# DATA-BASED FAULT DETECTION AND ISOLATION (FDI) METHODS FOR A NONLINEAR SHIP PROPULSION SYSTEM

by

Fang Liu

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTERS OF APPLIED SCIENCE
in the School
of
Engineering Science



SIMON FRASER UNIVERSITY

July 2004

# APPROVAL

**Name:** Fang Liu

**Degree:** Masters of Applied Science

**Title of Thesis:** Data-based Fault Detection and Isolation (FDI) Methods for a Nonlinear Ship Propulsion System

**Examining Committee:** Dr. Paul K.M. Ho, Chair
Professor, School of Engineering Science
Simon Fraser University

---

Dr. Mehrdad Saif, Senior Supervisor
Professor, School of Engineering Science
Simon Fraser University

---

Dr. William A. Gruver, Supervisor
Professor, School of Engineering Science
Simon Fraser University

---

Dr. John D. Jones, Examiner
Associate Professor, School of Engineering Science
Simon Fraser University

**Date Approved:** July 27. 2004

# SIMON FRASER UNIVERSITY

## Partial Copyright Licence



Bennett Library
Simon Fraser University
Burnaby, BC, Canada

# Abstract

In recent years there has been increasing interest in designing highly reliable systems that are safe and economical. Developed in the past decades, fault detection and isolation (FDI) techniques have played an important role in improving the reliability of dynamic systems. There have been various developments in this area for linear as well as nonlinear systems both in theory and applications. The present thesis is a novel attempt to study and compare the performance of fuzzy model-based, signal processing and pattern recognition FDI approaches on a nonlinear ship propulsion benchmark. This research focuses on data-based FDI methods; that is, they rely on no prior knowledge and limited mathematical information about the benchmark system.

A pattern-recognition-method-based FDI compares calculated minimum distances between an unknown pattern and each faulty sample pattern. As a result, the unknown pattern is considered to belong to the faulty pattern to which the smallest distance is derived. This method is applied to the FDI on three faults.

A fuzzy model is built from the historical data. To enhance the model accuracy, a real-coded genetic algorithm (GA) is implemented to search for optimal model parameters. Simulations are presented which focus on the building of an accurate fuzzy model and the successful FDI scheme on three faults in the system.

Wavelet signal processing methods are capable of revealing signal characteristics in time-frequency (time-scale) domains by monitoring and analyzing particular signals. It is shown that sensor faults, including abrupt and incipient faults, can be detected by this approach.

The research reveals the advantages and disadvantages of the above-mentioned methods: Dynamic Time Warping is a simple method, however, it is only suitable

for off-line implementations; the GA fuzzy model-based method is effective when a great deal of attention is paid to build an accurate model; the wavelet decomposition approach works well with sensor faults, while it is not able to detect other faults. It is also shown that the combination of the wavelet signal processing and fuzzy model-based approaches offers a significant improvement in detecting and isolating all the six prototypical faults in the ship propulsion benchmark.

# Dedication

To my husband Hui and my parents.

# Acknowledgements

It is my great pleasure to acknowledge the help of many in making this thesis possible.

My greatest thanks go to my Senior Supervisor Professor Mehrdad Saif for his continuous guidance and support for my study and research. His enlightening suggestions and help on the problems I encountered made my research life easier and meaningful. I would like to extend my appreciation to Dr. John Jones for serving as my examiner and Dr. William Gruver for serving on my Supervisory committee. My great appreciation also goes to Sunjaya Djaja, Xueman Li, Weitian Chen, and Allan Johnson for their significant help in reviewing this thesis.

I would like to thank the colleagues in our Control and Fault Diagnosis Lab and School of Engineering Science. Those who I shared the past years with and offered great help and wonderful friendships: Wen Chen, Weitian Chen, Xueman Li, Mahta Boozari, Qingguo Li, Guoyu Wang, Guangqing Jia, Qing Wu, Jinyun Ren, Yifeng Huang, Tong Jin, and too many others to list here.

Last but not least, special thanks to my husband Hui for his continuous encouragement and considerate suggestions, to my mother for her care and love through all these years, and to my sister Ying and my little nephew Zhenwei for all the laugher he has given me.

# Table of Contents

# List of Figures

# List of Tables

# Table of Acronyms

| | |
|---|---|
| CUSUM: | cumulative sum value |
| CWT: | Continuous Time Wavelet Transform |
| DTW: | Dynamic Time Warping |
| DWT: | Discrete Time Wavelet Transform |
| FCM: | Fuzzy C-means Algorithm |
| FDI: | Fault Detection and Isolation |
| FD : | Fault Diagnosis |
| FIS: | Fuzzy Inference System |
| GA: | Genetic Algorithm |
| GK : | Gustafson-Kessel Algorithm |
| HPF: | High Pass Filter |
| IDWT: | Inverse Discrete Time Wavelet Transform |
| LPF: | Low Pass Filter |
| NNs: | Neural Networks |
| PCA: | Principal Component Analysis |
| PLS: | Partial Least Squares |

# Chapter 1

# Introduction

Modern control systems are becoming more and more complex, usually including a large number of components (such as sensors, actuators, and computers, etc.). They require more reliable operations since system malfunctions may cause serious safety problems. It is well known that the *Chernobyl* nuclear disasters caused great loss of life and severe health problems for the victims. Economic and environmental factors also demand that stabilities of most systems are improved. A series of missile launch failures cost the United States over $300 million during a short time period from August 1998 to May 1999. In order to maintain a high level of safety, quality and reliability in controlled systems, it is very important that abnormal system operations and component faults are detected promptly. Therefore, there has been a surge of interest in research and applications on fault detection and isolation (FDI) techniques in the last three decades. Since early development in 1970's, this area has matured with the conception of various FDI methods. The main purpose of all FDI methods is to monitor system operations and, in the case of faults, accommodate the source of the faults so that timely corrective actions are taken. System reconfiguration can be accomplished afterwards by human operators or automatic configurations to maintain

nearly normal operations. However this topic is beyond the scope of this thesis. This thesis presents effective FDI solutions for a nonlinear system applications.

## 1.1 Fault Diagnosis: Definitions and Methodologies

A *fault* in a system can be considered as an undesired change which tends to degrade the overall system performance [32]. Some faults, when not treated properly and promptly, will lead to serious system breakdown, which is known as a *failure*. From the fault diagnosis point of view, faults can be divided into three groups [46]:

1. Actuator faults: faults associated with the actuator, such as damage in the bearings, deficiencies in force or momentum and so on.

2. Sensor faults: faults occurring with sensors, such as scaling errors, drift, dead zones, and so on.

3. Component faults: faults happening in the framework of the process, such as a struck valve, a broken or leaking pipe and so on.

Faults also can be categorized into *additive faults* and *multiplicative faults*, in terms of how the faults influence the system variables. Additive faults influence the variables additively, such as offsets of sensors; while multiplicative faults usually affect the system parameters by a product factor. Another way of categorizing faults is time-based, namely *abrupt faults* and *incipient faults*. An abrupt fault represents an undesired and sudden change, while an incipient fault is a fault that causes undesirable drifts away from healthy operating value and gradually grows with time.

*Fault detection* simply involves a decision based on the monitored data as to whether there is a fault or the system is running normally. *Fault isolation* is then executed to identify the type and location of a fault after the fault detection has triggered an alarm so that corrective actions can be made. These two steps combined are known as *fault detection and isolation* (FDI). The term *fault identification* in some literature refers to identifying the fault magnitude and the time of occurrence. Another frequently used term, *fault diagnosis*, is generally referred to as the combination of fault detection, isolation and identification. Figure 1.1 is a general framework for a fault diagnosis system. Firstly, adequate process data needs to be collected. Then the symptoms of the faults are extracted by various methods. A *symptom* is defined as a deviation of an observable variable from normal conditions [36]. It presents the primary information about the faults. Symptoms can be taken as different variables, parameters or functions in various FDI methods. For example, in model-based FDI approaches, a well used symptom is the *residual*, which is further described in Chapter 3. The symptoms obtained from the previous efforts help to distinguish different faults when a fault has been reported. Detailed information about the fault, including the fault magnitude and time of occurrence, is derived from further studies by the procedure of fault identification.

In order to compare various FDI approaches, it is useful to identify a set of criteria, as introduced in [40]. To mention a few:

1. Quick detection: The FDI system should be able to detect and isolate faults quickly enough to avoid severe damage during abnormal situations.

2. Isolability: The system is capable of distinguishing between different faults based on the monitored process data.

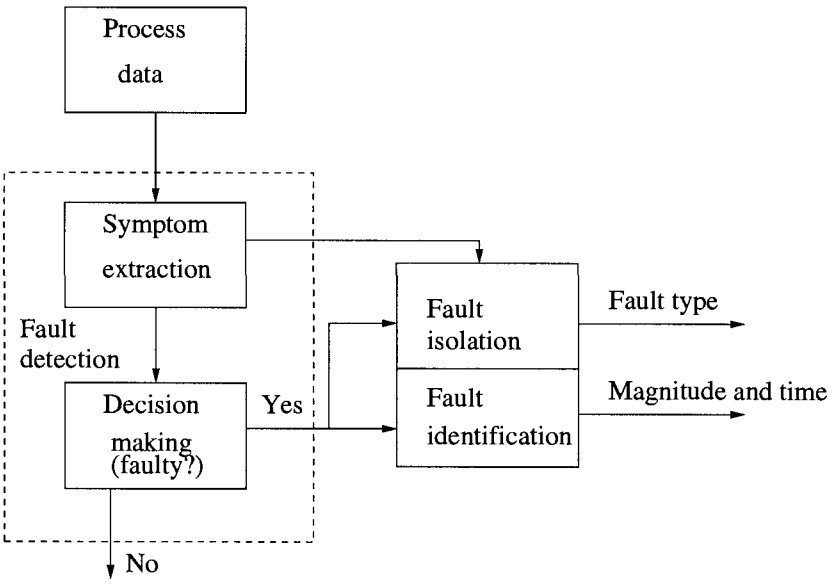


Figure 1.1: General framework for fault diagnosis

3. Robustness: The ability of the system to make a correct decision within the tolerance of system uncertainty and noise.

The effectiveness of the FDI method is also affected by factors other than faults, such as system uncertainties, noise and model mismatch. These might not influence the system operation very much under normal conditions, but can be very problematic to FDI. It is no surprise that there are trade-offs between robustness and the other two criteria, quick detection and isolability. How to detect and isolate faults quickly and correctly while being insensitive to uncertainties and noise is an important challenge to FDI approaches. In addition, incipient faults tend to be hidden by disturbances. Incipient faults do not necessarily cause immediate damage. However, if unheeded, they might lead to more serious problems.

The various FDI methods can be broadly classified into three categories: 1) model-based approaches, 2) classification methods, and 3) signal processing approaches. An

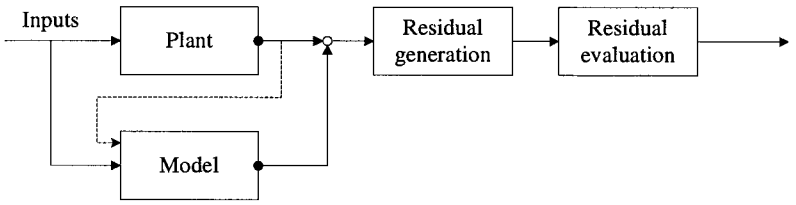


Figure 1.2: Scheme for the model-based fault detection

exhaustive classification and review can be found in [40] and [41].

### 1.1.1 Model-based FDI Techniques

As suggested by the name, model-based FDI approaches are based on models of the system. They are also associated with *analytical redundancy*, where the available inputs or measurements and a system model are used to cross-check the signal information so as to detect faults. Figure 1.2 depicts the general model-based FDI scheme. A system model is used to estimate the system behavior under normal operation. The deviations between model outputs and real system measurements generate the *residuals*. The residuals would be zero or small enough when the system is fault-free but noticeable when faults occur. The residuals are analyzed and evaluated for fault detection and isolation purposes.

There are two classes of model-based approaches to FDI: *mathematical models* based on the physics of the system and *data-based models*. In the former class, output observers, parity equations, and parameter estimations, etc. use static and dynamic relations among system variables to describe the system's behavior in physical or mathematical terms [36]. These approaches are based on the assumption that a fault causes changes to certain model parameters or states and detecting and isolating faults is possible by monitoring the estimated parameters or states. They make use

of mathematical functions and equations to estimate the system and its changes. One characteristic of these methods is that they require a comprehensive understanding and prior knowledge of the system physics and dynamics. In reality, the task of fault diagnosis is sometime difficult when prior understanding about the system is insufficient or incomplete. However, since the mathematical models reflect fundamental information of the system, these approaches are capable of detecting faults quickly, even incipient faults, which might not be easily noticed by other methods.

From a modelling perspective, there are data-based methods that do not assume any form of physics information and rely only on process history data to build the appropriate model. Among various data-based modelling methods, neural networks (NNs) have been an active area of research due to their learning ability, their "black box" properties, and their capability to model nonlinear systems [3]. Fuzzy systems also attract a lot of attention because of their interpretability over neural networks. The combination of neuro-fuzzy networks are actually fuzzy systems implemented using neural structures. Details are provided in [8]. Neural networks and fuzzy models are derived from the system's historical data so they can be viewed as "data-based models". These methods are attractive because the physical or mathematical models for complex systems are usually hard to obtain.

Since system parameters may change with time, and disturbances and noise which can affect the system behavior are unknown, it is not possible to obtain a perfect system model in practice. One important task of model-based FDI techniques is to deal with the model mismatch caused by inaccurate modelling, and the robustness problem caused by system uncertainties.

### 1.1.2 Classification Methods

The principle of the *classification* method is to classify the data into groups based on some specific features of each group. They are also referred to as pattern recognition methods since they are often applied to recognition problems [24]. The classifications are based on historical and monitored data. There are two main categories of classification: supervised and unsupervised classifications. The former needs reference patterns for learning. These reference patterns have been classified and labelled by prior work or experience. In the latter, there are no reference or sample patterns and we seek to find groups in the data and features which discriminate the groups.

There are many classification methods, including geometric, statistical, neural and fuzzy classifiers. Geometric classifiers use the notion of geometric distance. Distances between a certain symptom vector of the unknown pattern and that of different fault groups are calculated and the pattern is considered to belong to the group with the shortest distance [24, 25].

Multivariate statistical methods such as Principal Component Analysis (PCA) or Partial Least Squares (PLS) models are mainly used in on-line process monitoring [49]. Relative data processing algorithms are developed [10] to resolve outlier problems when PCAs are implemented. Dynamic process monitoring based on neural networks and PCA is implemented in [9]. Dynamic time warping has been used on speech recognition and is also applied in the fault diagnosis area [11, 22].

Neural networks and fuzzy inference systems use nonlinear discriminant methods and have also been applied in the field of FDI. They generate classification results by evaluating the input variables based on NNs or fuzzy systems, which are trained by the provided classification samples.

### 1.1.3 Signal Processing Technique Based Approaches

Signal processing approaches are another group of well-established methods used in practice. The signals which carry most information about faults of interest are studied and some particular symptoms are derived to represent features of the faults. Further decision making is then straightforward. These methods use the correlation function, covariance, power spectral densities or auto-regressive-moving-average models. Frequency analysis is suitable for detecting the signals which contain particular meaningful frequency information such as power system current. The wavelet transform is used in detecting abrupt sensor faults [47, 50]. Wavelet networks combine the attributes of the wavelet transform and neural networks and have applications in fault detection [52].

Signal processing methods focus on particular signal characteristics without bothering to reveal the system structure or properties in a deeper manner. They are capable of dealing with noise and can be used either independently or jointly with model-based approaches.

## 1.2 Thesis Outline

In this thesis, three different approaches to FDI for a nonlinear ship propulsion system are presented. This thesis is organized as follows:

In Chapter 2, a brief description of the ship propulsion system, which is the benchmark used throughout this thesis, and a fault scenario including six prototypical faults is given, and relevant research is reviewed. The following three chapters present FDI research on this benchmark from three perspectives. In Chapter 3, the principle and application of a pattern recognition method referred to as Dynamic Time Warping

(DTW) is introduced and then a simulation is implemented to detect and isolate specific faults in the ship propulsion system.

For a model-based approach, a genetic fuzzy model is derived in Chapter 4. Basic algorithms for building fuzzy models from numerical data, and an optimization scheme using a genetic algorithm working on a tentative model are illustrated. Generated residuals go through a residual evaluation process for detection and isolation decisions.

Chapter 5 develops a wavelet-decomposition-based signal processing method for detecting sensor faults. The goal of detecting all six faults is achieved using the combination of the fuzzy model-based approach and wavelet decomposition, which provides a complete and effective diagnosis. This concept can be extended to other applications.

In each chapter, the relative background knowledge is first introduced. Applications to FDI are illustrated and followed with simulation results on the ship propulsion system.

The differences, advantages and deficiencies of the three approaches are discussed in Chapter 6, then followed by concluding remarks and comments on future work.

# Chapter 2

# Ship Propulsion Benchmark

Many different FDI approaches for nonlinear systems have been developed, but only a few have been implemented. To evaluate various approaches, a nonlinear ship propulsion system is introduced in this thesis as a benchmark. The simulation model of the ship propulsion benchmark was defined and developed by Izadi-Zamamnabadi and Blanke at the University of Aalborg. The complete description of the benchmark is found in [18]. All the data used in this benchmark have been generated by a Simulink model of the ship propulsion system. In this chapter, the system description is first presented. Then the system fault scenario is described.

## 2.1 Ship Propulsion System Description

The ship propulsion benchmark has one engine and one propeller for a marine vehicle. The structure of the system is depicted in Figure 2.1.

The ship propulsion system consists of six main sections:

• The coordinated control part: calculates the set-points for the shaft speed $n_{ref}$ and the propeller pitch $\theta_{ref}$.

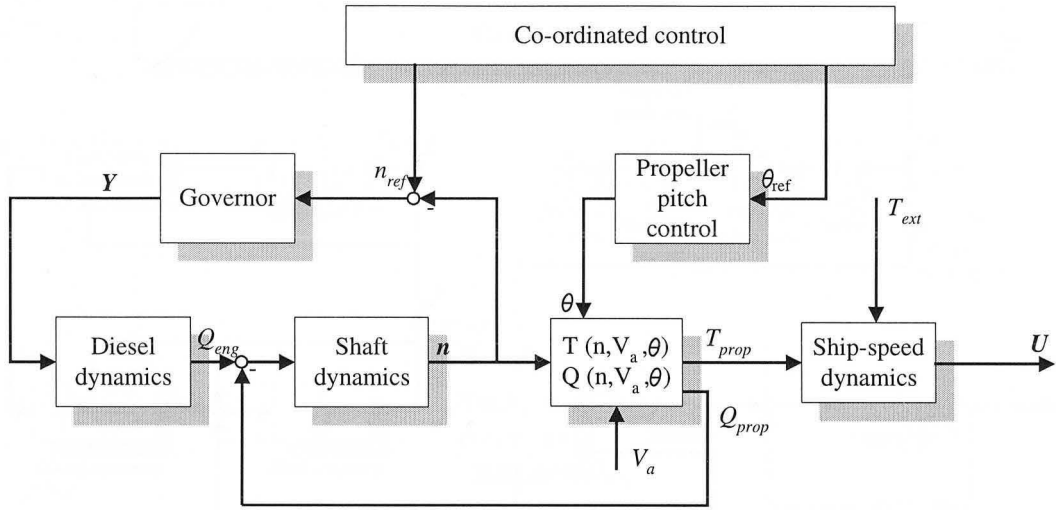


Figure 2.1: Structure of the ship propulsion System

- Propeller pitch control system: controls propeller pitch $\theta$.
- Governor: controls the fuel index $Y$ of the diesel engine.
- Diesel engine: generates torque $Q_{eng}$ which is then transformed to the shaft speed.
- Shaft dynamics: describes the shaft speed $n$ based on the difference between engine torque $Q_{eng}$ and propeller torque $Q_{prop}$.
- Propulsion characteristics: describes the propeller thrust $T_{prop}$ and torque $Q_{prop}$ based on shaft speed $n$, propeller pitch $\theta$, and water speed $V_a$.
- Ship speed dynamics: generates ship speed $U$ based on the propeller thrust $T_{prop}$, external force $T_{ext}$ and hull resistance.

The purpose of ship propulsion system is to maintain the ship's ability to propel and maneuver itself. There are two basic control loops in the propulsion system: the propeller pitch control loop and the shaft speed control loop (governor, diesel dynamics and the shaft dynamics). The later is coupled to the former through propulsion

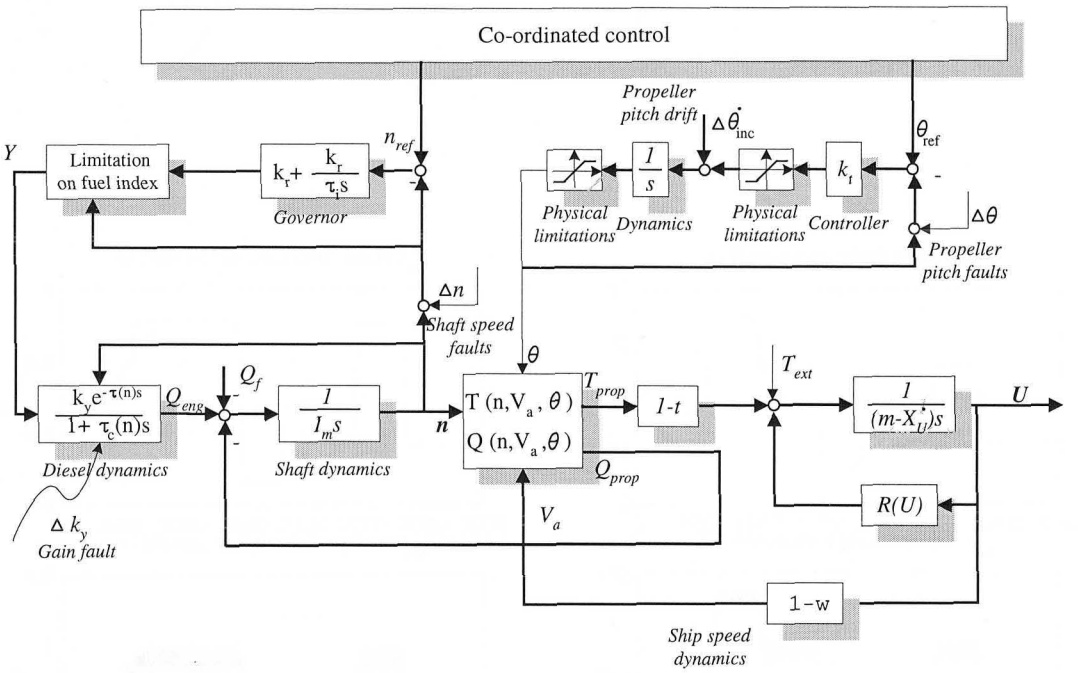


Figure 2.2: Detailed structure of the ship propulsion system with generic faults indicated

characteristics.

There are two known inputs for the propulsion system: the shaft speed set point $n_{ref}$ and the propeller pitch set point $\theta_{ref}$. There are unknown inputs like external force $T_{ext}$ and the friction torque $Q_f$. Measured outputs include the diesel engine shaft speed $n_{mf}$, the fuel index $Y_m$, the propeller pitch position $\theta_{mf}$, and the ship speed $U_m$. Figure 2.2 shows a more detailed system block diagram with the generic faults.

Figure 2.3 shows the input and output signals of the healthy system when the simulation runs for 3500 seconds.

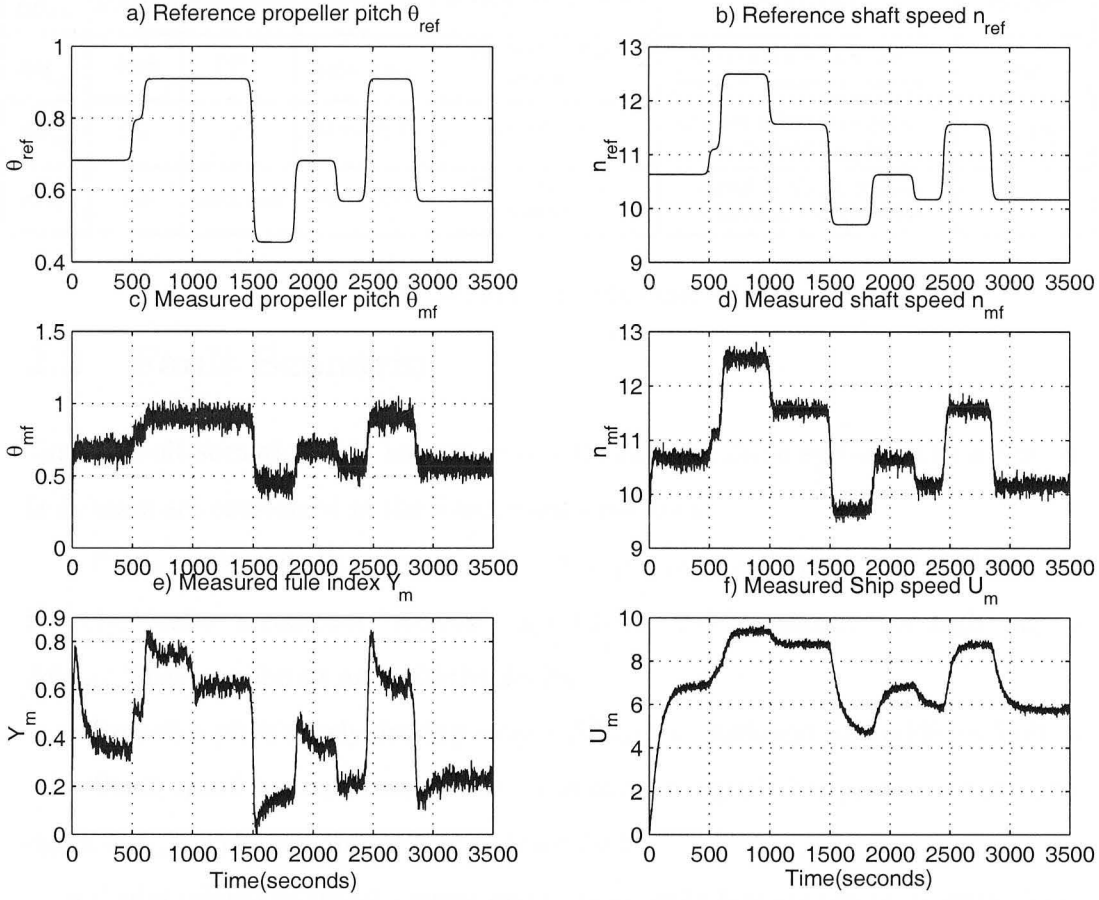


Figure 2.3: Simulated system inputs and outputs

| Fault | Type | Magnitude | Time period | End-effect | Consequences | Severity level |
|---|---|---|---|---|---|---|
| $\Delta\theta_{high}$ | *high* | *1* | *180s~210s* | decreased or reversed ship velocity | maneuvering risk delayed operation/cost increase | high |
| $\Delta\theta_{low}$ | *low* | *-0.7* | *1890s~1920s* | increased ship velocity(acceleration) | collision risk | very high |
| $\Delta\dot{\theta}_{inc}$ | *incipient* | $\frac{0.00001}{s+0.00001}$ | *800s~1700s* | gradual speed change | cost increase | medium |
| $\Delta n_{high}$ | *high* | *13*[a] | *680s~710s* | decreased ship velocity | delayed operation/ cost increase, maneuvering risk | high |
| $\Delta n_{low}$ | *low* | *4* | *2640s~2670s* | acceleration | collsion risk/cost increase over-speed danger | very high |
| $\Delta k_y$ | *low* | *20% loss* | *3000s~3500s* | decreased ship velocity | diesel overload/delayed operation, cost increase | medium |

Table 2.1: Six faults in the fault scenario

## 2.2 Fault Scenario

Various fault scenario could happen in a real ship propulsion system. The following fault cases are considered in the benchmark simulation:

- Pitch position faults: the propeller pitch position measurement could have the following faults: a constant "too low" signal $\Delta\theta_{low}$ and a constant "too high" signal $\Delta\theta_{high}$. They are abrupt and additive faults.
- Propeller pitch slowly drifting away : $\Delta\dot{\theta}_{inc}$ is an incipient and additive fault.
- Shaft speed measurement faults: a maximum signal $\Delta n_{high}$ and a minimum signal $\Delta n_{low}$. They are abrupt and additive faults.
- Faults related to diesel engine: one or more cylinders are out of function $\Delta ky$, an abrupt and multiplicative fault.

Table 2.1 lists six types of faults in the fault scenario, their magnitudes and time durations, as well as the end-effects, consequences, and the severity levels caused by the considered faults.

For various FDI schemes, the following requirements are introduced as criteria:

1. Quick detection: According to the severity of each fault type, sensor feedback faults ($\Delta\theta_{low}, \Delta\theta_{high}, \Delta n_{low}, \Delta n_{high}$) should be preferentially detected within 2 time samples. The incipient fault $\Delta\dot{\theta}_{inc}$ should be detected within 100 time samples, and the gain fault $\Delta k_y$ within 5 time samples.

2. Low false detection possibility.

3. Low missed detection possibility.

4. Robustness: FDI should be robust to system uncertainties and noise, which have been considered and simulated in this model.

## 2.3 System Models

The nonlinear models and detailed descriptions are given for the six subsystems.

### 2.3.1 Coordinated Control

As mentioned in section 2.1, the objective of the coordinated control part is to generate the reference control signals $n_{ref}$ and $\theta_{ref}$. Figure 2.4 illustrates the four main parts of the coordinated control subsystems.

- Combinator curve: sets coordinated command shaft speed $n_{com}$ and propeller pitch $\theta_{com}$ according to the handle position and the selected mode.

$$\begin{bmatrix} n_{com} \\ \theta_{com} \end{bmatrix} = F_{coor}(h, mode) \tag{2.1}$$

where $h$ stands for the handle position and *mode* is the selected mode between economy operation and maneuvering operation. Economy operation mode is used during long-distance voyage and cruise, and maneuvering operation mode is mainly used

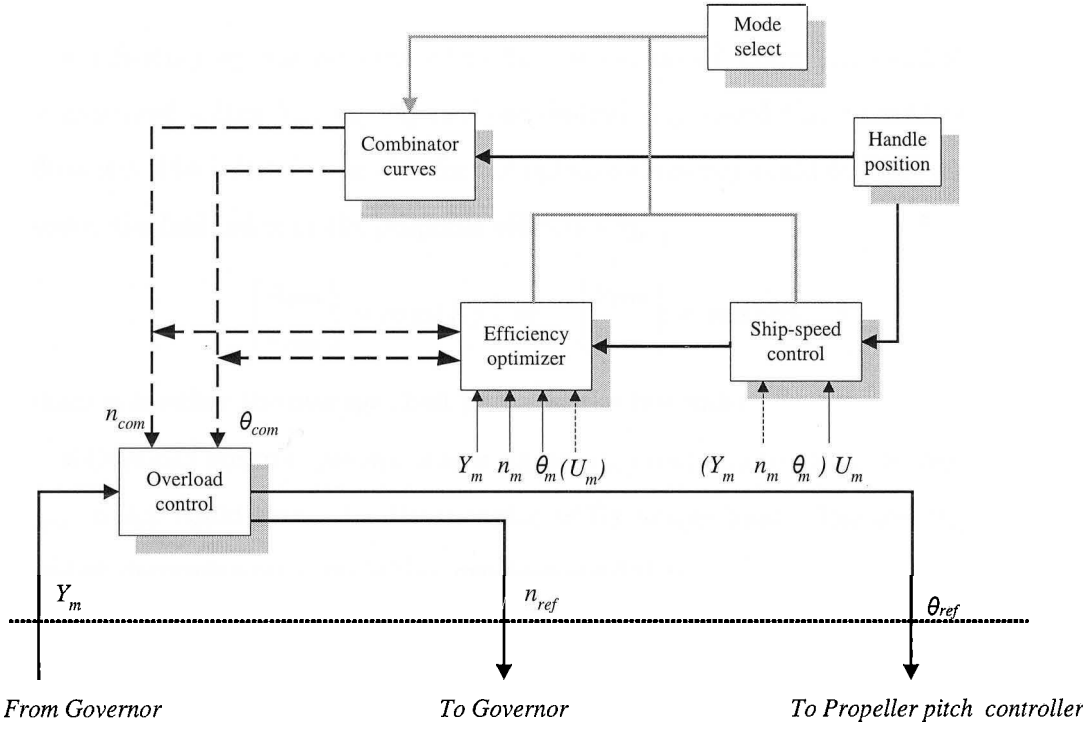


Figure 2.4: Structure of the coordinated control subsystems

when the ship is approaching harbor or sailing out of a harbor. $F_{coor}$ will generate different command values under the two modes for different voyage situation.

• Ship speed control: maintains a set ship speed $U_{ref}$ based on the measured speed $U_m$ and the handle position $h$.

$$U_{ref} = F_{speed}(h, U_m) \tag{2.2}$$

• Efficiency optimizer: determines the set command values $n_{com}$ and $\theta_{com}$ based on measured values $Y_m$, $n_m$, $\theta_m$ and the desired ship speed $U_{ref}$ to achieve optimal efficiency. The reference variable for the optimal efficiency could be the average shaft power, the fuel index or the propeller efficiency $\eta_{prop}$.

$$\begin{bmatrix} n_{com} \\ \theta_{com} \end{bmatrix} = min\{E\} \quad or \quad \begin{bmatrix} n_{com} \\ \theta_{com} \end{bmatrix} = max\{\eta_{prop}\} \tag{2.3}$$

where E is either the average shaft power or the fuel index.

• Overload control: generates new signals $n_{ref}$ and $\theta_{ref}$ to avoid too big $n_{com}$ and $\theta_{com}$, which could bring the diesel engine to its torque limit. The fuel index $Y_m$ is used to determine an approaching overload condition.

$$\begin{bmatrix} n_{ref} \\ \theta_{ref} \end{bmatrix} = F_{over}(n_{com}, \theta_{com}, Y_m) \tag{2.4}$$

For a thorough and detailed description of the functions, readers are referred to [20].

### 2.3.2 Propeller Pitch Control

The propeller pitch control subsystem use a P-controller to determine the propeller pitch $\theta$, as Figure 2.5 shows.

The following equations illustrate the relationship between the measured propeller pitch $\theta_m$, measurement noise $\nu_\theta$, the leakage fault $\Delta\dot{\theta}_{inc}$, and the pitch sensor fault $\Delta\theta$.

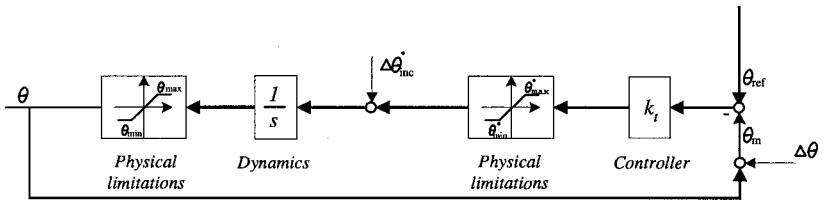


Figure 2.5: Structure of the propeller pitch control subsystem

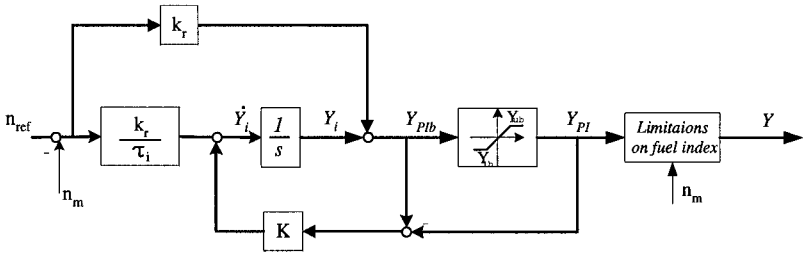


Figure 2.6: Structure of the governor subsystem

$\dot{\theta}_{max}$ and $\dot{\theta}_{min}$ are the rate limits set by the hydraulic pump capacity and geometry, while $\theta_{max}$ and $\theta_{min}$ are the physical limits for propeller blade travel.

$$\begin{aligned} \theta_m &= \theta + \nu_\theta + \Delta\theta \\ u_{\dot{\theta}} &= k_t(\theta_{ref} - \theta_m) \\ \dot{\theta} &= max(\dot{\theta}_{min}, min(u_{\dot{\theta}}, \dot{\theta}_{max})) + \Delta\dot{\theta}_{inc} \\ \theta &= max(\theta_{min}, min(\theta, \theta_{max})) \end{aligned} \tag{2.5}$$

### 2.3.3 Governor

The governor is basically a PI controller as shown in Figure 2.6. It uses the difference between the shaft speed reference $n_{ref}$ and the measured shaft speed $n_m$ as inputs and generates the output fuel index $Y$. An anti-windup gain $K$ is included to prevent the integrator from building up to a large value which will result in large overshoots. The PI controller has the following functions.

$$\begin{aligned} n_m &= n + \nu_n + \Delta n \\ \dot{Y}_i &= \frac{k_r}{\tau_i}((n_{ref} - n_m) - K(Y_{PI} - Y_{PIb})) \\ Y_{PIb} &= Y_i + k_r((n_{ref} - n_m)) \\ Y_{PI} &= min(max(Y_{PIb}, Y_{lb}), Y_{ub}) \end{aligned} \tag{2.6}$$

where $\nu_n$ represents the measurement noise, $Y_{lb}$ and $Y_{ub}$ are the lower and upper bounds, respectively, for the saturation of the actuator, and $\Delta n$ is the measurement fault.

### 2.3.4 Diesel Engine

Diesel engine dynamics generate a torque $Q_{eng}$ based on the fuel index $Y$. The following equations illustrate the relationships between $Q_{eng}$ and $Y$.

$$\frac{Q_{eng}}{Y}(s) = \frac{(k_y + \Delta k_y)e^{-\tau s}}{1 + \tau_c s} \tag{2.7}$$

where $\tau_c$ denotes the time constant and $k_y$ is the gain constant. The calculation of time delay $\tau$ is provided in [20].

### 2.3.5 Shaft Dynamics

The following differential equation illustrates the shaft speed $n$ based on the engine torque $Q_{eng}$, the propeller torque $Q_{prop}$, and the friction torque $Q_f$.

$$I_m \dot{n} = Q_{eng} - Q_{prop} - Q_f \tag{2.8}$$

### 2.3.6 Propulsion Characteristics

The developed propeller thrust and propeller torque are determined by the equations:

$$\begin{aligned} T_{prop} &= T_{|n|n}(\theta)|n|n + T_{|n|V_a}(\theta)|n|V_a \\ Q_{prop} &= Q_0|n|n + Q_{|n|n}(\theta)|n|n + Q_{|n|V_a}(\theta)|n|V_a \end{aligned} \tag{2.9}$$

where $V_a$ denotes the velocity of the water, and $Q_0$ represents the torque produced by the CP-propeller when the pitch is zero. The coefficients $T_{|n|n}, T_{|n|V_a}, Q_{|n|n}, Q_{|n|V_a}$ are complex functions of the pitch angle $\theta$.

### 2.3.7 Ship Speed Dynamics

The ship speed dynamics are described by the nonlinear differential equations:

$$\begin{aligned} (m - X_{\dot{U}})\dot{U} &= R(U) + (1-t)T_{prop} - T_{loss} - T_{ext} \\ U_m &= U + \nu_U \end{aligned} \tag{2.10}$$

where $R(U)$ is the hull resistance of the ship in the water and $X_{\dot{U}}$ denotes the added mass in surge. The $T_{ext}$ represents the external force imposed on the ship motion by wind and waves, $T_{loss}$ is the excess drag force to the rudder, and $\nu_U$ is the measurement noise.

## 2.4 Prior Research on The Ship Propulsion Benchmark

There has been a considerable amount of research on the implementation of the ship propulsion benchmark to test various FDI and fault tolerant control methods. Among them some work focuses on FDI of three faults, including the two shaft speed sensor faults and diesel engine gain fault. Bemdtsen and Izadi-Zamanabadi in [3] trained a neural network to model the input-output relationships of the structure. Residuals were then derived and an adaptive cumulative sum value (CUSUM) algorithm was designed to isolate the shaft speed sensor faults and diesel engine gain fault. There were no details about the detection time in this paper. The statistical method CUSUM

was also used by Blanke [4] with the emphasis on analyzing re-configuration possibilities. Izadi-Zamanabadi, Blanke, and Katebi [21] built a neural fuzzy logic system based on structural modelling, then applied an adaptive threshold to a residual to detect faults. Their simulation showed that it is possible to isolate gain and shaft speed faults from this single residual under certain assumptions and knowledge of the possible occurrence of the two faults. This paper did not provide any information on detection and isolation time for these two faults. In works of Blanke and Lootsma, adaptive observers were designed for detecting and isolating the three faults [5]. Good detection was found possible though isolation was only possible if assumptions were made about the fault signature. Nonlinear observers were also implemented on FDI of the three propeller pitch sensor faults. The detection results were compared by Schreier in [33].

There is also research concerning all the six faults. Geometric approaches have been used to implement on FDI for ship propulsion system [25, 26]. Five of the six faults were detected; however, noise pollution was not considered and the incipient fault was not detected. Only two of the detected five faults met the required detection time. An adaptive two-stage extended Kalman filter was built and a set of statistical detection variables were formed by Zhang [53] to detect the occurrence of a fault and furthermore, to identify the fault type. Their work fulfilled some of the benchmark requirements in the face of some prescribed perturbations in the model and disturbances of external signals. In addition, Amann et.al [1] designed five fuzzy output observers and compared the results from the five observers in detecting different types of faults. The results showed that the fuzzy observers provided satisfactory properties in the detection of sensor faults, but did not enable the detection of other faults.

# Chapter 3

# Pattern-Recognition-Based FDI

By definition, *pattern recognition* refers to the discrimination between different classes. From the perspective of pattern recognition, many instances of the problems being studied can be represented as patterns containing specific characteristics, and our task is to discriminate between them. This can also be referred to as *classification.* Pattern recognition problems are very often encountered in industry as well as in our daily life. For example, classifying a thousand people into three groups of a) seniors, b) middle-aged persons, c) children is a simple pattern recognition task. The application of pattern recognition to industry and research is to explore mathematical and technical aspects of different patterns, and to create various techniques to discern them. It is an interdisciplinary area covering engineering, artificial intelligence, computing science, biology and many other fields.

Pattern recognition techniques can be broadly divided into two families: supervised classification (discrimination) and unsupervised classification (clustering) [45]. In supervised classification we have a set of data samples with associated labels or class types. In unsupervised classification, the raw data have no labels and known characteristics, so we have to discover the data features which distinguish different

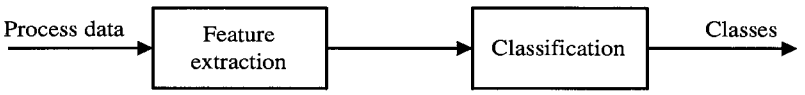


Figure 3.1: General scheme of pattern recognition

data types. Pattern recognition can be considered as a method for performing data-mining tasks, which is especially true for the unsupervised pattern recognition.

The process of pattern recognition generally consists of two parts: feature extraction and classification, as presented in Figure 3.1. After a set of data associated with the problem of interest has been collected, feature extraction identifies the characteristic of each class to distinguish among them. A feature extractor is used to perform this critical step. The next step is to design a decision scheme that uses the extracted features to classify the patterns in an optimal way. A classifier performs this classification. The information extracted by the feature extractor may suggest what type of classifier to use for a given application.

The following classification methods have been used to solve FDI problems [24]:

• Statistical Classifiers : When the statistical distribution functions of the classes are known, a statistical classifier is an appropriate approach. The well-known Bayes classifier, based on Bayes formula from probability theory, minimizes the expected probability of misclassification. A Priori knowledge of pattern distribution functions for each class is needed to perform Bayes classification. Multivariate statistical methods, and in particular Principle Component Analysis (PCA) and Partial Least Squares (PLS) are designed to model correlation structures among the process variables. The correlation structures are imposed by the physical mechanism which governs the process operation. These methods optimally reduce the large number of correlated variables into a small number of fictitious uncorrelated variables (i.e. principal components), so that the evolution of the process can be observed in the space of principal

components.

• Geometric Classifiers: These are based on computation of geometric distances. A minimum distance classifier is adopted if each class is represented by a single prototype and a new pattern belongs to the class with the minimum distance prototype. The nearest-neighbor classifier distinguishes a new pattern by measuring its distances from the sample patterns and choosing the class to which the nearest neighbor belongs.

• Fuzzy Classifiers: Fuzzy classifiers are good at dealing with uncertainties in pattern recognition. A pattern may belong to several classes with various grades of membership from 0 to 1. Fuzzy clustering algorithms perform the classification by giving each unknown pattern a set of grades of membership which determine the degree to which it belongs to the corresponding classes [7, 42].

• Neural networks: Neural networks are built with a structure of a input layer, an output layer, and hidden layers. A neural network is trained with a sample pattern by adjusting the weights of the network. It can be used to classify arbitrary unknown patterns [39].

To implement the FDI using pattern recognition, a general framework requires the following steps:

1. The collection of the historical data (i.e. patterns) from the normal operation and from all the faults of interest.

2. The extraction of the features from collected historical data.

3. Based on features derived from the previous step, a decision is made about which particular class a new pattern belongs to. The FDI task is performed by deciding whether the new pattern belongs to a healthy class or a particular faulty class.

Since building an accurate dynamic nonlinear model and stochastic disturbance model is a large task for many industrial processes, pattern recognition has significant advantages and is a widely used method in FDI. Pattern recognition methods need no insight into system models and inner structures. They perform the FDI task based on collected data. Therefore, adequate data are necessary to extract required characteristics from the data. When the mathematical model of a system is not readily available, while historical data (both healthy and faulty) is easy to get, pattern recognition can be a good approach.

## 3.1 The Principle of Dynamic Time Warping

The Dynamic Time Warping (DTW) method is a flexible pattern matching method, originally used in speech recognition for identifying isolated and connected words [37]. It is capable of compensating for the temporal mismatch caused by different speech speeds. The principle of DTW is to compare two dynamic patterns that may not be perfectly aligned and measure the similarity by calculating a minimum distance between them. Patterns to be tested are considered to have similar, but possibly expanded or contracted temporal correlations. DTW uses the principle of dynamic programming to nonlinearly warp the patterns and shift, extend or compress the patterns in such a way that the similar characteristics are captured to an optimal extent.

Suppose two time sequences, $x$ and $y$, possess similar characteristic shapes but these shapes do not line up on the time axis. In Figure 3.2 (a), the traditional Euclidean distance generates a pessimistic similarity measure because of the assumption that the $i^{th}$ point in one sequence is aligned with the $i^{th}$ point in the other sequence, where the dashed lines show the aligned pairs. In order to find the similarity between

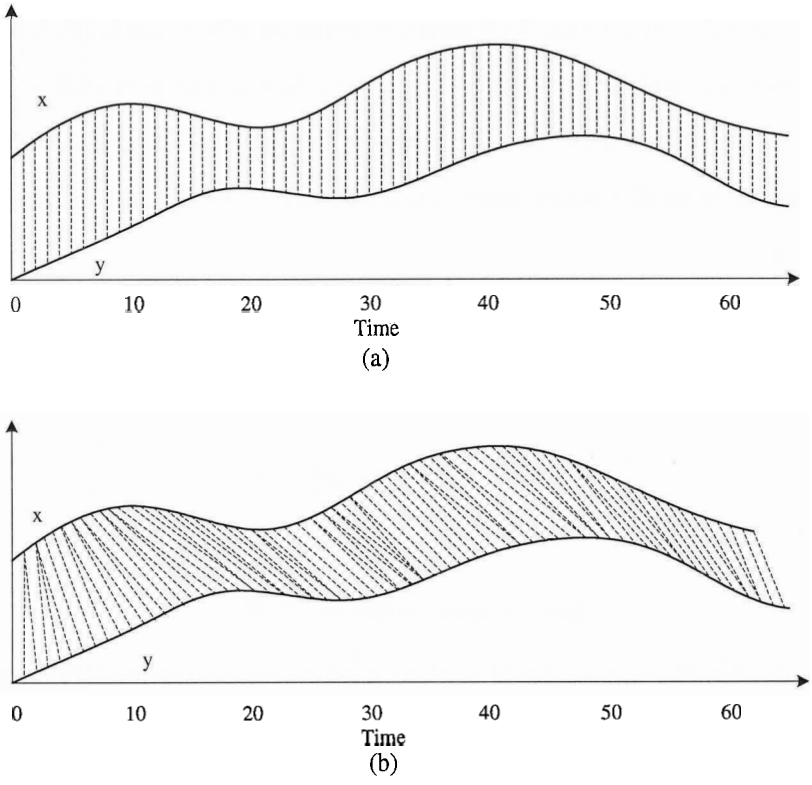


Figure 3.2: Similarity measurements by: (a)Euclidean distance (b) Dynamic time warping

these two sequences, we need to "warp" the time axis of one sequence to achieve a better alignment. DTW is able to find an alignment between the two sequences that allows a more sophisticated distance measure to be calculated. Rather than comparing the value of the sequence $x$ at point $i$ to the sequence $y$ at the same point $i$, DTW searches the space of mappings between points from the time sequence $x$ to that of $y$, as Figure 3.2 (b) depicts. For example, we may find that point $i$ in sequence $x$ corresponds to $i+6$ in sequence $y$, and $j$ in sequence $x$ corresponds to $j-2$ in sequence $y$. As a result, a minimum distance is eventually generated between two sequences and the similarity is symbolized by the distance, which doesn't have a physical meaning in this figure.

Suppose there are two time sequences of patterns $S$ and $U$, of length $m$ and $n$, respectively:

$$S = \{s_1, s_2, \ldots, s_i, \ldots, s_m\} \tag{3.1}$$

and

$$U = \{u_1, u_2, \ldots, u_j, \ldots, u_n\} \tag{3.2}$$

Firstly, DTW constructs an $m$-by-$n$ matrix $D$ where the $(i^{th}, j^{th})$ element of the matrix contains the local distance $d(s_i, u_j)$ between the two points $s_i$ and $u_j$. Each matrix index $(i, j)$ corresponds to the alignment between the point $s_i$ and $u_j$. The local distance is calculated by:

$$D(i,j) = d(s_i, u_j) = (s_i - u_j)(s_i - u_j)^T \tag{3.3}$$

The warping path $R$ of length $K$ is a continuous set of matrix indices $[i, j]$ that defines a mapping between $S$ and $U$.

$$R = r_1, r_2, \ldots, r_k, \ldots, r_K \quad and \quad max(m,n) \leq K \leq m+n \tag{3.4}$$

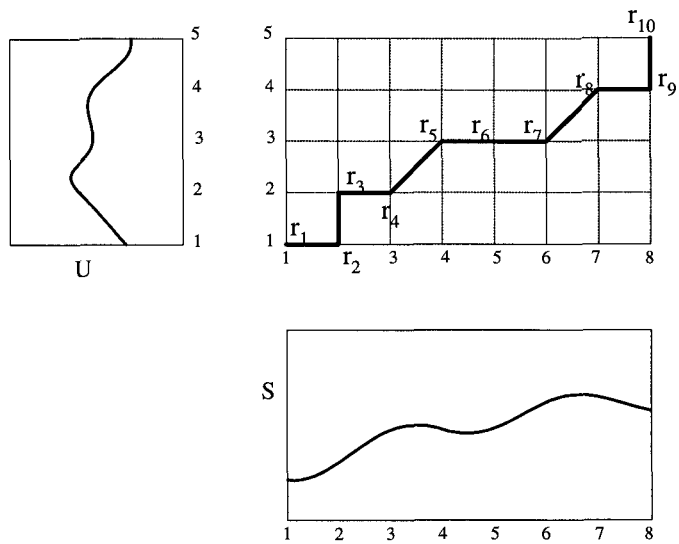


Figure 3.3: Warping path of an 8-point time sequence $S$ and a 5-point time sequence $U$

The $k^{th}$ element of $R$ is defined as:

$$r_k = [i, j]_k \tag{3.5}$$

As a simple example, consider an 8 point time sequence $S$ and a 5 point time sequence $U$, as shown in Figure 3.3. The element of $D(4,3)$ is the distance between $s_4$ and $u_3$. The warping path, which is subject to certain constraints and is searched by the principle of dynamic programming, contains 10 points in this figure. The point $r_5 = [4,3]$ demonstrates that the alignment (or mapping) between $s_4$ and $u_3$ is on the warping path.

The warping path is typically subject to several constrains:

• Endpoint constraints: The first and the last point of the two patterns are matched together, therefore:

$$r_1 = [1,1] \qquad r_K = [m,n] \tag{3.6}$$

• Monotonicity constrains: The patterns should be compared in time order. Given $r_k = [i, j]$ and $r_{k+1} = [i', j']$, $i' \geq i$ and $j' \geq j$ is satisfied.

• Local continuity constraints: This restricts the allowable steps in the warping path to adjacent points (including diagonally adjacent points). Given $r_k = [i, j]$ and $r_{k+1} = [i', j']$, the following is satisfied: $i' - i \leq 1$ and $j' - j \leq 1$. This constrains the warping path, such that local point $(i, j)$ can only be reached from points $((i-1), j)$,$((i-1), (j-1))$, or $(i, (j-1))$.

Figure 3.4 is a sample of two univariate patterns under the above mentioned constraints. There are many warping paths satisfying these constraints. The principle

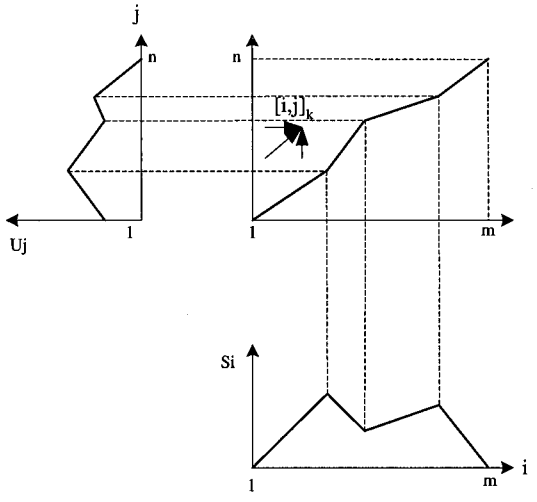


Figure 3.4: DTW comparison between two univariate patterns under the endpoint, monotonicity and local continuity constraints

of dynamic programming is used to search the warping path iteratively and for each iteration, a minimum distance is calculated.

The cumulative distance $D_c$ at point $(i, j)$ is defined as the weighted sum of the local distance $d(s_i, u_j)$ and the minimum of the cumulative distances in the previous

cells. This can be expressed as:

$$D_c(i,j) = min \left\{ \begin{array}{c} D_c(i-1,j) + d(s_i,u_j) \\ D_c(i-1,j-1) + 2d(s_i,u_j) \\ D_c(i,j-1) + d(s_i,u_j) \end{array} \right\} \tag{3.7}$$

where the weight 2 is used for cumulative distance calculation when the current point in warping path is reached via a diagonally adjacent cell. The initial condition is:

$$D_c(1,1) = 2d(s_1,u_1) \tag{3.8}$$

The final accumulated distance is defined as:

$$D_{dtw} = \frac{D_c(m,n)}{m+n} \tag{3.9}$$

This algorithm can be extended to multivariate time sequences $S$ and $U$:

$$S = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_i, \ldots, \mathbf{s}_m\} \tag{3.10}$$

and

$$U = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_j, \ldots, \mathbf{u}_n\} \tag{3.11}$$

where $\mathbf{s}_i$ and $\mathbf{u}_j$ are vectors with the same number of variables.

## 3.2 Dynamic-Time-Warping-Based FDI

Faults or abnormal events in continuous processes usually generate dynamic patterns consisting of a large number of process variables. Classification methods can be used to assess these dynamic patterns. To extract the corresponding feature from a particular faulty case, one needs to investigate the process variables and data. However, it is unlikely that every occurrence of a fault will make the process deviate from the normal

operating condition in exactly the same way. From the fault diagnosis point of view, one challenge is that the patterns arising from the same fault may last for different times and exhibit different magnitudes. Thus, our pattern recognition methods should be able to discriminate the faults despite varying characteristics, i.e. two sets of data arising from the same fault should be put in the same group even though the fault occurred at different times and has different magnitudes.

Several pattern recognition methods have been proposed to address these requirements. Neural networks (NNs) and Principal component analysis (PCA) have been successfully applied to FDI in large industrial systems. Since many of these methods assume steady-state conditions, when the dynamics of the system change erroneous fault diagnosis may occur [22]. DTW has been used to successfully isolate the faults in dynamic multivariate processes where fault detection has already been performed[22].

The data is preprocessed before being calculated by the DTW algorithm, so as to fit the algorithm better. For a distance measurement based algorithm like DTW, step or abrupt ramp signals of different magnitudes tend to increase the distance measurements unnecessarily. Furthermore, as mentioned in the previous section, this algorithm should be able to classify the patterns independently of the magnitude of the fault. Therefore, measures should be taken to get rid of the magnitude and time difference while distorting the original patterns to the minimum extent. A traditional way to scale a set of data is to subtract the average and divide by the standard deviation. The "average"and the "standard deviation"of a step-type signal depend on how many data points are included before and after the occurrence of the step. Subtracting the estimated "average" and dividing the raw data by the "standard deviation" will render the scaling procedure dependent on the duration of the pattern. This is an undesirable side effect since the method should be independent of the

duration of the pattern. A desirable method is able to remove the nonstationarity independent of the duration of the signal. For example, a high-pass filter can remove low frequency components like steps or ramps from a signal. After the high-pass filtering, the data are normalized to a standard deviation of one. This is to remove the effect of the magnitude of the remaining frequency components and the engineering units. Since white noise will degrade the algorithm performance, data also need to be processed to remove the high frequency noise. A low pass filter is then employed to deal with high frequency noise.

The unknown pattern must go through these preprocessing steps before the DTW algorithm is applied. Consequently, the similarity assessment scheme works with patterns that are different from those present in the raw data.

Generally, DTW based FDI performs the comparison as follows:

1. The unknown pattern, a healthy pattern, and a set of dynamic patterns of known past faults are collected.

2. All the patterns are filtered with a high-pass filter.

3. All the patterns are normalized to a standard deviation of one.

4. All the patterns are filtered with a low-pass filter.

5. The preprocessed unknown pattern is compared with each of the preprocessed faulty and the healthy pattern to get a series of distance measurements.

6. The unknown pattern is considered to belong to the group with which it has the smallest distance. Further details may be found in [22].

## 3.3 DTW Algorithm Applied to Ship Propulsion Benchmark

The above-mentioned method is applied to the ship propulsion benchmark. We consider three types of faults in the simulation: velocity measurement "too high" $\triangle n_{high}$, velocity measurement "too low" $\triangle n_{low}$, and gain fault $\triangle k_y$.

In this research, the sample patterns are created by simulating 1000 seconds with only one fault happening during the simulation time. Four sample patterns are collected in $S_1, S_2, S_3$,and $S_4$. The fault type, duration, and magnitude of these sample patterns are shown in Table 3.1. The magnitude of the fault in each sample pattern is chosen based on the fault scenario described in Chapter 2. The healthy data is also collected as a fourth pattern. Thus, by comparing dynamic patterns with the four existing patterns, the unknown pattern is classified as the type corresponding to the sample from which it has the smallest distance.

| Sample Pattern | Fault | Fault time duration | Magnitude | Simulation time (s) |
|---|---|---|---|---|
| $S_1$ | $\Delta n_{high}$ | 300s-600s | 13 | 1000 |
| $S_2$ | $\Delta n_{low}$ | 200s-600s | 5 | 1000 |
| $S_3$ | $\Delta k_y$ | 500s-900s | 0.2 | 1000 |
| $S_4$ | None | N/A | N/A | 1000 |

Table 3.1: Sample pattern descriptions

The output variables in the three considered faulty cases of $\triangle n_{high}$, $\triangle n_{low}$, and $\triangle k_y$ are studied for the purpose of detection and isolation. In order to be computationally efficient, only $n_{mf}$ is analyzed and taken as the one-dimensional pattern for this approach. Figure 3.5 depicts the simulated $n_{mf}$ of the four sample patterns:

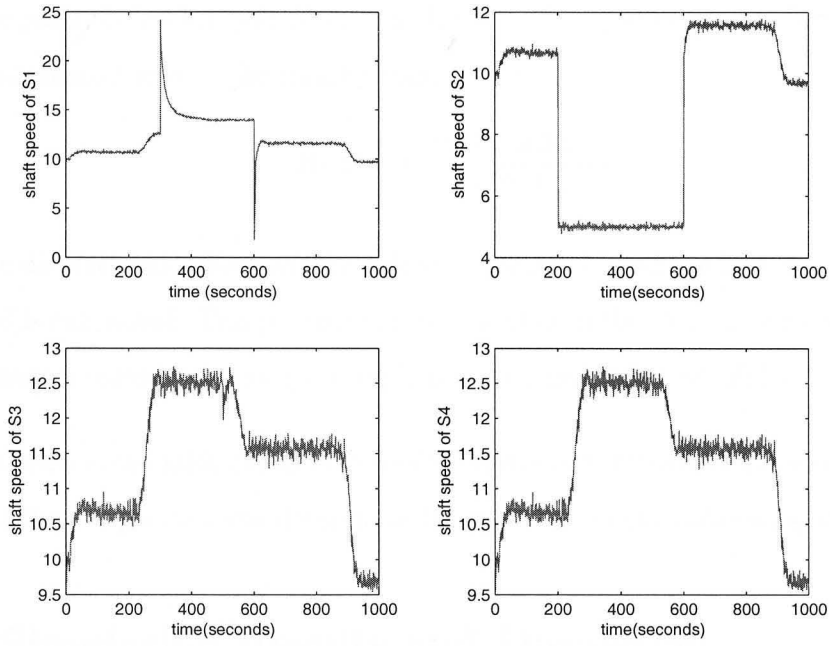


Figure 3.5: Simulated $n_{mf}$ signals of the four sample patterns

The $n_{mf}$ signal of faulty cases and healthy case are stored in a vector group $S's$, and that of the unknown data is stored in vector $U's$. The algorithms go through the following steps:

1. All the patterns are filtered by a high-pass filter. There is a tradeoff between filtering out the unwanted low frequency components and keeping the other parts of the data. The first-order high-pass filter does not have a sharp frequency response, but produces smoother output signals with the least distortion of the other components. Therefore, a first-order Butterworth filter is chosen for this purpose with cut-off frequency of 0.02 Hz. The transfer function is:

$$H_h(z^{-1}) = \frac{0.9695 - 0.9695z^{-1}}{1 - 0.9391z^{-1}} \tag{3.12}$$

2. All the patterns are normalized to a standard deviation of one.

3. The patterns are low pass filtered with a cut off frequency of 0.28 Hz to remove measurement noise. The transfer function is:

$$H_l(z^{-1}) = \frac{0.32 + 0.32z^{-1}}{1 - 0.36z^{-1}} \tag{3.13}$$

4. The distance value between the unknown pattern $U$ and the first sample vector in $S$ is calculated. This procedure is repeated until the comparison between the unknown pattern and all the sample patterns have been calculated.

5. In the decision making stage, the smallest distance between the unknown pattern and the sample patterns determines the diagnosis of the unknown pattern.

## 3.4 Simulation Results and Discussion

The simulation runs 10 times to randomly generate 10 different patterns, each with varied fault conditions, varied fault durations, and varied fault magnitudes. Table 3.2 presents the detailed information of these patterns and the simulation results. For each unknown pattern $U$, the first four columns of the table describe the fault information. The fifth to eighth columns are the calculated minimum distance to the four sample patterns $S_1, S_2, S_3$, and $S_4$, respectively. The smallest distance is underlined in each row. The details of four sample patterns are shown in Table 3.1. The last column in Table 3.2 is the decision of the fault isolation based on the minimum distance. It shows that in the 10 simulated runs, decisions based on the cumulative distance are correct.

Figure 3.6 is the warping path for calculating the distance between unknown pattern $U_9$ and four sample patterns.

There are several points we need to pay attention to for this approach:

| Unknown patterns | Fault type | Time duration /Magnitude | Simulation time | $S_1$ | $S_2$ | $S_3$ | $S_4$ | Decision |
|---|---|---|---|---|---|---|---|---|
| $U_1$ | $\Delta n_{high}$ | 350s-500s/13 | 800 | 0.0106 | 0.231 | 0.197 | 0.2473 | $S_1$ |
| $U_2$ | $\Delta n_{low}$ | 440s-600s/5 | 800 | 0.2936 | 0.0212 | 0.2037 | 0.1751 | $S_2$ |
| $U_3$ | None | N/A | 800 | 0.2651 | 0.1468 | 0.0458 | 0.0270 | $S_4$ |
| $U_4$ | $\Delta k_y$ | 450s-650s/0.4 | 800 | 0.2432 | 0.1742 | 0.0468 | 0.0644 | $S_3$ |
| $U_5$ | $\Delta n_{low}$ | 350s-480s/5 | 1000 | 0.2368 | 0.0205 | 0.2003 | 0.1755 | $S_2$ |
| $U_6$ | $\Delta n_{high}$ | 680s-780s/9 | 1000 | 0.0378 | 0.2415 | 0.2051 | 0.2431 | $S_1$ |
| $U_7$ | $\Delta n_{high}$ | 560s-680s/13 | 1000 | 0.0099 | 0.2522 | 0.2240 | 0.2728 | $S_1$ |
| $U_8$ | $\Delta n_{low}$ | 240s-310s/5 | 600 | 0.2156 | 0.1529 | 0.3018 | 0.2615 | $S_2$ |
| $U_9$ | $\Delta k_y$ | 660s-1100s/0.2 | 1100 | 0.1998 | 0.2113 | 0.0394 | 0.0724 | $S_3$ |
| $U_{10}$ | $\Delta k_y$ | 350s-1000s/0.2 | 1000 | 0.2399 | 0.1598 | 0.0001 | 0.0332 | $S_3$ |

Table 3.2: DTW results for three possible fault cases and healthy case

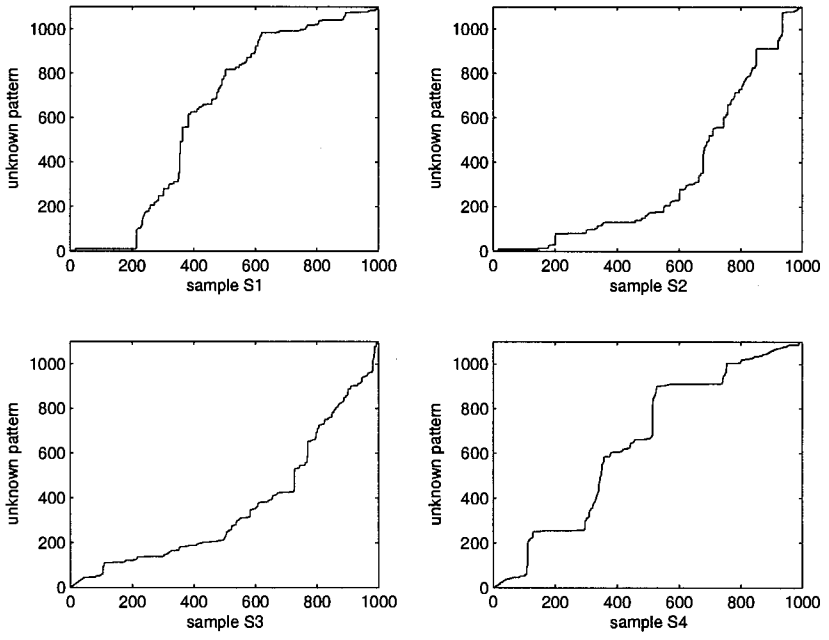


Figure 3.6: Warping path for calculating the DTW of unknown pattern $U_9$

1. There are no special requirements on choosing the four sample patterns. Each sample pattern of a reasonable time duration symbolizes a particular fault pattern. The time duration for these sample patterns are not necessarily the same. Since only one fault is included in one pattern, the time duration of 1000 seconds is chosen only for the reason of simplicity.

2. While some FDI methods assume that the system is in a steady state, this method is capable of dealing with dynamic problems.

3. The results show that DTW can perform fault detection and isolation in one step by including the healthy pattern as one of the sample patterns. Otherwise a separate detection step may be required while DTW is used to perform fault isolation.

4. The smallest distance should be significantly smaller than the rest of distances. If the DTW results show several samples with distances smaller than the rest of the samples, and among which distances are similar, then it could imply a new unknown pattern or more consideration needs to be taken into account for the right decision. The DTW distance between the unknown sample and each fault sample is a random variable, as is the difference between these distances. The statistical distribution of these random variables and the differences between them will determine the best choice of criteria both for detecting and isolating the correct fault, and for deciding when there is insufficient information to do so. Hence, developing rigorous statistical criteria to improve the isolation performance is recommended as future work.

5. The disadvantage of DTW method is that it is an off-line approach. This is because the pattern recognition methods need entire information of the fault, including starting states and ending states to classify the pattern. Thus it is not suitable for a real-time implementation. However, it is a useful tool to analyze the system off-line and to find the right patterns.

# Chapter 4

# Genetic Fuzzy Model-Based FDI

The model-based FDI method has attracted a great amount of research interest. As illustrated in Chapter 1, the main concern with this method is building an accurate model to estimate the system's normal behavior. Since complex industrial systems are generally nonlinear, the focus has been on nonlinear modelling.

There are many different ways to build a nonlinear model to predict the system's behavior based on numerical data. Neural networks (NNs) and fuzzy systems are two areas where considerable progress has been made in the past decades [15, 39, 42].

NNs consider a fixed topology of neurons connected by links in a predefined way. The weights between these connections are initialized by random numbers. The weights are refined by the training data and an optimal solution is sometimes achieved. However, since the knowledge is encoded within a "blackbox", it lacks the ability to explain itself in a human-comprehensible way.

Fuzzy systems express the relationship between variables in the form of "if-then" rules. A fuzzy system as a universal approximator is capable of uniformly approximating any nonlinear function on a compact universe to any degree of accuracy [6, 43]. It has also been proven that fuzzy systems are functionally equivalent to a class of radial

basis function networks [30]. The ease of interpretability of fuzzy systems makes them more understandable to humans and, therefore, their knowledge could be helpful for further utilizations and decisions by people. However, there is a tradeoff between the interpretability and accuracy of a fuzzy model [43].

Designing a fuzzy model from data can be decomposed into two main phases: 1) *rule generation* and 2) *system optimization* [15]. The rule generation can be further divided into two steps: a) *rule induction* and b) *rule-base optimization.* The former generates a basic system and a rule base, and the latter aims to refine the rule base. System optimization for complex multivariate systems involves selecting variables, reducing the rule base and optimizing the number of fuzzy sets.

In this chapter, a genetic fuzzy model-based fault diagnosis method and its implementation for the ship propulsion benchmark are presented. An optimization method: genetic algorithm (GA) is used for the fuzzy rule-base optimization. Figure 4.1 shows a GA-fuzzy model based FDI scheme, which is summarized in the following steps:

**Step 1-** Data collection. First of all, in order to get an accurate model, it is important to collect adequate data for the problem of interest. Numerical data are obtained from the system inputs and measurable outputs. The selected inputs and outputs are sampled and the generated data are preprocessed to remove noise.

**Step 2-** Model structure selection. This includes the selection of input and output variables from the collected data, and the structure of the model. Prior knowledge, combined with some statistical techniques, such as correlation analysis, can be used to determine the appropriate structure of input and output variables. Input, output, and product spaces are determined thereafter.

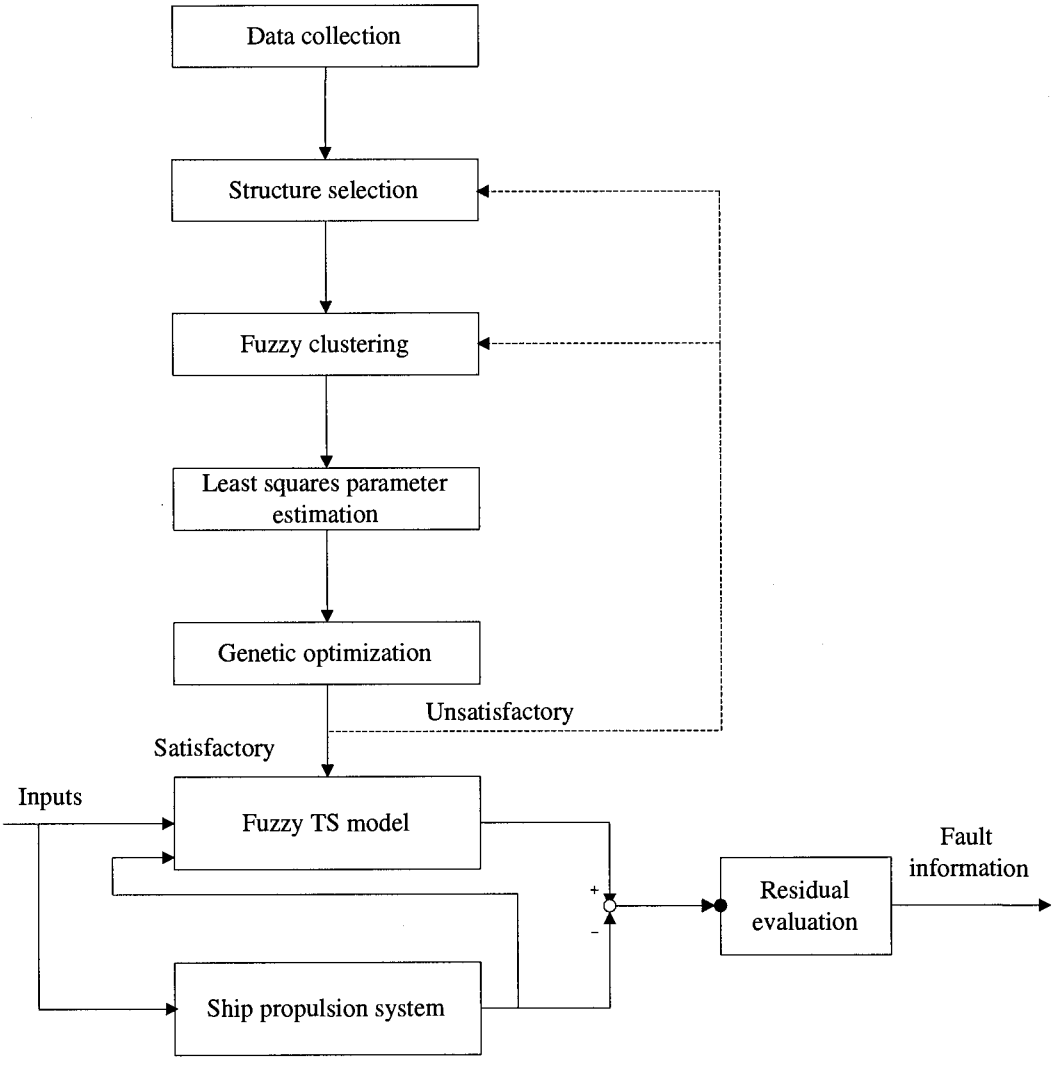


Figure 4.1: A GA fuzzy model-based FDI scheme

**Step 3-** Fuzzy clustering. Fuzzy clustering is a method used to decompose the product space into several partial fuzzy regions so that a rule base can be built which represents the input-output relationships. Each region in the product space will be represented by a rule in the fuzzy rule base. An appropriate number of clusters is adopted to determine how many partial fuzzy regions the product space is classified into. Cluster centers and the corresponding degrees of membership obtained through the clustering algorithm indicate the extent that a data set belongs to each cluster. The generated degrees of membership of different clusters are projected onto the input and output space. A predefined membership function is used and the antecedent parameters of each rule are approximated by means of projection.

**Step 4-** Least squares parameter estimation. After the antecedent parameters of the rule base have been decided, the output space is represented by either a set of fuzzy sets or a system of linear equations, depending on what type of fuzzy system has been chosen. In this implementation, as the outputs are approximated by linear equations of the inputs, the parameters of the equations are searched using the method of least squares estimation. Step 3 and step 4 together perform the task of rule induction.

**Step 5-** Genetic optimization. So far, the rule base and parameters have been obtained. An optimization method, namely the genetic algorithm, is now developed to refine the antecedent and consequent parameters so that model accuracy is improved. This step performs rule optimization. Since the nonlinear system being studied has only a few measured variables, system optimization is not necessary.

**Step 6-** Residual generation. The fuzzy model built by the previous steps is used to predict the behavior of the process under normal operating conditions. The difference between the output of fuzzy systems, i.e. estimated results, and the observed values are usually referred to as residuals in the fault diagnosis literature.

**Step 7-** Residual evaluation. The residuals generated from the previous stage are further evaluated and the FDI decision is then made. The system dynamics and other uncertainties are taken into consideration in the fault detection and isolation decision.

The principle of fuzzy identification is introduced in the first section of this chapter. The detailed concepts and principles involved in step 2 to 5 are then presented, followed by FDI simulation results of residual generation and evaluation with the ship propulsion benchmark.

## 4.1 Fuzzy Identification

*Fuzzy identification* usually refers to the process of deriving fuzzy models from data. The modelling framework considered in this thesis is based on rule-based fuzzy models, which describe relationships between variables by means of "if-then" rules. There are two types of rule-based fuzzy models: *Mamdani* and *Takagi-Sugeno* fuzzy models. In this section, the fundamental concepts of these two fuzzy models, including fuzzy sets, fuzzy rule bases and fuzzy logic systems, are introduced.

### 4.1.1 Fuzzy Sets and Fuzzy Logic Systems

A *fuzzy set* is in contrast to an ordinary set where a crisp decision is derived regarding whether or not a member belongs to the set. It is employed to represent the notion of fuzziness, which is often encountered in language description, in a mathematical way. A fuzzy set $U$ is characterized by a membership function $\mu : U \rightarrow [0, 1]$, where any member in $U$ is represented by the degree it belongs to $U$ : $\mu \in [0, 1]$. The values 0 and 1 indicate no membership and full membership respectively. Grades between 0 and 1 indicate that the data point has partial membership in a fuzzy set $U$.

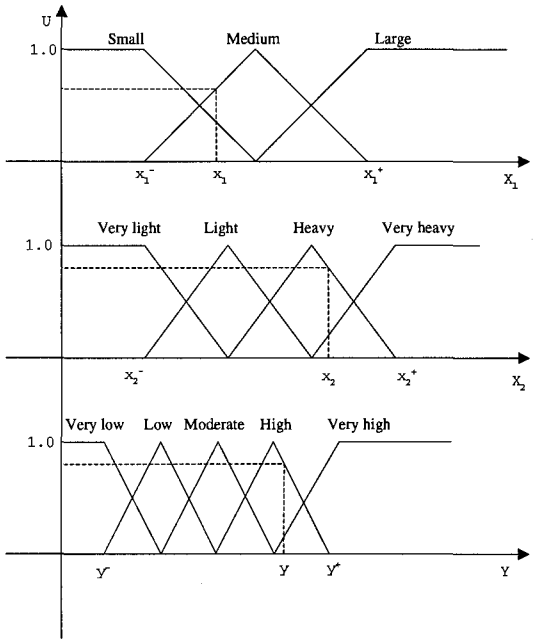


Figure 4.2: Grid partitioning fuzzy sets in a 2-input 1-output system

In Figure 4.2, the domain $[x_1^-, x_1^+]$ of input variable $x_1$ is partitioned into three fuzzy sets: *small*, *medium* and *large*. Similarly, variable $x_2$ has 4 fuzzy sets in domain $[x_2^-, x_2^+]$. The domain $[y^-, y^+]$ of output $y$ is decomposed into 5 fuzzy sets. These

fuzzy sets are derived by equally partitioning the corresponding domain into different numbers of subregions.

There could be an if-then rule describing the relationship between output $y$ and inputs $(x_1, x_2)$ such as:

$$\textbf{If } x_1 \textit{ is } \textit{Medium} \textbf{ and } x_2 \textit{ is } \textit{Heavy}, \textbf{ then } y \textit{ is } \textit{High} \tag{4.1}$$

where the "if" part is called the *antecedent* and the "then" part is called the *consequent.*

A basic *fuzzy logic system* consists of a fuzzy rule base and a *fuzzy inference system* (FIS) [43]. The fuzzy rule base includes a series of fuzzy if-then rules, and the FIS maps the input fuzzy sets to the output fuzzy sets according to the fuzzy rule base principles. Suppose the training data set contains $N$ data pairs. Each pair is composed of an $n$-dimensional input vector $\mathbf{x}$ and an $m$-dimensional output vector $\mathbf{y}$. So that $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ and $\mathbf{y} = [y_1, y_2, \ldots, y_m]^T$ . The fuzzy rule base of a Mamdani-type model is made of $R$ if-then rules described as:

$$\begin{aligned} &\textbf{If } x_1 \textit{ is } A_{i1} \textbf{ and } \ldots \textbf{ and } x_n \textit{ is } A_{in}, \\ &\textbf{then } y_1 \textit{ is } B_{i1} \textbf{ and } \ldots \textbf{ and } y_m \textit{ is } B_{im}, \quad i = 1, \ldots, R. \end{aligned} \tag{4.2}$$

where $A_{i1}, \ldots, A_{in}$ and $B_i, \ldots, B_{im}$ are antecedent and consequent fuzzy sets, respectively.

In a Mamdani fuzzy system with inputs and outputs being real-valued variables, there needs to be a fuzzifier and a defuzzifier to accomplish the transformation between real values and fuzzy sets, as depicted in Figure 4.3.

Each crisp datum $x_j$ is first fuzzified to a membership function value $\mu_{ij}(x_j)$ for fuzzy set $A_{ij}$ before the fuzzy rules can be applied. The outputs inferred from the inference system are defuzzified into crisp values by defuzzifiers, such as the center average defuzzifier and the maximum defuzzifier [43].

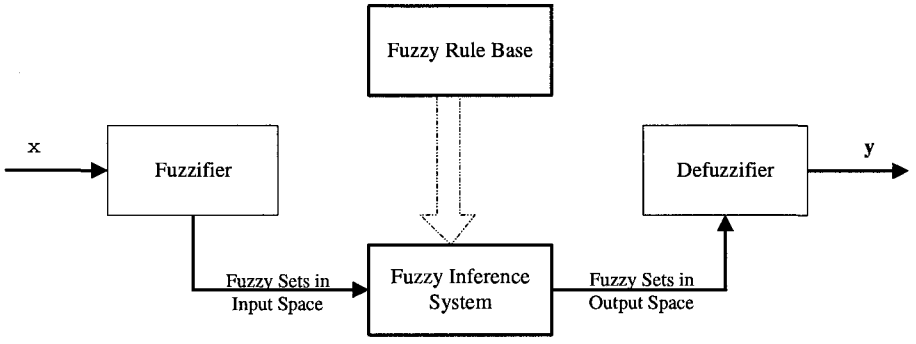


Figure 4.3: Basic configuration of a Mamdani fuzzy logic system with fuzzifier and defuzzifier

The FIS uses the rule base to evaluate the fuzzy inputs based on a number of ways to interpret the if-then rules in (4.2). Details are provided in [43].

### 4.1.2 Takagi-Sugeno Fuzzy Models

In contrast to the Mamdani model which derives the output from the if-then rules with both antecedents and consequents using fuzzy membership functions, the *Takagi-Sugeno* (TS) model [38] makes use of antecedent fuzzy sets and a consequent linear equation of the input variables to form the rule base, as illustrated in Equation 4.3. Takagi-Sugeno (TS) fuzzy model has been successfully applied to many practical problems. Figure 4.4 depicts the structure of a TS fuzzy system with real-valued input x.

$$\begin{aligned} &\textbf{If} \quad x_1 \quad is \quad A_{i1} \quad \textbf{and} \quad \ldots \quad \textbf{and} \quad x_n \quad is \quad A_{in}, \\ &\textbf{then} \quad g_i = p_{i1}x_1 + \ldots + p_{in}x_n + p_{i(n+1)}, \quad i = 1, \ldots, R. \end{aligned} \tag{4.3}$$

where $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ is the input vector, and $A_{i1}, \ldots, A_{in}$ are the antecedent fuzzy sets. The consequent part of the $i$th rule $g_i$ is a linear combination of the inputs and $p_{i1}, \ldots, p_{i(n+1)}$ are real-valued parameters.

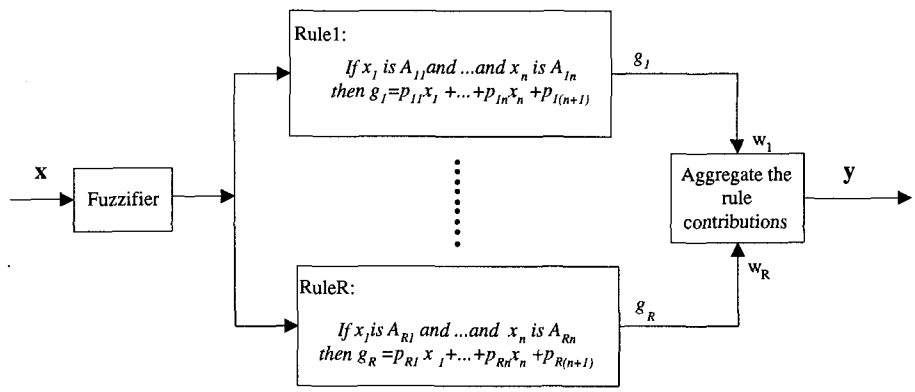


Figure 4.4: Basic configuration of a Takagi-Sugeno fuzzy logic system

The model output is computed by aggregating the individual rule contributions:

$$y = \frac{\sum_{i=1}^{R} w_i g_i}{\sum_{i=1}^{R} w_i} \tag{4.4}$$

where $w_i$ is the *degree of activation* of the $i$th rule calculated by using the product operator:

$$w_i = \prod_{j=1}^{n} \mu_{ij}(x_j), i = 1, 2, \ldots, R. \tag{4.5}$$

The TS model approximates a nonlinear system using the combination of several linear systems. The input space is decomposed into several regions. Then each region is represented by a linear equation. The consequent parameters can be optimized by means of nonlinear optimization methods.

There are two tasks for fuzzy modelling: *structure identification* and *parameter adjustment*. The former is to determine the partitioning of the input-output space, the antecedent and consequent parameters, and the initial position of membership functions. The latter deals with optimizing the parameters under the structure identified by the former step [30].

As previously mentioned, there is a tradeoff between readability and precision for a fuzzy model. Although more rules render greater precision, they are more difficult for users to interpret. The TS model is more suitable for situations where accuracy is considered more important than linguistic interpretability, while the Mamdani model has both antecedents and consequents as fuzzy sets and offers high interpretability.

## 4.2 Model Structure Selection

In step 2 of Figure 4.1, a common approach to modelling a dynamic system is to represent the identification of a dynamic system as a static regression problem. The followings are the three most commonly used model types [2].

- Nonlinear Finite Impulse Response (NFIR) model which uses the past input data to predict the future output data.

$$\hat{y}(k+1) = F(u(k), \ldots, u(k-n_u+1)) \tag{4.6}$$

  where $\hat{y}(k+1)$ represents the predicted output, and $y(k)$ and $u(k)$ are respectively the system output and input at time instance $k$. Parameter $n_u$ is the model order which should be chosen based on the particular problem. The function $F$ denotes the relationship which is mapped by the fuzzy model.

- Nonlinear Auto Regressive with eXogeneous Input (NARX) model which establishes a relation between the past input-output data and the predicted data.

$$\hat{y}(k+1) = F((y(k), y(k-1)\ldots, y(k-n_y+1), u(k), \ldots, u(k-n_u+1)) \tag{4.7}$$

  where $n_y$ denotes the order of past output data used as model input.

- Nonlinear Auto-Regressive Moving Average (NARMA) model which includes the past prediction errors $e(k) = \hat{y}(k) - y(k)$ into the equation.

$$\hat{y}(k+1) = F(y(k), y(k-1), \ldots, y(k-n_y+1), u(k), \ldots, u(k-n_u+1), e(k), \ldots, e(k-n_e)) \tag{4.8}$$

Choosing appropriate inputs, outputs, and the regression order requires prior knowledge of particular problems or statistical analysis of them. Unsuitably small order of the regression system results in inaccurate modelling, whereas an inappropriately large selection of the regression order causes modelling system "overfitting".

Once the model structure has been decided, the product space, which is composed of input space and output space, is readily obtained.

## 4.3 Fuzzy Clustering

### 4.3.1 Product Space Partitioning

For a fuzzy model to be constructed from numerical data, a set of fuzzy rules is generated from the desired input-output pairs and these fuzzy rules are used as a mapping from inputs to outputs. During the stage of rule induction, the product space is divided into fuzzy regions and fuzzy rules are consequently developed.

There are two kinds of methods for rule induction. The first one is called grid partitioning. Suppose the domain intervals are $[x^-, x^+]$ and $[y^-, y^+]$, for input $x$ and output $y$, respectively. The grid partitioning defines a number of fuzzy sets for each variable and these fuzzy sets are shared by all the rules. A common way to do grid partitioning is to divide each input variable domain into a given number of subsets and assign each subset a fuzzy membership function, as showed in Figure 4.2. This type

of partitioning does not necessarily involve any physical meaning and the membership functions and rules generated thereafter can be considerably many, which degrades the interpretability of the fuzzy model. Hence the usual procedure is to optimize the grid structure as well as the rule consequent parameters. Orthogonal transformation methods and similarity measures have been used to simplify the rule base to lower the system complexity and preserve accuracy as much as possible [35, 48].

Fuzzy clustering is a second unsupervised classification method used to deal with region partitioning and fuzzy membership function construction. The principle of product space clustering is to approximate a nonlinear regression problem by decomposing it into several local linear subproblems [2]. The data pairs are gathered into homogeneous groups, called clusters, by fuzzy clustering and a rule is associated with each group. More similarity can be found between members in the same cluster than between members belonging to different clusters. Fuzzy clustering allows the data to belong to several clusters simultaneously with different degrees of membership for each of these clusters. Different from the grid method, fuzzy sets are not shared by all the rules. Instead, each of them is used only in a particular rule.

### 4.3.2 Fuzzy C-means Clustering

A large family of fuzzy clustering methods is based on the optimization of the Fuzzy c-means function. The Fuzzy c-means clustering (FCM) was first introduced by Dunn in 1973 [12]. It partitions data into homogeneous groups and a rule is associated with each group. Suppose there are $N$ data pairs which need to be clustered into $c$ groups. The similarity in FCM clustering is measured by a distance norm from the $k$th data pair to the $i$th cluster:

$$D_{ikA}^2 = ||\mathbf{z}_k - \mathbf{v}_i||_{\mathbf{A}}^2 = (\mathbf{z}_k - \mathbf{v}_i)^T \mathbf{A} (\mathbf{z}_k - \mathbf{v}_i) \tag{4.9}$$

where $\mathbf{A}$ is a norm-inducing matrix, commonly chosen as the identity matrix so that Euclidean distance is achieved. Distances are calculated from the $kth$ data vector $\mathbf{z}_k$ in the product space to a prototype vector $\mathbf{v}_i$ of the $ith$ cluster. The prototype is sought and updated interactively in the partitioning process.

For a data matrix $\mathbf{Z}$, the FCM algorithm yields $\mathbf{U}$ and $\mathbf{V}$ which minimize the c-means objective function as shown in (4.10):

$$J(\mathbf{Z};\mathbf{U},\mathbf{V}) = \sum_{i=1}^{c}\sum_{k=1}^{N}(\mu_{ik})^m||\mathbf{z}_k - \mathbf{v}_i||^2_{\mathbf{A}} \tag{4.10}$$

where $m$ is a weighting exponent determining the fuzziness of the resulting clusters, usually set to 2. Matrix $\mathbf{U} = [\mu_{ik}]$ consists of membership functions, with the $(i,k)th$ element being the degree of membership from the $kth$ data pair to the $ith$ cluster under the probabilistic constraint:

$$\sum_{i=1}^{c}\mu_{ik} = 1 \quad \forall k = 1,\ldots,N \tag{4.11}$$

$$\mathbf{V} = [\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_c], \quad \mathbf{v}_i \in \mathbb{R}^n \tag{4.12}$$

where $\mathbf{V}$ is the matrix of the prototype vectors for $c$ clusters.

Suppose that the data set is $\mathbf{Z}$ and the number of clusters is $c$. The following are the procedures used in performing FCM clustering: firstly, initialize the partition matrix $\mathbf{U}$, then repeat the following steps with $l = 1,2,\ldots$ until $||\mathbf{U}^{(l)} - \mathbf{U}^{(l-1)}|| < \epsilon$ where $\epsilon$ is a small number used to indicate when sufficient accuracy is achieved.

1. Compute the cluster prototype(means):

$$\mathbf{v}_i^{(l)} = \frac{\sum_{k=1}^{N}(\mu_{ik}^{(l-1)})^m\mathbf{z}_k}{\sum_{k=1}^{N}(\mu_{ik}^{(l-1)})^m}, \quad 1 \le i \le c \tag{4.13}$$

2. Compute the distance:

$$D_{ikA}^2 = ||\mathbf{z}_k - \mathbf{v}_i||_{\mathbf{A}}^2 = (\mathbf{z}_k - \mathbf{v}_i)^T \mathbf{A}(\mathbf{z}_k - \mathbf{v}_i), \quad 1 \leq i \leq c, \quad 1 \leq k \leq N \quad (4.14)$$

3. Update the partition matrix:

   If $D_{ikA} > 0$ for $1 \leq i \leq c, \quad 1 \leq k \leq N$

$$\mu_{ik}^{(l)} = \frac{1}{\sum_{j=1}^{c} (D_{ikA}/D_{jkA})^{2/(m-1)}}, \quad (4.15)$$

   otherwise

   $\mu_{ik}^{(l)} = 0$ if $D_{ikA} > 0$,and $\mu_{ik}^{(l)} \in [0,1]$ with $\sum_{j=1}^{c} \mu_{ik}^{(l)} = 1$.

The FCM algorithm is suitable for clusters with comparable size and shape or when the clusters are well separated [15]. It imposes a spherical shape on the clusters, regardless of the actual data distribution.

### 4.3.3 Gustafson-Kessel Clustering

Gustafson-Kessel (GK) algorithm is an extension of the FCM algorithm which employs an adaptive distance norm [16]. In order to detect clusters of different geometrical shapes, it makes use of the fuzzy covariance matrix $F_i$ as shown below:

$$\mathbf{F}_i = \frac{\sum_{k=1}^{N} (\mu_{ik})^m (\mathbf{z}_k - \mathbf{v}_i)(\mathbf{z}_k - \mathbf{v}_i)^T}{\sum_{k=1}^{N} (\mu_{ik})^m} \quad 1 \leq i \leq c \quad (4.16)$$

The norm inducing matrix is then reformed as:

$$\mathbf{A}_i = [\rho_i det(\mathbf{F}_i)]^{\frac{1}{n}} \mathbf{F}_i^{-1}, \quad 1 \leq i \leq c \quad (4.17)$$

where $\rho_i$ denotes the cluster volume, usually fixed as 1. Similar to the steps of the FCM algorithm, the procedure in partitioning by GK algorithm is simply using the new $\mathbf{A}_i$ for cluster $i$ at each iteration of step 2 in section 4.3.2.

An advantage of the GK algorithm over the FCM is that it can divide the data set into groups of different shapes and orientations, geometrically, while the FCM algorithm can only be used to classify the data into the same types of shapes. It has been shown that the GK algorithm is especially suitable for the TS fuzzy model [2].

### 4.3.4 Projection and Membership Function Selection

After unsupervised clustering, the data pairs have been classified into $c$ groups in the product space. Membership functions are then generated by projecting the clusters onto each variable direction. For details, please refer to [2, 17].

Figure 4.5 illustrates how to project the classified clusters onto two-dimensional space, where $c_{11}, c_{12}, c_{13}$ and $c_{21}$,$c_{22}$, $c_{23}$ are the projection of cluster centers on each dimension. It is assumed that in the two dimensional product space [ $x$ $y$ ] three clusters are generated by fuzzy clustering. The FCM and the GK clustering both return not only the matrix of centers of the clusters, but also the degree of membership matrix $\mathbf{U}$. The $c \times N$ dimensional matrix $\mathbf{U}$ contains the grades of membership of each data pair to each cluster. The membership grades are plotted for each variable in dashed curves as shown in Figure 4.5. To construct the fuzzy inference system based on the fuzzy clustering, one has to transfer the membership grades to the fuzzy membership functions, and further build a fuzzy rule base on them. Triangular, trapezoidal, and Gaussian membership functions are the most widely used membership functions. The triangular membership function, shown in (4.18), was chosen in

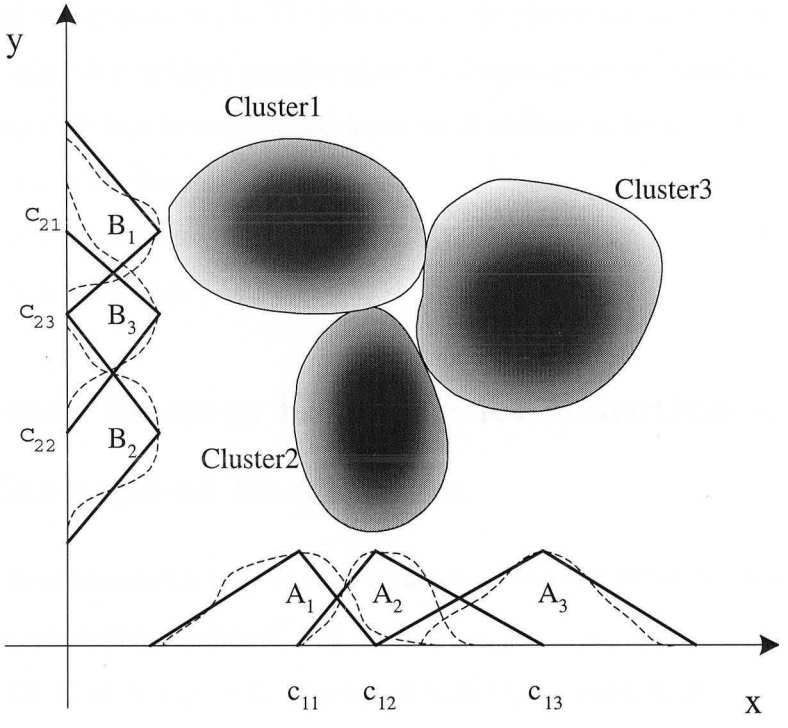


Figure 4.5: Project the clusters onto two dimensional space and use the triangular membership function to initialize the fuzzy system

this approach.

$$\mu(x; a, b, c) = max(0, min(\frac{x-a}{b-a}, \frac{c-x}{c-b})) \tag{4.18}$$

where $a, b, c$, are the three vertices of each triangular membership function. They are determined by projecting the membership grades from product space to each variable direction. For simplicity, the three cluster centers are taken as the three vertices having membership grade of 1. The leftmost or the rightmost vertices are simply set to a much bigger (or smaller) number than the boundary of the variable domains.

The fuzzy rule base is constructed from the 3 clusters in Figure 4.5 as follows:

Rule 1: if x is $A_1$ then y is $B_1$

Rule 2: if x is $A_2$ then y is $B_2$

Rule 3: if x is $A_3$ then y is $B_3$

## 4.4 Least Squares Estimate for Selection of Consequent Parameters

The consequent parameters $p_{iq}$ of each individual rule are obtained as the result of least square estimation. Define $\theta_i^T = [\ p_{i1}\ \ \ldots\ \ p_{iq}\ \ \ldots\ \ p_{in}\ \ p_{i(n+1)}\ ]$, let $\mathbf{X}_e$ denote the matrix $[\mathbf{X}\ \mathbf{1}]$ of $N \times (n+1)$ dimensions with input space $\mathbf{X}$ of:

$$\mathbf{X}_e = [\ \mathbf{X}\ \ \mathbf{1}\ ] = [\ \mathbf{x}_1\ \ \mathbf{x}_2\ \ \ldots\ \ \mathbf{x}_n\ \ \mathbf{1}\ ] \tag{4.19}$$

The $kth$ row of matrix $\mathbf{X}_e$ is $[\ x_{k1}\ \ x_{k2}\ \ \ldots\ \ x_{kn}\ \ 1\ ]$, and the output $\mathbf{y} = [\ y_1\ \ y_2\ \ \ldots\ \ y_N\ ]^T$. Let $\mathbf{W}_i$ denote:

$$\mathbf{W}_i = \begin{bmatrix} w_i(x_1) & 0 & \ldots & 0 \\ 0 & w_i(x_2) & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & w_i(x_N) \end{bmatrix} \tag{4.20}$$

The $N \times N$ matrix $\mathbf{W}_i$ has its $k$th diagonal element $w_i(x_k)$ as the degree of activation of the $kth$ row data of matrix $\mathbf{X}$ regarding $ith$ rule. The consequents of the $ith$ rule is the weighted least squares solution of $\mathbf{y} = \mathbf{X}_e\theta_i + \epsilon$ [2], which gives:

$$\theta_i = [\mathbf{X}_e^T\mathbf{W}_i\mathbf{X}_e]^{-1}\mathbf{X}_e^T\mathbf{W}_i\mathbf{y} \tag{4.21}$$

## 4.5 Genetic Algorithm Optimization

Since model accuracy is critical for a model-based FDI, and some of the system information might not be captured completely by initial fuzzy clustering and projection, an optimization approach is required to enhance the model performance. The Genetic Algorithm (GA) is an optimization technique which has received much attention in recent literature and it is appealing for powerful and parallel searching. The GA has found various applications in fields of pattern recognition and neural networks. In this section, a GA optimization method is applied on both antecedent and consequent parts of the fuzzy model to improve the model performance [34].

The Genetic Algorithm is a stochastic process performing parallel search over a complex space [31]. It maintains a population of $P(t)$, a set of individuals, for generation $t$ and uses a performance criterion *fitness* to evaluate each individual. Each individual represents a potential solution to the problem at hand. During the search for a global optimum, some individuals undergo stochastic transformations by means of *genetic operators* to form new individuals. New individuals, called child $O(t)$, are evaluated. Next, a new population is formed by selecting more fit individuals from the parent individuals and the child individuals. The GA recursively updates the optimization in each generation $t$ until it converges to the best individual. A general structure for the GA is as follows [13]:

**begin**

t← 0 ;

initialize P(t) ;

evaluate P(t) ;

**while** (*not termination condition*) **do**

**begin**

recombine P(t) to yield O(t) ;

evaluate O(t) ;

select P(t+1) from P(t) and O(t) ;

t ← $t+1$ ;

**end**

**end**

In GAs, an encoded individual set $x_1, x_2, \ldots, x_h$, which consists of $h$ parameters to be optimized is usually presented by a *chromosome*. In a binary coded GA, a chromosome containing $h$ segments of bit strings takes the form as follows:

$$\underbrace{10...0}_{x_1}\underbrace{101...11}_{x_2}\ ...\ \underbrace{11...00}_{x_h}$$

It also can be seen as an encoded possible solution to the problem of interest. A real-coded GA has chromosomes represented as decimal number strings and is considered in this thesis for its better performance on multidimensional and continuous problems over binary GAs [29]. For simplicity, examples in this section are shown in the format of binary strings. A set of chromosomes compose a population. In each generation, the initial population could be chosen as randomly generated or suboptimum chromosomes.

There are three genetic operators in GAs: *selection*, *crossover* and *mutation*.

1. Selection: the *selection* process copies chromosomes, known as parent chromosomes, and puts them into the mating pool for further genetic operations. The chance that a chromosome will be copied is based on the chromosome's fitness value. Good chromosomes are chosen and further operations work on them to locate possibly better solutions. Several methods are available for the selection operator such as Roulette wheel parent selection, linear selection and tournament selection.

2. Crossover: once the parents are chosen, the next operator is the *crossover* of the parents. Crossover, as the analog from the biological term, refers to the blending of the genetic information of parent chromosomes and the production of offsprings for the next generation. The GA decides whether or not the crossover should take place by a predefined parameter called *crossover probability.* When a randomly generated number is less than the crossover probability, the GA decides not to perform crossover. Otherwise, crossover will take place and a random splicing point is chosen in a string. Thus, two new strings are created by swapping all the characters from the splicing position to the end of the string. For example, if the splicing number is randomly generated as 5, the crossover between:

   001011011010 and

   011101101100

   will result in two new strings:

   001011101100 and

   011101011010

   These child strings are then placed in the child population.

3. Mutation: even though the selection and crossover operations have enlarged the

searching space, the GA employs another operator called *mutation* to further enlarge the searching space and keep the GA away from local optima. A *mutation probability* determines the frequency at which mutations occur. For example, a mutation of the string 001011101100 could result in a new chromosome of 001011111100, as the eighth bit is flipped.

The selection and crossover operations produce a large number of different individuals. A mutation, even occurring with a very small probability, helps prevent the population from stagnating. The power of a GA comes from the fact that it contains a rich set of individuals of great diversity.

The iteration repeats again with selection for the next generation. This iterative process continues until all user-specified criteria are met (for example, one hundred generations, or a string is found to have a fitness exceeding a certain threshold).

## 4.6 Simulation Results and Discussion

### 4.6.1 Fuzzy Model of The Ship Propulsion Benchmark

As noted in Chapter 2, the input signals for the benchmark are the shaft speed set point $n_{ref}$ and the propeller pitch $\theta_{ref}$. Measured outputs are the diesel engine shaft speed $n_{mf}$, the fuel index $Y_m$, the propeller pitch position $\theta_{mf}$, and the ship speed $U_m$. As in Chapter 3, three faults are considered in this chapter: velocity measurement "too high" $\Delta n_{high}$, velocity measurement "too low" $\Delta n_{low}$, gain fault $\Delta k_y$.

According to Figure 4.1, to employ the GA-fuzzy model based FDI on this particular problem the first step is the data collection. To achieve that, simulation of the ship propulsion model run for 3500 seconds and a set of training data was collected and stored in a Matlab data file. Figure 4.6 presents the output signal.

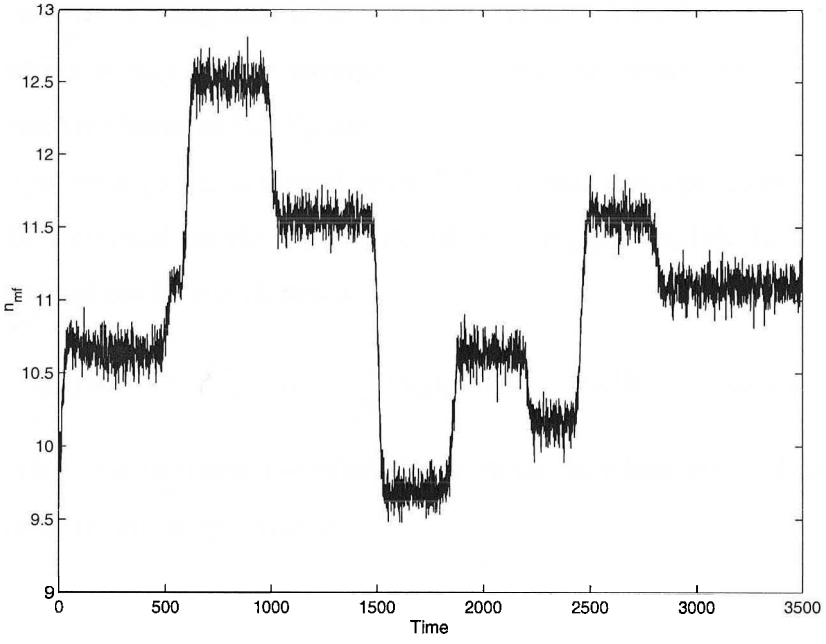


Figure 4.6: Output of the training data

The output signal is preprocessed by a low-pass filter to remove noise. The next step for building the fuzzy model is to choose an appropriate model structure, i.e. determine input and output signals for this problem. Generally speaking, statistical properties such as covariance and correlation can be used to determine the number of useful signals for modelling. In this particular problem, an analysis of structure models of the ship propulsion system leads to identifying several subsystems carrying possible redundant information [21]. Further analysis of the torque sub-system and knowledge about the intended use of the model suggested the following model:

$$n(k+1) = f(\theta(k-1), \theta(k), U(k), U(k-1), Y(k-1), Y(k)) \tag{4.22}$$

Since the dynamics of the ship speed are much slower than the shaft speed, for the purpose of the analysis in this paper, including $U_m(k)$ in the input variables is

sufficient, instead of using both $U(k)$ and $U(k-1)$ [21]. Therefore, the fuzzy model has the output of $n_{mf}$ for the purpose of detecting and isolating the three faults. Input signals are chosen as $U_m, Y_m$ and $\theta_{mf}$.

Consistent with (4.22), a second order NFIR model is adopted and $n_{mf}(k+1)$ needs to be estimated by the inputs: $\theta_{mf}(k-1)$, $\theta_{mf}(k)$, $U_m(k)$, $Y_m(k-1)$, and $Y_m(k)$. The final model was chosen as:

$$\hat{n}_{mf}(k+1) = F(\theta_{mf}(k-1), \theta_{mf}(k), U_m(k), Y_m(k-1), Y_m(k)) \tag{4.23}$$

where $\hat{n}_{mf}(k+1)$ is the predicted value of $n_{mf}$ at the time instance $k$. Therefore, the input space $\mathbf{X}$ in our fuzzy model is:

$$\mathbf{X} = [\ \theta_{mf}(k-1)\ \ \theta_{mf}(k)\ \ U_m(k)\ \ Y_m(k-1)\ \ Y_m(k)\ ] \tag{4.24}$$

The output space $\mathbf{y}$ consists of $n_{mf}$ only. The product space $\mathbf{Z}$ is then built so that it can be decomposed by fuzzy clustering:

$$\mathbf{Z} = [\ \mathbf{X}\ \ \mathbf{y}\ ] = [\ \theta_{mf}(k-1)\ \ \theta_{mf}(k)\ \ U_m(k)\ \ Y_m(k-1)\ \ Y_m(k)\ \ n_{mf}(k+1)] \tag{4.25}$$

The following step used the FCM and GK algorithms to classify the product space into several clusters. In the preceding discussion in Section 4.3.2, prior knowledge of the number of clusters (and rules) $c$ is assumed. Regarding this FDI problem, some trial and error has been performed to obtain an appropriate number $c$. To avoid a complex structure for the fuzzy system, the cluster numbers should be limited. Nevertheless, choosing too small a number will not result in a model with satisfactory accuracy. Given the system model structure as previously defined, numbers $3, 4$ and $5$ were utilized separately as the cluster numbers and the corresponding fuzzy systems were built.

The antecedent parameters of the TS model rules are approximated by the projection of the fuzzy clusters onto input and output directions as illustrated in Section 4.3.4. Triangular membership functions are chosen for each of the generated fuzzy sets (i.e. clusters), as in Figure 4.5. The consequent values of the TS model are then finalized by the maximization of the reciprocal of mean-square error:

$$P = \frac{1}{J} \tag{4.26}$$

where

$$J = \frac{1}{N}\sum_{k=3}^{N}(\hat{n}_{mf}(k) - n_{mf}(k))^2 \tag{4.27}$$

where the $k$ starts from 3 due to the fact that the second order regression system is adopted. The choice of performance measure in (4.26) is consistent with the fitness criterion selected for the genetic algorithm discussed later in this section.

The performance of the inferred fuzzy system with $3, 4$, and 5 clusters is summarized in Table 4.1. According to the evaluation criteria, an accurate fuzzy model will present a large value of $P$. It is obvious that 3 clusters is the best solution among these three choices and the GK clustering considerably outperforms the FCM clustering as expected.

| Number of clusters | P (by FCM algorithm) | P (by GK algorithm) |
|---|---|---|
| 3 | $1.2252 \times 10^3$ | $5.3690 \times 10^3$ |
| 4 | 603.0551 | $1.3395 \times 10^3$ |
| 5 | 49.9896 | 54.5786 |

Table 4.1: Performance value P using FCM and GK algorithms with 3,4, and 5 clusters

Since the product space of the problem consists of 5 input variables, and the number of fuzzy clusters has been chosen as 3, the projection generates 3 triangular

membership functions for each variable. For the $jth$ variable in the product space, a series of membership parameters and further fuzzy sets are derived: parameters $(a_{1j}, b_{1j}, c_{1j})$ represent fuzzy set $A_{1j}$, parameters $(a_{2j}, b_{2j}, c_{2j})$ represent fuzzy set $A_{2j}$, and $(a_{3j}, b_{3j}, c_{3j})$ represent fuzzy set $A_{3j}$. The consequent parameters $\theta_i$ with the same format in Section 4.4 for each rule are determined by least-square estimation.

So far, the antecedent and consequent parameters of the TS fuzzy system have been derived. It is then possible to build the fuzzy rule base which consists of 3 fuzzy rules, each rule explaining the extent to which a particular data pair belongs to the corresponding cluster. The fuzzy system has a rule base of the following format:

$$\begin{aligned} &\textbf{If} \quad x_1 \quad is \quad A_{i1} \quad \textbf{and} \ldots \textbf{and} \quad x_5 \quad is \quad A_{i5}, \\ &\textbf{then} \quad g_i = p_{i1}x_1 + \ldots + p_{i5}x_5 + p_{i6}, \quad i = 1, 2, 3. \end{aligned} \tag{4.28}$$

As shown in Figure 4.1, a real-coded GA is implemented on the generated fuzzy model to enhance the system accuracy. In this particular problem, the initial chromosome is built by the fuzzy rule antecedent and consequent parameters derived by FCM (or GK) and the least squares algorithm. In the first generation, the population consists of the initial chromosome and other randomly generated chromosomes. Their parameters are uniformly distributed in the neighboring area of the parameters in the initial chromosome. For instance, a chromosome $c_l$ is presented as:

$$c_l = (a_{11}, b_{11}, c_{11}, \ldots, a_{ij}, b_{ij}, c_{ij}, \ldots, a_{35}, b_{35}, c_{35}, \theta_1, \theta_2, \theta_3) \tag{4.29}$$

where $a_{ij}, b_{ij}, c_{ij}$ are the triangular membership function parameters of the $ith$ rule and the $j$th variable. The consequent parameter vector $\theta_i$ contains the consequent parameters [ $p_{i1}$ $p_{i2}$ $p_{i3}$ $p_{i4}$ $p_{i5}$ $p_{i6}$ ] of the $ith$ rule.

The fitness criterion of a good chromosome is a large $P$ calculated as in (4.26) and (4.27). Mean-square-error $J$ and the fitness criterion $P$ are calculated for each chromosome in a population.

The selection function used is tournament selection. In tournament selection a number, *Tour*, of chromosomes is chosen randomly from the population and the best chromosome from this group is selected as parent. The parameter *Tour* has the value of 2 in this simulation.

A simple arithmetic crossover operator and a uniform mutation operators are chosen, and the crossover and mutation probabilities are set to be 0.95 and 0.05 respectively.

Two constraints are considered to keep the physical meaning of the triangular memberships:

- Since the three number group of $(a, b, c)$ represents the triangular membership function, $a < b < c$ should be ensured for each fuzzy set. Each new population should be constrained to meet this requirement.
- To avoid the optimization points from drifting too far away from the original points, the scope limit is set to be 0.35; i.e, the searching scope is about one third of the fuzzy set length, centered around the original points.

The original and new membership functions have been generated after 10000 generations as shown in Figure 4.7 and Figure 4.8. The system performance has been improved to $P = 6.6317e{+}005$. The re-organization and simplification of the updated membership functions can improve the interpretability of the fuzzy model. However, it is beyond the scope of this thesis.

Another set of data was collected as checking data for this fuzzy model. The output signal and estimated errors for this checking data are shown in Figure 4.9. Part (a) shows the system output signal $n_{mf}$. It can be seen from part (b) of this figure that the estimated error: $e = \hat{n}_{mf} - n_{mf}$ is quite small during most of the simulation time. The resulting $P$ value for the checking data is $2.7810e + 005$. There exist some time instances when the errors can not be ignored. This is because the

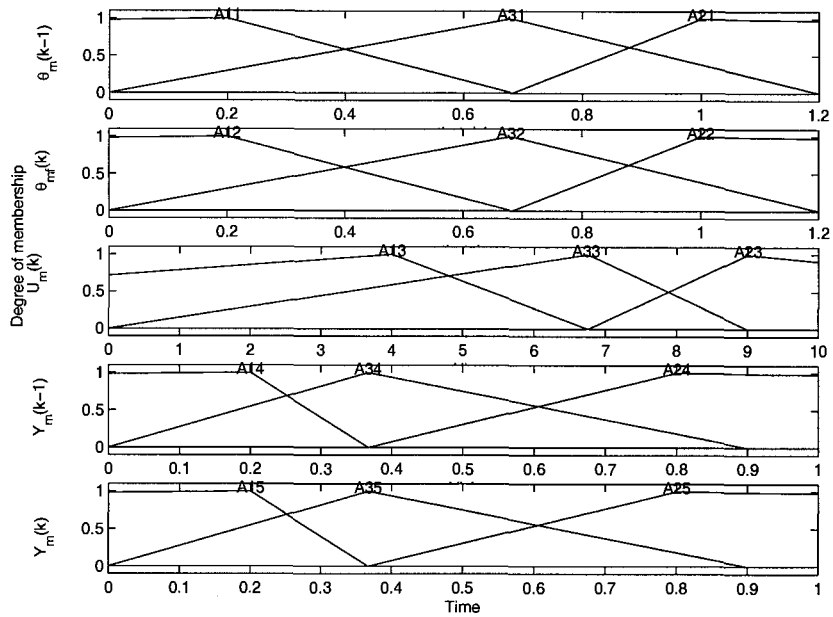


Figure 4.7: Fuzzy sets and membership functions before the GA

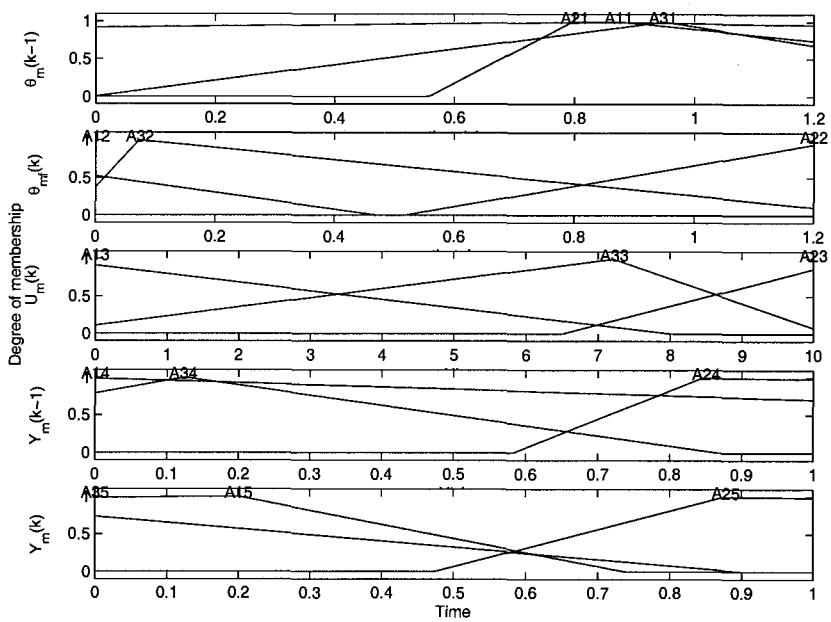


Figure 4.8: Fuzzy sets and membership functions after execution of GA

model can not capture the system's dynamics immediately. With the fault evaluation scheme presented in the next section, a satisfactory FDI method is performed without considering this little handicap.

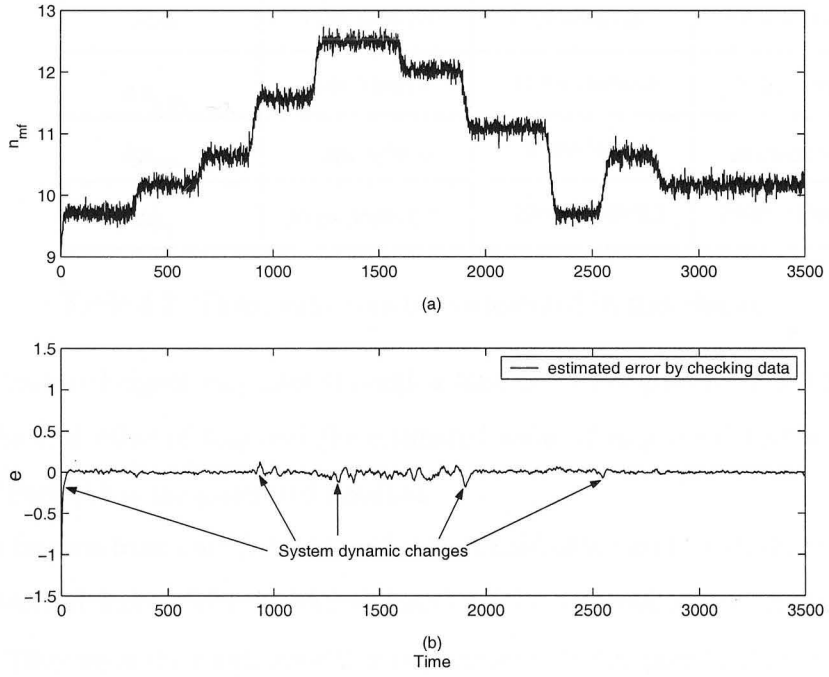


Figure 4.9: (a)Output signal $n_{mf}$ of checking data, (b)Estimated error of checking data

### 4.6.2 Residual Generation and Evaluation

After the model has been built, the residuals are generated by feeding input data into the fuzzy model, monitoring the corresponding estimated outputs from the model and comparing the model outputs with the actual measured values. The errors are considered as the residuals. Once the residual has crossed over a certain threshold, an alarm will be triggered, indicating a fault has occurred.

There are three fault scenarios and problems presented in this section. The corresponding information is listed in Table 4.2, in which fault scenario 1 is consistent with [20].

| Fault | Fault scenario 1 | Fault scenario 2 | Fault scenario 3 |
|---|---|---|---|
| $\Delta n_{high}$ | 680s-710s/13 | 1120s-1140s/16 | 1020s-1060s/15 |
| $\Delta n_{low}$ | 2640s-2670s/5 | 1620s-1660s/7 | 2110s-2150s/5 |
| $\Delta k_y$ | 3000s-3500s/0.2 | 2780s-3300s/0.2 | 2900s-3300s/0.2 |

Table 4.2: Three fault scenarios presented in this chapter

The measured signal $n_{mf}$ goes through a first-order low-pass filter to remove the noise. The real value of $n_{mf}$ and the estimated value of $n_{mf}$ are depicted in Figure 4.10 (a). Part (b) is the generated residual.

It can be seen from part (b) that with a threshold of 0.4 and $-0.4$, the three faults can be detected successfully. The three faults can be detected at 681, 2641, and 3002 seconds. They meet the quick detection requirements in Chapter 2. Figure 4.10 show changes in system dynamics also increase the residuals. However, compared to the residual magnitude when a fault occurs, these increase are small. Faults are readily distinguished from changes in system dynamics using a simple threshold criteria.

It can be observed that the magnitude of the residual caused by the $\Delta n_{high}$ fault is negative and falls sharply when crossing the threshold line, whereas the magnitude of the residuals caused by the $\Delta n_{low}$ fault is positive and rises sharply when crossing the threshold line. In addition, the residual caused by the $\Delta k_y$ fault is positive and has relatively small but constant magnitude. A decision scheme is adopted that if the magnitude of residual is bigger than 0.4 and has a change rate over 2, it is assumed to be caused by $\Delta n_{high}$. In contrast, if the magnitude of residuals is smaller than $-0.4$

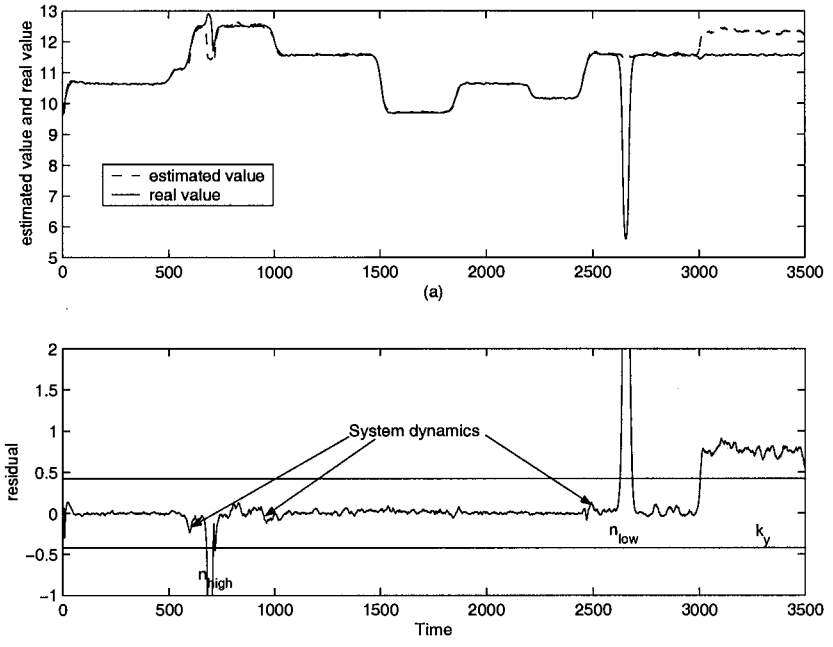


Figure 4.10: (a)Estimated value and real value of $n_{mf}$ (b) Generated residual in fault scenario 1

and has a change rate less than $-2$, it is assumed to be caused by $\Delta n_{low}$. Otherwise, when it is bigger than 0.4 and has a comparatively small change rate, it is considered a fault caused by $\Delta k_y$. Based on observations on the residual and the rate of residual changes in the presence of the three faults, fault isolation can be implemented by a fuzzy inference system with following rules:

1. **If** residual is *bigger than thresh* and resrate is *large* **then** the fault is $\Delta n_{low}$;
2. **If** residual is *smaller than -thresh* and resrate is *neglarge* **then** the fault is $\Delta n_{high}$;
3. **If** residual is *bigger than thresh* and resrate is *small* **then** the fault is $\Delta k_y$;

where resrate demonstrates $(residual(k) - residual(k-1))/\Delta T$ and $thresh = 0.4$. The fuzzy sets are depicted in Figure 4.11.

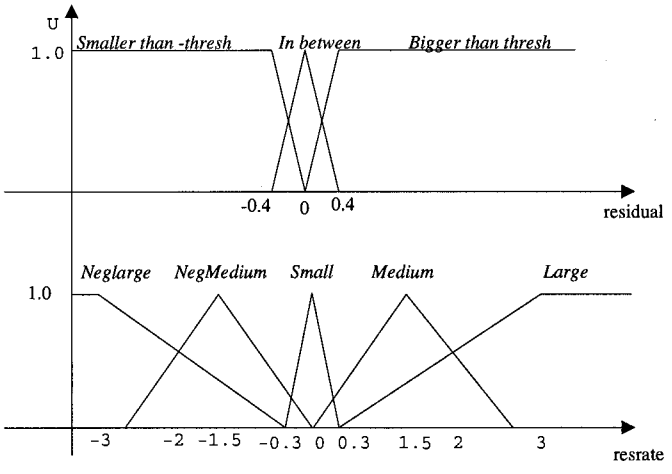


Figure 4.11: Fuzzy sets of the fuzzy interference system for fault isolation

Figure 4.12, 4.13 present the residuals in fault scenario 2 and 3, respectively. In Figure 4.13, it is noticed that the residual right after the $\Delta n_{high}$ fault exceeds the threshold. This may lead to false alarms. As the chance that a fault happens right

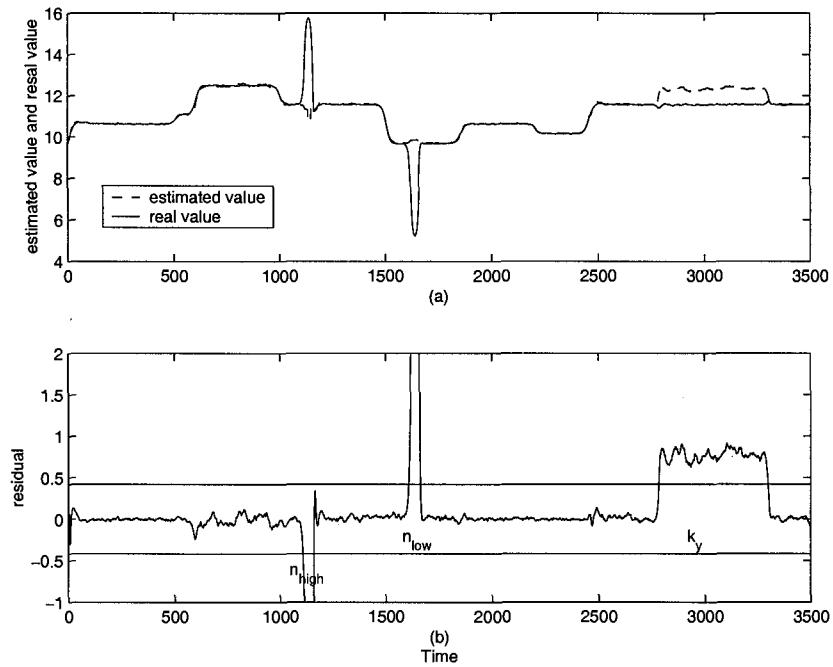


Figure 4.12: (a)Estimated value and real value of filtered $n_{mf}$ (b) Generated residual in fault scenario 2

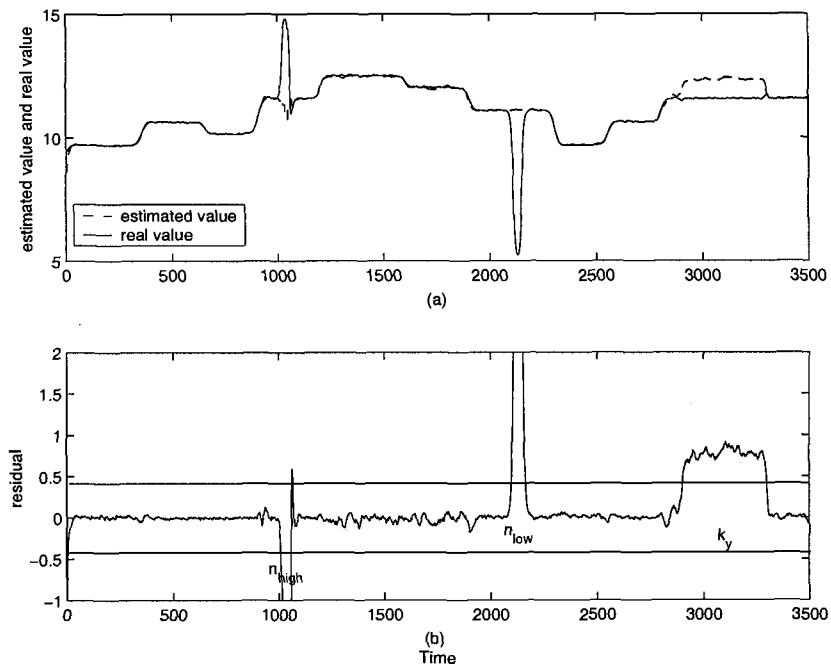


Figure 4.13: (a)Estimated value and real value of filtered $n_{mf}$ (b) Generated residual in fault scenario 3

after another fault is detected is very low, this possible alarm could be avoided by monitoring several more samples of the signal to make the final decisions.

## 4.7 Chapter Summary

The results in this chapter show that the GA fuzzy model-based FDI is able to detect and isolate three faults in the ship propulsion benchmark. The required time limits have been achieved and the three faults can be isolated successively. This demonstrates that the represented method of building the fuzzy model based on numerical data and the application to FDI renders satisfactory results. This method can be generalized to other FDI problems. Below are some considerations for the ship propulsion benchmark problem:

1. The training data used in building the fuzzy system contains 3500 input-output data pairs. Including more training data will take more computing resources and may result in the "overfitting" problem.

2. The training and checking data are preprocessed by a first order low pass filter with the transfer function of $H(s) = \frac{0.05}{s+0.05}$ to remove the influence of high frequency noise. Any monitored real-time data should be preprocessed by a similar LPF before the residuals are generated.

3. It has also been noted that the dynamics in this system have not been completely captured by the GA-fuzzy model. Fortunately, the threshold selected in this chapter is able to tolerate the errors caused by system dynamic changes. An alternative solution could be an adaptive threshold turning smaller in steady states and larger in dynamic change points.

4. Only one fuzzy model was built, thus only one residual is generated and taken

into consideration. This model functions well to detect and isolate the three faults. The work suggests that the task of detecting other three faults may be achieved using similar models. More fuzzy models of additional signals containing information about other possible faults can be constructed. For example, to detect faults $\triangle\theta_{high}$ and $\triangle\theta_{low}$, studies on signal $\theta_{mf}$ and relative modelling would be necessary. Using fuzzy modelling to detect incipient faults is left as promising future work. Since this thesis is intended to reveal properties of various FDI approaches, we will move on to a new approach in the next chapter.

# Chapter 5

# Wavelet Signal Processing Combined with Model-based Method

As discussed in Chapter 1, signal processing-based FDI methods have been investigated for linear and nonlinear systems. Because they are comparatively simple, these methods are suitable for many systems where accurate models are not available or are difficult to obtain. Wavelet analysis has perhaps been one of the most exciting developments in the last decades, bringing together researchers from several different fields such as image compression, signal processing, speech recognition, and so on. In the field of fault detection and isolation, wavelet based approaches have also been studied and interesting results have been derived [23, 50]. The wavelet transform accurately localizes the characteristics of a signal in both the time and frequency domains, where the multiscale representation of the signal helps identify the occurring instants of abrupt faults. The differences in the distribution of the signal energy on decomposed wavelet scales of the signal before and after the occurring instants of

faults are used to isolate various sensor faults [50]. In Kim's paper[23], the wavelet decompositions are used for nonstationary signal feature extraction in the process of a NN model based fault diagnosis. The wavelet neural network is introduced as an extension of a regular wavelet basis. It is a consistent function estimator having universal and $L^2$ approximation properties [51, 52]. Wang [44] trained a dynamic wavelet neural network, which carries out fault prognosis tasks and predicts future failures. Various research shows that wavelet analysis is a flexible and effective method to handle fault detection involving nonstationary signal information.

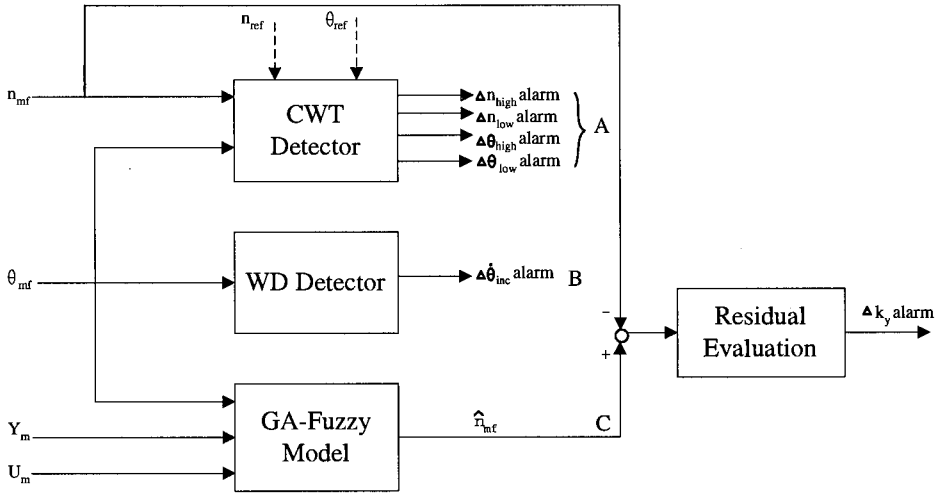


Figure 5.1: Scheme of the combined FDI method on the ship benchmark

In this chapter, the principle of wavelet analysis and its applications to FDI for the ship propulsion system are presented. There are five faults that can be successfully diagnosed by continuous or discrete wavelet time transform or decomposition, including an incipient fault. To detect and isolate all the six possible faults in this benchmark, a fuzzy model is included as an effective FDI approach for the fault which can not be detected by wavelet transform methods. The FDI scheme of wavelet transform

combined with model-based method is adopted as illustrated in Figure 5.1. The three detector outputs are A, B, and C. The detection and isolation logic is listed in the format of truth table in Table 5.1, where 1 denotes a triggered alarm, 0 denotes no alarm, and × denotes "don't care". It can been seen from this truth table that: 1) the outputs of CWT detector perform fault detection of four abrupt sensor faults, 2) the output of DWT detector indicates the incipient sensor fault, 3) when the CWT detector does not trigger an alarm and the residual from the GA-fuzzy model is bigger than the threshold, a gain fault is considered to have occurred.

| A | B | C | Fault diagnosis |
|---|---|---|---|
| 1 | × | × | $\Delta n_{high}$, $\Delta n_{low}$, $\Delta\theta_{high}$ or $\Delta\theta_{low}$ |
| × | 1 | × | $\Delta\dot{\theta}_{inc}$ |
| 0 | × | 1 | $\Delta k_y$ |

Table 5.1: Fault detection and isolation logic for combined FDI system

This scheme will be elucidated in greater detail in the following sections. First, the principle of wavelet transform is introduced in the next section.

## 5.1 The Principle of Wavelet Analysis

### 5.1.1 Continuous Time Wavelet Transform

Wavelet analysis is a new and promising set of time-scale (time-frequency) domain tools and techniques applicable to a wide range of signals. This section provides an introduction to wavelet transform theory. The beginning of the wavelet transform can be traced back to a paper by Grossman and Morlet [14]. They started to model a certain signal by a combination of translations and dilations of a simple, oscillatory

function of finite duration called a *wavelet.* The work is referred to as the *continuous wavelet transform* (CWT). There are other wavelet analysis techniques which have been developed based on CWT. Wavelet analysis allows the use of long time intervals where more precise low-frequency information is desired, and shorter intervals where high-frequency information is desired.

Let $L^2(R)$ denote the Hilbert space of measurable, square-intergrable one-dimensional functions. A function $f(t) \in L^2(R)$ satisfies:

$$\int_{-\infty}^{+\infty} |f(t)|^2 dt < \infty \tag{5.1}$$

The continuous-time wavelet transform of $f(t)$ with respect to a wavelet $\psi(t)$ is defined as the sum over all time of the signal multiplied by scaled, shifted versions of function $\psi(t)$ [14]:

$$Wf(s,u) = \frac{1}{\sqrt{s}} \int_{-\infty}^{+\infty} f(t)\psi^* \left( \frac{t-u}{s} \right) dt \tag{5.2}$$

where $\psi^*(t)$ denotes the complex conjugation of the *mother wavelet*, which is a real or complex-valued continuous-time function with the following two properties:

- The function $\psi(t)$ is a function with zero average, i.e:

$$\int_{-\infty}^{+\infty} \psi(t) dt = 0 \tag{5.3}$$

- The function $\psi(t)$ is a square-integrable function, i.e. it satisfies Equation 5.1. A mother wavelet is a waveform where most energy is confined to a finite duration. There are an infinite number of functions which qualify as mother wavelets.

The variable $s$ is referred to as the *scale* or *dilation variable*, demonstrating stretching or compressing of the mother wavelet. The variable $u$ is called *time shift* or *translation*, referring the delaying or hastening of the signal onset. The function $\psi^*(\frac{t-u}{s})$ is

then the dilation and translation of the mother wavelet function $\psi(t)$. Thus wavelet analysis produces a time-scale view of a signal.

Let $\bar{\psi}_s(t) = \frac{1}{\sqrt{s}}\psi\left(-\frac{t}{s}\right)$, then the continuous wavelet transform can also be written as a convolution product:

$$Wf(s,u) = f \star \bar{\psi}_s(u) \tag{5.4}$$

where $\star$ denotes convolution.

The result of applying CWT to a signal is a wavelet coefficient vector, which is a function of scale and translation. Each coefficient represents how closely correlated a scaled wavelet is with a portion of the signal which is determined by the translation. Continuous wavelet transform coefficients are precisely the time-scale view of the signal. On the other hand, the CWT also offers insight into both time and frequency domain properties of the signal. Since higher scales correspond to the most "stretched" wavelets, the more stretched the wavelet, the longer the portion of the signal with which it is compared, and thus the coarser the signal features measured by the wavelet coefficients. These coarser features, called *approximations*, providing basic shapes and properties of the original signal, correspond to low frequency components, whereas the low scale components capture the high frequency information and are called *details* in wavelet literature.

### 5.1.2 Discrete Time Wavelet Decomposition

Calculating continuous wavelet coefficients at every single scale is a fair amount of work and generates a lot of data. The discrete wavelet transform (DWT) is based on the wavelet analysis at particular scales and translations that are powers of two, such as $2, 4, 8, 16$, and so on. It is much more efficient and just as accurate as the CWT.

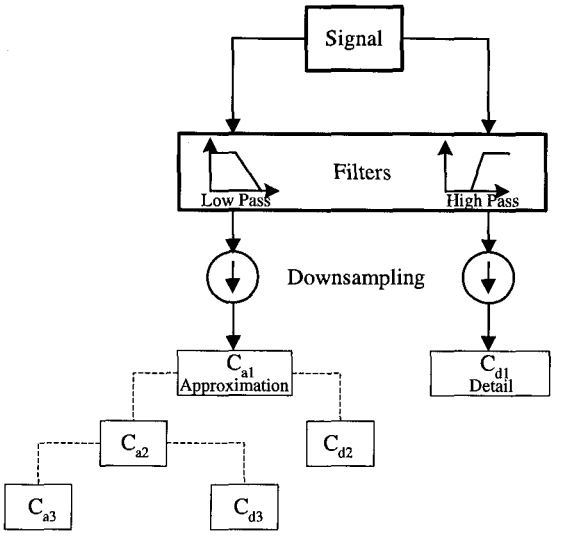


Figure 5.2: Scheme of multiple stage wavelet decomposition

The approximations of signals provide basic trends and characteristics of the original signal, whereas the details provide the flavor of the signal. The DWT reveals the approximation and detail properties of a signal at different stages (scales) through the process shown in Figure 5.2. The original signal passes through two complementary filters, whose characteristics are determined by the selected wavelet, and then emerges as two signals. The purpose of followed downsampling is to keep the generated $C_a$ and $C_d$ each half as long as the original signal, so as to avoid winding up with twice as much data. The outcome of the discrete wavelet transform is an approximation coefficient vector $C_a$ and a detail coefficient vector $C_d$, respectively representing the approximations and details of the analyzed signals. This process is also called wavelet decomposition. If the decomposition is repeated on the approximations in each stage, then the multiple stage DWT will break down one signal into many successively lower resolution components, as shown in Figure 5.2. At each stage, the approximation

coefficient $C_a$ represents the signal trend and the detail coefficient $C_d$ includes the information of noise or nuance. Because of the downsampling performed in each stage, the length of $C_a$ and $C_d$ is shortened in exponents of 2 for each stage. For example, $C_{a6}$ is cut off to $1/2^6$ to its original length.

The original signal can be reconstructed from the approximations and details by the process of applying inverse discrete wavelet transform (IDWT).

## 5.2 Wavelet Analysis Combined With Model-Based FDI

### 5.2.1 The Modulus Maximum and Singularity

In the literature of the wavelet transform, the term "*modulus maximum*" refers to any point $(s_0, u_0)$ such that $|Wf(s_0, u)|$ is locally maximum at $u = u_0$. This implies:

$$\frac{\partial Wf(s_0, u_0)}{\partial u} = 0 \tag{5.5}$$

This local maximum should be a strict local maximum in either the right or the left neighborhood of $u_0$, to avoid any local maximum when $|Wf(s_0, u)|$ is constant.

If the mother wavelet $\psi(t)$ is chosen as the first-order derivative of the Gaussian low pass function $\zeta(t)$, let $\bar{\zeta}_s(t) = \frac{1}{\sqrt{s}}\zeta\left(\frac{-t}{s}\right)$, then $\bar{\psi}_s(t) = s\frac{d\bar{\zeta}_s(t)}{dt}$ and:

$$Wf(s, u) = f(u) \star s\frac{d\bar{\zeta}_s(u)}{du} = s\frac{d}{du}[f(u) \star \bar{\zeta}_s(u)] \tag{5.6}$$

It is shown in Figure 5.3 that the CWT of $f(t)$ is equivalent to the process of low-pass filtering $f(t)$ first, and then taking the first-order derivative of the resulting signal. The wavelet modulus maxima are the maxima of the first-order derivative of $f$ smoothed by $\bar{\zeta}_s$. The CWT of the signal $f(t)$ has modulus maxima around the singular

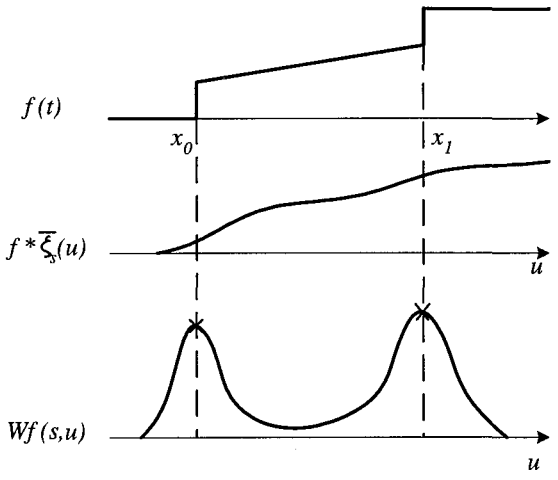


Figure 5.3: Original signal f(t), smoothed signal $f \star \bar{\zeta}_s(u)$, and wavelet transform $Wf(s,u)$

points. This can be used in distinguishing between the signal singular points and noise [28]. It has been proven that the mean value of continuous wavelet coefficient of a stationary random signal is zero. In addition, when the scale variable $s$ increases, the variance of the coefficient converges to zero [47]. As an instance of stationary random signals, stationary noise has the characteristic that the wavelet transform modulus maxima will quickly decay with increasing scale. Since the monitored signals usually consist of deterministic signals and noise, the wavelet transform of monitored signals is the sum of the wavelet transform of these two parts. The modulus maxima which correspond to singularity points of deterministic signals will increase or slowly decay, whereas the modulus maxima which correspond to noise will quickly decay.

The singularities and edges of a signal are detected by checking the wavelet transform modulus maxima at various scales. Thus, abrupt sensor faults occurring on a signal are detectable by finding the abscissa where the wavelet modulus maxima converge at large scales [27]. The wavelet transform has proven to be particularly useful

in detecting abrupt sensor faults [47, 50].

### 5.2.2 CWT for Detecting Sensor Faults in The Ship Propulsion Benchmark

The fault scenario described in Table 2.1 is considered in this chapter, in which there are four abrupt sensor faults: $\triangle n_{high}$, $\triangle n_{low}$, $\triangle \theta_{high}$ and $\triangle \theta_{low}$, and an incipient sensor fault$\triangle k_y$ and a gain fault $\triangle \dot{\theta}_{inc}$.

As presented in Figure 5.1, the continuous wavelet transform is implemented to detect abrupt sensor faults according to the properties illustrated in Section 5.2.1. Figure 5.4 respectively shows the monitored signals $\theta_{mf}$ and $n_{mf}$ in fault scenario shown in Table 2.1.

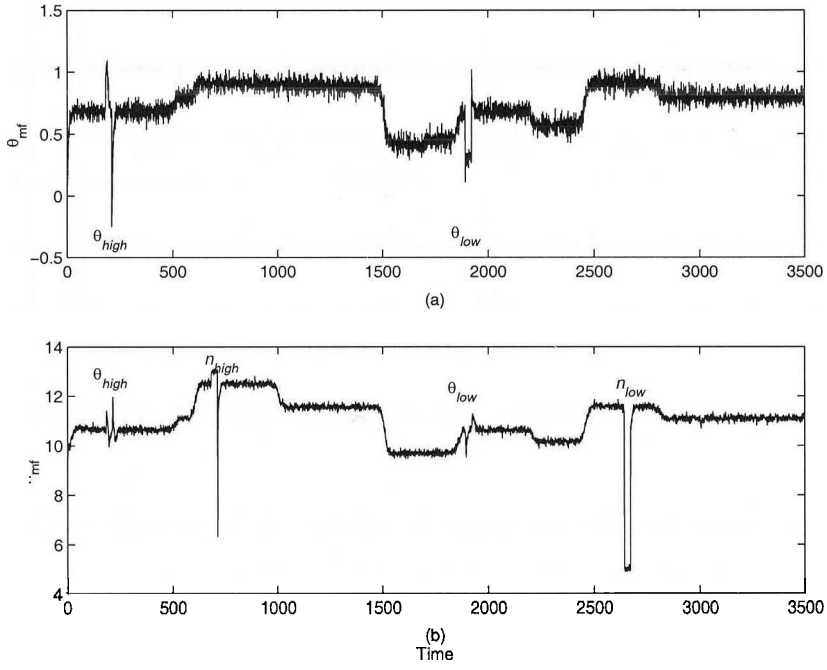


Figure 5.4: (a)Monitored $\theta_{mf}$ in the fault scenario (b) Monitored $n_{mf}$ in the fault scenario

One can observe that the $n_{mf}$ signal is not only affected by the faults of $\triangle n_{high}$ and $\triangle n_{low}$, but also by the faults of $\triangle \theta_{high}$ and $\triangle \theta_{low}$; while $\theta_{mf}$ is only affected by the faults of $\triangle \theta_{high}$ and $\triangle \theta_{low}$. This is explained by the system structure in which the shaft speed control loop is coupled to the propeller pitch control loop, as illustrated in section 2.1.

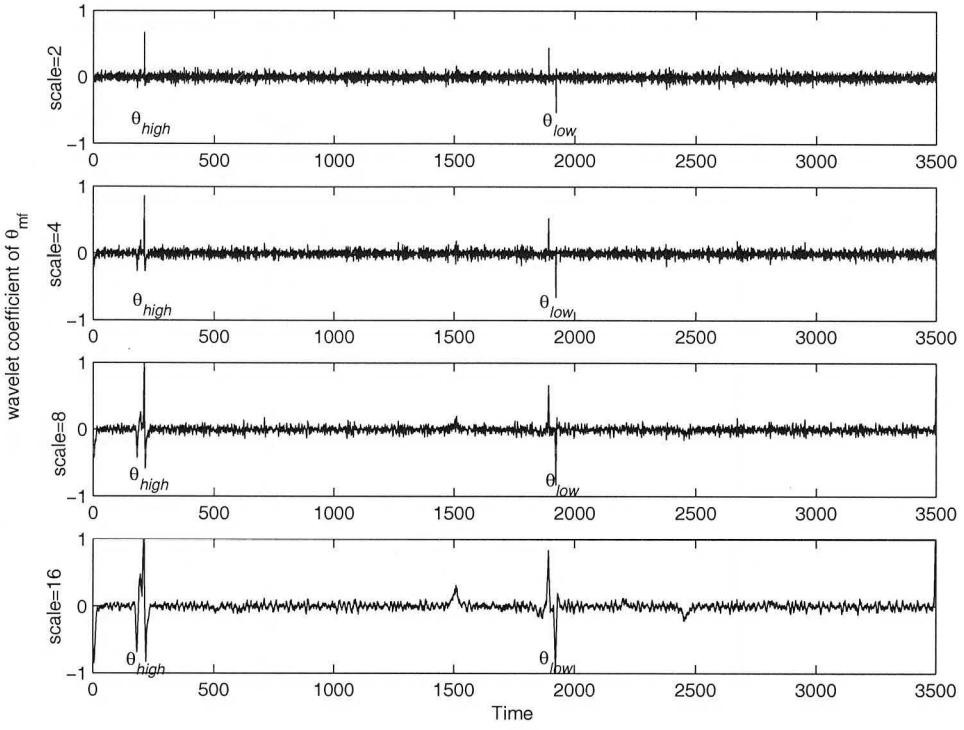


Figure 5.5: Wavelet coefficients of $\theta_{mf}$ at scales 2,4, 8 and 16

The CWT coefficients of the monitored signals on different scales are calculated separately and presented in Figure 5.5 and 5.6. The test has been done on the output signal $n_{mf}$ and $\theta_{mf}$ at different scales of 2,4, 8 and 16. The first derivative of the Gaussian function,'db1', is chosen as the mother wavelet function for these transformations.

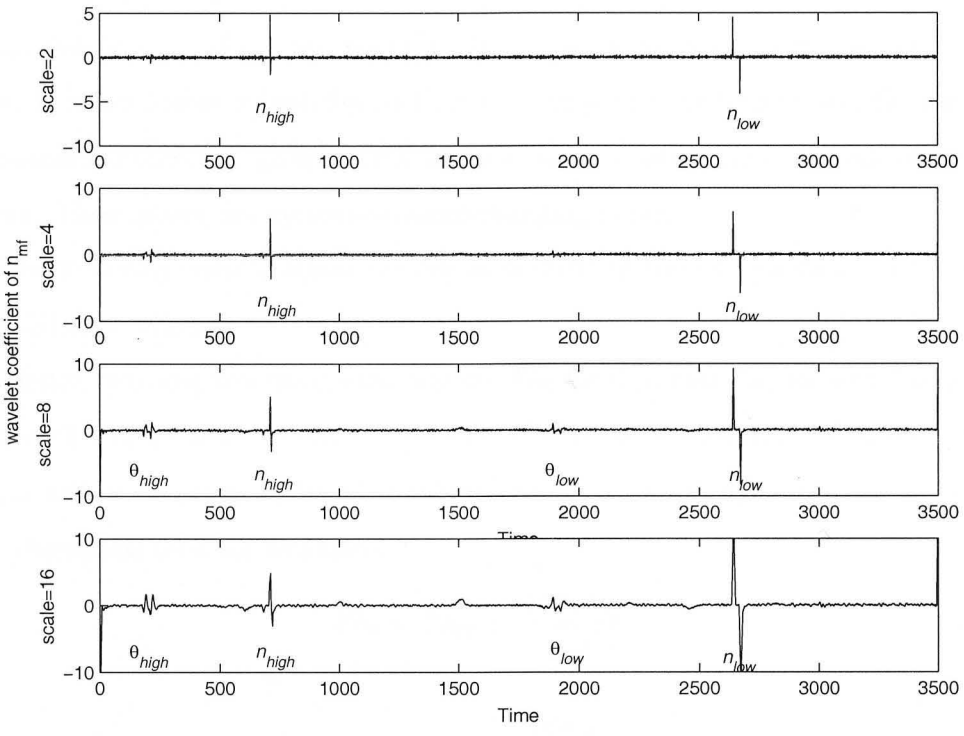


Figure 5.6: Wavelet coefficients of $n_{mf}$ at scales 2,4, 8 and 16

The series of CWT diagrams show an easy method to phase out the effects of noise on the monitored data based on modulus maximum values of CWT coefficients. From Figure 5.5 we can see that the modulus maxima of the wavelet transform at fault occurring points ($\triangle\theta_{high}$ and $\triangle\theta_{low}$) gradually grow with increasing scale. At the same time, the amplitudes of CWT of noise decay. From Figure 5.6, it can be seen that the modulus maxima of the wavelet transform at fault-occurring points ($\triangle n_{high}$, $\triangle n_{low}$ and $\triangle\theta_{high}$, $\triangle\theta_{low}$) also keep the same value or gradually grow with increasing scale. It is noticeable in both figures that at a comparatively larger scale, the wavelet transform modulus also grows a little larger at some points other than fault-occurring points. These points are system-dynamic-changing points.

The following considerations are key in addressing the FDI problem for the ship propulsion benchmark.

Firstly, two adaptive thresholds, namely $Th_\theta$ for $\theta_{mf}$, and $Th_n$ for signal $n_{mf}$, are used with a scale of 16 to ensure that only the modulus maxima at fault-occurring points will be detected, rather than falsely detecting system dynamic change points. The thresholds used are as follows:

$$Th_\theta = Th_{\theta 0} \pm |\frac{15\triangle\theta_{ref}}{\triangle T}| \tag{5.7}$$

and

$$Th_n = Th_{n0} \pm |\frac{15\triangle n_{ref}}{\triangle T}| \tag{5.8}$$

where $\Delta T$ is a time interval. The $n_{ref}$ and $\theta_{ref}$ are two input signals in the benchmark system. After some trial and error, the $Th_{n0}$ or $Th_{\theta 0}$ were set to $\pm 0.4$ and $\pm 0.2$ respectively on the two figures of scale 16. For both $Th_n$ and $Th_\theta$, one positive and one negative threshold are generated. To get a positive threshold for either $Th_n$ or $Th_\theta$, a positive value of corresponding $Th_{n0}$ or $Th_{\theta 0}$ needs to be used. At the same time, + should be selected as the sign of the second factor in (5.7) and (5.8), and vice

versa. The adaptive thresholds go through a low-pass filter with transform function $H(s) = \frac{0.04}{s+0.04}$ to make their behavior smoother. At system-dynamic-change points, the thresholds become larger so as to tolerate the system dynamic changing, thereby avoiding possible false alarms. Figure 5.7 shows the adaptive thresholds on a wavelet transform of scale 16 on monitored signal $\theta_{mf}$:

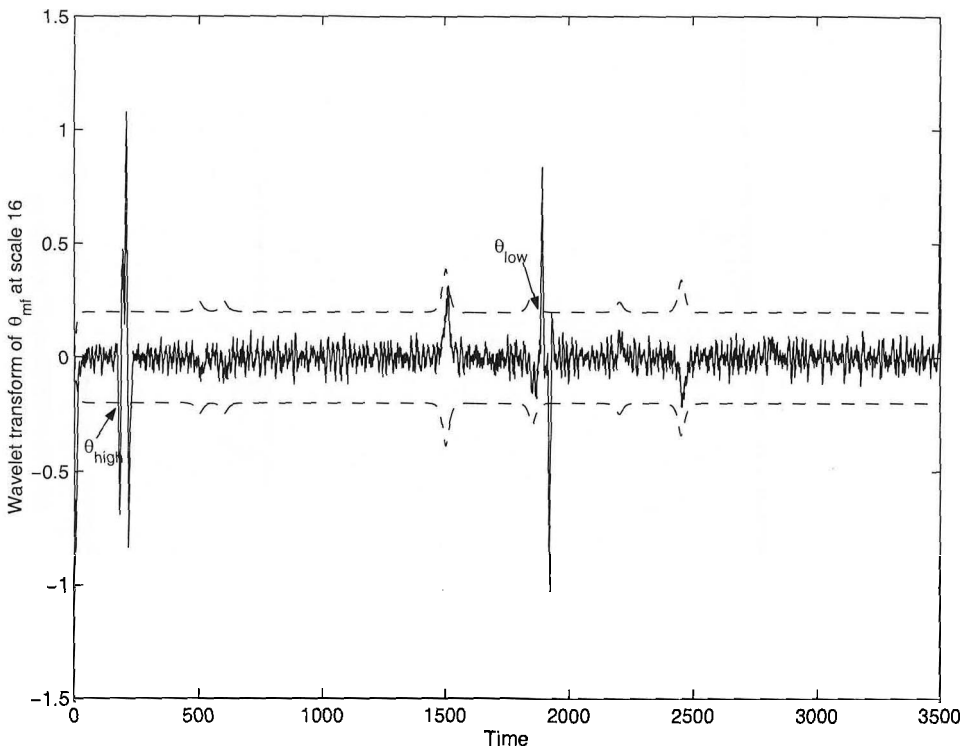


Figure 5.7: Wavelet transform coefficients of $\theta_{mf}$ at scale 16

Four types of faults $\triangle n_{high}$ , $\triangle n_{low}$ , $\triangle \theta_{high}$ and $\triangle \theta_{low}$ are detectable by monitoring the CWT of $n_{mf}$. Similarly, two types of faults, $\triangle \theta_{high}$ and $\triangle \theta_{low}$, can be detected by monitoring the CWT of the signal $\theta_{mf}$. These faults can be detected very promptly, all within one second.

For the purpose of fault isolation between two different fault types: $\triangle \theta$ and $\triangle n$,

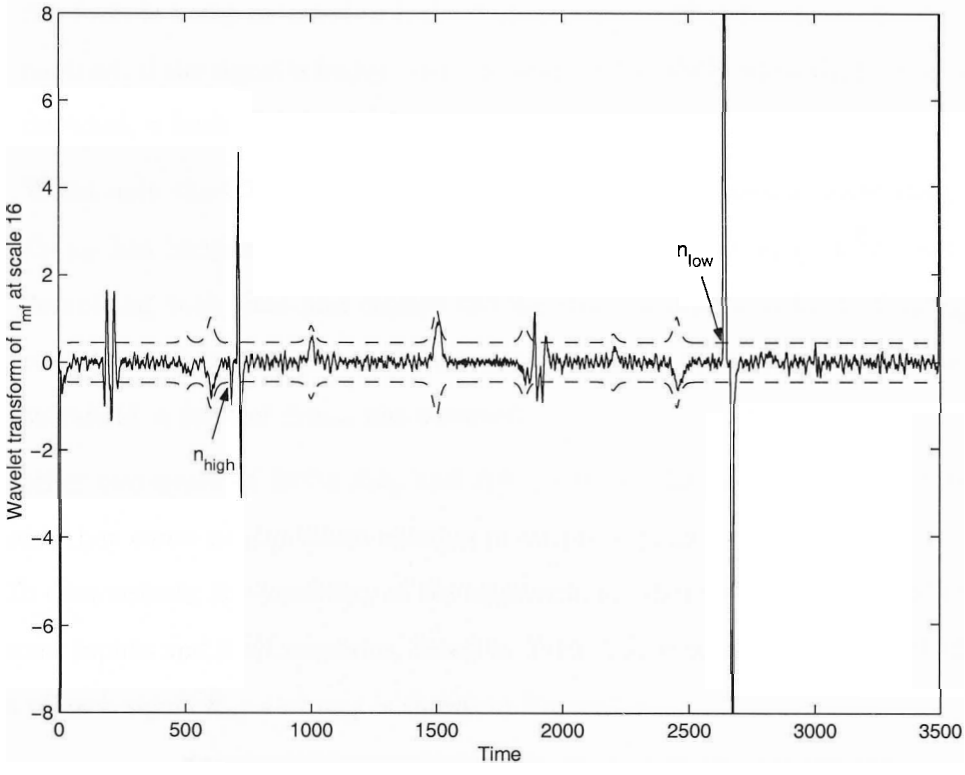


Figure 5.8: Wavelet transform coefficients of $n_{mf}$ at scale 16

the wavelet transforms of $\theta_{mf}$ and $n_{mf}$ need to be considered jointly. The following possibilities exist during the FDI process:

1. When the CWTs of both signals detect faults at the same time, which means either $\triangle\theta_{high}$ or $\triangle\theta_{low}$ has happened, check the CWT on scale 16 of the signal $\theta_{mf}$. If the coefficients are crossing the negative threshold at the point when the fault is being detected, a fault of $\triangle\theta_{high}$ is considered to have occurred. In contrast, if the signal is higher than the positive threshold when the fault is being detected, a fault of $\triangle\theta_{low}$ has occurred.
2. When only the CWT of the $n_{mf}$ detects faults, which means either $\triangle n_{high}$ or $\triangle n_{low}$ has happened, check the CWT on scale 16 of the $n_{mf}$. If the signal is decreasing with time and crosses the negative threshold, a fault of $\triangle n_{high}$ is considered to have occurred. In contrast, if the signal crosses over the positive threshold, a fault of $\triangle n_{low}$ has occurred.

The other two types of faults $\triangle k_y$ and $\triangle\dot{\theta}_{inc}$ can not be detected by this method because they cause no significant changes in output signals.

To demonstrate the feasibility of the approach, another set of checking data with different inputs and fault scenarios, listed in Table 5.2, is tested. The wavelet transform of each signal $\theta_{mf}$ and $n_{mf}$ is shown in Figure 5.9.

| Fault | Time duration / Magnitude | Fault | Time duration / Magnitude |
|---|---|---|---|
| $\Delta n_{high}$ | 1020s-1060s/15 | $\Delta\theta_{high}$ | 580s-610s/2 |
| $\Delta n_{low}$ | 2110s-2150s/5 | $\Delta\theta_{low}$ | 1700s-1750s/-0.9 |
| $\Delta k_y$ | 2900s-3300s/0.2 | $\Delta\dot{\theta}_{inc}$ | 800s-1700s/0.00001 |

Table 5.2: Fault scenario for checking the CWT approach

This method is easy and accurate, and even though the fault magnitude itself

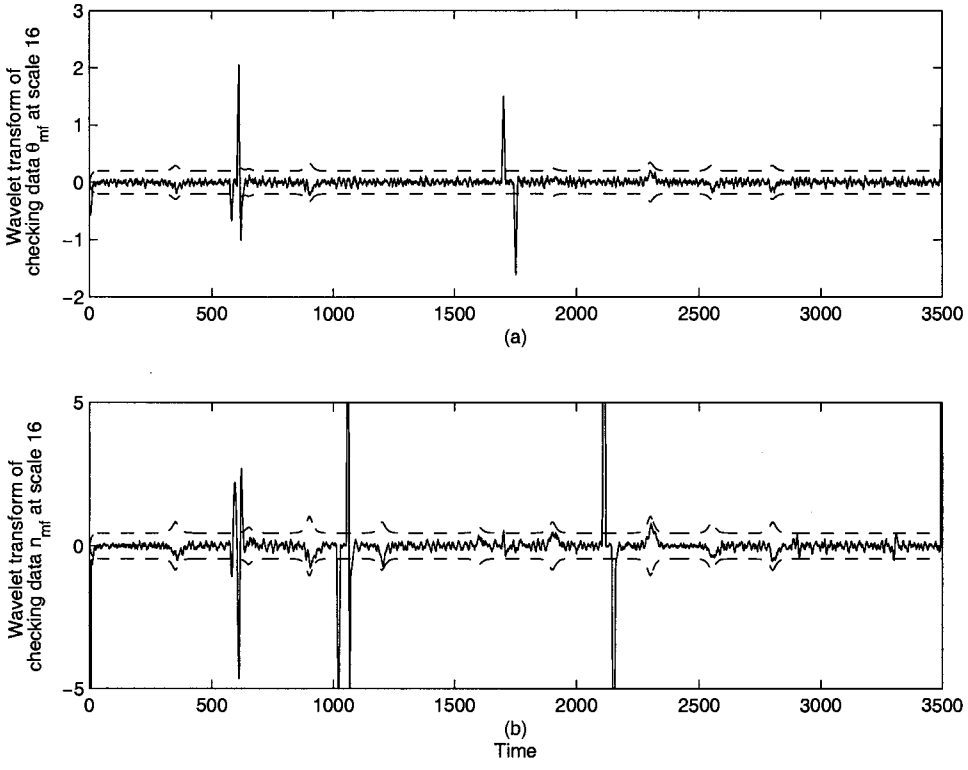


Figure 5.9: Test the CWT approach: (a)Wavelet transform coefficients of $\theta_{mf}$ at scale 16 (b)Wavelet transform coefficients of $n_{mf}$ at scale 16

is small, the modulus maxima of the CWT coefficients are significantly large. This technique can readily be applied to other fault detection problems.

### 5.2.3 DWT for Detecting and Identifying Incipient Faults

Detecting incipient faults is always a challenge in FDI because they tend to be unnoticed amidst the system uncertainties and noise. Serious damage could result to a system due to the delayed detection and the failure to follow-up with corrective solutions.

One reason for the difficulty in detecting incipient faults is because gradual changes are so much submerged in signals polluted by noise and uncertainty that an appropriate detection criteria is difficult to set. Wavelet analysis is useful in de-noising signals and revealing signal trends that are otherwise hidden by noise. The overall trends are recovered and the subsequent comparison and detection could be performed. In terms of wavelet analysis, trend is the lowest frequency part of the signal and corresponds to the greatest scale value. As the scale increases, the resolution decreases, producing a better estimate of the unknown trend. Another way to think of this is in terms of frequency. Since successive approximations progressively possess less high-frequency information, with the higher frequencies removed, what is left is the overall trend of the signal.

In this section, a discrete time wavelet decomposition based FDI approach for the incipient fault $\triangle\dot{\theta}_{inc}$ is presented. The assumptions are that all the sensor faults have been detected by the CWT as shown in section 5.2.2 and the incipient fault happens in the steady state. Since the sensor faults have been detected by the method based on the CWT, what remains is to detect the leakage incipient fault $\triangle\dot{\theta}_{inc}$ and the gain

fault $\triangle k_y$, as shown in Figure 5.1. As the propeller pitch control loop is not coupled with other control loops, it can be readily seen that the fault $\triangle k_y$ does not affect the signal $\theta_{mf}$, while there is no doubt that $\triangle \dot{\theta}_{inc}$ will affect the sensor measurement signal $\theta_{mf}$.

In this section, we use the DWT as a tool to analyze the properties of signal $\theta_{mf}$ and derive a feasible method to detect $\triangle \dot{\theta}_{inc}$. The signal $\theta_{mf}$ is first studied in healthy and faulty cases and meaningful observations are made on it. From section 5.1.2, the multiple level wavelet decomposition is capable of revealing the signal trend, which is represented by the approximation $C_a$. The more stages the decomposition has, the more fundamental shape the $C_a$ represents.

A 6-level wavelet decomposition is performed on signal $\theta_{mf}$ and the mother wavelet is selected as the first-order derivative of the Gaussian function. Since the length of the resulting approximation coefficient vector $C_{a6}$ has been shortened to $\frac{1}{2^6}$ of the length of its original, to compare and reveal the original signal's trend, the approximation vectors are reconstructed to the original length of the signal being analyzed.

In Figure 5.10 and Figure 5.11, the upper box 's' shows the original signal $\theta_{mf}$ in the healthy and faulty case, respectively. There is hardly any perceivable differences during the time period 800s-1500s when $\triangle \dot{\theta}_{inc}$ happens. The $a_6$ in the second row is the reconstructed approximation coefficient $c_{a6}$, and $d_1, \ldots, d_6$ are the reconstructed detail coefficients of $C_{d1}, \ldots, C_{d6}$. The original signal 's' is composed of the sum of the sixth level reconstructed approximation $a_6$ and reconstructed details from all the six levels, i.e. $s = a_6 + d_6 + d_5 + d_4 + d_3 + d_2 + d_1$. To better analyze the signal $a_6$, a finer description of $a_6$ of $\theta_{mf}$ in the faulty case is presented in Figure 5.12. The reconstructed $a_6$ presents a stair-like structure which reveals the overall trends of the signal. The dotted line between 800s and 1500s is the $a_6$ approximation in the healthy

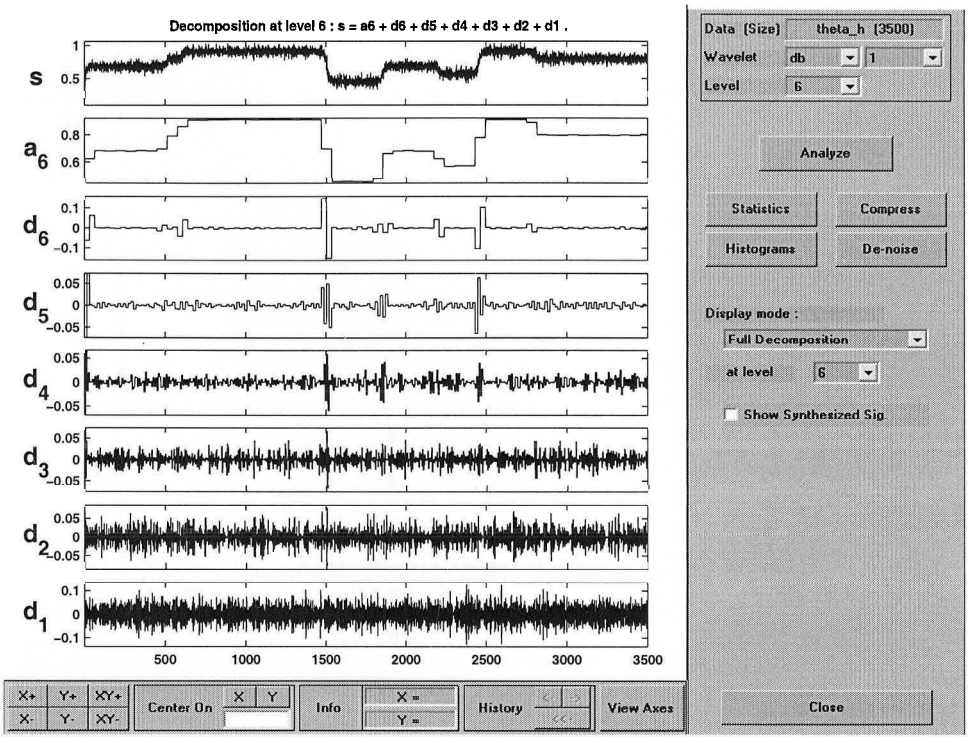


Figure 5.10: Six level wavelet decomposition of $\theta_{mf}$ in a healthy case

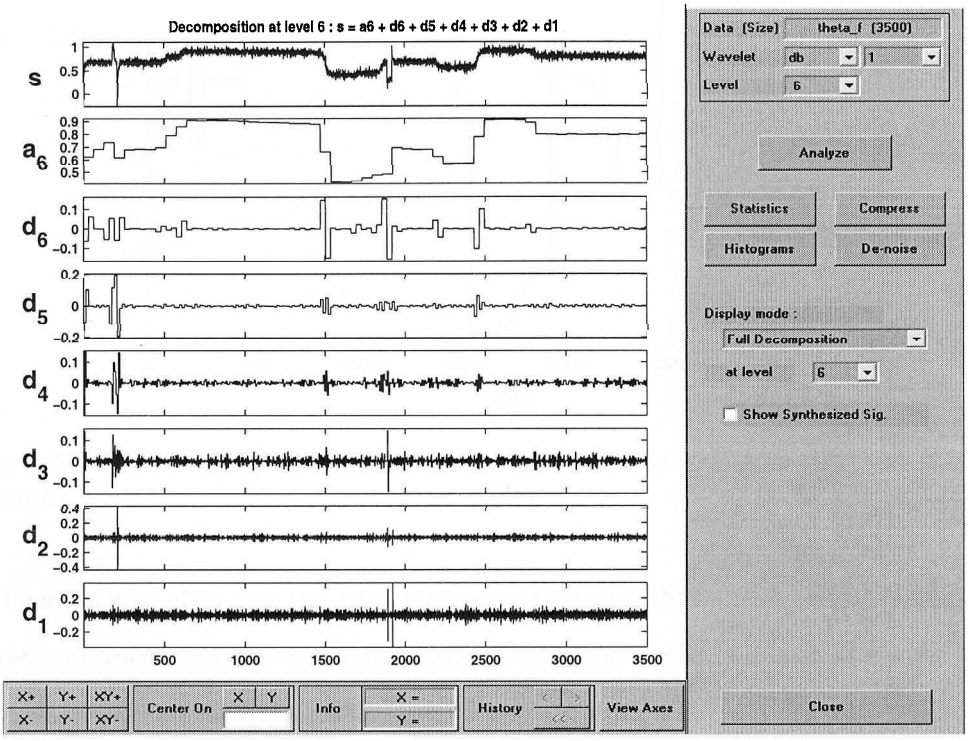


Figure 5.11: Six level wavelet decomposition of $\theta_{mf}$ in the fault scenario

case.

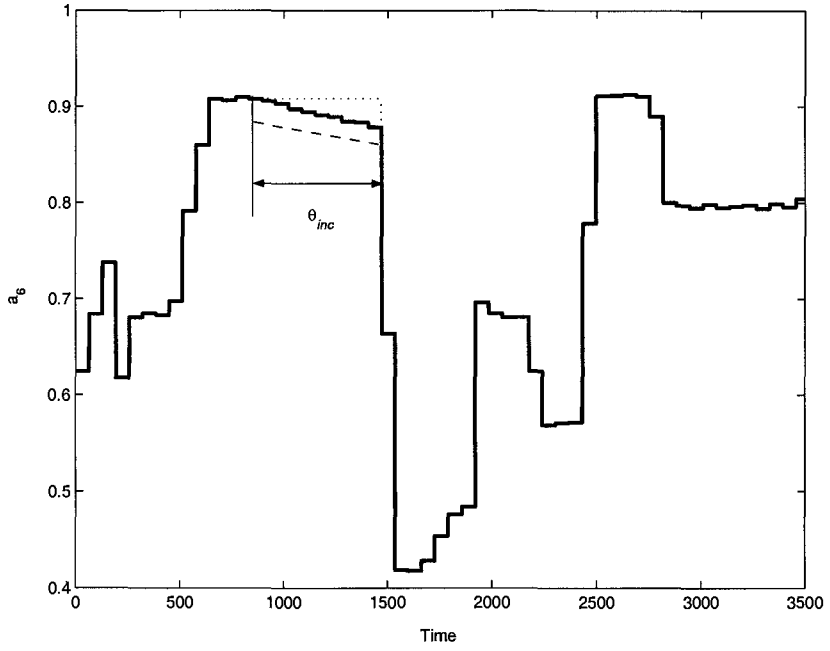


Figure 5.12: Reconstructed sixth level approximation $a_6$ of $\theta_{mf}$ in the fault scenario (zoomed in)

Comparison of $a_6$'s in the time interval 800s-1500s shows that the incipient fault causes the reconstructed approximation to decrease slowly while in the same time interval of healthy case the reconstructed approximation keeps a constant value.

The observations of the faulty signal show that the leakage incipient fault has two properties which can be used to distinguish it from the healthy operation and other fault cases. During the steady state, in the absence of incipient fault $\triangle\dot{\theta}_{inc}$, the mean value of the reconstructed approximation signal $a_6$ stays at a steady level. The incipient fault causes the $a_6$ of signal $\theta_{mf}$ to degrade gradually. In addition, the slope of $a_6$ in the faulty part stays at a constant level. A detection and isolation approach can be derived based on the preceding characteristics of signal $a_6$ in this

ship propulsion benchmark:

1. The value of the reconstructed approximation $a_6$ of the monitored signal $n_{mf}$ at time $k$ is compared to that value of the preceding time instance.
2. If the value is decreasing monotonically, a moving window with 50 consecutive $a_6$ samples is created. Then a linear fitting is performed to obtain a linear approximation of the windowed slope.
3. With the window shifting, if a consecutive number of slopes stay around a constant value, or the deviation of these slopes is around or less than 2 percent of the preceding slope, the slope is considered a constant. In this case, an incipient fault is assumed to have occurred.

This consecutive number of constant slopes required to trigger an alarm could be fixed at 100 so that the criteria for detection time could be met. However, to avoid the false detection, this number is chosen as 200. There is a tradeoff between the false detection probability, missed detection probability, and required detection time limits. The detection time could be shorter and meet the time requirement if the incipient fault is considered to be sufficiently serious that the price of a higher probability of false detection can be afforded. Since the severity of the incipient fault is not so high, the detection time is therefore set to be 200.

### 5.2.4 Fuzzy Model Based FDI

The wavelet signal analysis methods are able to detect and isolate five faults that occur in the ship propulsion benchmark, as illustrated in the preceding sections. There are still some faults which do not exhibit significant influence on a particular measurable signal. Therefore, they tend to be difficult to detect by the signal processing methods,

such as fault $\triangle k_y$ in this problem. A previously illustrated fuzzy model-based FDI is implemented in this section to detect the $\triangle k_y$ fault, as shown in Figure 5.1. The basic principles of the GA-fuzzy model-based FDI approach have been provided in Chapter 4. The inputs to a GA-fuzzy model are still the same $\theta_{mf}$, $Y_m$, $U_m$, and the model output is the $\hat{n}_{mf}$. The difference between the model output $\hat{n}_{mf}$ and measured signal $n_{mf}$ generates the residual.

From the discussion in last chapter, it is clear that the large residuals of this model are caused by three possible faults: $\triangle n_{high}$, $\triangle n_{low}$ and $\triangle k_y$. Hence, if the CWT sounds a sensor alarm indicating either of these two sensor faults, the residual of fuzzy model will not be taken into consideration. When the CWT does not detect any sensor faults and the residual of GA-fuzzy model exceeds the threshold, a gain fault $\triangle k_y$ is considered to have occurred, as summarized in Table 5.1. This saves efforts of employing another fuzzy system to distinguish the three faults as what has been done in Chapter 4. The simulation results are similar to that in Chapter 4.

## 5.3 Chapter Summary

There are following discussions and concerns regarding the implementation of the wavelet analysis method and combined FDI system on the ship propulsion system.

1. The wavelet analysis methods are accurate and timely in diagnosing sensor faults. It combines the information from time, scale and frequency domains in analyzing the signal.

2. For simplicity, we seek to use as few scales as possible to clearly show the evolution of the continuous wavelet coefficient to make faults distinguishable from the noise and system dynamic change points.

3. Distinguishing between the two $\Delta n$ faults and two $\Delta\theta$ faults is not always necessary. Knowing that a fault is occurring on shaft speed measurement is sufficient to locate the problem and take corrective actions. The same reason applies to $\Delta\theta$ faults.

4. There are no requirements on prior mathematical or mechanical knowledge of the system so this method is relatively easy to implement for sensor faults. However, because of the lack of mathematical insight into the system, it might not be as powerful and complete when used in detecting faults in which many signals are involved.

5. Where combined models are used to isolate faults, relative timing must be considered. In this case this is not an issue for two reasons. Firstly, the CWT method is fast to detect abrupt sensor faults which need very prompt correction actions. Secondly, the gain fault has a severity level "medium" and allows a longer detection time than the abrupt sensor faults. Even though the gain fault decision is delayed until the CWT decision is made, it is still safe and within the detection time requirements.

6. As shown in Figure 5.1, the six faults can be detected and isolated by the wavelet analysis and model-based combination. The combined signal processing and model-based FDI uses appropriate methods to detect different faults. This has been shown to be a successful approach in terms of ease of implementation, promptness of response and completeness in types of faults detected for the ship propulsion benchmark.

# Chapter 6

# Summary and Future Work

Three different FDI methods, the dynamic time warping (DTW), the GA-fuzzy model based approach, and wavelet signal processing method have been applied to a non-linear ship propulsion benchmark. Comparisons and conclusions have been made on these three approaches:

1. Pattern recognition methods reveal a significant insight into FDI problems. Including DTW, they perform supervised and unsupervised classifications, without knowledge of the particular system's mathematical properties. Instead, certain other features such as statistical or geometrical criteria play an important role in helping the classification. DTW, as presented in Chapter 3, is capable of extracting the pattern similarity by calculating a minimum distance and therefore, successfully classifying the fault patterns. Since it needs a significantly long time period of data to complete the distance calculation, its application to real time systems is limited.

2. The GA-fuzzy model based FDI presented in this thesis builds a fuzzy system based on numerical data. The fuzzy model was optimized by a GA to search for

a global minimum. The generated fuzzy rules can be expressed in if-then format to supply the user with human-comprehensible knowledge. This knowledge can be stored as expert knowledge for future studies of the same system.

3. Even though in building the GA-fuzzy model in this thesis, some structural analysis was considered, it is not necessary to assume structural analysis in practice. A purely data-driven model could be built using statistical methods in selecting suitable model structures. In reality, it is very difficult, if not impossible, to obtain the full model of a complex system. This makes the data-driven, fuzzy model based methods a promising area for future work.

4. The signal processing approaches, particularly wavelet signal processing, successfully solve the FDI problem for abrupt sensor faults. Wavelet analysis allows monitoring of the changes and reveals the properties of relative signals in time, frequency and scale domains. Wavelet analysis methods come up with FDI decisions promptly and precisely.

   Detecting incipient faults is usually a challenge for FDI. It is difficult to diagnose, even with observers. This thesis presents a new thought based on DWT and its application to the ship propulsion benchmark. It is able to detect the incipient fault by taking advantage of the fact that wavelet decomposition approximation coefficients show the basic trend of the signal.

5. Table 6.1 is the summary of the three methods implemented on the ship propulsion benchmark. This thesis presents simulation results that the DTW and GA-fuzzy-model-based approaches can detect and isolate three faults, and wavelet analysis is capable of detecting and isolating five sensor faults. The applications of DTW and GA-fuzzy model approaches to the other three faults are left as future work. It is promising that the GA-fuzzy model approach can detect the

| Fault | DTW | GA-fuzzy model | Wavelet analysis |
|---|---|---|---|
| $\Delta n_{high}$ | ✓ | ✓ | ✓ |
| $\Delta n_{low}$ | ✓ | ✓ | ✓ |
| $\Delta k_y$ | ✓ | ✓ | × |
| $\Delta \theta_{high}$ | | | ✓ |
| $\Delta \theta_{low}$ | | | ✓ |
| $\Delta \dot{\theta}_{inc}$ | | | ✓ |

Table 6.1: Detection results for the three methods on ship propulsion benchmark

$\triangle\theta$ faults if the correct model structure and residual are selected. Detecting the incipient fault will be a challenge for the DTW and GA-fuzzy model methods.

6. Chapter 5 also provides a new concept for a hybrid system to perform the multiple fault detection and isolation tasks. The continuous wavelet transform method at different scales can be used to detect the sensor faults, and the different behaviors of these signals at fault-occurring points are used to help isolate them. The incipient sensor fault is detectable after certain wavelet decomposition and reconstruction. A fuzzy model is built to detect the $\triangle k_y$ fault, since the statistical or frequency changes caused by this fault are not significant enough to use the wavelet decomposition approach. It is a good result since all the six faults are detected and isolated using the above procedures.

It is recommended that future work includes:

1. A rigorous investigation of the statistics of fault detection and isolation decision criteria, including optimal settings of decision thresholds and the tradeoff between false detection probabilities and missed detection probabilities, is a good area for future work.

2. A moving windowed Dynamic Time Warping approach could be investigated to detect and isolate faults in a short time interval. This would overcome the primary disadvantage that the DTW method requires a considerable amount of data before any comparison and conclusion can be reached, leaving it suitable only for off-line applications. The window is of specific length and the calculation of the minimum distance with sample patterns is repeated each shift. The shorter the window, the faster the DTW will detect faults.

3. More residuals could be generated by more GA-fuzzy models to predict the system behavior on different variables. Hence, more faults could be detected and isolated based on analysis and consideration of these additional residuals.

4. As it can be seen from the hybrid system in Chapter 5, both the combining of methods to yield better performance and the matching of a particular method to a particular class of faults depending on the fault characteristics are promising areas for future research.

# Bibliography


[1] P. Amann et.al, "Benchmark application of fuzzy observers to fault detection on a ship propulsion system", *In Proc. European Control Conference* 1999 (ECC'99).

[2] R. Babuka, *Fuzzy modeling for control*, Kluwer Academic Publishers, 1998.

[3] J. D. Bemdtsen and R. Izadi-Zamanabadi, "FDI using neural networks-application to ship benchmark engine gain", technical report, Aalborg University, Denmark.

[4] M. Blanke, et.al, "Fault monitoring and re-configurable control for a ship propulsion plant", *Journal of Adaptive Control and Signal Processing*, pp. 671-688, 1998.

[5] M. Blanke and T. F. Lootsma, "Adaptive observer for diesel fault detection in ship propulsion benchmark", technichal report, Aalborg University, Denmark.

[6] J. J. Buckley and T. Feuring, "Fuzzy and neural: interactions and applications", *ser. Studies in Fuzziness and Soft Computing*, Heidelberg, Germany: Physica-Verlag, 1999.

[7] L. Y. Cai and H. K. Kwan, "Fuzzy classifications using fuzzy inference networks", *IEEE Trans. Systems, Man, and Cybernetics-Part B: cybernetics*, vol.28, no.3, Jun 1998.

[8] J. M. F. Calado, M. J. G. C. Mendes, J. M. G. Sa da Costa and J. Korbicz, "Neuro and neuro-fuzzy hierarchical structures comparison in FDI: case study", *15th Triennial World Congress*, Barcelona, Spain, 2002 IFAC.

[9] J. Chen and C.M. Liao "Dynamic process fault monitoring based on neural network and PCA ", *Journal of Process Control*, vol. 12, pp. 277-289, 2002.

[10] L. H. Chiang, R. J.Pell, and M. B. Seasholtz, "Exploring process data with the use of robust outlier detection algorithms", *Journal of Process Control*, vol. 13, pp. 437-449, 2003.

[11] J. Colomer, J. Melemdez, and F. I. Gamero, "Pattern recognition based on episodes and DTW: application to diagnosis of a level control system", *QR'02 Proceeding, 16th International Workshop on Qualitative Reasoning*, Barcelona, Spain, Jun 2002.

[12] J. C. Dunn,"A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters," *Journal. Cybern*, vol. 3, no. 3, pp. 32-57, 1973.

[13] M. Gen and R. Cheng, "Genetic algorithm engineering optimization", John Wiley and Sons Inc. 2000.

[14] A. Grossgrable and J. Morlet, "Decompositions of Hardy functions into square integrable wavelets of constant shape", *SIAM Journal of Mathematical Analysis*, 15(4), pp. 723-736, Jul 1984.

[15] S. Guillaume, "Designing fuzzy inference systems from data: an interpretability-oriented review", *IEEE Trans. Fuzzy Systems*, vol. 9, no. 3, Jun 2001.

[16] D. Gustafson and W. Kessel, " Fuzzy clustering with a fuzzy covariance matrix", *Proc. IEEE CDC*, San Diego, CA, USA, pp. 761-766, 1979.

[17] H. Hellendoorn and D. Driankov, *Fuzzy model identification*, Springer, 1997.

[18] R. Izadi-Zamanabadi and M. Blanke, "Ship propulsion system as a benchmark for fault-tolerant control." technical report, Control Engineering Dept., Aalborg University, Aalborg, Denmark, 1998.

[19] R. Izadi-Zamanabadi and M. Blanke, "A ship propulsion system as a benchmark for fault-tolerant control", *Control Engineering Practice*, vol. 7, no. 2, pp. 227-239, 1999.

[20] R. Izadi-Zamanabadi, and M. Blanke, "A ship propulsion system model for fault-tolerant control", Department of Control Engineering, Aalborg University, Aalborg, Denmark.

[21] R. Izadi-Zamanabadi, M. Blanke, and S. Katebi, " Structural modeling and fuzzy-logic based diagnosis of a ship propulsion benchmark", *IFAC* 2000.

[22] A. Kassidas, P. A. Tayor and J.F.Macgregor, "Off-line diagnosis of deterministic fault in continuous dynamic multivariable processes using speech recognition methods", *Journal of Process Control*,vol. 8, no. 5-6, pp. 381-392, 1998.

[23] K. Kim and A. G. Parlos, "Induction motor fault diagnosis based on neuropredictors and wavelet signal processing", *IEEE/ASME Trans. Mechatronics*, vol.7, no.2, Jun 2002.

[24] S. Leonhardt and M. Ayoubi, "Methods of Fault diagnosis", *Control Engineering and Practice*, vol. 5, no. 5, pp. 683-697, 1997.

[25] T. F. Lootsma, R. Izadi-Zamanabadi, and H. Nijmeijer, " A geometric approach to diagnosis applied to a ship propulsion problem", report, Department of Control Engineering, Aalborg University, Aalborg, Denmark.

[26] T.F.Lootsma, "Observer-based fault detection and isolation for nonlinear systems", Ph.D thesis, Aalborg University, Aalborg, Denmark, 2001.

[27] S. Mallat, *A wavelet tour of signal processing*, Academic Press, 1999.

[28] S. Mallat and W. L. Hwang, "Singularity detection and processing with wavelets", *IEEE Trans. Information theory*, vol.38, no.2, Mar 1992.

[29] Z. Michalewic, *Genetic Algorithms+Data Structures=Evolution Programs*, 2nd ed. New York: Springer-Verlag, 1994.

[30] S. Mitra and Y. Hayashi, "Neuro-fuzzy rule generation : survey in soft computing framework", *IEEE Trans. Neural Networks,* vol. 11, no. 3, May 2000.

[31] S. K. Pal and P. P. Wang, *Genetic algorithm for pattern recognition*, CRC Press, 1996.

[32] R. J. Patton, P. M. Frank, and R. N. Clark, *Issues of Fault Diagnosis for Dynamic Systems*, Springer, 2000.

[33] G. Schreier and P. M. Frank, " Fault tolerant ship propulsion control: sensor fault detection using a nonlinear observer", *In Proc. European Control Conference* 1999, (ECC'99).

[34] M. Setnes and H. Roubos, "GA-fuzzy modeling and classification: complexity and performance", *IEEE Trans. Fuzzy Systems*, vol. 8, no. 5, Oct 2000.

[35] M. Setnes, R. Babuska, U. Kaymak, and H. R. van Nauta Lemke, "Similarity measures in fuzzy rule base simplification", *IEEE Trans. Systems, Man, and Cybernetics-Part B:Cybernetics*, vol. 28, No. 3, Jun 1998.

[36] S. Simani, C. Fantuzzi, and R. J. Patton, *Model-based fault diagnosis in dynamic systems using identification techniques*, Springer, 2003.

[37] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition", *IEEE Trans. Acoustics, Speech, and Signal Proc.*, vol. ASSP-26(1), pp. 43-49, 1978.

[38] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its application to modeling and control ", *IEEE Trans. Systems, Man and Cybernetics*, vol. 15, no. 1, pp.116-132, 1985.

[39] W. L. Tung and C. Quek, "GenSoFNN: a genetic self-organizing fuzzy neural network", *IEEE Trans. Neural Networks*, vol. 13, no.5, Sep 2002.

[40] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, "A review of process fault detection and diagnosis Part I: quantitative model-based methods", *Computers and Chemical Engineering* 27, pp. 293-311, 2003.

[41] V. Venkatasubramanian, R. Rengaswamy, and S. N. Kavuri, "A review of process fault detection and diagnosis Part II: qualitative models and search strategies", *Computers and Chemical Engineering* 27, pp. 313-326, 2003.

[42] J. S. Wang and C.S.George Lee, "Self-adaptive neuro-fuzzy inference systems for classification applications", *IEEE Trans. Fuzzy Systems*, vol.10, no.6, Dec 2002.

[43] L. X. Wang, *Adaptive fuzzy systems and control design for stability analysis*, Prentice Hall, Englewood Cliffs, New Jersey 07632, 1994.

[44] P. Wang and G. Vachtsevanos, *Fault prognosis using dynamic wavelet neural networks*, School of electrical computer engineering, Georgia Institute of Technology, Atlanta.

[45] A. Webb, *Statistical pattern recognition*, London: Arnold, 1999.

[46] Y. Xiong, *Robust fault diagnosis in linear and nonlinear systems based on unknown input, and sliding mode functional observer methodologies*, Ph.D thesis, Simon Fraser University, Canada, 2001.

[47] H. Ye, G. Z. Wang, and C.Z. Fang, " Wavelet application in fault detection", *ACTA Automatica Sinica*, vol. 23, no. 6, Nov 1997.

[48] J. Yen and L. Wang, "Simplifying fuzzy rule based models using orthogonal transformation methods", *IEEE Trans. Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 29, no. 1, Feb 1999.

[49] S. Yoon and J. F. MacGregor, "Fault diagnosis with multivariate statistical models part I: using steady state fault signatures", *Journal of Process Control*, vol. 11, pp. 387-400, 2001.

[50] J. Q. Zhang and Y. Yan, "A wavelet-based approach to abrupt fault detection and diagnosis of sensors", *IEEE Trans. Instrumentation and Measurement*, vol. 50, no. 5, Oct 2001.

[51] J. Zhang et.al,"Wavelet neural networks for function learning", *IEEE Trans. Signal Processing* , vol. 43, no. 6, Jun 1995.

[52] Q. Zhang and A. Benvensite, "Wavelet networks", *IEEE Trans. Neural Networks*, vol. 3, no. 6, Nov 1992.

[53] Y. Zhang and N. E. Wu, "Fault diagnosis for a ship propulsion benchmark: part1", *Proc. 14th IFAC World Congress*, pp. 569-574, 1999.