A RECURRENT ADAPTIVE TIME DELAY NEURAL NETWORK FOR FAULT DETECTION AND ISOLATION FOR THE SATELLITE'S ATTITUDE CONTROL SYSTEM

Shu ping Zhao

A Thesis

In

The Department

Of

Electrical & Computer Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Applied Science
in Electrical & Computer Engineering
at Concordia University
Montreal, Quebec, Canada

January 2007

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-28934-1
Our file Notre référence
ISBN: 978-0-494-28934-1

ABSTRACT

A Recurrent Adaptive Time Delay Neural Network for Fault Detection and Isolation for the Satellite's Attitude Control System

Shu ping Zhao

This thesis investigates a new Fault Detection and Isolation (FDI) scheme for the satellite's attitude control system by using a recurrent adaptive time delay neural network. The results obtained reveal that the proposed new scheme works quite well for detecting and isolating faults in the reaction wheel which cause the satellite to behave abnormally corresponding to either pitch, yaw or roll axes. Moreover, the promising robustness and insensitivity of the proposed neural network scheme due to external disturbances and noise have also demonstrated.

The results presented do indeed demonstrate the satisfactory capabilities and potential advantages of the proposed neural network based fault detection and isolation methodology. The specific faults considered are due to both voltage and current faults in the reaction wheels employed in the attitude control system of a satellite. Both multiple and simultaneous fault signatures and individual fault patterns have been investigated and the results presented validate the very good performances obtained by the proposed neural network. Furthermore, the recovery natures of these faults have also been investigated in several case studies in which the satellite operates under continuous setpoint operating changes.

To my father Mr. Younian Zhao

and to the memory of Mrs. Yuanxiang Zhao, my mother.

Contents

List of Figures

List of Tables

Chapter 1

Introduction: General Background and Research Goals

1.1. Brief Introduction to Attitude Control Subsystem

Since the late 1950s, studies on spacecraft attitude control have achieved fruitful results. The attitude control subsystem (ACS) aims on stabilizing and orienting the spacecraft in desired setpoint positions despite the influence of external disturbances [1-3]. Three types of control techniques are commonly implemented to meet design requirements, namely: *Gravity-gradient control*, *spin control* techniques and *three – axis control* techniques [3]. *Gravity-gradient control* orientates the spacecraft pointed toward the Earth by utilizing its inertial properties. This technique takes advantage of the fact that an elongated object in a gravity field tends to align its longitudinal axis through the Earth's center. The alignment is generated by torque which is symmetric around the undershoot vector to eliminate the effect caused by the yaw axis around the undershoot vector. This tendency is adopted on spacecraft in orbits without yaw orientation requirements.

Spin stabilization is a technique in which the entire spacecraft is made to rotate. As a result, its angular momentum vector remains approximately fixed in inertial space.

The shape of the spinner affects the behavior of the technique. Disk-shaped spinners are passively stable while pencil-shaped vehicles are not. Spinners can survive for long periods without attention and they provide components a thermally benign circumstance as well as a scanning motion for sensors. However, the main disadvantages of spin stabilization also need to be considered. First of all, the vehicle mass properties must be controlled to ensure the desired spin direction and stability. Secondly, the angular momentum vector requires more fuel to reorient than a vehicle with no net angular momentum does, therefore, reducing the implementation of this technique for payloads need to be pointed frequently.

Nowadays, the most common attitude control technique is the *three-axis control* in which they maneuver and can be stable and accurate, depending on sensors and actuators that are used. The control torque about the axes of 3-axis systems comes from combinations of momentum wheels, reaction wheels, control moment gyros, thrusters or magnetic torques. Briefly, these systems take two forms: one utilizes momentum bias by placing a momentum wheel along the pitch axis; another is called zero momentum with a reaction wheel on each axis. In this thesis, the attitude control is implemented based on the second approach that is zero momentum.

Generally, four reaction wheels (3 active and 1 redundant) are used on the spacecraft and that need to be controlled. Each active reaction wheel is aligned with one of the body axis of spacecraft and it can rotate in either direction and provide reaction torque for the related axis control. The redundant one will be excited in case of any of the other three wheels failed. For instance, when facing with secular disturbances, the wheel will be drifted toward saturation. In this case, an external torque, such as magnetic torque will be applied to force the wheel speed back to zero. This process is called momentum dumping [3].

In order to solve the problem of attitude control, researchers have developed a number of techniques based on both classical control and modern control [1-4]. With the development of these techniques, intelligent-based approaches have also been introduced for performing accurate attitude control design [5-7]. In this thesis, the research is based on a fictitious satellite which has three reaction wheels as actuators to achieve attitude control on three axes. Three classical PID controllers are independently designed for three separate control loops as controllers.

Reaction wheels are momentum exchange devices which provide reaction torque as well as store angular momentum to the spacecraft. They operate in environments with disturbances and unpredicted external influences. The performance of reaction wheels impact the attitude control significantly. Therefore, for the spacecraft attitude control, they play a crucial role for stability and control. The internal faults in wheels should be detected and isolated as soon as possible to avoid causing serious damage to the spacecraft attitude control. Generally, one may consider three types of faults in the reaction wheel, namely: (a) bus voltage drop, (b) motor current drop and, (c) temperature fault. Temperature is highly related to the viscous friction, which is the main friction factor of the wheel as discussed subsequently in the following chapter. The temperature fault will cause the wheel to operate abnormally. However, given that the internal temperature for the wheel is regulated and does not change much, and based on the fact that the spacecraft operate robustly with small temperature changes, only the first two types of fault are studied in this thesis. The bus voltage should be sufficiently high to avoid elimination of the voltage headroom. Moreover, a low bus voltage not only reduces the capacity of the torque but also causes the attitude of spacecraft to become seriously uncontrollable. Motor current drop has the same impact on the vehicle as the bus voltage drop does. Once the faults occur, the reaction wheel will lose part of the power and consequently lose capability to provide enough reaction torque to follow the attitude setpoint change command. A high fidelity mathematical model of a reaction wheel [8] will be discussed in Chapter 2.

1.2. Literature Review of Fault Detection and Isolation

1.2.1. Basic Concepts for the Fault Diagnosis Problem

Two basic terminologies are introduced to facilitate understanding the fault diagnosis problem. The information collected here can be traced back to the *SAEFPROCESS* Technical Committee [9] documentation.

- The term 'fault' is defined as an unexpected change of system function. That is an un-permitted deviation of at least one characteristic property or parameter of the system from the acceptable, usual or standard condition. Such a fault or malfunction often causes an unacceptable deterioration of the performance of the system or even leads to dangerous situations.
- The terminology 'failure' is denoted as complete breakdown of a system component or function. Generally, feedback from a faulty sensor can quickly result in instability causing a failure in a control.

Therefore, the goal of fault diagnosis is clear in the sense that it tries to avoid system shut-down, breakdown and even catastrophes, therefore a fault needs to be detected and isolated early before it causes a failure.

A fault diagnosis system generally comprises of three stages [10]:

- Fault detection: to determine the presence of a fault in the system or not;
- Fault isolation: to determine the location of the fault. For instance, which sensor or actuator has become faulty; and
- Fault identification: to estimate the size, type and time-variant behavior of a fault.

Consequently, the basis of the diagnosis is the fault detection step which must be performed properly in order to able to execute the further isolation and identification tasks. Fault isolation plays an equally important role as detection does in reality in the sense that it provides information on which parts of the system need to be substituted or changed to avoid serious damage to the system. Despite the importance of system reconfiguration problem, fault identification is not investigated in this thesis. The term FDI is commonly adopted to indicate fault detection and isolation in the literature as studied in this thesis.

1.2.2. Classification of Fault Diagnosis Methods

Generally, redundancy is required to detect and isolate faults in a system, which is used to make consistency checks between related variables. There exist two kinds of redundancy, namely: *hardware redundancy* and *analytical redundancy.* Hardware redundancy adopts extra control components. For instance, use multiple sensors, multiple components to measure and control a particular variable. Voting schemes are typically applied to a system using hardware redundancy to detect and locate the faulty sensor. This redundancy method is reliable and widely used in most industries. However, the main disadvantages of it are the need for extra components and the additional maintenance cost and extra space to accommodate the redundant equipments [11]. Since for some applications space is very limited, such as in a spacecraft, it is inconvenient and not practical to implement this method.

Recently, researchers have focused on the development of analytical redundancy approach. A diagnosis system applying analytical redundancy is called *model based diagnosis system*. In this scheme, the detection, isolation and determination of faults are achieved by a comparison between the available measurements of system components and a *priori* information represented by the system's mathematical model. Figure1.1 illustrates the concepts of hardware and analytical redundancy. It is clear that the difference of the two methods is the source of the expected value: one is from the model while another one is from redundant sensors.

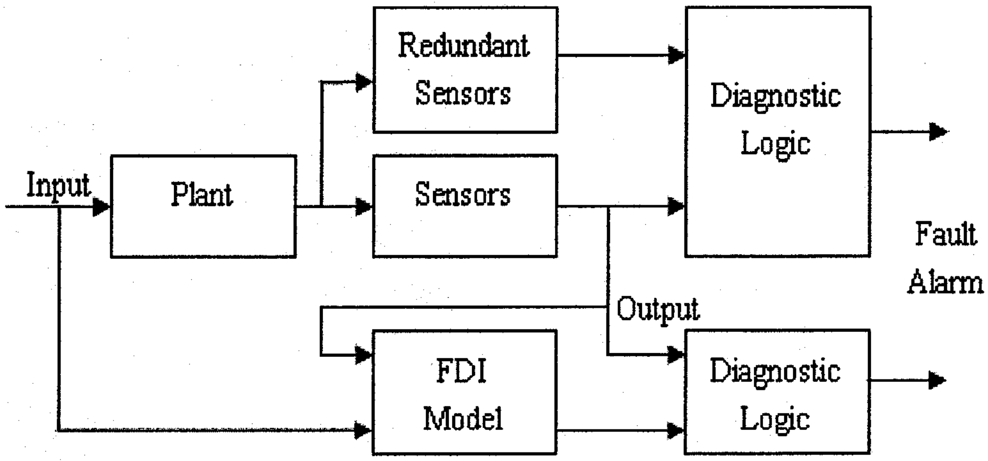

Figure 1.1 Comparisons between hardware and analytical redundancy schemes

The aim of model-based FDI scheme is to generate information about the fault such as the location and the timing by using measurements provided in the system. Generally, this method comprises of two main stages as illustrated in Figure 1.2.

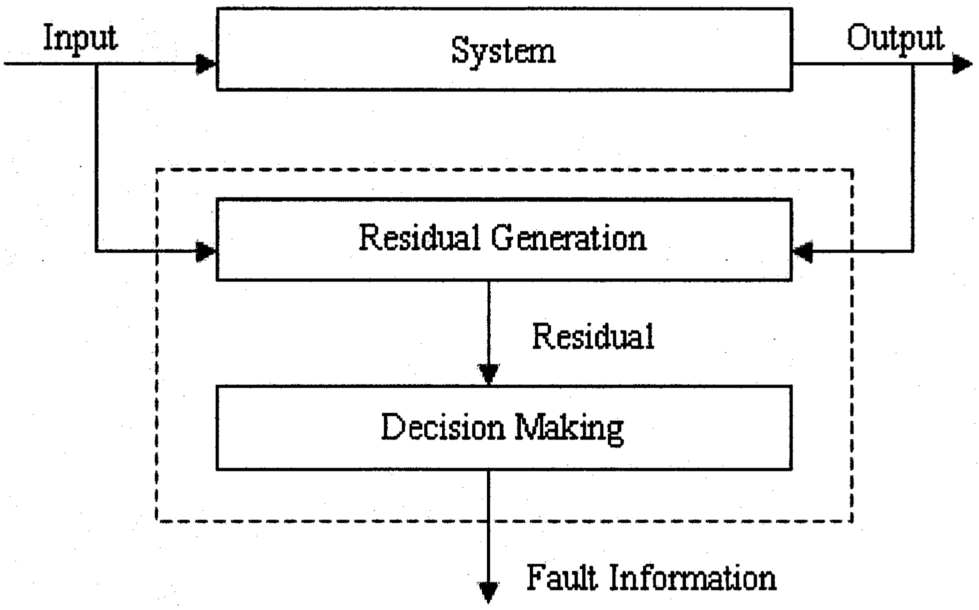

Figure 1.2 Model-based fault diagnosis

The purpose for residual generation is to process the measurable input and output of the monitored system to generate the indicated fault signal, termed *residual.* Clearly, the residual should be normally close to zero in fault free situation in ideal cases and should be distinguishably different from zero when a fault occurs. This residual generation stage attracts representative fault symptoms from the system and these symptoms are carried by the residual. For a good FDI scheme, the residual should contain as much fault information as possible, however at the same time be robust to system disturbances and noises. After the residual is generated, it will be sent to the decision-making stage for fault likelihood examinations. To determine whether a fault is occurred or not a decision rule needs to be applied here. In this process, the decision may be based on techniques such as threshold test, generalized likelihood ratio test, and sequential probability ratio test, etc. A threshold test is commonly used, which is based on either instantaneous values or moving averages of the residuals. Once the residual exceeds the threshold for a considerable amount of time, a fault might be judged to have occurred in the system. Generally, the stage of residual generation is the crucial part due to the fact that the decision can be made much easier based on a well generated residual than a poorly generated one.

Generally, model-based FDI methods can be divided into three main subcategories: *observer-based approaches*, *parity vector (relation) methods* and *parameter estimation methods* [10-13]. Details about their properties and their differences will be discussed in the following chapter.

With the development of the above techniques, another type of method known as *intelligent and learning-based methods* has attracted researchers' attention [14]. It can be broadly classified as a process–history based technique which needs to use a large amount of process historical data. They are typically made up of an antecedent part (series of events) and a consequence part, which maps these events to a known fault. Process history information enters the system in the form of antecedents and consequences. Thus these involve an explicit mapping of known symptoms to root causes [15].

Being classified as quantitative methods, neural network based methods for FDI scheme have received increasing attention in the past few years. Neural network and fuzzy logic techniques are being investigated as powerful modeling and decision making tools [16-20]. The use of these methods is considered as an important extension to the model-based FDI approaches. They have the potential to 'learn' the plant model from input-output data or 'learn' fault knowledge from past experiences, and they can be used as function approximators to construct the analytical model for residual generation, or as supervisory schemes to make the fault analysis decisions [17]. The nonlinear modeling capability of neural networks has been utilized for nonlinear fault diagnosis problems [18-22]. Meanwhile, expert systems and fuzzy logic have also been used in model based fault diagnosis [23-27].

1.2.3. Desirable Characteristics of a FDI System

Some desirable characteristics of a FDI system are listed here. Researchers can use them to benchmark various FDI approaches as well as use them as guidelines to design FDI systems [28]:

1. Early Detection and diagnosis: It is important and desirable for a FDI system that it is capable of performing early and accurate detection to avoid system breakdown.
2. Isolability: Isolability refers to the ability of FDI system to distinguish different faults and localize them.
3. Robustness: The FDI system should be robust to various noises, disturbances and uncertainties of the operating system. The threshold curves have to be chosen conservatively to avoid false alarms due to the noises and disturbances.
4. Novelty Identifiability: The FDI system should be able to recognize the occurrence of an unknown, novel fault and not misclassify it as normal operation.
5. Multiple Fault Identifiability: This is an important and difficult requirement for the FDI system due to the interacting nature of most faults.

6. Explanation Facility: The FDI system should provide explanations on how the fault originated and propagated throughout the system leading to the current situation.
7. Adaptability: The FDI system should be adaptable to the changes in external inputs or structural changes.
8. Reasonable Storage and Computational Requirement: There is a tradeoff between these requirements and a reasonable compromise is desirable.

1.2.4. Model-based FDI Methods

1.2.4.1. Observer Based Approaches

The basic idea for the observer-based approach is to estimate the outputs of the system from the available measurements by using either Luenberger observer in a deterministic environment or Kalman filters in a noisy environment. The output estimation error or its weighted value is served as the residual. The advantage of using the observer is the flexibility in selecting its gains which leads to a rich variety of FDI schemes [29-31].

In order to obtain the general structure of an observer, the discrete-time, time-invariant linear dynamic system under consideration is modeled in the state space representation as follows.

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ \quad y(k) = Cx(k) \end{cases} \tag{1.1}$$

where $u(k) \in \Re^r$, $x(k) \in \Re^n$ and $y(k) \in \Re^m$, and A, B and C are matrices with proper dimensions. Assuming that all the matrices are known, an observer can be applied to reconstruct the system states or variables based on the measured inputs and outputs $u(k)$ and $y(k)$, that is

$$\begin{cases} \hat{x}(k+1) = A\hat{x}(k) + Bu(k) + He(k) \\ \quad e(k) = y(k) - C\hat{x}(k) \end{cases} \tag{1.2}$$

The observer scheme described by equation (1.2) is depicted in Figure 1.3.

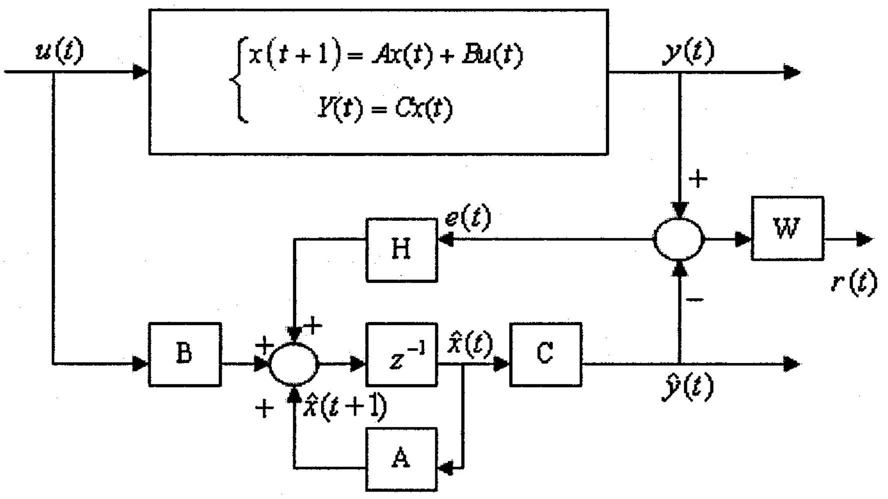

Figure 1.3 The process and the state observer architecture

For the state estimation error $e_x(k)$, we have

$$\begin{cases} e_x(k) = x(k) - \hat{x}(k) \\ e_x(k+1) = (A - HC)e_x(k) \end{cases} \tag{1.3}$$

The state error $e_x(k)$ (and the state error $e(k)$) is required to vanish asymptotically, that is $\lim_{k\to\infty} e_x(k) = 0$. (1.4)

This will be ensured by proper design of the observer gain H.

1.2.4.2. Parity Vector (Relation) Approaches

The parity vector (relation) approach is the oldest method which has been applied since the early development of FDI. Typically, there are two ways to arrange hardware redundancy. One is using sensors having identical or similar functions to measure the same variable, whereas the other approach is using dissimilar sensors to measure different variables but with their outputs being related to each other. The basic idea of parity vector method comes from the latter one that is to provide a proper check of the parity (consistency) of the measurements of the monitored system variables.

The measurement equation for a general problem of n-dimensional vector using m sensors may be expressed as:

$$y(k) = Cx(k) + f(k) + \xi(k) \tag{1.5}$$

where $x(k) \in \Re^n$ is the state vector, $y(k) \in \Re^m$ is the measurement vector, $f(k)$ is the vector of sensor faults, $\xi(k)$ is the noise vector and C is the matrix with proper size. If $m > n$ and $rank(C) = n$, this implies that the number of measurements is greater than the number of variables to be sensed, and inconsistency in the measurement data then can be used initially for fault detection and isolation. This technique has been successfully applied to fault diagnosis schemes for navigation systems where relations between gyroscope readings and accelerometer assemblies provide analytical forms of redundancy [32-33].

For FDI purposes, the vector $y(k)$ can be combined into a set of linearly independent parity equations to generate the parity vector (residual):

$$r(k) = Vy(k) \tag{1.6}$$

The residual generation scheme based on direct redundant measurements is shown in Figure 1.4.

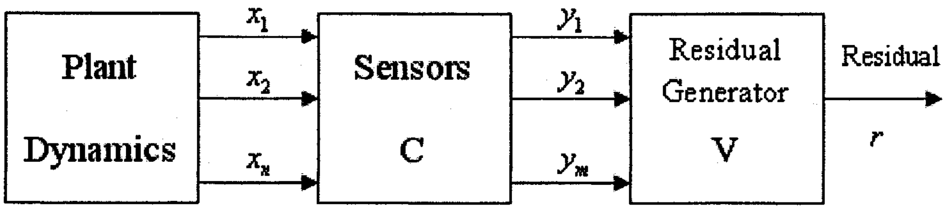

Figure 1.4 Residual generation via parallel redundancy

1.2.4.3. Parameter Estimation Approaches

Model-based FDI can also be achieved by the use of system identification techniques [32]. This approach is based on the assumption that the faults are reflected in the physical system parameters such as friction, mass, viscosity, inductance, capacitance, etc. The basic idea of the detection method is that the parameters of the actual process are repeatedly estimated on-line using well known parameter estimation methods and the results are compared with the parameters the reference model obtained initially under the faulty-free condition. Any substantial discrepancy indicates a fault. This approach normally uses the input-output mathematical model of a system in the following form:

$$y(k) = f(P, u(k)) \tag{1.7}$$

where P is the model coefficient vector which is directly related to physical parameters of the system. The function $f(\cdot,\cdot)$ can take either linear or non-linear forms.

To generate residuals using this approach, an on-line parameter identification algorithm should be used. If one has the estimate of the model coefficient at time step $k-1$ as $\hat{P}_{k-1}$, the residual can then be defined in either of the following ways:

$$\begin{cases} r(k) = \hat{P}_{k-1} - P_0 \\ r(k) = y(k) - f(\hat{P}_{k-1}, u(k)) \end{cases} \tag{1.8}$$

where P_0 is the normal model coefficient vector.

It is not easy to achieve fault isolation using the parameter estimation method because the parameters being identified are model parameters which cannot always be converted back to the system physical parameters. However, the faults are represented by variations in physical parameters. Moreover, [35] proposed an influence matrix approach to overcome the isolation difficulty. The idea is basically to identify the influence of each physical parameter on the residual.

1.2.5. Intelligent and Learning Based Methods

1.2.5.1. Fuzzy Logic Based Approaches

Fuzzy logic, as one kind of intelligent-based method, has received a lot of attention in FDI problems [25-27]. Obviously, the decision making stage of fault detection is a logical decision process that transforms quantitative knowledge (residual signals) into qualitative statements (normal or faulty). Due to the fact that the residual contains not only the information about faults but they are also contaminated by noises and disturbances, so that the residual will be non-zero even in fault-free cases. Therefore, it seems very natural to deal with the logical decision making problem with the aid of fuzzy logic since fuzzy logic shows advantages in handling such cases in uncertain and complicated situations based on incomplete information. The appealing feature of fuzzy logic is that it constitutes a powerful tool for modeling vague and imprecise facts and is therefore highly suited for the applications here.

1.2.5.2. Neural Network Based Approaches

A neural network is a processing system that consists of a number of highly interconnected units called neurons [16, 18-19]. Each neuron maps the mathematical function between its inputs and outputs and the neurons are interconnected by a large number of weighted link named weights. The inputs are connected to either the inputs of the system or the outputs of the other neurons in the system. The output of one neuron affects the outputs of other neurons and all neurons connected together can perform complex processes. The mathematical model used in traditional FDI scheme can be sensitive to modeling errors, noise and disturbance. However, no mathematical model is needed when a system implement a neural network. Once the output of a system is known, neural networks can be trained to represent the relationships between inputs and outputs of a system. A well trained neural network can generalize when presented with inputs not appearing in the training data and it also capable to make intelligent decisions in cases of noisy or corrupted data.

1.3. Research Motivation

Instead of using mathematical models, a neural network can be used to generate residual signal as well as to isolate faults and to provide more reliable and practical applications for a FDI scheme. The main feature of neural networks is their learning ability. They are capable of learning from examples. Therefore, they can be trained to represent relationships between the faulty conditions and the residual data. Different types of neural networks have been introduced successfully as an FDI scheme for satellite attitude control [20-22]. The motivation for this thesis is to explore the possibility of a FDI scheme with the use of an adaptive time delay recurrent neural network architecture [36].

1.4. Research Objectives and Contributions of the Thesis

According to the motivation stated in the previous section, the objective of this thesis is to develop a practical scheme based on an adaptive time delay recurrent neural network for fault detection and isolation in reaction wheels of satellites. Assuming the satellite changes its angle from 0 degree (deg) and the attitude range change is restricted from 0 degree to 10 degree, the research goal is to determine whether the proposed recurrent neural network is capable of detecting the faults that has occurred in reaction wheels or not under these circumstances. If it is feasible, a parallel bank of filters can be established to construct the entire FDI scheme. For instance, filter A is used to detect faults when the satellite attitude changes from 0 degree to 10 degree, filter B is used for the range 10 degree to 20 degree, and so on. Therefore, the entire FDI scheme for the satellite can be achieved by these banks of neural network filters.

In order to achieve these objectives, three neural networks are employed to supervise the dynamics of the reaction wheels on the three axes separately and independently. The neural network architecture applied in this thesis is the adaptive time delay recurrent neural network which represents accurately the nonlinear relationship between the wheel torque signal and the input reference command signal to the wheel.

1.5. Research Methodology

On each axis, a well trained neural network is employed to provide the estimated reaction torque signal. By comparing the difference between the estimated signals and the actual reaction torques, networks are able of identifying the existence of faults in the system.

The algorithm developed in this thesis consists of three stages:

1. Thresholds Generation: The threshold signal is generated in recall phase in this thesis. First of all, a well trained time delay adaptive recurrent neural network is selected (trained) to model the dynamics of the reaction wheel under fault free operation on one axis. The network has two inputs: one is the torque command voltage signal (TCV) and the other is the one step delay of the output of the network which is the estimated reaction torque. The difference between the actual and the estimated reaction torque signals are passed through a moving average filter to generate the residual signal error for a particular setpoint. This procedure needs to be repeated for obtaining the residual signals for different setpoints. Next, a residual error set including residual curves for these individual setpoints is set up and the mean value and the standard deviation of the set need to be calculated. Finally, the suitable parameters for ensuring that the threshold curve that is capable of providing a false alarm free detection is implemented.
2. Threshold Testing and Fault Detection: For fault testing, the residual signals generated in the first stage need to be compared with the corresponding threshold curve. If the residual error exceeds the threshold curve for a considerable period of time, say 20 minutes in this thesis, a fault can be considered as being present in that corresponding reaction wheel.

3. Fault Isolation: By examining the threshold testing results of each axis independently, one is able to provide the information on fault detection such as fault location, fault occurrence time and even fault magnitude symptoms such as small, medium, large etc.

In order to perform the above goals in this thesis, a simulation model of the entire attitude control system and the reaction wheel are constructed with the aid of MATLAB (Version 7.20) and SIMULINK. The data collection, preprocessing, neural network development and implementation as well as the simulation results on assessment and comparative studies are conducted in MATLAB and its associated toolboxes.

1.6. Outline of the Thesis

In chapter 2, a brief overview of the outline of the attitude control system will be provided. The details about dynamic characteristics of a reaction wheel and its MATLAB model are introduced.

In chapter 3, after a brief introduction about neural networks, a time delay adaptive recurrent neural network FDI scheme will be developed. The choice of neural network parameters will also be investigated.

The simulation results corresponding to the time delay adaptive recurrent neural network FDI scheme will be conducted in chapter 4. The general fault detection results, robustness and isolation tests are based on individual setpoint changes. Two fault patterns are randomly generated to mimic the wheel operating under faulty condition in real life. Six case studies are conducted to facilitate and achieve good understanding about the nature of this neural network to provide:

1. Multiple fault detection results under continuous setpoint changes.
2. Multiple fault recovery results under continuous setpoint changes.
3. Robustness for the network operating under continuous setpoint changes.
4. Isolation test for network operating under continuous setpoint changes.
5. Fault detection and recovery on one axis under continuous setpoint changes with the other two axes being also faulty.
6. Fault detection and recovery detection by using different definition of the residual error under continuous setpoint changes.

In chapter 5, the summary of our observations based on the above case studies is detailed as well as some suggestions for future work are stated.

Chapter 2

Introduction to the Attitude Control System (ACS)

2.1. Mission Specifications

The entire research is based on a hypothetical satellite which is utilized to investigate the attitude dynamics and control of a spacecraft. The satellite is launched into a 700 km circular Low Earth Orbit, sun-synchronous (98.2^{o}) orbit. With a velocity of approximately 7.5 km/s, the orbit period is 98.8 minutes. Orbit selection is chosen based on science requirements, orbit lifetime, ground station coverage, and radiation concerns.

The satellite has three axes and each of them has its independent control loop for position control. The stabilization of the satellite is achieved by using a 3-wheel assembly, with three active and one being redundant. Each reaction wheel is aligned with each axis separately. Three separate PID control loops are designed to meet the attitude pointing accuracy specification which is within 0.2^{o} in all the three axes. The details regarding a single control loop design will be illustrated in the following sub-section.

2.2. Single Axis Attitude Control Using PID

The block diagram of the attitude control loop for a single axis is shown in Figure 2.1. The control loop contains four components: sensor, controller, actuator, and satellite body. In practice, some noises are added on the sensors as well as external disturbances are imposed on the body.

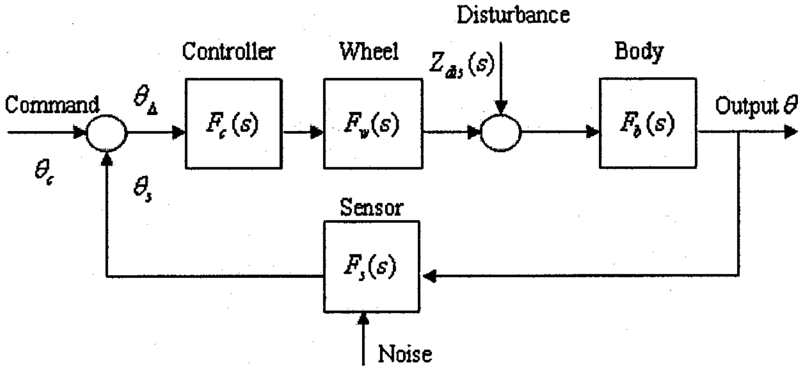

Figure 2.1 Single-axis attitude control block diagram [8]

The transfer function of the system is given by

$$\theta = \theta_\Delta F_c F_w F_b + Z(s) F_b \tag{2.1}$$

where θ is the controlled attitude angle, and θ_c is the command angle,

$$\theta_\Delta = \theta_c - \theta_s \tag{2.2}$$

$$\theta_s = \theta F_s \tag{2.3}$$

Combining these three equations and setting the disturbance $Z(s)$ to zero, the transfer function of the closed-loop system is obtained as:

$$F(s) = \frac{\theta(s)}{\theta_c(s)} = \frac{F_c(s)F_w(s)F_b(s)}{1 + F_c(s)F_w(s)F_b(s)F_s(s)} \tag{2.4}$$

By assuming that the control command $\theta_c(s) = 0$, the disturbance transfer function can be derived as:

$$D(s) = \frac{\theta(s)}{Z_{dis}(s)} = \frac{F_b(s)}{1 + F_c(s)F_w(s)F_b(s)F_s(s)} \tag{2.5}$$

2.2.1. Sensors

Table 2.1 shows a summary of typical sensors that are used in aerospace industry with their performance and physical characteristics.

Table 2.1 Typical attitude determination and control subsystem (ADCS) sensors [3]

Sensor	Typical Performance Range	Wt Range (kg)	Power (w)
Inertial Measurement Unit (Gyros & Accelerometers)	Gyro Drift rate = 0.003 deg/hr to 1 deg/hr, accel. Linearity = 1 to $5\times10^{-6} g/g^2$ over range of 20 to 60 g	1 to 15	10 to 200
Sun Sensors	Accuracy = 0.005 deg to 3 deg	0.1 to 2	0 to 3
Star Sensors (Scanners & Mappers)	Attitude accuracy = 1 arc sec to 1 arc min; 0.0003 deg to 0.01 deg	2 to 5	5 to 20
Horizon Sensors ● Scanner/Pipper ● Fixed Head (Static)	Attitude accuracy: 0.1deg to 1 deg (LEO) < 0.1 deg to 0.25 deg	 1 to 3 0.5 to 3.5	 5 to 10 0.3 to 5
Magnetometer	Attitude accuracy = 0.5 deg to 3 deg	0.3 to 1.2	<1

The details for the sensors are introduced below [3]:

Sun sensors are visible-light detectors which measure one or two angles between their mounting base and incident sunlight. They are widely used for instance in the normal attitude determination system, the initial acquisition or failure recovery system, or in an independent solar array orientation system. Moreover, they are accurate and reliable. Their accuracy feature is less than 0.01deg which is good but it is not guaranteed achievable forever. Despite requiring clear fields of view, they have become the common choice. In practice, in order to overcome their limitations, they are usually fixed near the ends of the vehicle to achieve a visible field of view.

Star sensors are the most popular sensors for high-accuracy missions. They are of two types: scanners or trackers. A scanner's field of view has multiple slits which observe the passage of stars. The attitude of vehicle is derived based on several star crossings. Scanners are used on spinning spacecraft while trackers are used on 3-axis attitude stabilized spacecraft to track one or more stars for obtaining two or three axes attitude information. They are not only capable of tracking the stars as bright spots, but also identify the star pattern which they are viewing, as well as report the sensor's orientation compared to an inertial reference. For the highest accuracy missions, a combination of star trackers and gyros are usually used to balance the cost consideration. They complement each other in the sense that the gyros are used for initial stabilization, during periods of sun or moon interference in the trackers while the star trackers are used to provide a high-accuracy, low frequency, external reference unavailable to the gyros.

Horizon sensors are infrared devices that detect the contrast between the cold of deep space and the heat of the Earth's atmosphere (about 30 km above the surface in the sensed band). They provide Earth-relative information directly for Earth-pointing spacecraft, which may simplify onboard processing. Horizon crossing indicators that are also called pippers, are used on spinning spacecraft to measure Earth phase and chord angles which, together with orbit and mounting geometry, define two angles to the Earth (undershoot) vector. Scanning horizon sensors use a rotating mirror or lens to replace (or augment) the spinning spacecraft body. In order to improve performance and redundancy, they are usually used in pairs. For circular orbits, the staring sensors work best.

Magnetometers are simple, reliable, lightweight sensors that measure both the direction and size of the Earth's magnetic field. When compared to the Earth's known field, their output can be used to establish the spacecraft's attitude. However, their accuracy is not as good as that of star or horizon references, therefore their data are often combined with the data from Sun or horizon sensors to improve their accuracy. A magnetometer is used to control the polarity of the torque output when a spacecraft that is using magnetic torques passes through the magnetic-field reversals during each orbit.

GPS receivers are commonly known as high-accuracy navigation devices. They have been used for attitude determination by employing the differential signals from antennas on a spacecraft. Generally, this kind of sensor can be used not only as a back–up sensor but also to provide the promise of lower cost and weight for Low Earth Orbit (LEO) mission implementations.

Gyroscopes are inertial sensors which measure the speed or angular rotation from an initial reference without any knowledge of an external or an absolute reference frame. Due to the lack of an external, absolute reference, they are often used in combination with other external references such as Sun sensors or star sensors for precision attitude sensing of spacecraft. When used with external references such as star trackers, gyros enable one to offer smoothing and higher frequency information while the star trackers offer low frequency, absolute orientation information. Individual gyros provide one or two axes information, and are often grouped together as an Inertial Reference Unit (IRU) for three full axes. IRUs with accelerometers added for position or velocity sensing are called Inertial Measurement Units (IMUs).

For our satellite used in this thesis, the Earth/Horizon sensor is implemented to obtain the pitch and roll angles. A realistic sensor is more likely to be represented by a block which represents a delay between the input and output signals, namely

$$F_s = \frac{K_s}{1+T_s s} \tag{2.6}$$

However, in order to simplify our problem in this thesis, an ideal sensor is represented by a simple block whose transfer function is assumed and expressed as:

$$F_s = K_s = 1 \tag{2.7}$$

Therefore, the block diagram of the closed-loop can be simplified as shown below:

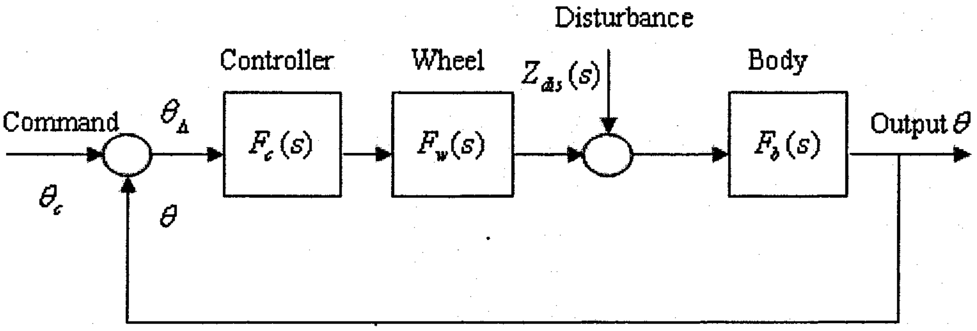

Figure 2.2 A simplified single-axis attitude control block diagram

2.2.2. Actuators

The actuators used in this thesis for attitude control are three-reaction wheel assemble consisting of three active reaction wheels on each axis of the spacecraft and one redundant wheel. Reaction wheels are momentum exchange devices which provide reaction torque to a spacecraft and store angular momentum [8]. The mathematical model of reaction wheel is deduced from Newton's second law. Additional terms which are a function of the temperature and the bus voltage are included here to assess the performance beyond the normal speed ranges. Moreover, the disturbance and noise terms are also factors that are considered for evaluating the operation. A typical reaction wheel is driven by an inertial brushless DC motor, which includes a rotating flywheel, typically suspended on ball bearings.

The fundamental block diagram of a nearly ideal reaction wheel model [8] is shown in Figure 2.3.

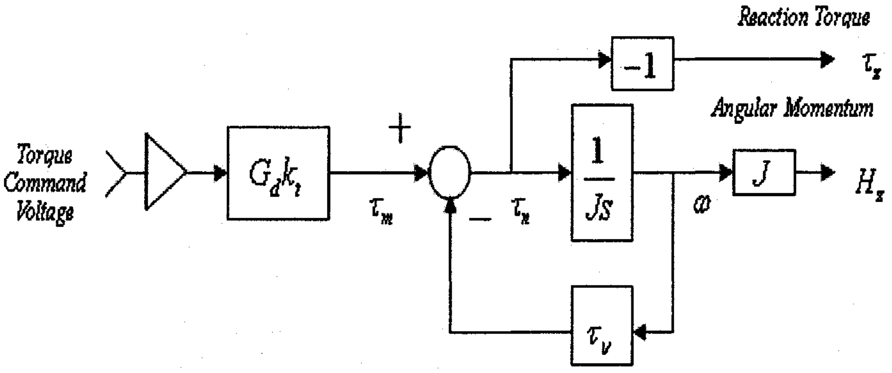

Figure 2.3 A nearly ideal reaction wheel model block diagram [8]

The input signal of the wheel is a torque command voltage which controls the motor current directly and then controls the motor reaction torque indirectly. The angular momentum stored in the flywheel H_z is the product of flywheel inertial and the angular velocity of the wheel, that is

$$H_z = J\omega \tag{2.8}$$

According to the Newton's third law, the reaction torque generated to the spacecraft is opposite to the net torque, namely

$$\tau_z = -\tau_n \tag{2.9}$$

According to the Newton's second law, the net torque can also be deduced from the rate of change of the angular momentum as

$$\tau_n = \frac{\partial H_z}{\partial t} = \frac{\partial (J\omega)}{\partial t} = J\frac{\partial \omega}{\partial t} \tag{2.10}$$

In order to simplify the problem, the coulomb friction is ignored and only viscous friction is considered here. Therefore,

$$\tau_n = \tau_m - \tau_v \omega \tag{2.11}$$

$$\tau_m = T_c G_d k_t \tag{2.12}$$

where T_c is the torque command voltage.

From equations (2.8) to (2.12) and with the use of the Laplace transform, the wheel transfer function from the torque command voltage to the reaction torque is obtained as,

$$F_w(s) = \frac{\tau_z(s)}{T_c(s)} = \frac{-G_d k_t Js}{Js + \tau_v} \tag{2.13}$$

On the other hand, the block diagram shown in Figure 2.4 is a detailed and a high fidelity diagram of a reaction wheel [8]. The diagram illustrates the fundamental relationships that exist for a high fidelity mathematical model of a reaction wheel system. This detailed reaction wheel model is utilized in this thesis to substitute for the real wheel. The diagram consists of five sub-blocks, namely: motor torque control, speed limiter, EMF torque limiting, motor disturbances and bearing friction and disturbances.

Further information about each sub function blocks will be given in details in the following sub-sections.

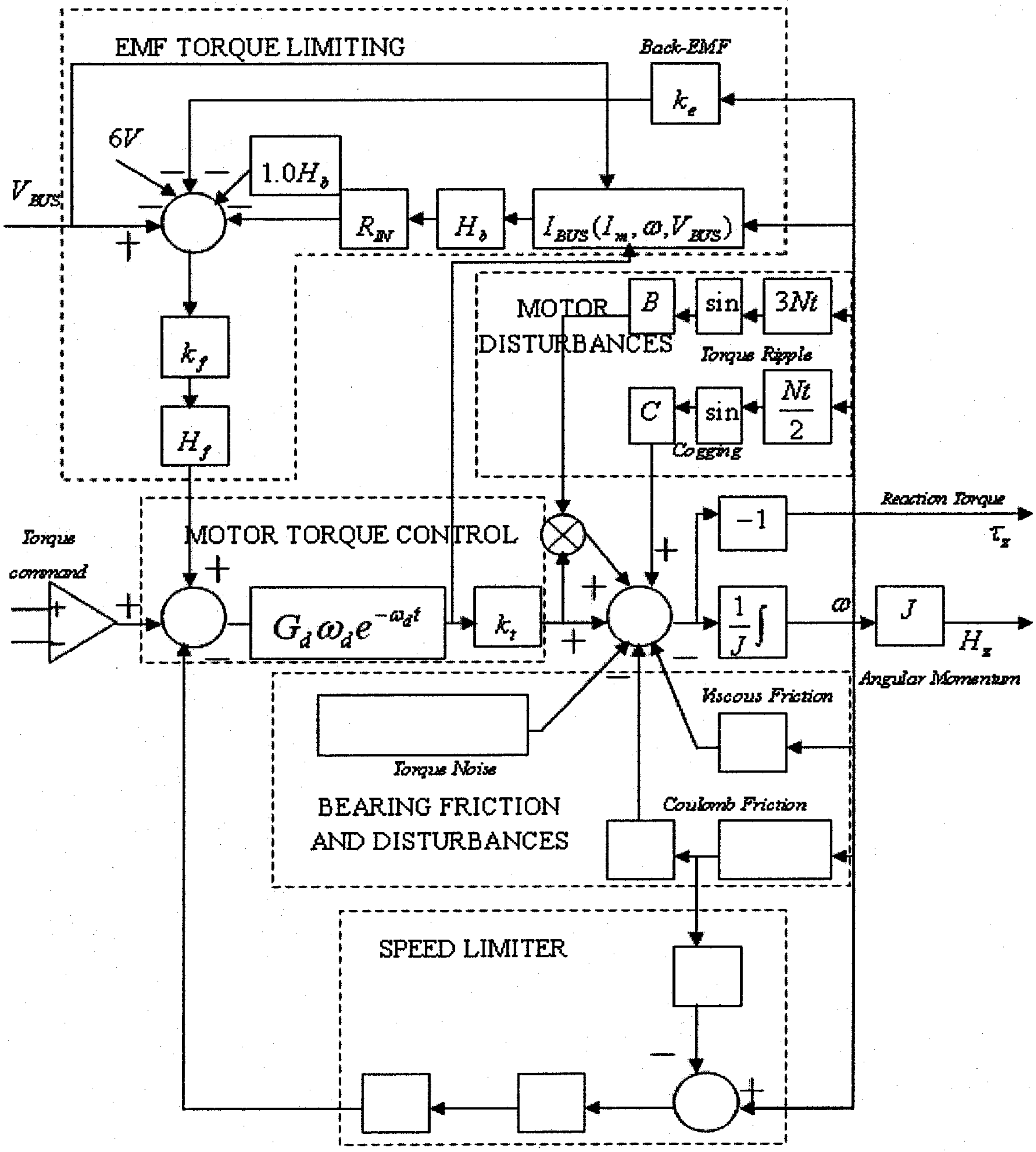

Figure 2.4 A detailed and high fidelity reaction wheel block diagram [8]

The reaction wheel employed herein is the ITHACO's standard Type A reaction wheel. The typical parameters used in this diagram are provided in Table 2.2. [8].

Table 2.2 A typical parameter values of Type A reaction wheels [8]

Variable	Nomenclature	Units	Type A RWA
G_d	Diver Gain	A/V	0.19
ω_d	Driver Bandwidth(-3 dB)	$rad/\sec$	2000
k_t	Motor Torque Constant	$N-m/A$	0.029
k_e	Motor Back-EMF	$V/rad/\sec$	0.029
k_s	Overspeed Circuit Gain	$V/rad/\sec$	95
ω_s	Overspeed Circuit Threshold	$rad/\sec$	690
τ_c	Coulomb Friction	$N-m$	0.002
J	Flywheel Inertial	$N-m-s^2$	0.0077
N	Number of Motor Poles	$-$	36
B	Motor Torque Ripple Coefficient	$-$	0.22
C	Cogging Torque Amplitude	$N-m$	0
R_{IN}	Input Resistance	Ω	2.0
P_q	Quiescent Power	W	3.0
R_B	Bridge Resistance	Ω	2.0
	Torque Command Range	V	5
	Torque Command Scale Factor	$N-m/V$	0.0055
K_f	Voltage Feedback Gain	V/V	0.5
θ_a	Torque Noise Angle Deviation	rad	0.05
ω_a	Torque Noise High Pass Filter Frequency	$rad/\sec$	0.2

Motor Torque Control Block

The motor driver is a voltage controlled current source with a gain G_d. The motor current I_m is directly proportional to the torque command voltage. This is then passed through a torque constant k_t to generate a motor torque τ_m which is proportional to I_m. In this thesis, the torque command voltage is chosen in the range $[-5V, 5V]$.

Speed Limiter Block

The purpose of utilizing a speed limiter circuit is to prevent the flywheel reaching unsafe speeds. The limiter circuit uses an analog tachometer circuit to sense wheel speed ω which is then compared with an established threshold ω_s. Once the wheel speed ω exceed ω_s, the circuit provides a high-gain negative feedback k_s to the torque command. In order to trigger the negative feedback, a heavy-side function H_s is employed in this diagram. This function is given according to

$$\begin{cases} H_s = 0 for|\omega| < \omega_s; \\ H_s = 1 for|\omega| \geq \omega_s; \end{cases} \tag{2.14}$$

EMF Torque Limiting Block

The increasing back-EMF k_e of the motor may cause the motor torque to become bounded at high flywheel speed when the motor operate in low bus voltage conditions. This situation influence the motor in two ways: on one hand it eliminates the voltage headroom ultimately. On the other hand, if the back-EMF is kept on increasing, it will cause the pulse-width-modulation of the motor to saturate, which reduces the torque capacity. Moreover, since the motor torque is coupled directly to the bus voltage, from the disturbance point of view, any fluctuations in bus voltage will be sensed as torque disturbances.

The back-EMF limiting is coupled to power consumption by voltage drops in the input filter due to the bus current level. The voltage drop is the product of the bus current I_{BUS}, and the filter input resistance R_{IN}. An approximate power consumption model is adopted below:

$$P_{INPUT} = \frac{V_{BUS}}{V_{BUS} - 1}\left[\frac{\tau_m^{\ 2}}{k_t^2}R_B + \frac{0.04|\tau_m|V_{BUS}}{K_t} + P_q + \omega\tau_m \frac{k_e}{k_t}\right] \tag{2.15}$$

where P_{INPUT} is the power consumption. By dividing it with V_{BUS} , the bus current I_{BUS} is deduced as follows

$$I_{BUS} = \frac{1}{V_{BUS} - 1}\left[\frac{\tau_m^{\ 2}}{k_t^2}R_B + \frac{0.04|\tau_m|V_{BUS}}{K_t} + P_q + \omega\tau_m \frac{k_e}{k_t}\right] \tag{2.16}$$

Combining the above relationships together,

$$k_t = \frac{\tau_m}{I_m} \tag{2.17}$$

and equation (2.16) and (2.17) yield:

$$I_{BUS} = \frac{1}{V_{BUS} - 1}\left[I_m^{\ 2}R_B + 0.04|I_m|V_{BUS} + P_q + \omega I_m k_e\right] \tag{2.18}$$

which indicates that the bus current is dependent on the motor current I_m, wheel speed ω, and bus voltage V_{BUS}.

In order to eliminate the voltage drop when the power is not being drawn form the bus, for instance, during a deceleration when energy is being deposed from the flywheel, a heavy-side function H_b is introduced in the block diagram. Moreover, a reverse polarity protection diode drop of $1V$ is also dependent on H_b.

$$\begin{cases} H_b(I) = 1 forI > 0; \\ H_b(I) = 0 forI \le 0; \end{cases} \tag{2.19}$$

Another heavy-side function H_f is applied in this sub-block and is defined as:

$$\begin{cases} H_f(V) = 0 for V > 0; \\ H_f(V) = 1 for V \leq 0; \end{cases} \qquad (2.20)$$

Motor Disturbances Block

The high frequency disturbances are usually caused by the torque motor in a reaction wheel due to the motor excitation and the magnetic construction. The reaction wheel adopted here use brushless DC motors, which exhibit torque ripple at the commutation frequency, and cogging at a frequency corresponding to the number of motor poles and rate of rotation.

Torque ripple is the amount of variation in the motor torque caused by the commutation method and the shape of the back-EMF. In cases where discrete commutation is implemented with sinusoidal back-EMF, such as in ITHACO's reaction wheels, the torque ripple in a perfectly aligned motor is classically 13.3% overshoot-to-overshoot of the commanded motor torque, or about 7% rms. For simplicity, the block diagram approximates the ripple wave as a pure sine wave although its actual shape is a truncated rectified sine wave. The amount of torque ripple mainly depends on the torque ripple frequency which is essentially the commutation rate.

Cogging is a disturbance torque which is always present in a conventional brushless DC motor, which is ignored here to simplify the system model.

Bearing Friction and Disturbances Block

The friction in a reaction wheel can be cataloged as viscous friction and coulomb friction. The viscous friction τ_v that is varied with speed and temperature is generated in the bearings due to the bearing lubricant. Since the viscosity is temperature dependent, the lubricant has a strong sensitivity to temperature. For the ITHACO's Tpye A reaction wheels used here, the viscous friction can be approximately modeled as [8]:

$$\tau_v = \left(0.049 - \frac{0.0002}{^oC}\left(T + 30^oC\right)\right) \times 10^{-2} \frac{N-m}{rad/\sec} \tag{2.21}$$

The coulomb friction τ_c is caused by rolling friction within the bearings. In a reaction wheel application with direction reversals, the bearing stiction will cause a disturbance which is characterized by a torque discontinuity as the wheel passes through zero speed. The rolling friction is defined as the smallest amount of torque, which if applied continuously, will keep the fly wheel rotating. The breakaway torque is the smallest amount of torque which will start the flywheel from a stalled condition. The resulting torque discontinuity for crossing through zero speed is therefore the sum of the rolling friction and the breakaway torque. In most cases, as well as in this thesis, the coulomb friction can be assumed as twice of the rolling friction, neglecting the breakaway torque difference [8].

Torque noise is the very low frequency torque variation from the bearings caused by lubricant dynamics, which can be specified as a deviation from the ideal location of motor at any constant speed. In the block diagram of the wheel, the torque noise is assumed as a sine wave at the high pass filter frequency and is approximately modeled as:

$$\tau_a = J\theta_a\omega_a^2 \sin\omega_a t \tag{2.22}$$

where J is the flywheel inertial, θ_a is the torque noise angle and ω_a is the torque noise high pass filter frequency.

2.2.3. Body Dynamics

According to Newton's second law, $\vec{\tau} = \dfrac{d\vec{H}}{dt}$. If the body reference system has angular velocity $\vec{\omega}$ as observed from the inertial reference frame, then the torque need to revised to

$$\vec{\tau} = \dot{\vec{H}} + \vec{\omega} \times \vec{H} \tag{2.23}$$

Recalling that

$$\vec{\omega} \times \vec{H} = \left(\omega_y H_z - \omega_z H_y\right) i + \left(\omega_z H_x - \omega_x H_z\right) j + \left(\omega_x H_y - \omega_y H_x\right) k \tag{2.24}$$

the Euler's moment equation can be deduced from the above two equations according to

$$\begin{cases} \tau_x = \dot{H}_x + \omega_y H_z - \omega_z H_y \\ \tau_y = \dot{H}_y + \omega_z H_x - \omega_x H_z \\ \tau_z = \dot{H}_z + \omega_x H_y - \omega_y H_z \end{cases} \tag{2.25}$$

Relating the angular momentum components to the angular velocity components, yields

$$\begin{cases} H_x = I_{xx}\omega_x - I_{xy}\omega_y - I_{xz}\omega_z \\ H_y = I_{yy}\omega_y - I_{xy}\omega_x - I_{yz}\omega_z \\ H_z = I_{zz}\omega_z - I_{xz}\omega_z - I_{yz}\omega_y \end{cases} \tag{2.26}$$

By assuming that the spacecraft body frame aligned with the principle axes, where the products of inertias are zero, result in

$$\begin{cases} \tau_x = \dot{\omega}_x I_{xx} + \omega_y \omega_z \left(I_{zz} - I_{yy}\right) \\ \tau_y = \dot{\omega}_y I_{yy} + \omega_z \omega_x \left(I_{xx} - I_{zz}\right) \\ \tau_z = \dot{\omega}_z I_{zz} + \omega_x \omega_y \left(I_{yy} - I_{xx}\right) \end{cases} \tag{2.27}$$

where x, y, z now represent the principal axes of inertia.

Equation (2.27) is utilized for model construction in this thesis. In order to simplify the PID controller design, a rigid and decoupled system is considered here, in which the coupling effects are ignored, namely

$$\begin{cases} \tau_x = \dot{\omega}_x I_{xx} \\ \tau_y = \dot{\omega}_y I_{yy} \\ \tau_z = \dot{\omega}_z I_{zz} \end{cases} \tag{2.28}$$

Therefore, the transfer function of the body dynamics for a single axis may be expressed as,

$$F_b = \frac{1}{Is^2} \tag{2.29}$$

where I is the inertia of the related axis.

2.2.4. External Disturbance Torques

Operating in space, the spacecraft experiences many types of external environmental disturbance torques. Four of them are mainly consider here, namely: gravitation torque, solar pressure torque, magnetic torque and aerodynamic torque. They are described in details below.

The gravitational torque arises since the gravitational force varies over the unsymmetrical mass distribution of the satellite body. Since the radius vector from the center of Earth to the center of the mass of the spacecraft varies in the body frame-of-reference, the gravity gradient torque varies throughout the orbit. In this thesis, the maximum gravity gradient torque during the whole period is assumed to be bounded by:

$$DIS_{gg} = 1.8 \times 10^{-6} N - m \tag{2.30}$$

For solar pressure torque, it is generated by an accumulative force imparted by the Sun on the spacecraft body orbiting the Earth and the offset of the spacecraft optical center from the spacecraft mass center. This pressure is highly dependent on the surface of the spacecraft. In this thesis, the worst case of solar pressure torque is assumed as [3]:

$$DIS_{sp} = 6.6 \times 10^{-6} N - m \tag{2.31}$$

Due to the inaccuracy of the spacecraft's magnetic dipole vector and the current loops within the spacecraft, it is difficult to determine the Earth's magnetic torque accurately all the time. Generally, a dipole model is used to estimate the value of this torque. Moreover, in this case, the maximum value of this torque is assumed to be bounded by [3]:

$$DIS_{mf} = 4.5 \times 10^{-5} N - m \tag{2.32}$$

The aerodynamics disturbance torque is due to the accumulative force imparted by the molecules found in the upper atmosphere and the offset of the spacecraft aerodynamic center from the spacecraft mass center. This torque is also related to the atmospheric density which significantly varies with solar activity. For a preliminary design, a rough estimate of the maximum value for this aerodynamics disturbance torque is used as [3]:

$$DIS_{ad} = 3.4 \times 10^{-6} N - m \tag{2.33}$$

For simplicity, the maximum external disturbance torque is assumed as the sum of the above four torques, namely

$$DIS = DIS_{gg} + DIS_{sp} + DIS_{mf} + DIS_{ad} = 5.68 \times 10^{-5} N - m \tag{2.34}$$

For the construction of the ACS model in this thesis, it is assumed that the external disturbance torque is a normally distributed random signal with zero mean and variance of $DIS^2 = \left(5.68\times10^{-5}\right)^2$. However, for the PID controller design, it is assumed as a step function with the step value of $5.68\times10^{-5} N-m$, which is the maximum disturbance torque discussed above. Under this assumption, the Laplace transform of the external disturbance torque becomes:

$$Z_{dis} = \frac{DIS}{s} = \frac{5.68\times10^{-5}}{s} \tag{2.35}$$

2.2.5. PID Controller Design

Let us assume that the design specifications are detailed as follows:

1. The desired maximum overshoot is less than 20%,
2. The 5% settling time is 30 seconds, and
3. The steady state error is 0.2^o.

A PID controller is designed as the control component in each axis. The PID controller can be represented as

$$F_c = K_d\left(s+z_{c1}\right)\frac{\left(s+z_{c2}\right)}{s} \tag{2.36}$$

For the sake of convenience, the transfer functions of the each block in the control loop in Figure 2.1 are rewritten below

$$\begin{cases} F_c = K_d\left(s+z_{c1}\right)\dfrac{\left(s+z_{c2}\right)}{s} \\ F_w = \dfrac{-G_d k_t Js}{Js+\tau_v} \\ F_s = 1 \\ F_b = \dfrac{1}{Is^2} \end{cases} \tag{2.37}$$

Therefore, the transfer function of the open-loop system can be expressed as:

$$G_{ol} = \frac{-G_d k_t J K_d (s+z_{c1})(s+z_{c2})}{(Js+\tau_v)Is^2} \tag{2.38}$$

Obviously, the system is a type 2 system which results in zero steady state error for step function input. Therefore, the steady state error corresponding to this system is caused by the external disturbance torques. The disturbance transfer function is given by

$$D(s) = \frac{\theta(s)}{Z_{dis}(s)} = \frac{F_b(s)}{1+F_c(s)F_w(s)F_b(s)F_s(s)} \tag{2.39}$$

By using the final value theorem, the steady state error of the system is obtained as

$$Error_{ss} = \lim_{s\to 0} sD(s)Z_{dis}(s) = \frac{sZ_{dis}(s)F_b(s)}{1+F_s(s)F_c(s)F_w(s)F_b(s)} \tag{2.40}$$

Assuming that the maximum disturbance torque is applied to the spacecraft as discussed above corresponding to equation (2.35), we get

$$Error_{ss} = \frac{\tau_v(5.68\times 10^{-5})}{G_d k_t J z_{c1} z_{c2} K_d} \tag{2.41}$$

Given that the steady state accuracy requirement is 0.2^o, we have

$$Error_{ss} < 0.2^o = \frac{0.2\pi}{180} \tag{2.42}$$

Hence,

$$K_d > -\frac{\tau_v(5.68\times 10^{-5})}{G_d k_t J z_{c1} z_{c2} \frac{0.2\pi}{180}} \tag{2.43}$$

Based on the above equation, the value of K_d in the PID controller is determined.

2.3. Three Axes Attitude Control System

Three PID controllers are utilized independently to meet the design specifications by following the design methodology stated in the previous section. Figure 2.5 illustrates the three axes attitude control system block diagram. The net torque applied on one axis is affected by the other two axes. Equation (2.27) repeated and revised below for convenience represents the coupled effects of the other two axes, which is selected here for constructing the controllers.

$$\begin{cases} \tau_x + \tau_{disx} = \dot{\omega}_x I_{xx} + \omega_y \omega_z (I_{zz} - I_{yy}) \\ \tau_y + \tau_{disy} = \dot{\omega}_y I_{yy} + \omega_z \omega_x (I_{xx} - I_{zz}) \\ \tau_z + \tau_{disz} = \dot{\omega}_z I_{zz} + \omega_x \omega_y (I_{yy} - I_{xx}) \end{cases} \qquad \text{(Revised 2.27)}$$

where τ_{dis} represents the effects of environmental and internal noise and disturbances.

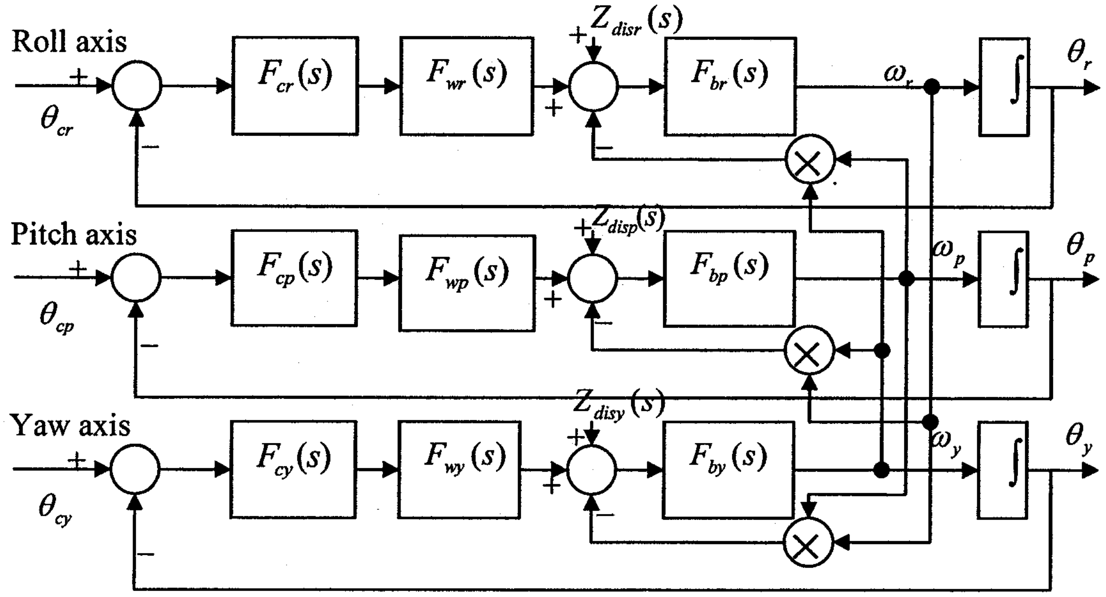

Figure 2.5 A three axes ACS block diagram

2.4. Initial Conditions and Parameters in Normal Satellite Operation

The initial conditions and parameters corresponding to the healthy operation of the satellite are summarized in Table 2.3.

Table 2.3 The parameter values in normal operating condition

Nomenclature	Units	Normal Value or Range
Bus Voltage	V	8
Temperature	^0C	23
Initial Body Attitude for One Axis	deg	0
Initial Body Rate for One Axis	$rad/\sec$	-1.0×10^{-4} to 1.0×10^{-4}
Initial Wheel Speed	$rad/\sec$	20 to 30
Setpoint Change for One Axis	deg	0 to 10
S/C Inertial Matrix	Kgm^2	$\begin{pmatrix} 17 & 0 & 0 \\ 0 & 15 & 0 \\ 0 & 0 & 22 \end{pmatrix}$

2.5. The Performance of the PID Controller

Figure 2.6 is chosen here to represent the continuous position setpoint change on any one axis of the satellite. The accuracy requirement is also shown explicitly on the diagram as an accuracy upper bound. Clearly, the satellite follows its position command at each setpoint change, which indicates that the PID controller meets its design requirements and specifications and the attitude control system tracks its setpoint command quite well under fault free and healthy operation.

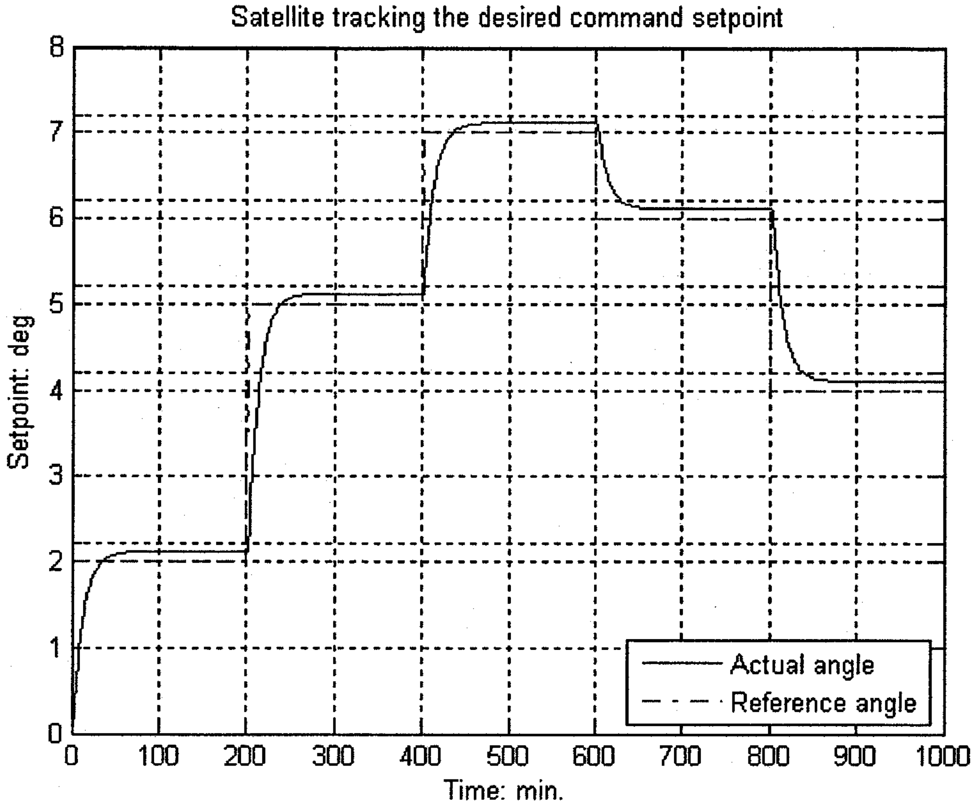

Figure 2.6 The body attitude performance with a PID controller

2.6. Conclusions

The basic concepts of an attitude control system are briefly introduced in this chapter and the dynamics of the reaction wheels are provided in details. To meet the desired pointing accuracy requirements for each axis, separate PID control loops for each axis are designed. Simulation results illustrate that the performance of the PID controllers acceptably met the design specifications.

Chapter 3

Neural Network Observer-based Fault Detection and Isolation for the Reaction Wheels

3.1. General Introduction to Neural Networks

Artificial neural networks traced back to biology has been studied and widely used in the areas of control, signal processing, and pattern recognition and fault diagnosis. A neural network is defined as a massively parallel distributed processor made up of simple processing units, which has a capability for storing knowledge contained in data. This kind of unit is called neuron and the connection between two different neurons is called synaptic weight. The procedure to store knowledge or learning process is called a learning algorithm, which modify the synaptic weights in an orderly fashion to attain a desired design objective. An important characteristic of a trained network is its learning ability, implying that a well trained network is capable of generating a reasonable output with an unseen input [19].

According to [19], neural networks possess three main properties that are suitable for the application considered in this thesis:

- Nonlinearity: A neural network is made up of an interconnection of nonlinear neurons, making itself a nonlinear system. This property makes it suitable for modeling dynamical systems which are normally highly nonlinear.
- Input-output Mapping: A popular neural network paradigm called supervised learning involves modifications of the synaptic weights through a set of training samples. Each sample contains a unique input and a corresponding desired output. The synaptic weights are modified to minimize the difference of the network output and the corresponding desired output when presented with a training sample randomly selected from the training data set. The training procedure will stop when the network reaches a state where there are no further significant changes in weights.
- Adaptability: Neural networks have a built-in capability to adapt its synaptic weights to changes in the surrounding environment.

3.1.1. Neuron Model

A neuron is the basic processing unit of a neural network. It represents a transformation function from input p to output $\varphi(p)$. A neuron with R-element input vector is shown in Figure 3.1.

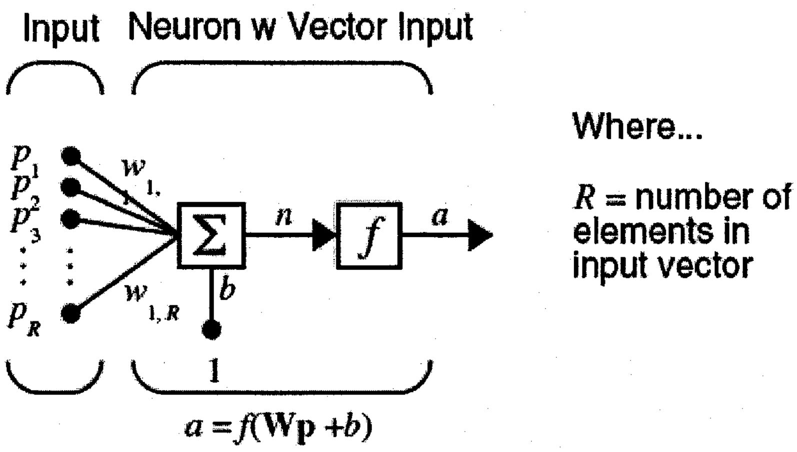

Figure 3.1 A neuron with R-element input vector [37]

The relation between input p and output a is governed by:

$$a = f\left(Wp + b\right) \tag{3.1}$$

where W is the weight matrix, b is called a bias and f represents the activation function. A hyperbolic tangent sigmoid function is generally used as the activation function in all the neurons. This kind of activation function is defined as

$$\tan sig\left(x\right) = \frac{2}{1+e^{-2x}} - 1 \tag{3.2}$$

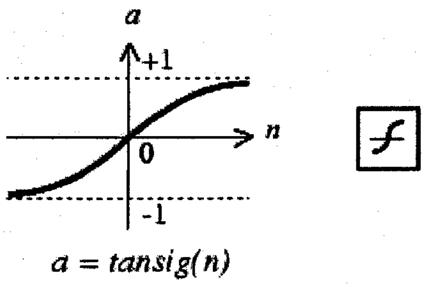

Figure 3.2 A hyperbolic tangent sigmoid activation function [37]

3.1.2. Network Architectures

The network architecture represents how the neurons are arranged and interconnected together. According to [19], network architectures are broadly divided into two categories:

a) Feed-forward networks

A single-layer feed-forward network with R input elements and S neurons in the layer is shown below in Figure 3.3. Each element of the input vector p is connected to all the individual neurons through the weight matrix W and each neuron output forms a column vector a which represents the output of the network.

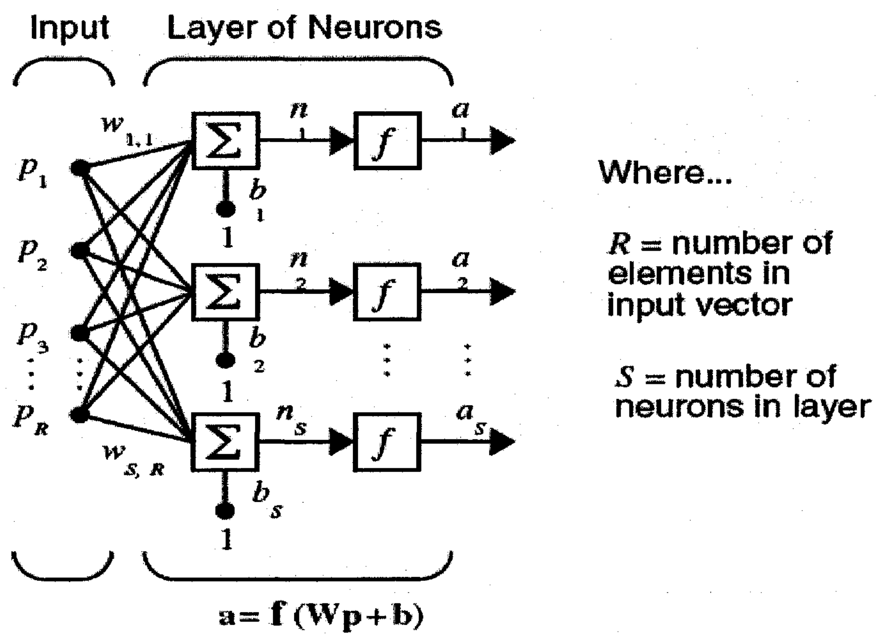
Input
Layer of Neurons
$w_{1,1}$
p_1
p_2
p_3
p_R
$w_{S,R}$
Σ
n_1
n_2
n_S
f
a_1
a_2
a_S
b_1
b_2
b_S
1
Where...
R = number of elements in input vector
S = number of neurons in layer
$\mathbf{a} = \mathbf{f}(\mathbf{Wp}+\mathbf{b})$

Figure 3.3 A single-layer feed-forward neural network [37]

In practice, a single-layer feed-forward network is usually used to map a simple nonlinear combination of inputs, however, if the layers are cascaded, they are able to represent arbitrary complex nonlinear mappings. The most important characteristic of a multilayer feed-forward network is that it is capable of learning the map of any complexity. A three layer feed-forward network, one input layer, one hidden layer and one output layer, is capable of representing an arbitrary mapping between the input and output variables.

b) Feed-back networks

A recurrent or a feedback network distinguishes itself from a feed-forward network in which it has at least one feedback loop. The presence of feedback loops has a profound impact on the learning capability of the network and its performance. Involving the use of particular branches composed of unit-delay elements, the recurrent network processes a nonlinear dynamical behavior.

Mainly, there are two kinds of recurrent networks: Elman networks and Hopfield networks. Elman networks are two-layer back-propagation networks, with the addition of a feedback connection from the output of the hidden layer to its input. This feedback path allows Elman networks to learn to recognize and generate temporal patterns, as well as spatial patterns. The Hopfield network is used to store one or more stable target vectors. These stable vectors can be viewed as memories that the network recalls when provided with similar vectors that act as a cue to the network memory.

3.1.3. Network Learning Methods

Network learning is defined as a procedure of synaptic weights adjustment to capture the information contained in the training data. By changing these synaptic weights, the network can generate correct outputs when presented with different inputs. Generally, the learning methods are divided into two categories: unsupervised learning and supervised learning [19].

- Unsupervised Learning

  Unsupervised learning requires no target output vector values, and hence no comparison of network outputs with a set of predetermined desired outputs. The learning set consists solely of input vectors, and the learning algorithm modifies synaptic weights so as to produce consistent outputs. The learning process in essence extracts the statistical properties of the learning set and group similar vectors into similar classes.

- Supervised Learning

  In contrast to unsupervised learning, the learning set of supervised learning contains both input vectors and the corresponding desired output vectors. After the output of the network for a given input vector is computed and compared to its desired target, the difference or error is fed back so that the synaptic weights are adjusted according to an algorithm that tends to minimize this error. The vectors in the training data sets are supplied randomly and sequentially to the network and the learning procedure is repeated until the error for the entire training data set reaches an acceptable low level that was defined a *priori* [19].

3.1.4. Network Training Styles

Briefly, the network training style can be grouped into either batch training in which the weights and biases are updated after all the inputs are represented or incremental training in which the weights and biases of the network are updated each time when an input is presented to the network [37, 38]. Batch learning can be called as epoch learning while instantaneous learning is the synonym for the incremental learning.

Usually, the training style can also be distinguished as on-line versus off-line learning. In off-line training, all the data are stored and can be used repeatedly while in on-line training, each data is discarded after it is represented and the weights are updated. Batch learning is always off-line while incremental learning is always on-line.

For the neural network used in this thesis, an intermediate method is adopted for the training, which is named mini-batch [38]. By using this training method, the parameters (weights and delays) are initialized before the training and during the training the following steps is repeated: process certain number of training data (two or more, but not all the training data) first, then update the weights and delays. This particular number is one of the training parameters denoted by an update period P in this thesis. In other words, the network treats the P data points as a batch similar to the batch training. After the weights and delays are updated, the network discards the P data points and the next P data inputs in the training set are represented to the network.

3.2. Adaptive Time Delay Neural Network (ATDNN)

For conventional networks, each neuron calculates the weighted sum of the inputs directly and then passes it through a nonlinear activation function while for the time delay neural network, a certain delay associated with each weight is introduced to each neuron [36]. Figure 3.4 illustrates the structure of the adaptive time delay neuron. This dynamic neuron is capable of representing the relationships between events in time. The input–output relationship of this neuron is governed by

$$y(t) = \sigma(\sum_{i=1}^{N} w_i x_i(t-\tau_i)) \tag{3.3}$$

where $w_i^{'}s$ are weights of the neuron, $\tau_i^{'}s$ are delays and $\sigma(.)$ is the activation function.

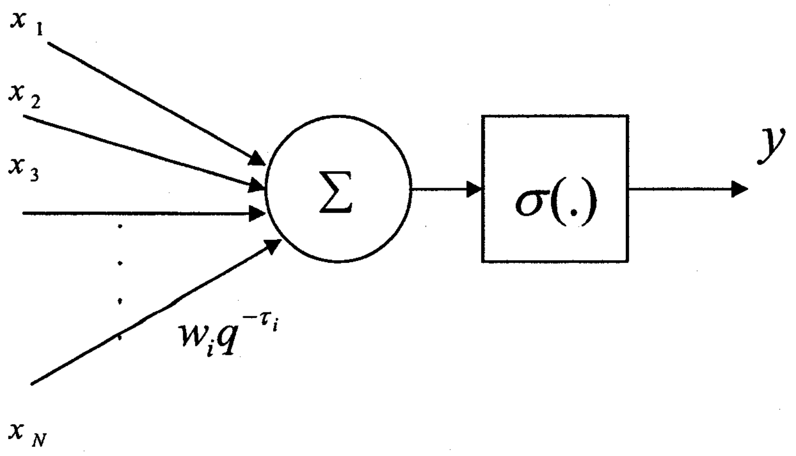

Figure 3.4 The structure of a dynamic neuron, $q^{-\tau}$ is the shift operator [36]

A dynamic multilayer feed-forward network is now constructed by using this dynamic neuron instead of the conventional static neurons in order to obtain a feedforward ATDNN as indicated in Figure 3.5.

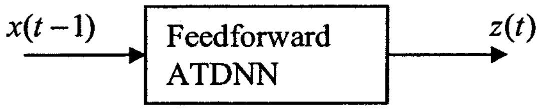

Figure 3.5 The architecture of a feed-forward adaptive time delay neural network [36]

3.2.1. Adaptive Time Delay Neural Networks for Nonlinear System Mapping

In this thesis, a three layer $6 \times 3 \times 1$ adaptive time delay neural network is constructed for each axis of the satellite to map the nonlinearity characteristic of the wheel. The update interval for parameters P is chosen as 2 in this thesis. During the training phase, the series-parallel architecture of this adaptive time delay neural network is used to map the nonlinear system with the following input-output nonlinear autoregressive moving average model.

$$y(t) = f[y(t-1), y(t-2), ..., y(t-N_s), u(t-1), u(t-2), ..., u(t-M_s)] \tag{3.4}$$

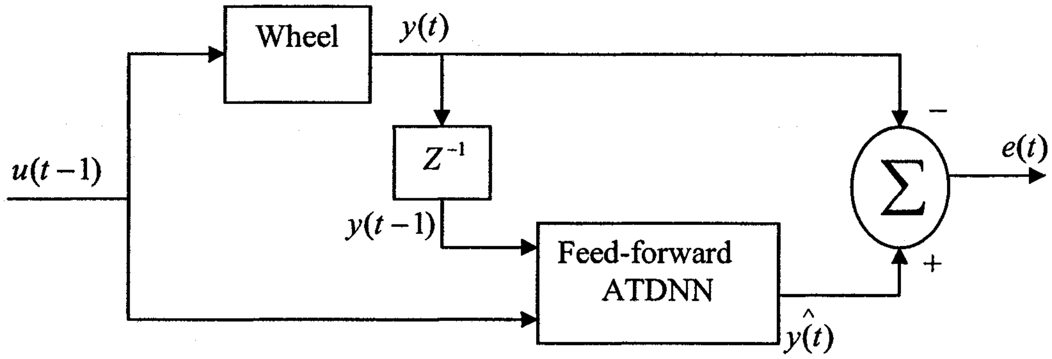

Figure 3.6 A series – parallel architecture of the adaptive time delay neural network [36]

By substituting $y(t-1) = \hat{y}(t-1)$, a parallel architecture commonly named Recurrent ATDNN and as shown in Figure 3.7 can be obtained which is used to generate an estimated wheel output during the recall phase.

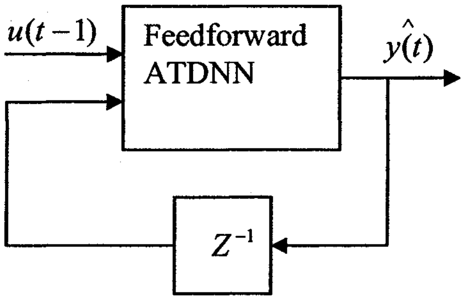

Figure 3.7 A recurrent adaptive time delay neural network [36]

3.2.2. Adaptation Laws for the Recurrent Adaptive Time Delay Neural Networks [36]

3.2.2.1. Output Layer Adaptation Laws

In this layer, the weights and the delays need to be updated. For weights adaptation laws we have:

$$\begin{cases} \Delta w_{kj}^{3} = \eta \delta_{k}^{3}(t) \dfrac{\partial o_{k}^{3}(t)}{\partial w_{kj}^{3}}; \\ \delta_{k}^{3}(t) = y_{k}(t) - o_{k}^{3}(t); \\ \dfrac{\partial o_{k}^{3}(t)}{\partial w_{kj}^{3}} = \sigma'(net_{k}^{3}(t))[o_{j}^{2}(t-\tau_{kj}^{3}) + \\ \sum_{p=1}^{N^2} w_{kp}^{3} \sigma'(net_{p}^{2}(t-\tau_{kp}^{3})) \sum_{n=1}^{N^1} w_{pn}^{2} \sigma'(net_{n}^{1}(t-\tau_{pn}^{2}-\tau_{kp}^{3})) w_{n2}^{1} \dfrac{\partial o_{k}^{3}(t-\tau_{n2}^{1}-\tau_{pn}^{2}-\tau_{kp}^{3}-1)}{\partial w_{kj}^{3}}] \end{cases} \tag{3.5}$$

The weights are updated each P period resulting in the update expression for w_{kj}^{3} as

$$w_{kj}^{3}(new) = w_{kj}^{3}(old) + \frac{1}{P}\sum_{p=1}^{P} \Delta w_{kj}^{3} \tag{3.6}$$

where w_{kj}^{l} is the associated weight between the k^{th} neuron in the l^{th} layer (output layer) and the j^{th} neuron in the $(l-1)^{th}$ layer, and τ_{kj}^{l} is the delay associated with the weight w_{kj}^{l}, and P is defined as the update period of the adaptive time delay neural network as mentioned before.

For the delays, we have:

$$\begin{cases} \Delta \tau_{kj}^{3} = \eta \delta_{k}^{3}(t) \dfrac{\partial o_{k}^{3}(t)}{\partial \tau_{kj}^{3}}; \\ \delta_{k}^{3}(t) = y_{k}(t) - o_{k}^{3}(t); \\ \dfrac{\partial o_{k}^{3}(t)}{\partial \tau_{kj}^{3}} = \sigma'(net_{k}^{3}(t)) \sum_{p=1}^{N^2} w_{kp}^{3} \sigma'(net_{p}^{2}(t-\tau_{kp}^{3})) \sum_{n=1}^{N^1} w_{pn}^{2} \sigma'(net_{n}^{1}(t-\tau_{pn}^{2}-\tau_{kp}^{3})) \\ [w_{n2}^{1} \dfrac{\partial o_{k}^{3}(t-\tau_{n2}^{1}-\tau_{pn}^{2}-\tau_{kp}^{3}-1)}{\partial \tau_{kj}^{3}} + w_{n1}^{1} \dfrac{\partial u(t-\tau_{n2}^{1}-\tau_{pn}^{2}-\tau_{kp}^{3}-1)}{\partial \tau_{kj}^{3}}] \end{cases} \tag{3.7}$$

Therefore, the law for delay adjustment is summarized as

$$\tau_{kj}^{3}(new) = \tau_{kj}^{3}(old) + \frac{1}{P}\sum_{p=1}^{P}\Delta\tau_{kj}^{3} \tag{3.8}$$

3.2.2.2. Hidden Layer Adaptation Laws

Similar to the previous case, we have:

$$\begin{cases} \Delta w_{ji}^{2} = \eta\delta_{j}^{2}(t)\dfrac{\partial o_{j}^{2}(t)}{\partial w_{ji}^{2}}; \\ \delta_{j}^{2}(t) = \sum_{q=1}^{N^{3}}\delta_{q}^{3}(t)\sigma' net_{q}^{3}(t)w_{qj}^{3}; \\ \dfrac{\partial o_{j}^{2}(t)}{\partial w_{ji}^{2}} = \sigma'(net_{j}^{2}(t))[o_{i}^{1}(t-\tau_{ji}^{2}) + \\ \sum_{n=1}^{N^{1}} w_{jn}^{2}\sigma'(net_{n}^{1}(t-\tau_{jn}^{2}))w_{n2}^{1}\sigma'(net_{q}^{3}(t-\tau_{n2}^{1}-\tau_{jn}^{2}-1))\sum_{p=1}^{N^{2}} w_{qp}^{3}\dfrac{\partial o_{p}^{2}(t-\tau_{n2}^{1}-\tau_{jn}^{2}-\tau_{qp}^{3}-1)}{\partial w_{ji}^{2}}] \end{cases} \tag{3.9}$$

$$w_{ji}^{2}(new) = w_{ji}^{2}(old) + \frac{1}{P}\sum_{p=1}^{P}\Delta w_{ji}^{2} \tag{3.10}$$

where w_{ji}^{l} is the associated weight between the j^{th} neuron in the l^{th} layer (hidden layer) and the i^{th} neuron in the $(l-1)^{th}$ layer.

Similarly, for the delay update we have:

$$\begin{cases} \Delta\tau_{ji}^{2} = \eta\delta_{j}^{2}(t)\dfrac{\partial o_{j}^{2}(t)}{\partial\tau_{ji}^{2}}; \\ \delta_{j}^{l}(t) = \sum_{q=1}^{N^{3}}\delta_{q}^{3}(t)\sigma'(net_{q}^{3}(t))w_{qj}^{3}; \\ \dfrac{\partial o_{j}^{2}(t)}{\partial\tau_{ji}^{2}} = \sigma'(net_{j}^{2}(t))\sum_{n=1}^{N^{2}} w_{jn}^{2}\sigma'(net_{n}^{1}(t-\tau_{jn}^{2}))[w_{n2}^{1}\sigma'(net_{q}^{3}(t-\tau_{n2}^{1}-\tau_{jn}^{2}-1)) \\ \sum_{p=1}^{N^{2}} w_{qp}^{3}\dfrac{\partial o_{p}^{2}(t-\tau_{n2}^{1}-\tau_{jn}^{2}-\tau_{qp}^{3}-1)}{\partial\tau_{ji}^{2}} + w_{n1}^{1}\dfrac{\partial u(t-\tau_{n1}^{1}-\tau_{jn}^{2}-\tau_{qp}^{3}-1)}{\partial\tau_{ji}^{2}}] \end{cases} \tag{3.11}$$

where τ_{ji}^{l} is the delay associated with the weight w_{ji}^{l}.

$$\tau_{ji}^{2}(new) = \tau_{ji}^{2}(old) + \frac{1}{P}\sum_{p=1}^{P}\Delta\tau_{ji}^{2} \tag{3.12}$$

3.2.2.3. Input Layer Adaptation Laws

For the weight update we have:

$$\begin{cases} \Delta w_{ji}^{1} = \eta\delta_{j}^{1}(t)\dfrac{\partial o_{j}^{1}(t)}{\partial w_{ji}^{1}}; \\ \delta_{j}^{1}(t) = \sum_{q=1}^{N^2}\delta_{q}^{2}(t)\sigma' net_{q}^{2}(t)w_{qj}^{2}; \\ \dfrac{\partial o_{j}^{1}(t)}{\partial w_{ji}^{1}} = \sigma'(net_{j}^{1}(t))[o_{i}^{0}(t-\tau_{ji}^{1}) + w_{j2}^{1}\sigma'(net_{q}^{3}(t-\tau_{j2}^{1}-1)) \\ \sum_{p=1}^{N^2} w_{qp}^{3}\sigma'(net_{p}^{2}(t-\tau_{j2}^{1}-\tau_{qp}^{3}-1)\sum_{n=1}^{N^1} w_{pn}^{2}\dfrac{\partial o_{n}^{1}(t-\tau_{j2}^{1}-\tau_{pn}^{2}-\tau_{qp}^{3}-1)}{\partial w_{ji}^{1}}] \end{cases} \tag{3.13}$$

where w_{ji}^{l} is the associated weight between the j^{th} neuron in the l^{th} layer (input layer) and the i^{th} input.

Similarly, for the delay update we have

$$\begin{cases} \Delta \tau_{ji}^{1} = \eta\delta_{j}^{1}(t)\dfrac{\partial o_{j}^{1}(t)}{\partial \tau_{ji}^{1}}; \\ \delta_{j}^{1}(t) = \sum_{q=1}^{N^2}\delta_{q}^{2}(t)\sigma'(net_{q}^{2}(t))w_{qj}^{2}; \\ \dfrac{\partial o_{j}^{1}(t)}{\partial \tau_{ji}^{1}} = \sigma'(net_{j}^{1}(t))[w_{j1}^{1}\dfrac{\partial u(t-1-\tau_{j1}^{1})}{\partial \tau_{ji}^{1}} + w_{j2}^{1}\sigma'(net_{q}^{3}(t-\tau_{j2}^{1}-1)) \\ \sum_{p=1}^{N^2} w_{qp}^{3}\sigma'(net_{p}^{2}(t-\tau_{j2}^{1}-\tau_{qp}^{3}-1))\sum_{p=1}^{N^2} w_{pn}^{2}\dfrac{\partial o_{n}^{1}(t-\tau_{j2}^{1}-\tau_{pn}^{2}-\tau_{qp}^{3}-1)}{\partial \tau_{ji}^{1}}] \end{cases} \tag{3.14}$$

where τ_{ji}^{l} is the delay associated with the weight w_{ji}^{l}. The number of neurons in output layer is donated as q here.

3.3. Neural Network Observer-based FDI Scheme

3.3.1. General Idea

In this section, a neural network observer-based FDI scheme for reaction wheels is developed and detailed. Three independent observers are designed for each wheel on the three axes to execute the FDI mission. The block diagram of the proposed scheme is shown in Figure 3.8.

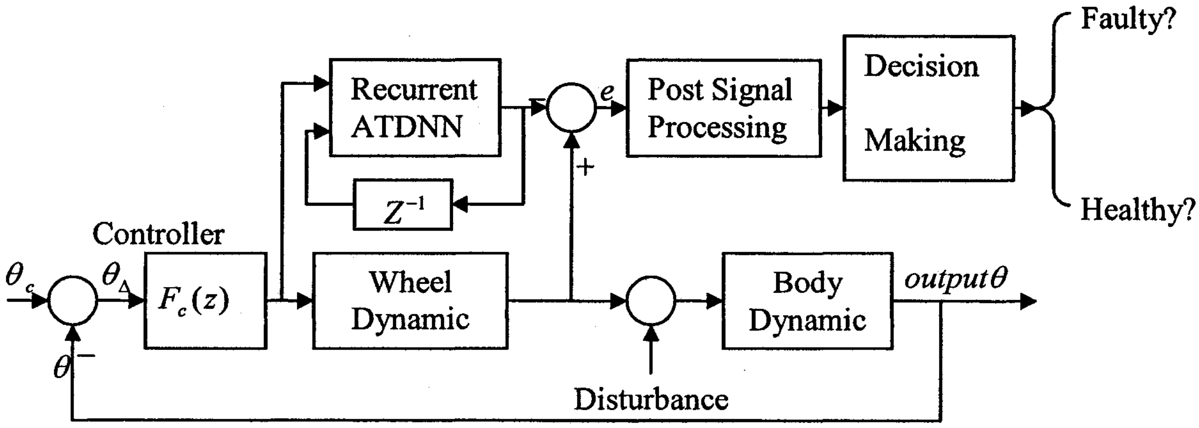

Figure 3.8 The neural network observer-based FDI scheme in the recall phase

The recurrent adaptive time delay neural network is implemented here to generate the estimated wheel output. Comparing the estimated with the actual wheel output, the residual error signal is generated. After some signal processing, the generated residual error is used to compare with the pre-defined residual based on the fault free operation of the wheel. Once the corresponding residual signal has been exceeded the threshold for a considerable period of time (for instance 20 minutes here), a fault can be concluded to have occurred.

The proposed recurrent neural network has two inputs, one is the output of the PID controller and the other one is the one step delay from its output. It also needs two parameters, namely: weights and delays but during the recall phase those parameters which are obtained during the training phase are never adjusted. As mentioned before, during the training phase a feed-forward adaptive time delay neural network is used. Figure 3.9 depicts the diagram of the neural network during the training phase. The inputs for the ATDNN are outputs from the PID controller and one step delay of the wheel output. The error e is the difference between the actual torque and the estimated torque from the neural network, which is used to adjust the weight W and the delay τ to minimize itself to the desired performance index. Once the error signal reaches the acceptable error tolerance, the training can be stopped and the network is considered as being well trained for generating the estimated data of the wheel as close as the actual data. The recurrent network which executes the recall task adopts these parameters which provide good estimated data under fault free circumstances. Once a fault has occurred in the wheel, the neural network still generates the estimated data that is corresponding to the normal and healthy case of the wheel. Therefore, the error between the actual and the estimated torques becomes large when a fault is present and will exceed the corresponding threshold, which is the purpose of the FDI.

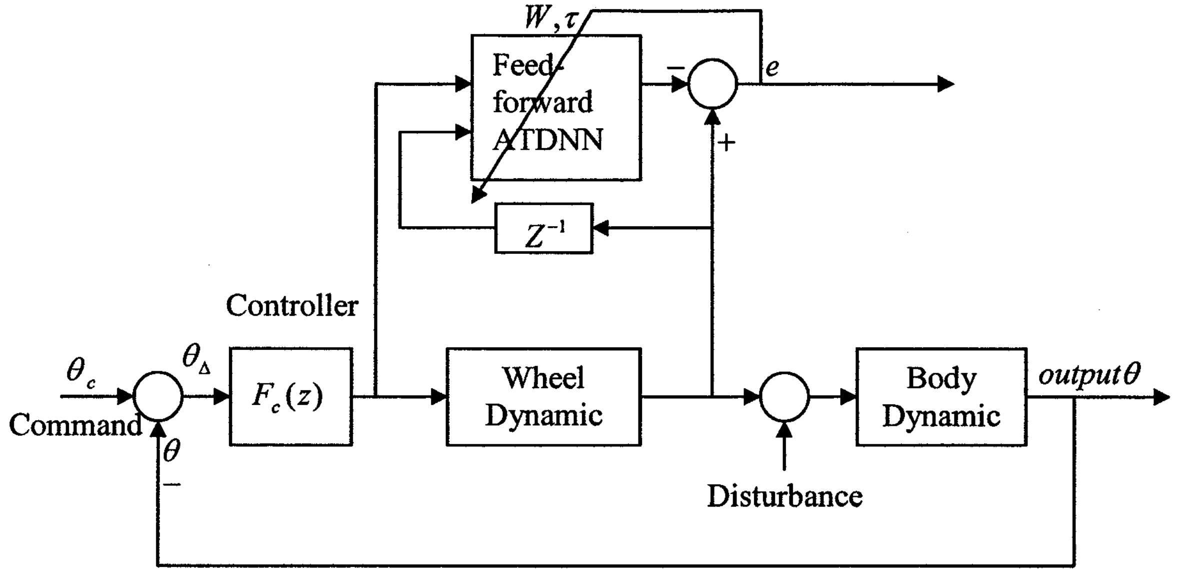

Figure 3.9 A neural network observer-based FDI scheme for the network training phase

3.3.2. Training Phase of the Neural Network Observer-based FDI Scheme

3.3.2.1. Training Data Collection

The data for the network training are the output of the PID controller and the corresponding reaction torque signal. With the aid of MATLAB, the wheel model is simulated to operate normally with a sampling period of 0.1 minute. During the first 200 minutes, the wheel will change its position from an initial angle of 0° to the required setpoint of 2° and then it continues to change to another setpoint of 6° in the following 200 minutes. The data collected from the output of the PID controller and the output of the wheel are collected as training input signals 1 and 2, respectively. The typical training data signals for the training of the neural networks for one axis are shown in Figure 3.10 and Figure 3.11 as well as the corresponding setpoint changes as shown in Figure 3.12.

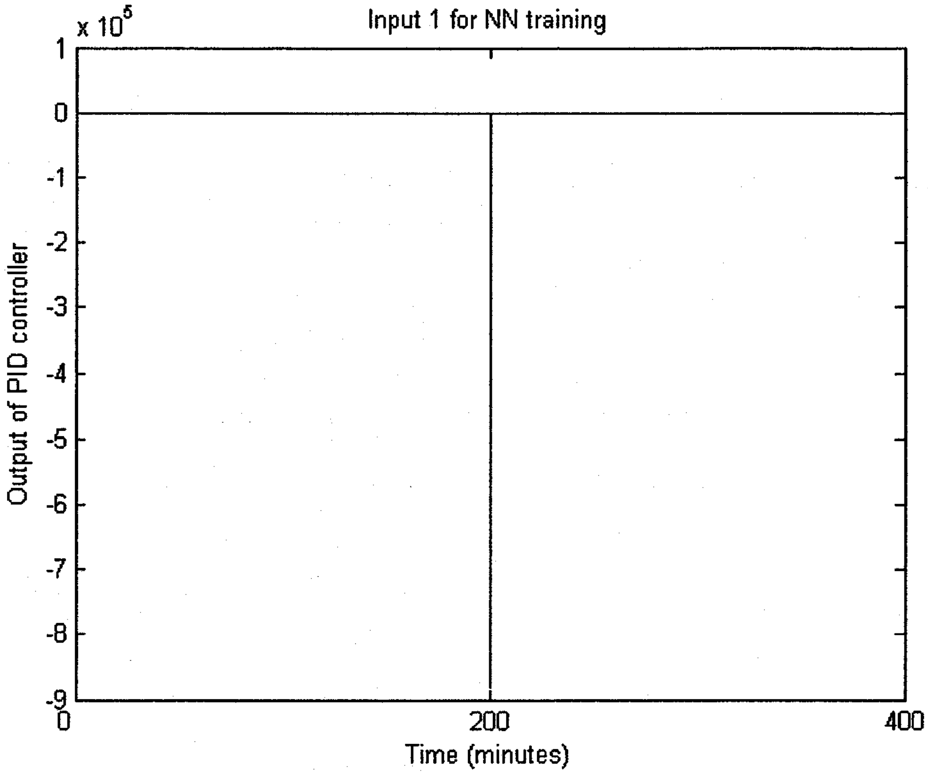

Figure 3.10 Network training data 1 from the controller

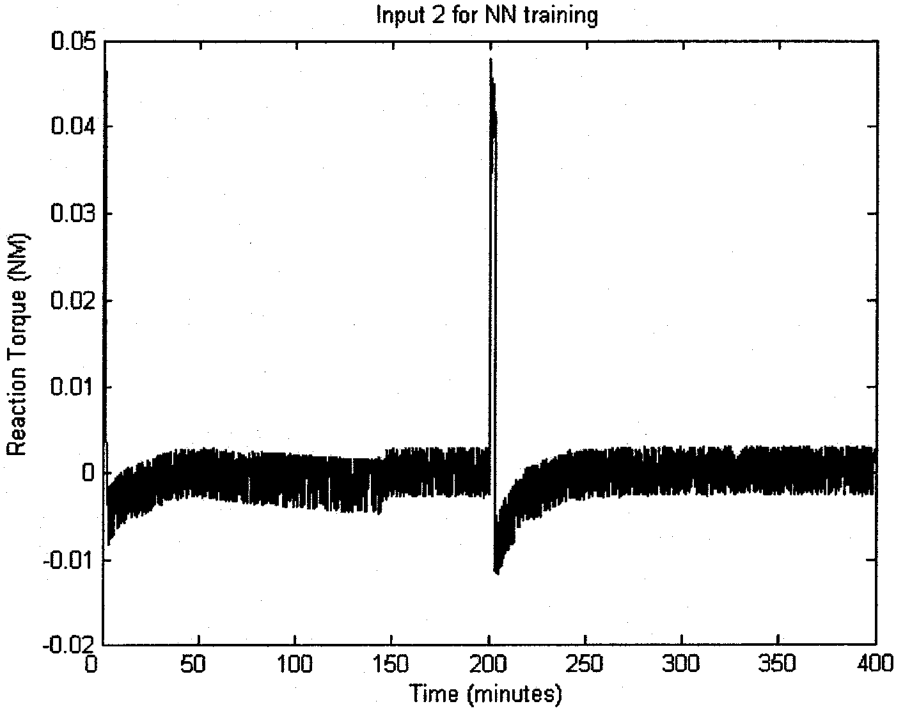

Figure 3.11 Network training data 2 from the wheel

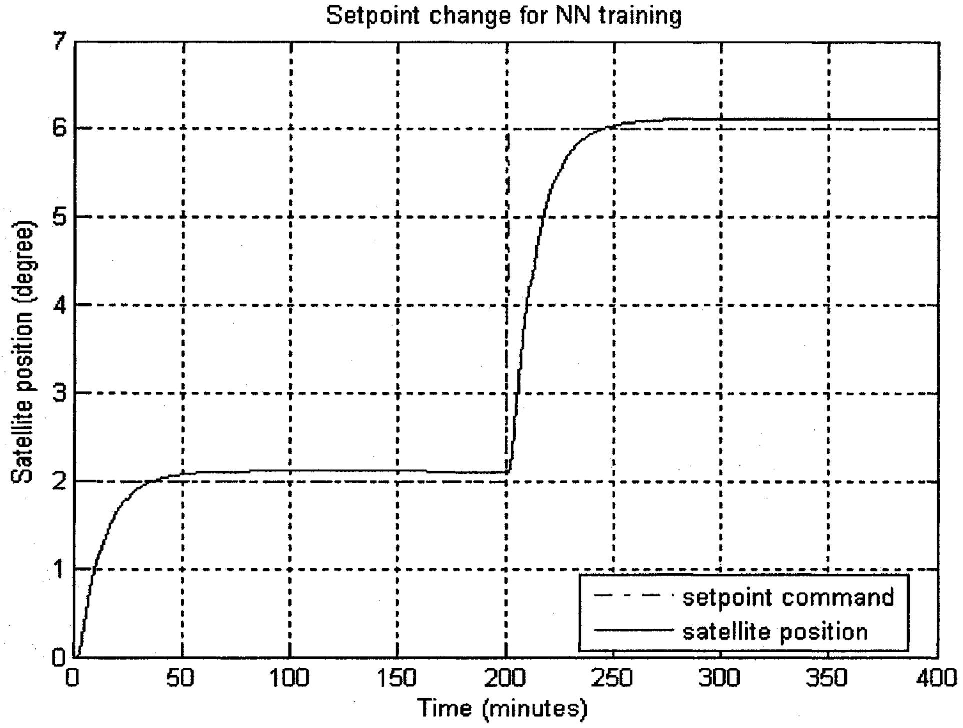

Figure 3.12 Satellite setpoint change command for training and the actual position

3.3.2.2. Training Data Pre-Processing

Given that the activation function adopted in this network is a tangent sigmoid function, one needs to normalize the training data set into the range $[-1,1]$ to ensure that each input has an equal impact on the training set for good and fair representation.

3.3.2.3. Network Initialization

The weights are randomly generated in the range [-0.5, 0.5] and the delays are pre-defined in the range [0, 5]. The potential influences caused by the initial weights are avoided by repeating the training process several times, which is 20 times in this thesis.

3.3.2.4. Network Training

With the initial weights and the delays specified as indicated earlier, the network is trained on-line with an adaptive learning rate for the weights and a fixed learning rate for the delays. The sample interval for the adjustment P is chosen as 2, which implies that the weights and delays are adjusted after processing each 2 data points. The weight learning rate is initialized as 0.85 and the adjusting rule is borrowed from *traingd*, a MATLAB defined training method. Specifically,

- To implement this method, the initial network outputs and errors need to be calculated in advance. At each sampling interval P, new weights are calculated using the current learning rate as well as the new outputs and errors.
- On the one hand, if the new error exceeds the old error by more than a ratio, e.g.1.04, this implies that the current learning rate is too high and therefore needs to be decreased by multiplying it with a factor, e.g. 0.7. The new weights are also discarded and the weights are kept as before.
- On the other hand, if the new error is less than the old error, then the learning rate is increased by multiplying it with the factor 1.05 in this thesis.

The learning rate for the delays is fixed as 0.001 and the maximum delay $\tau_{\max}$ is selected as 200. The weights and the delays need to be stored during the training according to the sampling interval. After several training epochs, once the network becomes saturated or the magnitude of the mean square error Mse reaches 10^{-5}, then the training process is terminated. The final training result for a typical scenario including the corresponding performance and the mean square error is shown in Figure 3.13.

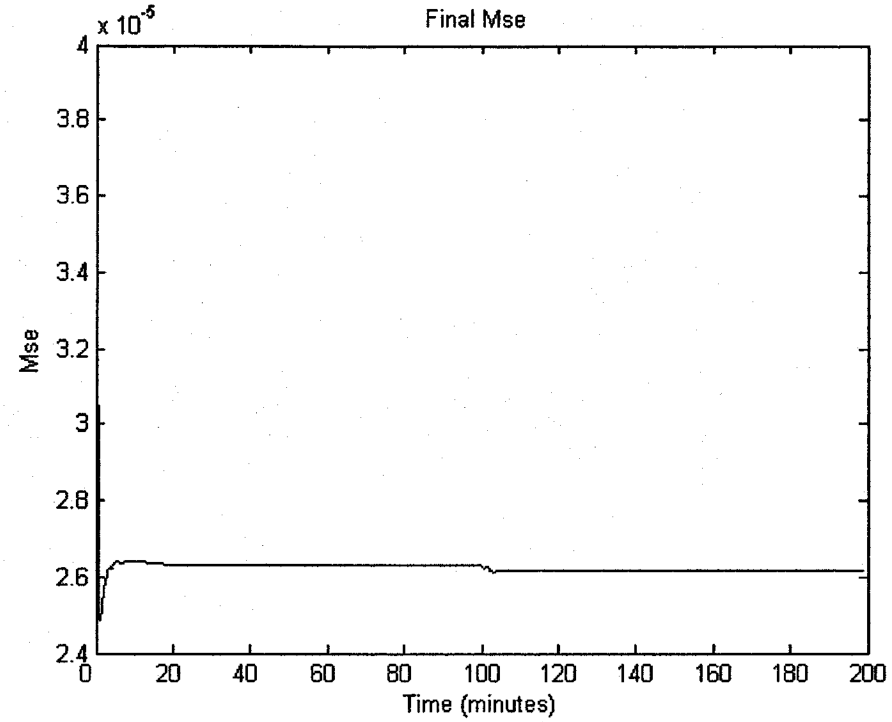

Figure 3.13 The MSE performance

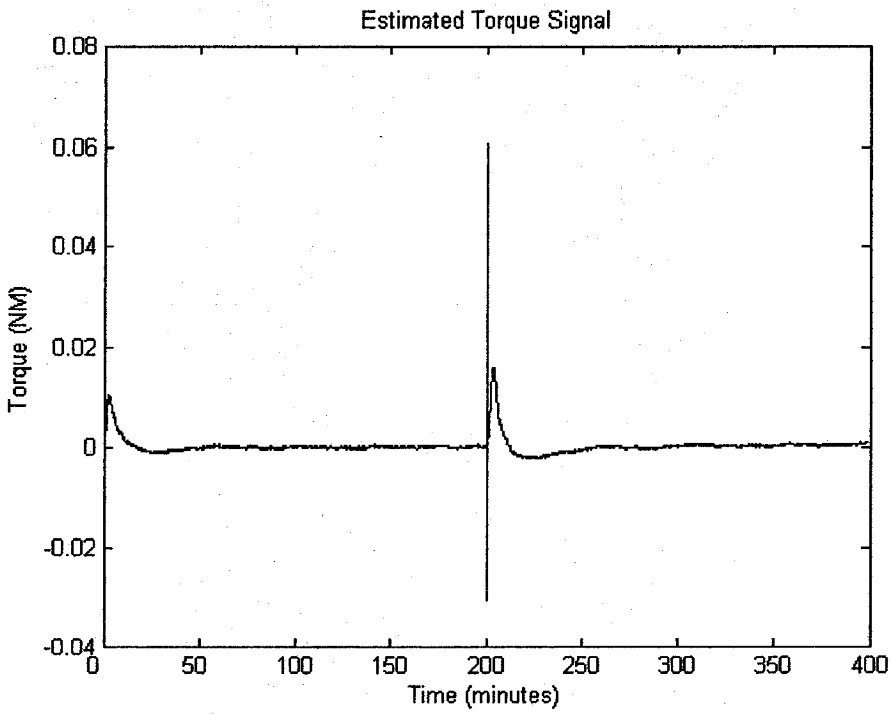

Figure 3.14 The estimated torque signal generated during the training neural network

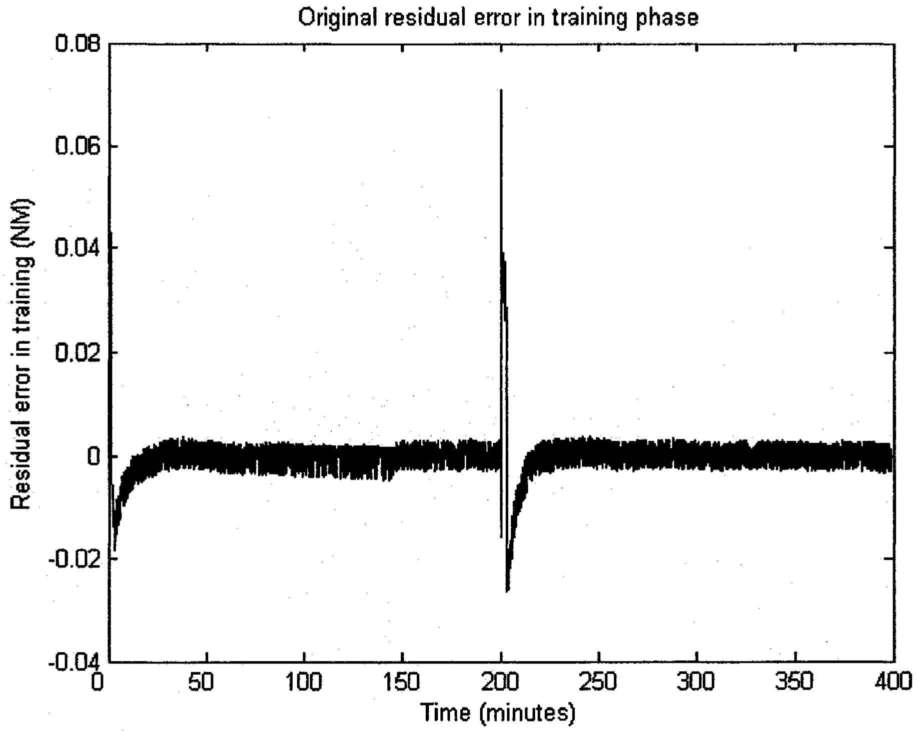

Figure 3.15 The actual residual error generated during the training phase

As illustrated from Figure 3.15, after the spike which is due to the sepoint change, the error is neatly around zero, which shows that the trained neural network models the desired reaction wheel output quite well.

3.3.2.5. Parameter Determination

After completing 20 different training scenarios, the parameters of the network to be used for the recall phase should be collected properly. Since the weights and the delays in the last epoch of the training approach are stable, the mean value of the last 300 sampled intervals is calculated for each training process. After gathering these 20 mean values, the mean of the previous 20 mean values is calculated and is then considered as the final mean value of the weights and delays for the trained network.

3.3.3. Threshold Generation for the Neural Network Observer-based FDI Scheme

3.3.3.1. The Residual Error Generation

The threshold for fault detection is generated based on the residual errors obtained under fault free operation of the wheel. The residual error is defined as the difference between the actual reaction torque from the wheel and the estimated values from the neural network. Once the instantaneous residual error is made available, it needs to pass through a low pass filter to be smoothened and so the moving average value of the residual error will be substituted for the raw residual error as the proposed residual signal in the rest of this thesis. For this purpose, the moving average filter window size is to be considered as a design parameter. Large window size will output smooth residual signal by scarifying its accuracy. Therefore, a compromise is needed here. Based on a number of experiments, the window size is decided as 250, which means that the residual is generated on the mean value of each batch data in 25 minutes. A typical residual error signal is shown in Figure 3.17.

3.3.3.2. Threshold Curve Determination

The threshold curves should be capable of bounding the residual error signal when the wheel is operating under normal situation. Once the error signal has exceeded this boundary, either its upper or/and its lower bound, for a considerable time period (e.g. 20 minutes) the decision making block can make judgment on the existence of a fault. The formula that we use to calculate the threshold is given by

$$X_{threshold} = \overline{X}_{residual} + \alpha \times \sigma \tag{3.15}$$

where the value σ is the standard deviation of the residual signals and α is a factor which can be different for the corresponding upper or lower boundary curves. The larger the value of α, the more conservative the threshold curve will be, however, it also implies that it has lower capability for use in the case of small fault detection.

The general way for generating the threshold is based on the training phase. That is, the residual error is collected during the training phase. In this thesis, we propose to compute the threshold based on the recall phase, in which the residual error is collected for the recall network whose parameters are provided by the training network. In other words, the purpose of the training network is that it is responsible for providing the parameters that are used by the recall network. The advantage of this method is that it facilitates the thresholds being independent of the training phase, which is feasible and practical. For instance, it is impractical to train a network for a long time and then use the generated threshold for fault detection.

Although the network may be well trained at setpoint changes from 2° to 6°, it may have strong representation capability and may be able of mapping the setpoint changes up to higher degrees. However no unique threshold is able to map all the setpoint changes, and therefore the parameters for generating the thresholds need to be adjusted. In this thesis, one set of thresholds has been found be capable of bounding all the setpoint changes from 2.5° to 7.5° during the fault free operation of the satellite. The parameters that are used to generate this threshold are listed below:

- for the threshold obtained based on training:
  a) The window size for the residual error is set to 305;
  b) α range is from 25 to -9.

- for the threshold obtained based on recall:

  a) The window size for the residual error is set to 250;

  b) The range of α is set from 5 to -5 or 3 to -5.

The threshold curves based on the training approach is shown in Figure 3.16 while the comparable threshold curves obtained based on the recall approach is shown in Figure 3.17.

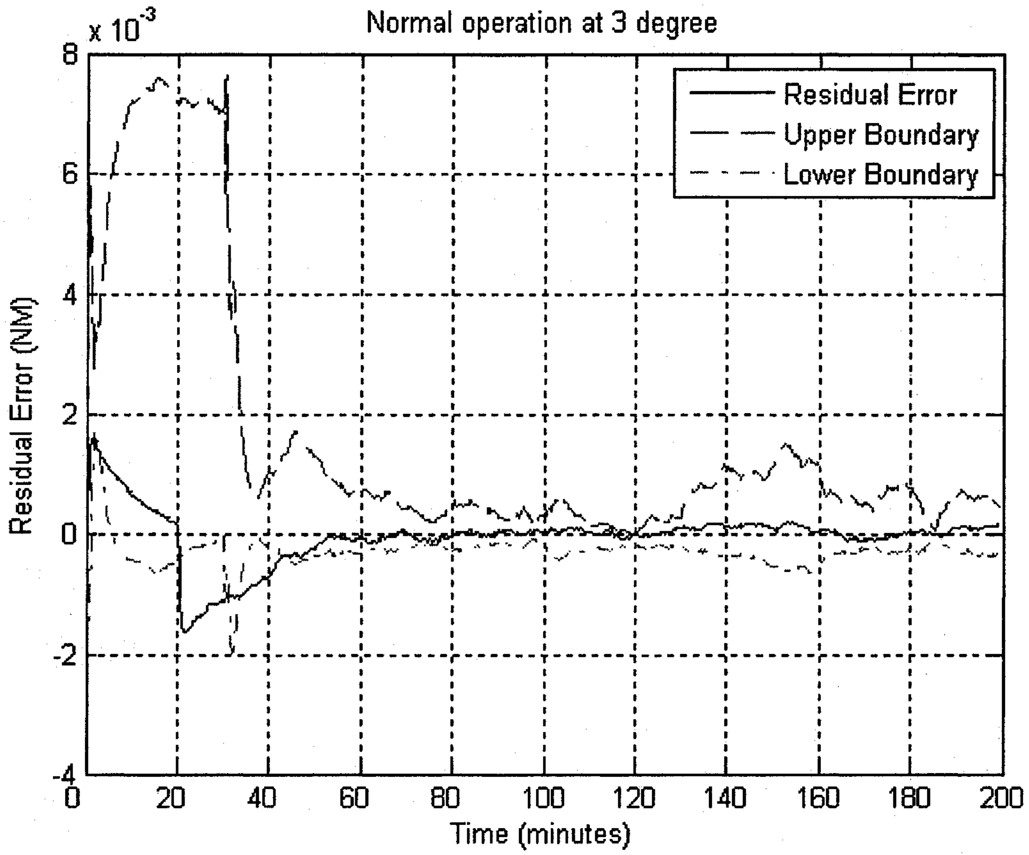

Figure 3.16 The threshold curves for the residual error test based on the training approach

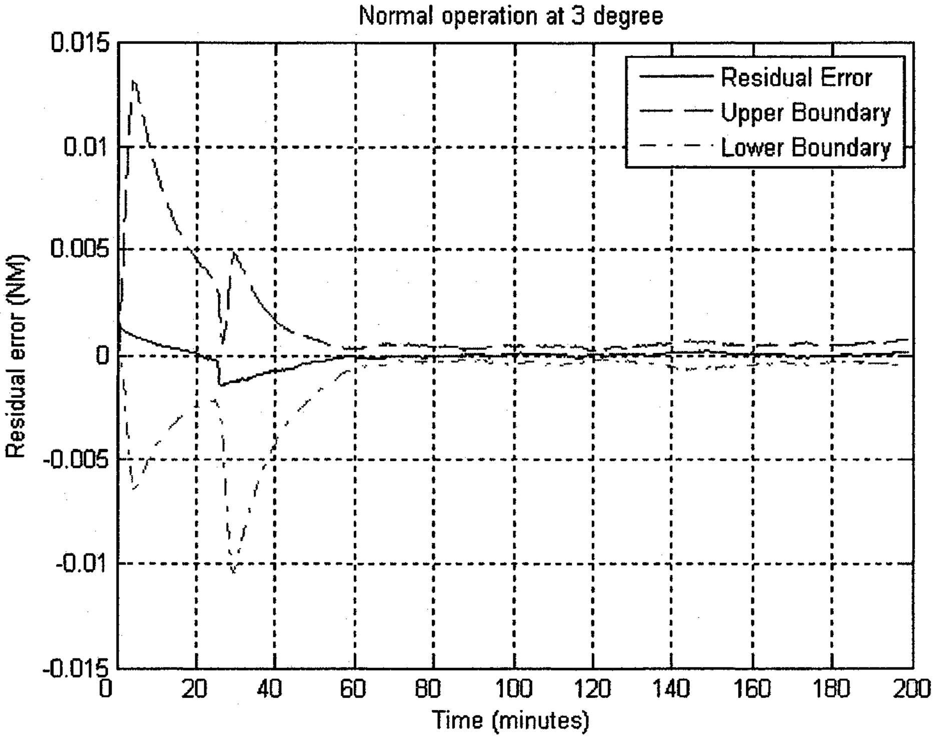

Figure 3.17 The threshold curves for the residual error test based on the recall approach

Both the above two threshold curves are formed under a fault free operation of the satellite and they provide absolutely false alarm free detection results. It should be stated here that false alarm free results do not imply that the actual torque signal at no time exceeds the threshold curve under fault free operation. However, if the time the residual error exceeds the thresholds is less than our detection time (20 min), then this situation will not be considered as a false alarm. This is due to the fact that the entire system is adaptive and nonlinear, and the disturbances imposed on the satellite also impact the wheel performance. Moreover, a false positive situation will occur when the fault is removed from the wheel, during which the signal tends to move back inside the threshold curves. Therefore, no threshold curve can guarantee that the residual error will never exceed the thresholds under normal operation of the wheel.

3.3.4. Decision Making for the Neural Network Observer-based FDI Scheme

Once the residual signal has exceeded the threshold curves from a given point of time for more than 20 minutes, a fault would then be considered to have occurred at that point. Although the three axes are coupled dynamically together, the isolation study conducted in the next chapter shows that the fault detection is only caused by the fault in that axis itself. Details are provided extensively in the next chapter.

3.4. Conclusions

After the basic concepts of neural network are introduced, a recurrent adaptive time delay neural network is constructed in this chapter. The entire FDI scheme with this recurrent ATDNN is provided in details. Simulation results of the neural network training phase provided that the trained recurrent adaptive time delay neural network is capable of modeling the wheel output quite well.

Chapter 4

Simulation Results of the Proposed FDI Scheme for the Reaction Wheels

4.1. Individual Setpoint Detection

For all the cases that are studied in this subsection, the fault has occurred at t = 100 minutes in a single axis and the other two axes do indeed operate under normal and healthy conditions.

4.1.1. Voltage Fault Studies

4.1.1.1. Fault Detection for a Voltage Drop

Briefly, the network is capable of detecting faults corresponding to voltage changes from 8V to 0V, 1V, 2V, 3V and 4V; however it can not detect smaller voltage changes such as 8V to 5V. A typical V_{bus} fault detection result is shown in Figure 4.1.

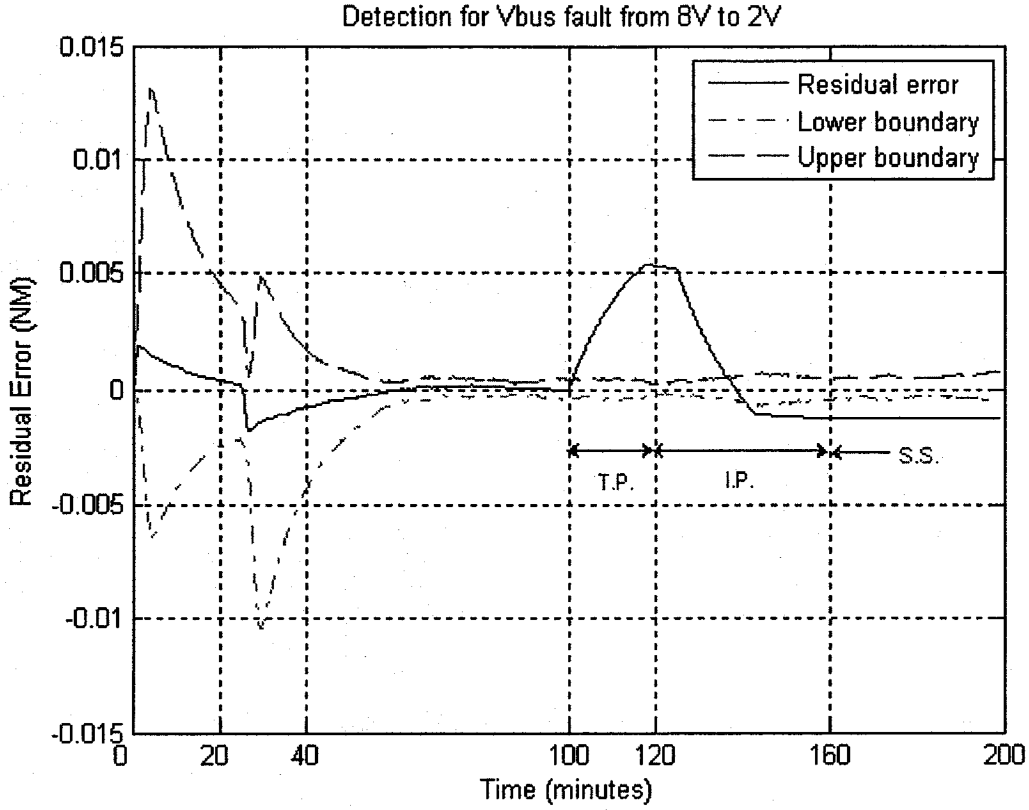

Figure 4.1 The detection result for a bus voltage fault

For the sake of convenience, in this thesis the term *transient phase* (T.P.) is used to describe the time period from the period t = 100 minutes (the beginning of the fault occurrence) to t = 120 minutes, the term *intermediate phase* (I.P.) is used to describe the period from t =120 minutes to t = 160 minutes and the term *steady state* (S.S.) is used to denote the time t > 160 minutes, as shown explicitly on Figure 4.1.

Table 4.1 and Table 4.2 summarize the results for a number of detection scenarios for satellite operating at different setpoints. In these tables D denotes Detected and UD denotes Undetected. Table 4.1 depicts the results whose threshold is generated by the recall network approach with the following detailed procedure as follows:

- First, a recall network is constructed with weights and delays obtained from a training network.

- This network is then used to generate the residual errors under normal operation for each individual setpoint from 2.5 deg to 7.5 deg.
- Next, the mean value of the residual error curves and their standard deviations are computed.
- Finally, the proper parameters for the threshold curves in which α is capable of providing false alarm free detection for as many setpoints as possible are obtained. For Table 4.1, the range for α is chosen as [-5, 3]. If the upper boundary parameter is increased to 5 (for the sake of symmetry with the lower boundary), the detection results will be slightly different from those shown in Table 4.1. These results are not shown here.

Table 4.1 The Voltage fault detection results based on the recall approach

		8V –0V	8V –0.5V	8V –2V	8V –3V	8V–4V	Average
2.5 Deg	T. P.	D with 4.4min delay	UD	D with 0.7min delay	D with 2.2min delay	UD	2.4333 min
	I. P.	D but ambiguous from t = 125.2 min to t = 131.3 min (6.1mins)	D after t = 155.3 min (35.3 min delay)	D but ambiguous from t = 138.4 min to t = 142.7 min (4.3mins)	D but ambiguous from t = 129.6 min to t = 160.9 min (31.5mins)	UD	19.3 min
	S. S.	D	D	D	D	D with 15.5 min delay t = 175.5 min	3.1 min
3 Deg	T. P.	D with 3.1 min delay	UD	D with 0.5min delay	D with 1.6 min delay	UD	1.7333 min
	I. P.	D but ambiguous from t = 122.8 min to t = 127.9 min (5.1mins)	D after t = 147.5 mins (27.5min)	D but ambiguous from t = 137.2 min to t = 141.1 mins (3.9mins)	D but ambiguous from t = 126.1 min to t = 152.6 min (26.5mins)	UD	15.7500 min

	S. S.	D	D	D	D	D with 5.3min delay t =165.3min	5.3/5 = 1.06min
3.5 Deg	T. P.	D with 3.8min delay	UD	D with 0.6min delay	D with 1.9 min delay	UD	2.1min
	I. P.	D but ambiguous from t = 120.3 min to t = 27.3 min (7mins)	D after T =151.1mins (31.1min)	D but ambiguous from t = 137.2 min to t = 141 min (3.8mins)	D but ambiguous from t = 126.4 min to t = 152.5 min (26.1 mins)	UD	17 min
	S. S.	D	D	D	D	D with. 15.2min delay t = 175.2min	15.2/5 = 3.04 min
4 Deg	T. P.	D with 2.9min delay	UD	D with 0.6 min delay	D with 1.8min delay	UD	1.7667 min
	I. P.	D but ambiguous from t = 122 min to t = 127.3 min (5.3mins)	D after t = 151.1 min (31.1min)	D but ambiguous from t = 137.3 min to t = 141 min (3.7mins)	D but ambiguous from t = 126.8 min to t = 150.5 min (23.7 mins)	UD	15.9500 min
	S. S.	D	D	D	D	D with 4.4 mins delay t = 164.4 min but ambiguous after t = 188.9 min	4.4/5 = 0.88min
4.5 Deg	T. P.	D with 6.3 mins delay	UD	D with 0.8min delay	D with 2.8mins delay	UD	3.3min
	I. P.	D but ambiguous from t = 122.6 min to t = 127.6min (5mins)	D after t = 151.9mins (31.9)	D but ambiguous from t = 137.1min to t = 140.7min (3.6 mins)	D but ambiguous from t = 126.1min to t = 151.7min (25.6 mins)	UD	16.525 min
	S. S.	D	D	D	D	D with 12.3min delay t = 172.3mins	12.3/5 = 2.46 min

5 Deg	T. P.	D with 3 min delay	UD	D with 0.7min delay	D with 1.7 min delay	UD	1.8min
	I. P.	D but ambiguous from t =123.8 min to t =128.4 min (4.6mins)	D after t= 148mins (28min).	D but ambiguous from t = 137.1 min to t = 140.7 min (3.6mins)	D but ambiguous from t = 126.3 min to t = 152.4 min (26.1mins)	UD	15.575 min
	S. S.	D	D	D	D	D with 9.2min delay t = 169.2mins	9.2/5 = 1.84min
5.5 Deg	T. P.	D with 2.6mins delay	UD	D with 0.5 min delay	D with 1.7min delay	UD	1.6min
	I. P.	D but ambiguous from t =123.2 min to t =127.9min (4.7mins)	D after t = 154.9mins (34.9min)	D but ambiguous from t =137.12 min to t =140.7 min (3.6mins)	D but ambiguous from t =126.7 min to t =152.4 min (25.7mins)	UD	17.225min
	S. S.	D	D	D	D	D with 12.1mins delay t= 172.1mins	12.1/5 = 2.42min
6 Deg	T. P.	D with 3.2min delay	UD	D with 0.7min delay	D with 2.2 min delay	UD	2.0333min
	I. P.	D but ambiguous from t =122.9 min to t =127.9 min (5mins)	D after t = 145.8 min (25.8min)	D but ambiguous from t =137 min to t =140.7 min (3.7mins)	D but ambiguous from t =126.5 min to t =152.1 min (25.6mins)	UD	15.0250 min
	S. S.	D	D	D	D	D with 12.1 mins delay t =172.1mins	12.1/5 = 2.42min
6.5 deg	T. P.	D with 0.2mins delay	UD	D with 0.2mins delay	D with 0.2 mins delay	UD	0.2min
	I. P.	D but ambiguous from t =122.9 min to t =127.9 min (5mins)	D after t = 152.1mins (32.1min)	D but ambiguous from t =137.1 min to t =141 min (3.9mins)	D but ambiguous from t =126.4 min to t =153.3 min (26.9mins)	UD	16.9750 min

	S. S.	D	D	D	D	D with 14.5mins delay t = 174.5min	14.5/5 = 2.9min
7 deg	T. P.	D with 1.9mins delay	UD	D with 0.4mins delay	D with 0.9mins delay	UD	1.0667 min
	I. P.	D but ambiguous from t =122.9 min to t =127.6 min (4.7mins)	D after t = 152.4mins (32.4min)	D but ambiguous from t =136.9 min to t =140.7 min (3.8mins)	D but ambiguous from t =126.3 min to t =152.2 min (25.9mins)	UD	16.7 min
	S. S.	D	D	D	D	D with 14.3mins delay t = 174.3min	14.3/5 = 2.86min
7.5 deg	T. P.	D with 2.9mins delay	UD	D with 0.6mins delay	D with 1.7mins delay	UD	1.7333min
	I. P.	D but ambiguous from t =121 min to t =127.4 min (6.4mins)	D after t= 154.2mins (34.2min)	D but ambiguous from t =137.1 min to t =141 min (3.9mins)	D but ambiguous from t =126.1 min to t =151.9 min (25.8mins)	UD	17.5750 min
	S. S.	D	D	D	D	D with 11.6mins delay t = 171.6min	11.6/5 = 2.32min

Table 4.2 summarizes the detection results by using the thresholds that are gathered from the training phase approach with its corresponding procedure are given by:

- First, generate the residual errors that are obtained from the training under normal operations but only maps two setpoints. For the sake of convince, call them residual error1. Since the training phase is based on 20 different initial conditions, there are 20 residual error1s available.
- Find the mean value of the residual error1 curves and their standard deviation.
- Construct a recall network with weights and delays as obtained from the training phase described above.

- Use the recall network to generate the residual error under normal operation for each setpoint of the scenarios. For the sake of convince, call them residual error2.
- Finally, find the proper parameters for the threshold in which α is capable of providing false alarm free detection for all those residual error2 curves. For Table 4.2, the range for α is set to [-9, 25].

Table 4.2 The V_{bus} fault detection results (threshold based on the training approach)

		8V –0V	8V –0.5V	8V –2V	8V–3V	8V –4V	Average
2.5 Deg	T. P.	D with 9.3min delay	UD	D with 2.5min delay	D with 8.7min delay	UD	6.8333 min
	I. P.	D but ambiguous from t = 121.2 min to t = 134.7 min (13.5mins)	UD	D but ambiguous from t = 133.8 min to t = 142.5 min (8.7mins)	D but ambiguous from t = 126.8 min to t = 167.4 min (40.6mins)	UD	20.9333 min
	S. S.	D	D with 1.9 min	D	D	D with 21 min delay t = 181 min	4.58 min
3 Deg	T. P.	D with 13.9 min delay	UD	D with 2.4min delay	D with 9 min delay	UD	8.4333 min
	I. P.	D but ambiguous from t = 120.6 min to t = 127.8 min (7.2mins)	UD	D but ambiguous from t = 133.3 min to t = 141.5 min (8.2mins)	D but ambiguous from t = 121.5 min to t = 160.4min (38.9mins)	UD	18.1 min
	S. S.	D	D	D	D	D with 9.7 min delay t = 169.7min	9.7/5 = 1.94mins
3.5 Deg	T. P.	D with 9.3min delay	UD	D with 2.4 min delay	D with 8.8 min delay	UD	6.8333 min

	I. P.	D but ambiguous from t = 119.3 min to t = 127.3 min (8mins)	UD	D but ambiguous from t = 133.3 min to t = 141.4 min (8.1mins)	D but ambiguous from t = 121.8 min to t = 160.3 min (38.5mins)	UD	18.2 min
	S. S.	D	D	D	D	D with 20 min delay t = 180min	20/5 = 4 min
4 Deg	T. P.	D with 13min delay	UD	D with 2.4 min delay	D with 9 min delay	UD	8.1333 min
	I. P.	D but ambiguous from t = 120 .5 min to t = 127.4 min (6.9mins)	UD	D but ambiguous from t = 133.3 min to t = 141.4 min (8.1mins)	D but ambiguous from t = 121.5 min to t = 159.8min (38.3 min)	UD	17.7667 min
	S. S.	D	D	D	D	D with 4.5 min delay t= 164.5 but ambiguous after t = 188 min	4.5/5 = 0.9mins
4.5 Deg	T. P.	D with 14.2 mins delay	UD	D with 2.8min delay	D with 9.1mins delay	UD	8.7 min
	I. P.	D but ambiguous from t = 120.5 min to t = 127.5min (7mins)	UD	D but ambiguous from t = 133.2 min to t = 141.3min (8.1 min)	D but ambiguous from t = 121.1 min to t = 160.2 min (39.1 min)	UD	18.0667 min
	S. S.	D	D	D	D	D with 17.8 min delay t = 177.8	17.8/5 = 3.56 min
5 Deg	T. P.	D with 13.8 min delay	UD	D with 2.3min delay	D with 8.8 min delay	UD	8.3 min

	I. P.	D but ambiguous from t =120.8 min to t =128 min (7.2mins)	UD	D but ambiguous from t = 133.2 min to t = 141.3 min (8.1mins)	D but ambiguous from t = 121.7 min to t = 160.4 min (38.7mins)	UD	18 min
	S. S.	D	D with 0.2mins delay	D	D	D with 15.5min delay t = 175.5 min	3.14 min
5.5 Deg	T. P.	D with 9.2 mins delay	UD	D with 2.3 min delay	D with 8.7min delay	UD	6.7333 min
	I. P.	D but ambiguous from t =120.5 min to t =127.7 min (7.2mins)	UD	D but ambiguous from t =133.2 min to t =141.3min (8.1mins)	D but ambiguous from t =121.7 min to t =160.3 min (38.6mins)	UD	17.9667 min
	S. S.	D	D with 0.2 min delay t = 160.2 min	D	D	D with 10.8mins delay t = 170.8 min	2.2 min
6 Deg	T. P.	D with 9.2min delay	UD	D with 2.4min delay	D with 8.8 min delay	UD	6.8 min
	I. P.	D but ambiguous from t =120.5 min to t =127.8 min (7.3mins)	UD	D but ambiguous from t =133.2 min to t =141.3 min (8.1mins)	D but ambiguous from t =121.8 min to t =160.3 min (38.5mins)	UD	17.9667 min
	S. S.	D	D	D	D	D with 16.6 min delay T =176.6	16.6/5 = 3.32 min
6.5 deg	T. P.	D with 9mins delay	UD	D with 2.1mins delay	D with 5.6mins delay	UD	5.5667 min

	I. P.	D but ambiguous from t =120.5 min to t =127.7 min (7.2mins)	UD	D but ambiguous from t =133.2 min to t =141.5 min (8.3mins)	D but ambiguous from t =121.7 min to t =160.5 min (38.8mins)	UD	18.1 min
	S. S.	D	D	D	D	D with 17.4 mins delay t = 177.4 min	3.5 min
7 deg	T. P.	D with 9.2mins delay	UD	D with 2.2mins delay	D with 8.7mins delay	UD	6.7 min
	I. P.	D but ambiguous from t =120.6 min to t =127.6 min (7mins)	UD	D but ambiguous from t =133.2 min to t =141.3 min (8.1mins)	D but ambiguous from t =121.8 min to t =160.3 min (38.5mins)	UD	17.8667 min
	S. S.	D	D with 0.4mins delay t = 160.4 min	D	D	D with 17.4mins delay t = 177.4 min	3.56 min
7.5 deg	T. P.	D with 18.4 mins delay	UD	D with 1.9mins delay	D with 9.6mins delay	UD	9.9667 min
	I. P.	D but ambiguous from t =119.9 min to t =127.2 min (7.3mins)	UD	D but ambiguous from t =133.2 min to t =141.3 min (8.1mins)	D but ambiguous from t =122.1 min to t =160.0 min (38.9mins)	UD	18.1 min
	S. S.	D	D with 0.6mins delay t = 160.6 min	D	D	D with 17.4 mins delay t = 177.4 min	3.6 min

By comparing results shown in Table 4.1 and 4.2, one may arrive at the observation that the detection results are consistent with each other although the results obtained based on the recall approach still has a slight advantage in the sense that the delay time is shorter and the parameters (the window size, the value of α) for composing the thresholds are smaller.

4.1.1.2. Reliability Analysis for V_{bus} Faults

The detection results for large V_{bus} faults have been studied in the previous subsection. Clearly, the proposed neural network is capable of detecting large faults for different setpoints. However, for small V_{bus} signal changes in which the magnitude is less than 4V, for instance 8V to 5V, the input used for the network recall is not significantly changes by the fault occurrence; therefore, the network is unable to detect it as a fault regardless of which setpoint the fault has occurred. The simulation results are provided in Figure 4.2.

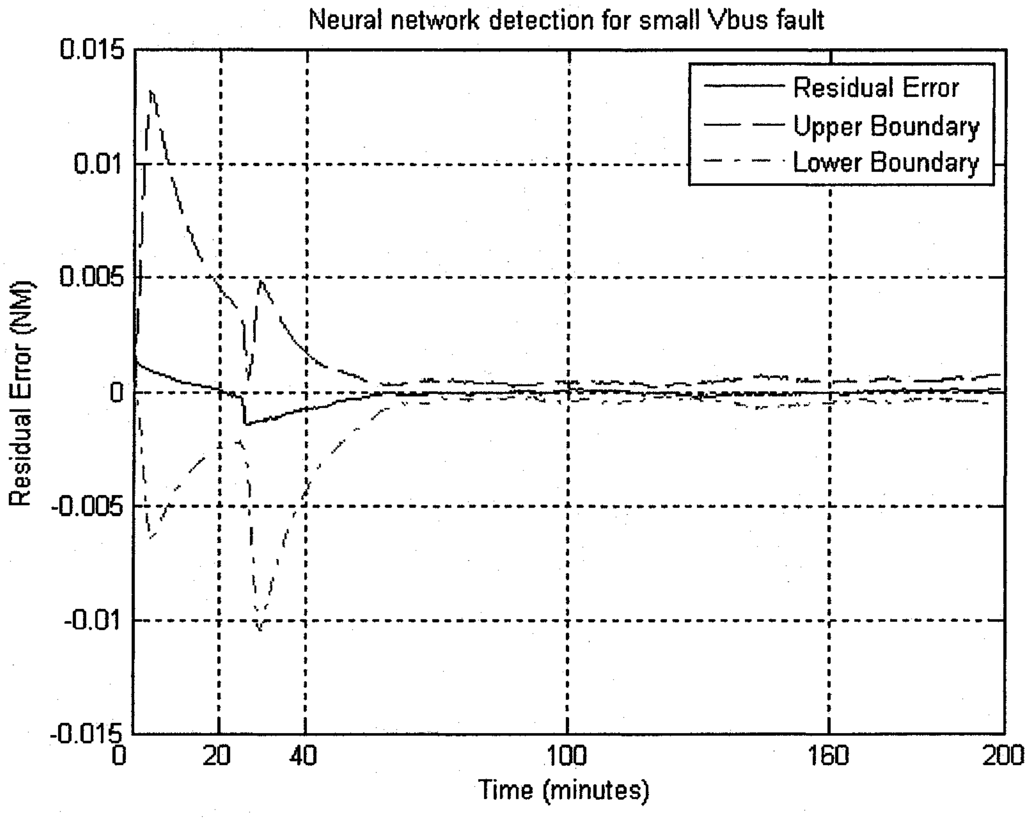

Figure 4.2 The neural network is unable to detect small V_{bus} change from 8V to 5V

In Figure 4.2, the fault has occurred at t = 100 minutes, however, the residual error did not cross over the thresholds even after its steady state t >= 160 minutes.

Therefore, the neural network has failed to detect this fault. This detection result is applicable to any fault whose magnitude is less than 4V. To investigate the reason for this behavior, the input to the neural network for small faults is studied and the results are shown in Figure 4.3 to Figure 4.5.

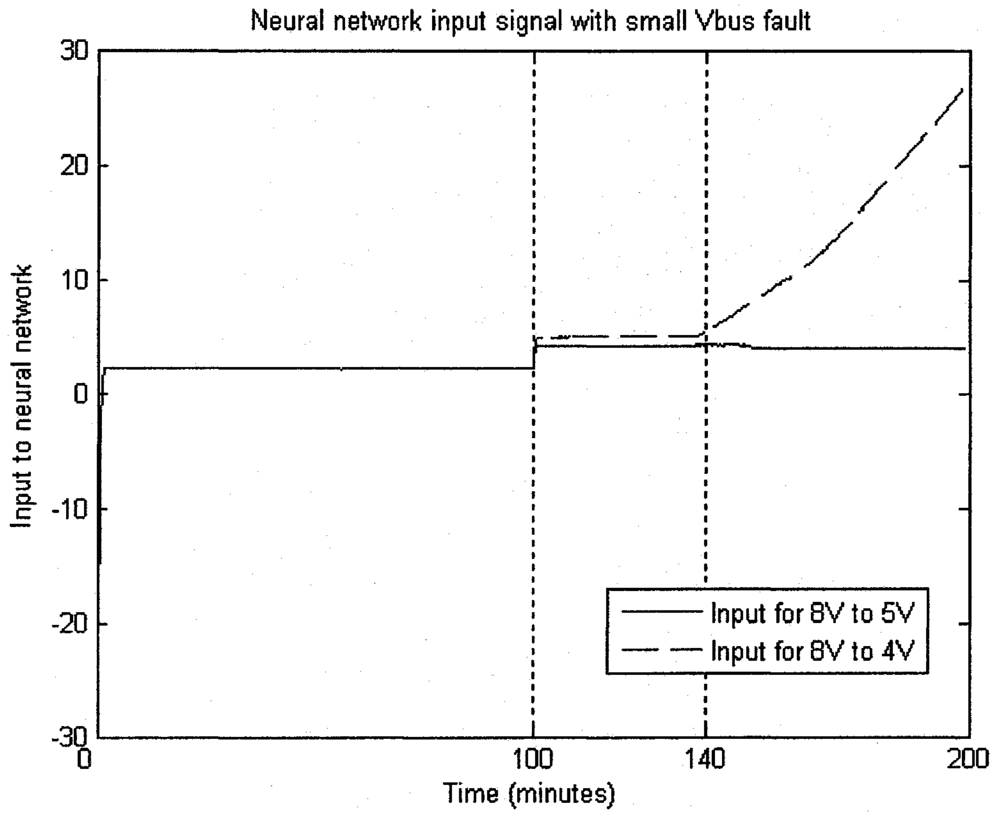

Figure 4.3 The neural network inputs for the 8V to 5V and 8V to 4V V_{bus} faults

Figure 4.3 shows the inputs to the network which reveals that even though the V_{bus} faults from 8V to 5V and 8V to 4V are not significantly different, the impacts they cause to the network input are significantly different. The input to the network for the 8V to 4V V_{bus} fault increases alike a ramp function after the transient period while the input for the 8V to 5V V_{bus} fault behaves like a step function. Consequently, the detection capability for these two changes is different as shown in Figure 4.4. For the 8V to 4V

V_{bus} fault, after the input has experienced a transient phase (t >= 140 minutes), the input does increase. The neural network has detected this change with about 20 minutes of delay, after t = 160 minutes, when the residual signal has exceeded the threshold boundary. Since the input for the 8V to 5V V_{bus} fault did not change much, the residual signal has remained inside the threshold boundary.

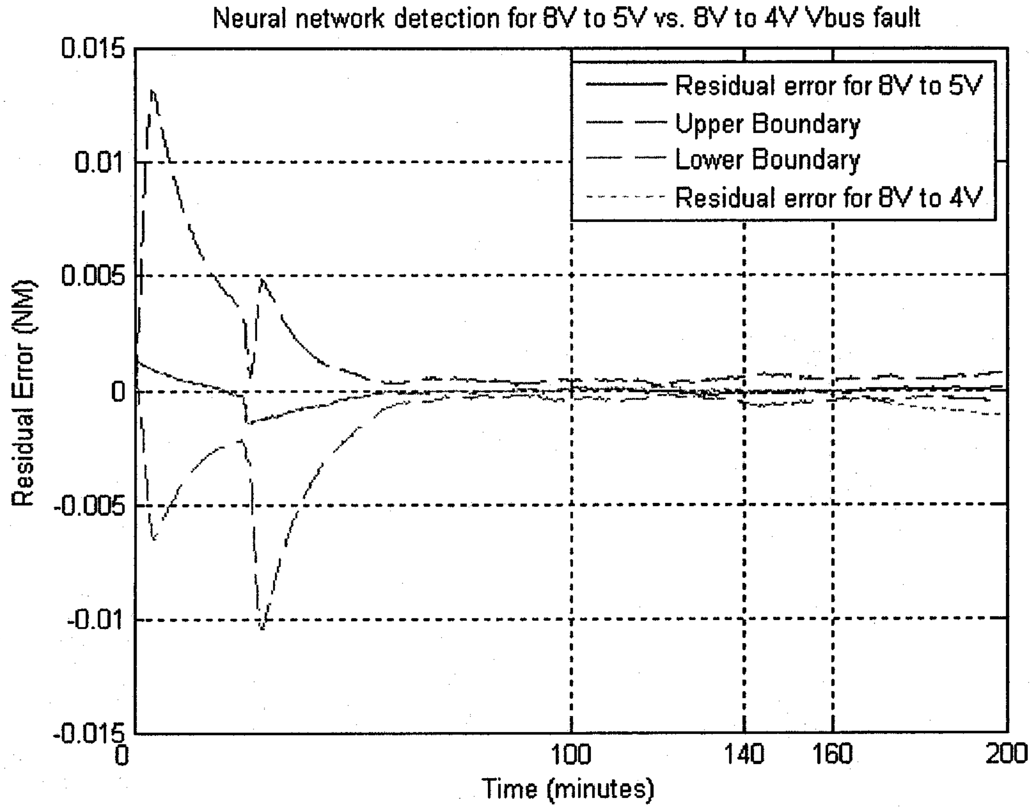

Figure 4.4 Detection for the 8V to 5V vs. 8V to 4V V_{bus} fault

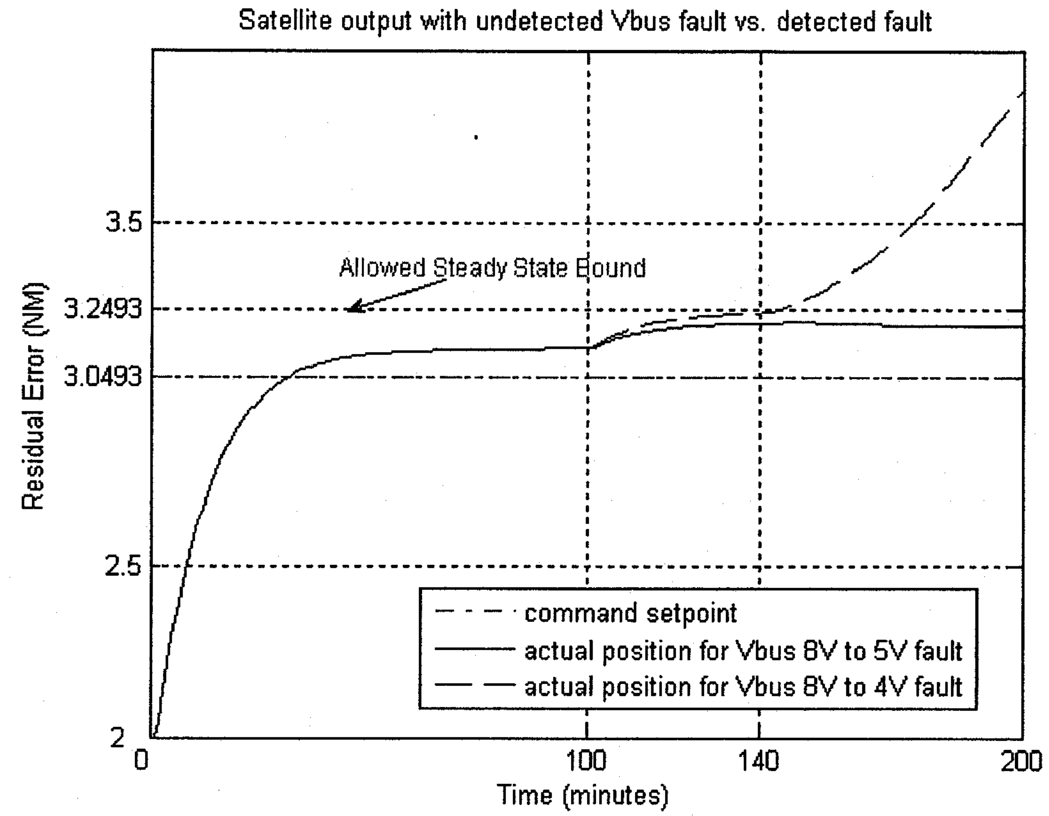

Figure 4.5 The satellite position change for the 8V to 5V vs. 8V to 4V V_{bus} fault

Figure 4.5 shows the satellite behavior under these two fault scenarios. Before the fault occurrence, the satellite behavior is undistinguishable and with the fault occurring at t = 100 minutes, one curve deviates from its orbit rapidly (after t = 140 minutes) while the other one just deviates away a small distance which is still acceptable and within the design steady state error accuracy requirement.

From this example it follows that small faults do not result in the satellite behaving sufficiently abnormally. However, this does not imply that one changes the definition of a fault. It is not true that small V_{bus} faults can not contribute to the satellite abnormal operation. Under some circumstances, for instance when V_{bus} fault has occurred simultaneously with another fault, a small V_{bus} fault is still capable of causing detectable influence. In some cases, it also has the tendency to “explode” with increase in time and eventually cause a failure to the entire system.

From the FDI scheme point of view, it is acceptable if the network fails to detect a small V_{bus} fault as long as it is unable to cause difficulty for the entire system. However, the more important aspect is that the neural network does not fail to detect a fault which can potentially cause the satellite system to behave abnormally.

4.1.2. Current Fault Studies

4.1.2.1. Fault Detection for Current Drop

Generally, current fault does exert more influence on the wheel than the voltage drop. The neural network is capable of detecting current faults from 1A to 0A, 0.1A, 0.2A, 0.3A and 0.4A. A typical current fault detection is shown in Figure 4.6.

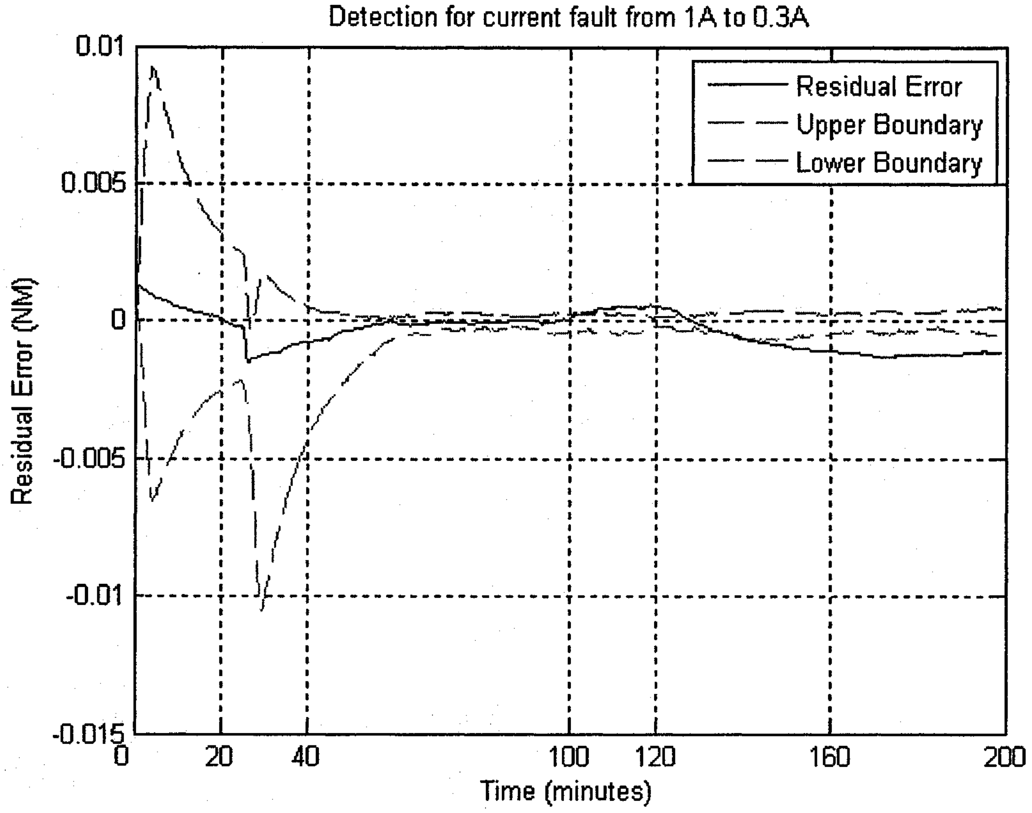

Figure 4.6 The detection results for the wheel current fault

Table 4.3 and Table 4.4 present a number of current fault detection results for the satellite operating at different setpoints. Table 4.3 shows the result when the threshold is generated from the recall approach network and the procedure is the same as in Table 4.1 (see page 69).

Table 4.3 Detection results for current faults (threshold is based on the recall approach)

		1A-0A	1A-0.1A	1A-0.2A	1A-0.3A	1A-0.4A	Average
2.5 Deg	T. P.	D with 0.9min delay	D with 1.2min delay	D with 1.9min delay	D with 4min delay	UD	2 min
	I. P.	D but ambiguous from t = 146.7 min to t = 154.6 min (7.9mins)	D but ambiguous from t = 145.7 min to t = 160.6 min (14.9mins)	D but ambiguous from t = 132.8 min to t = 164.7 min (31.9mins)	D but ambiguous from t = 129.3 min to t = 151.5 min (22.2mins)	D after t =127.6 min 7.6mins	16.9 min
	S. S.	D	D	D	D	D	0 min

3 Deg	T. P.	D with 0.6min delay	D with 0.8min delay	D with 1.1min delay	D with 2.8min delay	UD	1.3250 min
	I. P.	D but ambiguous from t = 143.3 min to t = 149.3 min (6mins)	D but ambiguous from t = 141.9 min to t = 154.9 min (13mins)	D but ambiguous from t = 130.2 min to t = 151.5 min (21.3mins)	D but ambiguous from t = 125.7 min to t = 137 min (11.3mins)	D after t = 125 mins (5 min)	11.3200 min
	S. S.	D	D	D	D	D	0
3.5 Deg	T. P.	D with 0.8 min delay	D with 1.0 min delay	D with 1.5 min delay	D with 3.2 min delay	UD	1.6250 min
	I. P.	D but ambiguous from t = 143.2 min to t = 150.2 min (7mins)	D but ambiguous from t = 141.8 min to t = 154.9 min (13.1mins)	D but ambiguous from t = 130.1 min to t = 150.9 min (20.8mins)	D but ambiguous from t = 125.2 min to t = 134.4 min (9.2mins)	D after t= 126.3 mins (6.3 min)	11.2800 min
	S. S.	D	D	D	D	D	0 min
4 Deg	T. P.	D with 0.7min delay	D with 0.8min delay	D with 1.4min delay	D with 2.4min delay	UD	1.3250 min
	I. P.	D but ambiguous from t = 142.9 min to t = 149.3 min (6.4mins)	D but ambiguous from t = 141.4 min to t = 154.9.min (13.5mins)	D but ambiguous from t = 129.9 min to t = 150.7 min (20.8mins)	D but ambiguous from t = 125.6 min to t = 136.9 min (11.3mins)	D after t = 126.2 mins (6.2min)	11.6400 min
	S. S.	D	D	D	D	D	0
4.5 Deg	T. P.	D with 1.0mins delay	D with 1.3mins delay	D with 2.4min delay	D with 6.3mins delay	UD	2.7500 min
	I. P.	D but ambiguous from t = 143.5 min to t = 150.5 min (7mins)	D but ambiguous from t = 141.7 min to t = 154.9 min (13.2mins)	D but ambiguous from t = 130.2 min to t = 151.5 min (21.3mins)	D but ambiguous from t = 125.2 min to t = 143.4min (18.2 min)	D after t= 126.3 mins (6.3 min)	13.2000 min
	S. S.	D	D	D	D	D	0min

5.0 Deg	T. P.	D with 0.8min delay	D with 0.9 min delay	D with 1.5min delay	D with 2.8 min delay	UD	1.5000 min
	I. P.	D but ambiguous from t =143.3 min to t =150.4 min (7.1mins)	D but ambiguous from t = 141.7 min to t = 154.9 min (13.2mins)	D but ambiguous from t = 130.3 min to t = 150.4 min (20.1mins)	D but ambiguous from t = 125.8 min to t = 141.5 min (15.7mins)	D after t = 127.3 min (7.3min)	12.6800 min
	S. S.	D	D	D	D	D	0
5.5 Deg	T. P.	D with 0.7min delay	D with 0.8min delay	D with 1.1 min delay	D with 1.9 min delay	UD	1.1250 min
	I. P.	D but ambiguous from t =142.4 min to t =149.0 min (6.6mins)	D but ambiguous from t =141.4 min to t =154.9 min (13.5mins)	D but ambiguous from t =129.9 min to t =150.9 min (21mins)	D but ambiguous from t =125.6 min to t =137.3 min (11.7mins)	D after t =127.0 min (7min)	11.9600 min
	S. S.	D	D	D	D	D	0
6 Deg	T. P.	D with 0.8 min delay	D with 1.0min delay	D with 1.8 min delay	D with 3.2 min delay	UD	1.7000 min
	I. P.	D but ambiguous from t =142.9 min to t =150.7 min (7.8mins)	D but ambiguous from t =141.9 min to t =154.9min (13mins)	D but ambiguous from t =130.1 min to t =150.9 min (20.8mins)	D but ambiguous from t =125.9 min to t =136.9 min (11mins)	D after t =126.3 min (6.3min)	11.7800 min
	S. S.	D	D	D	D	D	0
6.5 deg	T. P.	D with 0.2 min delay	D with 0.2 min delay	D with 0.2 min delay	D with 0.2 min delay	D with 0.2 min delay	0.2 min
	I. P.	D but ambiguous from t =142.9 min to t =149.2 min (6.3mins)	D but ambiguous from t =142.5 min to t =155.2min (12.7mins)	D but ambiguous from t =130.2 min to t =150.9 min (20.7mins)	D but ambiguous from t =125.3 min to t =132.5 min (7.2mins)	D but ambiguous from t =120.5 min to t =128.2 min (7.7mins)	10.9200 min
	S. S.	D	D	D	D	D	0

7 deg	T. P.	D with 0.5min delay	D with 0.6 min delay	D with 0.8min delay	D with 1.8 min delay	UD	0.9250 min
	I. P.	D but ambiguous from t=143.1 min to t=149.3 min (6.2mins)	D but ambiguous from t=142.4 min to t=155.3 min (12.9mins)	D but ambiguous from t=130.2 min to t=151.6 min (21.4mins)	D but ambiguous from t=125.1 min to t=131.5 min (6.4mins)	D after t=120.8 min (0.8min)	9.5400 min
	S. S.	D	D	D	D	D	0
7.5 deg	T. P.	D with 0.8 min delay	D with 1.0min delay	D with 1.4 min delay	D with 2.8 min delay	UD	1.5000 min
	I. P.	D but ambiguous from t=143.1 min to t=150 min (6.9mins)	D but ambiguous from t=141.4 min to t=154.9 min (13.5mins)	D but ambiguous from t=130.2 min to t=151.5 min (21.3mins)	D but ambiguous from t=125.3 min to t=136.9 min (11.6mins)	D after t=127.1 min (7.1min)	12.0800 min
	S. S.	D	D	D	D	D	0

Table 4.4 shows the results when the threshold is generated from the training phase approach and the procedure is the same as in Table 4.2 (see page 73).

Table 4.4 Detection results for current faults (threshold is based on the training approach)

		1A-0A	1A-0.1A	1A-0.2A	1A-0.3A	1A-0.4A	Average
2.5 Deg	T. P.	D with 3.9min delay	D with 5min delay	D with 6.5min delay	D with 13.8min delay	UD	7.3000 min
	I. P.	D but ambiguous from t= 138.4 min to t= 160.8 min (22.4mins)	D but ambiguous from t= 134.1 min to t= 163.7 min (29.6mins)	D but ambiguous from t= 129.9 min to t= 161 min (31.1mins)	D but ambiguous from t= 122.2 min to t= 161 min (22.2mins)	UD	26.3250 min
	S. S.	D	D	D	D	D	0

3 Deg	T. P.	D with 3.4min delay	D with 5min delay	D with 6.4min delay	D with 13.2min delay	UD	7min
	I. P.	D but ambiguous from t = 136.9 min to t = 153.9 min (17mins)	D but ambiguous from t = 132.9 min to t = 160.9 min (13mins)	D but ambiguous from t = 127.9 min to t = 161.3 min (33.4mins)	D but ambiguous from t =122.1 min to t =154.9 min (32.8mins)	D after t = 124.9 min (4.9min)	20.2200 min
	S. S.	D	D	D	D	D	0
3.5 Deg	T. P.	D with 3.9min delay	D with 5min delay	D with 6.3 min delay	D with 14.8 min delay	UD	7.5000 min
	I. P.	D but ambiguous from t = 137 min to t = 153.9 min (16.9mins)	D but ambiguous from t = 132.7 min to t = 161 min (28.3mins)	D but ambiguous from t = 127.8 min to t = 161.2 min (33.4mins)	D but ambiguous from t = 121.8 min to t = 153.8 min (32mins)	D after t= 125.9 min (5.9min)	23.3000 min
	S. S.	D	D	D	D	D	0
4 Deg	T. P.	D with 3.9min delay	D with 5min delay	D with 7.1min delay	D with 14.1min delay	UD	7.5250 min
	I. P.	D but ambiguous from t = 137 min to t = 152.2min (15.2mins)	D but ambiguous from t = 132.7 min to t = 161min (28.3mins)	D but ambiguous from t = 127.7 min to t = 161.3 min (33.6mins)	D but ambiguous from t = 122.1 min to t = 160.4min (38.3mins)	D after t = 25.6 mins (5.6min)	24.2000 min
	S. S.	D	D	D	D	D	0
4.5 Deg	T. P.	D with 4.2mins delay	D with 5.1mins delay	D with 7mins delay	D with 14.9mins delay	UD	7.32mins
	I. P.	D but ambiguous from t = 137.1 min to t = 155min (17.9mins)	D but ambiguous from t = 132.8 min to t = 160.9 min (28.1mins)	D but ambiguous from t = 127.8 min to t = 161.2 min (33.4mins)	D but ambiguous from t = 121.8 min to t = 160min (38.3mins)	D after t= 154 mins	11.2mins
	S. S.	D	D	D	D	D	1.22mins

5 Deg	T. P.	D with 3.9min delay	D with 5 min delay	D with 6.3min delay	D with 13.8 min delay	UD	7.2500 min
	I. P.	D but ambiguous from t =137.1 min to t =154.8 min (17.7mins)	D but ambiguous from t = 132.8 min to t = 161.9 min (28.1mins)	D but ambiguous from t = 127.8 min to t = 161.2 min (33.4mins)	D but ambiguous from t = 122.1 min to t = 160.1min (38mins)	D after t = 126.5 min (6.5min)	24.7400 min
	S. S.	D	D	D	D	D	0
5.5 Deg	T. P.	D with 3.2min delay	D with 4.9min delay	D with 6.2 min delay	D with 9.8 min delay	UD	6.0250 min
	I. P.	D but ambiguous from t =136.9 min to t =151.1 min (14.2mins)	D but ambiguous from t =132.8 min to t =161 min (28.2mins)	D but ambiguous from t =127.7 min to t =161.3 min (33.6mins)	D but ambiguous from t =122.1 min to t =160.3 min (38.2mins)	D after t =154.2 min (34.2min)	29.6800 min
	S. S.	D	D	D	D	D	0
6 Deg	T. P.	D with 3.9 min delay	D with 5 min delay	D with 6.4 min delay	D with 14.8 min delay	D with 18.9mins delay	9.8000 min
	I. P.	D but ambiguous from t =136.9 min to t =154.1 min (17.2mins)	D but ambiguous from t =132.9 min to t =161 min (28.1mins)	D but ambiguous from t =127.7 min to t =162.7 min (35mins)	D but ambiguous from t =122 min to t =159.9 min (37.9mins)	D but ambiguous from t =119 min to t =160.2 min (41.2mins)	31.8800 min
	S. S.	D	D	D	D	D	0
6.5 deg	T. P.	D with 3 min delay	D with 4.8 min delay	D with 5.8 min delay	D with 9.5 min delay	D with 18.5mins delay	8.3200 min
	I. P.	D but ambiguous from t =136.9min to t =153.8 min (16.9mins)	D but ambiguous from t =132.9min to t =161min (28.1mins)	D but ambiguous from t =127.8 min to t =161.3 min (33.5mins)	D but ambiguous from t =122 min to t =160.3 min (38.3mins)	D but ambiguous from t =120 min to t =128.4 min (8.4mins)	25.0400 min
	S. S.	D	D	D	D	D	0
7 deg	T. P.	D with 3.2 min delay	D with 4.9 min delay	D with 6.3 min delay	D with 15.6min delay	D with 18.8mins delay	9.7600 min

	I. P.	D but ambiguous from t=137 min to t=154 min (17mins)	D but ambiguous from t=132.9 min to t=161min (28.1mins)	D but ambiguous from t=127.8 min to t=161.6min (33.8mins)	D but ambiguous from t=121.8 min to t=1539 min (32.1mins)	D but ambiguous from t=119 min to t=126.4min (7.4mins)	23.6800 min
	S. S.	D	D	D	D	D	0
7.5 deg	T. P.	D with 3.3 min delay	D with 4.9 min delay	D with 6.0min delay	D with 13.2 min delay	UD	6.8500 min
	I. P.	D but ambiguous from t=136.9 min to t=153.9 min (17mins)	D but ambiguous from t=132.9 min to t=161min (28.1mins)	D but ambiguous from t=127.9 min to t=161.2 min (33.3mins)	D but ambiguous from t=122 min to t=140.8 min (18.8mins)	D after t=154.8 min (34.8min)	26.4000 min
	S. S.	D	D	D	D	D	0

4.1.2.2. Reliability Analysis for Current Faults

As in the voltage fault scenario, the network is unable of detecting the current faults whose magnitude is less than the one 0.6 A change as shown in Figure 4.7. The reason is also the same as in the small voltage fault cases, which is the influences that are caused by small faults do not provide sufficient change for the neural network to generate detectable results. Therefore, the neural network fails to detect these faults as shown in Figure 4.8 and 4.9.

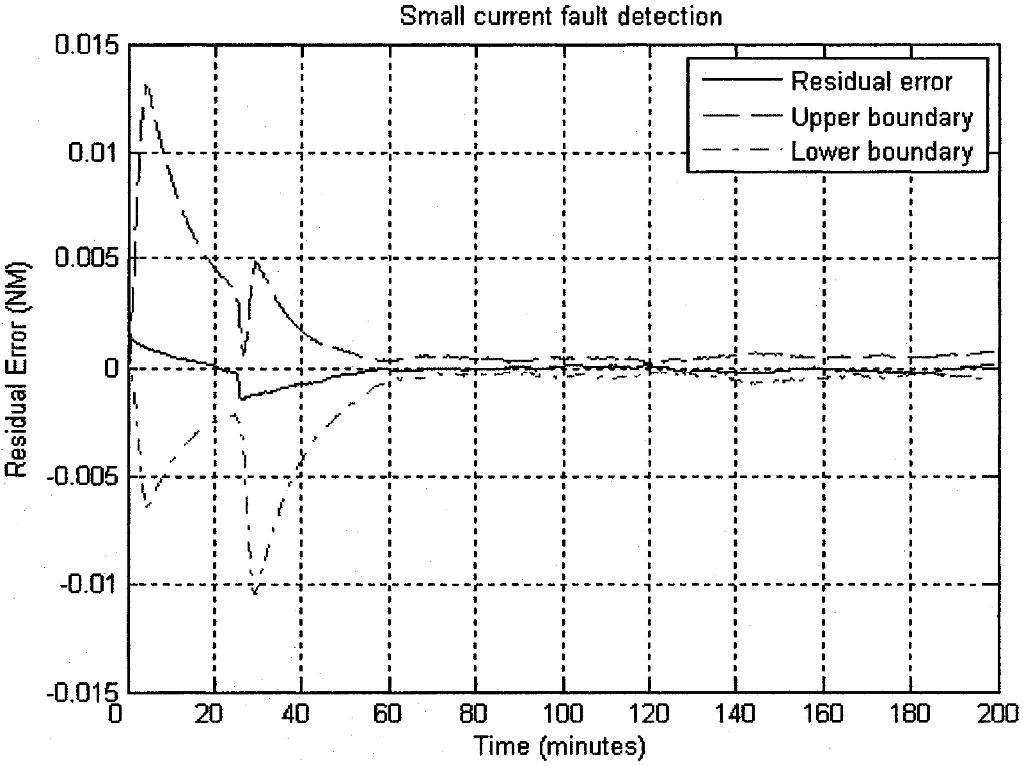

Figure 4.7 The small current fault is undetectable by the neural network

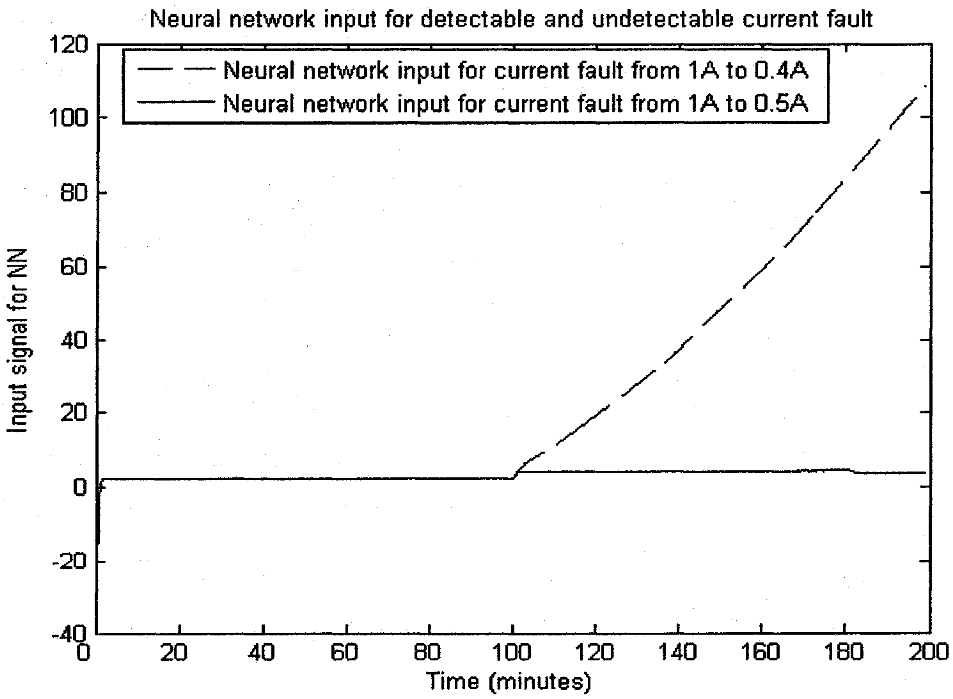

Figure 4.8 The neural network input resulting
from large detectable fault and small undetectable fault

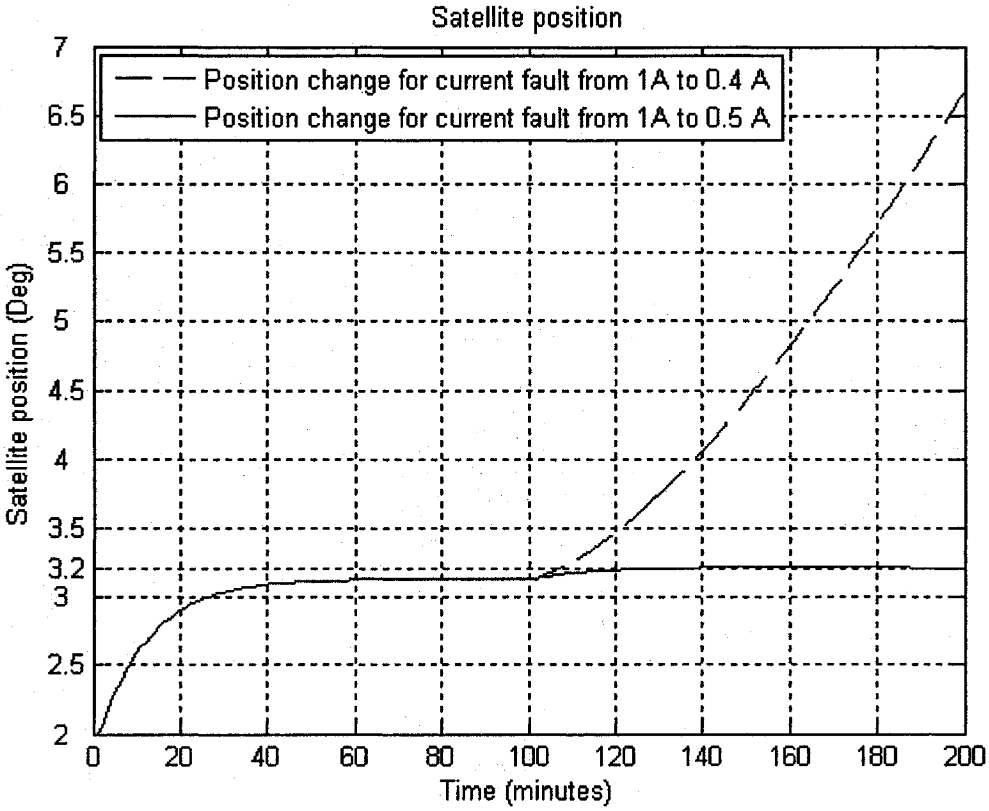

Figure 4.9 The satellite position changes corresponding to detectable and undetectable faults

4.1.3. Faults Isolation Study

Since the three axes are coupled together, the influences from the other two axes may affect the detection results in a given axis. Fortunately, the simulation results reveal that despite the operating conditions on the other axes affecting the performance on one axis somehow but they never cause or lead to a false alarm on that given axis. The fault isolation results for individual setpoints are illustrated below in Figure 4.10 to 4.12. The yaw axis is operating normally while the other two axes have faults in them. The detection results on the yaw axis show that the faults on the other axes do not lead to the yaw axis having any false alarms.

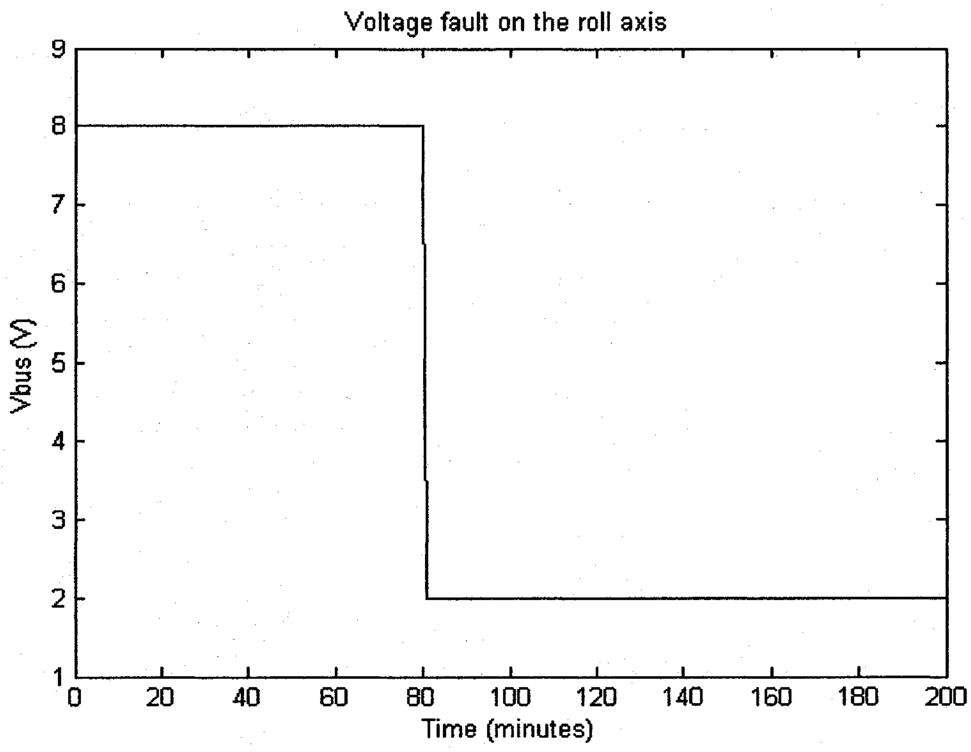

Figure 4.10 The V_{bus} fault on the roll axis at t = 80 minutes

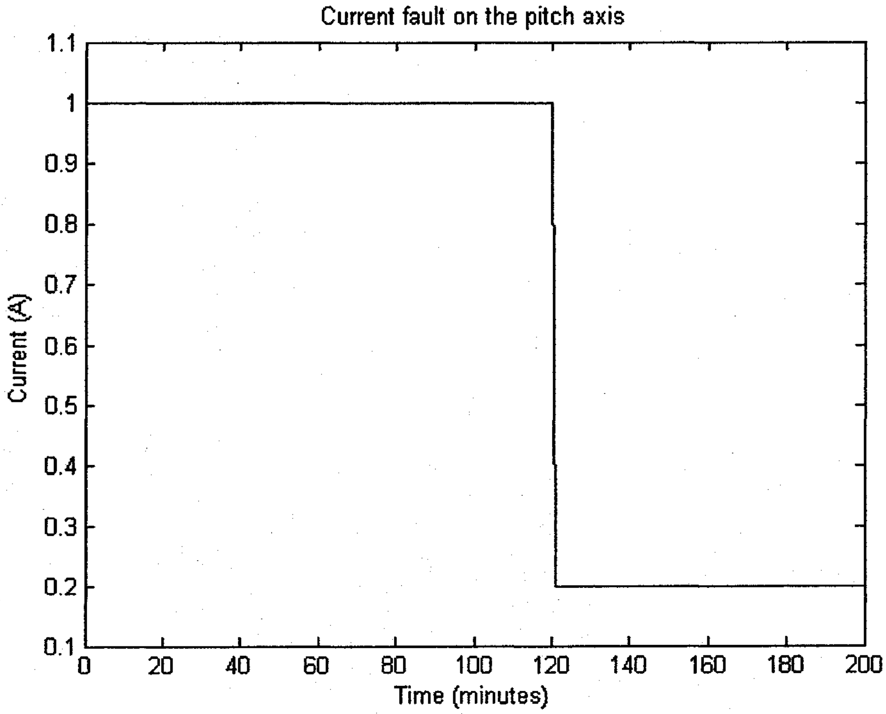

Figure 4.11 The current fault on the pitch axis at t = 120 minutes

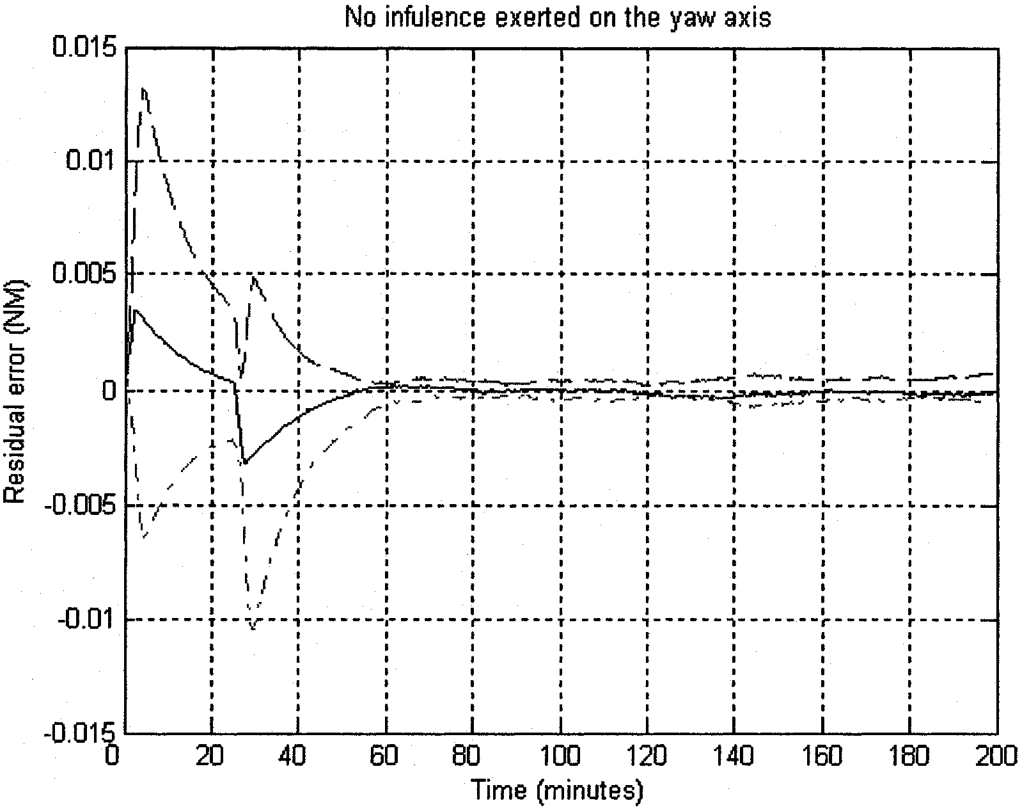

Figure 4.12 The yaw axis still operates under normal condition exhibiting no false alarms

Figure 4.12 shows that neither a single fault nor overlapped simultaneous faults on other axes can cause and lead to false alarm on the yaw axis. Similar results may also be observed for other axes and are not shown here.

4.1.4. Neural Network Robustness Study

In this subsection, the robustness of the neural network to external disturbances and noise is studied. The proposed neural network now operates under a fault free scenario but with higher noise levels. The case studies below are expected to measure the noise level that the neural networks can tolerate in providing false alarm free detection capability. The results are shown in Table 4.5.

Table 4.5 Robustness case studies for higher noise levels in all axes

Noise level increased by	3Deg	4 Deg	5 Deg

10%	Residual signal exceed from
	t =118.5 min to
	t = 119.2min (0.7 min) ;
	t = 128.6 min to
	t = 131.3 min (2.7 min);
	t = 131.8mins to
	t = 132.6min (0.6min) ;
	t =137.5min to
	t =137.8min (0.3mins);
	t =138min to
	t =138.4min (0.4mins);
	t =182.9min to
	t =186.5 min (3.6mins)
		Residual signal exceed from
		t = 115.9mins to
		t = 121.8mins (5.9mins);
		t = 185.2mins to
		t= 185.5mins (0.3 min);
			Residual signal exceed from
			t = 143.2mins to
			t = 144.7mins (1.5mins);
			t = 145.3mins to
			t = 146mins (0.7 min);
			t = 146.4mins to
			t = 146.7mins (0.3 min)

20%	Residual signal exceed from
	t =185.2mins to
	t = 185.9mins (0.7 min)
		Residual signal exceed from
		t = 110.1mins to
		t = 111.9mins (1.8mins);
		t = 114.6mins to
		t = 122.6mins (8mins);
		t = 123mins to
		t = 123.3mins (0.3 min);
		t = 124.5mins to
		t = 127.2mins (2.7mins);
		t = 183.1mins to
		t = 186.2min (3.1mins)
			Residual signal exceed from
			t = 121.3mins to
			t = 122mins (0.7 min);
			t = 126.8mins to
			t = 127.2 (0.4 min);
			t =129.4mins to
			t = 130.2mins (0.8mins)

30%	Residual signal exceed from
	t = 118.5mins to
	t =119mins (0.5 min);
	t = 183.1mins to
	t = 186.4mins (3.3mins)
		Residual signal exceed from
		t = 109.1mins to
		t = 113.5 (4.4mins);
		t = 114.2mins to
		t = 122.4mins (8.2mins);
		t = 125.6mins to
		t = 127.3mins (1.7mins);
		t = 185mins to
		t = 186mins (1min)
			None

40%	Residual signal exceed from
	t = 116.3mins to
	t = 119.9mins (3.6mins)
		Residual signal exceed from
		t = 110.2mins to
		t = 111.5mins (1.3mins);
		t = 116.4mins to
		t = 122mins (5.6min);
		t = 171.5mins to
		t = 172mins (0.5 min);
		t = 182.8mins to
		t = 187.9mins (5.1mins)
			None

50%	Residual signal exceed from
	t = 110.1mins to
	t = 111.8mins (1.7mins);
	t = 114.6mins to
	t = 116.1mins (1.5mins);
	t = 118.6mins to
	t = 121.4mins (2.8mins);
	t = 126.1mins to
	t = 127.7mins (1.6mins);
	t = 183.3mins to
	t = 186.1mins (2.8mins)
		Residual signal exceed from
		t = 118.5mins to
		t = 119.2mins (0.7mins);
		t = 182.9mins to
		t = 187.9mins (5mins)
			Residual signal exceed from
			t = 117.5mins to
			t = 118.7mins (1.2mins)

60%	Residual signal exceed from
	t = 110.1mins to
	t = 129.5mins (19.4mins);
	t = 183.2mins to
	t = 186.1mins (2.9mins)
		Residual signal exceed from
		t = 114.7mins to
		t = 121.7mins (7mins);
		t = 126.4mins to
		t = 127mins (0.6mins) ;
		t = 184.4mins to
		t = 185.9mins (1.5mins)
			None

70%	Residual signal exceed from
	t = 117.6mins to
	t = 119.2mins (1.6mins);
	t = 125.2mins to
	t = 130.7mins (5.5mins) ;
	t = 185.7mins to
	t = 185.9mins (0.2mins);
		Residual signal exceed from
		t = 110.1mins to
		t = 113.5mins (3.4mins);
		t = 114.2mins to
		t = 129.8mins (15.7mins);
		t = 171.4mins to
		t = 172.1mins (0.7mins);
		t = 185mins to
		t = 185.8mins (0.8mins)
			Residual signal exceed from
			t = 119.6mins to
			t = 130.9mins (11.3mins) ;
			t = 185.5mins to
			t = 186.3mins (0.8mins) ;
			t = 195.1mins to
			t = 195.8mins (0.7mins);

80%	Residual signal exceed from
	t = 109.8mins to t = 129.9mins (20.1mins);
	t = 183.0mins to t = 187.7mins (4.7mins) and
		Residual signal exceed from
		t = 109.2mins to t = 129.8mins (20.6mins);
		t = 184.1mins to t = 186.3mins (2.2mins)
			Residual signal exceed from
			t = 117.9mins to t = 124.5mins (6.6mins);
			t = 144.3mins to t = 144.5mins (0.2mins) ;
			t = 145.4mins to t = 146.2mins (0.8mins)

90%	Residual signal exceed from t = 109.5mins to t = 130.7mins (21.2mins)	Residual signal exceed from
		t = 110.1mins to t = 130.1mins (20mins);
		t = 184.0mins to t = 186.1mins (2.1min);
			Residual signal exceed from
			t = 117.0mins to t = 118.2mins (1.2mins) ;
			t = 133.0mins to t = 138.7mins (5.7mins);
			t = 184.3mins to t = 185.0mins (0.7mins);
			t = 185.4mins to t = 186.5mins (1.1mins);

100%	Residual signal exceed from
	t = 110.1mins to t = 122.5mins (21.6mins);
	t = 126.4mins to t = 127.8mins (1.4mins) ;
	t = 183.2mins to t = 187.8mins (4.6mins)
		Residual signal exceed from
		t = 115.0mins to t = 122.1mins (7.1mins);
		t = 125.8mins to t = 130.4mins (4.6mins);
		t = 183.2mins to t = 187.9mins (4.7mins)
			Residual signal exceed from
			t = 125.1mins to t = 130.4mins (21.6mins);
			t = 185.2mins to t = 185.8mins (5.4mins)

As may be observed and shown in Table 4.5, the neural network fault detection system is capable of working normally (that is no false positive alarms are detected) under less than 80% increase in the noise level when applied to all the three axes.

4.2. Case Studies: Detection of Multiple Faults under Continuous Setpoint Changes

In this subsection, detection of two fault patterns under three operating situations is studied. The robustness of the neural network under continuous setpoint changes is also investigated. The satellite is required to operate at each setpoint for 600 minutes and it is expected to change its position five times. During each setpoint, the satellite experiences voltage and current faults as well as the recovery of these faults. The impacts caused by the faults as well as the recovery characteristics are investigated. In order to compare and analyze the detection results, two aggregate fault patterns named F1 and F2 are designed as described in details below.

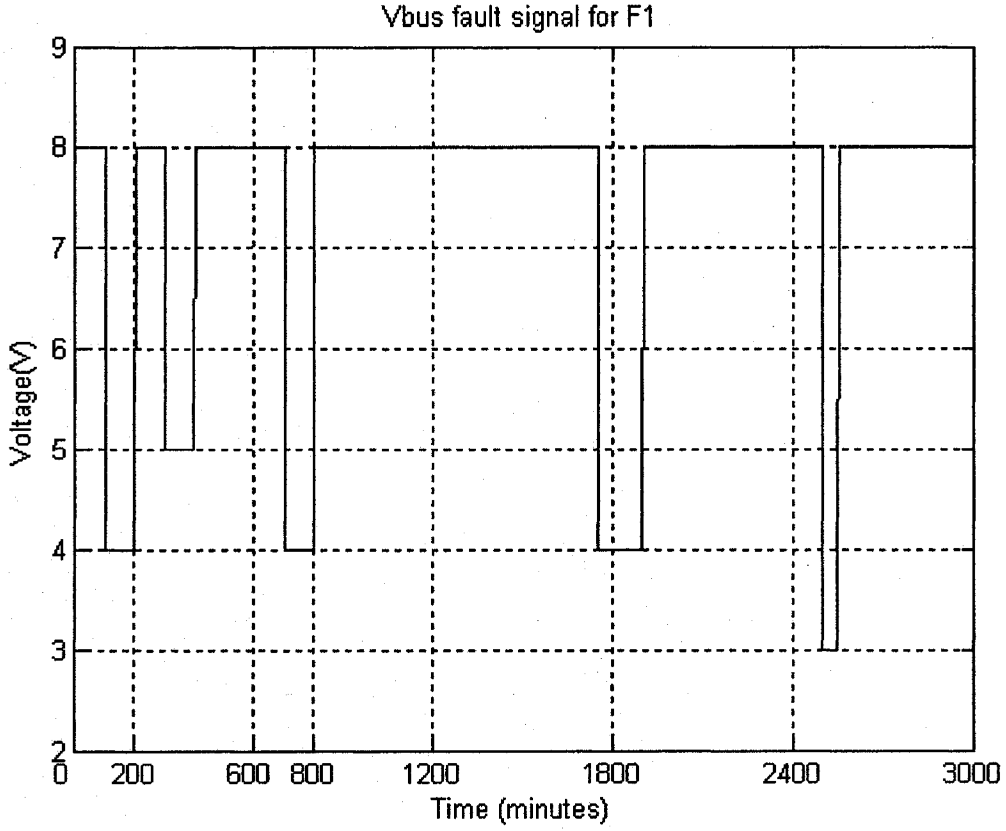

Figure 4.13 The V_{bus} fault signal patten for the fault pattern F1

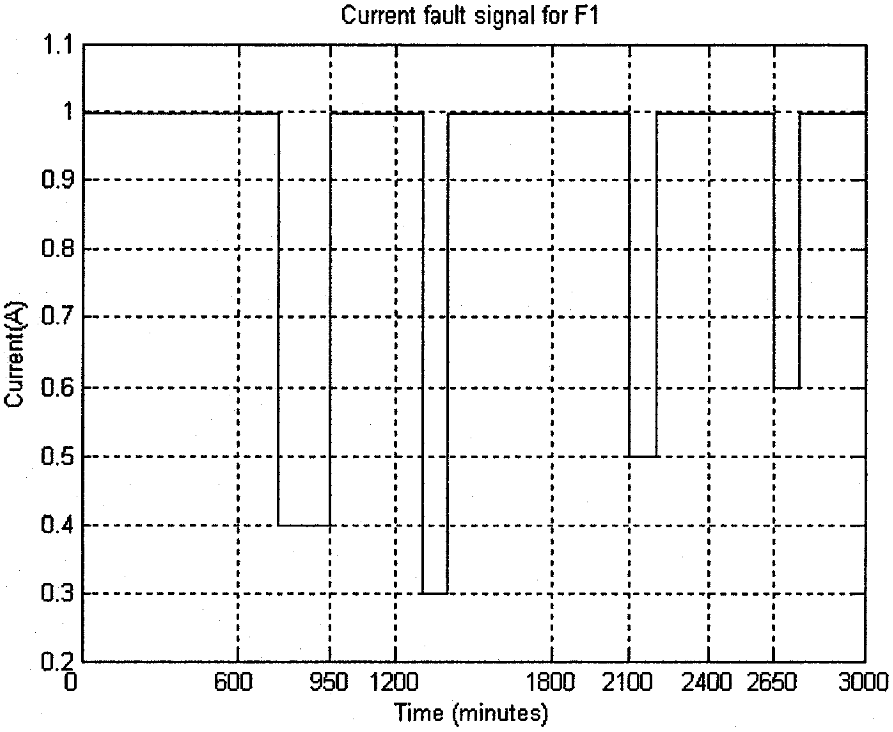

Figure 4.14 The Current fault signal patten for the fault pattern F1

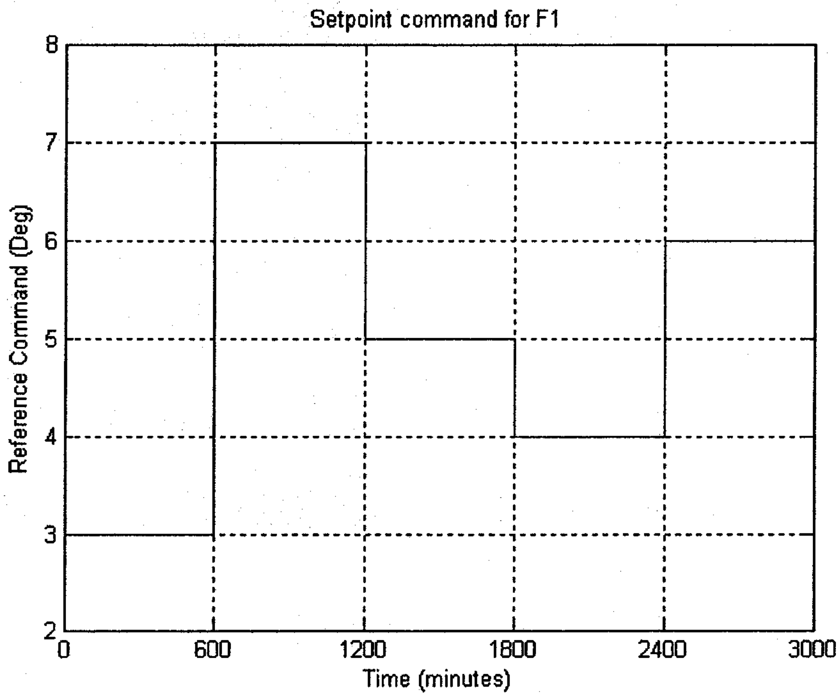

Figure 4.15 The setpoint change pattern for the fault pattern F1

The complete pattern for the fault F1 is detailed below:

- From 0 to 599 minutes: the satellite changes its position from 0 deg to 3 deg. At t = 100 minutes, the V_{bus} voltage fault from 8V to 4V is applied which is removed at t = 200 minutes. Later, at t = 300 minutes, the V_{bus} voltage fault is applied again, this time from 8V to 5V at t = 300 minutes, and which is removed (recovered) at t = 400 minutes.
- From 600 to 1199 minutes: at t = 600 minutes the satellite changes its position to 7 deg. At t = 700 minutes, the V_{bus} voltage fault from 8V to 4V is applied and before it is removed or recovered, a current fault signal at t = 750 minutes from 1A to 0.4A is applied. Later, at t = 800 minutes, the V_{bus} fault signal is recovered and at t = 950 minutes the current is also recovered.
- From 1200 to 1799 minutes: at t = 1200 minutes, the satellite changes its position to 5 deg. At t = 1300 minutes, the current fault from 1A to 0.3A is applied and which is recovered at t = 1400 minutes. Later at t = 1750 minutes, the V_{bus} fault from 8V to 4V is applied and it is not recovered within this setpoint.
- From 1800 to 2399 minutes: at t = 1800 minutes, the satellite changes its position to 4 deg. At t = 1900 minutes, the V_{bus} fault carried from the previous setpoint is recovered. Then at t = 2100 minutes, the current fault from 1A to 0.5A is applied, and this fault is removed at t = 2250 minutes.
- From 2400 to 2999 minutes: at t = 2400 minutes, the satellite changes its position to 6 deg. The V_{bus} fault at t = 2500 minutes from 8V to 3V is applied

and it is recovered soon at t = 2550 minutes. Later, a small current fault from 1A to 0.6A is applied at t = 2650 minutes and is recovered at t = 2800 minutes.

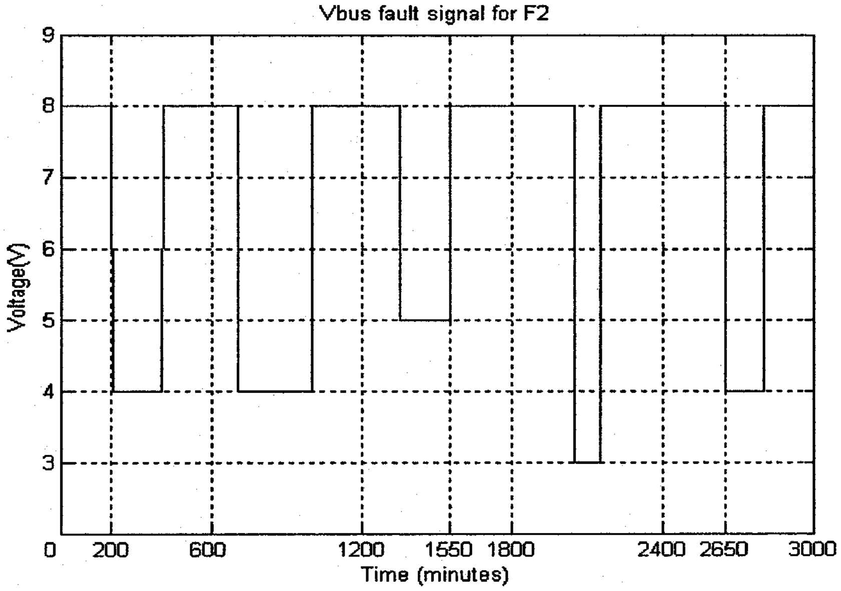

Figure 4.16 The V_{bus} fault signal pattern for the fault pattern F2

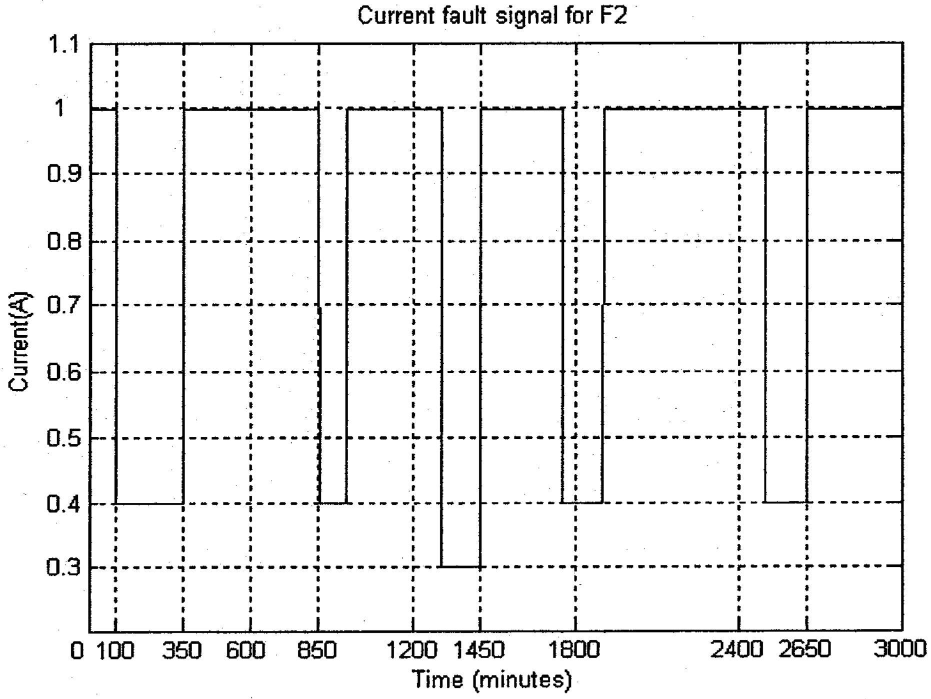

Figure 4.17 The current fault pattern for the fault pattern F2

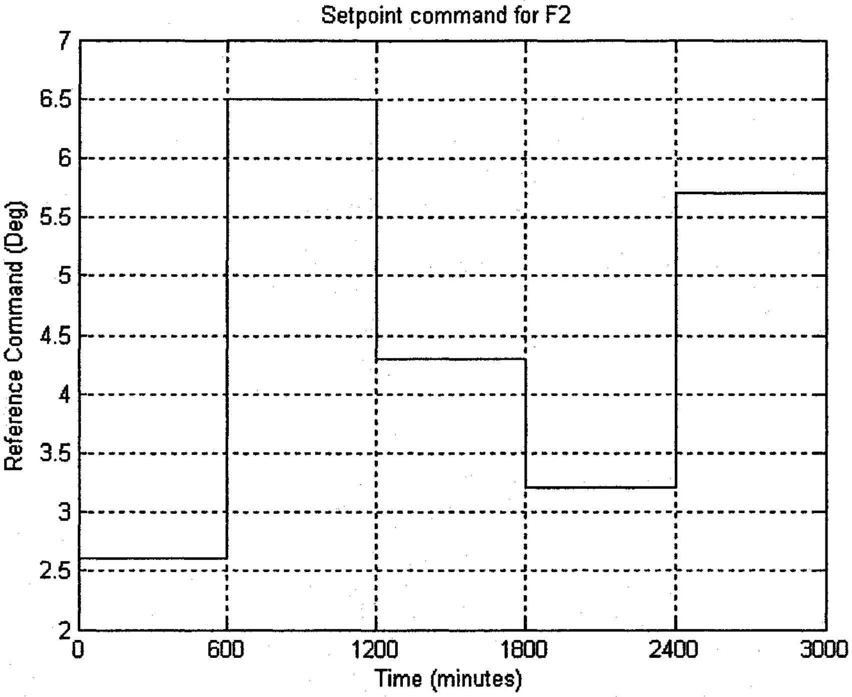

Figure 4.18 The setpoint change pattern for the fault pattern F2

The details for pattern F2 are explained below:

- From 0 to 599 minutes: the satellite changes its position from 0 deg to 2.6 deg. At t = 100 minutes, the current signal fault from 1A to 0.4A is applied and at t = 200 minutes a V_{bus} fault is also applied simultaneously. The satellite operated under these two simultaneous faults until t = 350 minutes when the current fault signal is removed. Later at t = 400 minutes, the voltage fault is recovered.
- From 600 to 1199 minutes: at t = 600 minutes, the satellite changes its position to 6.5 deg. At t = 700 minutes, the V_{bus} fault from 8V to 4V is applied and the current fault at t = 850 minutes from 1A to 0.4A is then applied simultaneously. Then at t = 950 minutes, the current signal fault is recovered first and later at t = 1000 minutes the V_{bus} fault is also recovered.

- From 1200 to 1799 minutes: at t = 1200 minutes, the satellite changes its position to 4.3 deg. At t = 1300 minutes, the current fault from 1A to 0.3A is applied and soon without being recovered, at t = 1350 minutes a small V_{bus} fault from 8V to 5V is applied simultaneously. Later at t = 1450 minutes, the current fault is recovered first, and then the V_{bus} fault is recovered at t = 1550 minutes. Moreover, at t = 1750 minutes, the current fault from 1A to 0.4A is applied which is carried out to the next setpoint change.
- From 1800 to 2399 minutes: at t = 1800 minutes, the satellite changes its position to 3.2 deg. At t = 1900 minutes, the current fault carried over from the previous setpoint is recovered. Then at t = 2050 minutes, the V_{bus} fault from 8V to 3V is applied and this large fault is removed at t = 2150 minutes.
- From 2400 to 2999 minutes: at t = 2400 minutes, the satellite changes its position to 5.7 deg. The current fault at t = 2500 minutes from 1A to 0.4A is applied and is recovered at t = 2650 minutes. Meanwhile, at t = 2650 minutes, the V_{bus} fault from 8V to 4V is applied and is recovered at t = 2800 minutes.

4.2.1. Case 1: One Axis is Faulty and the Other Two Axes are Fault Free

4.2.1.1. The Faulty Axis (Yaw Axis) Has F1 Pattern Fault

The detection results corresponding to the F1 are shown blow:

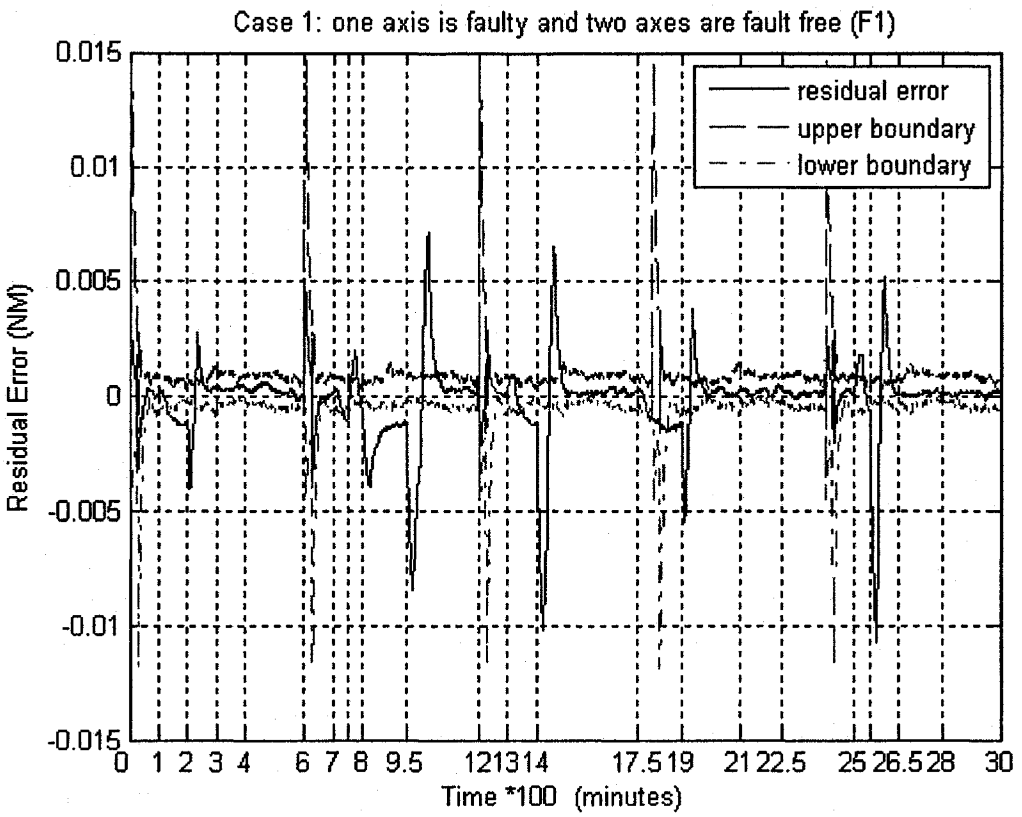

Figure 4.19 Case 1 detection results for the fault pattern F1

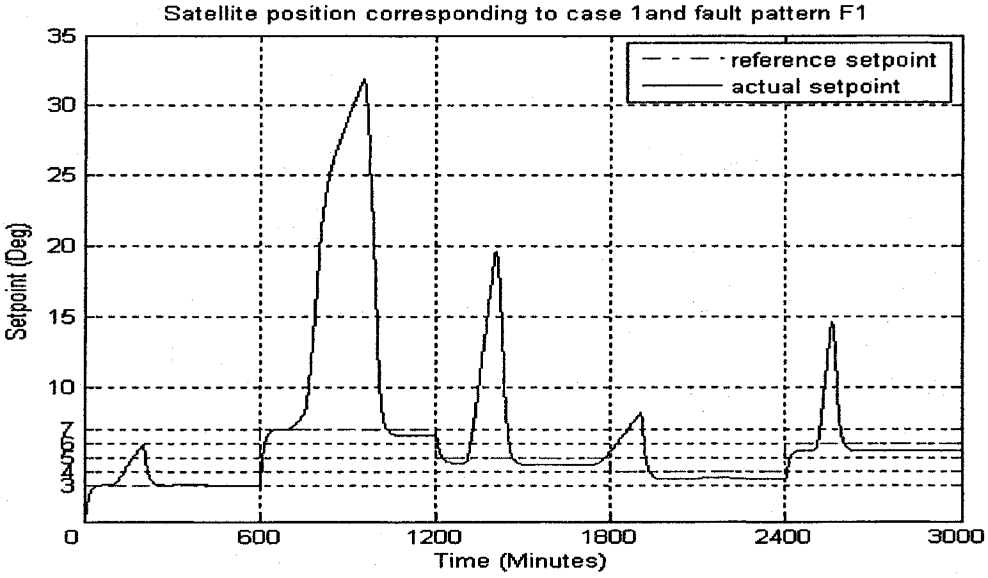

Figure 4.20 The positions of the satellite for case 1 corresponding to fault pattern F1

Each event, either fault detection or recovery, is labeled in Figure 4.19. In summary, once the fault has occurred the residual error exceeds the boundary and it would return back and stays inside the boundary once the fault is removed. The details for the individual setpoints are described below.

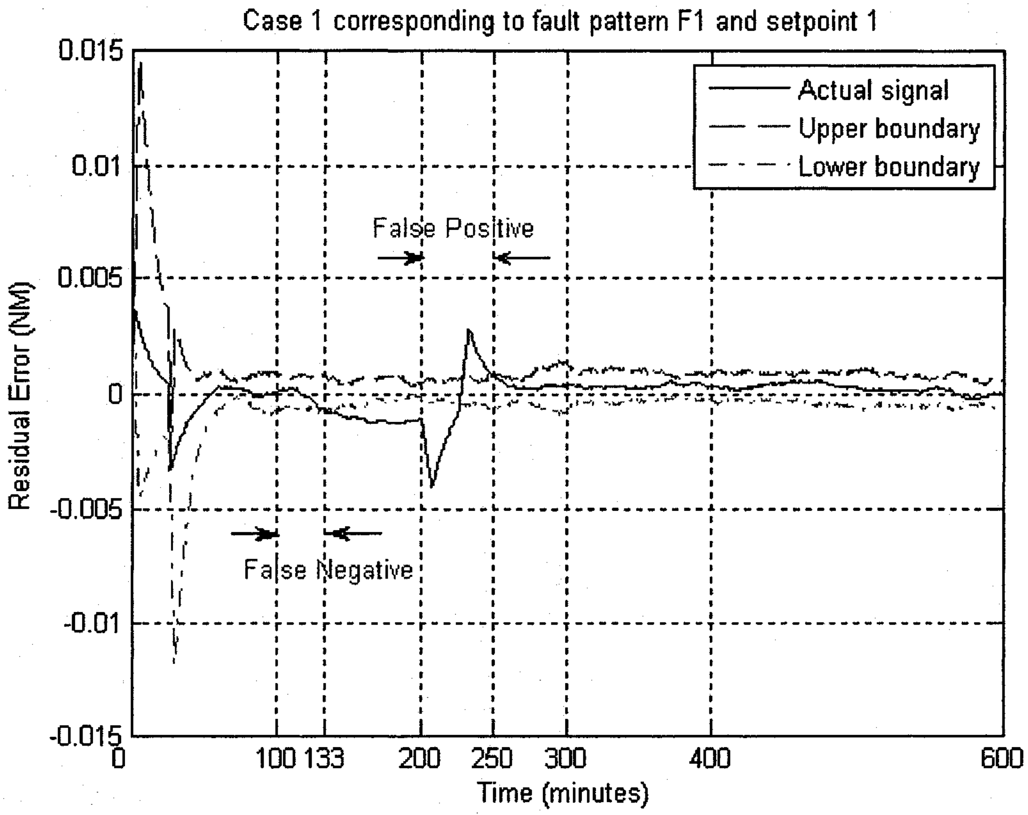

Figure 4.21 The detection results for setpoint 1 in case 1 for the fault pattern F1

With 33 minutes delay, the first fault caused the residual error to exceed the lower boundary at t = 133 minutes. This delay is known as *False Negative* which implies that there exist a fault but the neural network was not able to detect it. At t = 200 minutes, this fault was removed, before the residual error has a chance to enter the boundary it experienced another kind of delay known as, *False Positive,* which means that the network detects it as a fault although no fault exists. The details for this transient phase are as follows: at t = 207 minutes the residual curve reached its recovery undershoot then it ascended and at t = 226 minutes it entered the lower boundary; then exceeded the upper boundary 2 minutes later (at t = 228 minutes). At t = 232.8 minutes, it reached its overshoot then it descended and around 20 minutes later, at t = 250 minutes, it reentered the threshold and settles down inside the boundary. Since the second fault occurred at t = 300 minutes and was recovered at t = 400 minutes was a small fault, the neural network could not detect it implying that associated with the FDI scheme once a fault is removed it does not affect the detection characteristics of other faults.

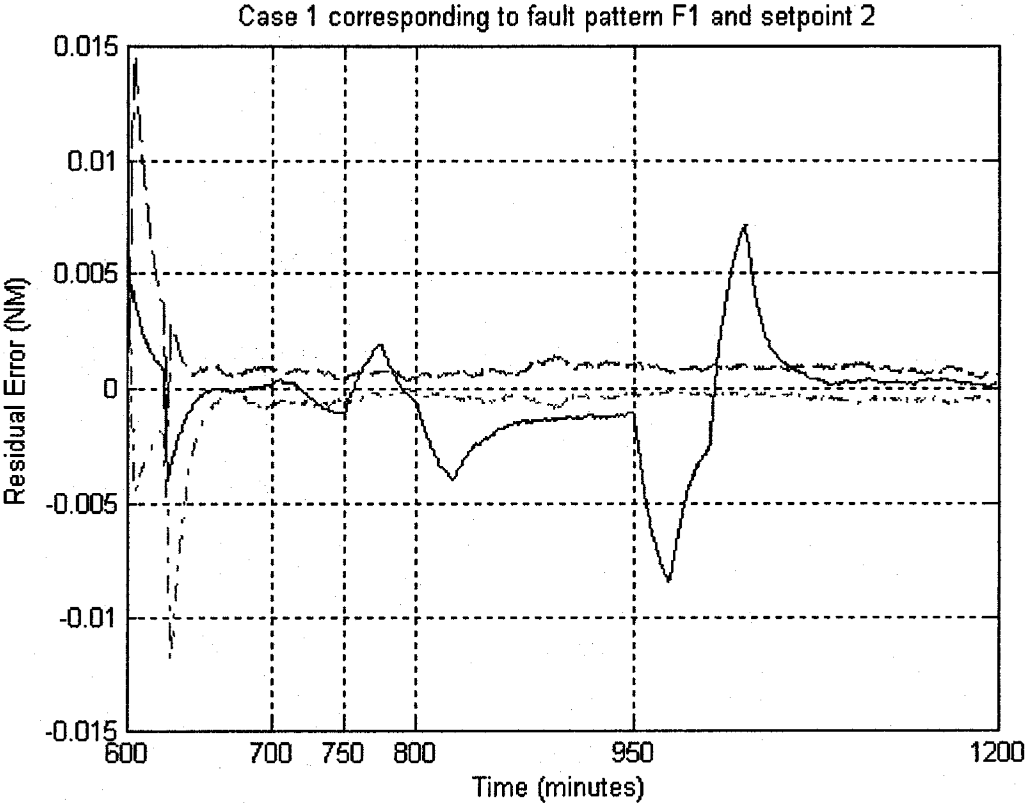

Figure 4.22 The detection for setpoint 2 in case 1 for the fault pattern F1

For the second setpoint, the V_{bus} fault was applied at t = 700 minutes and the residual error exceeded the lower boundary at t = 733 minutes. Unfortunately, since the current fault signal occurred at t = 750 minutes there is no sufficient time (20 minutes) to judge the voltage fault as faulty even though the residual has already exceeded the boundary. The curve ascended to respond to the current fault. At t = 753 minutes, the residual curve entered the lower boundary and at t = 761 minutes it exceeded the upper boundary then reached its overshoot at t = 775 minutes. Then, it began to descend. It crossed the upper and lower boundaries at t = 783 minutes and t = 792.4 minutes, respectively. Without much time to settle down, 7 minutes later the V_{bus} fault was recovered at t = 800 minutes. The residual curve responded to this change instantaneously by accelerating its descent speed. At t = 825 minutes, it reached its recovery undershoot and then ascended. Since the current in the wheel was still faulty, the residual curve could not enter the threshold. Finally, at t = 950 minutes, the current was recovered and the residual curve descended first and reached its recovery undershoot at t = 975 minutes and then ascended. From t = 987.4 minutes, the ascent rate slowed down and at t = 1002 minutes the residual curve changed its shape (reflection point). From t = 1005 minutes to t = 1007 minutes, the residual error crossed the threshold and at t = 1027 minutes, it reached its recovery overshoot and then descended. At t = 1060 minutes, the residual curve reentered the boundary and settled down within the two boundaries.

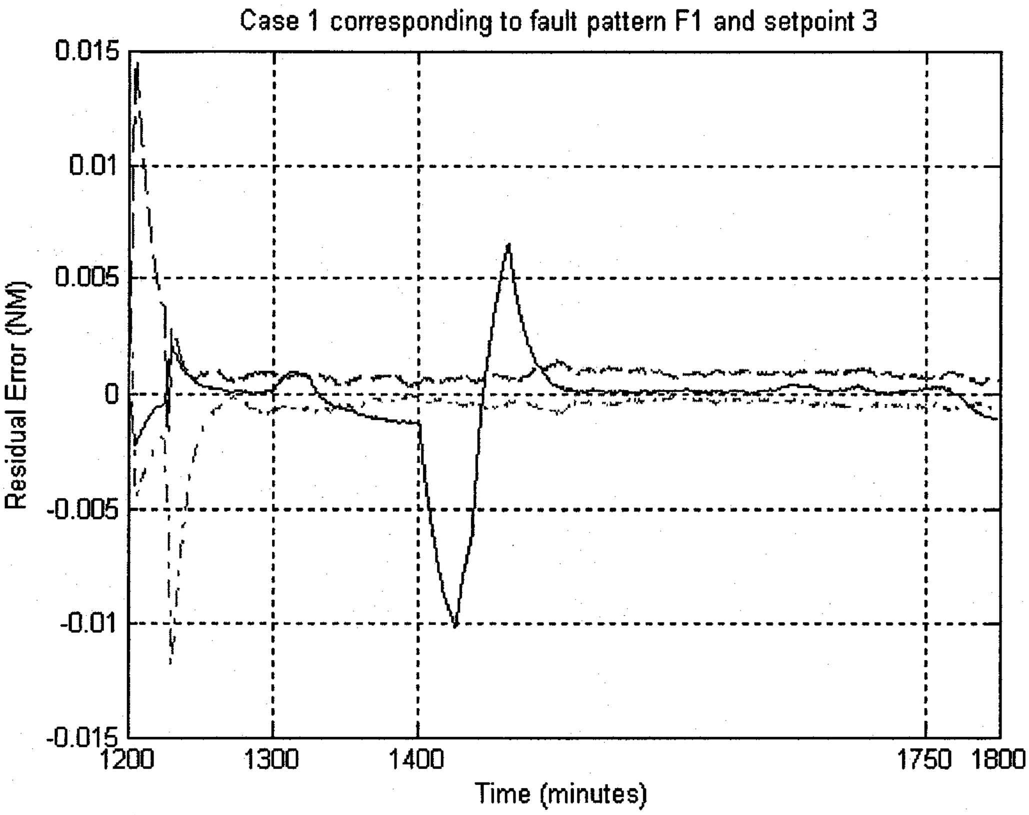

Figure 4.23 The detection results for setpoint 3 in case 1 for the fault pattern F1

For the third setpoint, the current fault signal was applied at t = 1300 minutes and the residual curve exceeded the upper boundary 6 minutes later but reentered it soon at t = 1325 minutes. It was expected to have a detection overshoot during this phase but a weak one. Then the residual error descended and crossed the lower boundary at t = 1344 minutes. At t = 1400 minutes, the fault was removed. The residual curve reached its recovery undershoot at t = 1426 minutes and overshoot at t = 1462 minutes. It then crossed the lower and upper boundaries at t = 1443.4 minutes, t = 1445.5 minutes, respectively. Finally, the residual error entered the boundary at t = 1484 minutes. At t = 1750 minutes, the voltage fault was applied and the residual curve crossed the threshold curve from t = 1780 minutes.

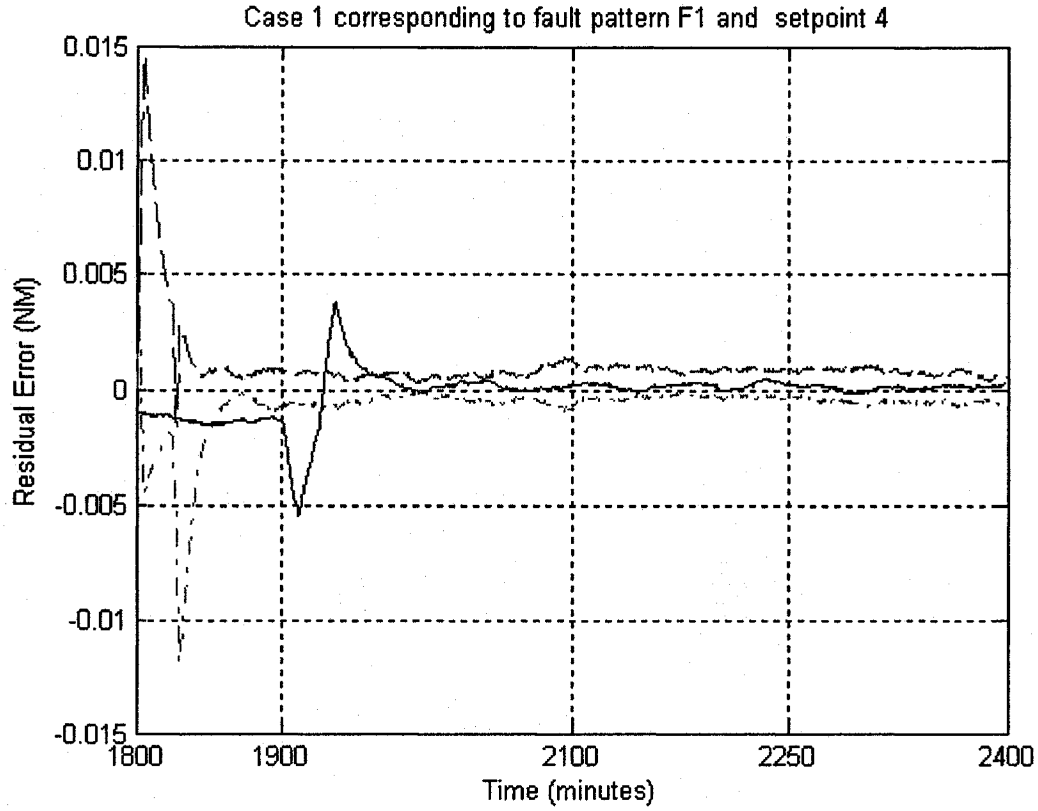

Figure 4.24 The detection results for setpoint 4 in case 1 for the fault pattern F1

At t = 1800 minutes, the satellite received a command to change its position. Since the voltage was abnormal, a spike due to the setpoint change was missed in this case. At t = 1852.3 minutes, the residual curve exceeded the lower boundary horizontally and tended to settle down there. At t = 1900 minutes, the voltage fault was removed and the residual curve responded to it by descending to reach its recovery undershoot at t = 1912 minutes. Then, it entered the lower boundary at t = 1928 minutes. It took 2 minutes to exceed the upper boundary and reached its recovery overshoot at t = 1937 minutes. Finally, at t = 1965 minutes the residual curve reentered the boundary and settled inside there. From t = 2100 to t = 2250 minutes, a small current fault (1A to 0.5A) has occurred and consistent with the precious detection results, the neural network did not detect it as a fault.

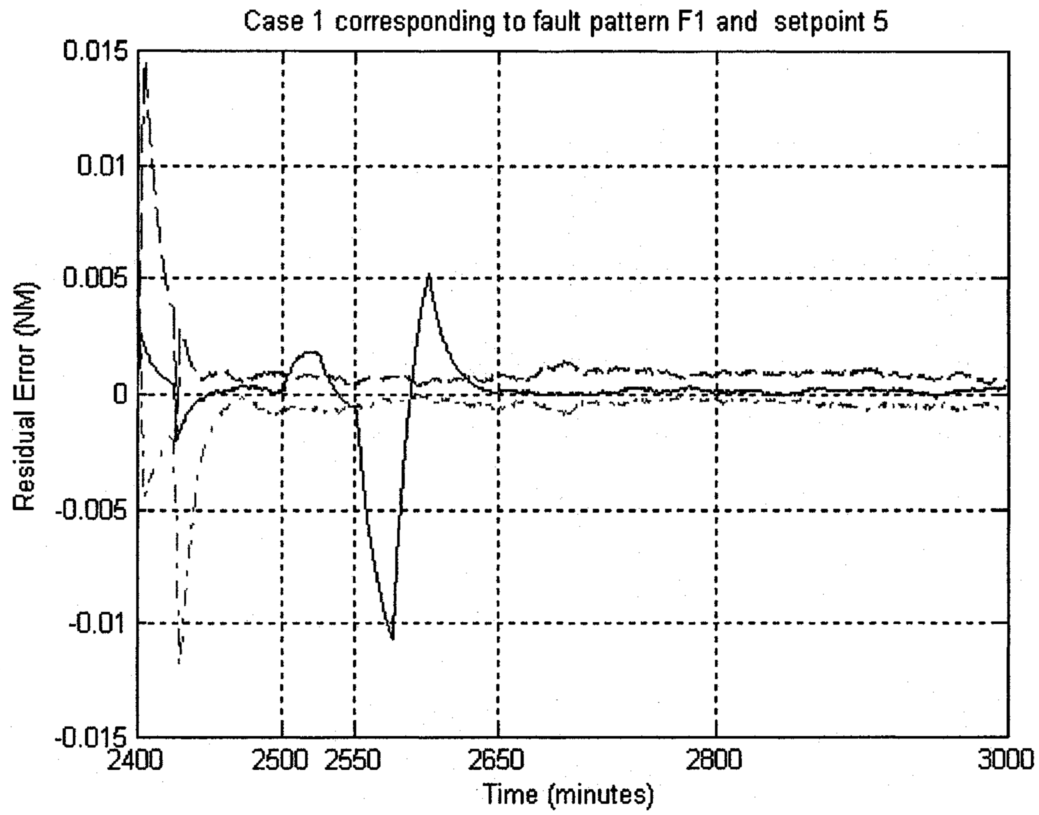

Figure 4.25 The detection result for Setpoint 5 in case 1 for the fault pattern F1

For the last setpoint change, a large V_{bus} fault (8V to 3V) has occurred at t = 2500. Four minutes later at t = 2504 minutes, the residual curve exceeded the upper boundary and reentered it at t = 2530 minutes. Since the residual curve crossed outside the boundary for more than 20 minutes, it was detected as a fault here. It kept on descending and crossed the lower boundary at t = 2547 minutes. At t = 2550 minutes, the voltage fault was removed. At t = 2575.9 minutes, the residual curve reached its recovery undershoot and it crossed the lower boundary and upper boundary at t = 2588.2 minutes and t = 2589.4 minutes, respectively. It reached its recovery overshoot at t = 2601.4 minutes and then descended and finally reentered the threshold boundaries at t = 2627.5

minutes. A small current fault (1A to 0.6A) has occurred at t = 2650 minutes and was recovered at t = 2800, where the neural network could not detect it as a fault.

4.2.1.2. The Faulty Axis Has F2 Pattern Fault

The detection results corresponding to F2 are shown blow:

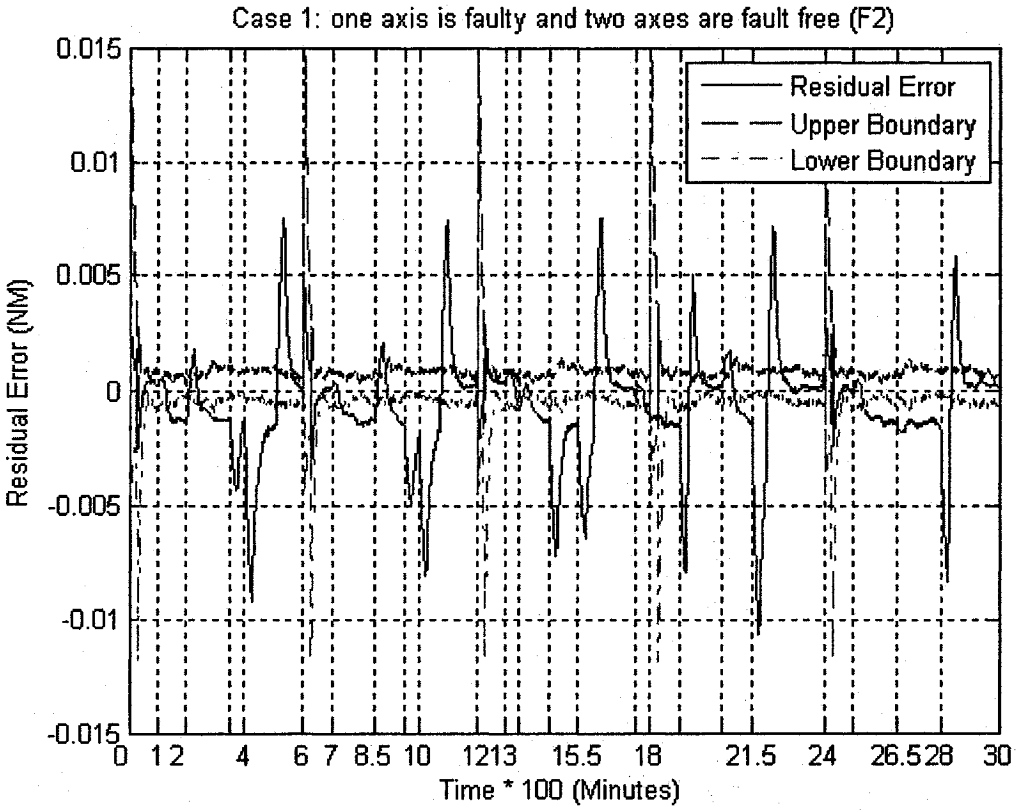

Figure 4.26 The case 1 fault detection results for the fault pattern F2

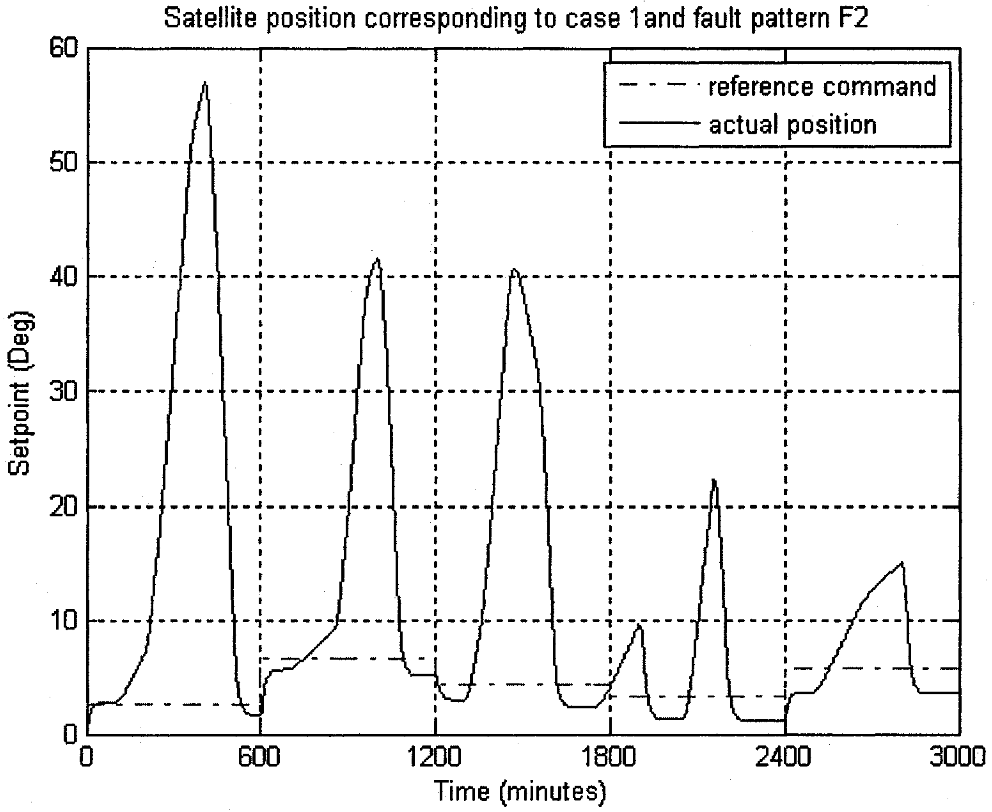

Figure 4.27 The satellite positions for case 1 corresponding to the fault patter F2

Figure 4.27 indicates that the faults in the F2 pattern cause the satellite to deviate from its reference command significantly. The detection results for the individual setpoints are described details below.

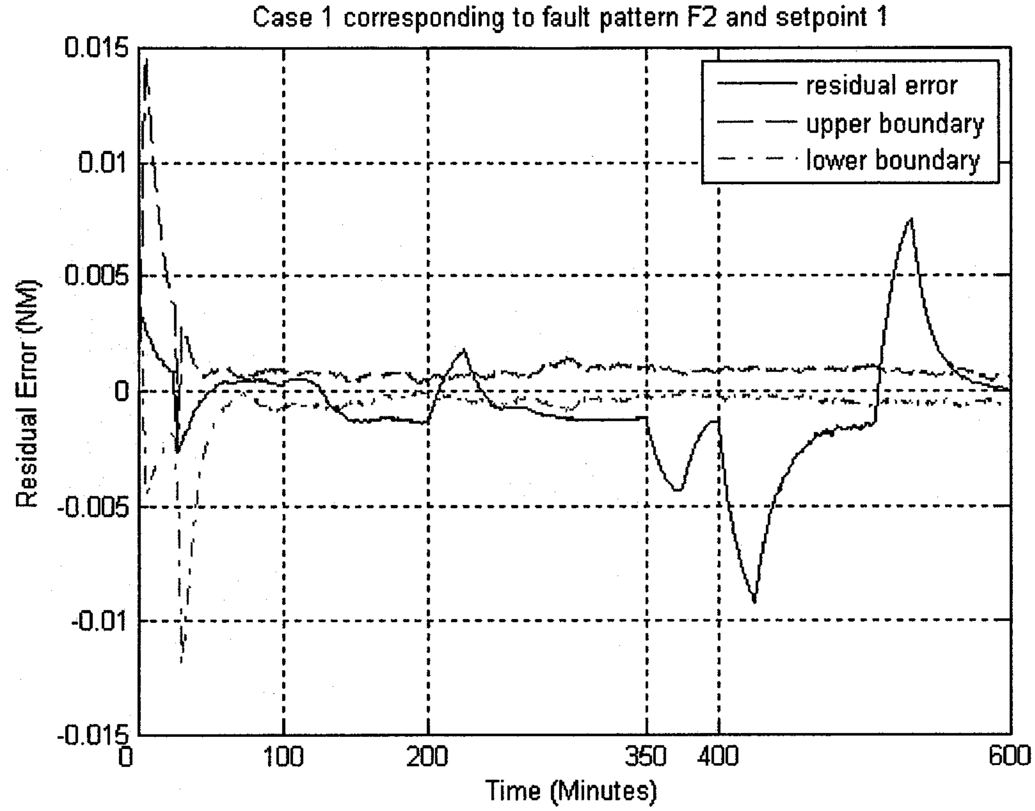

Figure 4.28 The detection results for setpoint 1 in case 1 For the fault pattern F2

For the first setpoint, the faults are imposed together. The recovery order of the faults is consistent with their occurrence order. At t = 100 minutes, the current fault was applied and 35 minutes later the residual curve exceeded the lower boundary. After the specified a priori 20 minutes detection time, at t = 155 minutes the fault was detected. At t = 200 minutes, the V_{bus} fault was applied and the neural network responded to this change immediately, that is it crossed the lower and upper boundaries at t = 205 minutes and t = 212 minutes, respectively and reached its detection overshoot at t = 225 minutes. The residual curve then reentered the upper boundary at t = 231 minutes. The residual error kept on descending and exceeded the lower boundary at t = 240 minutes. At t = 350 minutes, the current fault was recovered. At t = 375 minutes the residual curve reached its first recovery undershoot (caused by the current) and then ascended. Without sufficient allowed time for the transients to complete, at t = 400 minutes the V_{bus} fault was removed. The residual curve reached its second recovery undershoot at t = 425 minutes and then ascended. At t = 460 minutes, the residual curve tended to extend horizontally and changes its shape at t = 507.5 minutes (reflection point). The residual error kept on ascending and crossed the lower and upper boundaries at t = 509 minutes and t = 511.2 minutes, respectively. Finally the error reached its recovery overshoot at t = 532 minutes. At t = 563 minutes, it reentered the boundary and settled down there.

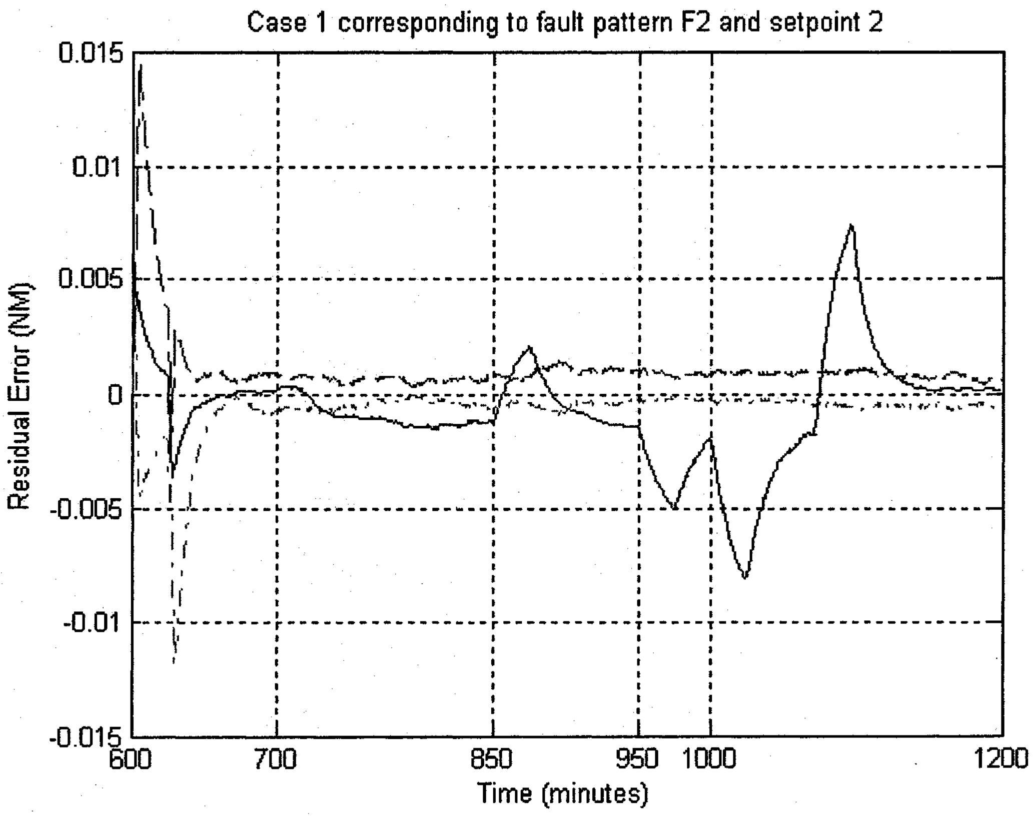

Figure 4.29 The detection for setpoint 2 in case 1 for the fault pattern F2

For the second setpoint, the faults were imposed together but the recovery order of the faults was contrary with their occurrence order. The occurrence and recovery of the current fault was processed before the application of the V_{bus} fault. However, it did cause extra difficulties for the detection scheme. With 35 minutes delay, the residual curve exceeded the lower boundary at t = 735 minutes due to the voltage fault. At t = 850 minutes, the current fault was applied. Three minutes later, the residual curve entered the lower boundary and crossed the upper boundary at t = 861 minutes. At t = 875 minutes, the residual curve reached its detection overshoot and then reentered the upper boundary at t = 881 minutes. It kept on descending and crossed the lower boundary at t = 902 minutes. At t = 950 minutes, the current was recovered. At t =975 minutes, the residual curve experienced the recovery undershoot due to the current fault. Without sufficient time allowed to complete the transient response, V_{bus} fault recovery occurred at t = 1000 minutes. At t = 1025 minutes, the residual curve reached its recovery undershoot again and around t = 1050 minutes the residual curve decreased the ascent rate and at t = 1071.5 minutes, it met its reflection point. At t = 1074 minutes and 1076 minutes, the residual curve crossed the lower and upper boundaries, respectively. At t = 1096 minutes, the residual error reached its recovery overshoot and began to descend. At t = 1127 minutes, the error reentered the boundary and settled down there.

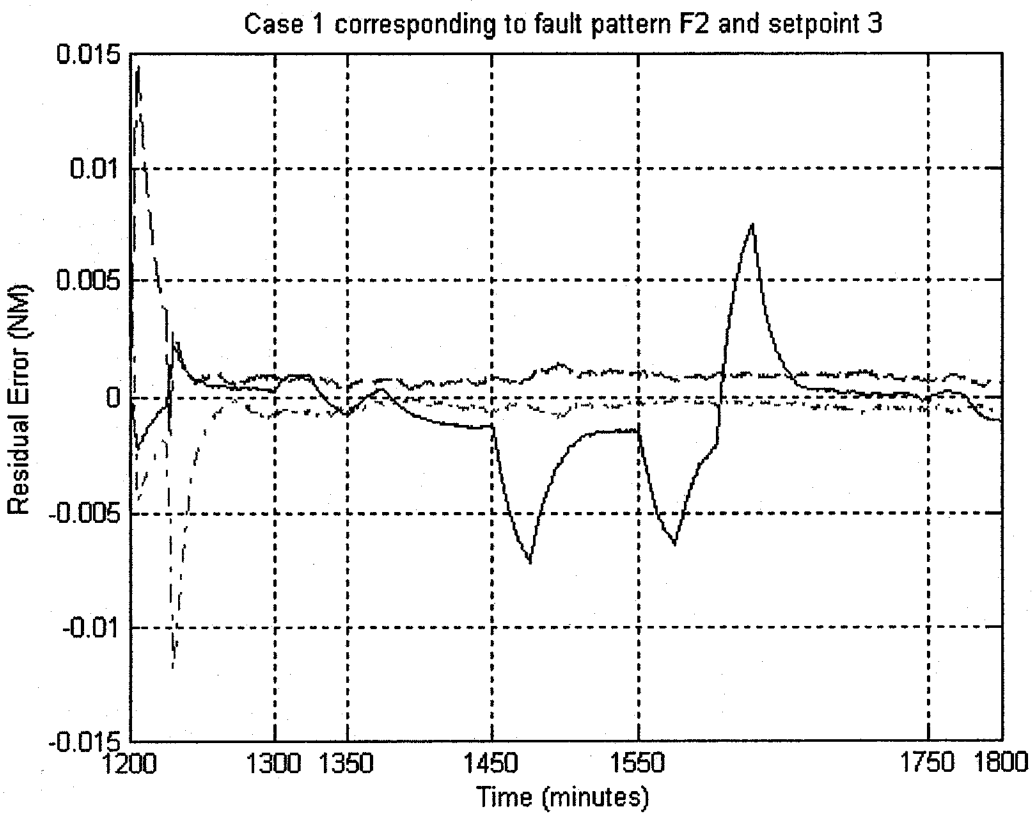

Figure 4.30 The detection for setpoint 3 in case 1 for fault pattern F2

For the third setpoint, an undetectable fault (8V to 5 V) was applied in addition to a detectable fault (1 A to 0.3 A) and the results showed that both faults were now detectable in this situation. The first fault occurred at t = 1300 minutes and 5 minutes later the residual error exceeded the upper boundary and then reentered it at t = 1326 minutes. The residual curve kept on descending and finally crossed the lower boundary at t = 1345 minutes. Five minutes later, a small V_{bus} fault has occurred (at t = 1350 minutes). The residual curve reentered the lower boundary at t =1353.6 minutes and without reaching the upper boundary it exceeded it at t = 1385 minutes. At t = 1450 minutes, the current fault was recovered. At 1476 minutes, the residual curve reached its recovery undershoot and around t = 1492.5 the residual curve reached its first recovery steady state. At t = 1550 minutes, the V_{bus} fault was removed and its transient took 106 minutes to settle down. At t = 1575.6 minutes the residual curve reached its recovery undershoot and at t = 1604 minutes it changes its shape (reflection point). The residual curve entered the lower and upper boundaries at t =1606.5 minutes and t = 1608.3 minute, respectively and then reached the recovery overshoot at t = 1629 minutes. At t = 1656 minutes, the residual curve reentered the boundary and settled down there. At t = 1750 minutes, the current fault was applied again, and the residual curve exceeded the boundary at t = 1781 minutes.

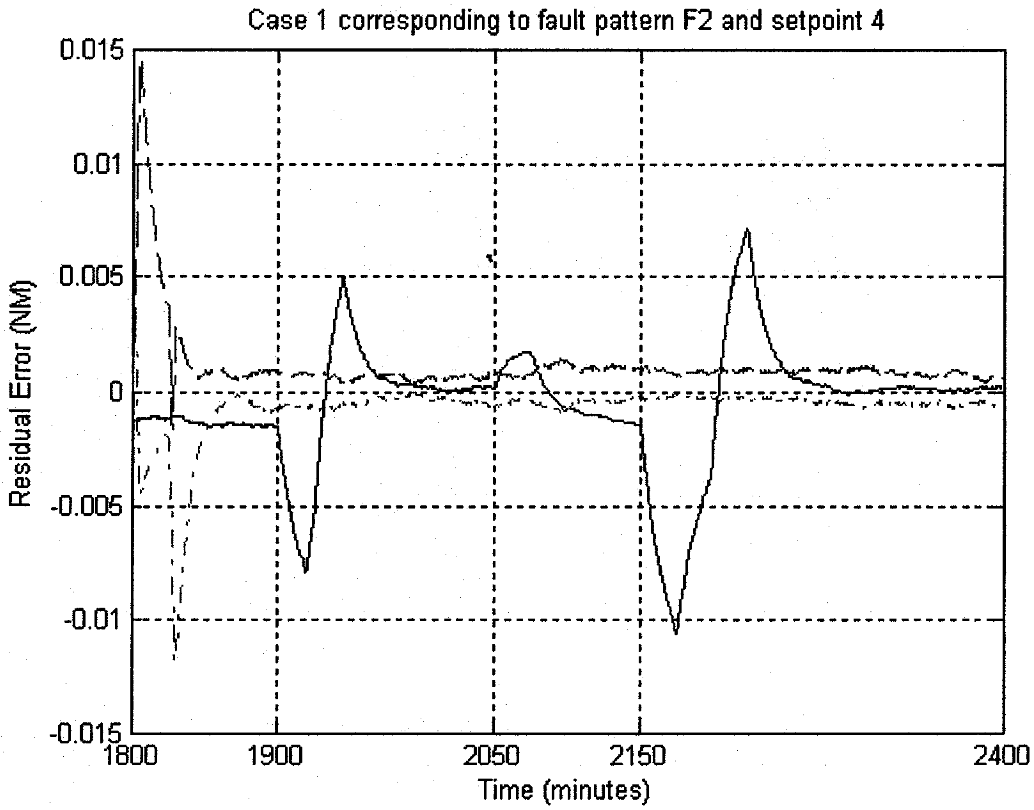

Figure 4.31 The detection results for setpoint 4 in case1 for the fault pattern F2

For the fourth setpoint, with the current fault applied at t = 1800 minutes the residual curve missed a spike due to the setpoint change. At t = 1852.3 minutes, the residual curve exceeded the lower boundary horizontally. At t = 1900 minutes, the current fault signal was removed. The residual curve reached the recovery undershoot at t = 1920 minutes and overshoot at t = 1945 minutes and finally entered the boundary at t = 1970 minutes. At t = 2050, the V_{bus} fault was applied and 4 minutes later the residual curve exceeded the upper boundary and reentered it at t = 2079 minutes. Finally, the error exceeded the lower boundary at t = 2101 minutes. At t = 2150 minutes, the V_{bus} fault was recovered. At t = 2176 minutes, the residual curve reached its recovery undershoot and began to ascend. Around t = 2198 minutes the residual curve met its reflection point. At t = 2224 minutes the residual error reached its recovery overshoot and at t = 2249 minutes, the residual curve settled down in the boundary.

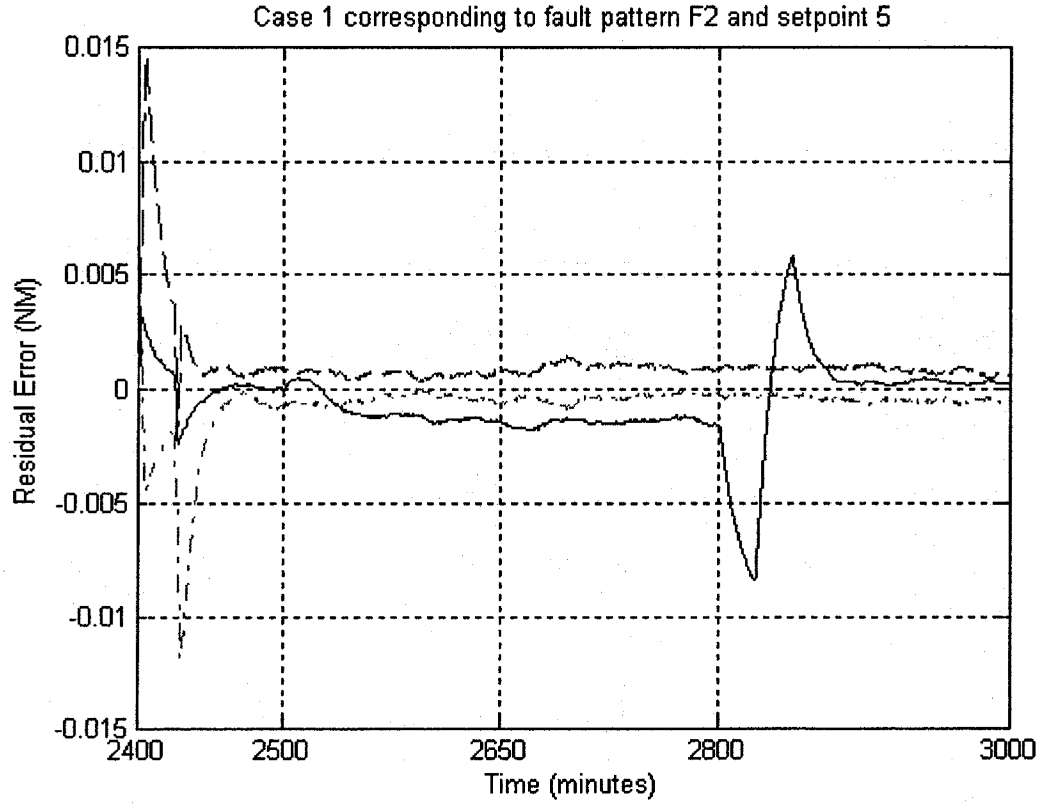

Figure 4.32 The detection results for setpoint 5 in case 1 for the fault patter F2

For the fifth setpoint, another interesting fault pattern is studied that is the current recovery and the V_{bus} fault have occurred simultaneously. As a result, the transient phase due to the current removal has disappeared. At t = 2500 minutes, the current was faulty, and 36 minutes later, at t = 2536 minutes the residual curve exceeded the lower boundary. At t = 2650 minutes, the current was removed and the V_{bus} signal became faulty. Without any change, the residual curve kept its tendency. At t = 2800 minutes, the V_{bus} fault was recovered. The residual curve reached its recovery undershoot at t = 2825.6 minutes and crossed the lower and upper boundaries at t = 2835.3 minutes and t = 2837 minutes, respectively and then it reached its recovery overshoot at t = 2851 minutes. Finally, the residual curve settled down into the boundary after t = 2874 minutes.

Based on the previous 10 setpoint change studies, we are able to make the following observation. Namely, the neural network is capable of providing good fault detection results in general. However, it seems that after the recovery of a fault the satellite tracking point accuracy has been impacted, which is due to the nonlinearity of the system. Since the entire system is nonlinear, the impact caused by the fault can not be removed completely as the fault is recovered. Therefore, there exists an increasing requirement that the PID controller parameters need to be fine tuned after the recovery of each fault, which is called self-tuning in the control area and which is clearly beyond the scope of this thesis.

4.2.2. Case 2: One Axis is Fault Free and the Other Two are Faulty

4.2.2.1. Two Faulty Axes with F1 Fault Pattern

The detection results corresponding to the fault pattern F1 in case 2 are shown blow:

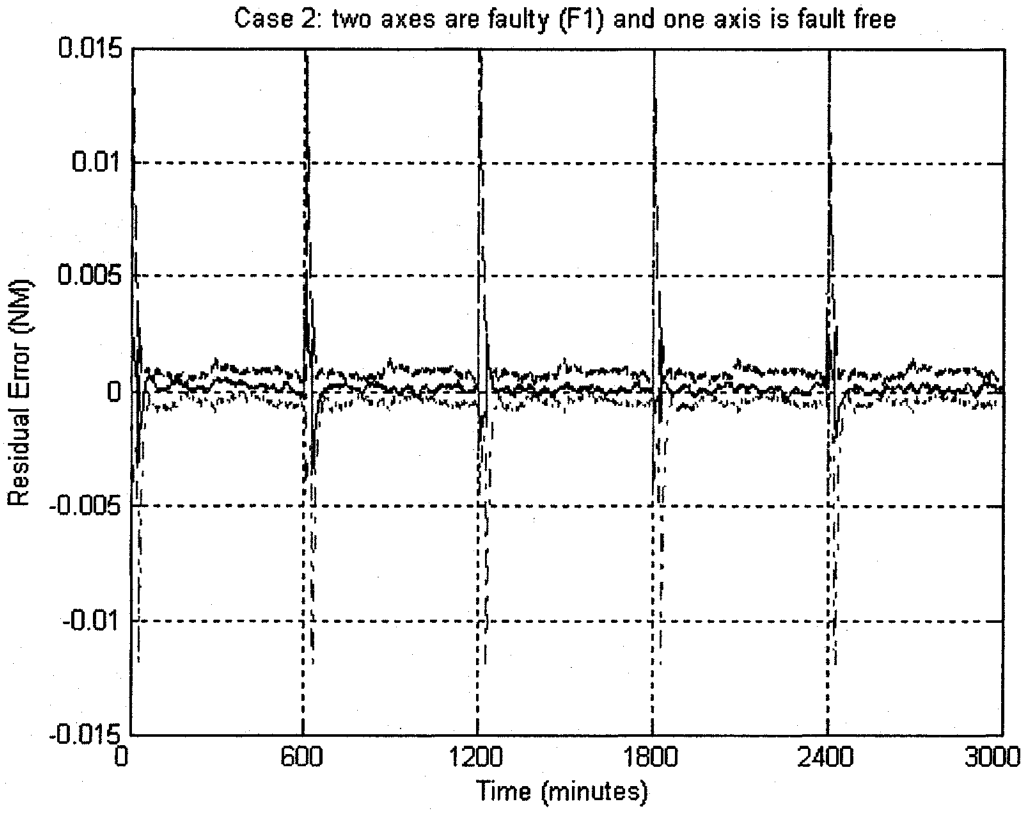

Figure 4.33 The detection results for the fault patter F1 in case 2

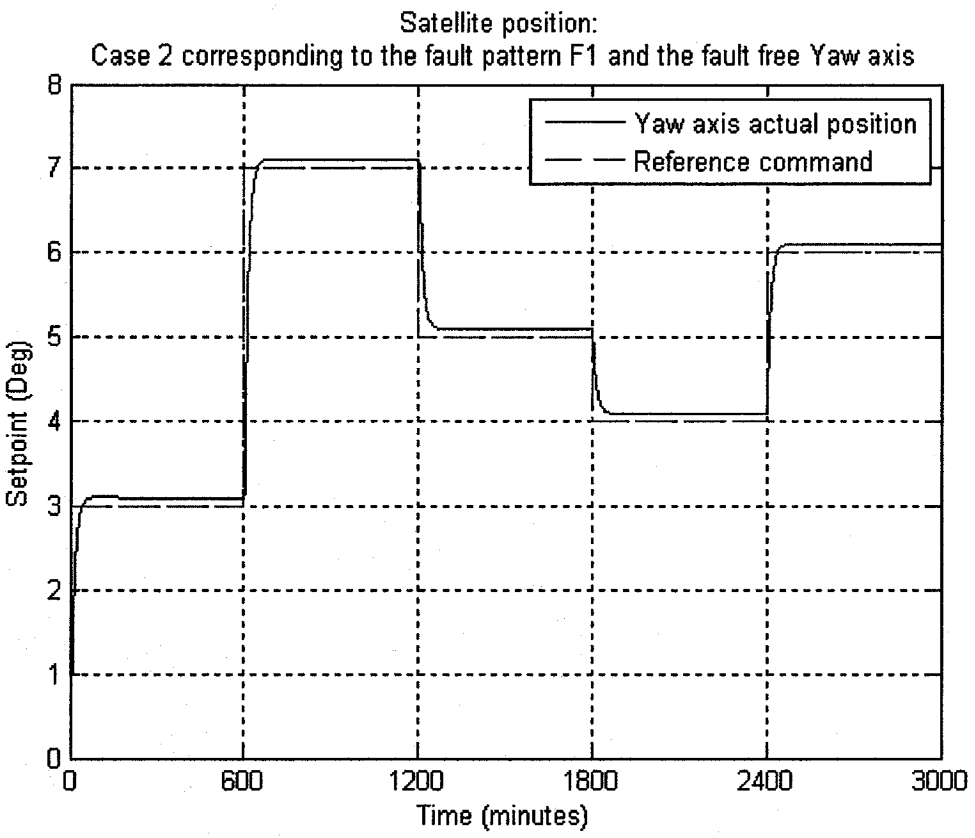

Figure 4.34 The satellite position for the fault free axis in case 2 and the fault pattern F1

From Figure 4.33, it may be concluded that the two faulty axes did not cause any adverse influence on the fault free axis (Yaw axis) despite the strong nonlinear interaction among all the three axes. These results are also supported by the results provided in Figure 4.34.

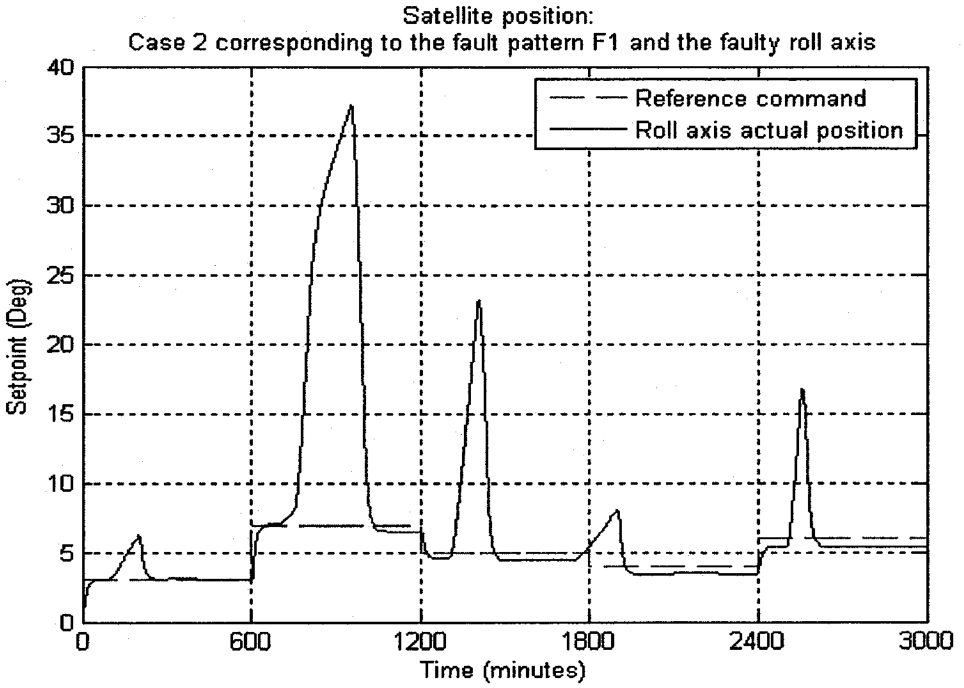

Figure 4.35 The satellite position for the faulty roll axis in case 2 and the fault pattern F1

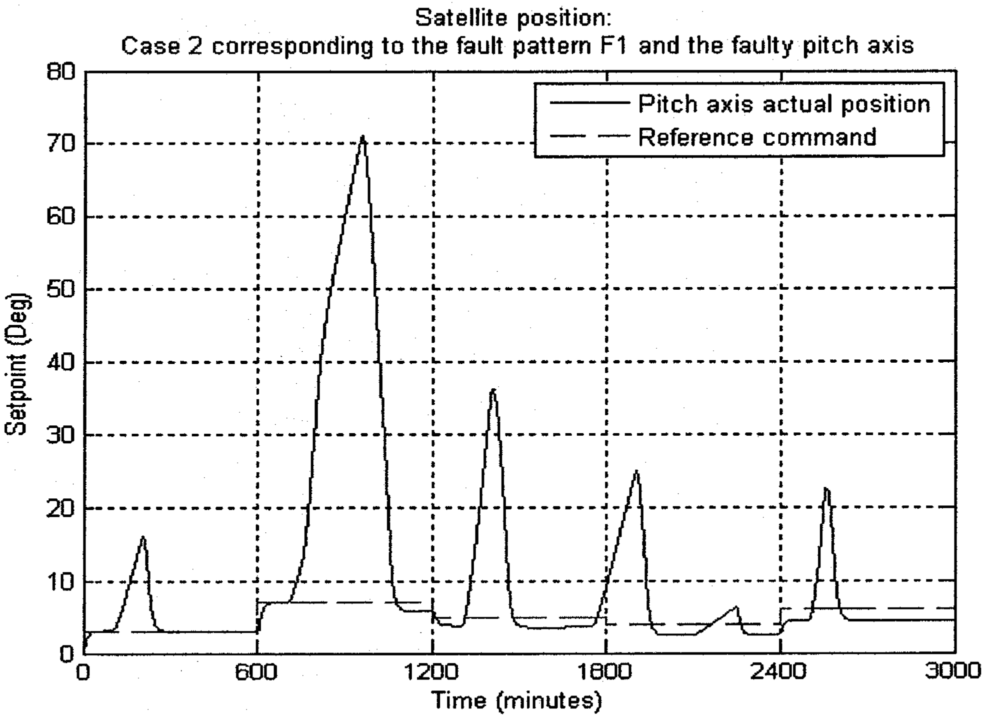

Figure 4.36 The satellite position for the faulty pitch axis in case 2 and the fault pattern F1

4.2.2.2. Two Faulty Axes with Fault Pattern F2

The detection results corresponding to the fault pattern F2 in case 2 are shown blow:

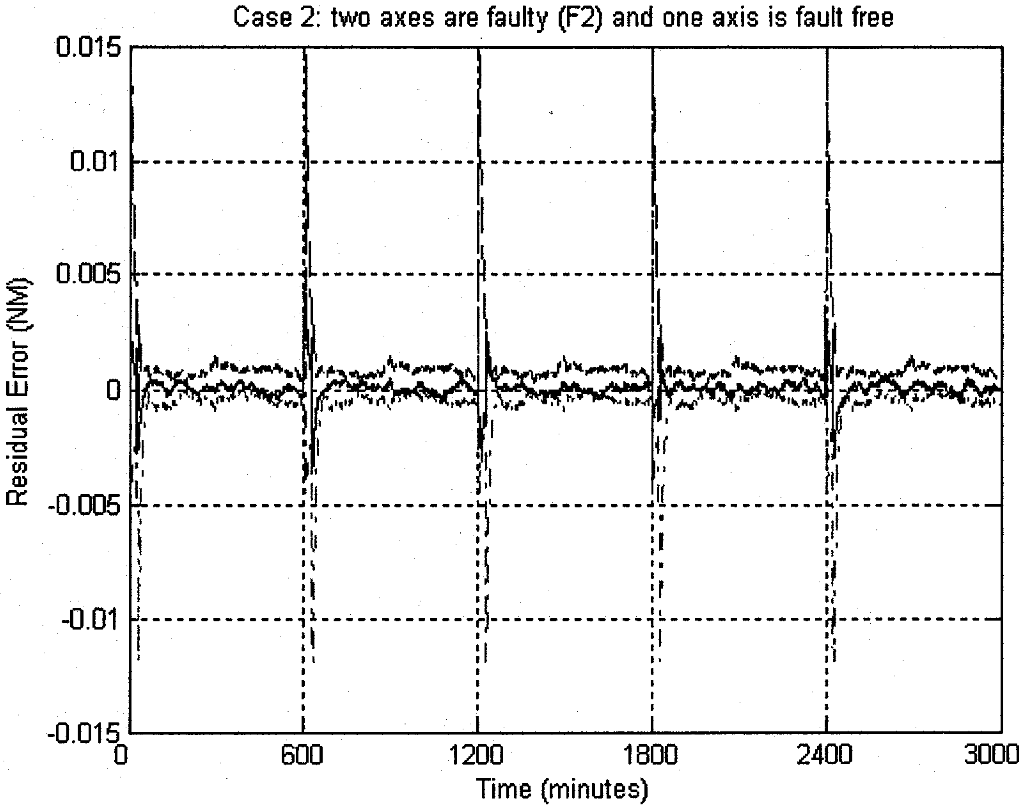

Figure 4.37 The detection results for the fault pattern F2 in case 2

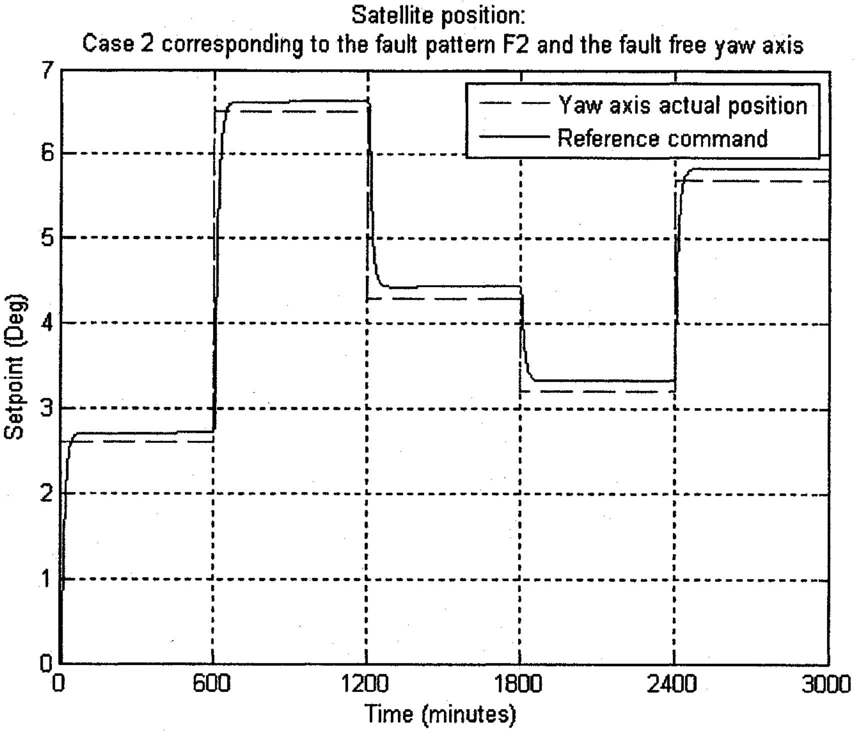

Figure 4.38 The satellite position for fault free yaw axis in case 2 and fault pattern F2

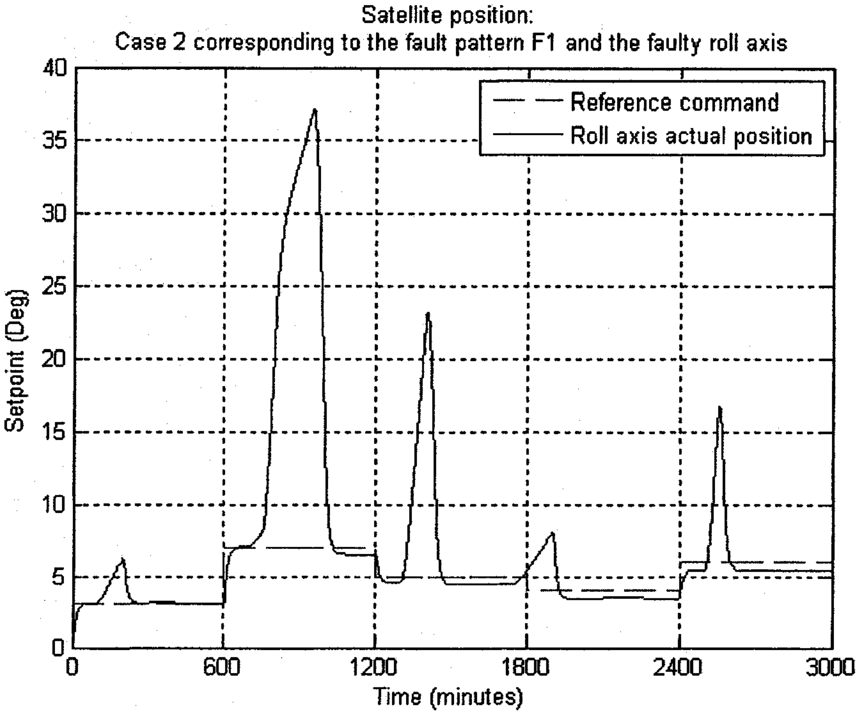

Figure 4.39 The satellite position for faulty roll axis in case 2 and fault pattern F2

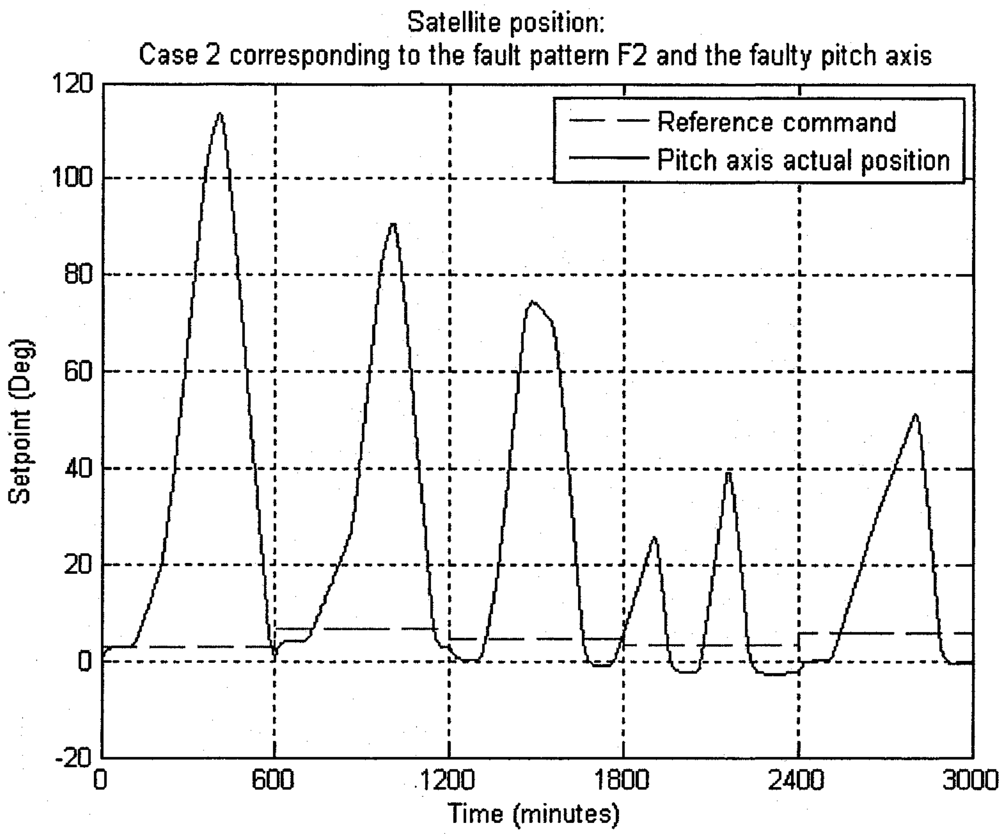

Figure 4.40 The satellite position for faulty pitch axis in case 2 and fault pattern F2

Figures 4.35, 4.36, 4.39, and 4.40 provide the satellite setpoint change in the other two axes under the fault pattern F1 and F2. It is revealed that the same fault pattern imposed on different axis lead to different detection results, which may be attributed to the presence of random disturbances and noise in the system. Therefore, under the certain circumstance of "small fault " it still may be possible that the detection scheme has the capacity to detect these faults even though the previous cases showed them to be undetectable.

4.2.3. Case 3: All Three Axes are Faulty

In this subsection, three axes have been subjected to the same fault pattern. The fault detection study is performed on the yaw axis.

4.2.3.1. Three Axes Have Fault Pattern F1

The detection results corresponding to the fault pattern F1 in case 3 are shown blow:

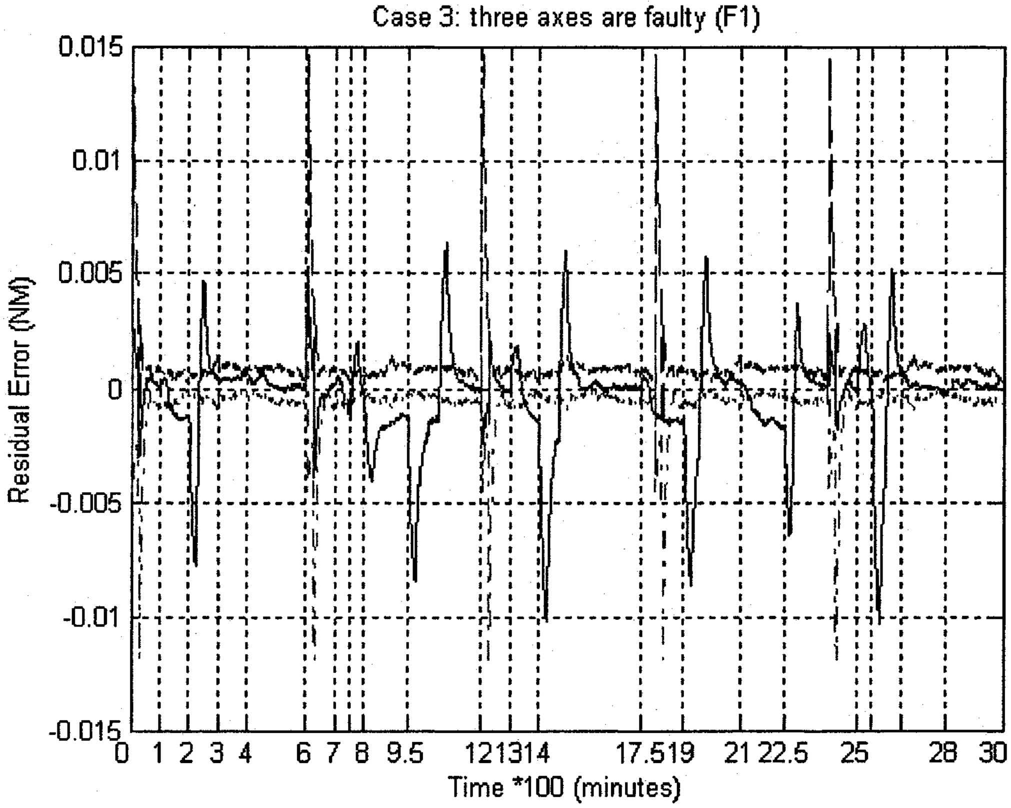

Figure 4.41 The detection results for the fault pattern F1 in case 3

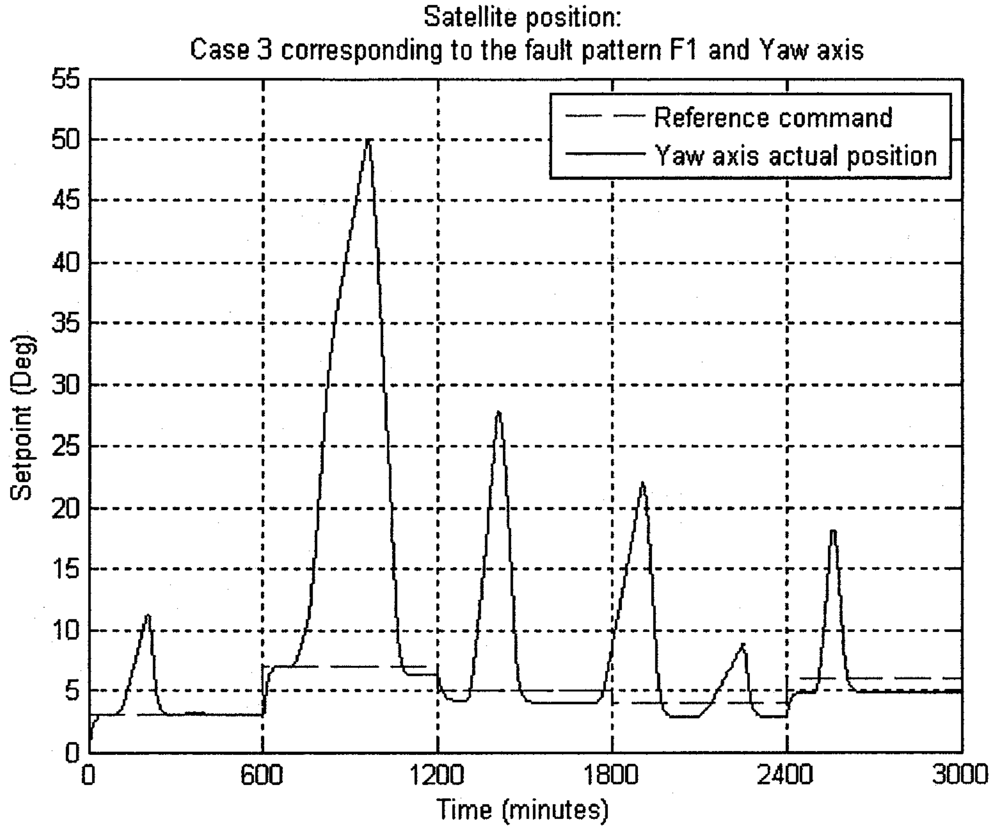

Figure 4.42 The satellite yaw axis position for the fault pattern F1 in case 3

The detection results for the individual setpoints are described in details below.

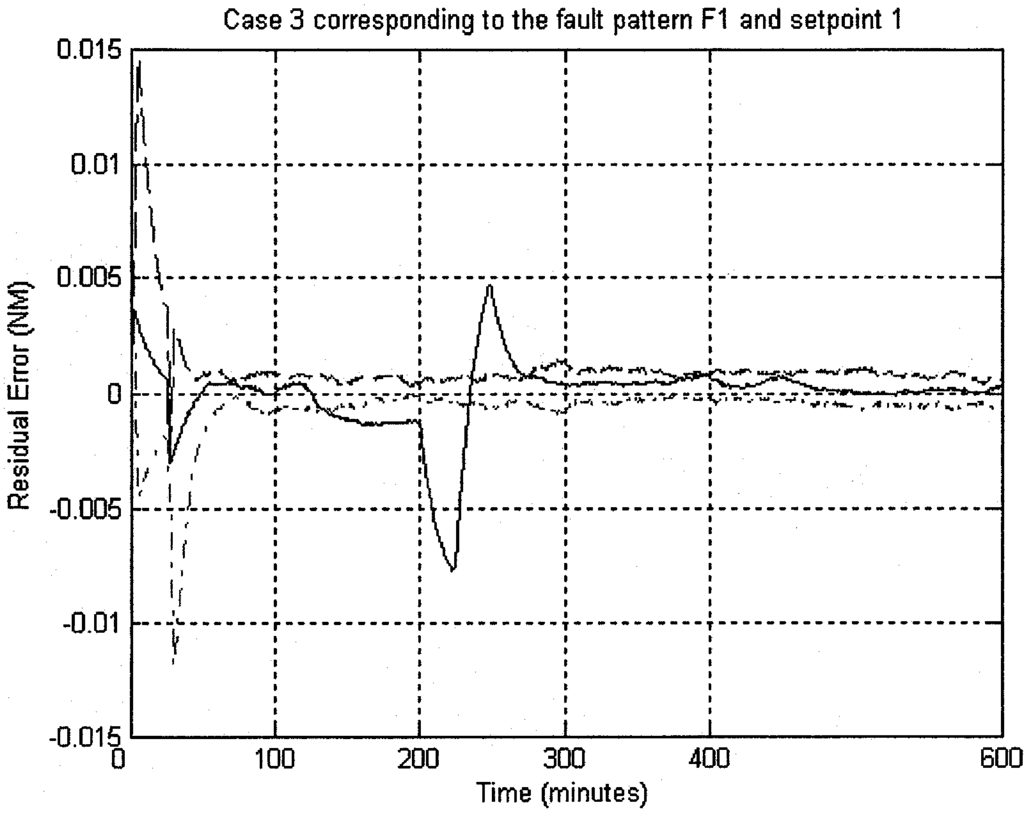

Figure 4.43 The detection results for the fault pattern F1 in case 3 and setpoint 1

The wheel has experienced voltage fault twice during this setpoint. Thirty-five minutes after the first voltage fault has happened, the residual curve exceeded the lower boundary (at t = 135.3 minutes). According to the fault decision criterion used in this thesis, namely 20 minutes, this fault was detected at t = 155 minutes. At t = 200 minutes, this fault was removed. The details for the transient phase are as follows: at t = 222.3 minutes, the residual curve reached its recovery undershoot and then it ascended and at t = 233.7 minutes the error entered the lower boundary. At t = 248.7 minutes, the residual curve reached its overshoot and then it descended and 20 minutes later, at t = 272.3 minutes, it entered the threshold again and settled down there. The second voltage fault has occurred at t = 300 minutes and it was recovered at t = 400 minutes where it was still undetected.

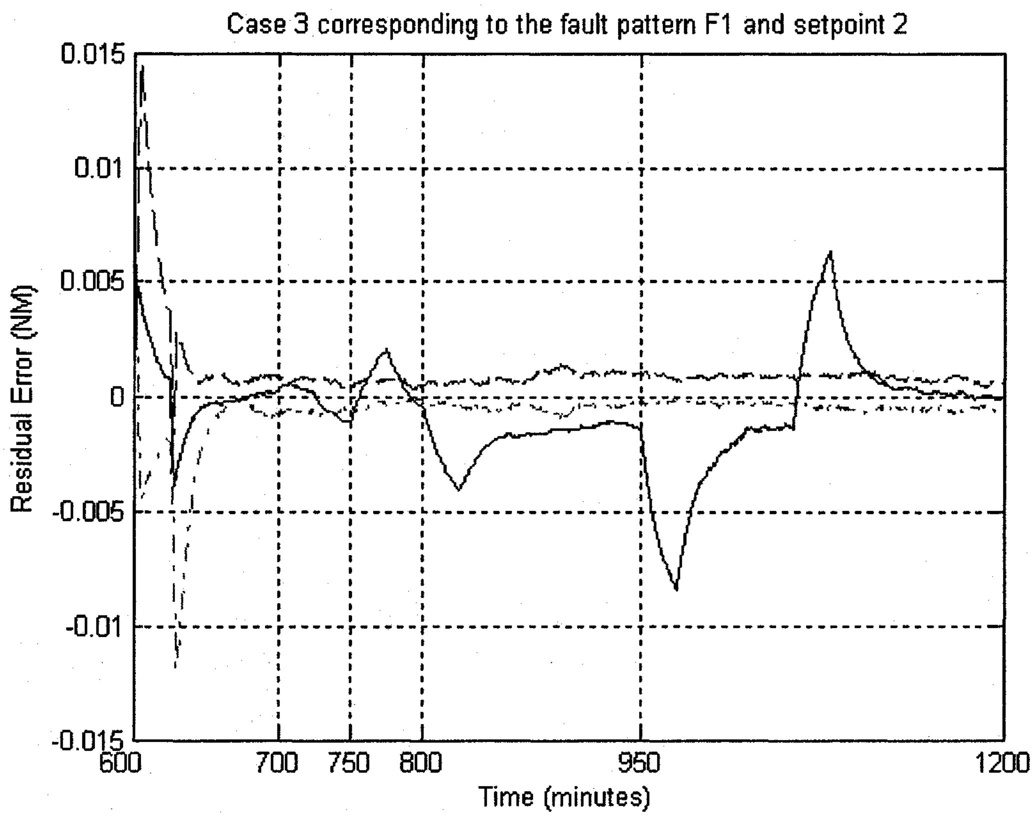

Figure 4.44 The detection results for the fault pattern F1 in case 3 and setpoint 2

For this setpoint, the residual curve exceeded the lower boundary at t = 737 minutes due to the voltage fault that has occurred at t = 700 minutes and it stayed beneath the lower boundary until the current fault has happened at t = 750 minutes. Since the current fault has occurred so close to the first fault there was no sufficient time to determine that the fault has occurred. Therefore, under this circumstance that fault was not detected. At t = 753 minutes, the residual curve entered the boundary and kept to ascend and cross the upper boundary at t = 761 minutes. At t = 775 minutes, the residual curve reached its detection overshoot and began to descend. At t = 783 minutes, the error reentered the boundary and at t = 792.4 minutes it exceeded the lower boundary eventually. At t = 800 minutes, the voltage fault signal was removed. The residual curve reached its recovery undershoot at t = 825 minutes and then began to ascend and settled down beneath the lower boundary. At t = 950 minutes, the current fault was removed. The residual curve reached its recovery undershoot at t = 975 minutes and began to ascend. From t = 990 minutes, the residual curve decreased its ascent rate and at t = 1055.3 minutes, it met its reflection point. At t = 1057 minutes, the residual curve entered the lower boundary and at t = 1060 minutes it crossed the upper boundary. At t = 1080 minutes, the error reached its recovery overshoot and began to descend. At t = 1105 minutes, the residual curve reentered the boundary and settled down there.

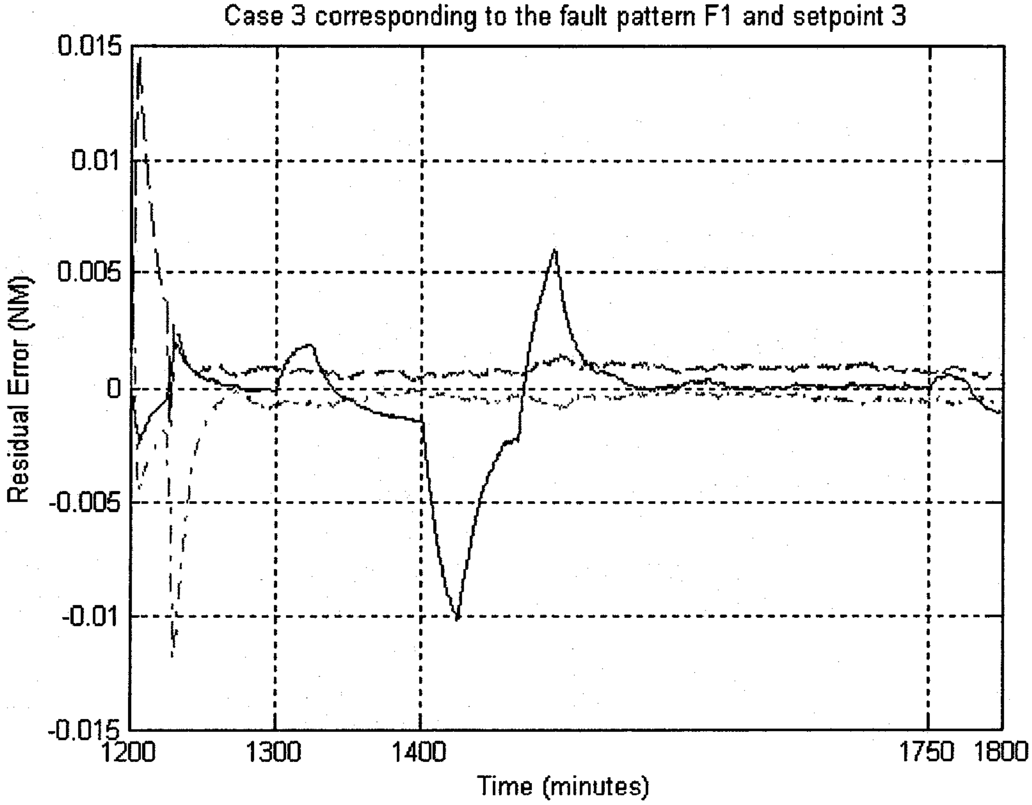

Figure 4.45 The detection results for the fault pattern F1 in case 3 and setpoint 3

The residual curve exceed the upper boundary with 5 minutes delay at t = 1304.7 minutes when the current signal became faulty. At t = 1323 minutes, the residual curve reached its detection overshoot and began to descend. At t = 1331 minutes, the error entered the upper boundary and at t = 1354 minutes it crossed the lower boundary. The fault signal was recovered at t = 1400 minutes. The residual curve reached its recovery undershoot at t = 1425.3 minutes and then ascended and at t = 1466 minutes the residual curve changes its shape (reflection point). At t = 1470 minutes, the residual curve reentered the boundary and at t = 1472 minutes it crossed the boundary again. At t = 1491.4 minutes, the residual curve reached its recovery overshoot and descended into the boundary at t = 1514.4 minutes. The residual curved settled in the boundary until t = 1783 minutes when the voltage fault was applied again. With 33 minutes delay, the residual curve exceeded the boundary again due to the present of the voltage fault.

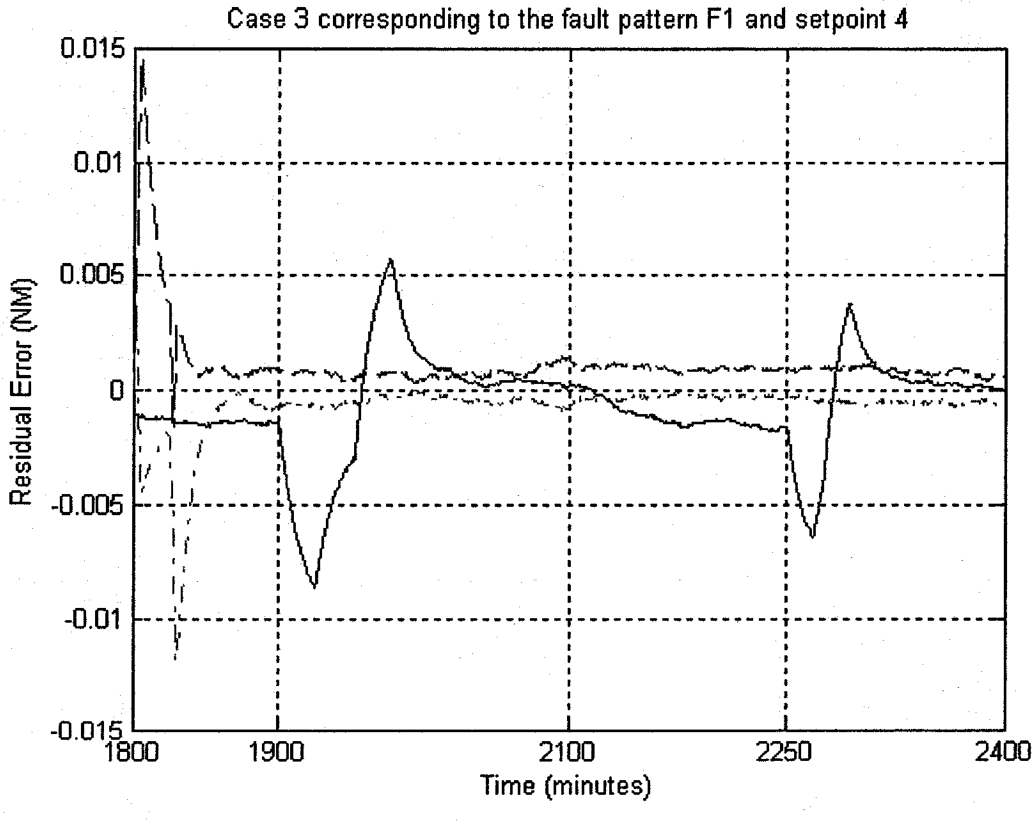

Figure 4.46 The detection results for the fault pattern F1 in case 3 and setpoint 4

At t = 1852 minutes, the residual curve crossed the lower boundary. Since the voltage fault signal was removed at 1900 minutes, the residual curve reached its recovery undershoot at t = 1926 minutes and at t = 1952 minutes the curve met its reflection point. At t = 1956 minutes, the residual curve entered the lower boundary and at t = 1959 minutes, it crossed it. At t = 1977.2 minutes, the residual curve reached its recovery overshoot. Eventually, it reentered the boundary at t = 2021.3 minutes and settled down there. Since a small current fault signal was applied at t = 2100 minutes, the residual curve exceeded the lower boundary at t = 2125 minutes. At t = 2250 minutes, the current signal became normal. The curve reached its recovery undershoot at t = 2268.3 minutes and crossed the lower boundary and upper boundaries at t = 2281.5 minutes and at t = 2284.3 minutes, respectively. Then, the residual curve reached its recovery overshoot at t = 2293.3 minutes. Finally, the residual curve reentered the boundary at t = 2316 minutes.

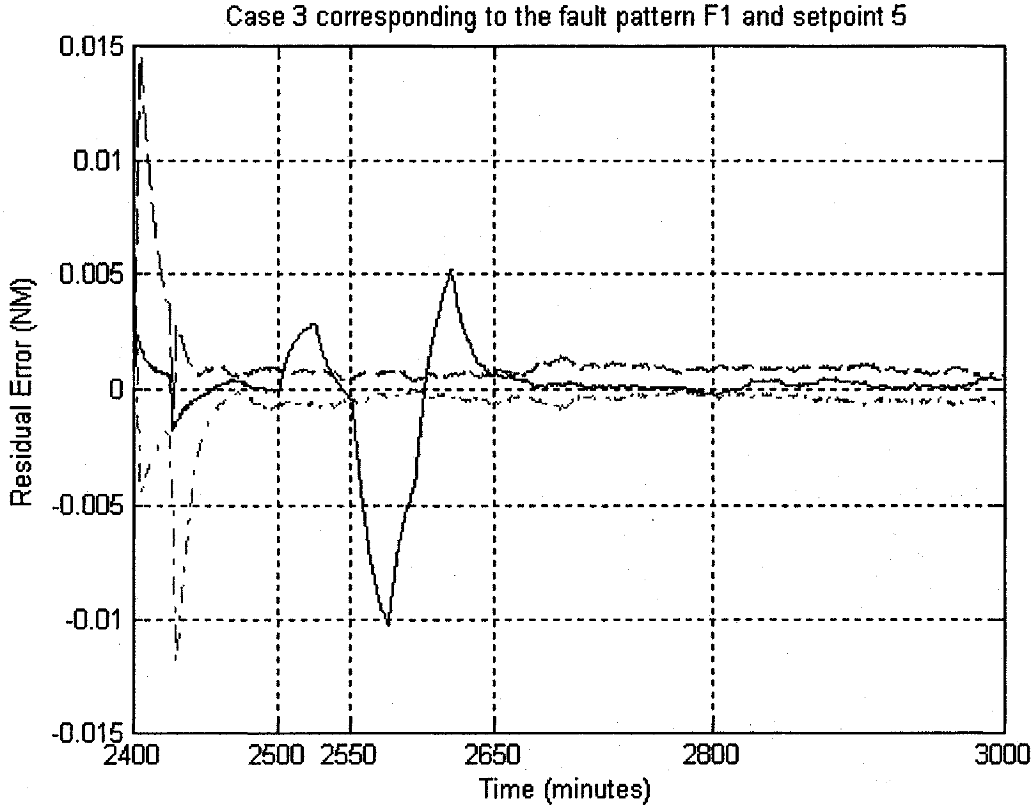

Figure 4.47 The detection results for the fault pattern F1 in case 3 and setpoint 5

The residual curve exceeded the boundary with 3.8 minutes delay when the voltage fault was applied at t = 2500 minutes. It reached its detection overshoot at t = 2523 minutes and entered the upper boundary at t = 2537.5 minutes. At t = 2551 minutes, the residual curve exceeded the lower boundary and coincidently the fault signal was removed at t = 2550 minutes. At t = 2575.8 minutes, the residual curve reached it recovery undershoot and at t = 2595 minutes, it met its reflection point. At t = 2601 minutes and t = 2602.6 minutes, the residual curve crossed the lower boundary and upper boundaries, respectively. At t = 2619.8 minutes, the residual curve reached its recovery overshoot and at time t = 2645 minutes, the residual curve reentered the boundary and settled down there. A small undetected current fault happened at t = 2650 minutes and was removed at t = 2800 minutes.

4.2.3.2. Three Axes with Fault Pattern F2

The detection results corresponding to the fault pattern F2 in case 3 are shown blow.

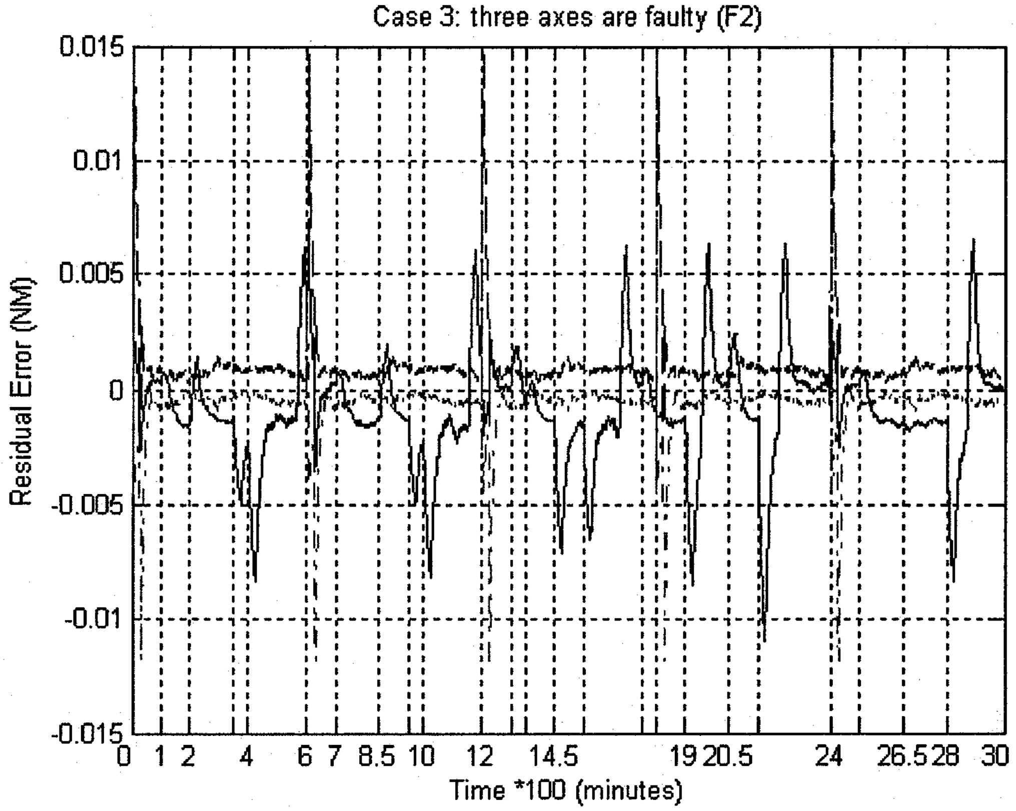

Figure 4.48 The detection results for the fault pattern F2 in case 3

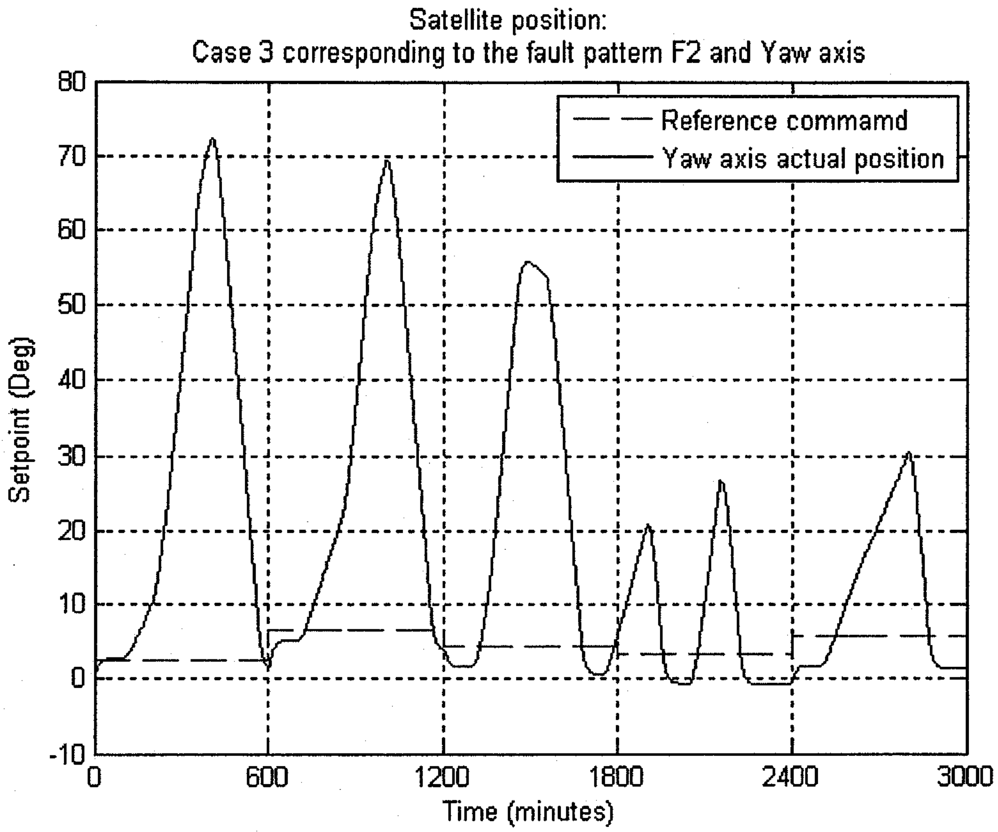

Figure 4.49 The satellite yaw axis position for the fault pattern F2 in case 3

The detection results for the individual setpoints are described in details below.

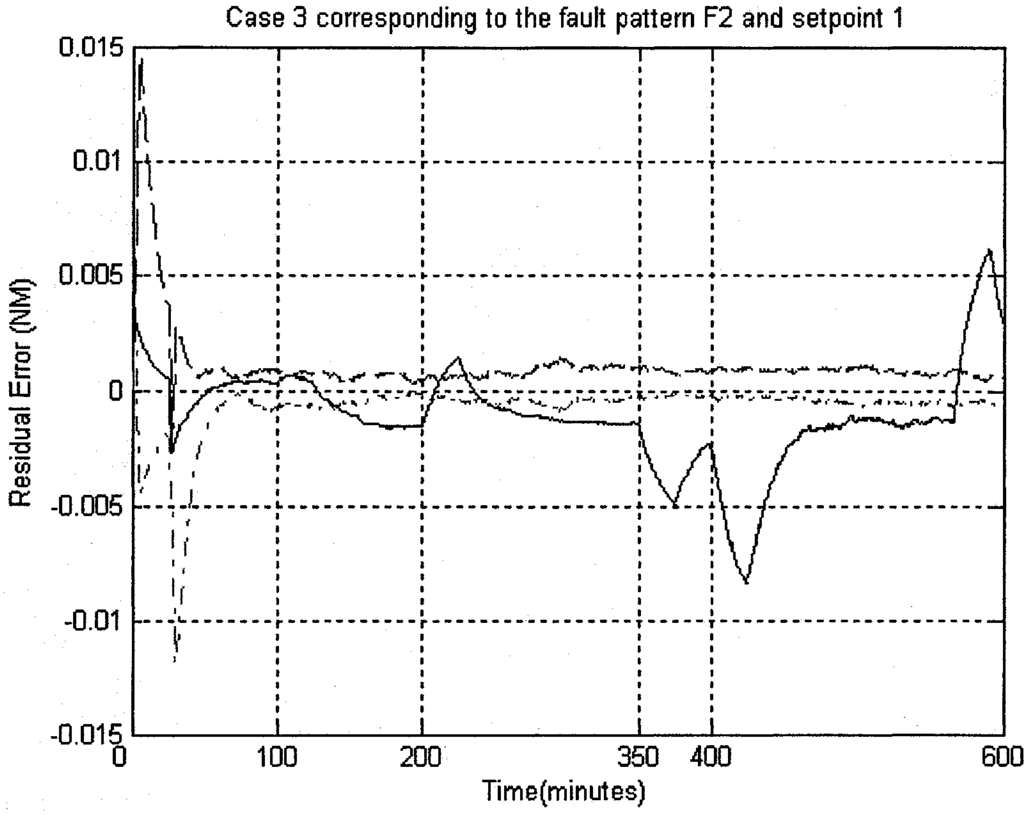

Figure 4.50 The detection results for the fault pattern F2 in case 3 and setpoint 1

Due to the presence of the current fault which has occurred at t = 100 minutes, the residual curve exceeded the lower boundary at t = 141.7 minutes. At t = 200 minutes, the voltage fault signal was also applied and the residual curve responded to this fault instantaneously by changing its descent tendency to an ascent. At t = 206.5 minutes, the residual curve entered the lower boundary and at t = 212.4 minute it crossed the upper boundary and then reached its detection overshoot at t = 225 minutes. The error descended and at t = 229.6 minutes the residual curve entered the upper boundary and exceeded the lower boundary at t = 248.8 minutes and then settled down beneath the lower boundary. At t = 350 minutes the current signal became normal, the residual curve reached its recovery undershoot at t = 375 minutes and then the residual curve ascended.

At t = 400 minutes, the voltage fault signal was also removed. The residual curve reached its recovery undershoot at t = 424.6 minutes then around t = 460 minutes the residual curve stopped ascending and began to extend horizontally. At t = 565.6 minutes, the error began to ascend and reached its recovery overshoot at t = 590.5 minutes and then descended. At t = 600 minutes, the descent seemed to not has finished yet.

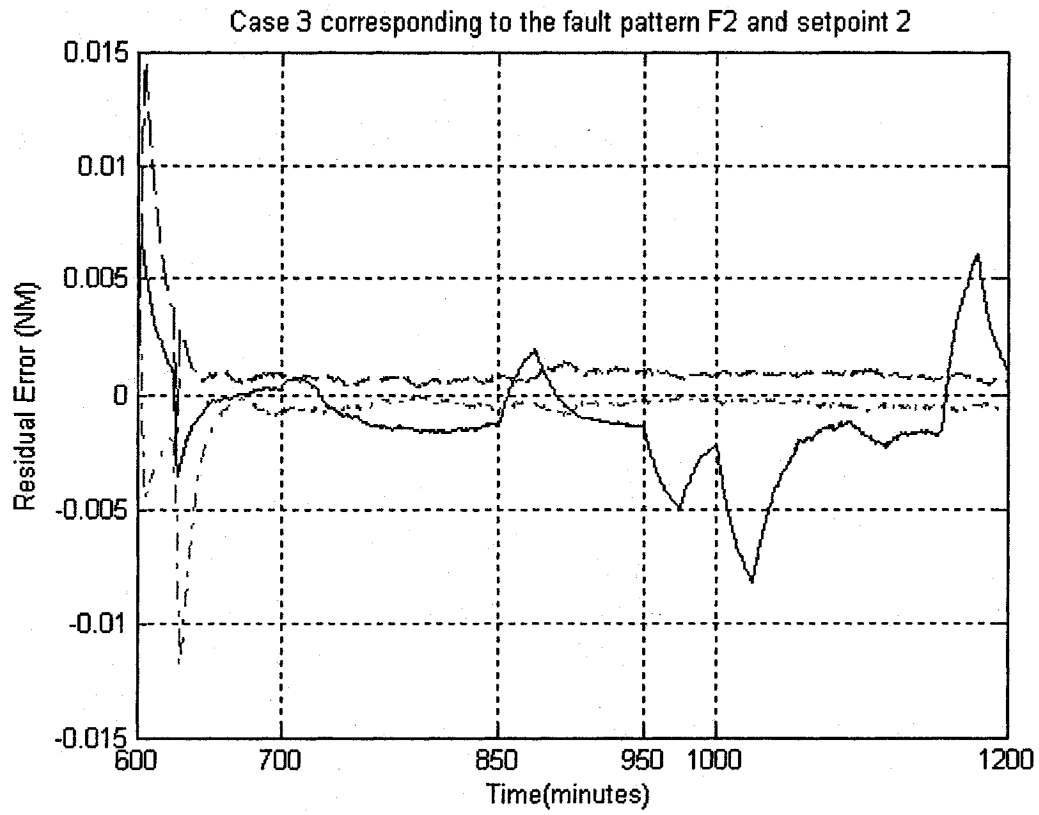

Figure 4.51 The detection results for the fault pattern F2 in case 3 and setpoint 2

At t = 600 minutes, the position change command was applied and it blocked the pervious recovery transient phase. This situation did not affect the FDI performance but it exerted negative influence on the satellite position pointing as shown in Figure 4.49. At t = 700 minutes, the voltage fault signal was applied and after 39.7 minutes the residual curve exceeded the lower boundary and settled beneath the boundary. At t = 850 minutes, the current signal became faulty and the residual curve ascended due to this fault. At t =

853 minutes, the error entered the lower boundary and at t = 860.8 minutes it exceeded the upper boundary. At t = 875.2 minutes, the residual curve reached its detection overshoot and began to descend. At t = 950 minutes, the current signal became normal and the residual curve reached its recovery undershoot at t = 975.2 minutes and then it ascended. Without sufficient time to complete its transient, at t = 1000 minutes, the voltage signal was also removed. At t = 1025.2 minutes, the residual curve reached its recovery undershoot and began to ascend. At t = 1057.4 minute, the residual curve stopped its ascent tendency and extended horizontally until t =1155 minutes. From that reflection point, t = 1155 minutes, the residual cùrve began ascending and reached its recovery overshoot at t = 1180 minutes. Then, it descended while the transient was not completed at t = 1200 minutes.

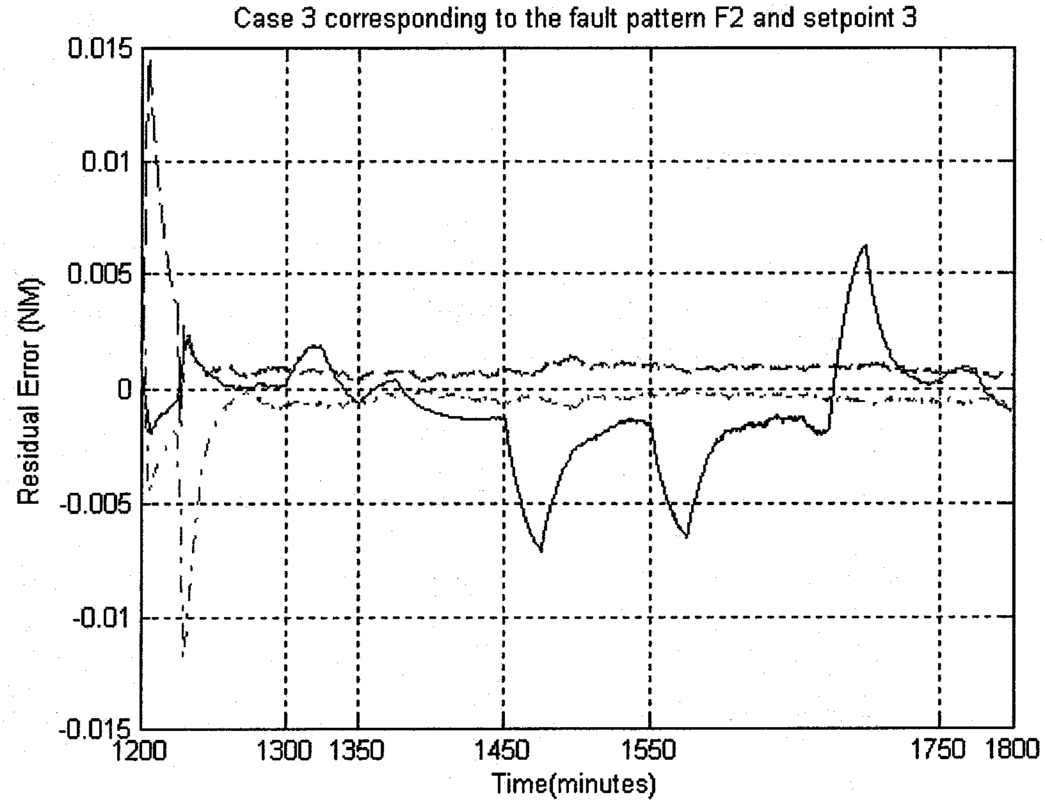

Figure 4.52 The detection results for the fault pattern F2 in case 3 and setpoint 3

At t = 1300 minutes, the current fault from 1V to 0.3V was applied and the residual curve exceeded the boundary with short delay, at t = 1304.3 minutes. At t = 1325.5 minutes, the residual curve reached its detection overshoot and at t = 1331.4 minutes it reentered the upper boundary. However, since this period (from 1304.3 to 1331.4) is longer than the detection criteria time (that is 20 minutes), a fault could be considered to have occurred in this period. At t = 1350.2 minutes, the residual curve exceeded the lower boundary and coincidently at t = 1350 minutes the voltage signal was faulty. The residual curve entered the boundary at t = 1351.7 minutes and crossed it at t = 1384.3 minutes and then settled beneath the lower boundary. At t = 1450 minutes, the current signal became normal and the residual curve reached its recovery undershoot at t = 1475.4 minutes and then began its ascent. At t = 1496 minutes, the ascent rate decreased. At t = 1550 minutes, the voltage became normal and the residual curve reached its recovery undershoot at t = 1575.6 minutes. From t = 1597.7 minutes, the residual curve began to extend horizontally and at t = 1672.7 minutes, the residual curve began to ascend and then reached its recovery overshoot at t = 1697.8 minutes. At t = 1722.2 minutes, the error reentered the boundary. Later at = 1750 minutes, the current fault was applied again and with 37 minutes delay, at t = 1786.5 minutes, the residual curve exceeded the boundary again.

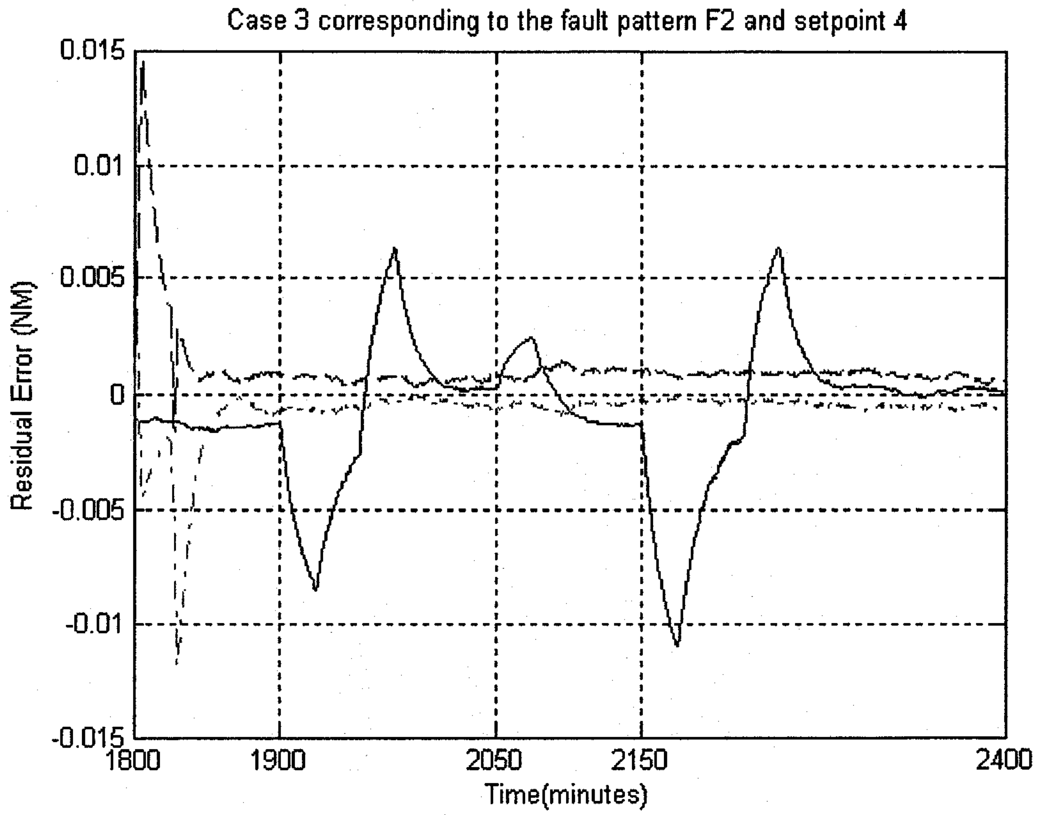

Figure 4.53 The detection results for the fault pattern F2 in case 3 and setpoint 4

The residual curve exceeded the lower boundary horizontally at t = 1852.4 minutes due to the current fault. At t = 1900 minutes, the current fault signal was removed. The residual curve reached its recovery undershoot at t = 1925.5 minutes and began to ascend. At t = 1955 minutes, the residual curve met its reflection point. At t = 1957.8 minutes and t = 1960.2 minutes, the residual curve exceeded the lower and upper boundaries, respectively and reached its recovery overshoot at t = 1980 minutes. Then, it descended and at t = 2010 minutes, it entered the boundary. At t = 2050 minutes, the voltage fault signal was applied and the residual curve exceeded the boundary with 4 minutes delay, at t =2053.5 minutes. It reached it detection overshoot at t = 2075.7 minutes and then descended and crossed the upper and lower boundaries at t = 2082.7 minutes and t =2101.4 minutes, respectively. At t = 2150 minutes, the voltage signal became normal and the residual curve reached its recovery undershoot at t = 2175.6 minutes and then ascended. At t = 2220 minutes, the residual curve changes its ascending shape (reflection point). At t = 2223 minutes and t = 2225.3 minutes, it crossed the lower and upper boundaries, respectively and reached it recovery overshoot at t = 2245.5 minutes. Then, it descended and reentered the boundary at t = 2269.9 minutes and settled down inside there.

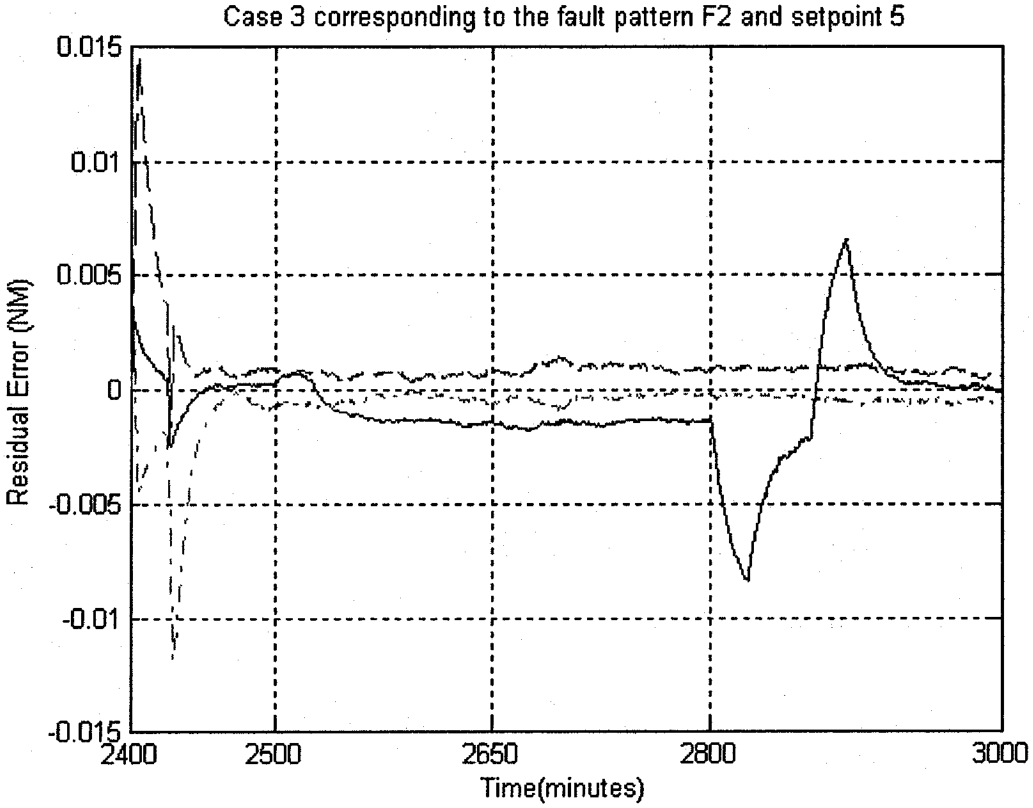

Figure 4.54 The detection results for the fault pattern F2 in case 3 and setpoint 5

The current fault signal was applied at t = 2500 minutes and with 39.7 minutes delay at t = 2539.7 minutes the residual curve exceeded the boundary. At t = 2650 minutes, the current became normal but the voltage became faulty. The residual curve did not change its tendency due to these two faults. At t = 2800 minutes, the voltage fault was removed and the residual curve reached its recovery undershoot at t = 2825.9 minutes.

The residual curve tended to extend horizontally from t = 2843.8 minutes and at t = 2868.8 minutes, the residual curve became to ascend and crossed the lower and upper boundaries at t = 2871.4 minutes and t = 2874.1 minutes, respectively. At t = 2893.8 minutes, the residual curve reached its recovery overshoot and descended. At t = 2919.6 minutes, it reentered the boundary and settled inside the boundary.

To summarize and specify the differences that are presented by the influences from other axes, Table 4.6 provides all the details during the application of faults and the recovery phases of these faults.

Table 4.6 Summary of the detection results corresponding to influences from other axes

Set point	Fault Seq.	Event	F1		F2	
			Case 1	Case 3	Case 1	Case 3
Sp1	1st fault	Occurrence	Detected with 33 mins. delay	Detected with 35.3 mins. delay	Detected with 35 mins. delay	Detected with 41.7 mins. delay
		Recovery	1. It took 7 min. to reach the undershoot; 2. It took 26 min. to reach the lower bound and 28 min. to the upper bound; 3. It took 32.8 min. to reach the overshoot; 4. It took 50 min. to reenter the boundary.	1. It took 22.3 min. to reach the undershoot; 2. It took 33.7 min. to reach the lower bound and 35.9 minutes to the upper bound; 3. It took 48.7 min. to reach the overshoot; 4. It took 72.3 min. to reenter the boundary.	1. It took 25 min. to reach the undershoot; 2. The transient did not complete.	1. It took 25 min. to reach the undershoot; 2. The transient did not complete.

		Properties	For detection: 2 more minutes delay caused by the influence For recovery: total 50 min. to complete transient in case 1 and total 72.3 min. in case 2. It elongates 22.3 minutes.		For detection: 6 minutes delay caused by the influence. For recovery: Not much difference. Both transients were not completed due to the occurrence of the next event.	
	2nd fault	Occurrence	Undetected	Undetected	1. It took 5 min. to exceed the lower bound and 12 min. for the upper bound; 2. It took 25 min. to reach the overshoot. 3. It took 31 min. to reenter the upper bound and 40 min. to exceed the lower bound.	1. It took 6.5 min. to exceed the lower bound and 12.4 min. for the upper bound; 2. It took 25 min. to reach the overshoot; 3. It took 29.6 min. to reenter the upper bound and 48.8 min. to exceed the lower bound.
		Recovery	None	None	1. It took 25 min. to reach the undershoot; 2. It extended 47.5 minutes horizontally. 3. It took 109 min. to reach the lower bound and 111.2 min. for the upper bound; 4. It took 132 min. to reach the overshoot; 5. It took 163 min. to reenter the boundary.	1. It took 24.6 min. to reach the undershoot; 2. It extended 105.6 minutes horizontally. 3. It took 166.7 min. to reach the lower bound and 169.5 min. for the upper bound; 4. It took 190.5 min. to reach the overshoot; 5. The transient did not complete.

		Properties	Since this fault is small, the neural network can not detect it.		For detection: 2 more minutes delay caused by the influence For recovery: total 163 min. to complete he transient in case 1 and it will take more than 200 minutes in case 2 since the transient is blocked by the occurrence of the next event. The influence elongated the procedure more than 40 minutes.	
Sp2	1st fault	Occurrence	Detected with 33 minutes delay	Detected with 37 minutes delay.	Detected with 35 minutes delay.	Detected with 39.7 minutes delay.
		Recovery	1. It took 25 min. to reach the undershoot;	1. It took 25 min. to reach the undershoot;	1. It took 25 min. to reach the undershoot; 2. It extended 21.5 minutes horizontally. 3. It took 74 min. to reach the lower bound and 76 minutes for the upper bound; 4. It took 96 min. to reach the overshoot; 5. It took 127 min. to reenter the boundary.	1. It took 25.2 min. to reach the undershoot; 2. It extended 97.6 minutes horizontally. 3. It took 157 min. to reach the lower bound and 159 minutes for the upper bound; 4. It took 180 min. to reach the overshoot; 5. The transient did not complete.
		Properties	For detection: 4 more minutes delay caused by the influence For recovery: not much difference.		For detection: 5 more minutes delay caused by the influence For recovery: total 127 min. to complete the transient in case 1 and it will take more than 200 minutes in case 2. The influence elongated the procedure more than 70 minutes.	

	2nd fault	Occurrence	1. It took 3 min. to exceed the lower bound and 11 min. for the upper bound; 2. It took 25 min. to reach the overshoot; 3. It took 33 min. to reenter the upper bound and 42.4 min. to exceed the lower bound	1. It took 3 min. to exceed the lower bound and 11 min. for the upper bound; 2. It took 25 min. to reach the overshoot; 3. It took 33 min. to reenter the upper bound and 42.4 min. to exceed the lower bound	1. It took 3 min. to enter the lower bound and 11 min. for the upper bound. 2. It took 25 min. to reach the overshoot. 3. It took 31 min. to reenter the upper bound and 52 min. to exceed the lower bound	1. It took 3 min. to enter the lower bound and 11 min. for the upper bound. 2. It took 25.2 min. to reach the overshoot. 3. It took 31 min. to reenter the upper bound and 52 min. to exceed the lower bound
		Recovery	1. It took 25 minutes to reach the undershoot; 2. It extended 14.6 minutes horizontally. 3. It took 55 min. to reach the lower bound and 57 min. for the upper bound; 4. It took 77 min. to reach the overshoot. 5. It took 110 min. to reenter the boundary.	1. It took 25 minutes to reach the undershoot; 2. It extended 65.3 minutes horizontally. 3. It took 107 min. to reach the lower bound and 110 min. for the upper bound; 4. It took 130 min. to reach the overshoot. 5. It took 155 min. to reenter the boundary.	1. It took 25 min. to reach the undershoot; 2. The transient did not complete.	1. It took 25.2 min. to reach the undershoot; 2. The transient did not complete.
		Properties	For detection: no difference. For recovery: total 110 min. to complete the transient in case 1 and 155 minutes in case 2. The influence elongated the transient 45 minutes.		No difference for both detection and recovery.	

Sp3	1st fault	Occurrence	1. It took 6 min. to exceed the upper bound and 25 min. to reenter it. 2. It took 44 minutes to exceed the lower bound	1. It took 4.7 min. to exceed the upper bound and 31 min. to reenter it. 2. It took 54 minutes to exceed the lower bound	1. It took 5 min. to exceed the upper bound and 26 min. to reenter it. 2. It took 45 min. to exceed the lower boundary.	1. It took 4.3 min. to exceed the upper bound and 31.4 min. to reenter it. 2. It took 50.2 min. to exceed the lower boundary.
		Recovery	1. It took 26 min. to reach the undershoot; 2 .It took 43.4 min. to reach the lower bound and 45.5 min. to the upper bound; 3. It took 62 min. to reach the overshoot; 4. It took 84 min. to reenter the boundary.	1. It took 25.3 min. to reach the undershoot; 2. It took 70 min. to reach the lower bound and 72 min. to the upper bound; 3. It took 91.4 min. to reach the overshoot; 4. It took 114.4 min. to reenter the boundary.	1. It took 26 min. to reach the undershoot; 2. It took 42.5 min. to decrease the ascend rate.	1. It took 26 min. to reach the undershoot; 2. It took 46 min. to decrease the ascend rate.
		Properties	For detection: the curve outside the boundary was extended: less detection delay and more ambiguous time. For recovery: total 84 min. to complete the transient in case 1 and 114.4 minutes in case 2 The influence elongated the transient 30 minutes.		For detection: the curve outside the boundary was extended: less detection delay and more ambiguous time. For recovery: not much difference.	
	2nd fault	Occurrence	It took 30 min. to exceed the lower boundary.	It took 33 min. to exceed the lower boundary.	It took 3.6 min. to enter the lower bound and 35 minute to exceed it.	It took 1.7 min. to enter the lower bound and 34.3 minute to exceed it.

		Recovery	None	None	1. It took 25.6 min. to reach the undershoot; 2. It extended 15 minutes horizontally. 3. It took 56.5 min. to reach the lower bound and 58.3 min. to the upper bound; 4. It took 79 min. to reach the overshoot; 5. It took 106 min. to reenter the boundary.	1. It took 25.6 min. to reach the undershoot; 2. It extended 75 minutes horizontally. 3. It took 125.4 min. to reach the lower bound and 128.2 min. to the upper bound; 4. It took 147.8 min. to reach the overshoot; 5. It took 172.2 min. to reenter the boundary.
		Properties	For detection: 3 minutes delay by the influence. For recovery: the fault was not recovered in this setpoint.		For detection: not much difference. For recovery: total 106 min. to complete the transient in case 1 and 172.2 minutes in case 2. The influence elongated the transient 66 minutes.	
Sp4	1st fault	Occurrence	1. Detected with 30 min. delay in the last setpoint. 2. It took 52.3 min. to exceed the lower boundary	1. Detected with 33 min. in the last setpoint. 2. It took 52.3 min. to exceed the lower boundary	1. Detected with 31 minutes delay in the last setpoint. 2. It took 52.3 min. to exceed the lower boundary	1. Detected with 36.5 minutes delay in the last setpoint. 2. It took 52.4 min. to exceed the lower boundary

		Recovery	1. It took 12 min. to reach the undershoot; 2. It took 28 min. to reach the lower bound and 30 min. to the upper bound; 3. It took 37 min. to reach the overshoot; 4. It took 65 min. to reenter the boundary.	1. It took 26 min. to reach the undershoot; 2. It took 56 min. to reach the lower bound and 59 min. to the upper bound; 3. It took 77.2 min. to reach the overshoot; 4. It took 121.3 min. to reenter the boundary.	1. It took 20 min. to reach the undershoot; 2. It took 32 min. to reach the lower bound and 34.5 min. to the upper bound; 3. It took 45 min. to reach the overshoot; 4. It took 70 min. to reenter the boundary.	1. It took 25.5 min. to reach the undershoot; 2. It took 57.8 min. to reach the lower bound and 60.2 min. to the upper bound; 3. It took 80 min. to reach the overshoot; 4. It took 110 min. to reenter the boundary.
		Properties	For detection: 3 minutes delay by the influence. For recovery: total 65 min. to complete the transient in case 1 and 121.3 minutes in case 2 The influence elongated the transient 56 minutes.		For detection: 6 minutes delay by the influence. For recovery: total 70 min. to complete the transient in case 1 and 110 minutes in case 2. The influence elongated the transient 40 minutes.	
	2nd fault	Occurrence	Undetected	It took 25 min. to reach the undershoot;	1. Detected with 4 minutes delay and 29 min. to reenter the upper bound; 2. It took 51 min. to exceed the lower boundary;	1. Detected with 4 minutes delay and 32.7 min. to reenter upper bound; 2. It took 51.4 min. to exceed the lower boundary;
		Recovery	None	1. It took 18.3 min. to reach the undershoot; 2. It took 31.3 min. to reach the lower bound and 34.3 min. to the upper bound; 3. It took 43.3 min. to reach overshoot; 4. It took 66	1. It took 26 min. to reach the undershoot; 2. It took 48 min. to change its shape. 3. It took 53.4 min. to reach the lower bound and 55.5 min. to the upper bound; 4. It took 74 min. to reach the	1. It took 26 min. to reach the undershoot; 2. It took 70 min. to change its shape. 3. It took 73 min. to reach the lower bound and 75.5 min. to the upper bound; 4. It took 95.5 min. to reach the

				min. to reenter the boundary.	overshoot; 5. It took 99 min. to reenter the boundary.	overshoot; 5. It took 119.9 min. to reenter the boundary.
		Properties	For detection: with influence from other axes, the small fault became detected For recovery: total 66 min. for the transient in case2		For detection: no difference. For recovery: total 99 min. to complete the transient in case 1 and 119.9 minutes in case 2. The influence elongated the transient 20 minutes.	
Sp5	1st fault	Occurrence	1. Detected with 4 minute delay and it took 30 minute to reenter the upper bound. 2. It took 47 minutes to exceed the lower boundary.	1. Detected with 4 min. delay and it took 37.5 minute to reenter the upper bound 2. It took 51 minutes to exceed the lower boundary.	It took 36 min. to exceed the boundary	It took 39.7 min. to exceed the boundary
		Recovery	1. It took 25.9 min. to reach the undershoot; 2. It took 38.2 min. to reach the lower bound and 39.4 min. to the upper bound; 3. It took 51.4 min. to reach the overshoot; 4. It took 77.5 min. to reenter the boundary.	1. It took 25.8 min. to reach the undershoot; 2. It took 51 min. to reach the lower bound and 52.6 min. to the upper bound; 3. It took 69.8 min. to reach the overshoot; 4. It took 95 min. to reenter the boundary.	None.	None.
		Properties	For detection: not much difference. For recovery: total 77.5 min. to complete the transient in case 1 and 95 minutes in case 2. The influence elongated the transient 17 minutes.		For detection: with 4 minutes delay caused by the influence. For recovery: No recovery properties.	
	2nd fault	Occurrence	Undetected	Undetected	No delay	No delay

		Recovery	None	None	1. It took 25.6 min. to reach the undershoot; 2. It took 35.3 min. to reach the lower bound and 37 min. to the upper bound; 4. It took 51 min. to reach the overshoot; 5. It took 74 min. to reenter the boundary.	1. It took 25.9 min. to reach the undershoot; 2. It took 71.4 min. to reach the lower bound and 74.1 min. to the upper bound; 4. It took 93.8 min. to reach the overshoot; 5. It took 119.6 min. to reenter the boundary.
		Properties	The fault was undetected in both cases.		For detection: no difference. For recovery: total 74 min. to complete the transient in case 1 and 119.6 minutes in case 2. The influence elongated the transient 46minutes.	

4.2.4. Case Study 4: Robustness Investigation

In this subsection, the robustness properties of the neural network scheme under case 1 with F1 fault pattern are investigated. The nominal ripple noise applied to the wheel is shown in Table 2.2 with a ripple value of 0.22. This ripple noise is increased to different levels at each setpoint as shown in Figure 4.55.

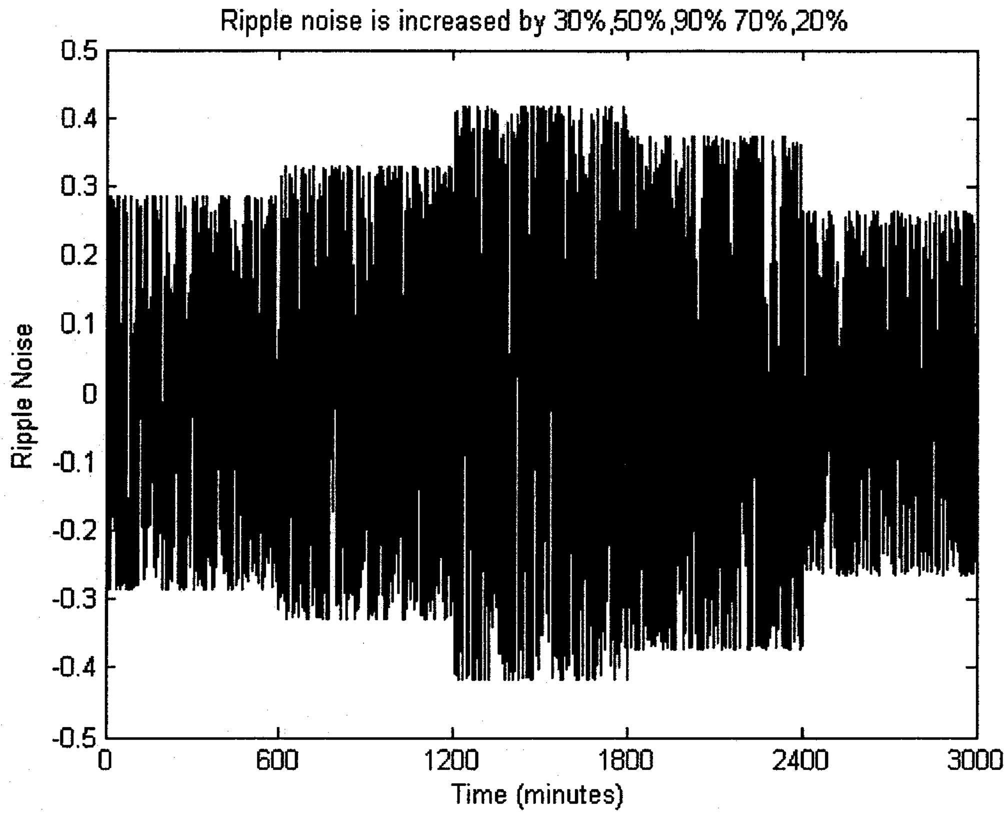

Figure 4.55 The increased ripple noises corresponding to different setpoint positions

The detection results associated under this changing noise value are shown below.

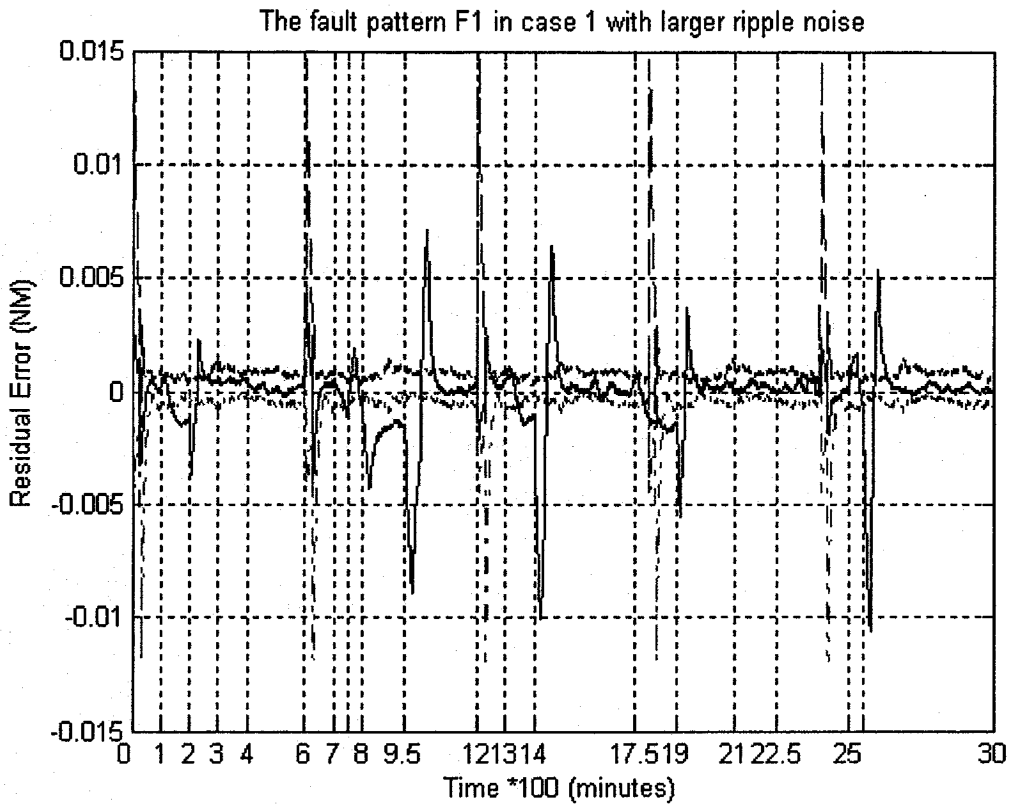

Figure 4.56 Detection results for the fault pattern F1 in case 1 with large noise

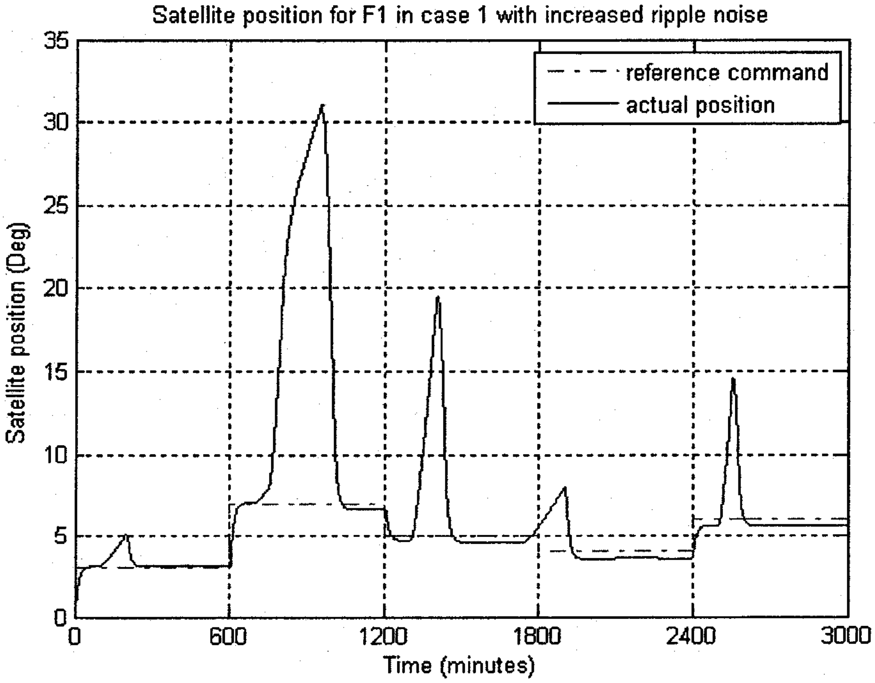

Figure 4.57 The satellite position for the fault pattern F1 in case 1 with large noise

As can be observed from the above simulation results, by increasing the ripple noise the rate of false alarms for detecting faults does not change and increase. This implies that the proposed neural network is quite robust to these uncertainties. Since the detection results (in terms of the detection time and recovery time) for a given fault pattern can be slightly different due to the randomly generated noise and disturbances that are applied to the system, a strict comparison between various noise and disturbances scenarios are not conducted here.

4.2.5. Case Study 5: Detection Results Using a Different Definition of the Residual Error

All the detection results that are provided in this thesis have the property that the residual signals decrease but the error curve lays below the threshold curves at steady state. There is however a more appropriate representation for the error signals. Given the fact that the threshold curve is generated based on the normal operation of the satellite and the neural networks are also trained to represent and capture the satellite behavior under normal conditions, if error signals in the wheel decrease and the neural networks always correspond to normal data, then the difference between the wheel output and the network output should increase rather than decrease as shown in our previous figures. The reason for this behavior is that the residual error adopted in this these is defined as $error = y - o$, where y is the output of the wheel and o is the output of the neural network. If, on the other hand, the error definition is changes to $error = o - y$, then the detection results figures will be more consistent with the above common sense interpretations. As an illustration, Figures 4.58 and 4.59 show the corresponding difference by using this new residual error definition.

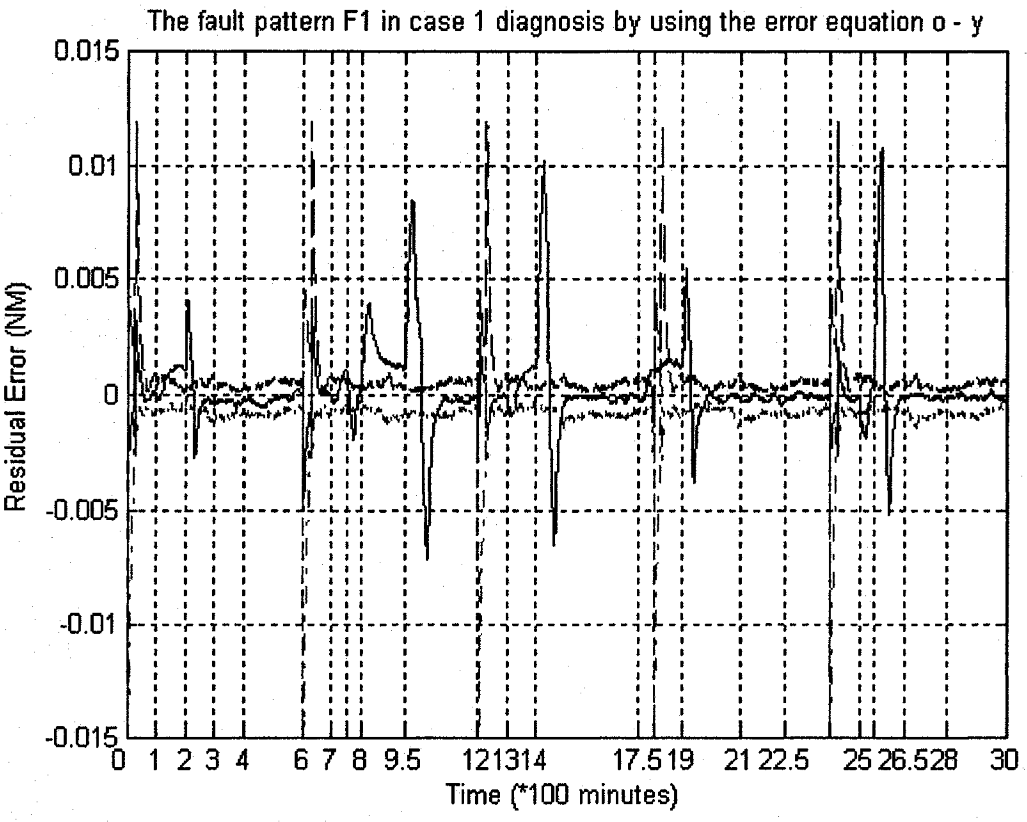

Figure 4.58 The detection results for the fault pattern F1 in case 1 using a different error definition

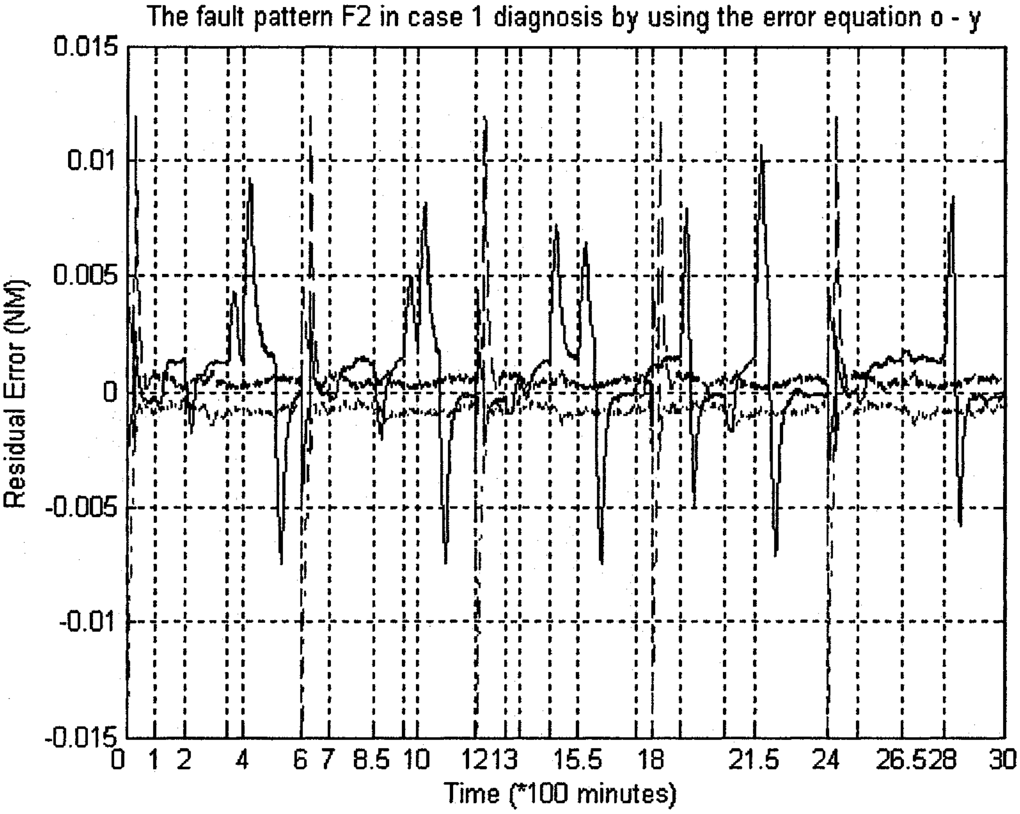

Figure 4.59 The detection results for the fault pattern F2 in case 1 using a different error definition

4.3. Conclusions

In this chapter, the FDI scheme for satellite attitude control by using an adaptive time delay neural network is investigated. The detection results for either V_{bus} fault or current fault in the reaction wheel for individual setpoint changes of the satellite are summarized and provided in details first. Next the robustness of the proposed neural network is also explored. Moreover, in order to understand the properties of the proposed neural network well, six case studies in which the satellite operates under continuous setpoint change circumstances as well as presenting multiple faults in the reaction wheel are also conducted and explained in details. Based on the problems investigated in this chapter, an overall recommendation about the proposed FDI scheme can be drawn that this new scheme is more than capable of its tasks. More conclusions are summarized in details in the following chapter.

Chapter 5

Conclusions and Further Work

In this thesis, a fault detection and isolation (FDI) scheme is proposed by using an adaptive nonlinear neural network system for the satellite's attitude control system. The following observations and conclusions may be drawn based on the studies that are conducted in previous chapters. Specifically:

1. The proposed FDI scheme building on adaptive recurrent neural networks performs quite satisfactorily. From the setpoint range of 2.5 to 7.5 deg, the network provides very good fault detection and isolation results for the satellite's attitude changes (either by increasing or decreasing its angle). Under fault free operation, the residual error curves remain inside the threshold curves and provide a free false alarm detection results. Corresponding to the wheel becoming faulty, the residual error will cross over the threshold curves except in some exceptional situations as elaborated later.
2. The neural networks are robust to the system noise and disturbances. Increasing the ripple noise will not cause any change to the detection results. For instance, no false alarms are reported and the recovery duration after the fault is removed has not been affected.

3. Once a fault occurs in the wheel the residual error will cross over the threshold curves and once the fault is removed and the wheel operates under a fault free condition, the residual error will move inside the threshold boundaries after some transient time.
4. The recovery process after a fault has been removed from the wheel shows an undershoot and an overshoot transient and during this pair there exist a reflection point. However, if the wheel is still under another and a second fault, only the undershoot behavior is present.
5. Depending on the magnitude of the fault, some large faults will exceed one boundary first and then move inside the boundary and after some ambiguous detection time, the other boundary is then crossed.
6. The magnitude of the faults effect the first time the residual error exceeds the threshold boundary. It generally takes shorter time to cross over the boundary corresponding to a large fault.
7. In practice, if a fault occurs very close to another one (that is not more than 50 minutes apart) it will cause a false negative detection for the first fault implying that the first fault cannot be detected. One of the reasons for this behavior is that the average steady state detection time for faults applied to the wheel is 60 minutes after the application of the faults. Anther reason is that one needs some time to determine the detection of the fault, which implies that the residual error has crossed the threshold boundary clearly for the duration of at least a considerable time (20 minutes in this thesis). Therefore, a sufficient separation

time is needed between the occurrences of two consecutive faults for our proposed FDI scheme.

8. As a mater of fact, two faults applied simultaneously do not cause the deviation of the residual error further away from the threshold however it impacts the satellite position significantly.
9. If an undetected fault is applied after a detectable fault, then it is possible that both faults become detectable by the neural network, however if one fault has occurred after the recovery phase of the other fault, then the first fault has no influence on the current fault detection.
10. The presence of fault on any of the other two axes does not affect the FDI detection results on the third axis. The existence of faults on the two axes do only somewhat elongate the fault recovery transient phase on the third axis, however they do not cause any false alarms (false positive flags) therefore they do not affect the fault isolation results corresponding to the third axis.
11. It was shown that setpoint changes corresponding to the satellite attitudes do not influence the nature of the fault detection and isolation scheme.
12. For a given single fault (not applied simultaneously with another fault) will produce similar detection behavior and recovery characteristics and overshoot magnitudes.
13. For a given type of a fault, the larger the magnitude of the fault is, the larger the recovery process transient overshoot takes.

14. The duration of the presence of a fault impacts the transient phase of the recovery process. For a given single fault which has longer duration, it will take a longer time for the recovery transient phase.
15. Changing the satellite setpoint during the presence of a fault will cause the recovery process of that fault to take a longer time. Therefore, it is recommended that no setpoint changes take place during wheel faults.

Overall, the proposed neural network performs very well as a fault detection and isolation scheme for the satellite attitude control. The advantages of the proposed neural network are listed as following:

- It is capable of providing reliable detection results for the faults which can cause the satellite behave abnormally.
- The proposed neural network possesses strong robustness capability for the uncertainties (noise and disturbances) exerted to the satellite system.

While the disadvantage of the proposed neural network is that the proposed neural network is not capable of distinguishing small fault signals with normal signals.

The following are some suggestions for further research to be conducted beyond this thesis:

1. Neural networks designed and implemented for the other two axes (roll and pitch axes) should be studied also to provide a full three axes fault detection system. The fault detection capabilities in conjunction with the coupling effects need to be investigated further.
2. A comparison with other types of recurrent neural networks should also be studied.

3. The self tuning of the control system after the recovery of each fault is an important attitude requirement. After each recovery of a fault, the entire system has been changed due to the nonlinearity nature of the satellite attitude system. The pre-designed and fixed controller parameters are unable to meet the desired pointing specifications any longer; therefore these parameters need to be adaptively adjusted for the proper execution of the mission.

References

[1] Hughes, P. C. (1986). *Spacecraft attitude dynamics*. New York: John Wiley & Sons, Inc.

[2] Kaplan, M. H. (1976). *Modern spacecraft dynamics and control*. New York: John Wiley & Sons, Inc.

[3] Wertz, J. R., & Larson, W. J. (Eds.). (1999). *Space mission analysis and design*. (3rd ed.). CA: Microcosm Press.

[4] Wertz, J. R. (Ed.). (1980). *Spacecraft attitude determination and control*. Hingham: Kluwer Academics Publishers.

[5] Vadali, S. R., Krishnan, S. & Singh, T. (Feb 22-24, 1993). Attitude control of spacecraft using neural networks. Presented at the *AAS/AIAA spaceflight mechanics meeting*. At Pasadena, California.

[6] Satyadas, A. & KrishnaKumar, K. (1996). EFM-based controller for space station attitude control: Application and analysis. In: F. Herrera & J.L. Verdegay (Eds.), *Genetic Algorithms and Software Computing (Studies in Fuzziness and Soft Computing, Vol. 8)* (pp.152–171). New York: Springer-Verlag.

[7] Lindblad, T., Lindsey, C. S., Minerskjold, M., Eide, A., Linden, T. & Shelton, R. O. (April, 1995). Attitude control systems for spacecrafts using neural networks and fuzzy logic. Presented at the *Forth International Workshop on Software Engineering and artificial Intelligence for High Energy and Nuclear Physics*. At Pisa.

[8] Bialke, B. (Feb 4-8, 1998). High fidelity mathematical modeling of reaction wheel performance. Presented at the *Guidance and Control Conference.* At Breckenridge, CO.

[9] Simani, S., Fantuzzi, C. & Patton, R. (2003). *Model-based fault diagnosis in dynamic systems using identification techniques.* London: Springer.

[10] Gertler, J. (1998). *Fault Detection and Diagnosis in Engineering Systems.* New York: Marcel Dekker.

[11] Isermann, R. (1997, May). Supervison, fault detection and fault diagnosis methods: An introduction. *Control Engineering Practice, 5* (5), 639-652.

[12] Patton, R. J., Frank, P. M., & Clark, R. N. (Eds.). (2000). *Issues of fault diagnosis for dynamic systems.* London: Springer-Verlag.

[13] Basseville, M. & Nikiforov, I. V. (1993). *Detection of abrupt changes: Theory and application.* NJ: Prentice Hall.

[14] Patton, R. J., Uppal, F. J., & Lopez-Toribio, C. J. (June 14-16, 2000). Soft computing approaches to fault diagnosis for dynamic system: A survey. Proceedings on I*FAC Symposium SAFEPROCESS 2000.* pp 298-311.

[15] Lapp, S. A., & Powers, G. A. (April, 1977). Computer-aided synthesis of fault-trees. *IEEE Transaction on Reliability, 26,* 2-13.

[16] Zurada, J. M. (1992). *Introduction to artificial neural systems.* St. Paul: West

[17] Antsaklis, P.J. & Passino, K.M. (Eds.). (1993). *An introduction to intelligent and autonomous control.* Norwell, MA: Kluwer Academic Publishers.

[18] Principe, J. C., Euliano, N. R., & Lefebvre, W. C. (1999). *Neural and adaptive systems: Fundamentals through simulations.* New York: John Wiley & Sons, Inc.

[19] Haykin, S. (1999). *Neural networks: A comprehensive foundation.* (2nd ed). CANADA: Prentice Hall.

[20] Stengel, R. F. (1993, Dec). Toward intelligent flight control. *IEEE Transactions on Systems, Man, and Cybernetics, 23* (6), 1699-1717.

[21] Sorsa, T., Koivo, H. N., & Koivisto, H. (1991, Aug). Neural networks in process fault diagnosis. *IEEE Transactions on Systems, Man and Cybernetics, 21* (4), 815-825.

[22] Maki, Y. & Loparo, K. A. (1997). A neural network approach to fault detection and diagnosis in industry processes. *IEEE Transactions System Technical, 5* (4), 529-541.

[23] Ayoubi, M. & Isermann, R. (1997, August). Neuro-fuzzy system for diagnosis. *Fuzzy Sets and Systems, 89* (3), 289-307.

[24] Passino, K. M. & Antsaklis, P. J. (1988, June). Fault detection and identification in an intelligent restructurable controller. *Journal of Intelligent and Robotic System, 1* (2), 145-161.

[25] Laukonen, E. G., Passino, K. M., Krishnaswami, V., Luh, G. C., & Rizzoni, G. (1995). Fault detection and isolation for an experimental internal combustion engine via fuzzy identification. *IEEE Transactions on Control Systems Technology, 3* (3), 347-355.

[26] Schneider, H. and Frank, P. (1996, May). Observer based supervision and fault detection in robots using nonlinear and fuzzy logic residual evaluation. *IEEE Transactions on Control Systems Technology*, *4* (3), 274-282.

[27] Frank, P. M. & Koppen-Seliger, B. (1997, Jan). Fuzzy logic and neural network application to fault diagnosis. *International Journal of Approximate Reasoning*, *16* (1), 67-88.

[28] Dash, S., & Venkatasubramanian, V. (2000, July). Challenges in the industrial application of fault diagnostic systems. *Computers & Chemical Engineering*, *24* (2), 785-791.

[29] Yoshimura, M., Frank, P. M., & Ding, X. (1997, Dec). Survey of robust residual generation and evaluation methods in observer-based fault detection system. *Journal of Process Control*, *7* (6), 403-424.

[30] Chen, J., Patton, R. J., & Zhang, H. Y. (1996). Design of unknown input observer and robust fault detection filters. *International Journal of Control*, *63* (1), 85-105.

[31] Liu, G. P. & Patton, R. J. (1998). *Eigenstructure assignment for control system design*. England: John Wieley & Sons.

[32] Ray, A. & Luck, R. (1991, Feb). An introduction to sensor signal validation in redundant measurement systems. *IEEE Control Systems Magazine*, *11* (2), 44-49.

[33] Desai, M. & Ray, A. (1984). A fault detection and isolation methodology theory and application. *Proceedings of the 1984 American Control Conference*. pp. 262-270.

[34] Delay, K. C., Gai, E., & Harrison, J. V. (1979). Generalized likelihood test for FDI in redundancy sensor configurations. *Journal of Guidance, Control & Dynamics*, *2* (1), 9-17.

[35] Chow, E., & Willsky, A. (1984, July). Analytical redundancy and the design of robust detection systems. *IEEE Transactions on Automatic Control*, *29* (7), 603-614.

[36] Yazdizadeh, A. (1997, June). *Identification of nonlinear systems using dynamic neural networks*. Montreal: Concordia PH.D. thesis.

[37] Demuth, H. & Beale, M. (2004). *Neural network toolbox user's guide*. (3rd ed). MA: The MathWorks, Inc.

[38] Sarle, W.S. (2002). *Internet FAQ archives.* Retrieved November 06, 2005, from http://www.faqs.org/faqs/ai-faq/neural-nets/part2/section-2.html