ q^T$, can attain only two values, 0 or 1 (observable or unobservable).

Example 4.4: Let us consider the crisp DES as in Figure 4.1 and check the observability at any state. Initially the system is in place *G*. If event α occurs at a time *t* then the system is only in place *P* and the state vector *q* is $[0 \quad 1]$. The dissimilarity matrix W can be calculated by using equation 4-29 is as:

$$W(q) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Then,

$$q \circ W \circ q^T = [1 \quad 0] \circ \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \circ \begin{bmatrix} 1 \\ 0 \end{bmatrix} = [1 \quad 0] \circ \begin{bmatrix} \max(0, 0) \\ \max(1, 0) \end{bmatrix} = \max(0, 0) = 0$$

Hence we conclude that the crisp DES is observable at the state *q*.

■

To measure the degree of ambiguity of states (i.e., the most ambiguous case is to be in all states with the same degree) in the FDES structure, a new fuzzy observability definition is needed. An important utilization of the concept of "fuzzy observability" is that it will indicate the degree of observability. In fault diagnosis applications the diagnoser can be in different states indicating faulty components at the same time (usually in the case of multiple failures; then there is ambiguity about where the failure is and, of course, how much it is.). To make a correct decision, one must have sufficient information available to distinguish fault types. Our aim in generalizing the observability analysis to the fuzzy observability case is that we wanted to give some definite meaning to being in the system states whenever possible. For example, we would like to assign a degree to the observability of the system. If the degree of observability is equal to "one" then the system is observable; in other words, there is no ambiguity (i.e., the system is only at one state at an instant of time) in the decision-making system (i.e., the diagnoser). A correct decision can be made about the situation of the system in this case. To show that the generalization of the proposed fuzzy observability definition depends on the dissimilarity matrix in equation 4-29; first the crisp version of the fuzzy observability definition is given in definition 4.8. The crisp DES observability definition then can be automatically generalized to fuzzy observability (FO) of FDES based on Definition 4-8.

Definition 4.9: An FDES modeled by $G = (\overline{Q}, \overline{\Sigma}, \bar{f}, \bar{h}, \bar{q}_0)$ is said to be observable with degree α with respect to the dissimilarity matrix $W(\bar{q})$ at a state $\bar{q}$ at any time *t* if $1 - \dfrac{\bar{q} \circ W \circ \bar{q}^T}{\bar{q} \circ \bar{q}^T} = \alpha$.

$$FO = FO(\bar{q}, W(\bar{q})) = 1 - \frac{\bar{q} \circ W \circ \bar{q}^T}{\bar{q} \circ \bar{q}^T} = \alpha \qquad (4\text{-}31)$$

where $\bar{q} \circ W \circ \bar{q}^T / \bar{q} \circ \bar{q}^T$ describes the degree of dissimilarity among states in $\bar{q}$. $FO(\bar{q}, W(\bar{q}))$ takes values in between 1 and 0 for all time. If $FO(\bar{q}, W(\bar{q}))$ is 1, then the FDEDS is completely observable at state $\bar{q}$ at any time t with respect to the dissimilarity matrix *W*. One can observe different observability degrees for different states $\bar{q}$ with respect to this definition.

Claim: Suppose $\bar{q}$ is a state vector in R^n. There is ambiguity (i.e., the considered system could be in all states with the same membership degree) at a given time t_1, if and only if $1 - \dfrac{\bar{q} \circ W \circ \bar{q}^T}{\bar{q} \circ \bar{q}^T}$ equals zero.

Proof: Let us take $\bar{q}$ as: $\bar{q} = [\mu_{q_1}(t_1) \;\; \mu_{q_2}(t_1) \;\; ,..., \;\; \mu_{q_n}(t_1)]$. Then,

$$\bar{q}\circ W\circ\bar{q}^T=[\mu_{q_1}(t_1)\ \ \mu_{q_2}(t_1)\ \ \dots\ \ \mu_{q_n}(t_1)]\circ\begin{bmatrix} 0 & \max(\mu_{q_1}(t_1),\mu_{q_2}(t_1)) & \max(\mu_{q_1}(t_1),\mu_{q_3}(t_1)) & \dots & \max(\mu_{q_1}(t_1),\mu_{q_n}(t_1)) \\ \max(\mu_{q_2}(t_1),\mu_{q_1}(t_1)) & 0 & \max(\mu_{q_2}(t_1),\mu_{q_3}(t_1)) & \dots & \max(\mu_{q_2}(t_1),\mu_{q_n}(t_1)) \\ . & . & & . & . \\ . & & & . & . \\ . & & . & . & . \\ \max(\mu_{q_n}(t_1),\mu_{q_1}(t_1)) & \max(\mu_{q_n}(t_1),\mu_{q_2}(t_1)) & \max(\mu_{q_n}(t_1),\mu_{q_3}(t_1)) & \dots & 0 \end{bmatrix}\circ\begin{bmatrix}\mu_{q_1}(t_1)\\ \mu_{q_2}(t_1)\\ .\\ .\\ .\\ \mu_{q_n}(t_1)\end{bmatrix}$$

This implies that, if the system is in all states at time t_1 (i.e., the most ambiguous case), then the component of $\bar{q}$, i.e., $\mu_{q_i}(t_1)$ will be same. Let us take $\mu_{q_i}(t_1)=\mu_q(t_1)$ hence

$$\begin{aligned}&\min(\mu_{q_1}(t_1),\min(\mu_{q_2}(t_1),\max(\mu_{q_1}(t_1),\mu_{q_2}(t_1))))=\mu_q(t_1),\ \min(\mu_{q_1}(t_1),\min(\mu_{q_3}(t_1),\max(\mu_{q_1}(t_1),\mu_{q_3}(t_1))))=\mu_q(t_1),\\ &\dots,\ \min(\mu_{q_1}(t_1),\min(\mu_{q_n}(t_1),\max(\mu_{q_1}(t_1),\mu_{q_n}(t_1))))=\mu_q(t_1),\ \min(\mu_{q_2}(t_1),\min(\mu_{q_3}(t_1),\max(\mu_{q_2}(t_1),\mu_{q_3}(t_1))))=\mu_q(t_1),\\ &\dots,\ \min(\mu_{q_2}(t_1),\min(\mu_{q_n}(t_1),\max(\mu_{q_2}(t_1),\mu_{q_n}(t_1))))=\mu_q(t_1),\dots,\ \min(\mu_{q_{n-1}}(t_1),\min(\mu_{q_n}(t_1),\max(\mu_{q_{n-1}}(t_1),\mu_{q_n}(t_1))))=\mu_q(t_1)\end{aligned} \tag{4-32}$$

which results in $\bar{q}\circ W\circ\bar{q}^T=\mu_q(t_1)$, then

$$\bar{q}\circ\bar{q}^T=\left[\mu_{q_1}(t_1)\ \ \mu_{q_2}(t_1)\ \ \dots\ \ \mu_{q_n}(t_1)\right]\begin{bmatrix}\mu_{q_1}(t_1)\\ \mu_{q_2}(t_1)\\ .\\ .\\ .\\ \mu_{q_n}(t_1)\end{bmatrix}=\max\left\{\mu_{q_1}(t_1),\ \ \mu_{q_2}(t_1),\ \ \dots\ \ ,\mu_{q_n}(t_1)\right\}$$

which results in $\bar{q}\circ W\circ\bar{q}^T=\mu_q(t_1)$. The dominator term $\bar{q}\circ\bar{q}^T$ cannot be zero since the system must be in some state at any time t with non-zero membership degrees. As a result, the equation $1-\dfrac{\bar{q}\circ W\circ\bar{q}^T}{\bar{q}\circ\bar{q}^T}$ becomes zero (i.e., $1-\dfrac{\bar{q}\circ W\circ\bar{q}^T}{\bar{q}\circ\bar{q}^T}=1-\dfrac{\mu_q(t_1)}{\mu_q(t_1)}=0$). The case, which makes the observability degree of the system a maximum, is that, the system must be in only one state with a membership degree one. For proof see the previous observability discussion.

Example 4.5: Let us consider again Figure 4.1 as an FDES and check its observability degree at a time t_1. Initially the system's state is *G* and *P* with membership degrees 0.9 and 0.1, respectively. Suppose that the possibility for the system to evolve from state *G* to *P*, *G* to *G* and *P* to *P*, at the same fixed time *t* are α_{GP} = 0.6, β_{GG} = 0.3 and σ_{PP} = 0.1, respectively. Then the state vector $\tilde{q}$ is computed, using the $\max(\circ)$ operation as:

$$\bar{q} = \max([\bar{q}_0 \circ \bar{f}(\alpha), \bar{q}_0 \circ \bar{f}(\beta), \bar{q}_0 \circ \bar{f}(\sigma)]),$$

$$\bar{q} = \max([0.9 \quad 0.1] \circ \begin{bmatrix} 0 & 0.3 \\ 0 & 0 \end{bmatrix}, \quad [0.9 \quad 0.1] \circ \begin{bmatrix} 0.6 & 0 \\ 0 & 0 \end{bmatrix}, \quad [0.9 \quad 0.1] \circ \begin{bmatrix} 0 & 0 \\ 0 & 0.1 \end{bmatrix})$$

$$= \max([0 \ \ 0.3], [0.6 \ \ 0], [0 \ \ 0.1]) = [0.6 \quad 0.3].$$

The dissimilarity matrix *W* can be calculated by using equation 4-29 as:

$$W(\tilde{q}) = \begin{bmatrix} 0 & 0.6 \\ 0.6 & 0 \end{bmatrix}$$

Then

$$FO = 1 - \bar{q} \circ W \circ \bar{q}^T \ / \ \bar{q} \circ \bar{q}^T = 1 - [0.6 \quad 0.3] \circ \begin{bmatrix} 0 & 0.6 \\ 0.6 & 0 \end{bmatrix} \circ \begin{bmatrix} 0.6 \\ 0.3 \end{bmatrix} / \, 0.6$$

$$= 1 - [0.6 \quad 0.3] \circ \begin{bmatrix} \max(0, \ 0.3) \\ \max(0.6, \ 0) \end{bmatrix} / \, 0.6$$

$$= 1 - \max(0.3, \ 0.3) / 0.6 = 1 - 0.3 / 0.6 = 0.5.$$

Hence we conclude that considered FDES is observable at degree 0.5 at the time t_1. To check the observability of the FDEDS, the FDES observability concept can be used. Hence, definition 6 is valid for FDEDS.

4.5 Conclusions

In this section, the FDES and FDEDS structures are introduced. They are general structures and can be used to construct a nonlinear dynamical system. Because of the dynamic aspects of the rule-bases, the resultant FDES is automatically a dynamical system called FDEDS. Theoretically, almost all of the ideas, definitions, algorithms, results, etc., are the same in such a case, but obviously derivations are a lot more complicated.

In this chapter, a new fuzzy observability concept and a simple observability checking method is also proposed to check nonlinear complex systems observability degree. This simple observability definition is based on a relationship on the state space, called dissimilarity relation represented by the matrix W. This matrix can be chosen in many different ways. But to decrease ambiguity, a systematic way was proposed to construct this matrix. This observability is different from the classical observability concept.

CHAPTER 5

KNOWLEDGE-BASED FDD

5.1 Necessity of Knowledge-Based FDD

Model-based FDD approaches may not provide accurate results in complex, nonlinear systems, because they make use of mathematical models of the nonlinear systems. A perfectly accurate mathematical model of a physical system is never available. This assumption has limited the success in practical applications. Usually, the parameters of the system may vary with time and the characteristic of the disturbances and noises are unknown, so that they cannot be modeled accurately. There is always a mismatch between the actual process and its mathematical model even under no fault conditions. Hence, information obtained is incomplete or uncertain in nonlinear, complex systems. It is essential to deal with the incomplete knowledge in an efficient way. A more suitable solution to process incomplete knowledge may be the utilization of knowledge-based techniques. A knowledge-based FDD is constructed by using analytical and heuristic knowledge. The features related to the system behavior are extracted from system characteristic values, while the system is working normal and faulty cases. After that, these features are used for FDD purposes. If no information is available on the fault-event relations, classification methods are used especially with neural network and fuzzy logic. If more information between events and faults is available, IF-THEN rule-based reasoning can be applied. A growing application area in knowledge-based FDD is fuzzy rule-based reasoning approaches.

5.2 Knowledge-Based FDD

Knowledge-based fault detection is constructed by analytical and heuristic symptom (event) generation. The features from system characteristic values (variances, amplitude, frequency, model parameters, state variables, transformed residuals, special noise, color, smell, vibration) are extracted, while the system is working under normal and faulty conditions using analytic and heuristic knowledge. Then the features containing faults are compared with the features of the non-faulty process and methods of change detection are applied.

5.2.1 FDES or FDEDS Based FDD

The structures of the DES, FDES and FDEDS are given in Chapter 4. These newly proposed structures are based on fuzzy IF-THEN rules. Hence, they can be considered as knowledge-based systems. In the next section, we will show how to use FDES or FDEDS structure to solve an FDD problem in nonlinear, complex systems. It has been applied in many engineering fields. Recently, the FDD problem has also been investigated via DES approach [8]. However, the DES approaches are not adequate in fault diagnosis applications, in which the states (e.g., a component health status) are somewhat uncertain (e.g., degree of fault) and vague even in a deterministic sense [22], [23]. Sometimes one may need to model systems that cannot be modeled by the current DES modeling methods due to the vagueness in the definitions of the states and/or events. In order to overcome these difficulties, the concepts of fuzzy state and fuzzy event can be used [11], [10]. Lin and Ying [13], [14] initiated the study of FDES by combining fuzzy set theory with DES to solve problems which are not possible to be solved by conventional DES. The research on the diagnostic problem for such systems with fuzziness is interesting and important. Hence, to deal with the incomplete knowledge in an efficient way, fuzzy sets and fuzzy logic can be used. Such representations and calculations are mathematically precise. In this part of the thesis to solve the fault diagnosis problem, knowledge-based FDES and FDEDS approaches are used.

5.2.2 Diagnosability/Fuzzy Diagnosability

The diagnosability problem is investigated extensively in the literature [17], [18]. Actually diagnosis means to determine faults occurring in a nonlinear system confidently and efficiently. The actuator and component faults are generally considered as an additive input explained in Chapter 1. Hence a nonlinear system model including all possible additive component and actuator faults can be given as:

$$\begin{aligned} \dot{x}(t) &= f(x(t), \bar{u}(t)) \\ y(t) &= g(x(t), u(t)) \end{aligned} \tag{5-1}$$

where $x \in R^n$ is the state vector, $\bar{u}(t) = (u, f_c) \in R^m \times R^q$, $u(t) \in R^m$ is a known input, $f_c \in R^q$ is an unknown fault vector including additive component and actuator faults, and $y \in R^p$ are the measured outputs and t is the time index. The observability and diagnosibility in nonlinear systems is defined as [93]:

Definition 5.1: A system given in (5-1) is said to be state observable on time $[t_0 \quad t_1]$ if the states x can be determined from the system equations and the time histories of the data $u[t_0 \quad t_1]$ and $y[t_0 \quad t_1]$.

Definition 5.2: A system given in (5-1) is said to be diagnosable on time $[t_0 \quad t_1]$ if it is possible to estimate the fault f_c on time interval $[t_0 \quad t_1]$ from the system equations and the time histories of the data $u[t_0 \quad t_1]$ and $y[t_0 \quad t_1]$, i.e., is diagnosable if f_c is observable on time interval $[t_0 \quad t_1]$ with respect to $u[t_0 \quad t_1]$ and $y[t_0 \quad t_1]$.

Definition 5.2 shows that the diagnosability problem can actually be considered as an observability problem and is solved by using an unknown input observer (UIO). The classical theory of observers is concerned with the reconstruction of the state from the input and output of the system. When the input is not completely available, an UIO is designed to estimate unknown inputs. It is known that a diagnosable system need not be observable, and vice versa [93].

Example 5.1:

$$\begin{cases} \dot{x}_1 = x_1 + u + f_c \\ \dot{x}_2 = -x_2 + x_1 \\ y = x_1 \end{cases}$$

which is diagnosable ($f_c = \dot{y} - y - u$) but not observable (x_2 is not observable with respect to u and y).

Example 5.2:

$$\begin{cases} \dot{x}_1 = x_2 + u \\ \dot{x}_2 = -x_2 - x_1 + f_c \\ y = x_1 + u \end{cases}$$

which is not diagnosable ($f_c = x_2 + \dot{x}_2 + y - u$ is not diagnosable with respect to u and y) but observable if f_c equals zero ($x_1 = y - u$, $x_2 = \dot{y} - \dot{u} - u$).

We have a result, which should be kept in mind.

Theorem: If system (5-1) is observable on time interval $[t_0 \quad t_1]$ then it is diagnosable on time interval $[t_0 \quad t_1]$ if and only if f_c is observable on time interval $[t_0 \quad t_1]$ with respect to $u[t_0 \quad t_1]$, $y[t_0 \quad t_1]$ and $x[t_0 \quad t_1]$.

The theorem clearly shows a ***relationship between diagnosability and observability concepts.*** A diagnosability definition may be given for nonlinear systems similar to definition 5.1. However, we are interested in the event-base systems. Hence, the nonlinear diagnosability definition should be adapted to the event-base systems.

5.2.2.1 Fault Model for DES

In the event-base systems, the faults can be considered as events. It is known that the actuator and component faults are modeled as additive inputs. Hence, to create a faulty DES model we can handle the faulty events as inputs. As a result, the fault model for a DES can be given by a five-tuple

$$\breve{G} = (Q, \breve{\Sigma}, f, h, q_0) \qquad (5\text{-}2)$$

where $\breve{\Sigma}$ (i.e., $\breve{\Sigma} = \{\Sigma, \ \Sigma_f\}$) is a set of events, Σ_f contains detectable and undetectable faulty events.

The meaning of the diagnosability for event-base systems is that, a faulty event-base system is diagnosable if it is possible to detect occurrence of failures by using observed events. Here, the diagnosability problem is handled as unknown input observability problem. Besides, a relation between diagnosibility and state observability for event-based systems is given similar to what is done for nonlinear systems. Thus, we should use the DES fault model to give the diagnosibility concept.

Definition 5.3: A DES modeled by $G = (Q, \Sigma, f, h, q_0)$ is said to be diagnosable at a state *q* with respect to the fault model $\breve{G} = (Q, \breve{\Sigma}, f, h, q_0)$, system inputs, states q_i and measurable outputs, if there exists some $T \rangle 0$, such that $\forall e$ the state trajectory $\Phi(q,e,t)$ creating an output trajectory $h(q,e,t)$, and $\forall y$ the state trajectory $\Phi(y,e,t)$ creating an output trajectory $h(y,e,t)$ if $h(\Phi(q,e,t),e,t) = h(\Phi(y,e,t),e,t)$ *then* $\Sigma_{fy} = \Sigma_{fq}$.

We have a result, which should be kept in mind.

Theorem: If a faulty DES modeled by $\breve{G} = (Q, \breve{\Sigma}, f, h, q_0)$ is observable at a state *q* then it is diagnosable if and only if Σ_f is observable (i.e., detectable) with respect to the system inputs, outputs and states.

Proof: If the faulty system is diagnosable at a state *q*, the event set Σ_f can be obtained by using the system inputs, outputs (by definition). Proof in the opposite direction is also trivial.

Using a similar approach, the diagnosibility for FDES can be given as:

Definition 5.4: A FDES modeled by $\overline{G} = (\overline{Q}, \overline{\Sigma}, \bar{f}, \overline{h}, \overline{q}_0)$ is said to be diagnosable at a state *q* with respect to the fault model $\hat{G} = (\overline{Q}, \hat{\Sigma}, \bar{f}, \overline{h}, \overline{q}_0)$, system inputs, states $\overline{q}_i$, and measurable outputs, if there exists some $T \rangle 0$, such that $\forall e$ the state trajectory $\Phi(q,e,t)$ creating an output trajectory $\overline{h}(q,e,t)$, and $\forall y$ the state trajectory $\Phi(y,e,t)$ creating an output trajectory $\overline{h}(y,e,t)$ if $\overline{h}(\Phi(q,e,t),e,t) = \overline{h}(\Phi(y,e,t),e,t)$ *then* $\Sigma_{fy} = \Sigma_{fq}$.

where $\hat{\Sigma}$ (i.e., $\hat{\Sigma} = \{\Sigma, \ \Sigma_f\}$) is a set of events, Σ_f contains detectable and undetectable faulty events.

To give diagnosability degree between zero (i.e., not diagnosable) and one (i.e., diagnosable), a new fuzzy diagnosability (FD) definition for FDES can be given as:

Definition 5.5: A faulty FDES modeled by $\hat{G} = (\overline{Q}, \hat{\Sigma}, \bar{f}, \bar{h}, \bar{q}_0)$ is said to be $\gamma\delta$ diagnosable at a state *q* with respect to the system inputs e_i, states q_i, and measurable outputs, if there exists $T \rangle 0$, such that $\forall e$ state trajectory $\Phi(q,e,t)$ creating an output trajectory $\bar{h}(q,e,t)$, $\forall y$ state trajectory $\Phi(y,e,t)$ creating an output trajectory $\bar{h}(y,e,t)$ if

$$\max_{[t_0,\ t_0+T]} \left\| \bar{h}(\Phi(q,e,t),e,t) - \bar{h}(\Phi(y,e,t),e,t) \right\| \le 1-\gamma \text{ then } \left\| \mu_y(t) - \mu_q(t) \right\| \le \delta \quad (5\text{-}3)$$

where, t_0 denote initial time and the norm is sup-norm. The expression $\max_{[t_0, t_0+T]} \left\| \bar{h}(q,e,t) - \bar{h}(q,e,t) \right\|$ and $\left\| \mu_y(t) - \mu_q(t) \right\| \le \delta$ stays between zero and one at time interval [t_0, t_0+T]. For a proof see observability concept given in Chapter 4.

After presenting the diagnosability concept, the next task is to find faults. They can be determined by using a diagnoser. A diagnoser can be considered as an unknown input observer. Hence, a diagnoser can be constructed as an observer. A state observer is usually used in order to reconstruct the state variables. For this purpose, Luenberger developed an observer design theory for linear systems [87]. This theory gives a complete answer to the observer design problem. However, the observer design problem is a more challenging task for nonlinear systems. In literature, there are many attempts to solve nonlinear observer design problem such as linearization approach, extended Kalman filter approach and extended Luenberger observer approaches. The extension of Luenberger observer focuses on analytic design techniques of the nonlinear observers and has been applied to realistic nonlinear systems [88], [89], [90]. The observer gain is chosen in such a way that the overall linearized error dynamic matrix have stable eigenvalues over a closed subset of the state space [91], [92]. A linear time-invariant (LTI) system can be modeled as:

$$\dot{x}(t) = A.x(t) + B.u(t)$$
$$y(t) = C.x(t) + D.u(t) \tag{5-4}$$

where $x \in \Re^k$ is the state vector, $y \in \Re^l$ is the output vector, $u \in \Re^r$ is the input vector at time *t*., a stable linear Luenberger observer with an observable pair (A, C) is given by

$$\dot{\tilde{x}}(t) = A.\tilde{x}(t) + B.u(t) + L(y(t) - \tilde{y}(t))$$
$$\tilde{y}(t) = C.\tilde{x}(t) + D.u(t) \tag{5-5}$$

It can be designed by placing the poles of the observer at any desired location such that the error signals ($error = x(t) - \tilde{x}(t)$) exhibit the desired dynamics. The extension of the Luenberger observer to nonlinear systems is straightforward. A nonlinear time varying dynamical system can be modeled as:

$$\dot{x}(t) = f(x(t), u(t), t)$$
$$y(t) = h(x(t), u(t), t) \tag{5-6}$$

A nonlinear Luenberger observer is given by

$$\dot{\tilde{x}}(t) = f(\tilde{x}(t), u(t), t) + L((y(t) - \tilde{y}(t)),\ t)$$
$$\tilde{y}(t) = h(\tilde{x}(t), u(t), t) \tag{5-7}$$

Though, there is a strong theory behind the linear Luenberger observers and there are strict analytical methods of selecting the observer gain vector *L(t)*, such results are not available for nonlinear Luenberger observers, yet. One of the methods is the gradient adaptation method that adjusts gain parameter *L(t).* In this method the optimum *L* is computed that minimizes the mean-square-error (MSE) along a given training trajectory

$\{u(t), y(t)\}_{0 \le t \le T}$. In this case the cost function and the gradient is given as:

$$J = \int_{t=t_0-T}^{t_0} \left\| y(t) - \tilde{y}(t) \right\|^2 dt$$
$$\frac{\partial J}{\partial L} = -2 \int_{t=t_0-T}^{t_0} (y(t) - \tilde{y}(t)) \cdot (\frac{\partial \tilde{y}(t)}{\partial L}) dt \tag{5-8}$$

where,

$$\frac{\partial \tilde{y}}{\partial L} = \frac{\partial f}{\partial \tilde{x}(t)} \frac{\partial \tilde{x}(t)}{\partial L} \tag{5-9}$$

If the steepest descent algorithm is used in the minimization then

$$L_{new} = L_{old} - \eta \frac{\partial J}{\partial L}$$

The term $\tilde{x}(t)$ in equation (5-9) can be solved as an initial value problem as:

$$\frac{\partial \dot{\tilde{x}}(t)}{\partial L} = \frac{\partial f}{\partial \tilde{x}(t)} \frac{\partial \tilde{x}(t)}{\partial L} - L \frac{\partial h}{\partial \tilde{x}(t)} \frac{\partial \tilde{x}(t)}{\partial L} + (y(t) - \tilde{y}(t)).[a]_{n \times n},$$
$$\partial \tilde{x}(t_0) = \tilde{x}_o,$$
$$\frac{\partial \tilde{x}(t_0)}{\partial L} = 0.$$

where $[a]_{n \times n}$ denotes an all-ones square matrix of the size of the state vector. The J and η denote the cost function and learning rate, respectively. The same algorithm can be implemented by using a neural network.

A Luenberger observer for fuzzy dynamical event-based systems can be constructed in the same way as in the nonlinear case. It is known that an FDEDS is actually a nonlinear system represented as shown in Chapter 4. In that chapter, the nonlinear model for an FDEDS is obtained as:

$$\begin{aligned} \dot{q}(t) &= f(q(t), e(t), t) \\ y(t) &= h(q(t), e(t), t) \end{aligned} \tag{5-10}$$

where $q \in \Re^n$ is the state vector including system faults which occurs continuously with different membership degrees $(\mu_{q_1}(t), ..., \mu_{q_n}(t))$ at an instant of time*,* where *n* is the number of states, $y \in \Re^m$ is the output event vector consisting of detectable events that occurs continuously with different membership degrees $(\mu_{e_1}(t), ..., \mu_{e_m}(t))$ at an instant of time, where *m* is the number of events, $e \in \Re^l$ is the input event vector containing detectable and undetectable events at time *t*. Using the same approach as in the nonlinear case, a Luenberger observer for FDEDS is given by,

$$\begin{aligned} \dot{\tilde{q}}(t) &= f(\tilde{q}(t), e(t), t) + L(y(t) - \tilde{y}(t),\ t) \\ \tilde{y}(t) &= h(\tilde{q}(t), e(t), t) \end{aligned} \tag{5-11}$$

The observer gain vector *L(t)* can be determined by using steepest decent algorithm as given before*.*

Another way to construct a diagnoser is to use an expert system. In this thesis, we proposed diagnoser structures based on expert systems. The FDES or FDEDS based diagnosers include a rule-base produced by using analytic or heuristic knowledge. In the rule-base, relations between events and fault types (*IF* event *then* fault type and its degree of failure) are used. The diagnoser design procedure will be given in the next section.

5.2.3 Diagnoser Design Procedure

In this part, the diagnoser will be constructed by using FDES and FDEDS approaches based on fuzzy rules derived by an expert or knowledge-based techniques. A diagnoser isolates faults and also gives information about the percentage of the occurred fault types. The diagnoser's input-output relations are as follows. The differences between measurable system and observer outputs are defined as residuals, which are in the time domain and applied to the diagnoser as inputs after labeling them as events (crisp/fuzzy) by the event generator. The diagnoser's (i.e., FDES and FDEDS) outputs are degrees of faults (fault percentage) related to the faulty components. This is accomplished by an event dependent fuzzy rule-base. To fuzzify crisp events, triangular and trapezoidal membership functions (or others) are used. The informal diagnoser construction procedure is:

1. Decide on the number of the states to show the faulty components and their usual conditions,
2. Construct fuzzy if-then rule-base to show the relation between events (inputs) and fault types (places).

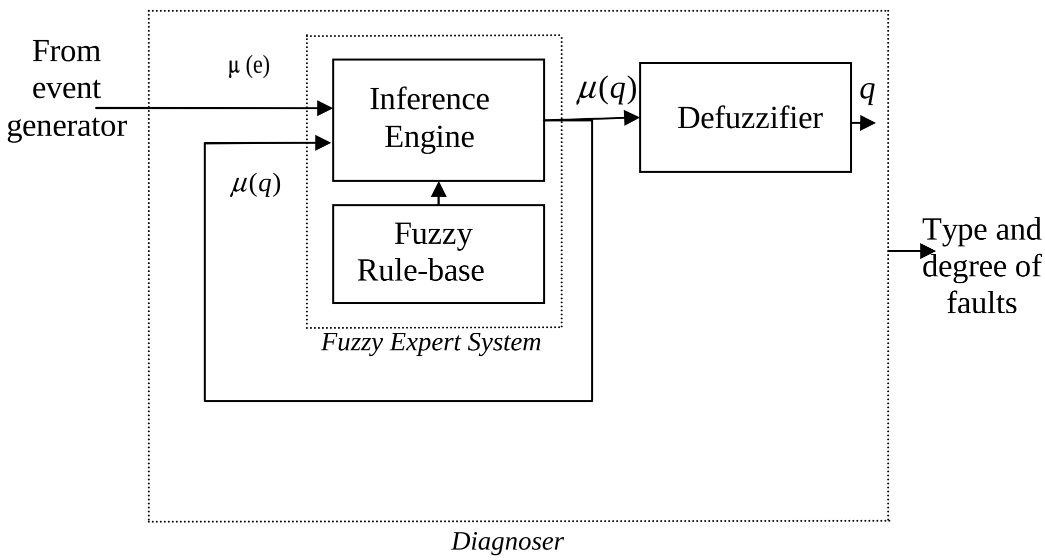

Figure 5.1: FDES diagnoser compact form

The proposed FDES and FDEDS diagnoser compact forms are given in Figures 5.1 and 5.2, respectively. In these figures, *q* and *e* show state and detectable (measurable) event vectors, respectively.

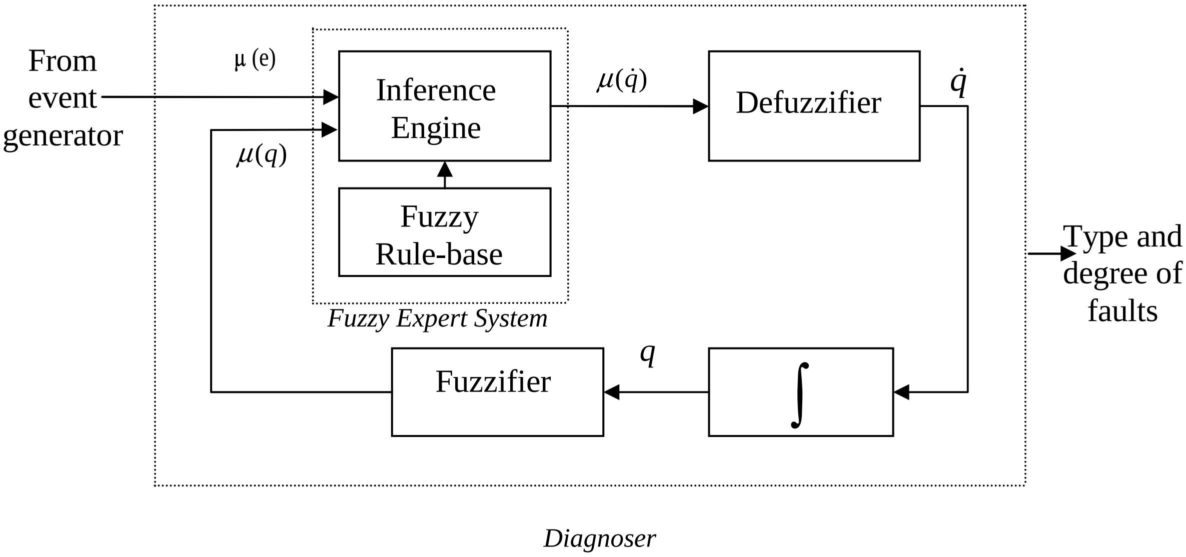

Figure 5.2: FDEDS diagnoser compact form

There are fuzzy expert systems within the diagnosers. The most commonly employed fuzzy expert systems are Mamdani and Sugeno type expert systems [12]. The Sugeno type expert system is computationally effective and works well with the optimization and adaptive techniques. The Mamdani type is widely accepted for capturing expert knowledge. It allows us to describe the expertise in more intuitive, more human-like manner. Hence, the Mamdani type expert system is preferred in the diagnosers. The applied inputs to the fuzzy expert systems are the fuzzified states and events occurring continuously all the time with different membership degrees.

The events, states and change in states are used to construct the rule-base. The rule-base structures used in FDEDS and FDES diagnosers are given in equation (4.8) and (4.9), respectively, are repeated here.

R^i **: IF** $\bar{e}_t,\ldots,\bar{e}_{t-N}$ is $A^i_1,\ldots,A^i_N$ **AND** $\bar{q}_t,\ldots,\bar{q}_{t-N}$ is $B^i_1,\ldots,B^i_N$ **THEN** ($\bar{q}_1$ is $C^i_1,\ldots,\bar{q}_N$ is C^i_N), $i = 1,\ldots,n$

R^i **: IF** $\bar{e}_t,...,\bar{e}_{t-N}$ is $A^i_1,...,A^i_N$ **AND** $\bar{q}_t,...,\bar{q}_{t-N}$ is $B^i_1,\ldots,B^i_N$ **THEN** ($\Delta\bar{q}_1$ is $C^i_1,\ldots,\Delta\bar{q}_N$ is C^i_N), $i = 1,...,n$

The second rule denotes a dynamical rule-base structure as explained in Chapter 4.

The inference, evaluation of fuzzy rules to produce an output for each rule, is the kernel of a fuzzy expert system. A fuzzy rule is a fuzzy relation, which is expressed as a fuzzy implication. According to the compositional rule of inference, conclusion can be obtained by taking the composition of related fuzzy set and fuzzy implication. Among the various fuzzy implications Mamdani's fuzzy implication method (minimum operation) associated with max-min composition is chosen in the diagnoser structure.

The fuzzifier unit in the FDEDS diagnoser performs the function of fuzzification, which is a subjective evaluation to transform incoming crisp data into a subjective value. Hence, it can be defined as a mapping from an observed input space to labels of fuzzy sets in a specified input universe of discourse. The defuzzification is another unit used in the diagnosers. Defuzzification is a conversion of fuzzy set produced by an inference engine into a crisp value. This process is necessary because a crisp state value is required. Unfortunately, there is no systematic procedure for choosing a defuzzification strategy. The COA method is used in the diagnosers.

5.2.4 Event Generation

In this thesis analytic and heuristic knowledge are used together to produce events.

5.2.4.1 Event Generation with Analytic Knowledge

The analytic knowledge on the process is used to obtain quantitative or analytical information. The characteristic values (variances, amplitude, frequency, model parameters, state variables or transformed residuals) are based on measured process variables and obtained by a data processing unit (i.e., limit value checking, signal analysis by the use of a model like ARMA and process analysis by using mathematical model with estimation techniques). The features related to the system characteristic values are extracted, while system is working under normal and faulty conditions. Then the features containing faults are compared with the normal features of the non-faulty process and methods of change detection and classification are applied. The resulting differences are residuals labeled as events.

5.2.4.2 Event Generation with Heuristic Knowledge

Heuristic symptoms are generated by using qualitative information from human operators. The heuristic characteristic values, special noise, color, smell, vibration etc., are obtained through human observation. The process history (e.g., maintenance, repair, former faults, lifetime etc.) is a further source of heuristic information. Fault statistics are also used to obtain heuristic information. By this way, heuristic symptoms are produced. Then the heuristic symptoms are represented as linguistic variables (e.g., small, medium, large). These linguistic variables are called events.

5.2.4.3 Rule Extraction

Rule-base construction is an important task in an expert system. A rule denotes the relationship between events and degree of faults in FDES and FDEDS diagnosers having an expert system. A rule-base is constructed by using analytic and heuristic knowledge. A logical and straightforward way to construct a rule-base is to use the system mathematical model (i.e., analytical knowledge). System’s faults can be determined by using system’s fault model and residuals (i.e., the system variables measured). If we do not know the system’s mathematical model we can obtain residuals by injecting faults to the system. Then the rules can be constructed by using optimization based approaches.

5.2.4.3.1 Rule Extraction with Genetic Algorithms

Genetic algorithms are powerful tools for optimization, which are capable of jumping over the local extremes and approaching to the global extreme. Genetic algorithms make use of the principles of natural biological evolution. As the candidates more suitable for adaptation to the environment have more chance to pass their features to next generations, the chromosomes that result in a higher fitness when applied to the fitness function of the genetic algorithm have more chance to pass their characteristics to regenerated chromosomes. In this way to better chromosomes are created in each generation. Crossover and mutation operators increase the variety that would introduce new features probably suitable for the task.

A rule $R = \{R_a, \quad R_c\}$ consists of two parts called antecedent R_a and consequent R_c, which is given in equations (4-8) and (4-9). To derive rules by using fault-event relations with a genetic algorithm, first a chromosome including rules should be constructed. A rule-base can be represented by a chromosome as:

$$Ch = \{R_a^1 \quad R_c^1 \quad R_a^2 \quad R_c^2 \quad ,..., \quad R_a^n \quad R_c^n\} \tag{5-12}$$

where *n* denotes the number of the rules in the rule-base. In this structure rules' antecedant parts R_a^i include residuals' membership values that are coded as linguistic variables such as big, medium, small etc. The rules' consequent parts R_c^i represent normalized faults degrees, which are known, between zero and one. The rule antecedent part can also be constructed by using residual histories as given in equation (4-8). The overall fitness calculation is done by *Fitness* function,

$$Fitness = \sum_{i=0}^{N-1} \left\| f_i^{created} - f_i^{calculated} \right\|^2 \tag{5-13}$$

where f_i shows fault vector. For calculating the fitness of each chromosome, the diagnoser decoded by the chromosome, is utilized. The steps of the genetic algorithm for the rule construction application can be given as:

1. Initialize a population of chromosomes with random 0’s and 1’s.
2. Assign the fitness value of each chromosome in the population.
    - Decode the chromosome.
    - Construct the rule-base.
    - Create the expert diagnoser by using this rule-base.
    - Simulate the system for creating faults.
    - Calculate the fitness value as in (5-13).
    - Assign this value to the chromosome as its fitness.
3. Put aside a few best chromosomes and update the stock with them.
4. Construct the new population from the old one using roulette wheel selection.
5. Apply crossover and mutation operations on the population.
6. If the predefined number of generations is not performed, go to step 2.
7. Take the best chromosome from the stock as the rule base constructed.

Based on our experiences on the rule-extraction with genetic algorithms, we can say that it does not work very well, because of the local minimum and convergence problems. The optimization algorithm sticks a local minimum while minimizing the cost function. Hence, other approaches should be used to construct a rule-base. If only analytic information is available on the fault-event relation, classification methods (i.e., k-means and fuzzy clustering, artificial neural network and probabilistic methods) can be used. If more information (i.e., analytic and heuristic information) about events and faults relation is available, expert knowledge and different methods of reasoning (i.e., probabilistic reasoning, probabilistic reasoning with fuzzy logic and reasoning with artificial intelligence [25], [27], [29], [32], and [54]) can be applied. The fuzzy logic reasoning (IF-THEN-rule system) is more appropriate. When fuzzy reasoning is utilized, it is possible to present the results in the form of possibility of faults and their sizes [33]. In the next section we will show how to construct a rule-base with k-means classification technique by using analytical knowledge.

5.2.4.3.2 Rule Extraction with K-means Clustering

K-means clustering can be best described as a partitioning method, which partitions the samples in the data set into mutually exclusive clusters. Unlike the hierarchical clustering methods, k-means clustering does not create a tree structure to describe the grouping in the data set, but rather creates a single level of clusters. Compared to hierarchical clustering methods, k-means is more effective for clustering large amounts of data. The number of clusters, *k*, needs to be determined at the onset. The idea behind k-means clustering is to divide the samples into *k* clusters such that some metric relative to the centroids of the clusters is minimized. Various metrics to the centroids that can be minimized include:

- maximum distance to its centroid for any sample,
- sum of the average distance to the centroids over all clusters,
- sum of the variance over all clusters,
- total distance between all samples and their centroids.

The metric to minimize and the choice of a distance measure will determine the shape of the optimum clusters.

Suppose we are given $X \in \Re^{m \times n}$, a set of m samples in n-dimensional space $\Re^n$ and an integer *k*. The problem is to determine a set of *k* centroids $\mu_1, \mu_2, ..., \mu_k$ in $\Re^n$, so as to minimize the sum-of-squares criterion

$$J = \sum_{c=1}^{k} \sum_{x_j \in S_c} \left\| x_j - \mu_c \right\|^2 \qquad (5\text{-}14)$$

where S_C denote set of centroids. A general algorithm is:

1. Randomly pick *k* samples in the data set as the initial cluster centroids $\mu_1^{(0)}, \mu_2^{(0)}, ..., \mu_k^{(0)}$. Set iteration *i* = 0.
2. Assign each sample x_j to the cluster with the nearest centroid $\mu_c^{(i)}$.
3. When all samples have been assigned, recalculate the positions of the *k* centroids

$$\mu_c^{(i+1)} = E\{x_j\}_{x_j \in \mu_c^{(i)}} \tag{5-15}$$

4. Repeat Steps 2 and 3 until the centroids no longer move.

The k-means algorithm uses an iterative procedure, which converges to one of the local minima. The computational complexity is *O* (*mnkT*), where T is the number of iterations. In practice, the number of iterations is generally much less than the number of samples. It is known that k-means are sensitive to initial starting conditions. Despite this limitation, the algorithm is used fairly frequently as a result of its ease of implementation. One way to find good optima is to do many runs of k-means, each from a different random starting point, and find the best minimum in terms of (5-14).

A rule-base is constructed by using k-means algorithm as follows. First, the residuals are obtained by using analytical knowledge. The k-means algorithm is applied to classify the residuals. In order to find good optima, k-means algorithm is repeatedly executed from a different random starting point. The best minimum is found among the results obtained from k-means runs in terms of (5-14). After deciding the number of the class centers *m*, the centers $\{C_1 \quad C_2 \quad ,..., \quad C_m\}$ are labeled as sub-events $\{e_1 \quad e_2 \quad ,..., \quad e_m\}$. As a result, a rule is constructed as:

IF $c_i\{e_i\}$ **THEN** q_i or **IF** $c_i\{e_i\}$ **THEN** Δq_i

To fuzzify the sub-events, triangular and trapezoidal membership functions (or others) are used. To construct a rule as in equation (4-8), the rules' antecedent parts can be composed of using sub-events such as $C_1 \quad C_2 \quad ,..., \quad C_N$ which are labeled as $\{e_1 \quad e_2 \quad ,..., \quad e_N\}$, where N is an integer related to the model order. A rule can be given by

Rule type 1: **IF** $C_1 \quad C_2 \quad ,..., \quad C_N \{e_1 \quad e_2 \quad ,..., \quad e_N\}$ **THEN** q_i

or,

Rule type 2: **IF** $C_1 \quad C_2 \quad ,..., \quad C_N \{e_1 \quad e_2 \quad ,..., \quad e_N\}$ **THEN** Δq_i

In rule type 2, the membership degree of the rule's antecedent part is calculated by multiplying the membership degrees with each other in the rules antecedent part. Next, those, which are too small, are set to zero membership degree. The remaining rule's antecedent parts are normalized among themselves. The consequent parts q_i's and Δq_i's of the rules are also known, because we assigned these faults to the system to generate residuals.

5.3 Conclusions

In this chapter, a brief introduction into the field of knowledge-based FDD is given, and it is shown how to apply the FDES and FDEDS to solve an FDD problem. A new fuzzy diagnosability definition and a relation between diagnosability and observability concepts are given. The FDES and FDEDS based FDD techniques are illustrated. A nonlinear and event based extension of the Luenberger observer is introduced. It is also shown that, by this way it is possible to construct a diagnoser. The FDES and FDEDS diagnoser design procedures are introduced, including an expert system. A new fuzzy event generation techniques and two rule extraction methods are addressed. The rules obtained with k-means clustering method are superior to the rules obtained with genetic algorithm based rule extraction in isolating faults. It is also addressed that how to obtain events by using analytical and heuristic knowledge. If no information about the fault-event relations is available a classification approaches can be used. Otherwise, If more information about the fault-event relations is available, a rule-based reasoning approach can be used.

CHAPTER 6

KNOWLEDGE-BASED FDD APPLICATIONS

6.1 Introduction

In the following section, several examples simulated are presented in order to test the FDD techniques studied in Chapter 5. The FDD techniques used are based on the FDES and FDEDS based approaches. In this thesis, we are mostly interested in component faults since it is a more challenging task to detect such kinds of faults than the actuator and sensor faults. Hence, single and multiple, additive, abrupt and incipient component faults are considered and simulated on the monitored systems. The following processes are described.

1. MIMO simulink model of an induction motor working with electrical mains.
2. MIMO simulink model of an unmanned helicopter flying a predefined trajectory.

6.2 FDD in an Induction Motor with FDES and FDEDS Approaches

6.2.1 System Architecture

An induction motor made by three stator windings and three rotor windings is given in Chapter 3. Here, we will show how to apply an FDES or FDEDS approach to the failure detection and diagnosis problem in an induction motor. The architecture is shown in figure 6.1. In this structure, the diagnoser unit can be constructed via the FDES or FDEDS approaches. In this structure, the events and states are fuzzy variables as mentioned before. Different types of FDES or FDEDS diagnoser structures and rule extraction methods can be proposed, but the aim is to show how to work the proposed FDES and FDEDS based failure diagnosis method. Hence a simple FDES and FDEDS structure and a rule extraction method are used in this study.

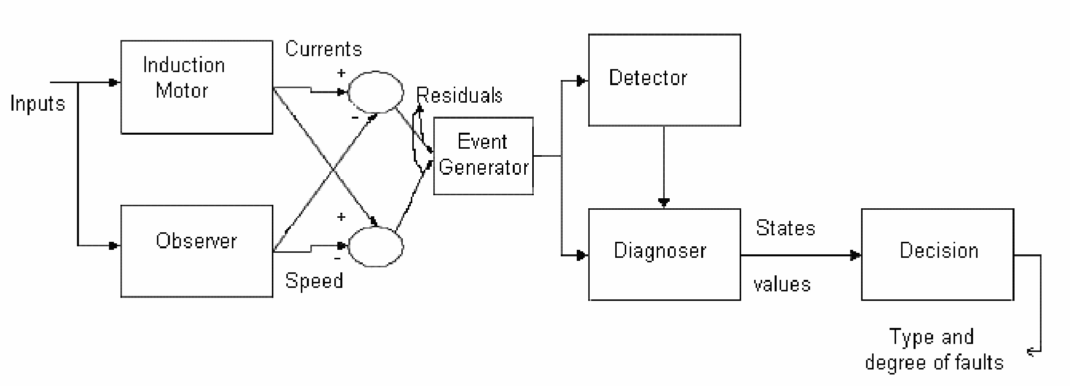

Figure 6.1: FDD system structure

By taking the difference between the system and observer outputs, current residuals and speed residual are defined. These residuals occurring continuously are in the time domain. In this structure, event generator labels motor current and speed residuals as events. Here, the observer unit can be constructed by various methods. It depends on the available analytic and heuristic knowledge about the system. For example, the mathematical model of the system can be used. However, a neural network, which simulates the input-output relation, can also be used as an observer. Depending on the events, detector produces an output. If there is a difference (i.e., the difference is normalized with respect to the nominal motor variables and it takes any value between -1 and 1. For instance, zero residual means there is no difference between measured signal and observer output) between the observer and system outputs or states, detector tells the diagnoser unit to start working. To prevent false alarms suitable threshold values are used in this unit.

In order to indicate a probable failure in the diagnoser; a fuzzy rule-base is used. The rules in the diagnosers are of the following form

IF event THEN C

The conclusion C represents the degree of failure in the system such as “low failure”, “medium failure” or “high failure”. The event represents the condition related to the residuals. In the diagnoser, events are combined and evaluated. Using a suitably designed event base system, inference procedures are performed in order to make a decision about failure types. Following the fuzzy inference procedure, the resulting membership function profiles are defuzzified. Next, they are compared with suitable thresholds (i.e., they are not necessarily the same). If one of them is above the associated threshold value, then the corresponding failure probabilities are made available at the output of the diagnoser. The overall FDD procedure can be summarized as:

1. Obtain residuals.
2. Label residuals as fuzzy events such as big, medium, small etc. (i.e., event generation).
3. Apply fuzzy events to the FDES or FDEDS diagnosers as inputs.
4. Obtain failure probabilities.

6.2.2 Event Generation

The event generator labels speed and current residuals. We called these labels as events. For example zero speed residual and zero current residual events are denoted by *e* for the FDES or FDEDS diagnosers. Table 6.1 shows these events. There are fuzzy events applied to the diagnoser. In Table 6.1, *ZR, SR, MR, BR* show zero residual, small residual, medium residual and big residual. If Table 6.1 is investigated further, one can see that there are 16 fuzzy events denoted by *e…u* to construct the FDES and FDEDS diagnosers. Here, the events have fuzzy values.

Table 6.1: Event for FDES and FDEDS diagnosers

Inputs of the Event Generator (EG)		Output of the EG
Speed Residual	Current Residual	
ZR	ZR	e
SR	ZR	f
MR	ZR	g
BR	ZR	h
ZR	SR	i
SR	SR	j
MR	SR	k
BR	SR	l
ZR	MR	m
SR	MR	n
MR	MR	o
BR	MR	p
ZR	BR	r
SR	BR	s
MR	BR	t
BR	BR	u

6.2.3 Rule Derivation

Rules in the diagnosers show the relationship between events and faults. The rule-base used in this case study is based on expert knowledge, which can be found in [6]. The information about the power loss could give intuition about the bearing condition. Overall loss in an induction motor consists of stator, rotor, core, stray-load, friction and winding losses. The frictions are mechanical losses due to the friction caused by bearing wear. Windage losses caused by the rotating elements of the motor comprise five to ten percent of the overall losses experienced by a healthy motor. If motor loss type can be calculated, rule derivation can be done, but these calculations are not easy. Thus, an alternative approach should be given to rule derivation. Firstly, it is known that bearing condition depends on motor speed. As bearing wear increases, the rotor speed is reduced. Furthermore, the motor cannot stay at high speed with increased bearing wear. This reduction in rotor speed increases the motor slip; as a result motor current increases. One can argue that this is not true in a low current low speed region or a high-current high-speed region (i.e., constant torque region). Actually the above condition itself excludes the possibility of an event associated with low current-low speed region or a high current-high-speed region [6]. The rule derivation about bearing faults is done by using the above knowledge via current and speed residuals (i.e., they are labeled as events). For example:

*"***If** *current residual is low and speed residual is low and state is* q_i **then** *the bearing condition is good or there is no change in the bearing condition".* Or, in general, *"***If** *an event occurs and state is* q_i, **then** *failure degree or change in failure degree"*

is the structure of a general rule to detect bearing condition and it is used to construct the FDES or FDEDS diagnosers.

Secondly, knowledge about winding insulation condition suggests that it is heavily dependent on motor current. This dependence is due to the Arrhenius life relation, which is dependent on the temperature of the stator or rotor windings. This relationship is expressed by the Arrhenius equation shown below [6].

$$\text{Life} = Ae^{\frac{B}{\theta}} \qquad (6\text{-}1)$$

where, *A* and *B* are determined by the properties of the insulating material used and θ is the absolute temperature in degrees Kelvin. The temperature of the windings is a direct result of the operating environment and the current going through the windings. Because of this, the motor current becomes the major contributing factor. Moreover rotor speed is not a necessary input. The rule derivation about winding failure due to insulation condition was done along with this information. For example, “**If** *current residual is low and speed residual is high and state is* q_i *,* **then** *winding failure is low or there is no change in winding condition* “. Or “**if** *an event occurs* **then** *winding failure degree or change in failure degree*”.

Finally, motor faults can be considered as a combination of bearing and stator winding failures. The basic idea is that, if a system produces residual then the motor has a failure. For example, a rule for motor failure can be given as follows.

*“***If** *zero speed residual and zero current residual occur and state is* q_i *,* **then** *zero motor failure or there is no change in motor condition”. Or “***if** *event occurs* **then** *motor failure degree or change in motor failure degree”.*

There are 127 rules in the FDES and FDEDS diagnosers to isolate and give fault priority and degree about induction motor bearing and stator winding faults. The set of rules is not given in here but if the diagnosers are in state 1 or 2 or 3 (i.e., see Figure 6.2) and the events occur as in Table 6.1, at this circumstance the rules will be as in Table 6.2 and 6.3 for FDES and FDEDS diagnosers. In these tables, *ZF, VSF, SF, MF, BF, VBF, ZCF*, *VSCF, SCF, MCF, BCF, VBCF* denote zero failure, very small failure, small failure, medium failure, big failure and very big failure, zero failure change, very small failure change, small failure change, medium failure change, big failure change and very big failure change, respectively.

Table 6.2: Some rules for FDES diagnoser

	Events		Bearing Failure	Winding Failure	Motor Failure
IF	e	***THEN***	ZF	ZF	ZF
	f		VSF	ZF	ZF
	g		SF	ZF	SF
	h		MF	ZF	SF
	i		ZF	VSF	ZF
	j		SF	VSF	SF
	k		MF	SF	MF
	l		MF	SF	MF
	m		ZF	SF	SF
	n		SF	MF	MF
	o		BF	MF	BF
	p		VBF	MF	BF
	r		ZF	BF	BF
	s		SF	BF	BF
	t		BF	BF	BF
	u		VBF	VBF	BF

Table 6.3: Some rules for FDEDS diagnoser

	Events		Change in Bearing Failure	Change in Winding Failure	Change in Motor Failure
IF	e	***THEN***	ZCF	ZCF	ZCF
	f		VSCF	ZCF	ZCF
	g		SCF	ZCF	SCF
	h		MCF	ZCF	SCF
	i		ZCF	VSCF	ZCF
	j		SCF	VSCF	SCF
	k		MCF	SCF	MCF
	l		MCF	SCF	MCF
	m		ZCF	SCF	SCF
	n		SCF	MCF	MCF
	o		BCF	MCF	BCF
	p		VBCF	MCF	BCF
	r		ZCF	BCF	BCF
	s		SCF	BCF	BCF
	t		BCF	BCF	BCF
	u		VBCF	VBCF	BCF

6.2.4 FDES and FDEDS Diagnoser Structures

In this sub section, the diagnosers will be constructed by using FDES and FDEDS frameworks. The FDES and FDEDS diagnoser structure will be the same. However, the rule-base used in the diagnosers will be different as explained in Chapter 4. In the diagnosers, places indicate whether a system is faulty or not. Figure 6-2 shows the diagnoser structures. A place description of the diagnosers is in Table 6.4. We didn't show all the implications and events' names in Figure 6.2 to avoid confusion. Let us assume that initially the system is in places 1, 2 and 3 with membership degrees 0.7, 0.8 and 0.9 (i.e., no failure in the system). If the event generator produces the event *u* and *t* with membership degrees 0.7 and 0.3, respectively (i.e., others are zero), then the diagnoser places will be 4, 5 and 6 at the same time because of the rules. Since event *u* and *t* show medium and big speed and current residual, we conclude that the motor stator winding and bearing have failures.

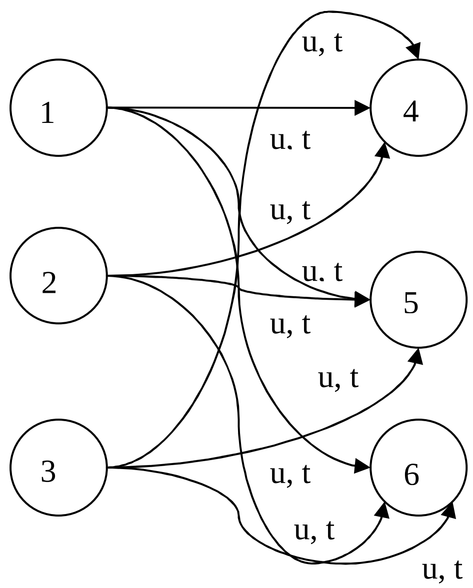

Figure 6.2: FDES and FDEDS diagnoser structure.

Table 6.4: Description of the diagnoser places

DES diagnoser places	Description of places
1	Zero motor failure
2	Zero bearing failure
3	Zero winding failure
4	Motor failure
5	Bearing failure
6	Winding failure

The computation of the diagnoser next state $\bar{q}$ can mathematically be best understood in the following example: suppose that the present state is $\bar{q}_0$ and events *t,* and *u* have occurred with membership degrees 0.3 and 0.7 respectively, as in Figure 6.2 (i.e., all remaining events have zero membership degrees). Then state transition matrices related to events whose membership values are not zero are given as (i.e., state transition matrices related to events whose membership values are assumed to be zero):

$$\bar{q}_0 = \begin{bmatrix}0.6 & 0.8 & 0.9 & 0 & 0 & 0\end{bmatrix}$$

$$\bar{f}(t,\bar{q}_0) = \begin{bmatrix} 0 & 0 & 0 & 0.3 & 0.3 & 0.3 \\ 0 & 0 & 0 & 0.3 & 0.3 & 0.3 \\ 0 & 0 & 0 & 0.3 & 0.3 & 0.3 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \bar{f}(u,\bar{q}_0) = \begin{bmatrix} 0 & 0 & 0 & 0.7 & 0.7 & 0.7 \\ 0 & 0 & 0 & 0.7 & 0.7 & 0.7 \\ 0 & 0 & 0 & 0.7 & 0.7 & 0.7 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Then, to calculate next fuzzy state ($\bar{q}$) we used fuzzy max ($\circ$) operation as:

$$\bar{q} = \max[(\bar{q}_0 \circ \bar{f}(t,\bar{q}_0)), (\bar{q}_0 \circ \bar{f}(u,\bar{q}_0)]$$

$$= \max[(0 \quad 0 \quad 0 \quad 0.3 \quad 0.3 \quad 0.3), (0 \quad 0 \quad 0 \quad 0.7 \quad 0.7 \quad 0.7)]$$

$$= [0 \quad 0 \quad 0 \quad 0.7 \quad 0.7 \quad 0.7]$$

where (∘) denotes fuzzy max-min operation. The result shows that the diagnosers are in states 4, 5 and 6 with membership degrees 0.7 at the same time after occurring events *u* and *t* with membership degrees 0.7 and 0.3, respectively.

6.2.5 Simulation Results

The overall FDES and FDEDS architectures are tested based on three different (i.e., single and multiple faults) scenarios using a simulation program.

Scenario I

In this scenario, while the system was working in real time, 10 % incipient bearing fault is created by changing the motor friction coefficient gradually from 0 to 1, between 2.4 and 3 seconds. The plots are in Figures 6.3 and 6.4.

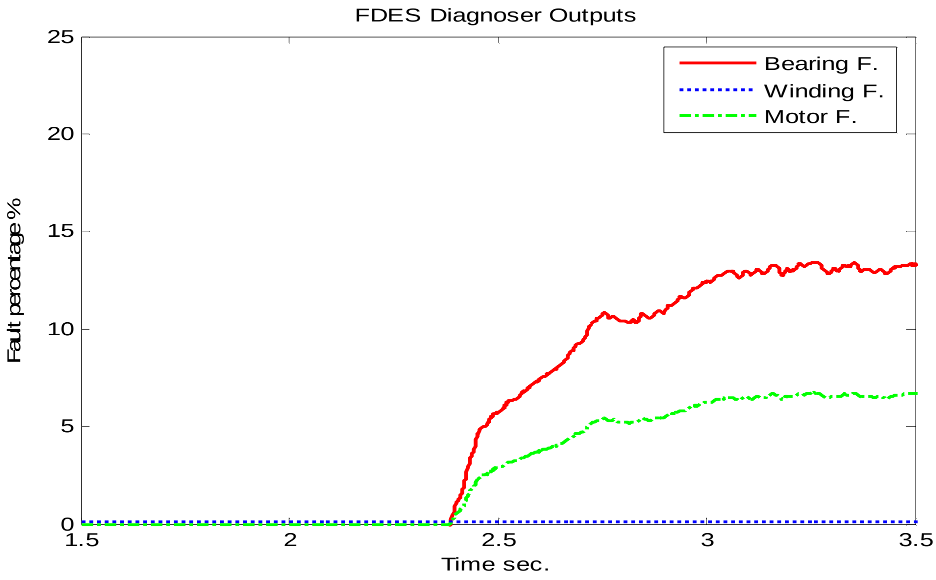

Figure 6.3: FDES diagnoser outputs.

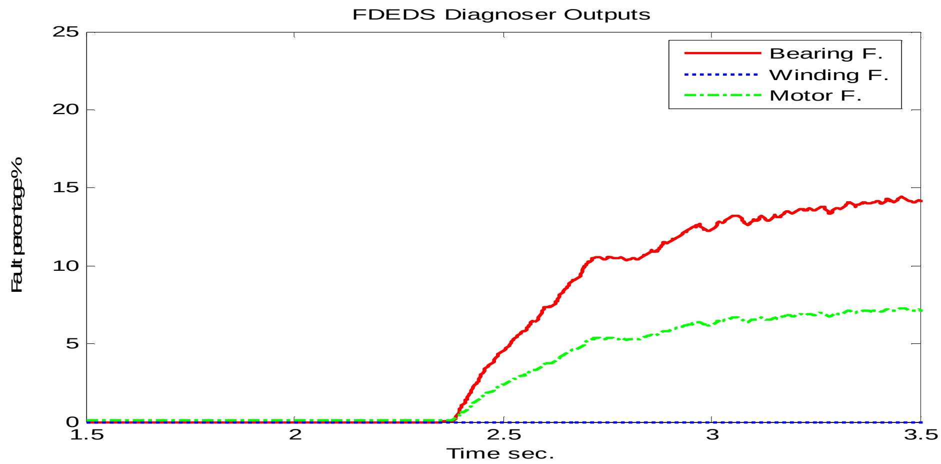

Figure 6.4: FDEDS diagnoser outputs.

Scenario II

In this scenario, while the system was working in real time, abrupt winding fault is created by decreasing the stator winding by an amount of 10 % amount at time 2.35 sec. Figures 6.5 and 6.6 denote the results obtained.

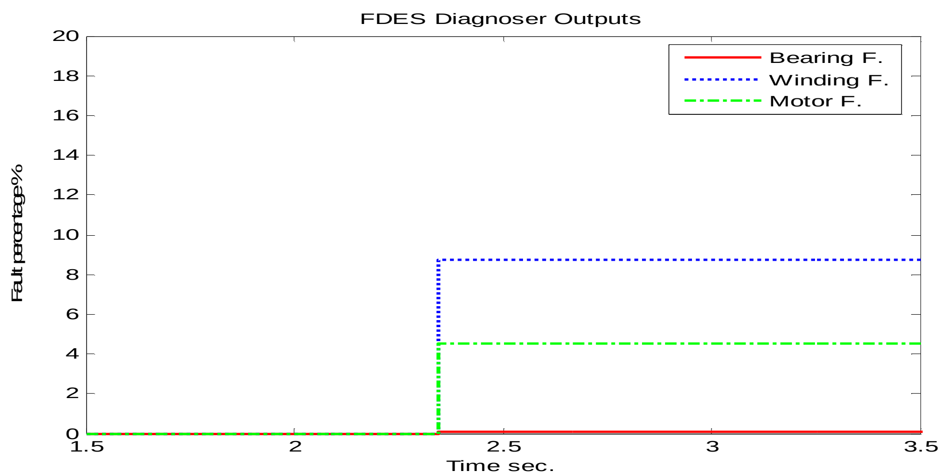

Figure 6.5: FDES diagnoser outputs.

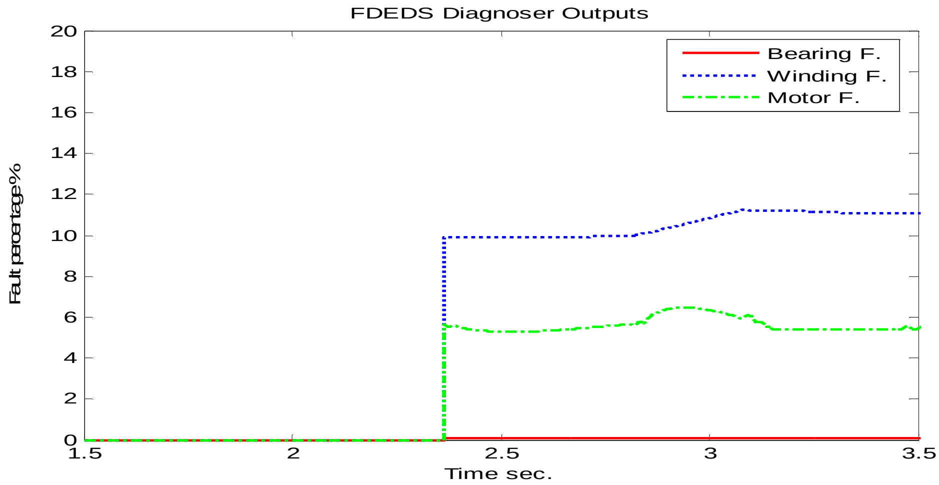

Figure 6.6: FDEDS diagnoser outputs.

Scenario III

In this scenario, while the system working in real time, abrupt winding and bearing faults are created at the same time (2.35 sec) by decreasing the stator winding by an amount of 5 % and changing friction coefficient from zero to 5 (i.e., multiple fault case). Figures 6.7 and 6.8 show the results obtained.

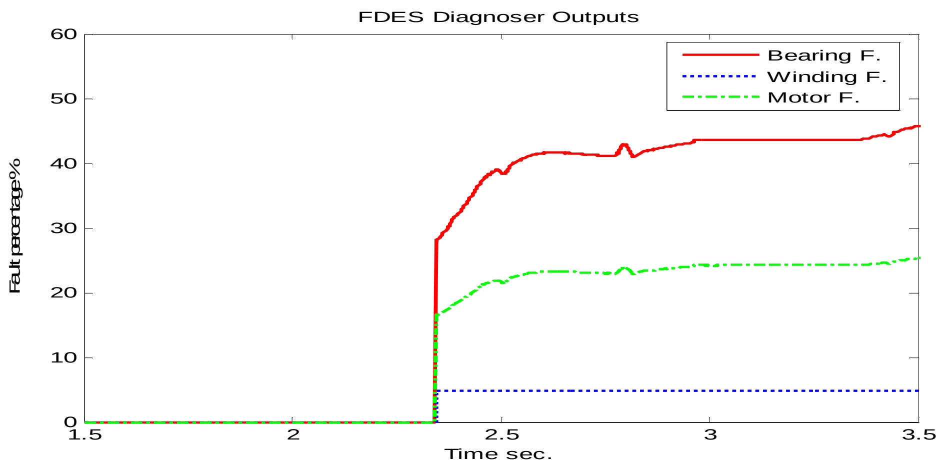

Figure 6.7: FDES diagnoser outputs.

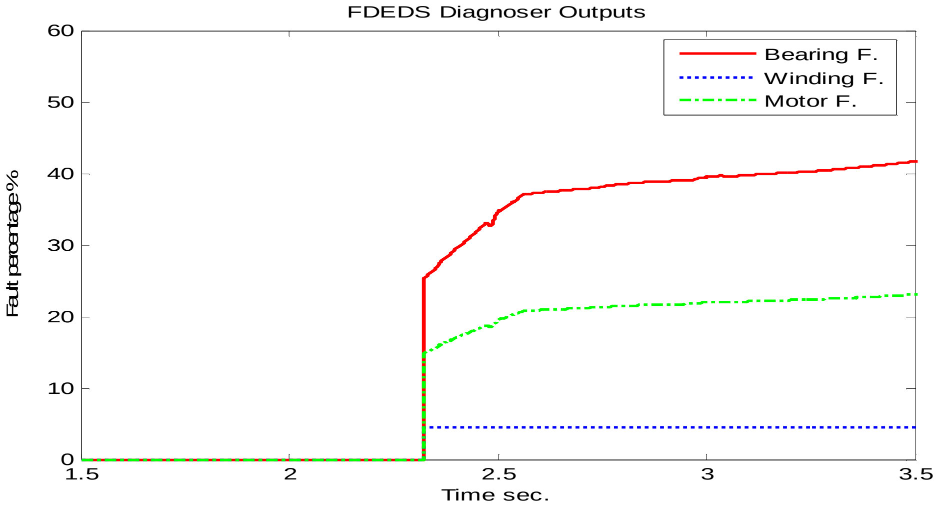

Figure 6.8: FDEDS diagnoser outputs.

As observed from the figures, the failure probabilities produced by the FDES and FDEDS diagnosers for the different components show expected variations over the time horizon.

6.3 FDD in an Unmanned Small Helicopter with FDES Approach

In literature one can find many fault diagnosis methods about helicopters. Many studies [5], [13], [20] are focused on detecting and identifying helicopter gearbox faults. Signal analysis techniques with pattern classification (Kalman filter approach [6], decision trees, learning vector quantization, multi-layer perceptrons, fuzzy ARTMAP, and Gaussian mixtures [20]) are used to diagnose helicopter gearbox faults. Feature vectors used to isolate helicopter faults are based on neuro-fuzzy system, reasoning [15] and signal processing approaches [13], [20] (Mean square RMS, kurtosis maximization). Two studies related to the actuator fault compensation and actuator and sensor faults in a helicopter can be seen in [14] and [6], respectively. In this case study to solve the fault diagnosis problem, an FDES approach based on a fuzzy rule-base is proposed. The approach proposed has been applied to a failure diagnosis problem in an unmanned small helicopter. In order to construct IF-THEN rules (i.e.

events-fault relations), the k-means classification algorithm is used since it is simple.

6.3.1 System

The continuous-time 6 degrees of freedom linear model of a helicopter can be described by the following dynamical state equations [4], [14]:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \qquad (6\text{-}2)$$

The state variable vector *x* (dimension 9) is defined as:

$$x = [u \quad v \quad w \quad p \quad q \quad r \quad \varphi \quad \theta \quad \psi]^T \qquad (6\text{-}3)$$

where *u, v* and *w* represent the longitudinal, lateral and vertical velocities (ft/sec), respectively; *p, q* and *r* represent the roll, pitch and yaw rates (rad/sec) respectively; φ, θ represent the roll and pitch attitude (rad); and ψ is the heading (rad). There are five control inputs, i.e., $u = [u_1 \quad u_2 \quad u_3 \quad u_4 \quad u_5]^T$ where u_1 is the lateral stick (inch), u_2 is the longitudinal stick (inch), u_3 is the collective stick (inch), u_4 is the pedal position (inch) and u_5 is the horizontal tail incidence (degree).

6.3.2 Fuzzy FDD Method

In order to perform fuzzy FDD method in an efficient way we should take into account all possible fault types occurring in an unmanned small helicopter, which means that the faults must not prevent the helicopter to continue its execution. The proposed FDD method consists of 2 stages: off-line rule extraction and on-line fault detection and identification.

6.3.2.1 Off Line Rule Extraction

Figure 6.9 denotes the off-line rule extraction procedure. The system’s fault model (i.e., system model including faults) and healthy model is used together to generate residuals. After generating residuals the k-means algorithm is applied to classify them (i.e., the system parameters or some variables related to sticks or percentage actuator faults are changed in a simulation program within 1 second time interval).

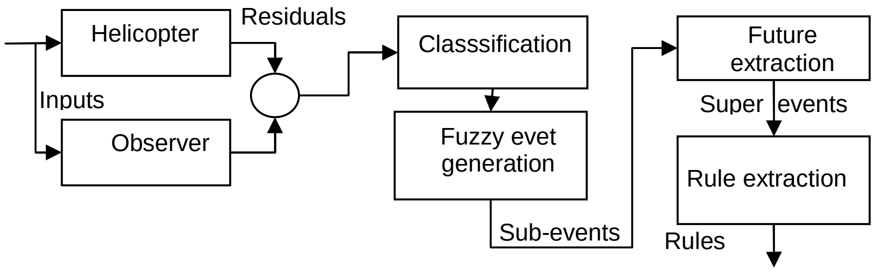

Figure 6.9: Off-line rule extraction procedure

Many simulations are done to decide the number of the class centers. After examining the relation between the obtained cost value and used class centers in the k-means classification algorithm, 101 centers (classes) are chosen. These centers are called sub-events and used as rules' antecedent parts. To fuzzify sub-events, triangular and trapezoidal membership functions (or others) are used. Rules' antecedent parts contain 20 sub-events (i.e., an event sequence). These event sequences are called feature vectors (i.e., super events). These super events are related to single, double and triple faults. As a result, the rule-base consists of 118 rules (i.e., the data base employed includes 118 different fault types).

6.3.2.2 On Line FDD

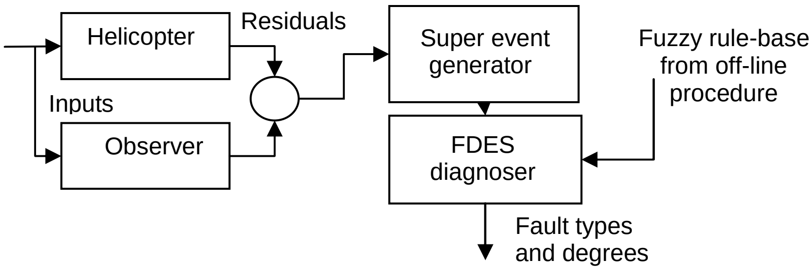

Figure 6.10: On-line FDD procedure

Figure 6.10 denotes on-line FDES based FDD procedure. There are 6 sensors in the system. They are used to measure helicopter body speeds (*u, v*, and *w*) and angular velocities (*p, q*, and *r*) on real time. These sensor outputs are used to obtain residuals. The super event generator labels the residuals as sub-events and create a super event (an event sequence) by taking past 20 sub-events in time period. The sub-events membership degrees are calculated by using Euclidean distance measure (the distance of measurement vectors to the predetermined class centers). Next, those, which are too small, are set to zero membership degree. The remaining sub-events are normalized among themselves. The membership degrees of super events are also calculated in this part as follows: first, the last 20 (length is the same as super events) sub-events (taking place in the related super events) membership degrees are multiplied with each other. Next, those, which are too small, are set to zero membership degree. The remaining super events are normalized among themselves. The super events have a dynamic structure; their membership degrees change with respect to time. Finally, the super events are applied to the FDES diagnoser as inputs. The FDES diagnoser is contains 12 fuzzy places (i.e., places show the system's faulty components related to the 4 different parameters and actuators) based on rules derived using k means classification technique in off-line stage. It isolates faults and also gives information about the percentage of the occurred fault types.

A fuzzy rule-base is used to isolate faults in the diagnoser. A typical rule structure is as follows:

$$\mathbf{R^i: IF}\ e_t,\ldots,e_{t-N} \text{ is } A^i_1\ \mathbf{THEN}\ (q_1 \text{ is } C^i_1,\ldots,q_m \text{ is } C^i_m),\ i=1,2,\ldots,n \qquad (6\text{-}4)$$

Here, N is the integer related to the model order and, $e_t \ldots e_{t-N}$ denote the past input vector related to the sub-events, $q_1 \ldots q_m$, show states A^i_1 and $C^i_1 \ldots C^i_m$ are linguistic values (labels) represented as fuzzy subsets of the respective universes of discourse. The diagnoser outputs are degree of faults (i.e., fault percentage) related to the faulty components. This is accomplished by using the COA defuzzification.

6.3.3 Simulation Results

In this study, two types of actuator faults called percentage and stuck actuator faults are created and used in the FDD algorithm. The overall FDES based FDD method is tested on three different scenarios using a simulation program. In the first scenario, we created abrupt faults. There is no fault between 0 and 0.02 second. 50 % abrupt faults in parameter 3 and parameter 4(i.e paramaters related to the helicopter main rotor blades) are created at time 0.02 second. Figure 6.11 shows simulation result obtained by using the FDES diagnoser.

In the second scenario, 50 % incipient faults in parameter 3 and parameter 4 and 20 % incipient fault in parameter 5 are created at time 0.02 sec. Figure 6.12 shows simulation result obtained by using the FDES diagnoser. As observed from the figure, the small incipient parameter fault can not be detected by the FDD algorithm. It stays under a predefined threshold levels.

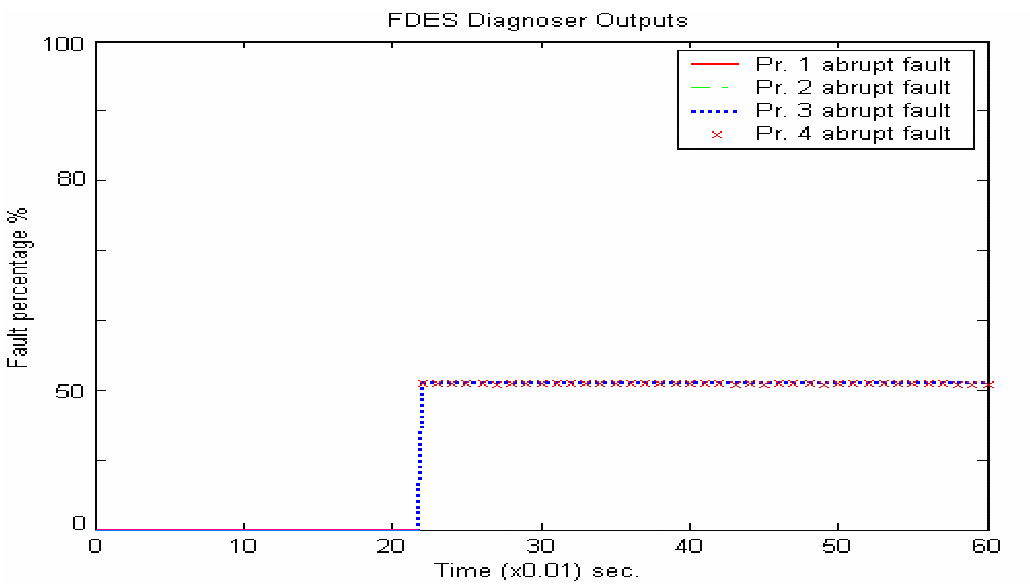

Figure 6.11: Multiple abrupt fault case

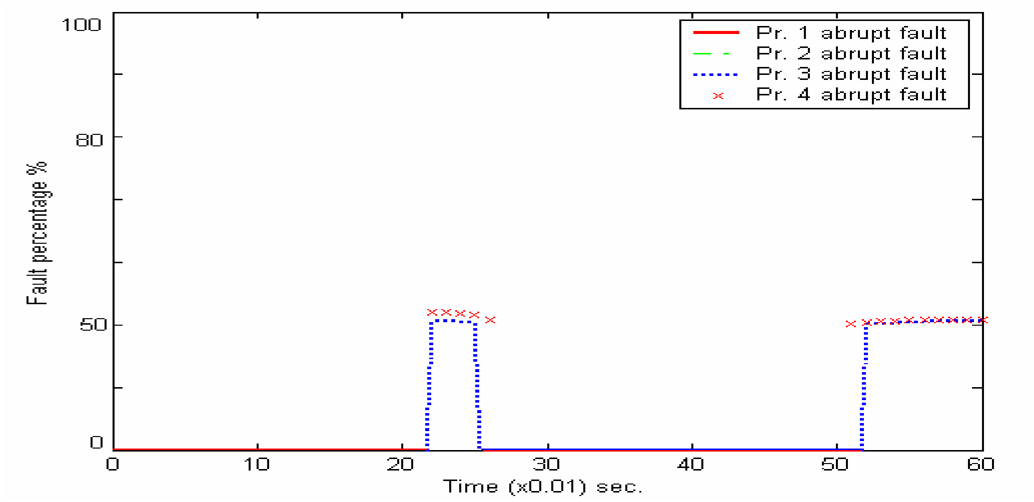

Figure 6.12: Multiple incipient fault case

In the third scenario, again multiple faults are created. There is no fault between 0 and 0.02 seconds. 50 % abrupt faults in parameter 3 and parameter 4 and 50 % stuck fault in actuator 1 (i.e., actuator begins to work normally, at the instant *t*; it sticks and remains at working condition of time *t*) are created at time 0.02 second. Figure 6.13 shows simulation result obtained by using the FDES diagnoser.

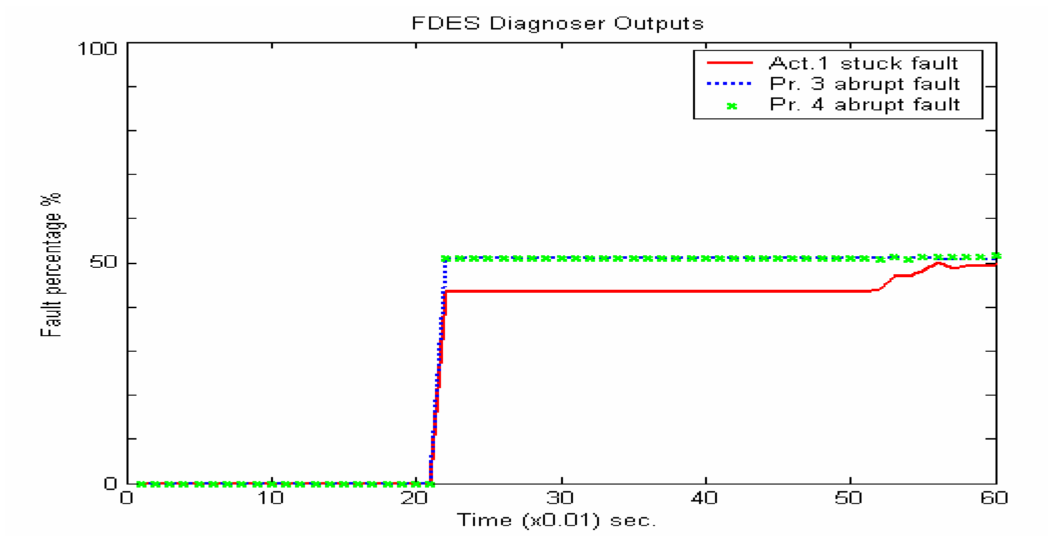

Figure 6.13: Multiple abrupt fault case

As observed from the figure, the fault probabilities obtained by the FDD algorithm for the different components show expected variations over the time horizon.

6.4 Conclusions

In this part of the thesis, the application of the FDES and FDEDS to the FDD problem was tested on an induction motor and unmanned small helicopter. The fuzzy rule-base employed is constructed using event-fault relations and k-means clustering algorithm. The results obtained shown that the FDES or FDEDS based FDD approaches can be used to isolate faults. Moreover, they can be used to give priorities and degrees about faults.

CHAPTER 7

CONCLUSION AND FURTHER RESEARCH

In this thesis three model-based and two new knowledge-based FDD methods, which are proposed first here, are suggested and implemented.

The main challenge in model-based FDD is to diagnose incipient faults in nonlinear dynamic systems under the assumption that input and output measurements are affected by the disturbances caused by faults occurring in a system. In the thesis, three model-based FDD approaches are introduced. They are the FDD methods utilizing PCA, the system identification and the inverse model based approaches. These techniques have been applied on a quadruple tank system, an induction motor and a gas pipeline system.

In the PCA-based fault detection algorithm, it might be enough to use only output measurements, which might provide enough information to detect certain types of faults. Moreover, it can also be argued that the PCA-based fault detection method can also provide satisfactory results for certain non-linear systems, for which the linear approximation captures most of the dynamics. Also we can say that the proposed PCA based FDD method is robust. By using this method to isolate faults, an adaptive threshold should especially be used in single fault cases. However, in multiple fault cases threshold based fault isolation techniques are not good enough. Hence, intelligent residual classification techniques or rule-bases to isolate faults should be used.

The system identification based FDD techniques are flexible, easy to implement and can be applied to a wide range of nonlinear dynamic systems as long as the systems states are available or they are observable. Applying constraints to the identification problem may also allow for additional information to be incorporated in the fault detection scheme, and can increase the power of the fault detection process. It is also possible to give priority about faults using the identification based FDD approach. The main drawbacks in the identification-based method are the local extremums of the associated global optimization problems and the requirement to know all system states. Moreover one can face convergence problems.

The inverse model based FDD techniques are also flexible, easy to implement and can be applied a wide range of nonlinear static systems. But a difficulty with the inverse model based FDD approach is the number of unknowns. Sometimes, it may not possible to determine faults if the number of unknown variables is greater than the number of equations in the model.

Another interesting topic discussed in this thesis is the knowledge-based FDD approaches. These methods are based on the FDES or FDEDS structures using IF-THEN rules. In the thesis, a brief description of these structures are given, and shown how to apply to solve an FDD problem. The FDD methods proposed were tested on an induction motor and an unmanned small helicopter. These approaches can be considered as an expert system application. In these approaches it is possible to assign priorities and degrees about faults.

The main problem in the FDES or FDEDS based approaches is to construct rules and obtain events. To cope with these difficulties a new fuzzy event generation technique and a classification based rule extraction method are proposed in the thesis. It is shown that, the clustering based event generation technique produces a satisfactory solution for the rule extraction problem. The FEDS or FDEDS are general structures. A system can be modeled as a nonlinear dynamic system by using these structures.

Moreover, components of a nonlinear system can be modeled separately and combined by using these structures. One can easily see that these new knowledge-based FDES and FDEDS based FDD methods are very simple.

Finally, in this thesis a model and an expert based diagnoser design procedures are represented. To construct a model-based diagnoser to isolate faults, a nonlinear extension of the Luenberger observer is implemented. It is also shown how to adjust the gain parameter of this diagnoser by using a gradient-based optimization method. To construct an expert system based diagnoser to isolate fault, an event base extension of the Luenberger observer is also introduced. The faults are handled as system inputs. Thus, the diagnosibility problem is taken as an observability problem. Hence, the nonlinear type Luenberger observers can be used as diagnosers.

Classical FDD methods are based on systems mathematical model such as FDD method utilizing PCA, system identification and inverse model based FDD. These methods handle generally single faults and do not give priority and degree about fault types. When we compare the results obtained by using the model and the new knowledge-based FDD methods, we can say that the proposed knowledge-based FDD techniques are more satisfactory than the model-based techniques. It is possible to detect single and multiple faults using these methods and assign priorities and degrees about fault types. Especially in the multiple fault cases knowledge-based methods perform better than model-based methods. In order to construct a model-based FDD method, mathematical models of nonlinear systems are needed. This assumption has limited the success in practical applications. A perfectly accurate mathematical model of a physical system is never available. Usually, the parameters of a system may vary with time and the characteristics of the disturbances and noises are unknown so that they cannot be modeled accurately. Hence, information obtained is incomplete or uncertain in nonlinear complex systems. It is essential to deal with the incomplete knowledge in an efficient way. A more suitable solution may be utilizing the knowledge-based techniques. In model-based FDD approaches the system states must be known perfectly or predicted. If this is not possible, fault-event relation, classification methods are used especially with neural network and fuzzy logic. If more information between events and faults is available, IF-THEN fuzzy rule-based reasoning can be applied. In this thesis, the newly proposed event based FDD methods based on FDES and FDEDS structures used fuzzy rules. The fuzzy rule-base employed is constructed using event-fault relations and k-means clustering algorithm. Actually, the dynamic aspects of the failure diagnosis depends on the definition of “fault” events; if those events are based on time histories, then the resultant FDES can be considered as a dynamical system. Theoretically, almost all of the ideas, definitions, algorithms, results, etc., will be the same in such a case, but obviously derivations are more complicated.

The observability and diagnosibility concepts related to the proposed FDES and FDEDS structures have been illustrated in the thesis. A new fuzzy observability and diagnosibility definition and relation between these concepts have also been given. It is possible to give observability and diagnosibility degrees by using these new definitions. It is not easy to check a system’s observability by using the observability definition. To overcome this difficulty a simple observability checking method is proposed, which is different from the classical observability concept. By using this new observability checking method one can easily check a system’s observability.

It is important to note that, the results obtained by using the FDES or FDES structures are of a general nature and are applicable not only to particular systems but also to a wide class of nonlinear complex dynamic systems. We believe that the knowledge-based approaches introduced here will show researchers a new way to cope with the FDD problem in nonlinear dynamic systems.

The research in this thesis has inevitably had to end before all the interesting topics for future FDD research could be explored. In the following, we will enumerate the most important topics for future research as long as FDD is concerned.

There are mainly two stages in an FDD: 1) Residual generation 2) Fault isolation. The main problem in model-based or knowledge based FDD is the residual generation. Most residual generation techniques are based on linear system model. For nonlinear systems, the traditional approach is to linearize the model around the system operating points. However, for systems with high nonlinearity and a wide dynamic operating range, the linearized approach fails to give satisfactory results. One solution is to use a large number of linearized models corresponding to a range of operating points. This means that a large number of FDD schemes corresponding to each operating points are needed. Hence, it is important to study residual generation techniques, which tackle nonlinear systems directly. There are some studies on the residual generation of nonlinear, complex, dynamic systems. There have been some attempts to use nonlinear observers to solve the nonlinear FDD problem, e.g., nonlinear unknown input observers including adaptive and sliding mode observers. In this thesis, the proposed nonlinear and event based extension of Luenberger observer can be used for this purpose. On the other hand, the analytical model, which the nonlinear observer approaches are based on are not easy to obtain in practice. Sometimes it is not possible to obtain an explicit mathematical model. To overcome this problem, it is desirable to find a universal approximate model, which can be used to represent real systems with an arbitrary degree of accuracy. Different approaches were proposed and they are currently under investigation [63] such as neural networks, fuzzy models, hybrid models and linear parameter varying (LPV) models.

Neural networks are powerful tools for handling nonlinear systems. Therefore, they are very suitable to deal with fault diagnosis problems. Up to know they have mainly exploited as fault classifier with steady state processes. Neural networks can also be used as residual or event generators, as models of nonlinear dynamic systems [63].

Fuzzy models can be used as residual or event generators. A nonlinear dynamic process can be described as a composition of Takagi Sugeno models. The main idea is to exploit the constructed models as an observer. Then the residuals are computed by the fusion of local residuals. The residuals can be considered as linguistic variables and labeled as events.

Hybrid models can be used to describe the behavior of any nonlinear dynamic system if they are described as a composition of several local affine models selected according to the system operating conditions. Instead of exploiting complicated nonlinear models obtained by nonlinear modeling techniques, it is possible to describe the plant by a collection of affine models. Such a compound system requires the identification of the local models from data. After the system model is constructed by these hybrid modeling techniques, they can be used as observers to produce residuals or events. The residuals are computed by fusing of local residuals. They can be used to generate events by treating them as linguistic variables.

Lastly, LPV models can be used to produce residuals or events. LPV models are powerful linear design tools for stability and performance of systems. It can be applied to complex systems. After system models are constructed by LPV modeling techniques, they can be used as observers to produce residual events. Adaptive residual generation techniques for nonlinear system can also be proposed, but these approaches are a lot more complex [63]. There are some ways that can be to decrease the complexity of the adaptive residual generation techniques, however, they require complicated estimation algorithms.

After obtaining the residuals or events, the data should be used to construct a rule base to isolate faults. Clustering techniques are mostly unsupervised methods that can be used to organize data into groups based on similarities among the individual data items. Different fuzzy clustering techniques can be exploited to convert data into a fuzzy rule-base. The rule base obtained can be improved by using neuro-fuzzy approaches.

REFERENCES

[1] Willsky A.S., “A survey of design methods for failure detection systems”, Automatica, vol. 12, pp. 601-611, 1976.

[2] Isermann R., Supervision, fault-detection and fault diagnosis methods-an introduction, Control Eng. Practice, vol. 5, no. 5, pp. 639–652, 1997.

[3] Ayoubi M., Isermann R., “Neuro-fuzzy systems for diagnosis”, Fuzzy Sets and Systems, no. 89, pp. 289-307, 1997.

[4] Rich S., Venkatasubramanian V., “Model based reasoning in diagnostic expert systems for chemical process plants”, Computer Chem. Eng., vol. 11, pp. 11-122, 1987.

[5] Nandi S., Toliyat H.A., “Fault diagnosis of electrical machines-A review”, Technical Report, Texas University, 1998.

[6] Goode P.V., Chow M.Y., “Using a neural/fuzzy system to extract heuristic knowledge of incipient faults in induction motors”, IEEE Trans. on Indust. Electr., vol. 42, no. 2, pp. 139-145, April 1995.

[7] Racoceanu D., Minca E., Zerhouni N., “Fuzzy Petri nets for monitoring and recovery”, Robotics and Automation, Proceedings. ICRA'03, IEEE International Conf., vol. 3, pp. 4318–4323, Taipei, Taiwan, September 14-19, 2003.

[8] Sampath M., et al, “Failure diagnosis using discrete event models”, IEEE Trans. on CST., vol. 4, no. 2, pp. 105-124, 1996.

[9] Prock J., “A new technique for fault detection using Petri nets”, Automatica, vol. 27, no. 2, pp. 239-245, 1991.

[10] Zadeh L.A., “Fuzzy sets”, Inform. Control, vol. 8, pp. 338-353, 1965.

[11] Santos E.S., “Max-min automata”, Inform. Control, vol. 13, pp. 363-377, 1968.

[12] Passino K.M., Yurkovich S., “Fuzzy Control”, Addison Wesley Longman Inc, 1998.

[13] Lin F., Ying H., “Fuzzy discrete event systems and their

observability", Proceeding Joint Int. Conf. of 9th Int. Fuzzy Systems Assoc. World Congr. 20th North Amer. Fuzzy Inform. Processes Soci., Vancouver, BC, Canada, pp. 1271-1276, July 25–28, 2001.

[14] Lin F., Ying H., "Modelling and control of fuzzy discrete event systems", IEEE Trans. on Syst. Man and Cybernetics, vol. 32, no. 4, pp. 408-415, August 2002.

[15] Lin F., Wonham W.M., "On observability of discrete event systems", Information Sciences, vol. 44, no. 3, pp. 173-198, 1988.

[16] Özveren C.M., Willsky A.S., "Observability of discrete event dynamic systems", IEEE Trans. on AC., vol. 35, no. 7, pp. 797-806, 1990.

[17] Lin F., "Diagnosibility of discrete event systems and its applications", J. DEDS, vol. 4, no. 2, pp. 197-212, 1994.

[18] Sampath M., et al., "Diagnosibility of discrete event systems", IEEE Trans. on AC., vol. 40, no. 9, pp. 1555-1575, 1995.

[19] Marino R., et al., "Adaptive input-output linearizing control of induction motors", IEEE Trans. on Automatic Control, vol. 38, no. 2, pp. 208-221, February 1993.

[20] Cassandrass C. G., Lafortune S., "Introduction to Discrete Event Systems", Boston, MA: Kluwer, 1999.

[21] Isermann R., Ballĕ P., "Trends in the application of model-based fault detection and diagnosis of technical processes", Control Eng. Practice, vol. 5, no. 5, pp. 709–719, 1997.

[22] Fuessel D., Isermann R., "Hierarchical motor diagnosis utilizing structural knowledge and self-learning neuro-fuzzy scheme", IEEE Trans. on Industrial Electronics, vol. 47, no. 5, pp. 1070-1077, October 2000.

[23] Isermann R., "On fuzzy logic applications for automatic control, supervision and fault diagnosis", IEEE Trans. on Systems, Man and Cybernetics – Part A: Systems and Humans, vol. 28, no. 2, pp. 221-235, March 1998.

[24] Dubois D., Prade H., "An introduction to fuzzy systems", Clinica Chimica Acta, no. 270, pp. 3-29, 1998.

[25] Dubois D., Prade H., "Fuzzy set-based methods in instance-based reasoning", IEEE Trans. on Fuzzy Systems, vol. 10, no. 3, pp. 322-332, June 2002.

[26] Cardoso J., Valette R., Dubois D., “Possibilistic Petri nets”, IEEE Trans. on Systems, Man and Cybernetics – Part B: Cybernetics, vol. 29, no. 5, pp. 573-582, October 1999.

[27] Cayrac D., Dubois D., Prade H., “Handling uncertainty with possibility theory and fuzzy sets in a satellite fault diagnosis application”, IEEE Trans. on Fuzzy Systems, vol. 4, no. 3, pp. 251-269, August 1996.

[28] Ulieru M., Norrie D., “Fault recovery in distributed manufacturing systems by emergent holonic re-configuration: a fuzzy multi-agent modeling approach”, Information Sciences, no. 127, pp. 101-123, 2000.

[29] Wang H., et al., “A framework of fuzzy diagnosis”, IEEE Trans. on Knowledge and Data Eng., vol. 16, no. 12, pp. 1571-1582, December 2004.

[30] Qiu D., “Supervisory control of fuzzy discrete event systems: a formal approach”, IEEE Trans. on Systems, Man and Cybernetics – Part B: Cybernetics, vol. 35, no. 1, pp. 72-88, February 2005.

[31] Ying H., et al., “A fuzzy discrete event system for HIV/AIDS treatment planning”, IEEE International Conf. on Fuzzy Systems, vol. 1, pp. 197-202, Budapest, Hungary, 25-29 July, 2004.

[32] Ulieru M., Isermann R., “Integrated fault detection and diagnosis”, IEEE/SMC Conf. on System Engineering Service Humans, vol. 1, pp. 815-835, Le Touquet, France, 1993.

[33] Ulieru M., “Fuzzy logic in diagnostic decision: Probabilistic networks”, Ph.D. dissertation, Technical University of Darmstadt, Germany, 1996.

[34] Karasu Ç., ”Small-size unmanned model helicopter guidance and control”, M.Sc. thesis, Middle East Technical University, Ankara, Turkey, November 2004.

[35] Vinay B., et al., ”Diagnosis of helicopter gearboxes using structure-based networks”, Proc.of the American Control Conference, pp. 1623-1627, Seattle Washington, June 1995.

[36] Constantino R., et al., “ Failure detection and identification and fault tolerant control using the IMM-KF with applications to the Eagle-Eye UAV”, Proceedings of the 37th IEEE Conference on Decision & Control, pp. 4208-4213, Tampa Florida, USA, December 1998.

[37] Wang W., "Identification of gear mesh signals by Kurtosis maximisation and its application to CH46 helicopter gearbox data", Proceeding of the 11th IEEE signal processing workshop on statistical signal processing, pp. 369-372, 6-8 August, 2001.

[38] Yen G. G., Meesad P., "An effective neuro-fuzzy paradigm for machinery condition health monitoring", IEEE Trans. on Systems, Man and Cybernetics – Part B: Cybernetics, vol. 31, no. 4, pp. 523-536, August 2001.

[39] Gillespie B. W., Atlas L. E., "Optimizing time-frequency kernels for classification", IEEE Trans. on Signal Processing, vol. 49, no. 3, pp. 485-496, March 2001.

[40] Yu X. H., "Actuator fault compensation for a helicopter model", Proceedings of IEEE Conference on Control Applications, vol. 1, pp. 2957-2969, 23-25 June, 2003.

[41] Amulya K., et al., "Hybrid reasoning for prognostic learning in CBM sytems", IEEE Aerospace Conference Proceedings, vol. 6, pp. 1747-1752, 10-17 March, 2001.

[42] Wen F., Deb S., "Signal processing and fault detection with application to CH-46 helicopter data", IEEE Aerospace Conference Proceedings, vol. 6, pp. 15-25, 18-25 March, 2000.

[43] Borsje H. J., "Fault detection in boilers using canonical variate analysis", Proceeding of American Control Conference, pp. 1167-1170, San Diego, California, June 1999.

[44] Fretheim T., Vincent T. L., Shoureshi R., "Optimization based fault detection for nonlinear systems", Proceeding of American Control Conference, pp. 1747-1752, Arlington, VA, June 25-27, 2001.

[45] Alessandri A., Parrisini T., "Neural state estimators for direct model based fault diagnosis", Proceeding of American Control Conference, pp. 2874-2878, Philadelphia, Pennsylvania, June 1998.

[46] Borairi M., Wang H., "Actuator and sensor fault diagnosis of non-linear dynamic system via genetic neural networks and adaptive parameter estimation technique", Proceeding of IEEE International Conference on Control Applications, pp. 278-282, Trieste, Italy, 1-4 September, 1998.

[47] Shumsky A. Y., "An approach to failure isolation in nonlinear systems", Proceeding of American Control Conference, pp. 3577-

3578, Baltimore, Maryland, June 1994.

[48] Gertler J., Luo Q., “Direct identification of nonlinear parity relations-a way around the robustness problem”, Proceeding of the 37th IEEE International Conference on Decision & Control, pp. 78-83, Tampa, Florida, December 1998.

[49] Stoustrup J., Niemann H., “Fault detection for non-linear systems-a standart problem approach”, Proceeding of the 37th IEEE International Conference on Decision & Control, pp. 96-101, Tampa, Florida, December 1998.

[50] Chowdhury F. N., “Goal-orianted system identification for safe operation of engineering problem”, Proceeding of American Control Conference, pp. 2711-2712, Chicago, Illinois, June 2000.

[51] Selmic R. R., Lewis F. L., “Multimodel neural networks identification and failure detection of nonlinear systems”, Proceeding of the 40th IEEE International Conference on Decision & Control, pp. 3128-3133, Orlando, Florida, December 2001.

[52] Kim Y. C., Agrawal V. D., Saluja K. K., “Multiple faults: modeling, simulation and test”, Proceeding of the 15th IEEE International Conference on VLSI Design, pp. 592-597, 2002.

[53] Moseler O., Isermann R., “Application of model-based fault detection to a brushless DC motor”, IEEE Trans. on Industrial Electronics, vol. 47, no. 5, pp. 1015-1020, October 2000.

[54] Chin H. C., “Fault section diagnosis of power system using fuzzy logic”, IEEE Trans. on Power Systems, vol. 18, no. 1, pp. 245-250, February 2003.

[55] Börner M., Isermann R., “Model-based detection of critical driving situations with fuzzy logic decision making”, Control Engineering Practice, article in press, accepted on 21 November 2004.

[56] Zhao J., Meyer F.J., “Maximal diagnosis of interconnects of random access memories”, IEEE Trans. on Reliability, vol. 52, no. 4, pp. 423-434, December 2003.

[57] Koushanfar F., Potkonjak M., Vincenttelli A. S., “On-line fault detection of sensor measurments”, Proceeding of IEEE, pp. 974-979, 22-24 October, 2003.

[58] Linaric D., Koroman V., “Fault diagnosis of a hydraulic actuator using neural network”, Proceeding of IEEE ICIT, pp. 108-111,

Maribor, Slovenia, 2003.

[59] Embrechts M. J., Benedek S., “Hybrid identification of nuclear power plant transients with artificial neural networks”, IEEE Trans. on Industrial Electronics, vol. 51, no. 3, pp. 686-693, June 2004.

[60] Marcos A., Balas G. J., Mylaraswany D., “Robust identification and residual generation application to a turbofan engine”, IEEE Aerospace Conference Proceedings, pp. 3384-3394, 2004.

[61] Yen G. G., Ho L. W., “Online multiple model based fault diagnosis and accommodation”, IEEE Trans. on Industrial Electronics, vol. 50, no. 2, pp. 296-312, April 2003.

[62] Frank P. M., Ding X., “Survey of robust residual generation and evaluation methods in observer based fault detection systems”, J. Process Control, vol. 7, no. 6, pp. 403-424, 1997.

[63] Chen J., Patton R. J., “Robust model based fault diagnosis for dynamic systems”, Kluwer Academic Publishers, 1999.

[64] Frank P. M., “Fault diagnosis in dynamic system using analytical and knowledge-based redundancy- a survey and some new results”, Automatica, vol. 26, no. 3, pp. 453-474, 1990.

[65] Desoer C. A., “Notes for a second course on linear systems”, University of California, Berkeley, August 1969.

[66] Patton R., “Fault tolerant control: the 1997 situation, “Proceeding of the 3rd Symposium on Fault Detection”, Supervision and Safety for Technical Processes (SAFEPROCESS’97), pp. 1033-1054, Hull University, Hull, UK, 1997.

[67] Bošković J., Mehra R., “Failure detection, identification and reconfiguration system for a redundant actuator assembly”, Proceeding of the 5th Symposium on Fault Detection, Supervision and safety for Tecnical Processes (SAFEPROCESS’2003), pp. 429-434, Washington D.C., USA, 2003.

[68] Kinnaert M., “Fault diagnosis based on analytical models for linear and nonlinear systems-a tutorial”, Procceding of the 5th Symposium on Fault Detection, Supervision and safety for Tecnical Processes (SAFEPROCESS’2003), pp. 37-50, Washington D.C., USA, 2003.

[69] Aström K.J., Albertos P., Blanke M., Isidori A., Santz R., Schaufelberger W., “Control of complex systems”, Springer Verlag 2000.

[70] Diamantaras K.I., Papadimitriou Th., “Blind signal separation using oriented PCA neural models”., ICASSP, IEEE, pp. 733-736, 2003.

[71] Edward J.J.A., “User’s guide to principal components”, Wiley-Interscience, New-York, 1991.

[72] Hagenblad A., Gustafsson F., Klein I., “A comparison of two methods for stochastic fault detection: The parity space approach and principal components analysis”, 13th IFAC Symposium on system identification, pp. 1090-1095, Rotherdam, The Nederlands, 27-29 August, 2003.

[73] Haiging W., Zhihuan S., Ping L., “Improved PCA with optimized sensor locations for process monitoring and fault diagnosis”, Proceeding of the 39th IEEE Conference on Decision and Control, pp. 4353-4358, Sydney, Australia, December 2000.

[74] Hiden H.G., Willis M.J., Tham M.T., Turner P., Montague G.A., “Non-linear principal components analysis using genetic programming”, Genetic Algorithms in Engineering Systems: Innovations and Applications, vol. 446, pp. 302-307, 2-4 September, 1997.

[75] Hotelling H.,”Analysis of a complex of statistical variables into principal components”, J. Educ. Psychol., vol. 24, pp. 417, 1933.

[76] Hyvärinen E., Karhunen J., Oja E., “Independent component analysis”, John Wiley and Sons, 2001.

[77] MacGregor J.F., “Data based methods for process analysis, monitoring and control”, 13th IFAC Symposium on System Identification, pp. 1019-1029, 2003.

[78] Misra M. et al., “Multivariate process monitoring and fault diagnosis by multi-scale PCA”, Computer and Chemical Engineering, vol. 26, pp. 1281-1293, 2002.

[79] Moya E., Sainz G.I., Grande B., Fuente M. J., Peran J.R., “Neural PCA based fault diagnosis”, Proceeding of the European Control Conference, pp. 809-813, 2001.

[80] Pöyhönen S., Jover P., Hyötyniemi H., “Independent component analysis of vibrations for fault diagnosis of an induction motor”, Proceeding of the IASTED international conference, Circuits, Signals and Systems, pp. 203-208, Cancun, Mexico, 19-21 May, 2003.

[81] Suykens J.A.K, Van Gestel T., Vandewalle J., De Moor B., "A support vector machine formulation to PCA analysis and its kernel version", IEEE Trans. on Neural Networks, vol.14, no. 2, pp. 450-477, March 2003.

[82] Vedam H., Venkatasubramanian V., "PCA-SDG based process monitoring and fault diagnosis", Control Engineering Practice, vol. 7, pp. 903-917, 1999.

[83] Zang H., Tangirala A.K., Shah S., "Dynamic process monitoring using multiscale PCA", Proceeding of the 1999 IEEE Canadian Conference on Electrical and Computer Engineering, pp. 1579-1584, Edmonton, Alberta, Canada, May 1999.

[84] Zeng X., Chen Y., Nakao Z., "Image feature representation by the subspace of nonlinear PCA", ICASSP, IEEE, pp. 228-231, 2002.

[85] Ljung L., "System Identification: Theory for the user", Englewood Cliffs, N.J., Prentice Hall, 1987.

[86] Johansson K.H., "The quadruple-tank process: a multivariable laboratory process with and adjustable zero", IEEE Trans. on Cont. Sys. Tech., vol. 8, no. 3, pp. 807-812, 2000.

[87] Luenberger, D.G., "Observing the state of a linear system", IEEE Trans. Military Electr., vol. 8, pp. 74-80, 1964.

[88] Elmas C., Zelaya-de la Parra H., "Application of a full-order Luenberger observer for a position sensorless operation of a switched reluctance motor drive," IEE Proc. Control Theory Appl., vol. 143, no. 5, pp. 401-408, September 1996.

[89] Song J., Lee K. B., Song J. H., I. Choy, Kim K. B., "Sensorless vector control of induction motor using a novel reduced order extended Luenberger observer," Conf. Record of the 2000 Industry Appl. Conf., vol. 3, pp. 1828-1843, 2000.

[90] Du T., Brdys M.A., "Implementation of extended Luenberger observers for joint state and parameter estimation of PWM induction motor drive," 5th European Conf. on Power Elctr. Appl., vol. 4, pp. 439-444, 1993.

[91] Cicceralla G., Dalla M. M., Germani A., "A Luenberger-like observer for nonlinear systems", Int. J. Control, vol. 57, pp. 537, 1993.

[92] Benamor S., Hammouri H., Couenne F., "A Luenberger-like observer for discrete time nonlinear systems", Proceedings of the 37th IEEE Conference on Decision & Control, pp. 4612-4613, Tampa, Florida USA, December 1998.

[93] Guerra M. R., Diop S., "Diagnosis of nonlinear systems using an unknown-input observer: an algebraic and differential approach", IEE Proc. Control Theory Appl., vol. 151, no. 1, pp. 130-135, Mexico, January 2004.

[94] Kılıç E., Leblebicioğlu K., "Fault diagnosis with dynamic fuzzy discrete event system approach", The Proceedings of the 14th Turkish Symposium on Artificial Intelligence and Neural Networks, pp. 55-62, Çeşme, İzmir, Turkey, 16-17 June, 2005.

[95] Kılıç E., Leblebicioğlu K., Halıcı U., "Bulanık renkli petri ağı kullanarak asenkron motorda hata teşhis ve tesbiti", Endüstri ve Otomasyon Dergisi, vol. 64, pp. 42-47, July 2002 (In Turkish).

[96] Kılıç E., Leblebicioğlu K., "Bulanık ayrık olaylı sistem yaklaşımı ile hata tanılama", TOK-2005 Ulusal Otomatik Kontrol Konferansı, pp. 179-186, İstanbul Teknik Universitesi, İstanbul,Turkey, 2-3 Haziran, 2005 (In Turkish).

[97] Kılıç E., Leblebicioğlu K., Halıcı U., "Eğitilmiş üyelik fonksiyonları kullanarak bulanık karar ağacı ile doğal gazlı boru hatlarında hata bulma", IEEE-SİU-2001, pp. 440-445, Girne, Cyprus, 2001 (In Turkish).

[98] Kılıç E., Leblebicioğlu K., Halıcı U., "Bulanık karar ağacı ile doğal gazlı boru hatlarında hata bulma", IEEE-SİU-2000, pp. 165-169, Belek, Antalya 2000 (In Turkish).

[99] Kılıç E., Leblebicioğlu K., Halıcı U., "Bulanık renkli Petri ağı kullanarak asenkron motorda hata teşhis ve tesbiti", TOK-2001 Ulusal Otomatik Kontrol Konferansı, pp. 181-189, Bursa, 2001 (In Turkish).

[100] Kılıç E., Leblebicioğlu K., Halıcı U., "Bulanık ayrık olaylı sistem yapısı ve bulanık gözlenebilme kriteri kullanarak sistemlerde hata tanılama", TOK-2002 Ulusal Otomatik Kontrol Konferansı, pp. 715-723, ODTÜ, Ankara 2002 (In Turkish).

[101] Kılıç E., Leblebicioğlu K., "Fault diagnosis with dynamic fuzzy discrete event system approach", accepted for publication in Lecture Notes in Artificial Intelligence.

[102] Kanev S., “Robust fault-tolerant control”, PhD. thesis, University of Twente, The Netherlands, March 2004.

APPENDIX A

RULE-BASE USED IN THE FDD PROGRAM TO DIAGNOSE THE HELICOPTER ACTUATOR AND PARAMETER FAULTS

There are many system parameters in the helicopter model but we dealt with only four of them. These parameters are numbered as 1, 2, 3 and 4. The parameters 1 and 2 are related to the helicopter wings and 3 and 4 are related to the helicopter main rotor and tail rotor blade. Parameter faults are restricted to percentage faults.

There are four actuators in the helicopter. All of the actuators are taken into account for fault diagnosis. There are two types of faults in the actuators. The first one is percentage fault and the other is stuck fault. Both of these fault types are considered for fault diagnosis.

The rule sets given below are used to diagnose parameters and actuators faults. In this representation each of the rows corresponds to a rule defined for a particular fault case. The first group of columns denotes a rule antecedent part; each column in this group stands for a distinct super event with a membership degree. Super events denoted by 1 are enabled with corresponding membership degrees whereas super events corresponding to zeros are disabled.

The first column after comma denotes a rule’s consequent part. The number after comma shows linguistic variables. The meaning of these numbers differs for stuck and percentage faults. For stuck faults 1 stands for no stuck error and 2 stands for stuck error. For actuator percentage fault 1 denotes no fault and 2 shows medium fault. Finally for parameter percentage faults 1, 2 and 3 stand for no fault, small fault and medium fault, respectively. The column in the parenthesis denotes weight and the last column shows connection parameter value.

1. RULES TO DIAGNOSE ACTUATOR-1 PERCENTAGE FAULTS

[Rules]

1[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
2[0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
3[0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
4[0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
5[0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
6[0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
7[0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
8[0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
9[0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
10[0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
11[0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
12[0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
13[0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
14[0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
15[0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
16[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
17[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 , 2 (1) : 1
18[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 , 2 (1) : 1
19[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 , 2 (1) : 1
20[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 , 2 (1) : 1
21[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 , 2 (1) : 1
22[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 , 2 (1) : 1
23[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 , 2 (1) : 1
24[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 , 2 (1) : 1
25[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 , 1 (1) : 1

2. RULES TO DIAGNOSE ACTUATOR-2 PERCENTAGE FAULTS

[Rules]

1[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
2[0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
3[0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
4[0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
5[0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
6[0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
7[0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
8[0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
9[0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
10[0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
11[0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
12[0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1

13[0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
14[0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
15[0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
16[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
17[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 , 2 (1) : 1
18[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 , 2 (1) : 1
19[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 , 2 (1) : 1
20[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 , 2 (1) : 1
21[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 , 2 (1) : 1
22[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 , 2 (1) : 1
23[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 , 2 (1) : 1
24[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 , 2 (1) : 1
25[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 , 1 (1) : 1

3. RULES TO DIAGNOSE ACTUATOR-3 PERCENTAGE FAULTS

[Rules]
1[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
2[0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
3[0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
4[0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
5[0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
6[0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
7[0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
8[0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
9[0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
10[0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
11[0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
12[0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
13[0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
14[0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
15[0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
16[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
17[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 , 2 (1) : 1
18[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 , 2 (1) : 1
19[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 , 2 (1) : 1
20[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 , 2 (1) : 1
21[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 , 2 (1) : 1
22[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 , 2 (1) : 1
23[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 , 2 (1) : 1
24[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 , 2 (1) : 1
25[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 , 1 (1) : 1

4. RULES TO DIAGNOSE ACTUATOR-4 PERCENTAGE FAULTS

[Rules]
1[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1

2[0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
3[0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
4[0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
5[0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
6[0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
7[0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
8[0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
9[0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
10[0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
11[0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
12[0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
13[0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
14[0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
15[0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
16[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
17[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 , 2 (1) : 1
18[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 , 2 (1) : 1
19[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 , 2 (1) : 1
20[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 , 2 (1) : 1
21[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 , 2 (1) : 1
22[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 , 2 (1) : 1
23[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 , 2 (1) : 1
24[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 , 2 (1) : 1
25[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 , 1 (1) : 1

5. RULES TO DIAGNOSE ACTUATOR-1 STUCK FAULTS

[Rules]

1[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
2[0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
3[0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
4[0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
5[0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
6[0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
7[0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
8[0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0, 2 (1) : 1
9[0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0, 2 (1) : 1
10[0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0, 2 (1) : 1
11[0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0, 2 (1) : 1
12[0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0, 2 (1) : 1
13[0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0, 2 (1) : 1
14[0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0, 2 (1) : 1
15[0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0, 2 (1) : 1
16[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0, 2 (1) : 1
17[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1, 1 (1) : 1

6. RULES TO DIAGNOSE ACTUATOR-2 STUCK FAULTS

[Rules]

1[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
2[0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
3[0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
4[0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
5[0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
6[0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
7[0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
8[0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
9[0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
10[0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
11[0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
12[0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
13[0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0, 2 (1) : 1
14[0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0, 2 (1) : 1
15[0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0, 2 (1) : 1
16[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0, 2 (1) : 1
17[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0, 2 (1) : 1
18[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0, 2 (1) : 1
19[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0, 2 (1) : 1
20[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0, 2 (1) : 1
21[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0, 2 (1) : 1
22[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1, 1 (1) : 1

7. RULES TO DIAGNOSE ACTUATOR-3 STUCK FAULTS

[Rules]

1[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
2[0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
3[0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
4[0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
5[0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
6[0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
7[0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
8[0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
9[0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
10[0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
11[0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
12[0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0, 2 (1) : 1
13[0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0, 2 (1) : 1
14[0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0, 2 (1) : 1
15[0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0, 2 (1) : 1
16[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0, 2 (1) : 1
17[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0, 2 (1) : 1
18[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0, 2 (1) : 1
19[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0, 2 (1) : 1

20[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0, 2 (1) : 1
21[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1, 1 (1) : 1

8. RULES TO DIAGNOSE ACTUATOR-4 STUCK FAULTS

[Rules]
1[1 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
2[0 1 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
3[0 0 1 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
4[0 0 0 1 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
5[0 0 0 0 1 0 0 0 0 0 0 0 0 0, 2 (1) : 1
6[0 0 0 0 0 1 0 0 0 0 0 0 0 0, 2 (1) : 1
7[0 0 0 0 0 0 1 0 0 0 0 0 0 0, 2 (1) : 1
8[0 0 0 0 0 0 0 1 0 0 0 0 0 0, 2 (1) : 1
9[0 0 0 0 0 0 0 0 1 0 0 0 0 0, 2 (1) : 1
10[0 0 0 0 0 0 0 0 0 1 0 0 0 0, 2 (1) : 1
11[0 0 0 0 0 0 0 0 0 0 1 0 0 0, 2 (1) : 1
12[0 0 0 0 0 0 0 0 0 0 0 1 0 0, 2 (1) : 1
13[0 0 0 0 0 0 0 0 0 0 0 0 1 0, 2 (1) : 1
14[0 0 0 0 0 0 0 0 0 0 0 0 0 1, 1 (1) : 1

9. RULES TO DIAGNOSE PARAMETER-1 PERCENTAGE FAULTS

[Rules]
1[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
2[0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
3[0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
4[0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
5[0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
6[0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
7[0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
8[0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
9[0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
10[0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
11[0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
12[0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
13[0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
14[0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
15[0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
16[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
17[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
18[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
19[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
20[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
21[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 , 2 (1) : 1
22[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 , 2 (1) : 1
23[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 , 2 (1) : 1
24[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 , 2 (1) : 1

25[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 , 3 (1) : 1
26[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 , 3 (1) : 1
27[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 , 3 (1) : 1
28[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 , 3 (1) : 1
29[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 , 1 (1) : 1

10. RULES TO DIAGNOSE PARAMETER-2 PERCENTAGE FAULTS

[Rules]
1[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
2[0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
3[0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
4[0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
5[0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
6[0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
7[0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
8[0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
9[0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
10[0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
11[0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
12[0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
13[0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
14[0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
15[0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
16[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
17[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0, 2 (1) : 1
18[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0, 2 (1) : 1
19[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0, 2 (1) : 1
20[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0, 2 (1) : 1
21[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0, 2 (1) : 1
22[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0, 2 (1) : 1
23[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0, 2 (1) : 1
24[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0, 3 (1) : 1
25[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0, 3 (1) : 1
26[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1, 1 (1) : 1

11. RULES TO DIAGNOSE PARAMETER-3 PERCENTAGE FAULTS

[Rules]
1[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
2[0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
3[0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
4[0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
5[0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
6[0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
7[0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
8[0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
9[0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1

10[0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
11[0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
12[0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
13[0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
14[0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
15[0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
16[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
17[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
18[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
19[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0, 2 (1) : 1
20[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0, 2 (1) : 1
21[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0, 2 (1) : 1
22[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0, 2 (1) : 1
23[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0, 2 (1) : 1
24[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0, 3 (1) : 1
25[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0, 3 (1) : 1
26[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0, 3 (1) : 1
27[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0, 3 (1) : 1
28[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1, 1 (1) : 1

12. RULES TO DIAGNOSE PARAMETER-4 PERCENTAGE FAULTS

[Rules]
1[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
2[0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
3[0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
4[0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
5[0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
6[0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
7[0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
8[0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
9[0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
10[0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
11[0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
12[0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
13[0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
14[0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
15[0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
16[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
17[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
18[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
19[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0, 2 (1) : 1
20[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0, 2 (1) : 1
21[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0, 2 (1) : 1
22[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0, 2 (1) : 1
23[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0, 2 (1) : 1
24[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0, 2 (1) : 1
25[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0, 3 (1) : 1

26[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0, 3 (1) : 1
27[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0, 3 (1) : 1
28[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0, 3 (1) : 1
29[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1, 1 (1) : 1

APPENDIX B

RULE-BASE USED IN THE FDD PROGRAM TO DIAGNOSE THE INDUCTION MOTOR FAULTS

The rule sets given below are used to diagnose an induction motor stator winding and bearing faults. To diagnose the stator winding faults following rule set is used in FDES and FDEDS diagnosers. Each of rows in the rule set corresponds to a rule defined for a particular fault case. The first group of columns in a row denotes a rule antecedent part. The numbers are linguistic variables related to the motor speed residual, motor current residual, zero stator winding fault and stator winding fault. For motor speed and current residuals; 1, 2 and 3 denote zero, medium and big speed or current residuals, respectively. Finally, for zero stator winding fault and winding fault last two columns of the first group show the membership degrees. In these columns 1, 2 and 3 stands for membership degrees of zero, medium and big, respectively.

The first column after comma denotes a rule's consequent part. The number after comma shows linguistic variables. These meanings of these numbers differ for FDES and FDEDS diagnosers. In these columns 1, 2 and 3 stands for membership degrees of zero, medium and big, respectively. For FDEDS diagnoser, in these columns 1, 2 and 3 stands for membership degrees of negative big changes, zero, positive big change in the for zero stator winding fault and winding fault, respectively. The column in the parenthesis denotes weight and the last column shows connection parameter value.

1. RULES FOR FDES DIAGNOSER TO DIAGNOSE THE MOTOR STATOR WINDING FAULT

[Rules]

1[1 1 1 1, 1 3 (1) : 1
2[1 1 1 2, 1 3 (1) : 1
3[1 1 1 3, 1 3 (1) : 1
4[1 1 2 1, 1 3 (1) : 1
5[1 1 2 2, 1 3 (1) : 1
6[1 1 2 3, 1 3 (1) : 1
7[1 1 3 1, 1 3 (1) : 1
8[1 1 3 2, 1 3 (1) : 1
9[1 1 3 3, 1 3 (1) : 1
10[1 2 1 1, 2 2 (1) : 1
11[1 2 1 2, 2 2 (1) : 1
12[1 2 1 3, 2 2 (1) : 1
13[1 2 2 1, 2 2 (1) : 1
14[1 2 2 2, 2 2 (1) : 1
15[1 2 2 3, 2 2 (1) : 1
16[1 2 3 1, 2 2 (1) : 1
17[1 2 3 2, 2 2 (1) : 1
18[1 2 3 3, 2 2 (1) : 1
19[1 3 1 1, 3 1 (1) : 1
20[1 3 1 2, 3 1 (1) : 1
21[1 3 1 3, 3 1 (1) : 1
22[1 3 2 1, 3 1 (1) : 1
23[1 3 2 2, 3 1 (1) : 1
24[1 3 2 3, 3 1 (1) : 1
25[1 3 3 1, 3 1 (1) : 1
26[1 3 3 2, 3 1 (1) : 1
27[1 3 3 3, 3 1 (1) : 1
28[2 1 1 1, 1 3 (1) : 1
29[2 1 1 2, 1 3 (1) : 1
30[2 1 1 3, 1 3 (1) : 1
31[2 1 2 1, 1 3 (1) : 1
32[2 1 2 2, 1 3 (1) : 1
33[2 1 2 3, 1 3 (1) : 1
34[2 1 3 1, 1 3 (1) : 1
35[2 1 3 2, 1 3 (1) : 1
36[2 1 3 3, 1 3 (1) : 1
37[2 2 1 1, 1 3 (1) : 1
38[2 2 1 2, 1 3 (1) : 1
39[2 2 1 3, 1 3 (1) : 1
40[2 2 2 1, 1 3 (1) : 1
41[2 2 2 2, 1 3 (1) : 1
42[2 2 2 3, 1 3 (1) : 1
43[2 2 3 1, 1 3 (1) : 1
44[2 2 3 2, 1 3 (1) : 1

45[2 2 3 3, 1 3 (1) : 1
46[2 3 1 1, 1 3 (1) : 1
47[2 3 1 2, 1 3 (1) : 1
48[2 3 1 3, 1 3 (1) : 1
49[2 3 2 1, 1 3 (1) : 1
50[2 3 2 2, 1 3 (1) : 1
51[2 3 2 3, 1 3 (1) : 1
52[2 3 3 1, 1 3 (1) : 1
53[2 3 3 2, 1 3 (1) : 1
54[2 3 3 3, 1 3 (1) : 1

2. RULES FOR FDES FDEDS DIAGNOSER TO DIAGNOSE THE MOTOR STATOR WINDING FAULT

[Rules]

1[1 1 1 1, 2 2 (1) : 1
2[1 1 1 2, 2 1 (1) : 1
3[1 1 1 3, 2 1 (1) : 1
4[1 1 2 1, 2 2 (1) : 1
5[1 1 2 2, 1 1 (1) : 1
6[1 1 2 3, 1 2 (1) : 1
7[1 1 3 1, 2 2 (1) : 1
8[1 1 3 2, 2 2 (1) : 1
9[1 1 3 3, 2 3 (1) : 1
10[1 2 1 1, 2 2 (1) : 1
11[1 2 1 2, 2 1 (1) : 1
12[1 2 1 3, 2 1 (1) : 1
13[1 2 2 1, 3 3 (1) : 1
14[1 2 2 2, 3 2 (1) : 1
15[1 2 2 3, 3 2 (1) : 1
16[1 2 3 1, 2 3 (1) : 1
17[1 2 3 2, 2 2 (1) : 1
18[1 2 3 3, 2 2 (1) : 1
19[1 3 1 1, 2 2 (1) : 1
20[1 3 1 2, 2 2 (1) : 1
21[1 3 1 3, 2 2 (1) : 1
22[1 3 2 1, 2 1 (1) : 1
23[1 3 2 2, 2 1 (1) : 1
24[1 3 2 3, 2 1 (1) : 1
25[1 3 3 1, 2 1 (1) : 1
26[1 3 3 2, 2 1 (1) : 1
27[1 3 3 3, 2 1 (1) : 1
28[2 1 1 1, 3 2 (1) : 1
29[2 1 1 2, 3 2 (1) : 1
30[2 1 1 3, 3 2 (1) : 1
31[2 1 2 1, 3 1 (1) : 1
32[2 1 2 2, 3 2 (1) : 1
33[2 1 2 3, 3 2 (1) : 1

34[2 1 3 1, 3 1 (1) : 1
35[2 1 3 2, 3 2 (1) : 1
36[2 1 3 3, 3 3 (1) : 1
37[2 2 1 1, 1 1 (1) : 1
38[2 2 1 2, 1 1 (1) : 1
39[2 2 1 3, 2 1 (1) : 1
40[2 2 2 1, 1 1 (1) : 1
41[2 2 2 2, 2 2 (1) : 1
42[2 2 2 3, 1 3 (1) : 1
43[2 2 3 1, 1 1 (1) : 1
44[2 2 3 2, 1 2 (1) : 1
45[2 2 3 3, 2 1 (1) : 1
46[2 3 1 1, 2 1 (1) : 1
47[2 3 1 2, 3 2 (1) : 1
48[2 3 1 3, 3 3 (1) : 1
49[2 3 2 1, 3 1 (1) : 1
50[2 3 2 2, 3 2 (1) : 1
51[2 3 2 3, 3 3 (1) : 1
52[2 3 3 1, 3 3 (1) : 1
53[2 3 3 2, 3 2 (1) : 1
54[2 3 3 3, 3 3 (1) : 1

In order to diagnose the bearing fault the following rule-set is used in FDES and FDEDS diagnosers. The first group of columns in a row denotes a rule antecedent part. The numbers are linguistic variables related to the motor speed residual, motor current residual, zero bearing fault and bearing fault. For motor speed and current residuals 1, 2, 3, 4 and 5 denote zero, very small, small, medium and big speed or current residuals, respectively. Finally, for zero bearing fault and bearing fault last two columns of the first group show the membership degrees. In these columns 1, 2, 3, 4 and 5 stands for membership degrees of zero, very small, small, medium and big, respectively.

The first column after comma denotes a rule's consequent part. The number after comma shows linguistic variables. These meanings of these numbers differ for FDES and FDEDS diagnosers. In these columns 1, 2, 3, 4 and 5 stands for membership degrees of zero, very small, small, medium and big, respectively. For FDEDS diagnoser, in these columns 1, 2, 3, 4 and 5 stands for membership degrees of negative big changes, negative medium change, zero, positive medium change, positive big change in the for zero bearing fault and bearing fault, respectively. The column in the parenthesis denotes weight and the last column shows connection parameter value.

3. RULES FOR FDES DIAGNOSER TO DIAGNOSE THE MOTOR BEARING FAULT

[Rules]

1[1 1 1 1, 1 5 (1) : 1
2[1 1 2 2, 1 5 (1) : 1
3[1 1 3 3, 1 5 (1) : 1
4[1 1 4 4, 1 5 (1) : 1
5[1 1 5 5, 1 5 (1) : 1
6[1 2 1 1, 1 5 (1) : 1
7[1 2 2 2, 1 5 (1) : 1
8[1 2 3 3, 1 5 (1) : 1
9[1 2 4 4, 1 5 (1) : 1
10[1 2 5 5, 1 5 (1) : 1
11[1 3 1 1, 1 5 (1) : 1
12[1 3 2 2, 1 5 (1) : 1
13[1 3 3 3, 1 5 (1) : 1
14[1 3 4 4, 1 5 (1) : 1
15[1 3 5 5, 1 5 (1) : 1
16[2 1 1 1, 1 5 (1) : 1
17[2 1 2 2, 1 5 (1) : 1
18[2 1 3 3, 1 5 (1) : 1
19[2 1 4 4, 1 5 (1) : 1
20[2 1 5 5, 1 5 (1) : 1
21[2 2 1 1, 2 4 (1) : 1
22[2 2 2 2, 2 4 (1) : 1
23[2 2 3 3, 2 4 (1) : 1
24[2 2 4 4, 2 4 (1) : 1
25[2 2 5 5, 2 4 (1) : 1
26[2 3 1 1, 2 4 (1) : 1
27[2 3 2 2, 2 4 (1) : 1
28[2 3 3 3, 2 4 (1) : 1
29[2 3 4 4, 2 4 (1) : 1
30[2 3 5 5, 2 4 (1) : 1
31[3 1 1 1, 2 4 (1) : 1
32[3 1 2 2, 2 4 (1) : 1
33[3 1 4 4, 2 4 (1) : 1
34[3 1 5 5, 2 4 (1) : 1
35[3 2 1 1, 3 3 (1) : 1
36[3 2 2 2, 3 3 (1) : 1
37[3 2 3 3, 3 3 (1) : 1
38[3 2 4 4, 3 3 (1) : 1
39[3 2 5 5, 3 3 (1) : 1

40[3 3 1 1, 3 3 (1) : 1
41[3 3 2 2, 3 3 (1) : 1
42[3 3 3 3, 3 3 (1) : 1
43[3 3 4 4, 3 3 (1) : 1
44[3 3 5 5, 3 3 (1) : 1
45[4 1 1 1, 3 3 (1) : 1
46[4 1 2 2, 3 3 (1) : 1
47[4 1 3 3, 3 3 (1) : 1
48[4 1 5 5, 3 3 (1) : 1
49[4 2 1 1, 4 2 (1) : 1
50[4 2 2 2, 4 2 (1) : 1
51[4 2 3 3, 4 2 (1) : 1
52[4 2 4 4, 4 2 (1) : 1
53[4 2 5 5, 4 2 (1) : 1
54[4 3 1 1, 4 2 (1) : 1
55[4 3 2 2, 4 2 (1) : 1
56[4 3 3 3, 4 2 (1) : 1
57[4 3 4 4, 4 2 (1) : 1
58[4 3 5 5, 4 2 (1) : 1
59[5 1 1 1, 4 2 (1) : 1
60[5 1 2 2, 4 2 (1) : 1
61[5 1 3 3, 4 2 (1) : 1
62[5 1 4 4, 4 2 (1) : 1
63[5 1 5 5, 4 2 (1) : 1
64[5 2 1 1, 5 1 (1) : 1
65[5 2 2 2, 5 1 (1) : 1
66[5 2 3 3, 5 1 (1) : 1
67[5 2 4 4, 5 1 (1) : 1
68[5 2 5 5, 5 1 (1) : 1
69[5 3 1 1, 5 1 (1) : 1
70[5 3 2 2, 5 1 (1) : 1
71[5 3 3 3, 5 1 (1) : 1
72[5 3 4 4, 5 1 (1) : 1
73[5 3 5 5, 5 1 (1) : 1

4. RULES FOR FDEDS DIAGNOSER TO DIAGNOSE THE MOTOR BEARING FAULT

[Rules]

1[1 1 1 1, 3 3 (1) : 1
2[1 1 2 2, 1 1 (1) : 1
3[1 1 3 3, 1 1 (1) : 1
4[1 1 4 4, 1 1 (1) : 1
5[1 1 5 5, 1 1 (1) : 1
6[1 2 1 1, 3 4 (1) : 1
7[1 2 2 2, 3 3 (1) : 1
8[1 2 3 3, 4 2 (1) : 1
9[1 2 4 4, 1 2 (1) : 1

10[1 2 5 5, 1 4 (1) : 1
11[1 3 1 1, 3 3 (1) : 1
12[1 3 2 2, 4 4 (1) : 1
13[1 3 3 3, 4 5 (1) : 1
14[1 3 4 4, 4 5 (1) : 1
15[1 3 5 5, 4 3 (1) : 1
16[2 1 1 1, 3 3 (1) : 1
17[2 1 2 2, 4 4 (1) : 1
18[2 1 3 3, 5 5 (1) : 1
19[2 1 4 4, 5 5 (1) : 1
20[2 1 5 5, 4 5 (1) : 1
21[2 2 1 1, 4 4 (1) : 1
22[2 2 2 2, 3 3 (1) : 1
23[2 2 3 3, 4 4 (1) : 1
24[2 2 4 4, 5 5 (1) : 1
25[2 2 5 5, 4 5 (1) : 1
26[2 3 1 1, 4 5 (1) : 1
27[2 3 2 2, 3 3 (1) : 1
28[2 3 3 3, 4 4 (1) : 1
29[2 3 4 4, 4 5 (1) : 1
30[2 3 5 5, 5 5 (1) : 1
31[3 1 1 1, 4 1 (1) : 1
32[3 1 2 2, 3 5 (1) : 1
33[3 1 4 4, 4 3 (1) : 1
34[3 1 5 5, 3 3 (1) : 1
35[3 2 1 1, 3 3 (1) : 1
36[3 2 2 2, 3 3 (1) : 1
37[3 2 3 3, 3 3 (1) : 1
38[3 2 4 4, 3 3 (1) : 1
39[3 2 5 5, 3 3 (1) : 1
40[3 3 1 1, 5 5 (1) : 1
41[3 3 2 2, 4 4 (1) : 1
42[3 3 3 3, 3 3 (1) : 1
43[3 3 4 4, 3 3 (1) : 1
44[3 3 5 5, 4 4 (1) : 1
45[4 1 1 1, 5 3 (1) : 1
46[4 1 2 2, 5 4 (1) : 1
47[4 1 3 3, 4 4 (1) : 1
48[4 1 5 5, 5 3 (1) : 1
49[4 2 1 1, 5 5 (1) : 1
50[4 2 2 2, 4 4 (1) : 1
51[4 2 3 3, 3 3 (1) : 1
52[4 2 4 4, 3 3 (1) : 1
53[4 2 5 5, 3 3 (1) : 1
54[4 3 1 1, 5 5 (1) : 1
55[4 3 2 2, 4 4 (1) : 1
56[4 3 3 3, 4 3 (1) : 1
57[4 3 4 4, 4 3 (1) : 1

58[4 3 5 5, 4 2 (1) : 1
59[5 1 1 1, 4 1 (1) : 1
60[5 1 2 2, 4 5 (1) : 1
61[5 1 3 3, 4 4 (1) : 1
62[5 1 4 4, 4 4 (1) : 1
63[5 1 5 5, 3 1 (1) : 1
64[5 2 1 1, 5 5 (1) : 1
65[5 2 2 2, 4 3 (1) : 1
66[5 2 3 3, 3 5 (1) : 1
67[5 2 4 4, 3 3 (1) : 1
68[5 2 5 5, 3 3 (1) : 1
69[5 3 1 1, 5 5 (1) : 1
70[5 3 2 2, 5 5 (1) : 1
71[5 3 3 3, 4 4 (1) : 1
72[5 3 4 4, 3 3 (1) : 1
73[5 3 5 5, 3 3 (1) : 1

VITA

Erdal Kılıç was born in Kars on March 22, 1969. He received his B.S. and M.Sc. degrees in Electrical and Electronics Engineering from Karadeniz Technical University in January 1991 and October 1996, respectively. He worked in SEKA pulp and paper factories from August 1991 to March 1996. He also worked as a teaching assistant from October 1996 to November 1999 at the Ondokuz Mayıs University in the Department of Electrical and Electronics Engineering. Since November 1999 he has been a PhD student in the department of Electrical and Electronics Engineering at the Middle East Technical University. He has been in the Department of Delft Center Systems and Control of Delft Technical University in the Netherlands as a visiting researcher from February 2004 to November 2005. Since October 1999 he has been working as a teaching assistant in the department of Electrical and Electronics Engineering at the Middle East Technical University. His main areas of interest are systems, intelligent control, fuzzy logic, optimization and fault detection and diagnosis.

PUBLICATIONS

- Kılıç E., Leblebicioğlu K., Halıcı U., "Bulanık renkli Petri ağı kullanarak asenkron motorda hata teşhis ve tesbiti", Endüstri ve Otomasyon, vol. 64, pp. 42-47, July 2002.

- Kılıç E., Leblebicioğlu K., "Fault diagnosis with dynamic fuzzy discrete event system approach", accepted for publication in Lecture Notes in Artificial Intelligence.

- Kılıç E., Leblebicioğlu K., "Fault diagnosis with dynamic fuzzy discrete event system approach", The Proceedings of the14th Turkish Symposium on Artificial Intelligence and Neural Networks, pp. 55-62, Çeşme, İzmir, Turkey, 16-17 June, 2005.

- Kılıç E., Altaş İ.H., "Power factor correction of synchronous motor using fuzzy logic", Mathematical & Computational Applications, vol. 1, pp. 66-72, 1996.

- Kılıç E., Leblebicioğlu K., “Bulanık ayrık olaylı sistem yaklaşımı ile hata tanılama”, TOK-2005 Ulusal Otomatik Kontrol Konferansı, pp. 179-186, Istanbul Teknik Üniversitesi, İstanbul,Turkey, 2-3 Haziran, 2005 (In Turkish).

- Kılıç E., Leblebicioğlu K., Halıcı U., “Eğitilmiş üyelik fonksiyonları kullanarak bulanık karar ağacı ile doğal gazlı boru hatlarında hata bulma”, IEEE-SİU-2001, pp. 440-445, Girne, Cyprus, 2001 (In Turkish).

- Kılıç E., Leblebicioğlu K., Halıcı U., “Bulanık karar ağacı ile doğal gazlı boru hatlarında hata bulma”, IEEE-SİU-2000, pp. 165-169, Belek, Antalya 2000 (In Turkish).

- Kılıç E., Leblebicioğlu K., Halıcı U., “ Bulanık renkli Petri ağı kullanarak asenkron motorda hata teşhis ve tesbiti”, TOK-2001 Ulusal Otomatik Kontrol Konferansı, pp. 181-189, Bursa, 2001 (In Turkish).

- Kılıç E., Leblebicioğlu K., Halıcı U., “Bulanık ayrık olaylı sistem yapısı ve bulanık gözlenebilme kriteri kullanarak sistemlerde hata tanılama”, TOK-2002 Ulusal Otomatik Kontrol Konferansı, pp. 715-723, ODTÜ, Ankara 2002 (In Turkish).

- Kılıç E, Şenlik İ., Öbilgin G., “Bulanık mantık denetimli fırçasız doğru akım motoru”, Bursa 4. Bilgisayar-Haberleşme Sempozyumu, sayfa 210-214, Bursa, 1996 (In Turkish).

- Kılıç E, Şenlik İ., Öbilgin G., “Fırçasız doğru akım motorunun hızının denetimi için tasarlanan PI ve bulanık mantık denetleyicilerinin başarım karşılaştırması”, Elmeksem, Sayfa 305-311, Bursa, 1997 (In Turkish).

Journal papers to be submitted

- Kılıç E., Karasu Ç., Leblebicioğlu K., “Fault diagnosis in a Helicopter based on a dynamical fuzzy discrete event system approach”.

- Kılıç E., Leblebicioğlu K., “Fault diagnosis in an induction motor based on a fuzzy discrete event system approach”.