Fault Detection and Isolation in Reaction Wheels by Using Neural Network Observers

Zhongqi Li

A Thesis

in

The Department

of

Electrical and Computer Engineering

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

August 2005

Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 0-494-10242-X
Our file Notre référence
ISBN: 0-494-10242-X

ABSTRACT

Fault Detection and Isolation in Reaction Wheels by Using Neural Network Observers

Zhongqi Li

The attitude control of spacecraft has been widely investigated in the past years. In order to meet the requirement to maintain the spacecraft within 0.2^o of a pointing attitude in all three axes, we use a three-axis stabilized system with on-axis reaction wheels for control actuation. Normally, the actuator subsystem contains 4 reaction wheels (3 active and 1 redundant) and three magnetorquers used for momentum dumping (e.g. de-saturating the wheel speed). Each active reaction wheel is aligned with one of the body axis of satellite and acts as an actuator for the control loop of relative axis. In this sense, reaction wheels play an essential role in attitude control system. As we know, reaction wheels are momentum exchange devices which provide reaction torque to a spacecraft and store angular momentum. Reaction wheels' working condition and performance heavily affect the attitude control of a satellite. Any fault in reaction wheels should be detected as early as possible. Mainly, there are three kinds of faults that occur in reaction wheels and that heavily impact the performance of the wheel which deserves more attentions. They are bus voltage fault, current loss (power loss) and temperature fault.

There are many schemes suitable for fault detection and isolation (FDI) such as observer-based methods, parity space and parameter estimation techniques. This thesis presents a neural network observer-based scheme for the actuator fault detection and isolation in the spacecraft attitude control. The features of neural network, such as its intrinsic nonlinearity property, its ability to learn, generalize and parallel processing make it suitable to model a non-linear dynamic system, such as the reaction wheel in our problem. We introduce three Elman recurrent networks and each of them is specific for modeling the dynamics of the wheel on each axis separately and independently. After each network has been trained approximately, it can give accurate estimation of the reaction torque generated by the wheel on each axis. Through some post-processing of the error signal between the actual and estimated output, we can get three residual curves for the FDI purpose. By comparing with a linear observer-based FDI scheme, the neural network observer-based scheme developed in this thesis does show advantages as demonstrated in the simulation results.

ACKNOWLEDGEMENTS

I sincerely wish to express my gratitude to my supervisor Dr. Kash Khorasani for his full support, patient and expert guidance throughout my research work. His suggestions about my work and my life do help me a lot.

I would also like to thank the members of the Electrical and Computer Engineering Department at Concordia University for their help. And the thanks are also for many friends and colleagues for their support and encouragement.

Finally, I would like to take his opportunity to thank my parents, elder sister and elder brother. Their endless help and support really help me go though all these sad and happy days. I am sure they will be proud of me someday.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

General Background and Research Goals

1.1 Introduction

The attitude control of spacecraft has been widely studied since the late 1950s [1-4]. The attitude control subsystem stabilizes the spacecraft and orients it in the desired set point position in a very short time despite external disturbances torque acting on it. Generally, there exist three types of control techniques to meet the requirements [4]. The first one is Gravity-gradient control, which uses the inertial properties of a vehicle to keep it pointed toward the Earth. This relies on the fact that an elongated object in a gravity field tends to align its longitudinal axis through the Earth's center. The torques which cause this alignment decrease with the cube of the orbit radius, and are symmetric around the nadir vector, thus not influence the yaw of a spacecraft around the nadir vector. This tendency is used on simple spacecraft in orbits without yaw orientation requirements, often with deployed booms to achieve the desired inertias. Another is spin control technique. Spin stabilization is a passive control technique in which the entire spacecraft rotates so that its angular momentum vector remains approximately fixed in inertial space. The principle disadvantages of spin stabilization are (1) that the vehicle mass properties must be controlled to ensure the desired spin direction and stability and (2) that the angular momentum vector requires more fuel to reorient than a vehicle with no net angular momentum, reducing the usefulness of this technique for payloads that must be repointed frequently. And the most common attitude control technique today is three-axis control technique. They maneuver and can be stable and accurate, depending on their sensors and actuators. The control torques about the axes of 3-axis systems come from combinations of momentum wheels, reaction wheels, control moment gyros, thrusters or magnetic torquers. Normally, these systems take two forms: one uses momentum bias by placing a momentum wheel along the pitch axis; the other is called zero momentum with a reaction wheel on each axis. The latter one is introduced in this thesis for attitude control. Usually, there are four reaction wheels (3 active and 1 redundant) on the spacecraft to be controlled. Each active reaction wheel is aligned with one of the body axis of spacecraft; it can rotate in either direction and provide reaction torque for the related axis control. And the redundant one will be excited in case of any of the other three wheels fails. Especially, 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.

There are a number of techniques that can deal with the problem of attitude control from classical PID control to adaptive control [1-4]. And recently, many intelligent-based approaches have been introduced to realize the attitude control [5-7]. As discussed above, we design 3 independent control loops by the use of three reaction wheels as the actuators to achieve attitude control on all three axes. In this thesis, we apply the three classical PID controllers separately and the details about the controller design will be explained in Chapter 2.

As stated earlier, we can see reaction wheels play an essential role in the attitude control of spacecraft. Reaction wheels are momentum exchange devices which provide reaction torque to a spacecraft and store angular momentum. A high fidelity mathematical modeling of reaction wheel [8] will be discussed in Chapter 2. Therefore, the performance of reaction wheels heavily affects the attitude control significantly. Reaction wheels operate in the environments with all kinds of disturbances and unpredicted effects. Any fault happening inside the wheels should be detected and isolated as early as possible to avoid serious damage to the attitude control of spacecraft. Normally, there exist three types of faults in the wheel that deserve more attention. The first one is bus voltage fault. The bus voltage should be high enough to avoid elimination of the voltage headroom. And low bus voltage will results in reduced torque capacity and seriously will cause the attitude of spacecraft out of control. The same effect will happen when the motor current loses inside the wheel. That means the reaction wheel loses part of power and as a result cannot supply enough reaction torque to achieve proper set point change of attitude. Finally, we need to consider the temperature fault. As we will see in the Chapter 2, temperature is highly related to the viscous friction, which is the main friction factor of the wheel. The temperature fault will cause the wheel operate in abnormal operating conditions too.

1.2 Literature Reviews in Fault Detection and Isolation

1.2.1 Fault Diagnosis Terminology

Before we go through the problems of fault diagnosis, we need to address some common terminology.

These are based on information obtained from the SAEFPROCESS Technical Committee [9]. A 'fault' is defined as an unexpected change of system function. That is an unpermitted 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. On the other hand, the terminology 'failure' is denoted as complete breakdown of a system component or function. So we are clear that, in order to avoid system breakdown and catastrophes, a fault should be detected and isolated early before it causes a failure.

Accordingly, a fault diagnosis system normally consists of three steps:

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

Obviously, fault detection is the most essential step. It must be done properly before the other two steps can follow. Fault isolation is almost equally importance in practical use in the sense that we can know which parts of system have to be substituted or changed to keep the system from serious problems. While fault identification is not that kind of important if we do not consider system reconfiguration. Hence, we often use the term FDI to indicate fault detection and isolation in the literature.

1.2.2 Fault Diagnosis Methods Classification

Generally, to detect and isolate faults in the system, some sort of redundancy is required. The redundancy is used to make consistency checks between related variables. A traditional approach to fulfill the FDI problem is based on hardware or physical redundancy methods which use multiple sensors, components to measure and control a particular variable. Typically, a voting technique is applied to hardware redundancy system to decide whether a fault has occurred or not. This kind of method is very reliable and widely used in many practical industries. But the main disadvantages of hardware redundancy are the requirements of extra equipments, maintenance cost and the additional space to accommodate the redundant equipments [10]. Sometimes, the space is very limited in the system, such as a spacecraft.

Recently, many researchers have concentrated on the development of analytical or functional redundancy methods. In analytical redundancy system, we use more (not necessarily identical) ways to determine a variable, where one way uses a mathematical process model in analytical form. Based on analytical redundancy, there exist a large amount of methods termed model-based FDI techniques. Figure 1.1 illustrates the concepts of hardware and analytical redundancy.

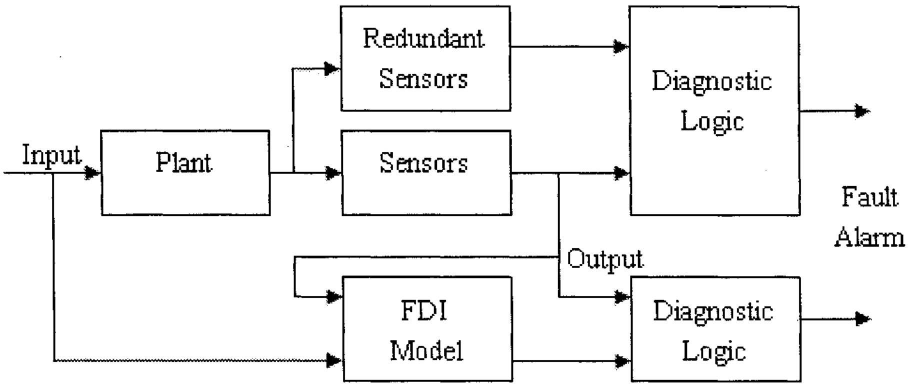


Figure 1.1 Comparison between Hardware and Analytical Redundancy Schemes

Model-based FDI methods can be divided into three main subcategories according to many literatures [11-13]. They are observer-based approaches, parity vector (relation) methods and parameter estimation methods. More details about their properties and difference will be discussed later.

All these model-based FDI methods share some common features in that they usually comprise two main stages as illustrated by Figure 1.2.

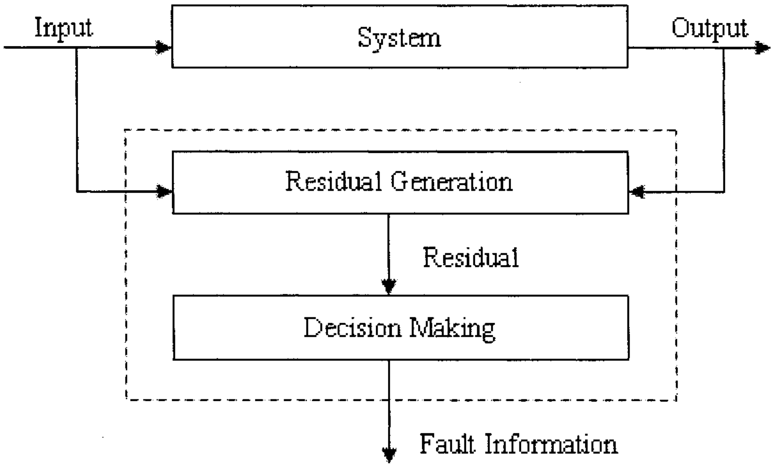


Figure 1.2 Conceptual Structure of Model-Based FDI

This two-stage structure was first suggested by Chow and Willsky [14] and is widely accepted. The first stage is residual generation. Its purpose is to use the available input and output information of the monitored system to generate the indicated fault signal, termed residual. Apparently, in ideal cases, the residual should be normally zero or close to zero in fault free situation, and is distinguishably different from zero when a fault occurs. This stage of residual generation attracts representative fault symptoms from the system and these symptoms are carried by the residual. For better FDI, the residual should contain as more fault information as possible while not sensitive to system disturbances and noises. After the residual is generated, it will be put into the second stage, namely the decision-making stage for fault likelihood examination and finally a determination about fault occurrence is made. On this process, the commonly used decision techniques are simple threshold test on instantaneous values or moving averages of the residuals, or it may consist of methods of statistical decision theory like generalized likelihood ratio test or sequential probability ratio testing. Normally, much attention has been paid on the stage of residual generation, since the decision can be made much more easily based on well generated residuals. And the efforts to improve the performance of the FDI system through well designed decision-making stage are considered as passive approaches.

Besides the hardware redundancy and model-based methods, there still exist a group of methods called intelligent and learning-based methods [15]. These methods make use of the large amount of process history data. In early years, the rule-based methods such as expert system-based FDI system were well known in practice [16]. 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. Recently, neural network and fuzzy logic techniques are being investigated as powerful modeling and decision making tools [17-20]. 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 experience, 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 ability of neural networks has been utilized for nonlinear fault diagnosis problems [18-20]. Meanwhile, expert systems and fuzzy logic have been used in model based fault diagnosis [21-26]. More details about fuzzy logic based FDI methods will be shown in next subsection and the neural network based methods will be discussed more in Chapter 4.

1.2.3 Desirable Attributes of a FDI System

We will list some desirable characteristics of a FDI system. We can use them to benchmark various FDI approaches and also use them as the guideline to design FDI systems [27].

1. Early Detection: Early detection is an important and highly desirable attribute which is more likely 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 and disturbances of the

monitored system. The thresholds should be chosen carefully to avoid the false alarm 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 to 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 them, so a reasonable compromise is desirable.

1.2.4 Observer Based Approaches

The basic idea behind the observer-based approaches 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 to select its gains which leads to a rich variety of FDI schemes [28-31].

In order to obtain the general structure of an observer, the discrete-time, time-invariant linear dynamic system under consideration is modeled in a state space form:

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

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 size. Assuming that all the matrices are known, an observer can be applied to reconstruct the system 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.

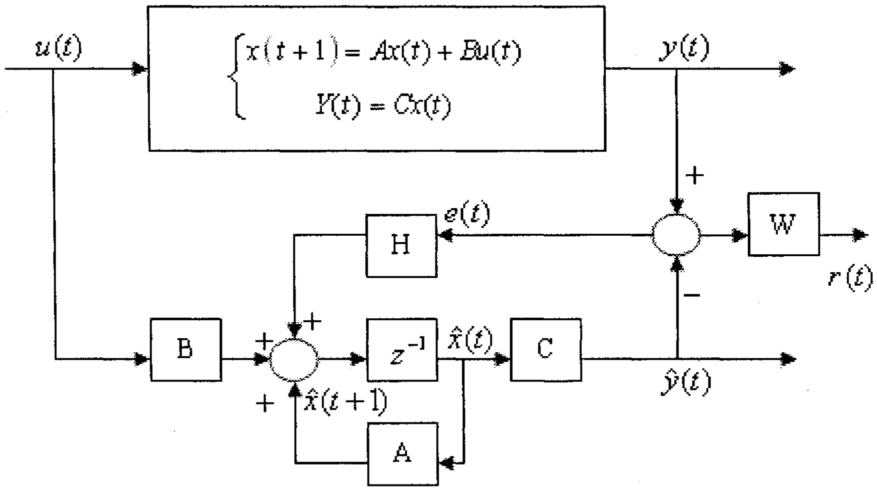


Figure 1.3 Process and State Observer

For the state estimation error $e_x(k)$, it follows from Equation 1.2 that

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

The state error $e_x(k)$ (and the state error $e(k)$) is required to vanish asymptotically, that is

$$\lim_{k\to\infty} e_x(k) = 0 \tag{1.4}$$

Through proper design of observer feedback H , we can make the observer asymptotically stable.

If the process is influenced by disturbance and faults, the system can be described as:

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) + Qv(k) + L_1 f(k) \\ \qquad y(k) = Cx(k) + Rw(k) + L_2 f(k) \end{cases} \tag{1.5}$$

where $v(k)$ and $w(k)$ are the non-measurable disturbance vectors at the input and output respectively; $f(k)$ fault signals at the input and output acting through L_1 and L_2, and they can represent actuator, process and sensor additive faults. The diagram of the system is shown in Figure 1.4.

For the state estimation error, the following equations hold if the disturbances $v(k) = 0$ and $w(k) = 0$,

$$e_x(k+1) = (A - HC)e_x(k) + L_1 f(k) - HL_2 f(k) \tag{1.6}$$

and the output error $e(t)$ becomes

$$e(k) = Ce_x(k) + L_2 f(k) \tag{1.7}$$

The vector $f(k)$ represents additive faults because they influence $e(k)$ and $x(k)$ by a summation. When sudden or permanent faults $f(k)$ occur, the state estimation error will deviate from zero. As well as $e(k)$, $e_x(k)$ shows dynamic behavior which are different from $L_1 f(k)$ and $L_2 f(k)$. Both $e_x(k)$ and $e(k)$ can be taken as residuals. Particularly, the residual $e(k)$ is the basis for different fault detection methods based on output estimation.

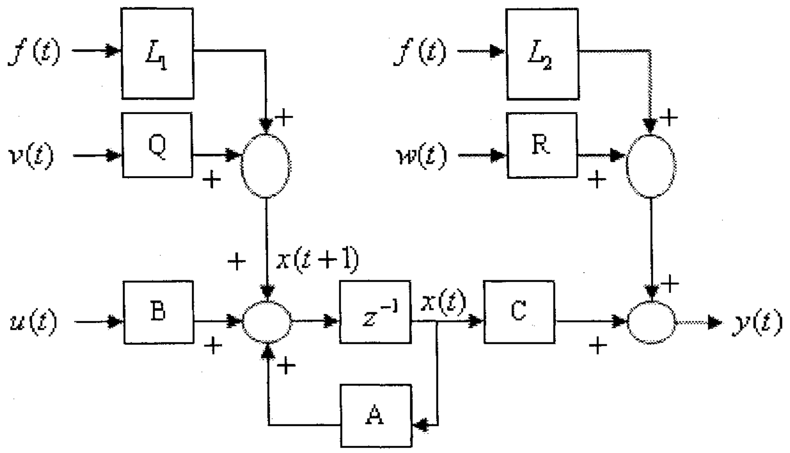


Figure 1.4 MIMO Process with Faults and Disturbances

Something we need to mention here is that the disturbances $v(k)$ and $w(k)$ will affect the sensitivity to faults of FDI system. In this case, the Kalman filter must be used instead of classical observers [32]. And many other techniques have been applied such as unknown input observer (UIO) and disturbance de-coupled residual generator design via eigenstructure assignment.

If faults appear as changes ΔA or ΔB of the parameters, the system becomes

$$\begin{cases} x(k+1) = (A+\Delta A)x(k) + (B+\Delta B)u(k) \\ \qquad\qquad y(k) = Cx(k) \end{cases} \tag{1.8}$$

where the state $e_x(k)$ and the output estimation $e(k)$ errors change to

$$\begin{cases} e_x(k+1) = (A-HC)e_x(k) + \Delta Ax(k) + \Delta Bu(k) \\ \qquad\qquad e(k) = Ce_x(k) \end{cases} \tag{1.9}$$

The changes ΔA and ΔB are then multiplicative faults [11]. In this case, the changes in residuals depend on the parameter changes, as well as input and state variable changes. Hence, the influence of parameter changes on the residuals is not as straightforward as in the case of the additive faults case.

1.2.5 Parity Vector (Relation) Approaches

The parity vector (relation) approaches has been applied to FDI problems since the early development of FDI. A complete survey can be found in [33]. To begin with this problem, let us mention that there are two typical ways to arrange hardware redundancy, one is the use of sensors having identical or similar functions to measure the same variable, the other is the use of dissimilar sensors to measure different variables but with their outputs being relative to each other. And 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.

The measurement equation of a general problem of measurement of n-dimensional vector using m sensors is:

$$y(k) = Cx(k) + f(k) + \xi(k) \tag{1.10}$$

where $x(k) \in \Re^n$ is the state vector, $y(k) \in \Re^m$ is measurement vector, $f(k)$ is the vector of sensor faults, $\xi(k)$ is a noise vector and C is the matrix with proper size. If $m > n$ and $rank(C) = n$, that is 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 where relations between gyroscope readings and accelerometer assemblies provide analytical forms of redundancy [34-36].

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

$$r(k) = Vy(k) \tag{1.11}$$

The residual generation scheme based on direct redundant measurements is shown in Figure 1.5.

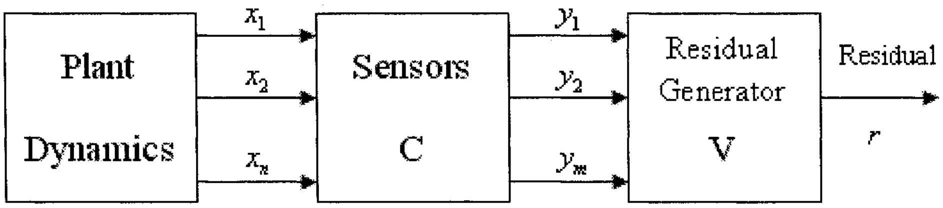


Figure 1.5 Residual Generation via Parallel Redundancy

In order to make $r(k)$ satisfy the usual requirement for a residual (zero-valued for the fault-free case), the matrix V must satisfy the condition:

$$VC = 0 \tag{1.12}$$

When this condition holds true, the residual (parity vector) only contains information on the faults and noise:

$$r(k) = v_1\left[f_1(k) + \xi_1(k)\right] + \cdots + v_m\left[\left[f_m(k) + \xi_m(k)\right]\right] \tag{1.13}$$

where v_i is ith column of V, $f_i(k)$ is ith element of $f(k)$ which denotes the fault in the ith sensor. This equation reveals that the parity vector only contains information due to faults and noise (uncertainty), and is independent of the unmeasured state $x(k)$. We can also see that the parity space (or residual space) is spanned by the columns of V, that is the columns of V form a basis for the space called 'parity space'. Moreover, a fault in the ith sensor implies a growth of the residual $r(k)$ in the direction v_i.

Using the notation of [37], a fault detection decision function is defined as:

$$DFD(k) = r(k)^T r(k) \tag{1.14}$$

If a fault occurs in the sensors, $DFD(k)$ will be greater than a predetermined threshold. Accordingly, the fault isolation decision function is then:

$$DFI_i(k) = v_i^T r(k); \;\; i \in \{1, 2, \cdots, m\} \tag{1.15}$$

For a given $r(k)$, a malfunctioning sensor is identified by computing the m values of $DFI_i(k)$. If $DFI_i(k)$ is the largest one of these values, the sensor corresponding to $DFI_i(k)$ is the one which is most likely to faulty.

In the parity space of view, the columns of V define m distinct fault signature directions. After a fault has been declared, it can be isolated by comparing the orientation of the parity vector to each of these signature directions. Indeed, the fault isolation function $DFI_i(k)$ is a measure of the correlation of the residual vector with fault signature directions. In order to isolate faults reliably, the generalized angles between fault signature directions should be as large as possible. Thus, optimal fault isolation performance will be achieved if:

$$\begin{cases} \min\{v_i^T v_j\}; i \neq j \in \{1,2,\cdots,m\} \\ \max\{v_i^T v_i\}; i \in \{1,2,\cdots,m\} \end{cases} \tag{1.16}$$

The traditional sub-optimal solution of the matrix V is to make [38]:

$$VV^T = I_{m-n} \tag{1.17}$$

If we combined Equation 1.17 with 1.12, we can get:

$$VV^T = I_m - C(C^T C)^{-1} C^T \tag{1.18}$$

The condition for the existence of a solution V for Equation 1.12 is that $rank(C) = n < m$. This implies that the outputs of the sensors are selected by a static relation. There are some algorithms about getting a complete solution of V. People can refer to Potter's algorithm [34] to get more details if interested.

But if $rank(C) = n > m$, the direction redundancy relation does not exist. In this situation, we may construct redundancy relations by collecting sensor outputs over a time interval (data window like $\{y(k-s), y(k-s+1), \cdots, y(k)\}$). This is known as 'temporal redundancy' or 'serial redundancy'. A number of literatures are concerned about this technique [39-43]. Figure 1.6 is one structure diagram about the parity relation approach for residual generation of dynamic systems suggested in [39]. Obviously, the residual signal can be defined as:

$$r(k) = V\left[Y(k) - HU(k)\right] \tag{1.19}$$

The parity relation approach can be used to design structured residual set for fault isolation. And the design for isolating sensor faults is very straightforward by using this parity vector-based approach as we see above. But for the actuator faults isolation, the structure set is more difficult to design.

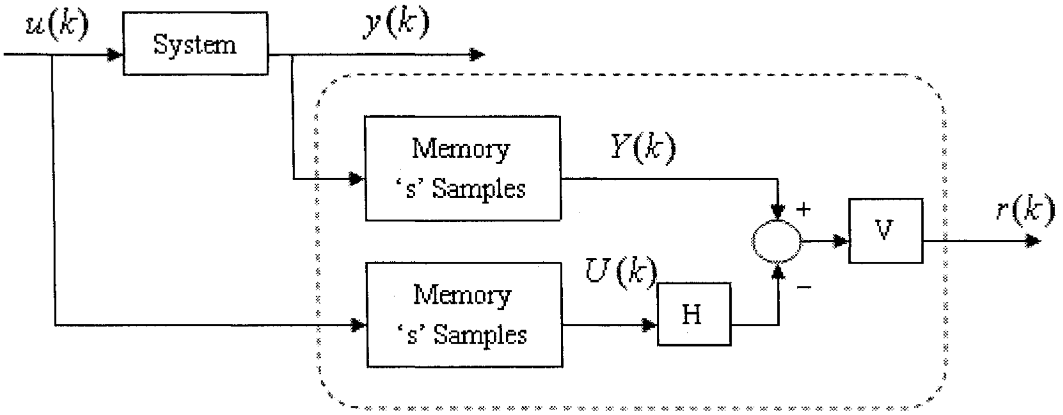


Figure 1.6 Residual Generation via Temporal Redundancy

1.2.6 Parameter Estimation Approaches

Model-based FDI can also be achieved by the use of system identification techniques [44-48]. 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 as a fault. This approach normally uses the input-output mathematical model of a system in the following form:

$$y(k) = f(P, u(k)) \tag{1.20}$$

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 formats.

The basic procedure for carrying out FDI using parameter estimation is:

- Establish the process model using physical relations;
- Determine the relationship between coefficients and process physical parameters;
- Estimate the normal model coefficients;
- Calculate the normal process physical parameters;
- Determine the parameter changes which occur for the various fault cases.

By carrying out the last step for known fault, a database of faults and their symptoms can be built up. During the system operation, the coefficients of the system model are periodically identified from the measurable inputs and outputs, and compared with the normal and faulty model parameters.

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

$$\begin{cases} r(k) = \hat{P}_{k-1} - P_0 \\ r(k) = y(k) - f(\hat{P}_{k-1}, u(k)) \end{cases} \tag{1.21}$$

where P_0 is the normal model coefficient.

It is not easy to achieve fault isolation using the parameter estimation method. This is 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, [49] proposed an influence matrix approach to overcome the isolation difficulty. The idea is to identify the influence of each physical parameter on the residual.

1.2.7 Fuzzy Logic Based Approaches

Fuzzy logic, as one kind of intelligent-based method, has received more and more attention in FDI problems [24-26]. Obviously, the decision making stage of fault detection is a logic decision process that transforms quantitative knowledge (residual signals) into qualitative statements (normal or faulty). Due to the fact that although the residual contains the information of faults, they are contaminated by the noises and disturbances, the residual will be non-zero even in fault-free cases. And it seems very natural to deal with this logical decision making problem with the aid of fuzzy logic since fuzzy logic shows advantage to handle such cases in an uncertain and complicated situation 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.

Fuzzy logic endows machine intelligence with the ability to make decisions based on shades of grey, instead of black-and-white information. Essentially, fuzzy processing can be divided into following steps. Firstly, the residuals are compared with membership functions (or degree-of-belief curves) which are often assumed to be of triangular shape. Secondly, the lower of the two antecedent outputs is selected. Then the output of all rules is combined. Finally, the center of gravity (or another averaging method) is used to defuzzify the output and lead to the possibility of definite decision making. The use of fuzzy logic has been proved to be applicable in the providing reliable and sufficient FDI in real industrial systems.

Moreover, through combination of fuzzy logic and traditional model-based methods, we can get so-called fuzzy observer as shown in Figure 1.7, which is helpful in FDI in non-linear dynamic systems. The main idea is to use the T-S fuzzy model which was developed in [50]. Using this model, a non-linear dynamic system is described by a number of locally-linearized observers. Under the fuzzy logic observer scheme, a number of local linear observers are designed and the state estimate is given by a fuzzy fusion of local observer outputs. The diagnostic signal – a residual is the difference between the estimated and real system outputs. More details about the design procedures and examples can be referred to [51].

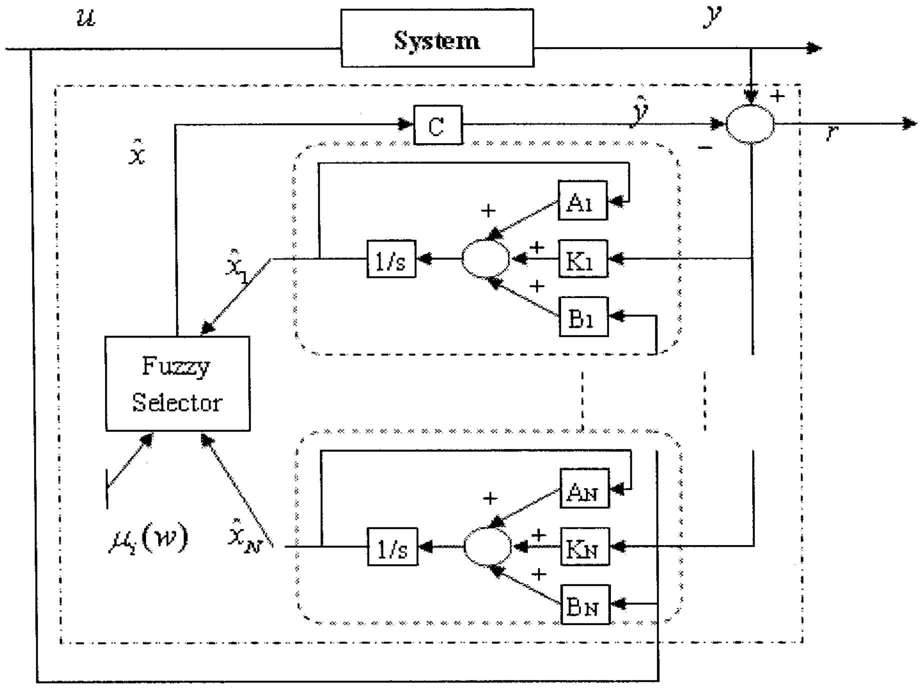


Figure 1.7 Fuzzy Observer

1.3 Research Motivation

When the bus voltage fault, current loss fault or temperature fault happens in the reaction wheel, we can observe change in the reaction torque generated by the wheel with fault and also change in the output angle of the related axis compared with fault free cases. This thesis is concentrated on the understanding of the behavior of reaction torques as a consequence of faults. The objective of this work is to develop some ways to detect and isolate the faults. By detection, we mean an approach to find out whether the working conditions of wheels are in normal range or not. And fault isolation means we need to decide the fault happens in the wheel on which axis of the spacecraft. There are many kinds of fault detection and isolation techniques which have been widely applied in many dynamic systems [9,52]. Artificial neural networks have the properties of inherent high nonlinearity, robustness to disturbances and noises and parallel operation. It has been demonstrated to successfully be applied in modeling of dynamic systems and fault diagnosis as well [53-55]. Hence, our motivation is to employ artificial neural network techniques as a reliable tool to detect and isolate faults in the reaction wheels and it does validate its usefulness and advantages over traditional generalized Luenberger linear observer-based techniques, which can be shown in the following chapters.

1.4 Research Objectives and Contributions of the Thesis

In view of the above discussion, the objectives of this thesis are to develop schemes based on neural networks for fault detection and isolation in the reaction wheels. Specifically, the goal is to decide whether a bus voltage fault, current loss fault or temperature fault has occurred in the reaction wheels and localize which wheel is faulty as well. In order to achieve these objectives, three neural networks are introduced to model the dynamics of the reaction wheels on all three axes separately and independently. Due to the dynamic property of the wheel, the neural network architecture we apply in this thesis is the Elman recurrent network [56] with backpropagation algorithm. The efficiency of the neural network observer-based FDI scheme is carefully investigated, and a comparative study is conducted with the performance of the generalized Luenberger linear observer-based scheme. The simulation results will prove the advantages of the neural network-based method we develop here.

1.5 Research Methodology

With the help of the neural networks employed on the reaction wheel of each axis, we can observe the estimated reaction torques from each wheel. Through these estimated signals, we are able to identify the existence of faults in the system and which wheel is faulty as well. Figure 1.8 shows the stages conducted in this thesis.

As shown in Figure 1.8, the algorithm developed in this thesis consists of three stages:

1. Residual Signals Generation: We design three Elman networks to model the dynamics of the reaction wheels on three axes separately. The inputs to each network are the torque command voltage signal (TCV) and the one step delay of actual reaction torque signal (RT) of the corresponding reaction wheel or the estimated reaction torque from the output of the network.

And the output of the network is the estimated reaction torque. Then after some post-processing of the differences between the actual and the estimated reaction torque signal, we can get residual for FDI purpose. Three residual signals are generated totally in this stage.

2. Threshold Testing and Fault Detection: The residual signals generated in the first stage will go through the corresponding threshold curves for testing. The threshold curve is generated through large sums of residual curves collected in the fault free cases. If the residual curve exceeds the threshold curve for a considerable time, then we consider there is a fault that has happened in the concerned reaction wheel.

3. Fault Isolation: With consideration on the threshold testing results from all these three axes, we can localize which wheel is faulty.

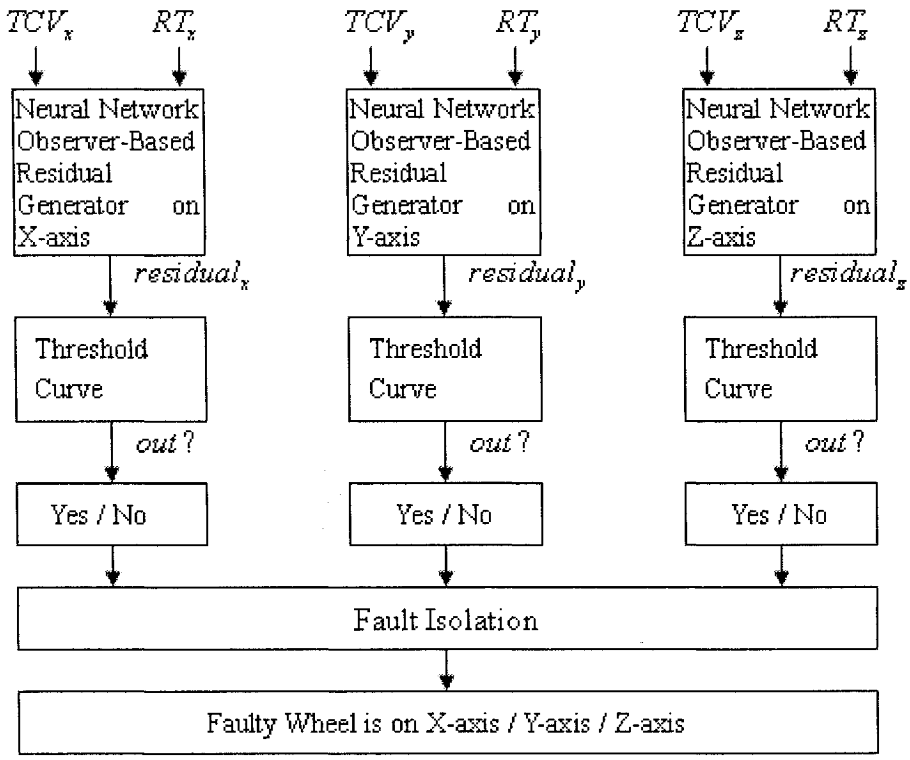

TCV_x
RT_x
TCV_y
RT_y
TCV_z
RT_z
Neural Network Observer-Based Residual Generator on X-axis
Neural Network Observer-Based Residual Generator on Y-axis
Neural Network Observer-Based Residual Generator on Z-axis
$residual_x$
$residual_y$
$residual_z$
Threshold Curve
Threshold Curve
Threshold Curve
out?
out?
out?
Yes / No
Yes / No
Yes / No
Fault Isolation
Faulty Wheel is on X-axis / Y-axis / Z-axis

Figure 1.8 The FDI Methodology by Using Neural Network

In order to perform the above tasks in this thesis, simulation model of the whole attitude control system and the reaction wheel are constructed with the help of software MATLAB (Version 7.01) and SIMULINK. And all the data accumulation, signal processing, neural network application and simulation result assessment are conducted in MATLAB and its associate toolboxes.

1.6 Outline of the Thesis

In Chapter 2, an outline of the attitude control system will be shown. And we will get to know the details about dynamic characteristics of the reaction wheel and their modeling in MATLAB.

Chapter 3 presents a simple generalized Luenberger linear observer-based scheme used for fault detection and isolation in the reaction wheels. The simulation results will server as a comparison of data with the neural network observer-based scheme.

In chapter 4, after a brief introduction about the neural networks, a neural network observer-based FDI scheme will be developed step by step. The suitable choice of neural network parameters will be investigated as well.

Then a comparative study between the neural network observer-based scheme and the linear observer-based scheme will be conducted in Chapter 5. And the comparative results will prove the efficiency and advantages of neural network observer-based scheme developed in this thesis.

In chapter 6, a brief summary of this thesis will be stated and some recommendations about the future work are discussed.

Chapter 2

Introduction to the Attitude Control System and Wheel Dynamics

2.1 Mission Specifications

In order to investigate the attitude dynamics and control of the spacecraft, a hypothetical satellite called MakSat is considered in this thesis. The MakSat will be launched into a 700 km circular Low Earth Orbit (LEO), sun-synchronous (98.2^o) orbit. With a velocity of approximately 7.5 km/s, the orbit has a period of 98.8 minutes. Orbit selection is driven by science requirements, orbit lifetime, ground station coverage, and radiation concerns.

In MakSat, we introduce three axis control techniques for attitude control. Specifically, we tend to achieve pointing accuracy for each axis through the reaction wheel aligned with each axis separately. The accuracy requirement of pointing attitude is within 0.2^o in all three axes in our thesis. To achieve this, three separate PID control loops will be used to control the three reaction wheels for control of each spacecraft axis. A detailed discussion about one single control loop will be shown and explained in the following sub-section.

2.2 Single Axis Attitude Control Using PID

The standard block diagram of a single axis attitude control loop is shown in Figure 2.1. As seen from the diagram, this control loop contains four main blocks: sensor block, controller block, actuator block and body dynamics block. Beside these, there are still some noises added on the sensors and some external disturbances imposing on the spacecraft.

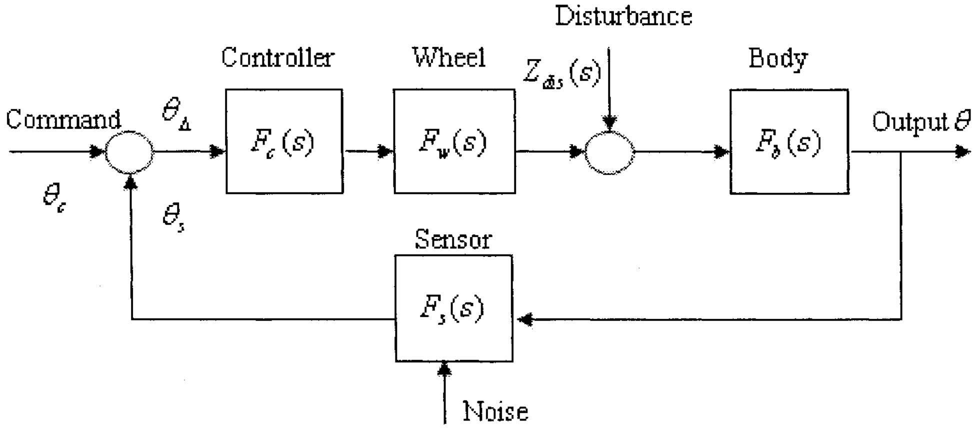


Figure 2.1 Single-Axis Attitude Control Block Diagram

Let us get the transfer function of the system first. Note that

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

where θ is the controlled attitude angle, and

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

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

Combining these three equations and set the disturbance $Z(s)$ to zero, we can get the transfer function as:

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

We can get the disturbance transfer function by assuming the control command $\theta_c(s)$ is zero,

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

The different function blocks of this control loop will be discussed in the following sub-sections.

2.2.1 Sensors [4]

Table 2.1 shows a summary of typical sensors used in airospace, as well as their performances and physical characteristics.

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, 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: 0.1 deg 1 deg (LEO) <0.1 deg to 0.25 deg	1 to 4 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

Table 2.1 Typical ADCS Sensors [4]

Sun sensors are visible-light detectors which measure one or two angles between their mounting base and incident sunlight. They are accurate and reliable, but require clear fields of view. Usually, sun sensors are mounted near the ends of the vehicle to get an unobstructed field of view. They can be used as part of the normal attitude determination system, part of the initial acquisition or failure recovery system, or part of an independent solar array orientation system.

Star sensors represents the most common sensor for high-accuracy missions. They can be scanners or trackers. Stars pass through multiple slits in a scanner's field of view, so we can get the spacecraft's attitude after several star crossings. Scanners are usually used on spinning spacecraft. While trackers are used on 3-axis attitude stabilized spacecraft to track one or more stars to obtain two or three axes attitude information. The most sophisticated units not only track the stars as bright spots, but identify which star pattern they view, and output the sensor's orientation compared to an inertial reference. For highest accuracy missions, we use a combination of star trackers and gyros. We use the gyros for initial stabilization and during periods of sun or moon interference in the trackers, while we use the star trackers 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 40 km above the surface in the sensed band). Horizon sensors provide Earth-relative information directly for Earth-pointing spacecraft, which may simplify onboard processing. Horizon crossing indicators (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 (nadir) vector. Scanning horizon sensors use a rotating mirror or lens to replace (or augment) the spinning spacecraft body. They are usually used in pairs for improved performance and redundancy.

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 helps us establish the spacecraft's attitude. But their accuracy is not as good as that of star or horizon references. They are often combined with Sun or horizon sensors to improve accuracy.

GPS receivers are commonly known as high-accuracy navigation devices. GPS receivers have been used for attitude determination by employing the differential signals from antennas on a spacecraft. Such sensors are low cost and weight for LEO missions but low accuracy.

Gyroscopes are inertial sensors which measure the speed or angle of rotation from an initial reference, but without any knowledge of an external, absolute reference. 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). Since Gyroscopes cannot provide any information of an external, absolute reference, they are often used for precision attitude sensing in spacecraft when combined with some external references such as Sun sensors or star sensors.

For the MakSat in our thesis, we use the Earth/Horizon sensor to get the pitch and roll angle. A realistic sensor is more likely to be represented by a $P-T_1$ block, which represents a delay between the input and output signals,

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

In order to simplify the problem in our thesis, we use an ideal sensor represented by a simple P block with the transfer function expressed below:

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

2.2.2 Actuators

The actuators here we use for attitude control are reaction wheel assemble consisting of 3 active reaction wheels on the three axes of the spacecraft and one more redundant wheel. Reaction wheels are momentum exchange devices which provide reaction torque to a spacecraft and store angular momentum [8]. Mathematical modeling of reaction wheel can be derived merely from an application of Newton's laws, but additional terms are included herein to assess performance beyond the normal speed range, and as a function of temperature and bus voltage. What is more, the disturbance and noise terms cannot be ignored in reality. A reaction wheel consists of a rotating flywheel, typically suspended on ball bearings, and driven by an inertial brushless DC motor.

A detailed time domain block diagram of a typical reaction wheel is shown in Figure 2.2. This diagram provides the fundamental relationships for a high fidelity mathematical model of reaction wheel system. This detailed reaction wheel model is used in our thesis to substitute the real wheel. There are five main sub-blocks in the diagram: motor torque control, speed limiter, EMF torque limiting, motor disturbances and bearing friction and disturbances. Further information about the sub function blocks will be explained in the following sub-sections.

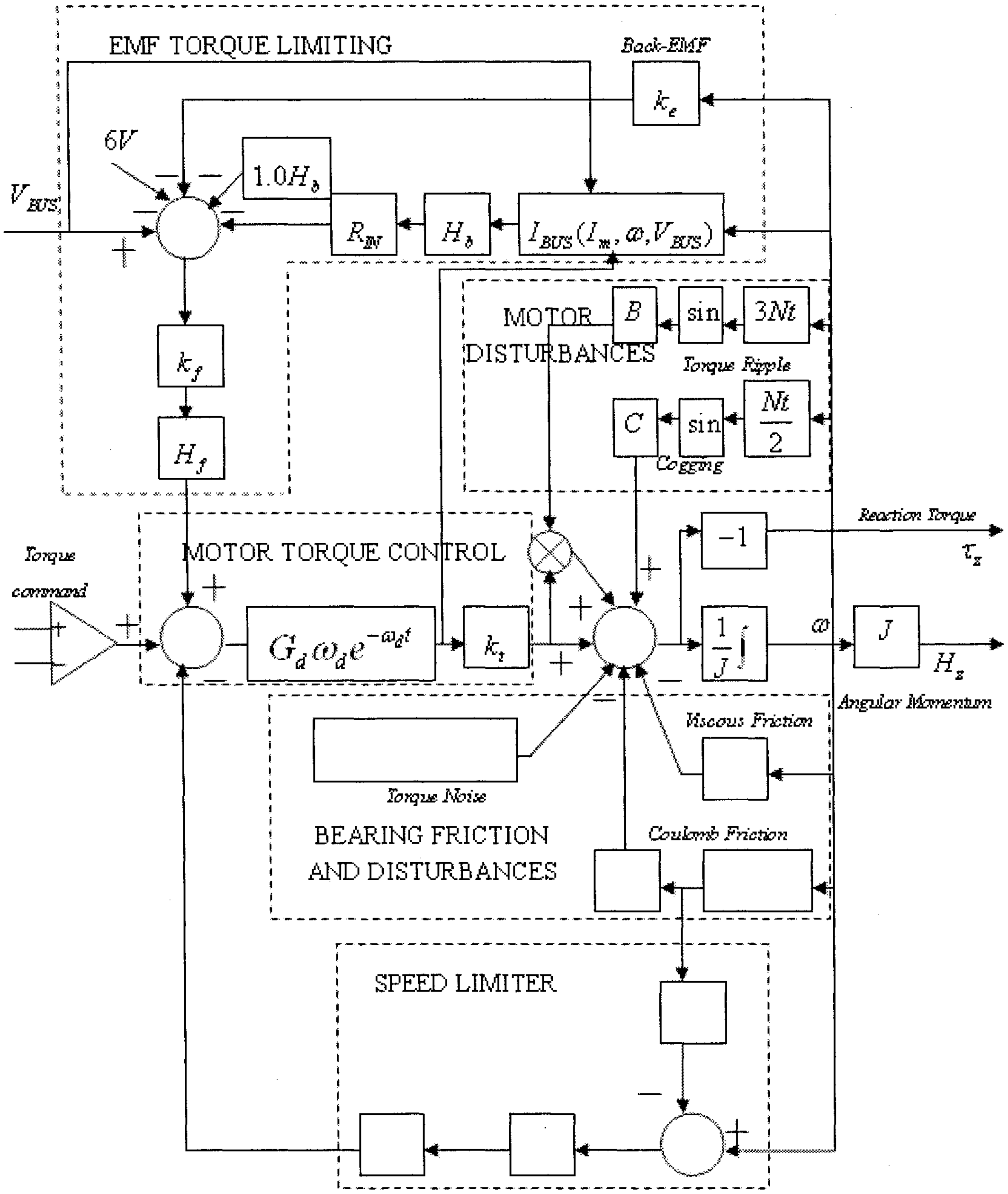


Figure 2.2 Detailed Reaction Torque Block Diagram [8]

And the reaction wheel applied in our thesis is the ITHACO's standard Type A reaction wheel. Its typical constant values used in this diagram are defined in Table 2.2 [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

Table 2.2 Typical Constant Values of Type A Reaction Wheels [8]

Motor Torque Control Block

The motor driver is essentially controlled by a torque command voltage with a gain G_d. And the result is a motor current I_m directly proportional to the voltage. Then through a torque constant k_t, it generates a motor torque τ_m. In our thesis, the torque command voltage is constrained by $[-5V, 5V]$.

Speed Limiter Block

A speed limiter circuit is employed to prevent the flywheel from reaching unsafe speeds, which is defined as a threshold ω_s. It uses an analog tachometer circuit to sense wheel speed. Once the speed ω is beyond the ω_s, the circuit provides it as a high-gain negative feedback k_s to limit the speed. In the diagram, we use a heavyside function H_s to enable this negative feedback.

$$\begin{cases} H_s = 0 \, for \, |\omega| < \omega_s; \\ H_s = 1 \, for \, |\omega| \geq \omega_s; \end{cases} \tag{2.8}$$

EMF Torque Limiting Block

For low bus voltage conditions, the motor torque may be limited at high flywheel speed due to the increasing back-EMF k_e of the motor. This eventually eliminates the voltage headroom, and reduces the torque capacity once the back-EMF increases to the point where the pulse-width-modulation of the motor driver is saturated. From a disturbance point of view, the available motor torque will be coupled directly to the bus voltage in this condition, and any fluctuations in bus voltage will be felt as torque disturbance.

The back-EMF limiting is mildly coupled to power consumption by voltage drops in the input filter due to the bus current level. This voltage drop is the product of the bus current I_{BUS}, and the filter input resistance R_{IN}. In order to get the expression of I_{BUS}, we use an approximate power consumption model:

$$P_{INPUT} = \frac{V_{BUS}}{V_{BUS}-1}\left[\frac{\tau_m^{\ 2}}{k_t^{\ 2}}R_B + \frac{0.04|\tau_m|V_{BUS}}{K_t} + P_q + \omega\tau_m\frac{k_e}{k_t}\right] \tag{2.9}$$

where P_{INPUT} is the power consumption, and when dividing V_{BUS} we can the bus current I_{BUS}.

$$I_{BUS} = \frac{1}{V_{BUS}-1}\left[\frac{\tau_m^{\ 2}}{k_t^{\ 2}}R_B + \frac{0.04|\tau_m|V_{BUS}}{K_t} + P_q + \omega\tau_m\frac{k_e}{k_t}\right] \tag{2.10}$$

Combining the relationship that

$$k_t = \frac{\tau_m}{I_m} \tag{2.11}$$

the above two equations yield:

$$I_{BUS} = \frac{1}{V_{BUS}-1}\left[I_m^{\ 2}R_B + 0.04|I_m|V_{BUS} + P_q + \omega I_m k_e\right] \tag{2.12}$$

In order to eliminate the voltage drop when the power is not being drawn form the bus, as during a deceleration when energy is being removed from the flywheel, we apply a heavyside function H_b to achieve this function. In addition, a reverse polarity protection diode drop of $1V$ is also dependent on H_b.

$$\begin{cases} H_b(I) = 1 for I > 0; \\ H_b(I) = 0 for I \le 0; \end{cases} \tag{2.13}$$

Another heavyside function applied in this sub-block is defined as:

$$\begin{cases} H_f(V) = 0 for V > 0; \\ H_f(V) = 1 for V \le 0; \end{cases} \tag{2.14}$$

Motor Disturbances Block

Type A reaction wheel model has employed 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 due to 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 14.3% peak-to-peak of the commanded motor torque, or about 7% rms. For some cases with misalignment, this value can be as high as 22% peak-to-peak, or 11% rms. Although the torque ripple wave shape is a truncated rectified sine wave, the block diagram approximates it with a pure sine wave for simplicity. In addition, the amount of torque ripple is highly dependent on the torque ripple frequency which is essentially the commutation rate. And the commutation rate is proportion to the number of poles, the number of phases and the rotation rate.

Cogging is a disturbance torque which is always present in conventional brushless DC motor. It is due to the change in reluctance of iron stator as the magnets in a conventional brushless DC motor are rotated. In some cases, this can lead to undesirable disturbances when operating near zero speed. But the ITHACO's ironless armature motor design, which is applied in reaction wheels of our thesis, has completely eliminated the cogging disturbances.

Bearing Friction and Disturbances Block

Normally, the friction in a reaction wheel can be mathematically broken down into viscous friction and coulomb friction.

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

$$\tau_v = \left(0.049 - \frac{0.0002}{{}^oC}\left(T + 30^oC\right) \right) \times 10^{-2} \frac{N-m}{rad/\sec} \tag{2.15}$$

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 flying 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, and our thesis as well, this coulomb friction can be assumed as twice of the rolling friction, neglecting the breakaway torque difference.

Torque noise is the very low frequency torque variation from the bearings, due to the lubricant dynamics. Light oils will result in low torque noise, but potentially more wear in the bearing and thus increase the drag torque. Grease can be used to reduce drag torque, but slight amounts of grease are a significant source of torque noise. Torque noise can be specified as a deviation from the ideal location of motor at any constant speed. As in the block diagram, this torque noise can be approximately modeled as:

$$\tau_a = J\theta_a\omega_a^2 \sin\omega_a t \qquad (2.16)$$

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

What we need to mention here is, for PID controller design, we only use the nearly ideal reaction wheel model for simplicity. Its block diagram is shown below in Figure 2.3.

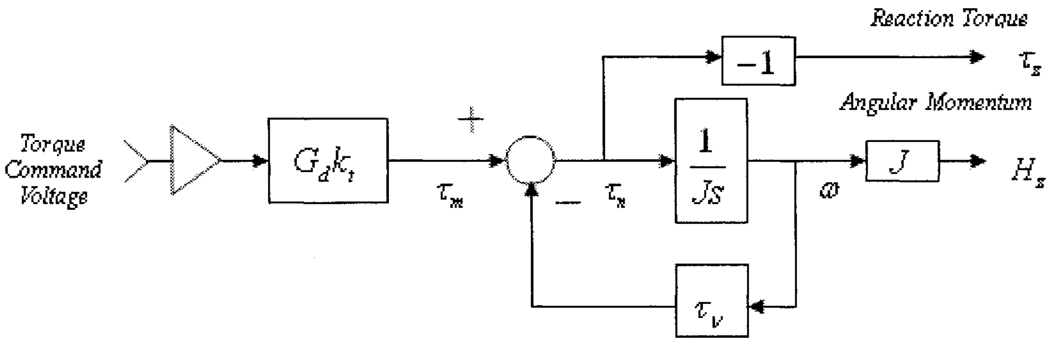


Figure 2.3 Nearly Ideal Reaction Wheel Model Block Diagram

The angular momentum stored in the flywheel is the product of flywheel inertial and the wheel speed, that is,

$$H_z = J\omega \qquad (2.17)$$

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

$$\tau_z = -\tau_n \tag{2.18}$$

And the net torque can be derived from the rate of change angular momentum,

$$\tau_n = \frac{\partial H_z}{\partial t} \tag{2.19}$$

Meanwhile, we can also use the motor torque minus frictional loss (we only use viscous friction for simplicity).

$$\tau_n = \tau_m - \tau_v \omega \tag{2.20}$$

$$\tau_m = T_c G_d k_t \tag{2.21}$$

where T_c is the torque command voltage.

From the above five equations and with the use of Laplace transform, we can get the wheel transfer function from the torque command voltage to the reaction torque,

$$F_w = \frac{-G_d k_t J s}{J s + \tau_v} \tag{2.22}$$

2.2.3 Body Dynamics

From Newton's second law, if the body reference system has angular velocity $\vec{\omega}$ as observed from the inertial reference frame, there exists a relationship between the torque and angular momentum:

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

Recalling that

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

Combining these two equations, we can get the Euler's moment equation:

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

Also,

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

Normally, we consider the case that the spacecraft body frame aligned with the principle axes, where the products of inertials are zero

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

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

Equation 2.27 is applied for model construction in our thesis. But for the PID controller design, we assume a rigid and decoupled system for simplicity, which means the coupling effects are omitted,

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

Then we can get the transfer function of the body dynamics for one single axis,

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

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 main of them are consider here: gravitation torque, solar pressure torque, magnetic torque and aerodynamic torque. They are explained more in the following.

The gravitational torque arises because 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. For instantaneous gravity gradient torque calculation, one can refer to [1]. For the problem here, we assume the maximum gravity gradient torque during the whole period is:

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

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. For the spacecraft used in our thesis, the worst case of solar pressure torque can be assumed as [4]:

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

Because the inaccuracy of the spacecraft's magnetic dipole vector and the current loops within the spacecraft, it is tough to determine the Earth's magnetic torque accurately all the time. Normally, we employ a dipole model to estimate the value of this torque. Let us assume the maximum value of this torque in our problem is [4]:

$$DIS_{mf} = 4.5\times10^{-5}N-m \tag{2.32}$$

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 preliminary design, we use a rough estimation maximum value for this aerodynamics disturbance torque as [4] :

$$DIS_{ad} = 3.4\times10^{-6}N-m \tag{2.33}$$

For simplicity, we assume the maximum external disturbance torque is the summation of these four maximum torques:

$$DIS = DIS_{gg} + DIS_{sp} + DIS_{mf} + DIS_{ad} = 5.68\times10^{-5}N-m \tag{2.34}$$

For the construction of ASC model in our thesis, we assume the external disturbance torque is a normally distributed random signal with zero mean and variance as $DIS^2 = \left(5.68\times10^{-5}\right)^2$.But for the PID controller design, we assume it 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 this external disturbance torque is:

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

2.2.5 PID Controller

In order to achieve the Earth pointing accuracy within 0.2^o, a PID controller should be applied. The PID controller can be represented as

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

As discussed above, we have the transfer function of the different blocks in the control loop in the Figure 2.1 and rewrite them here as:

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

So the transfer function of open loop system is:

$$G_{OL} = \frac{-G_d k_t J\left(s+z_{c1}\right)\left(s+z_{c2}\right)s}{\left(Js+\tau_v\right)Is^3} \tag{2.38}$$

Suppose the design specifications to be such like that the desired maximum percent overshoot is less than 20% and the 5% settling time is 30 seconds, based on the PID controller design algorithm explained in [57], we can get the values of z_{c1} and z_{c2} in the formula of PID controller.

Obviously, from Equation 2.38, without external disturbance torques, our attitude control system is a type 2 system with zero steady state error for step function input. Hence the steady state error of our system is due to the external disturbance torques. From disturbance transfer function of Equation 2.5, which is shown here again for convenience,

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

using the Final Value Theorem, the steady state error of our system becomes

$$Error_{ss} = \lim_{s \to 0} sD(s)Z_{dis}(s) = \frac{sZ_{dis}(s)F_b(s)}{1 + F_s(s)F_c(s)F_w(s)F_b(s)} \qquad (2.39)$$

Assume the maximum disturbance torque is applied on the spacecraft as discussed above in Equation 2.35, so we can get

$$Error_{ss} = -\frac{\tau_v\left(5.68 \times 10^{-5}\right)}{G_d k_t J z_{c1} z_{c2} k_d} \qquad (2.40)$$

In order to meet the pointing accuracy requirement of 0.2^o, that is

$$Error_{ss} < 0.2^o = \frac{0.2\pi}{180} \qquad (2.41)$$

We can get

$$k_d > -\frac{\tau_v\left(5.68 \times 10^{-5}\right)}{G_d k_t J z_{c1} z_{c2} \dfrac{0.2\pi}{180}} \qquad (2.42)$$

Based on this equation, we can decide the value of k_d in the PID controller formula. Actually, the PID design algorithm is a trial and error procedure to achieve lower output overshoot, shorter settling time and small pointing error within the accuracy requirement.

2.3 Three Axes Attitude Control System

Three PID controllers are designed separately and independently using the design method specified in section 2.2. What we want to emphasize is that, the construction of the three axes attitude control system is based on equation 2.27, that is

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

which means we need to include the coupled effects from other axes. Figure 2.4 shows the simulation block diagram of this three axes attitude control system. It is clear that the net torque applied on one axis is effected by the angular speed of other two axes.

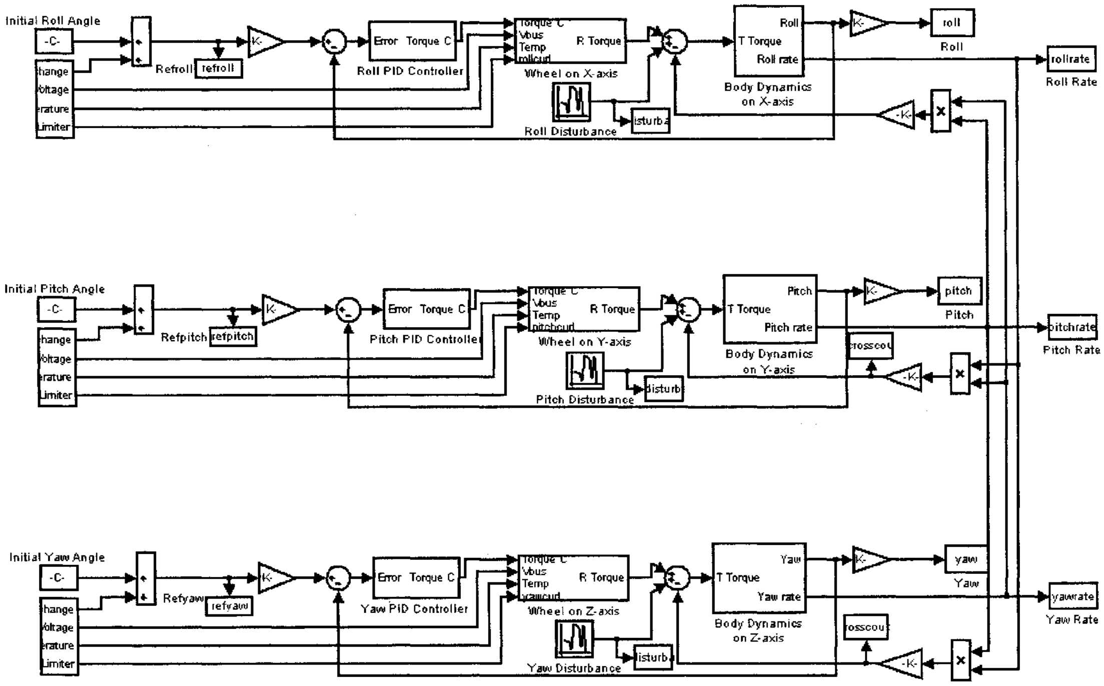


Figure 2.4 Three Axes ACS Block Diagram

2.4 Parameter Values in Normal Operation

Besides the parameter values of the reaction wheel which have been defined in Table 2.2, there are still some parameters and normal operating condition of our attitude control system. All these values are the same but independent for all three axes. They have been shown below in Table 2.3 before we start the simulations.

Nomenclature	Units	Normal Value or Range
Bus Voltage	V	8
Temperature	^0C	23
Initial Body Attitude for One Axis	deg	-180 to 180
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	-5 to 5
S/C Inertial Matrix	Kgm^2	$\begin{pmatrix} 17 & 0 & 0 \\ 0 & 15 & 0 \\ 0 & 0 & 22 \end{pmatrix}$

Table 2.3 Parameter Values in Normal Operating Condition

2.5 Simulation Results

Below is one typical example with the following operating conditions:

Initial body attitude for X axis: $94.3549\deg$

Initial body rate for X axis: $7.8260\times10^{-5} rad/\sec$

Initial wheel speed on X axis: $24.8598 rad/\sec$

Setpoint change on X axis: $4.5013\deg$

Initial body attitude for Y axis: $115.7066\deg$

Initial body rate for Y axis: $-9.6299\times10^{-5} rad/\sec$

Initial wheel speed on Y axis: $24.5647 rad/\sec$

Setpoint change on Y axis: $-2.6886\deg$

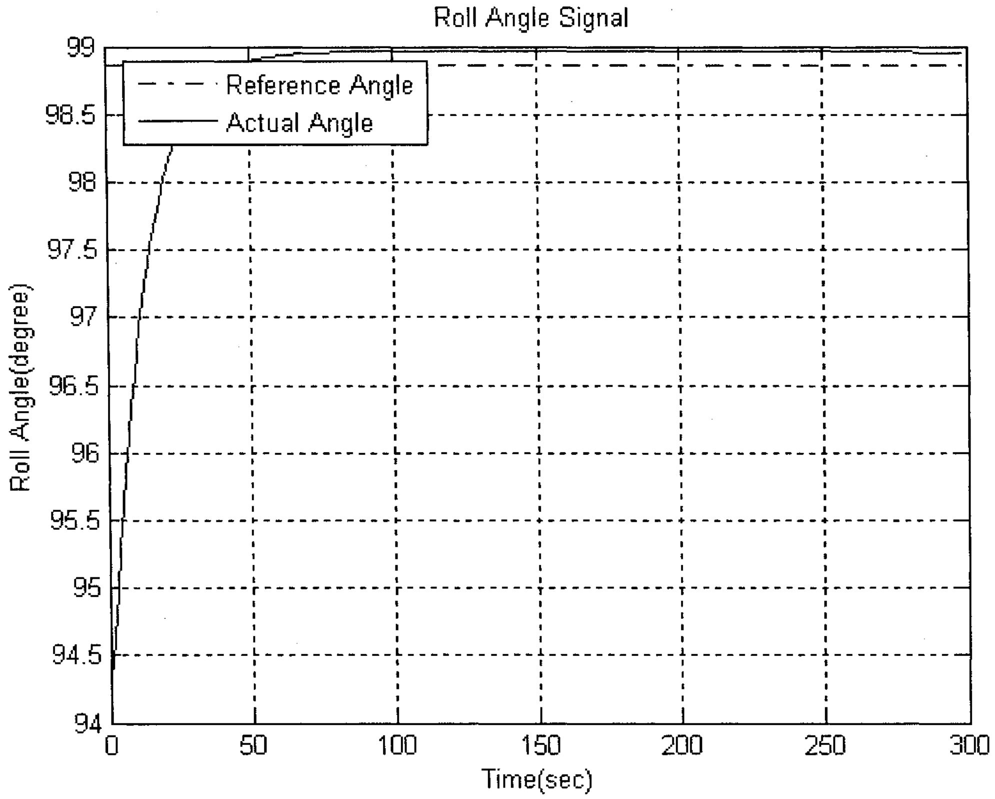
Initial body attitude for Z axis: $105.0973\deg$

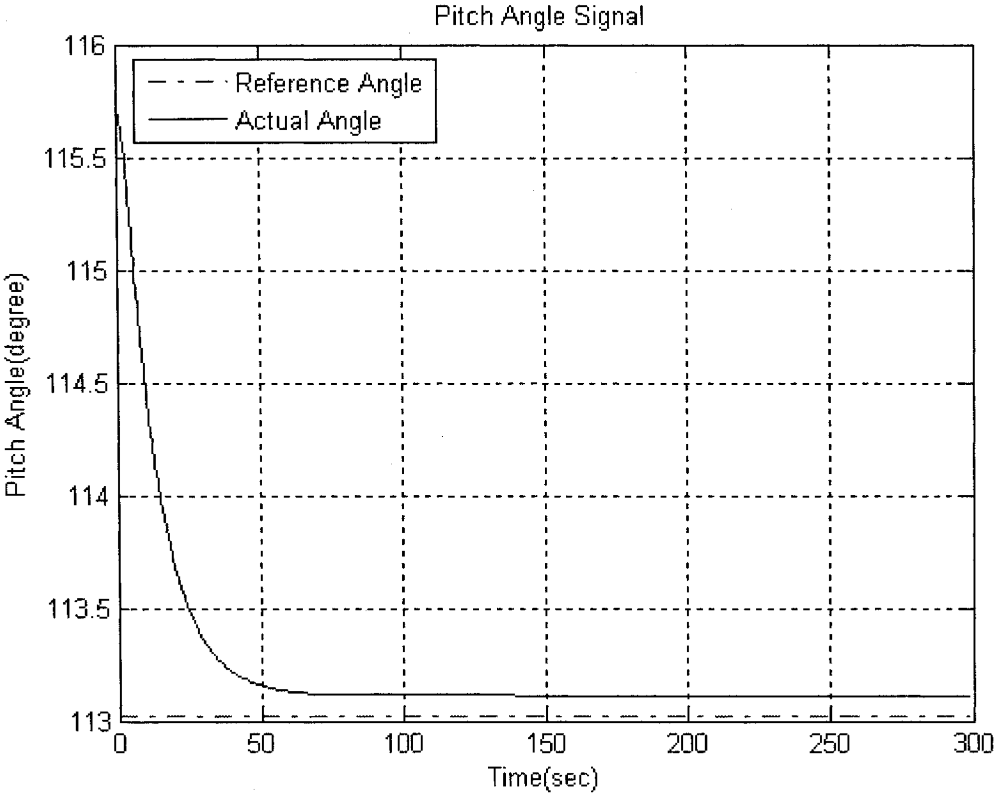
Initial body rate for Z axis: $2.3086\times10^{-5} rad/\sec$

Initial wheel speed on Z axis: $24.4470 rad/\sec$

Setpoint change on Z axis: $1.0684\deg$

The reference and actual angles are shown in the figures below. Clearly, these PID controller applied perform well and the attitude control system can track the command setpoint well with satisfied accuracy, overshoot and settling time.


Roll Angle Signal
Reference Angle
Actual Angle
Roll Angle(degree)
99
98.5
98
97.5
97
96.5
96
95.5
95
94.5
94
0
50
100
150
200
250
300
Time(sec)


Pitch Angle Signal
Reference Angle
Actual Angle
Pitch Angle(degree)
116
115.5
115
114.5
114
113.5
113
0
50
100
150
200
250
300
Time(sec)

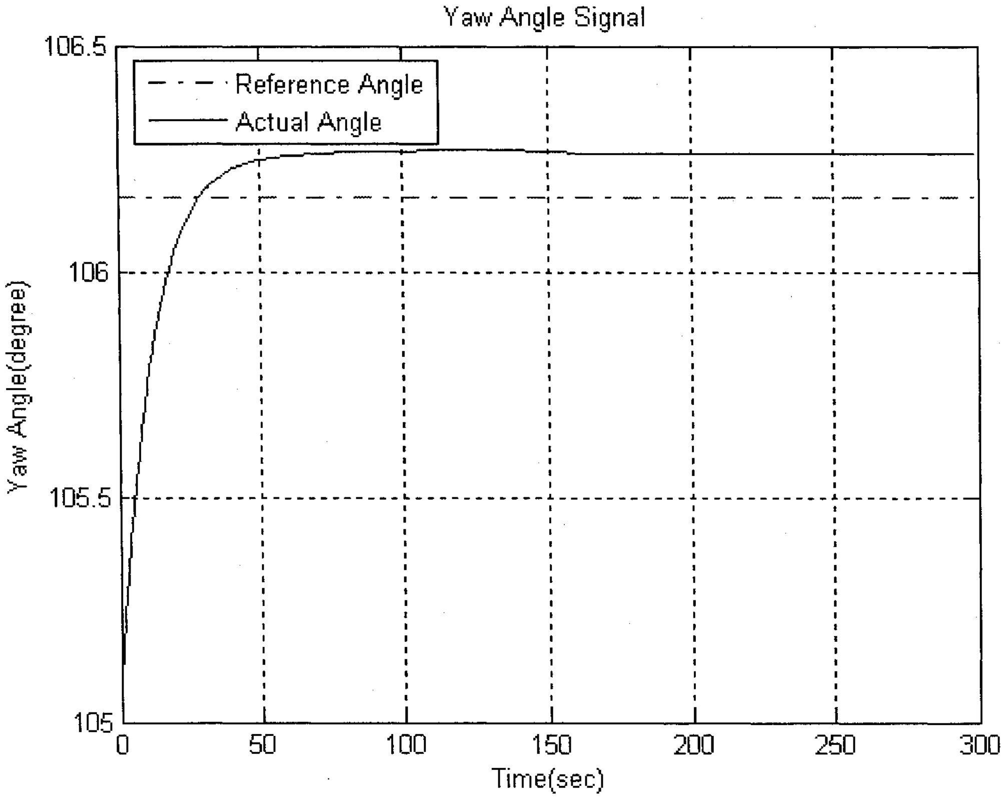


Figure 2.5 Body Attitude Performance with PID Controllers

2.6 Conclusions

In this chapter, a brief introduction to the attitude control system was given and a detailed description about the dynamics of the reaction wheels was provided. Three separate PID control loops were designed to achieve the desired pointing accuracy requirements. The effectiveness of these PID controllers was shown by the simulation results.

Chapter 3

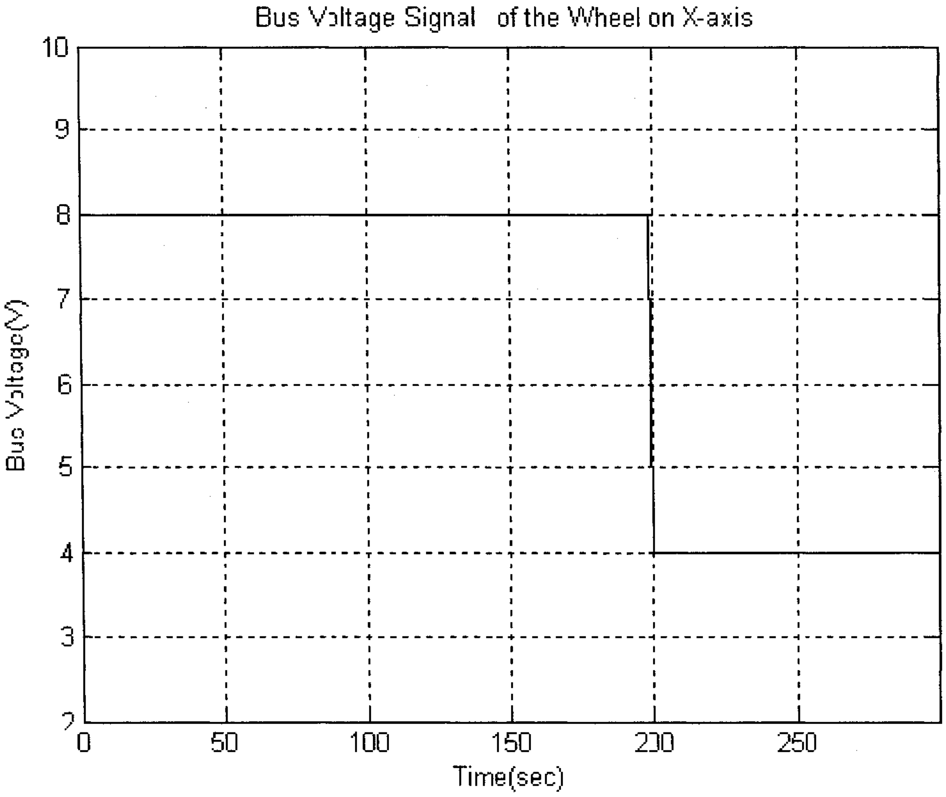
Linear Observer-based Fault Detection and Isolation in Reaction Wheels

3.1 Fault Detection and Isolation Problem Description

As we have discussed in Chapter 2, for the purpose of attitude control of spacecraft, three active and one redundant reaction wheels servers as the actuators of our system. From this point of view, reaction wheels play an essential role in the control system and their working condition should be carefully monitored. Any faults occurring inside the wheels should be detected and isolated quickly and correctly.

Through section 2.2.2, we know three parameter values in the wheel heavily impact the performance of the wheel. These three parameters are: bus voltage, motor current and temperature. Accordingly, the bus voltage fault, current loss (power loss) fault and temperature fault are three main faults that deserve our attention and discussion here.

As we know, the normal value of bus voltage is $8V$ in our thesis. If this bus voltage drops down, the motor torque may be limited at high speeds due to the increasing back-EMF of the motor and eventually results in reduced torque capacity of the wheel. When this value becomes too low, the whole attitude system will break down and the attitude of the spacecraft will be out of control. This can be easily observed from the Figure 3.1, which shows the attitude angle of X axis when the bus voltage of the wheel on X axis drops from $8V$ to $4V$ at 200 second after a set point change command was applied. Obviously, this type of fault should be detected and isolated early before the system becomes out of control.



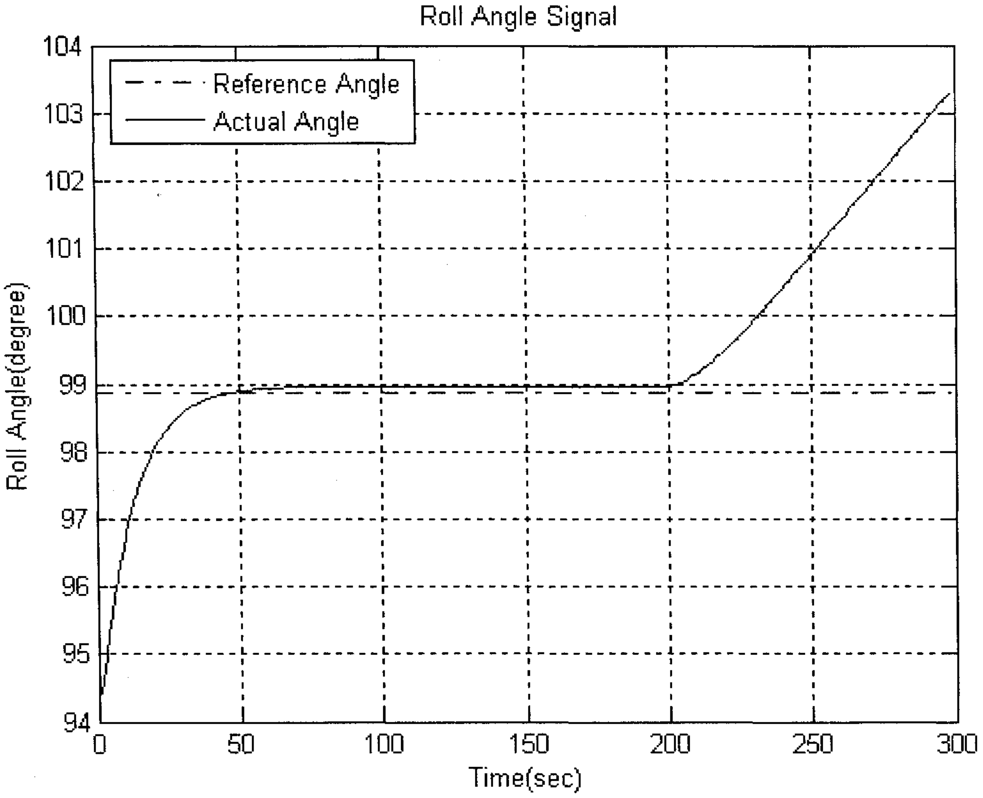


Figure 3.1 The attitude angle of X axis becomes divergent when the bus voltage drops too much

Similarly, since the motor torque is directly related to the motor current through one constant parameter k_t, when some kind of motor current loss happens in the reaction wheel, the motor torque will drop down accordingly. That means the wheel cannot supply enough motor torque for the attitude control. When the current loss becomes serious, the controlled attitude angle will become divergent as shown in Figure 3.2. Here, the current loss is model by the current multiplied by a parameter called current limiter with a normal value 1.

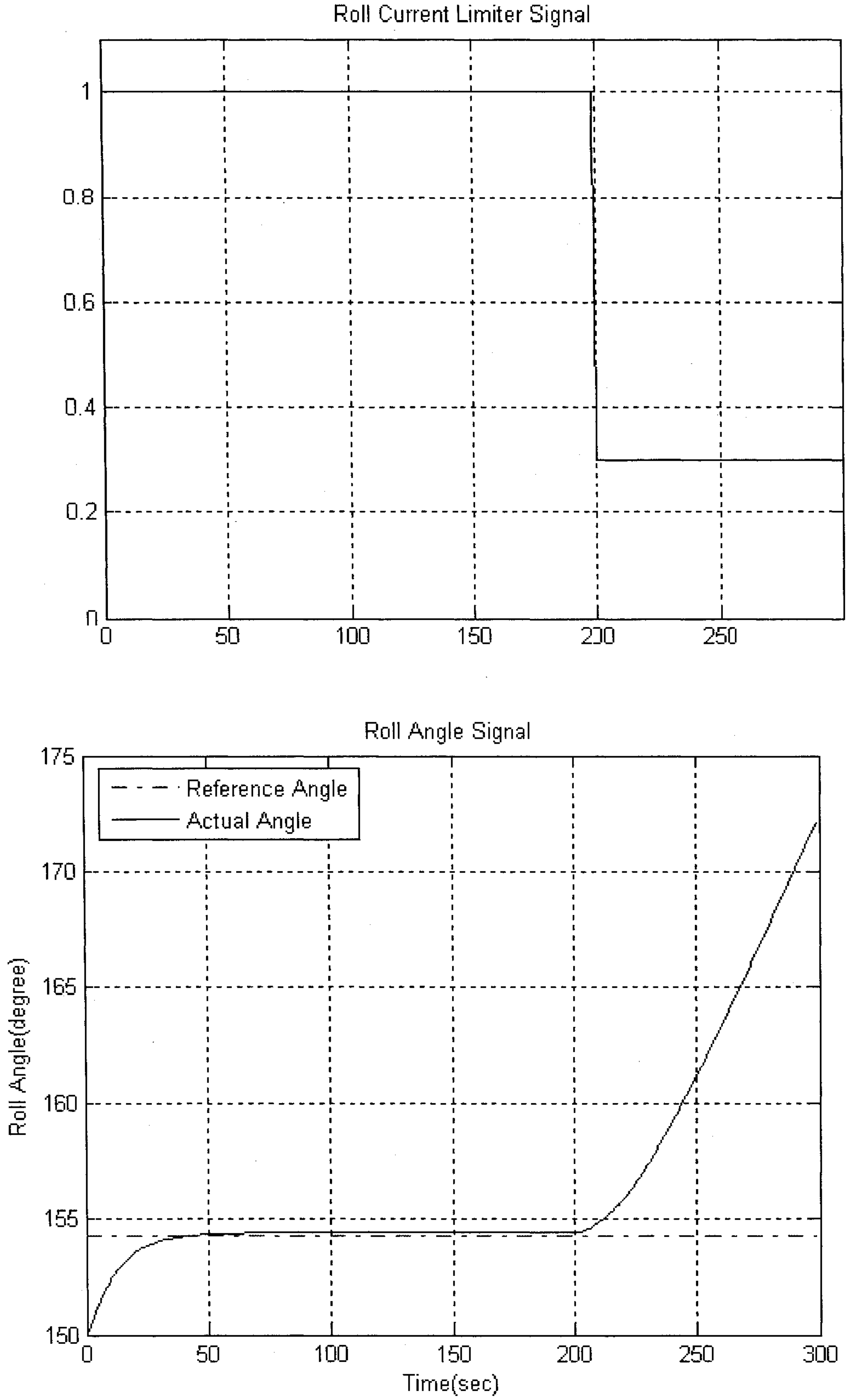


Figure 3.2 The attitude angle of X axis diverges when the current loss becomes serious

As we know, viscous friction is generated in the bearings due to the bearing lubricant. Through this friction, we can estimate the working condition of the bearings. When the bearings have been hurt seriously, this viscous friction will become much larger than that in normal condition. Since the temperature of the wheel is intensively related to the viscous friction in the wheel through the Equation 2.15, this suggests us to estimate the working condition of the bearings through monitoring the values of temperature of the wheel. If the temperature becomes too high, that means the bearings have been hurt much and they should be fixed.

3.2 Linear Observer-based Fault Detection and Isolation Structure

From what we have discussed in sub-section 1.2.4, in order to detect and isolate faults in actuators, the observer-based approaches are widely used in many literatures. The basic idea behind the observer-based approaches is to estimate the states of the system from the available measurements by using either Luenberger observer in a deterministic environment or Kalman filters in a noisy environment.

For our problems, the faults happen in the actuator block of the single-axis control block diagram in Figure 2.1. It is natural for us to design the observers to estimate the output values of the actuators, that is, the reaction torques generated by the wheels. In addition, to simplify the observer design algorithm and fault isolation, we design three independent observers for the wheels on the three axes separately. In this way, we can observe the value of reaction torque generated by each wheel at the same time. And combined with the observed (estimated) and actual values of reaction torques, we can get the information of three residual signals as well. In normal operation without faults, these three residual signals are small and around zero. Any fault happening in the wheel can be reflected by big change of residual signals from zero. A big change of residual on one axis from zero indicates the wheel aligned on that axis is likely to be faulty. By this scheme, we can detect and isolate the faulty wheel at the same time. The whole simulation diagram is plotted in Figure 3.3.

The details about the fault detection and isolation structure on one axis are shown in Figure 3.4. There are four steps for this fault detection scheme:

- Calculate the difference between the actual reaction torque and the estimate value from the linear observer. The design of linear observer will be shown in the next subsection;
- After calculating the abstract value of this difference signal, apply this new signal into a moving average filter to smooth the signal and this smoothed signal servers as the residual. Details about the moving average filter will be shown in Chapter 4;
- Through many cases (105 cases in our thesis) in normal operation (refer to Table 2.3) without fault, we can determine three threshold curves for all three axes separately for fault detection and isolation. Details will be shown in Chapter 4;
- Fault detection and isolation using the threshold curves: If any of these three residual signals exceed the corresponding threshold curve, fault has occurred in the reaction wheel aligned on that axis.

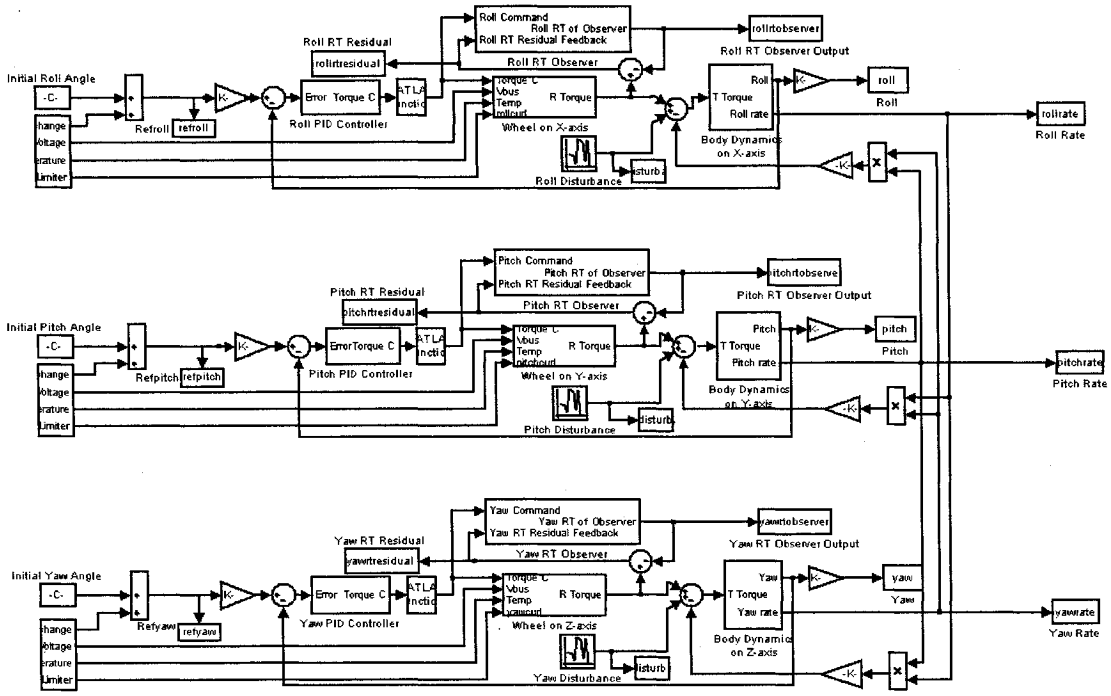


Figure 3.3 Linear Observer-based Fault Detection and Isolation Diagram

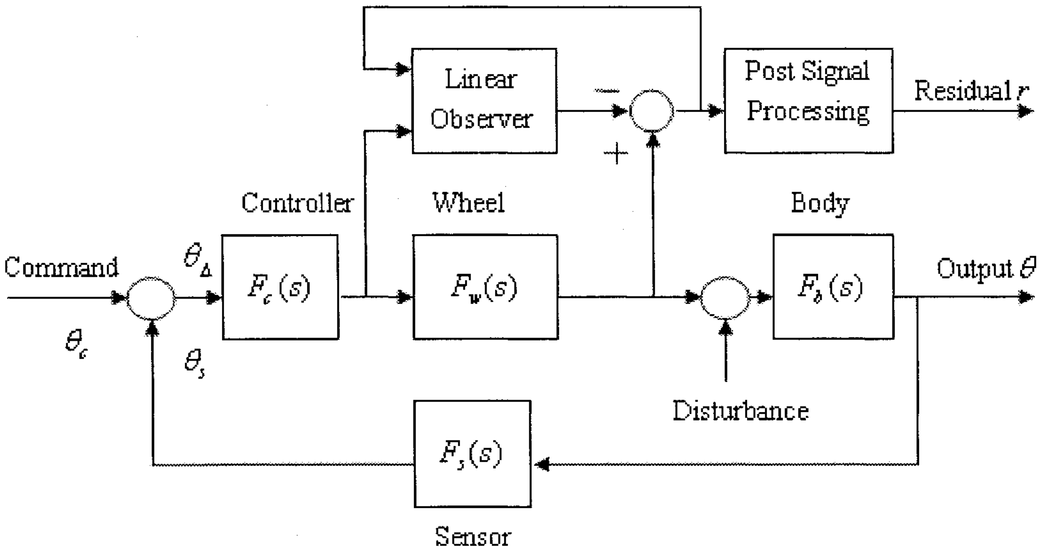


Figure 3.4 Linear Observer-based Fault Detection and Isolation Structure on One Axis

3.3 Linear Observer Design Algorithm

A general Luenberger observer design method is explained in [58]. For simplicity, we use a full order linear observer in our thesis. Its design block diagram is shown in Figure 3.5. The design procedure of our specific problem is:

- From the analysis before, we know the transfer function of the reaction wheel (nearly ideal model) is:

  $$F_w = \frac{-G_d k_t Js}{Js + \tau_v}$$

- Using this transfer function, we can get the state space form of the reaction wheel system

  $$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}$$

  with $A = [\text{-}0.04987]; B = [0.01563]; C = [0.01759]; D = [\text{-}0.00551];$.

- We calculate the eigenvalues of A , λ_A ;

- We make the observer 4 times faster than the original wheel system, that is making the eigenvalues of $(A - LC)$, $\lambda_{A-LC} = 4 \times \lambda_A$;

- By using the Ackermann's formula, we can calculate the value of feedback gain L in design diagram and we construct the linear observer as:

  $$\begin{cases} \dot{\hat{x}} = A\hat{x} + Bu + L(y - \hat{y}) \\ \hat{y} = C\hat{x} + Du \end{cases} \tag{3.1}$$

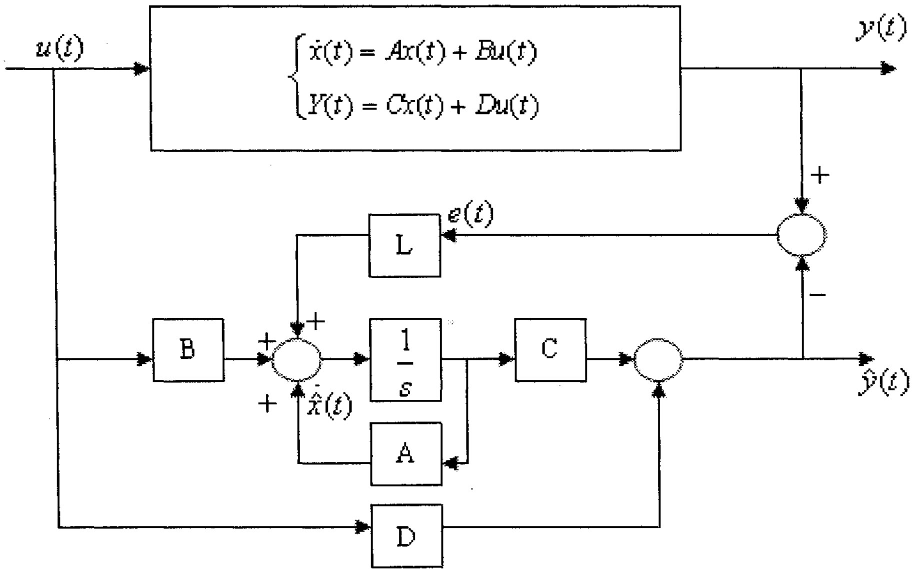
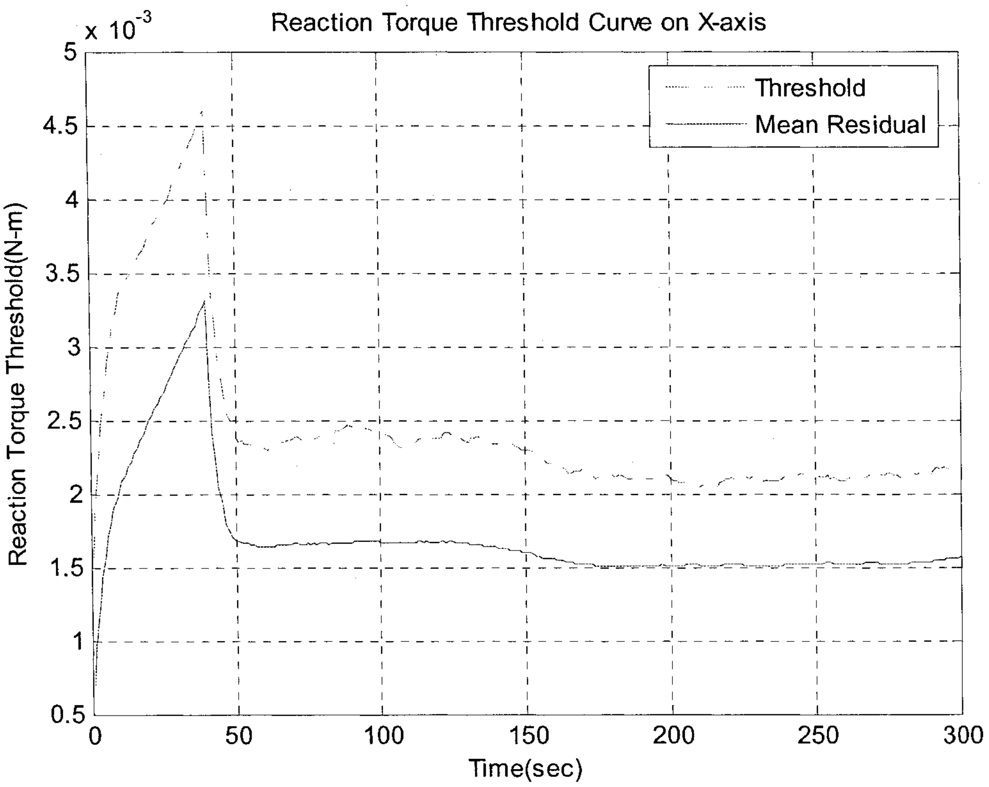


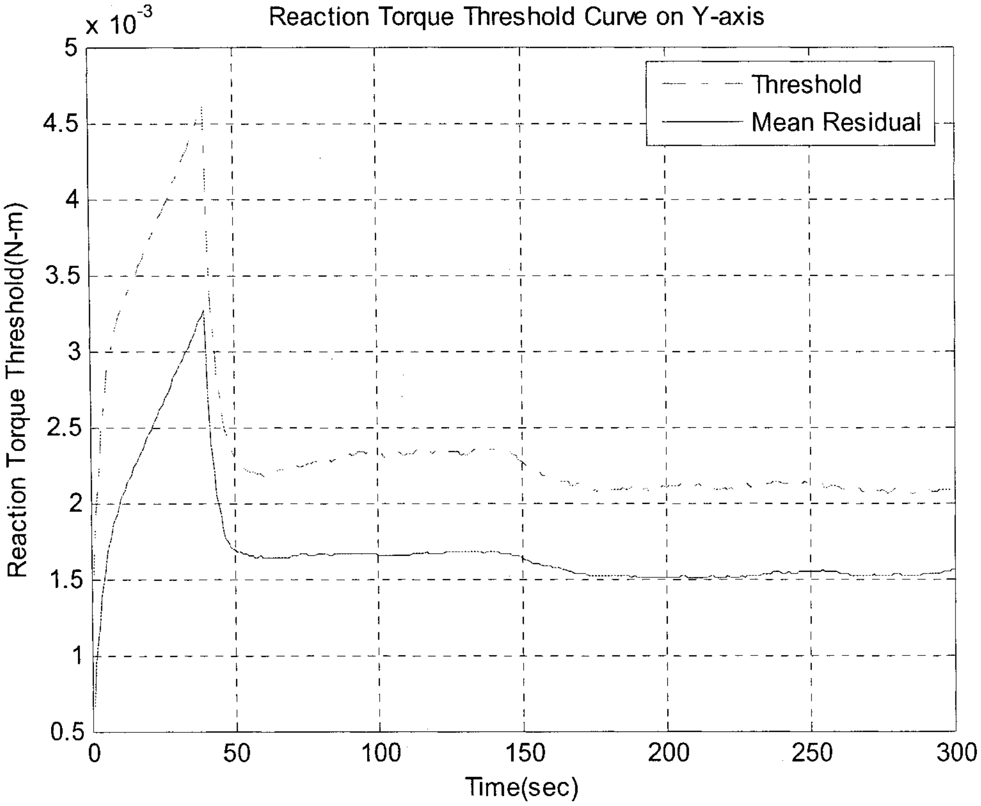
Figure 3.5 Full Order Linear Observer Design

3.4 Simulation Results of Linear Observer-based FDI

3.4.1 Threshold Curves Determination

As discussed in subsection 3.3, based on residual curves we accumulated in 105 cases in the normal operation (fault free), and after some post-processing, we can get three threshold curves for fault detection and isolation as shown in Figure 3.6:


Reaction Torque Threshold Curve on X-axis
x 10^-3
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
50
100
150
200
250
300
Time(sec)
Reaction Torque Threshold(N-m)
Threshold
Mean Residual


Reaction Torque Threshold Curve on Y-axis
x 10^-3
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
50
100
150
200
250
300
Time(sec)
Reaction Torque Threshold(N-m)
Threshold
Mean Residual

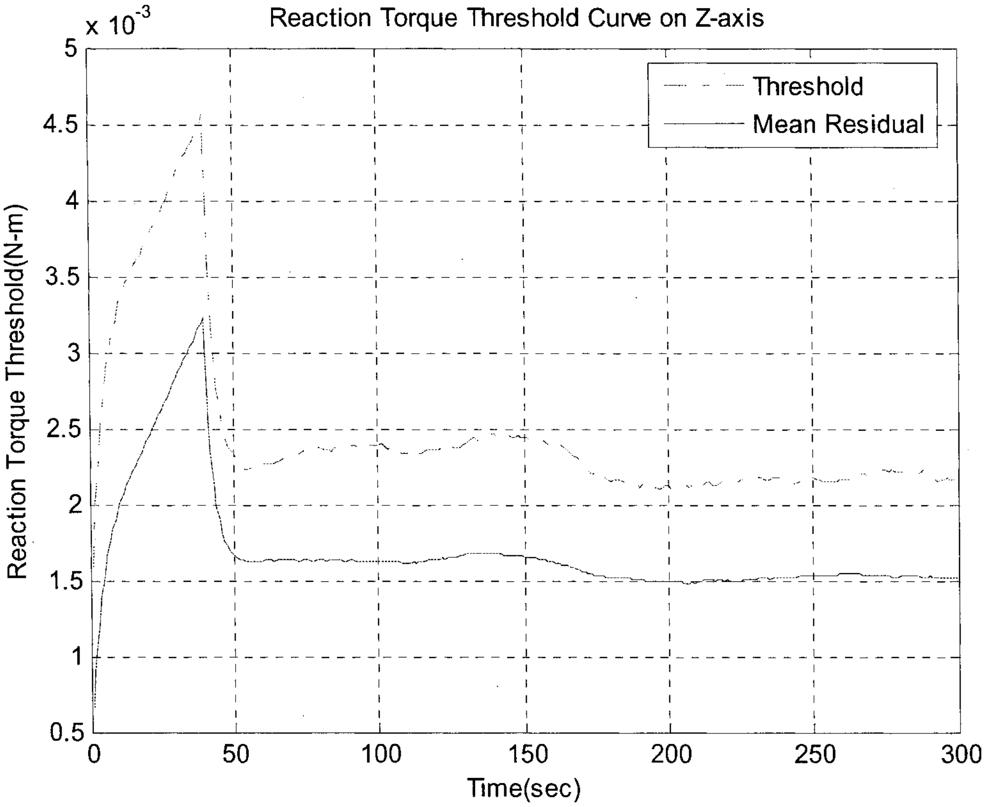


Figure 3.6 Three Threshold Curves of Linear Observer-based Scheme

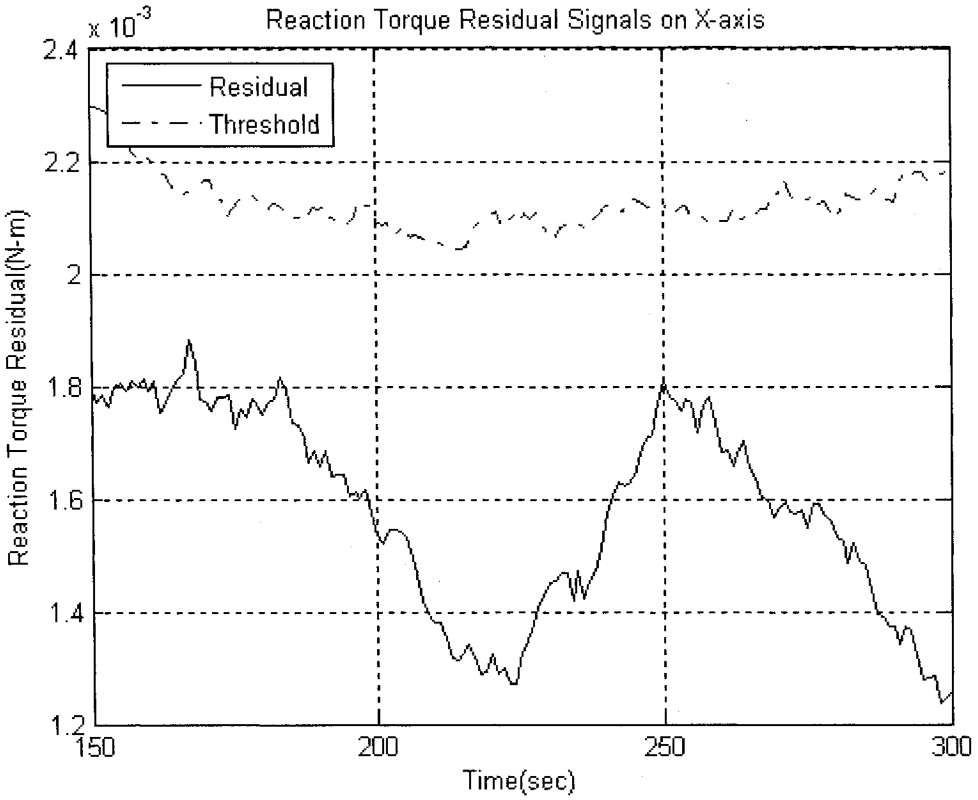
To verify the reliability of these threshold curves, we verify them with the original 105 groups of data to see whether these residual curves are beneath the threshold curves. Results are shown in Table 3.1. From this table, we can see these threshold curves are very reliable in normal operation.

Threshold Curve	$X-Axis$	$Y-Axis$	$Z-Axis$
Reliability	99%	99%	100%

Table 3.1 Threshold Curves' Reliability Test in Normal Operation

3.4.2 FDI Performance in Fault Free Cases

For the fault free cases in our problem, each of three residual curves is beneath its corresponding threshold curve. By using this linear observer-based scheme, no false detection has been made. A typical result is shown in Figure 3.7.



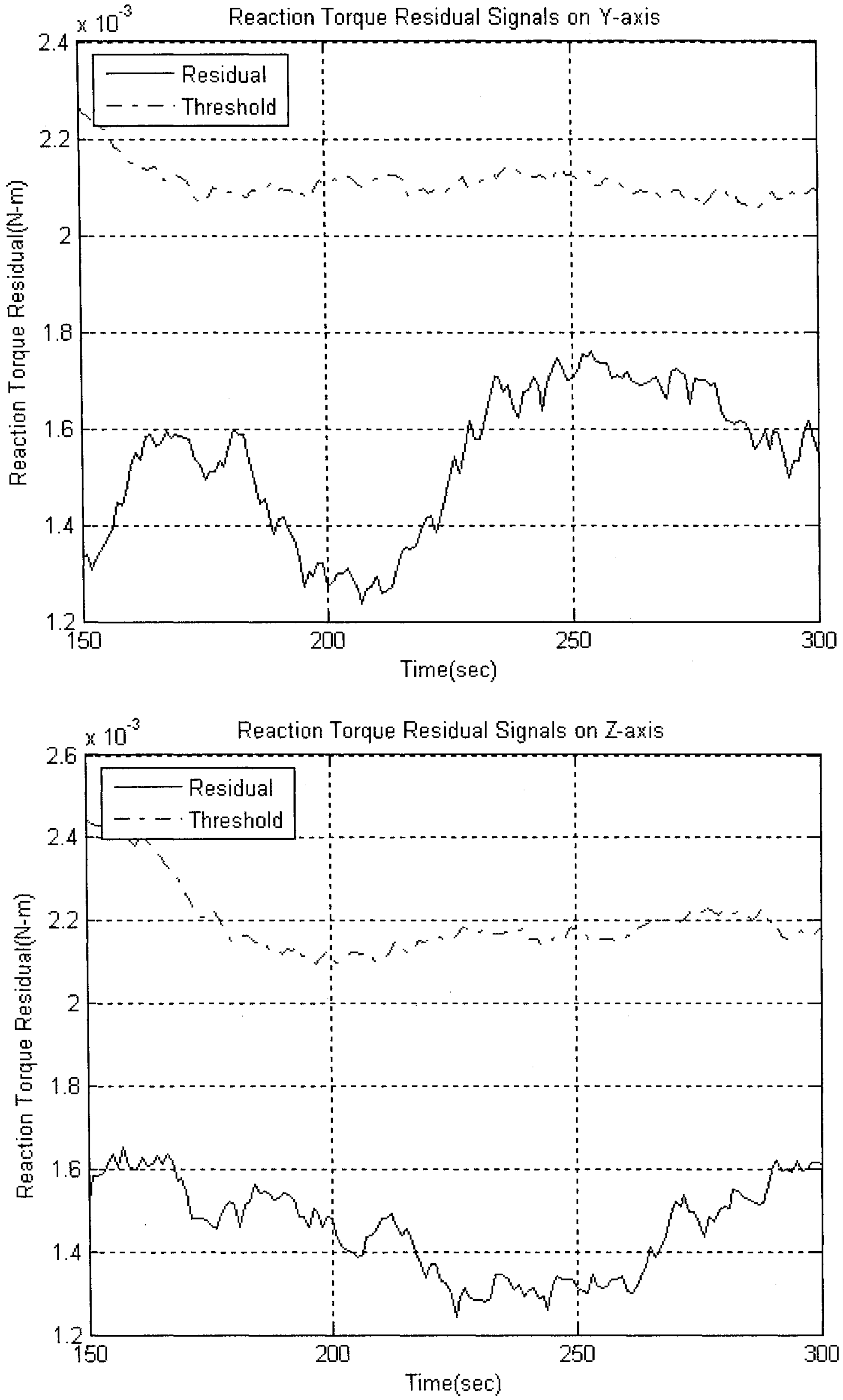
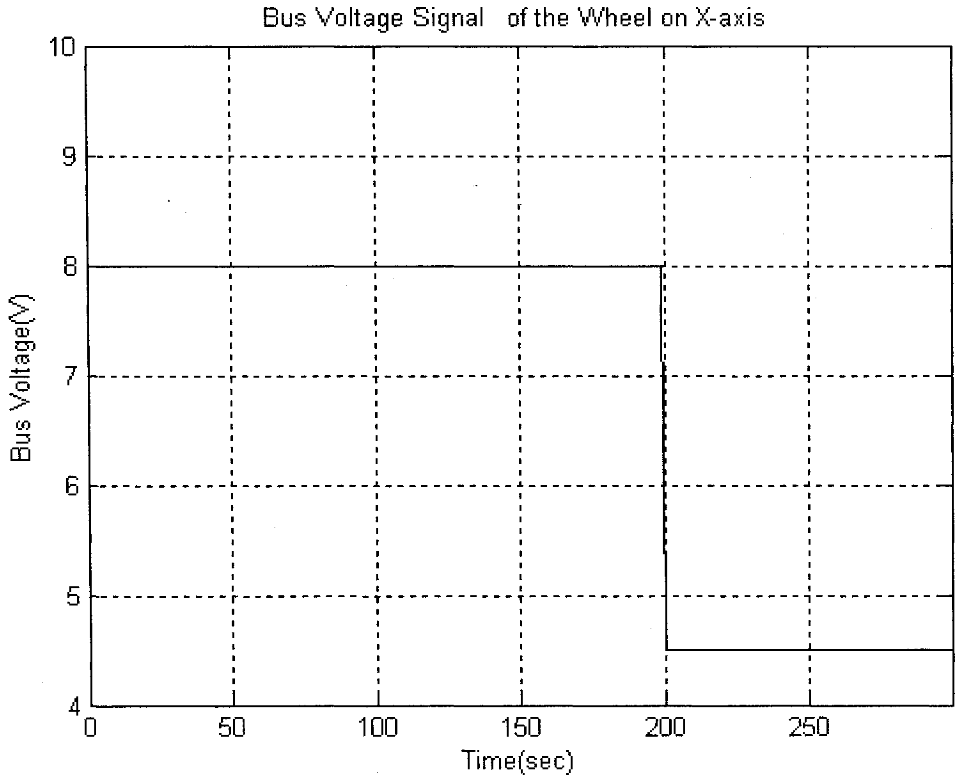


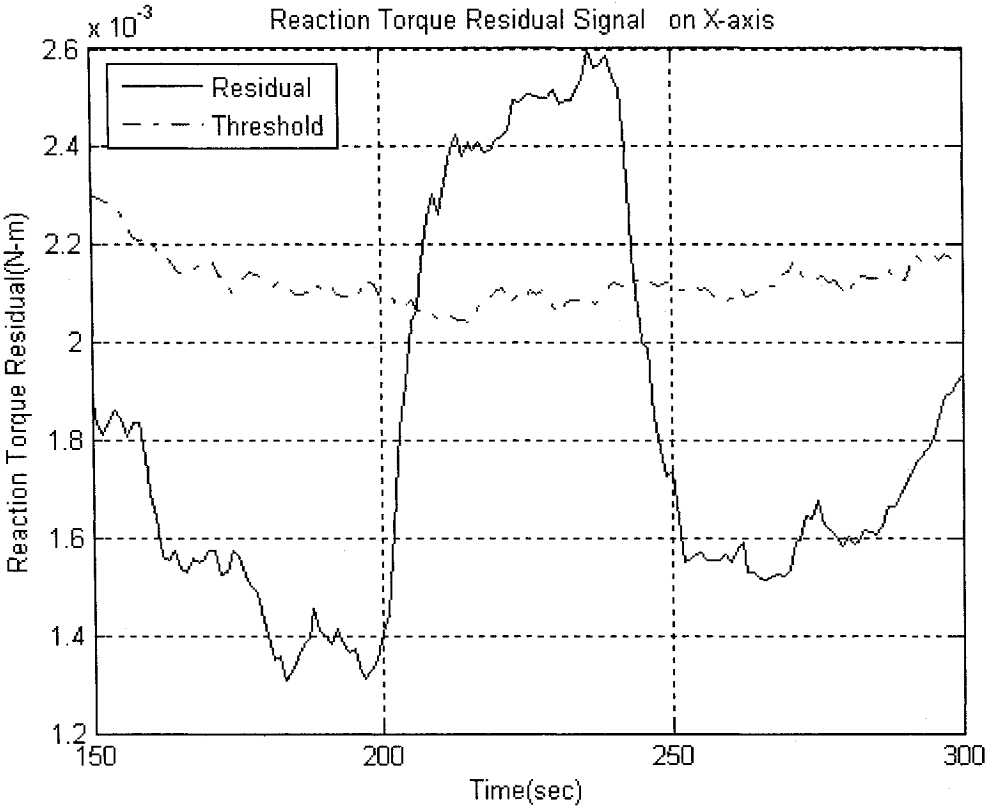
Figure 3.7 Fault Detection and Isolation Performance in Fault-free Case

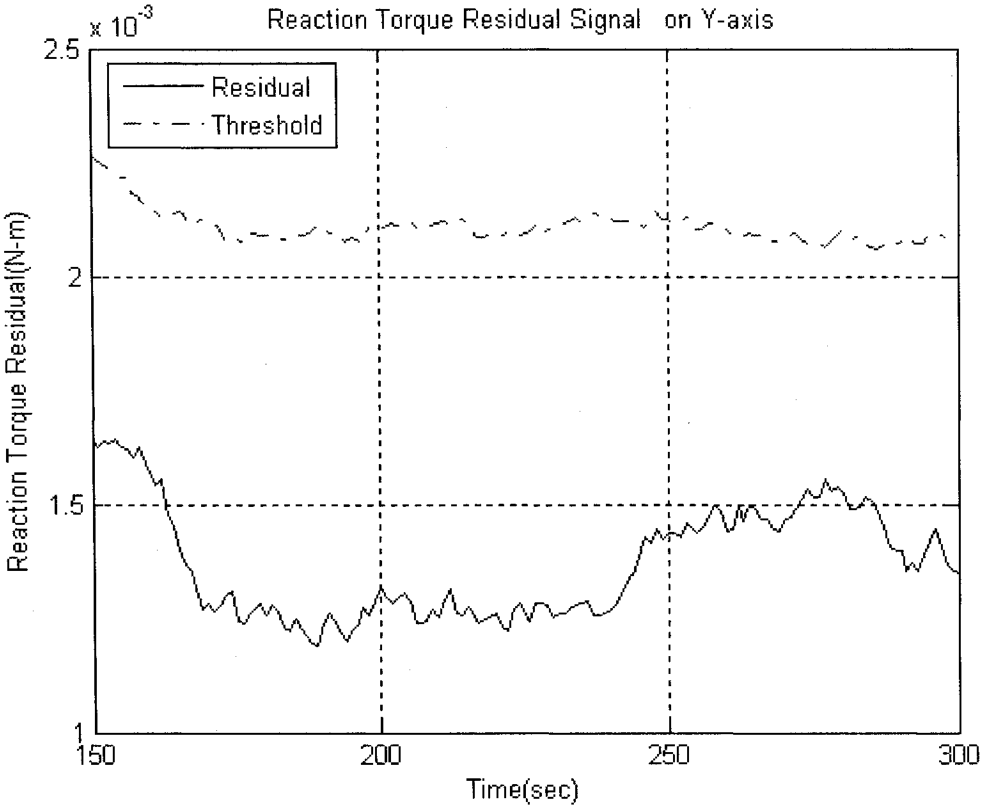
3.4.3 FDI Performance in Faulty Cases

Bus Voltage Faults

Figure 3.8 shows a case study of bus voltage fault detection and isolation. As we see, the bus voltage of the reaction wheel aligned on X axis dropped from the normal value $8V$ to $4.5V$ at 200 seconds after a set point change command was applied. The residual of the wheel on X axis exceeded the corresponding threshold curve shortly after the fault happened. At the same time, the residuals of the other two wheels are still beneath their related threshold curves and not affected by the fault happened in the wheel of X axis. So the bus voltage fault in the wheel of X axis is detected and isolated correctly, however, it is incorrectly identified as nonfaulty after less than 40 seconds even though the fault is still present.




x 10^-3
Reaction Torque Residual Signal on X-axis
Residual
Threshold
Reaction Torque Residual(N-m)
2.6
2.4
2.2
2
1.8
1.6
1.4
1.2
150
200
250
300
Time(sec)


x 10^-3
Reaction Torque Residual Signal on Y-axis
Residual
Threshold
Reaction Torque Residual(N-m)
2.5
2
1.5
1
150
200
250
300
Time(sec)

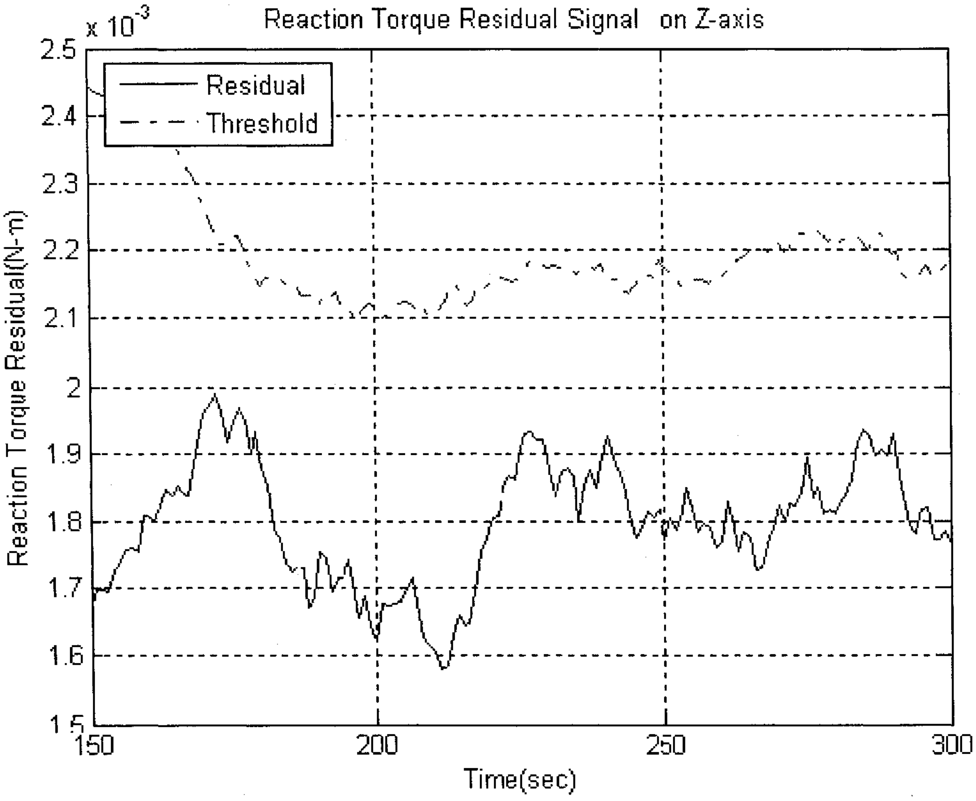
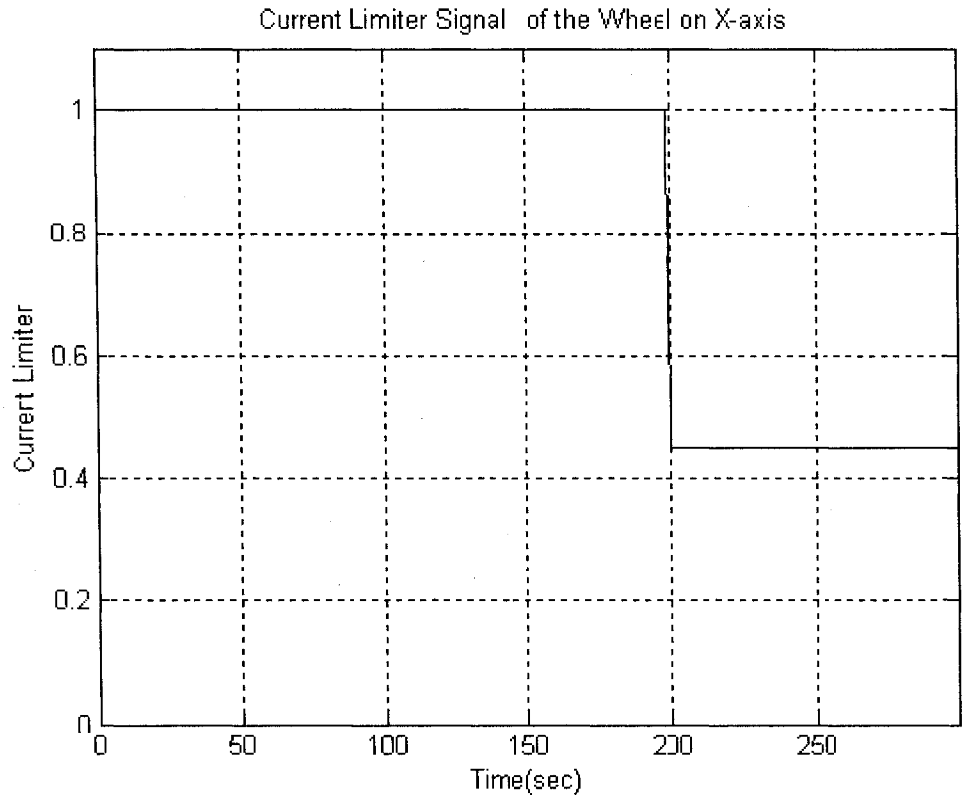
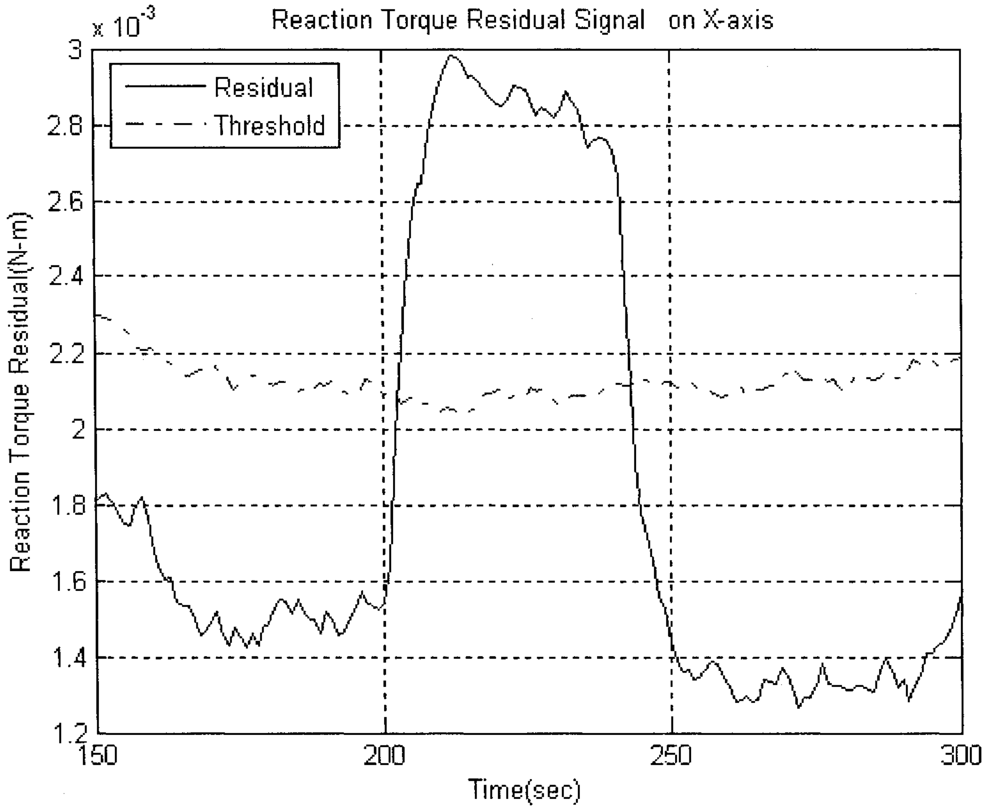


Figure 3.8 Bus Voltage Fault Detection and Isolation Case

Current Loss Faults

Similarly, a current loss fault in the wheel of X axis was properly detected and isolated as shown in Figure 3.9. The attitude of spacecraft was changed according to a set point change command at zero second. And the current limiter signals of the wheel on X axis dropped from 1 to 0.45 at 200 second, which represents 55% motor current loss. Accordingly, the residual of X axis increased above the threshold curve after some time delay and indicated that the wheel on X axis was faulty, however, it is incorrectly identified as nonfaulty after less than 40 seconds even though the fault is still present.


Current Limiter Signal of the Wheel on X-axis
Current Limiter
1
0.8
0.6
0.4
0.2
0
0
50
100
150
200
250
Time(sec)


Reaction Torque Residual Signal on X-axis
x 10^-3
Reaction Torque Residual(N-m)
Residual
Threshold
3
2.8
2.6
2.4
2.2
2
1.8
1.6
1.4
1.2
150
200
250
300
Time(sec)

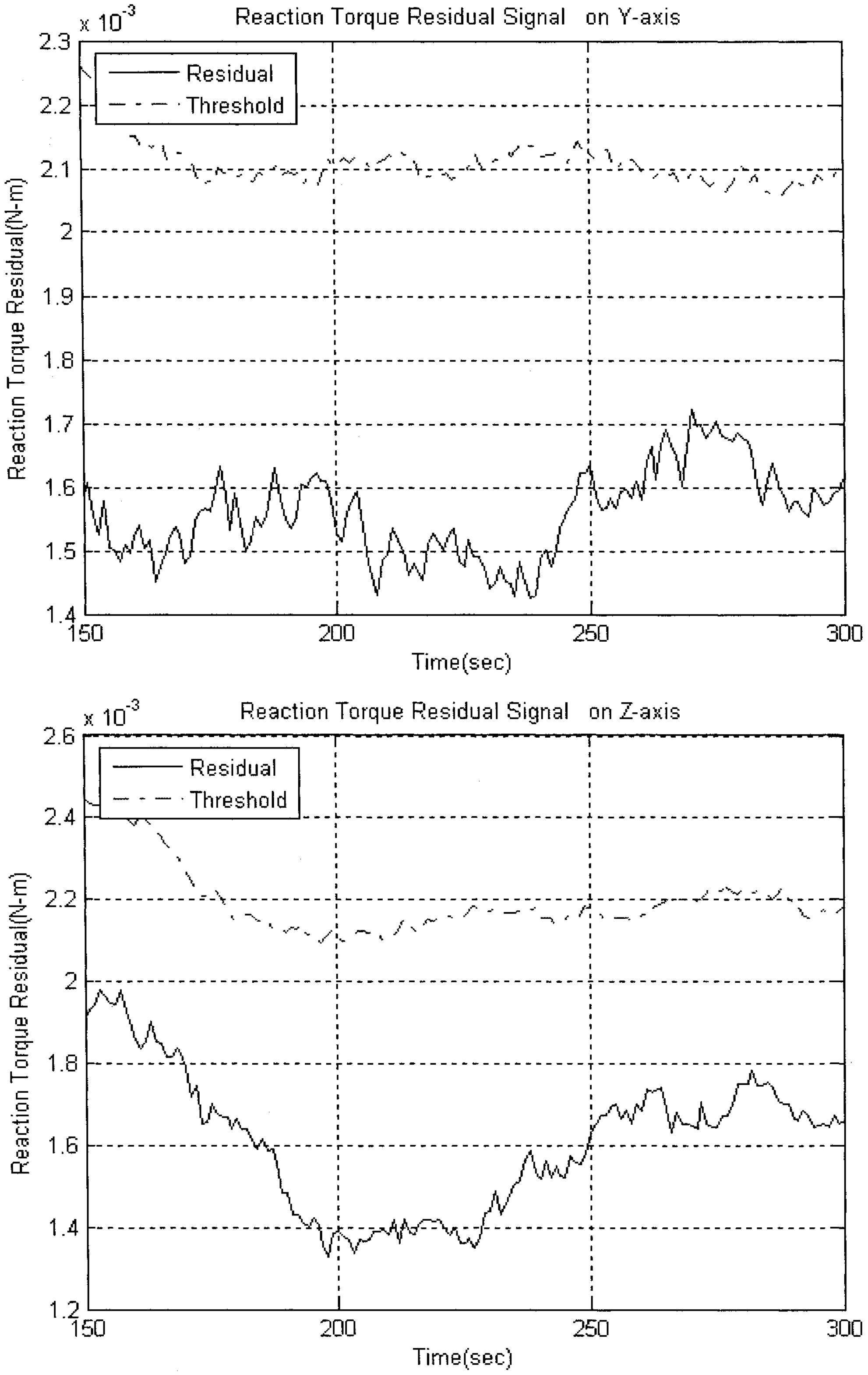
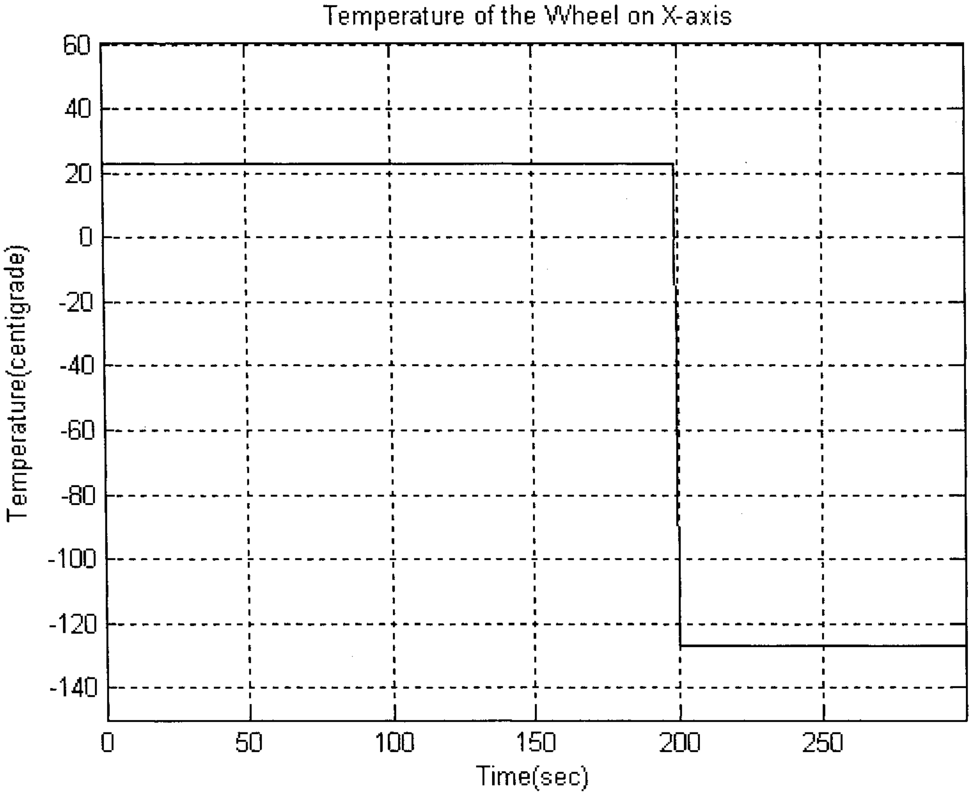
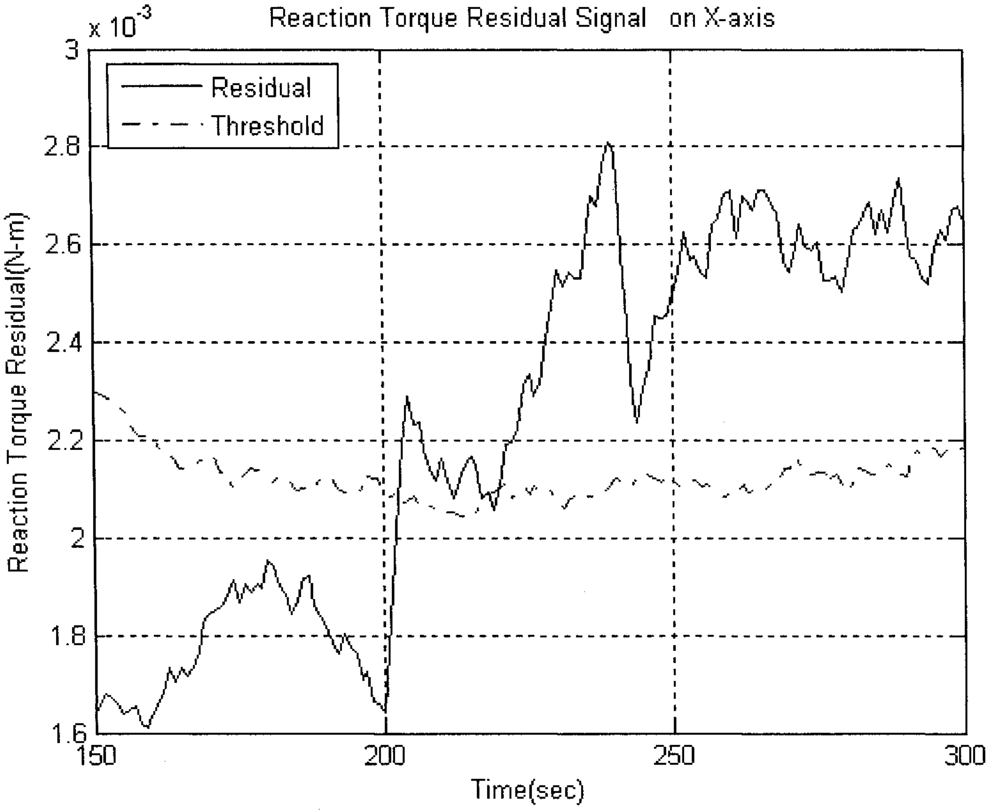


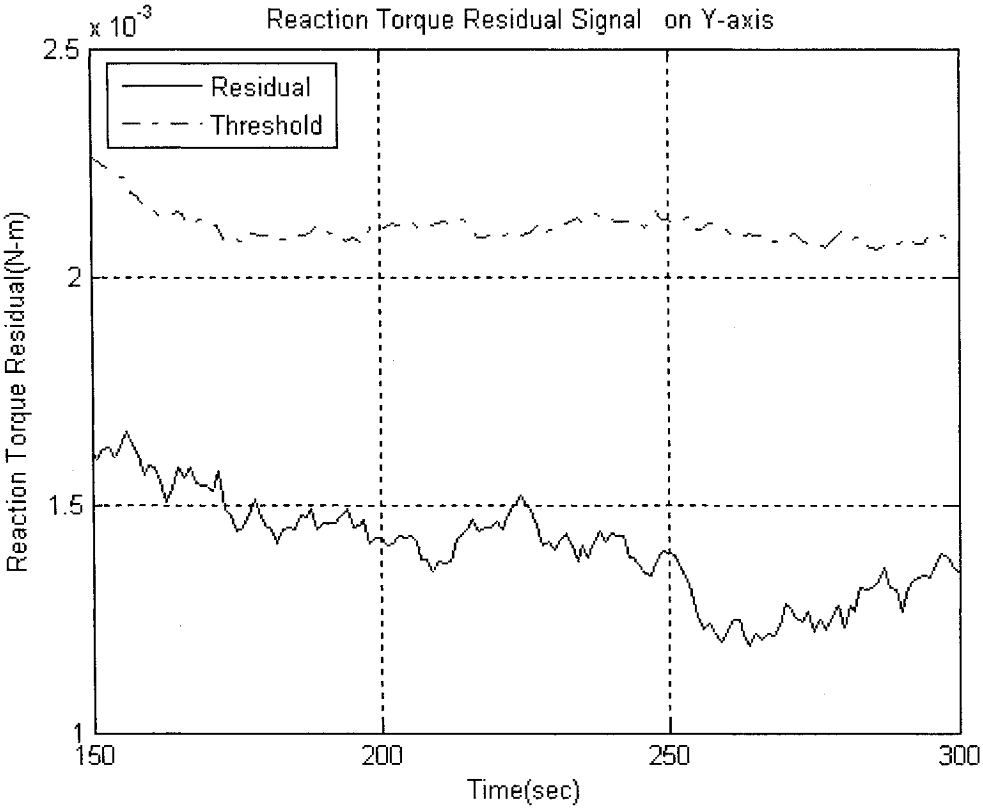
Figure 3.9 Current Loss Fault Detection and Isolation Case

Temperature Faults

The simulation results in Figure 3.10 also prove that this kind of linear observer-based scheme is effective in detecting and isolating a temperature fault occurring in the wheel on X axis.




x 10^-3
Reaction Torque Residual Signal on X-axis
Residual
Threshold
3
2.8
2.6
2.4
2.2
2
1.8
1.6
Reaction Torque Residual(N-m)
150
200
250
300
Time(sec)


x 10^-3
Reaction Torque Residual Signal on Y-axis
Residual
Threshold
2.5
2
1.5
1
Reaction Torque Residual(N-m)
150
200
250
300
Time(sec)

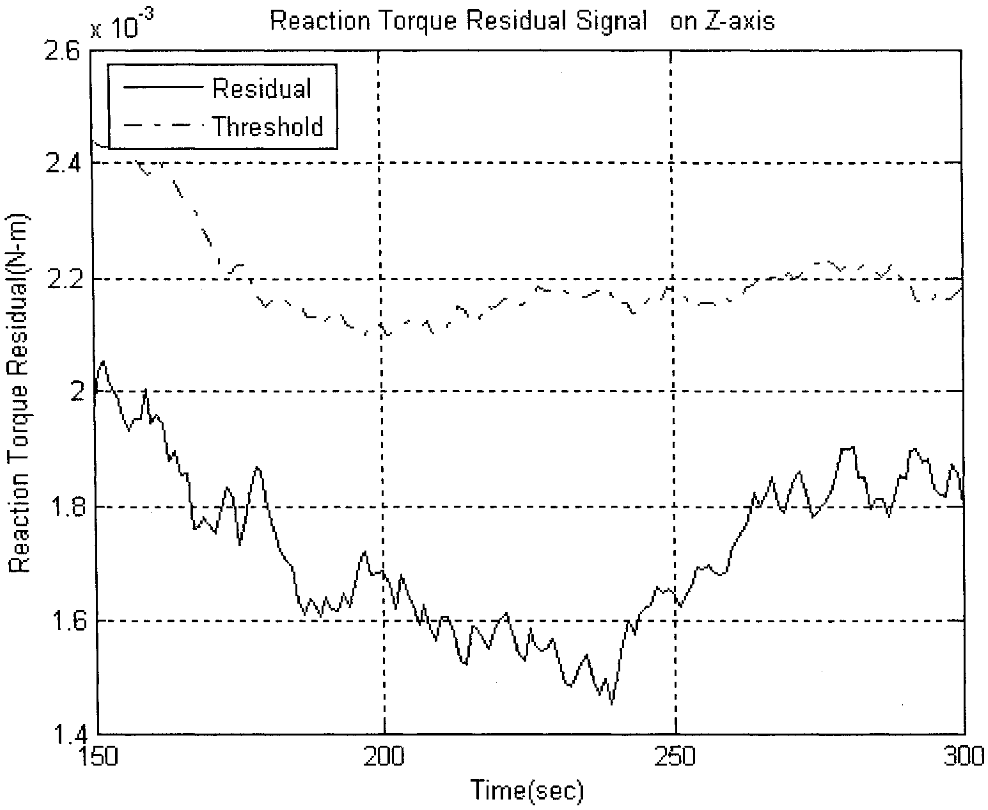
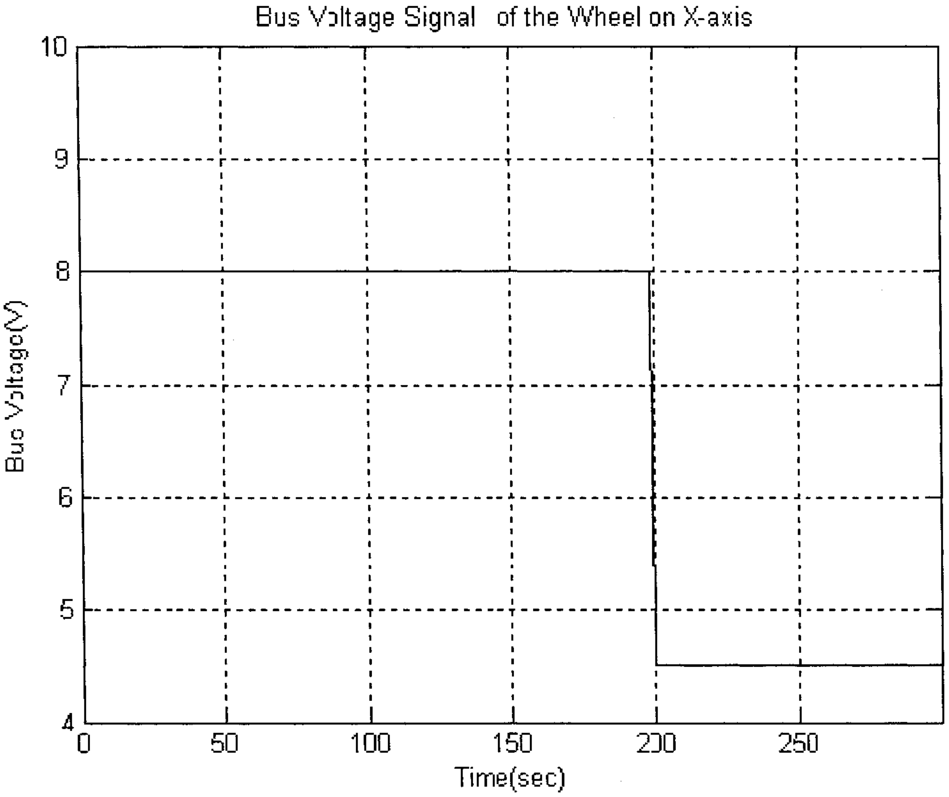


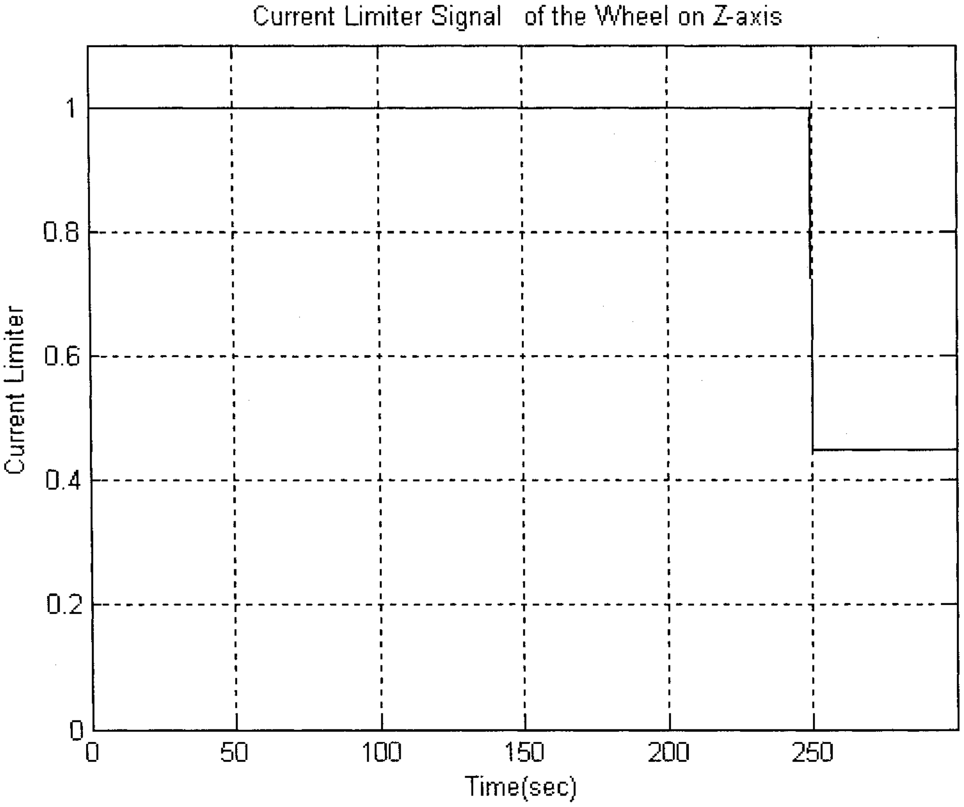
Figure 3.10 Temperature Fault Detection and Isolation Case

Multiple Faults (faults happen in more than one axis)

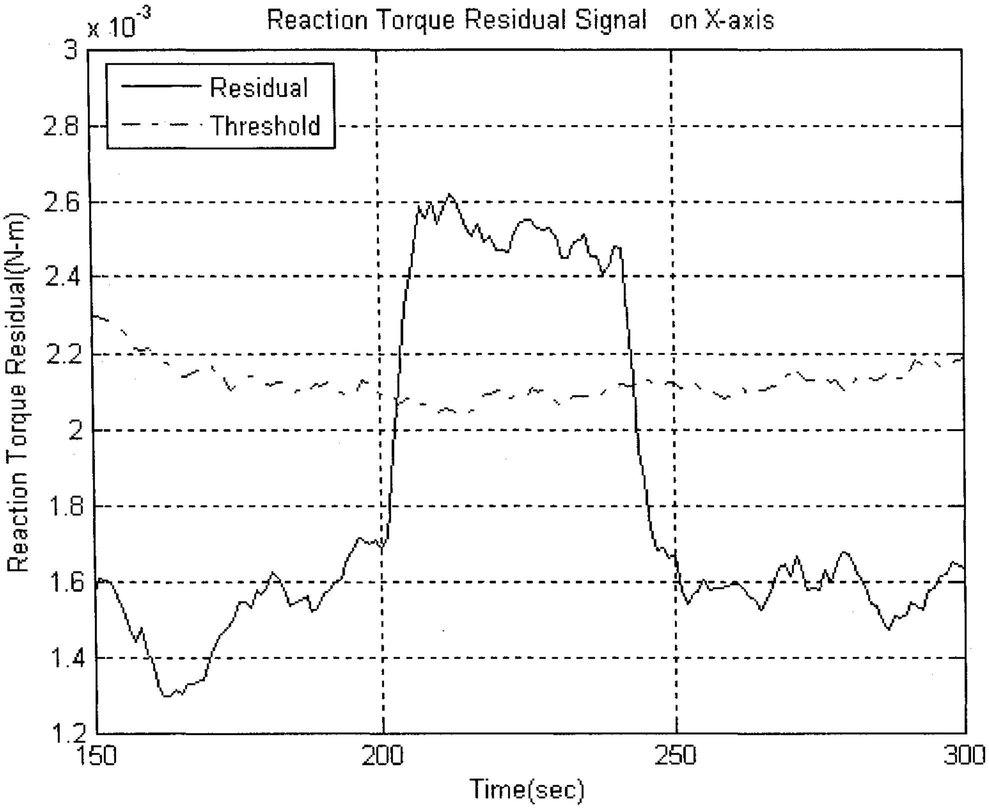
This case study is used to check the performance of this scheme under multiple faults, when more than one wheel was faulty at different time. Like this case, a bus voltage fault happened in the wheel of X axis at 200 second and followed a current loss fault in the wheel of Z axis at 250 second. Obviously, From Figure 3.11, these two faults were both detected and isolated initially, they are incorrectly identified as nonfaulty after less than 40 seconds even though the faults are still present.

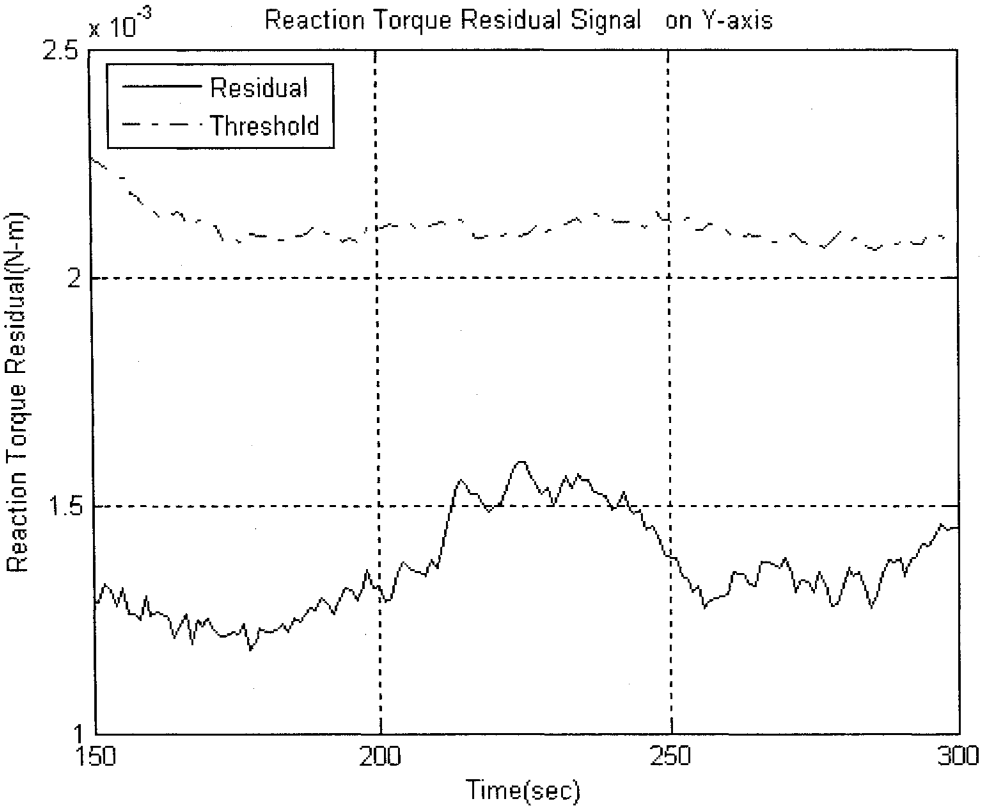


Bus Voltage Signal of the Wheel on X-axis
Bus Voltage(V)
10
9
8
7
6
5
4
0
50
100
150
200
250
Time(sec)



Current Limiter Signal of the Wheel on Z-axis
Current Limiter
1
0.8
0.6
0.4
0.2
0
0
50
100
150
200
250
Time(sec)


x 10^-3
Reaction Torque Residual Signal on X-axis
Residual
Threshold
Reaction Torque Residual(N-m)
3
2.8
2.6
2.4
2.2
2
1.8
1.6
1.4
1.2
150
200
250
300
Time(sec)


x 10^-3
Reaction Torque Residual Signal on Y-axis
Residual
Threshold
Reaction Torque Residual(N-m)
2.5
2
1.5
1
150
200
250
300
Time(sec)

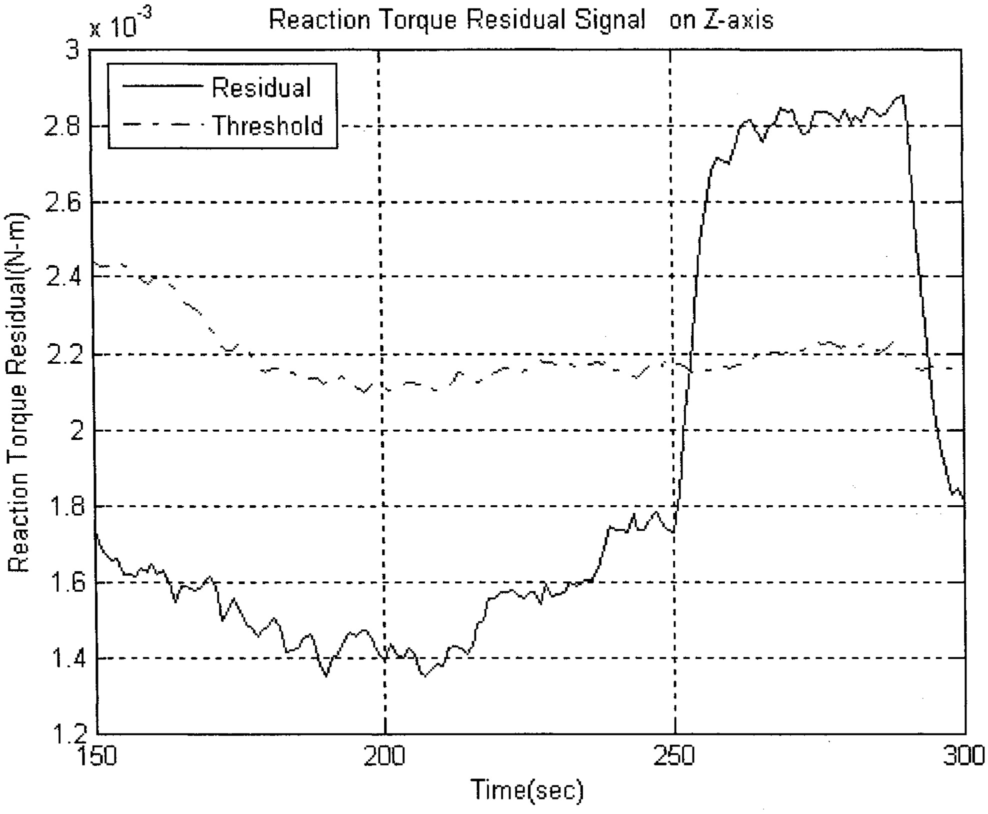


Figure 3.11 Multiple Faults Detection and Isolation Case

From previous simulation results, we can find that the linear observer-based scheme is not good enough for fault detection and isolation of reaction wheels. Due to the introduction of error feedback into the observers for stabilization, the observers will still follow the actual outputs of the wheels even when they became faulty. Thus the faulty wheel is easily incorrectly identified as nonfaulty after a short period even though the fault is still present.

3.5 Conclusions

In this chapter, firstly, we described the three types of faults happening in the reaction wheels. Next, we developed a linear observer-based scheme for FDI purpose in the wheels. This scheme is not quite suitable for fault detection and isolation in the wheels as shown in the simulation results. Based on this scheme, the faulty wheel is easily incorrectly identified as nonfaulty after a short period even though the fault is still present.

Chapter 4

Neural Network Observer-based Fault Detection and Isolation in Reaction Wheels

4.1 General Introduction to Neural Networks

Neural networks or precisely artificial neural networks was originated from the biological concept. But now it has been thoroughly studied and widely used in the areas of control, signal processing, pattern recognition and fault diagnosis as well. A neural network can be defined as a massively parallel distributed processor made up of simple processing units, which has a natural propensity for storing experiential knowledge and making it available for use. This kind of unit is called neuron and the connection between two different neurons is called synaptic weight. And the procedure to store experiential knowledge or learning process is called a learning algorithm, which modify the synaptic weights in an orderly fashion to attain a desired design objective. A most important criterion of a trained network is its generalization performance, which means the network can generate a reasonable output when it encounters a new input.

According to [53], neural networks show three main properties suitable for the application in the thesis:

- Nonlinearity: A neural network, made up of an interconnection of nonlinear neurons, is itself nonlinear. This property makes it suitable for modeling the dynamic system which is 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. And 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.
- Adaptive: Neural networks have a built-in capability to adapt its synaptic weights to changes in the surrounding environment.

4.1.1 Neuron Model

Neuron is the basic processing unit of 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 4.1.

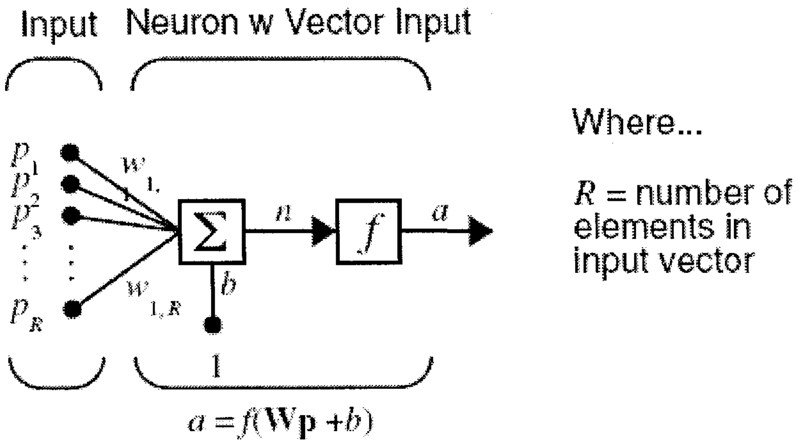


Figure 4.1 Neuron with R-element Input Vector [59]

The relation from input p to output a is:

$$a = f\left(Wp + b\right) \tag{4.1}$$

where W is synaptic weight matrix, b is a bias and f represents the activation (transfer) function. A lot of activation functions can be used and many of them are listed in the Matlab toolbox. Three of the commonly used are shown below in Figure 4.2.

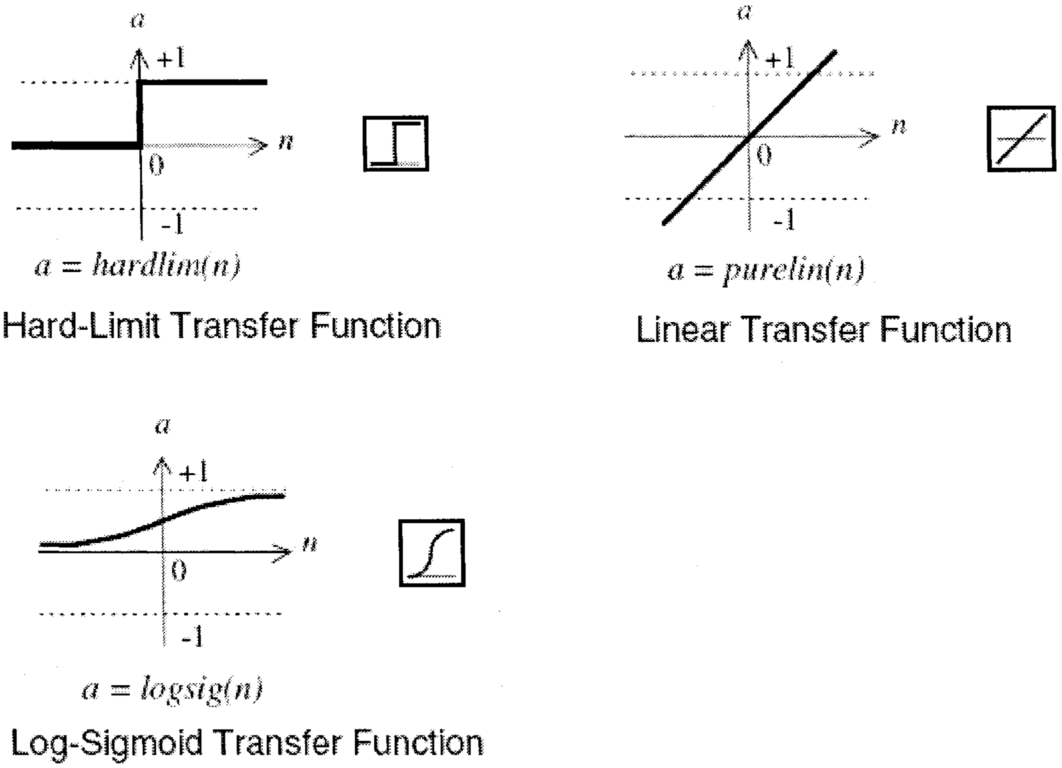


Figure 4.2 Three Commonly Used Activation Functions [59]

4.1.2 Network Architectures

Network architecture represents how the neurons of a network are arranged and interconnected together. Referring to [53], where the various existing network architectures and divided into three fundamental categories:

1. Single-layer feed-forward networks

A single-layer feed-forward network with R input elements and S neurons is shown below in Figure 4.3. In this kind of network, each element of input vector p is connected to each neuron input through the weight matrix W . And the neuron layer outputs form a column vector a , whose expression is shown at the bottom of the figure.

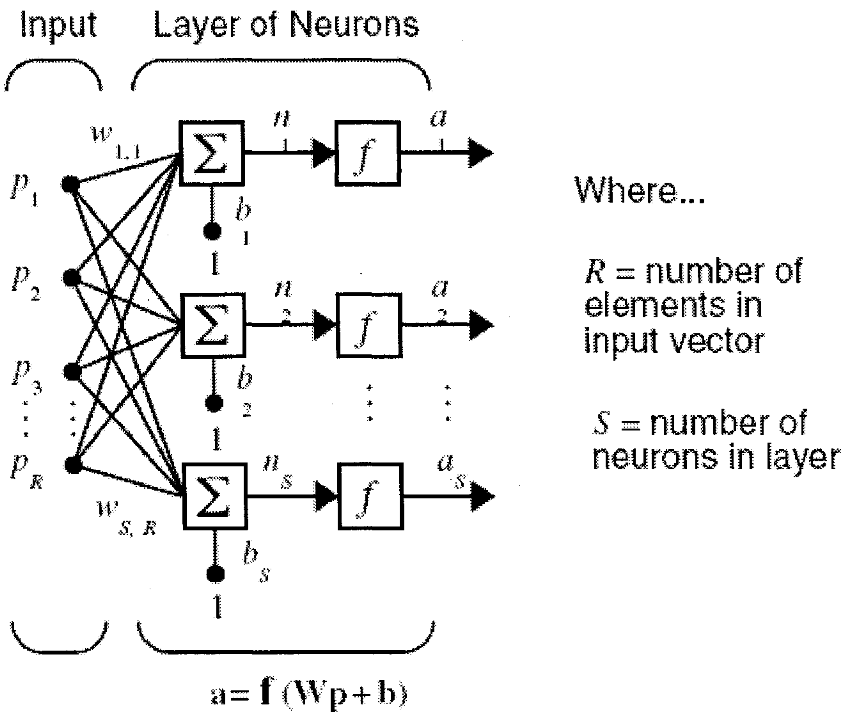


Figure 4.3 A Single-layer Feed-forward Network [59]

2. Multilayer feed-forward networks

A single-layer feed-forward network is not very powerful in terms of computational or representational capabilities. But if more layers contained in the network, it can even represent very complex nonlinear mapping from the inputs and outputs of a system. As shown in literature [53], a three layer (one input layer, one hidden layer and one output layer) fully connected feed-forward network with sigmoidal activation functions can represent an arbitrary mapping between the input and output variables. An example of four layers feed-forward network is shown in Figure 4.4 or its abbreviated representation in Figure 4.5. The equations from the inputs to the outputs are given at the bottom of the Figure 4.5.

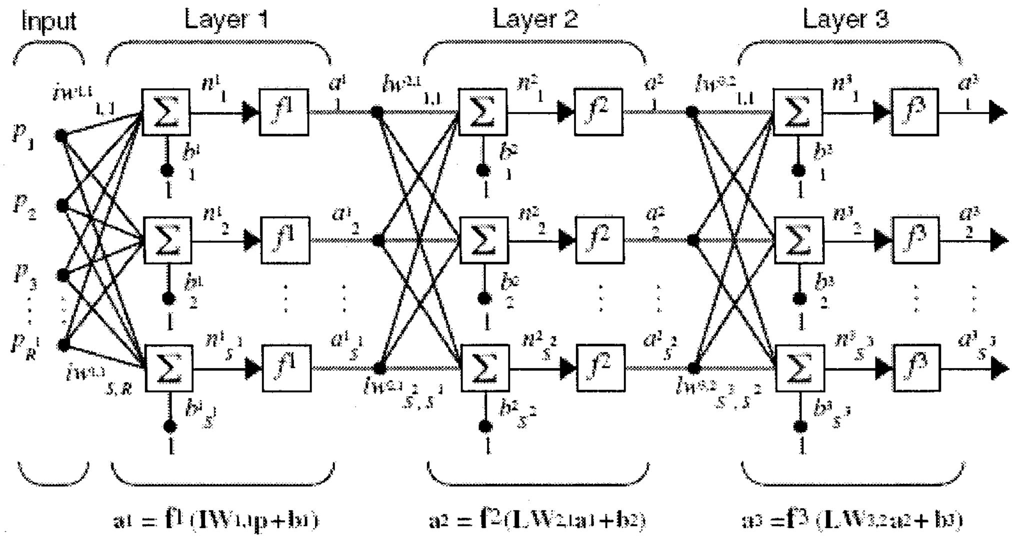


Figure 4.4 A Four Layers Feed-forward Network [59]

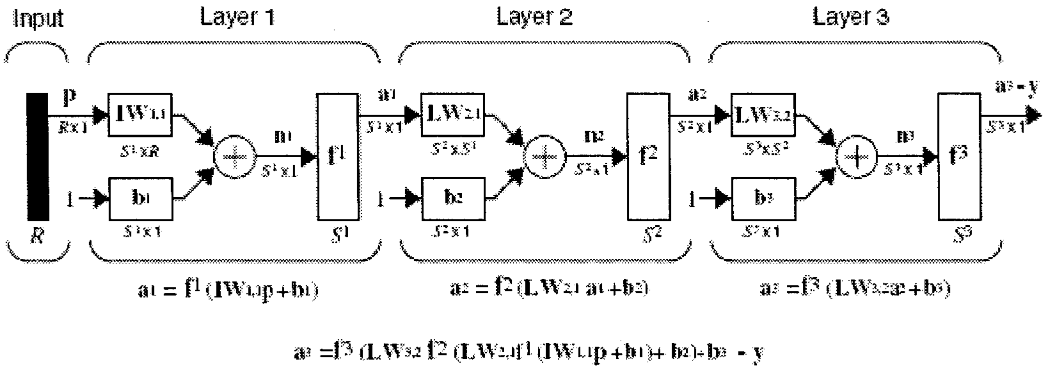


Figure 4.5 Abbreviated Notation of a Four Layers Feed-forward Network [59]

3. Recurrent networks

A recurrent network distinguishes itself from a feed-forward network in that 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 backpropagation 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. One can refer to [60] to get more information. 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. The basic work on the Hopfield network can be found in [61].

4.1.3 Network Learning

Network learning is defined as a procedure of synaptic weights change to capture the information contained in the training data. By adjusting these synaptic weights, the network can generate correct outputs when presented with different inputs. There exists many kinds of network learning (training) methods which can be found in Matlab Neural Network toolbox. Mainly, they are divided into two categories: unsupervised learning and supervised learning.

- 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 classes.

- Supervised Learning

Contracts to unsupervised learning, the learning set of supervised learning contains both input vectors and the corresponding desired output vectors as well. 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 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 defined prior [62].

The procedure of the most popular back-propagation training algorithm is specified as follows:

1) Initialize the weighs and biases. Set all the weights and biases to small random values in the interval $[-1,1]$;
2) If the stopping condition specified priori by the user is not satisfied, then proceed to steps 3 to 7, otherwise stop;
3) For each sample (input and output pair) in the training data set, follows step 4 to 6;
4) Feed-forward phase: Present the input vector $x_i = (x_1, x_2, \cdots, x_M)$ and specify the corresponding desired output $D = (d_1, d_2, \cdots, d_L)$, then calculate the network output $O = (o_1, o_2, \cdots, o_L)$. The sample could be new on each trial or samples from the training set could be presented cyclically until synaptic weights have been stabilized;

5) Back-propagation of error: The cost function E_p can be defined as the output error corresponding to each sample and expressed as:

$$E_p = \frac{1}{2}\sum_{k=1}^{L}\left(d_{pk} - o_{pk}\right)^2 \tag{4.2}$$

where o_{pk} and d_{pk} are the actual and desired outputs of the k th output neuron for p th sample, respectively. And the associated error to be propagated to the previous layers is defined as follows

$$\delta_k = \left(d_k - o_k\right) f_k'\left(Net_k\right) = o_k\left(1 - o_k\right)\left(d_k - o_k\right) \tag{4.3}$$

According to the chain rule, the weight adjustment term for each neuron in the output layer becomes:

$$\Delta_p w_{jk} = \eta o_{pk}\left(1 - o_{pk}\right)\left(d_{pk} - o_{pk}\right) o_{pj} \tag{4.4}$$

where η is a learning rate. And for each neuron in the hidden layers, the weight adjustment is

$$\Delta_p w_{ij} = \eta o_{pj}\left(1 - o_{pj}\right)\left(\sum_{k=1}^{L} \delta_{pk} w_{jk}\right) o_{pj} \tag{4.5}$$

More details about the procedure to get this adjustment can be found in [53].

6) Update and adjust weights and biases:

$$w_{jk}\left(n+1\right) = w_{jk}\left(n\right) + \Delta w_{jk} \quad \text{or} \quad w_{ij}\left(n+1\right) = w_{ij}\left(n\right) + \Delta w_{ij} \tag{4.6}$$

for the neurons in the output layer and hidden layers respectively. Or sometimes an additional momentum term is added to smooth the weights adjustment.

$$\begin{cases} w_{jk}\left(n+1\right) = w_{jk}\left(n\right) + \Delta w_{jk} + \alpha\left[w_{jk}\left(n\right) - w_{jk}\left(n-1\right)\right] \\ w_{ij}\left(n+1\right) = w_{ij}\left(n\right) + \Delta w_{ij} + \alpha\left[w_{ij}\left(n\right) - w_{ij}\left(n-1\right)\right] \end{cases} \tag{4.7}$$

where α is a momentum.

7) Stopping criterion:

Iterate the above calculations and operations by presenting new epoch of training samples to the network until the free parameters (weights) of the network stabilize their values and the average squared error computed over the entire training data set is at a minimum or acceptable small value. The orders of presentation of the training samples are randomized from epoch to epoch. The momentum and learning rate parameters are fixed or adjusted as the number of training iteration increases.

4.2 Neural Network's Applications in Fault Diagnosis

As discussed in section 4.1, neural networks have the good capability in pattern classification and function approximation. Recently, neural networks have been widely applied in system fault detection and isolation, especially in the non-linear dynamic systems [63-70]. A neural network can be used to model a MIMO non-linear system. After training, the network can give an estimated system output. Using the residual generation concept discussed in Chapter 1, the weighted difference between actual and estimated outputs is used as a residual for fault detection. When the magnitude of this residual exceeds a pre-defined threshold, the system is likely to be faulty. What is more, a second neural network can be connected to extract features from the residual for fault classification. In our thesis, for fault detection and isolation in the reaction wheel, we concentrate on modeling of the non-linear dynamic system with neural networks.

In order to model the non-linear dynamic systems, a multi-layer feed-forward networks and recurrent networks are two main classes of network to be used. Since the dynamic feedback has been introduced in the network, recurrent networks can be used to model dynamic systems. While the feed-forward network is only a static non-linear mapping between inputs and outputs, without modification, it cannot be used for dynamic system modeling. But we can combine the feed-forward network with some time delay units to enable it to represent dynamic systems. For an nth order non-linear dynamic system defined in Equation 4.8, the simplest way to model this system is to use a one step prediction model as shown in Figure 4.6.

$$y(k)=F\left(y(k-1),\cdots,y(k-n),u(k),\cdots,u(k-n)\right) \tag{4.8}$$

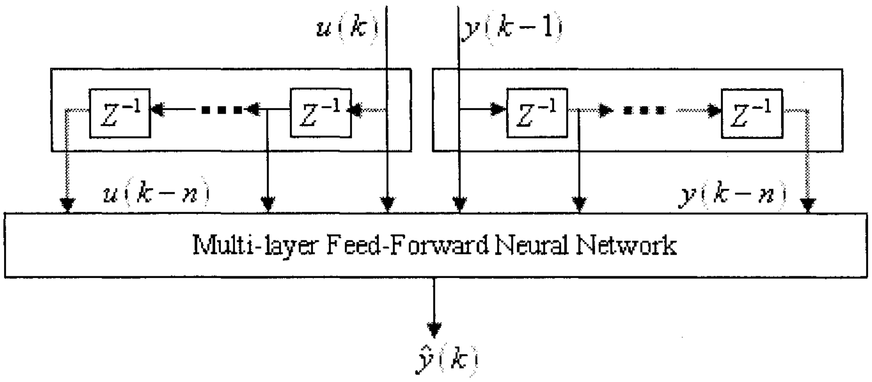


Figure 4.6 One Step Prediction Model by Using Neural Networks

As we know, the estimated output is:

$$\hat{y}(k)=NN\left(W,y(k-1),\cdots,y(k-n),u(k),\cdots,u(k-n)\right) \tag{4.9}$$

where NN represents a neural network map and W is the synaptic weights.

A neural network-based FDI scheme is shown in Figure 4.7 [63]. Here the neural network has been used as an alternative to the traditional state estimator such as Luenberger observer or a Kalman filter to estimate the system output. In the absence of faults, the residual is only due to the unmodeled noise and disturbance. When a fault comes, the residual deviates from zero in characteristic ways. In the second stage, another neural network is used to classify different fault situations according to the residual signals.

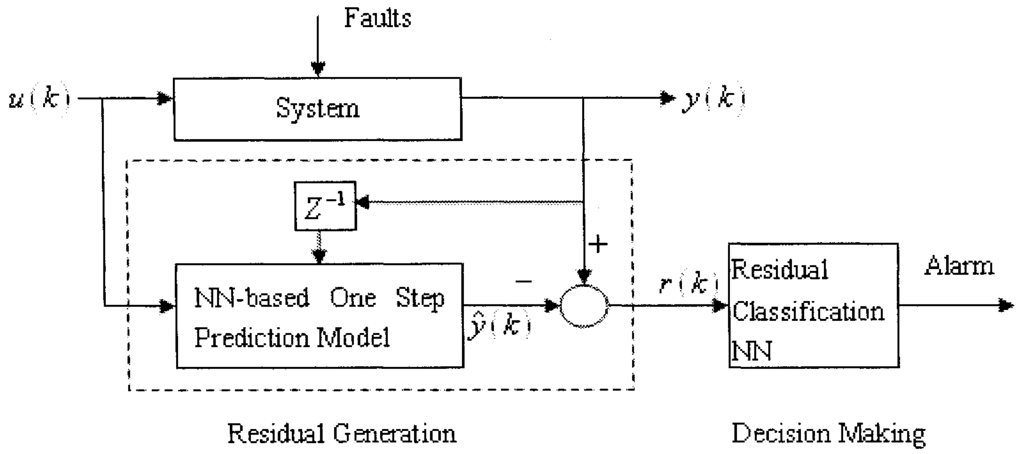


Figure 4.7 Neural Network-based FDI Scheme

4.3 Neural Network Observer-based FDI Scheme

In this section, we are going to develop a neural network observer-based scheme for fault detection and isolation in reaction wheels. Similar to the linear observer-based method in Chapter 3, three independent observers are designed separately for the wheels on the three axes to simplify design algorithm and fault isolation. The block diagram is shown in Figure 4.8.

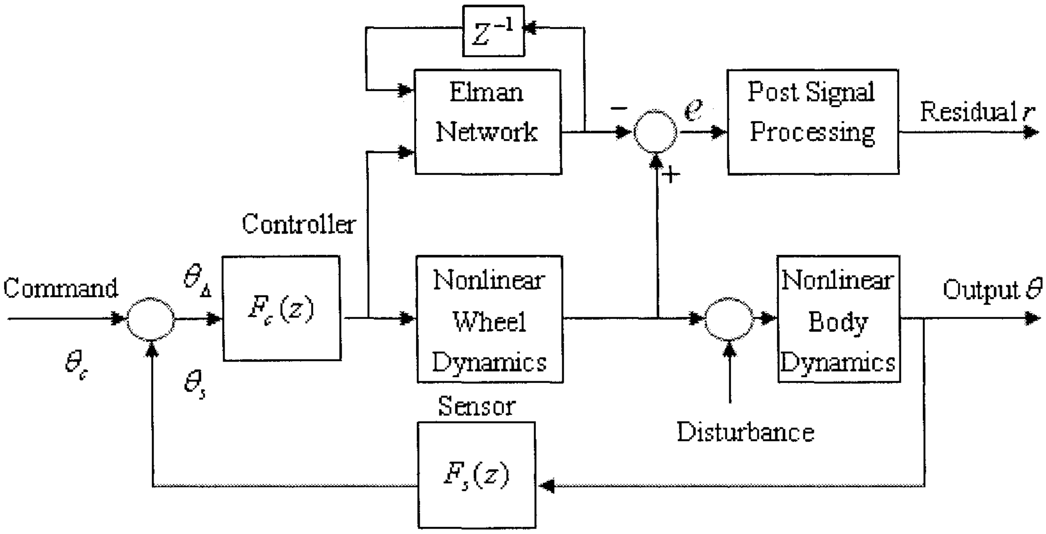


Figure 4.8 Neural Network Observer-based FDI Scheme in Reaction Wheels (Recall Stage)

What we need to mention is that another scheme is used in Figure 4.9 during the training phase of the neural network. The main difference between these two schemes is as follows: When the network is being trained to model the dynamics of the reaction wheel, the delay value introduced into the network input is the actual reaction torque. And we use the difference e between the actual torque and the estimated torque from the network to tune the network synaptic weights W to minimize the difference. When the network has been trained, that network inputs becomes the delayed value of the estimated torque from the network. In this way, when the wheel becomes faulty, the neural network still represents the wheel dynamics in fault free case and gives the estimated torque in normal case. Thus a big difference between the actual and estimated torques will occur for FDI purpose.

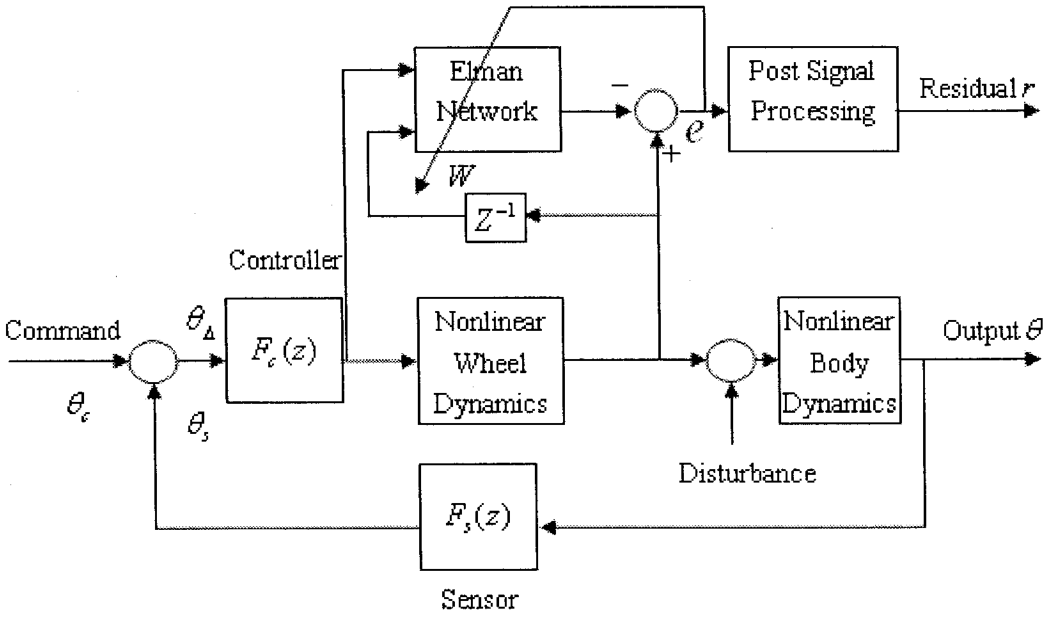


Figure 4.9 Neural Network Observer-based FDI Scheme during Network Training Phase

4.3.1 Network Architecture Selection

In order to model the dynamics of the reaction wheel, we choose the one kind of recurrent networks called Elman network [46]. The structure of one of three-layer Elman network is shown below in Figure 4.10. This network has a feedback from the output of the hidden layer to the input of the hidden layer. This recurrent connection allows the Elman network to detect and generate time-varying patterns or model dynamic functions. And the delay in the connection stores values from the previous time step, which can be used in the current time step. Obviously, more hidden neurons or layers are needed if the function being fit increases in complexity.

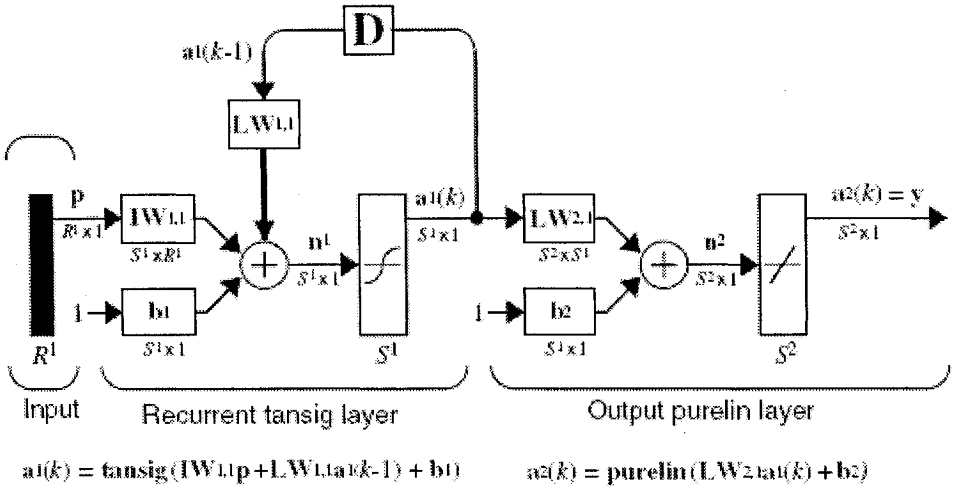


Figure 4.10 Structure of One Three-layer Elman Network [59]

In our thesis, we create a three-layer Elman network $NN_{2\times25\times1}$ to model the reaction wheel dynamics. That is the Elman network contains 2 input elements, 25 hidden neurons and 1 output. One of the input elements is the torque command voltage generated by the PID controller (has been constrained in$[-5,5]$). The other input element is the one time step delay value from the actual reaction torque signal when the network is being trained as in Figure 4.9 or from the network output when the network has been trained and used in fault detection and isolation as in Figure 4.8 and where the network output is the estimated reaction torque.

4.3.2 Network Parameters Selection

We use hyperbolic tangent sigmoid function as the activation function in the hidden layer and output layer. This kind of activation function is defined as

$$\tan sig\left(x\right)=\frac{2}{1+e^{-2x}}-1 \tag{4.10}$$

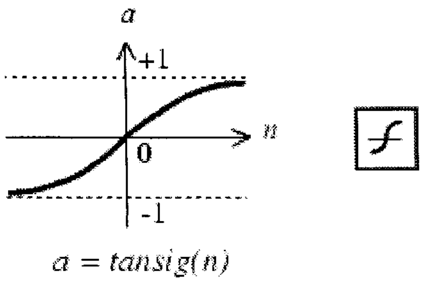


Figure 4.11 Tan-Sigmoid Activation Function [59]

The network performance function *mse* (mean square errors) is used to evaluate the network performance.

The training procedure of an Elman network is given in [56]. At each epoch, first, the entire input sequence is presented to the network, and its output is calculated and compared with the target sequence to generate an error sequence. Then for each time step, the error is backpropagated to obtain gradients of errors for each weights and biases. These gradients are used to update the synaptic weights with training function *traingdx*, the gradient descent training function with momentum and adaptive learning rate backpropagation. The weights are adjusted by the Equation 4.4, 4.5 and 4.7, which are summarized here again for convenience. This training algorithm is similar to the algorithm of epochwise Back-propagation through time [57].

$$\begin{cases} w_{jk}(n+1) = w_{jk}(n) + \Delta w_{jk} + \alpha\left[w_{jk}(n) - w_{jk}(n-1)\right] \\ w_{ij}(n+1) = w_{ij}(n) + \Delta w_{ij} + \alpha\left[w_{ij}(n) - w_{ij}(n-1)\right] \\ \Delta w_{jk} = \eta o_k(1-o_k)(d_k - o_k)o_j \\ \Delta w_{ij} = \eta o_j(1-o_j)\left(\sum_{k=1}^{L}\delta_k w_{jk}\right)o_j \end{cases}$$

where the momentum value $\alpha = 0.7$ and the original adaptive learning rate $\eta = 0.1$.

For each epoch, if performance decreases toward the goal, then the learning rate is increased by the pre-defined factor. If performance increases toward the goal, the learning rate is decreased by another pre-defined factor.

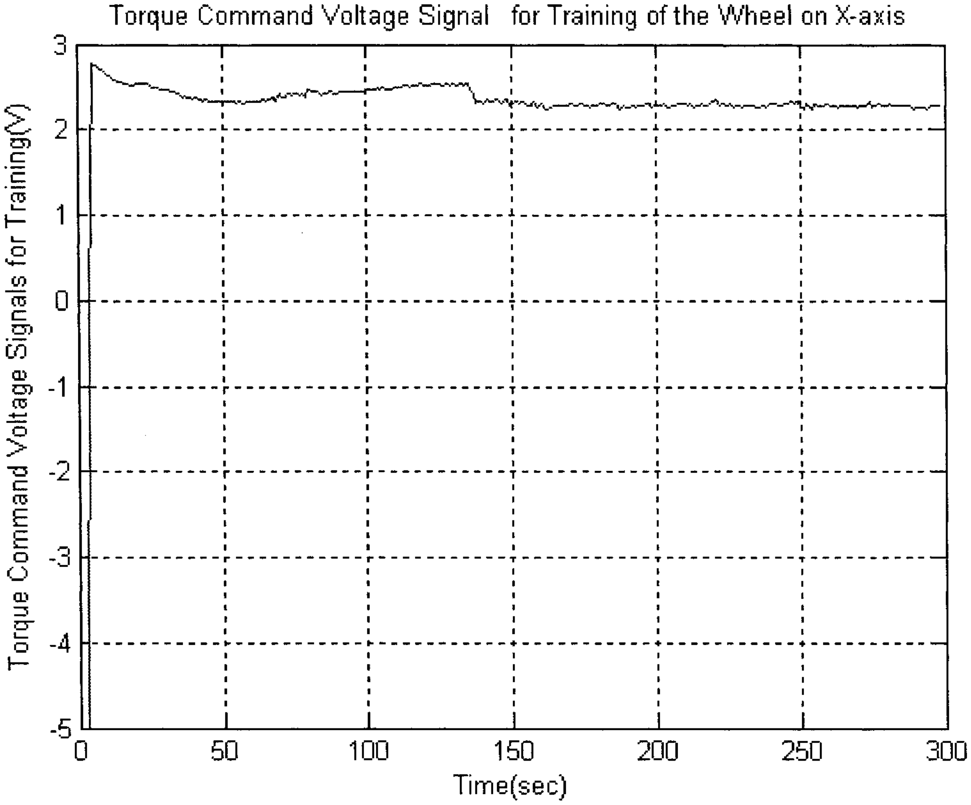
4.3.3 Procedures of the Neural Network Observer-based FDI Scheme

For this neural network observer-based fault detection and isolation scheme in reaction wheels, we are going to design three separate networks for FDI in the wheel on the three axes independently. The procedures of the scheme on each axis are very similar. Here are four main steps we need to follow for this a single axis wheel FDI:

1. Elman Back-propagation Network Training

   1) Training data collection

   As discussed above, the training data we need for the network training are the torque command voltage signal and the corresponding reaction torque signal. The torque command voltage and one time delay of corresponding reaction torque serve as the network inputs and the corresponding reaction output serves as the desired network output. First, let the spacecraft operates normally under an initial condition for the attitude control system according to the system working range specified in Table 2.3. Then we collect the information about the torque command voltage and reaction torque from the wheel since a set point change command is applied. The training data signals for the training of network on X axis are shown in Figure 4.12.



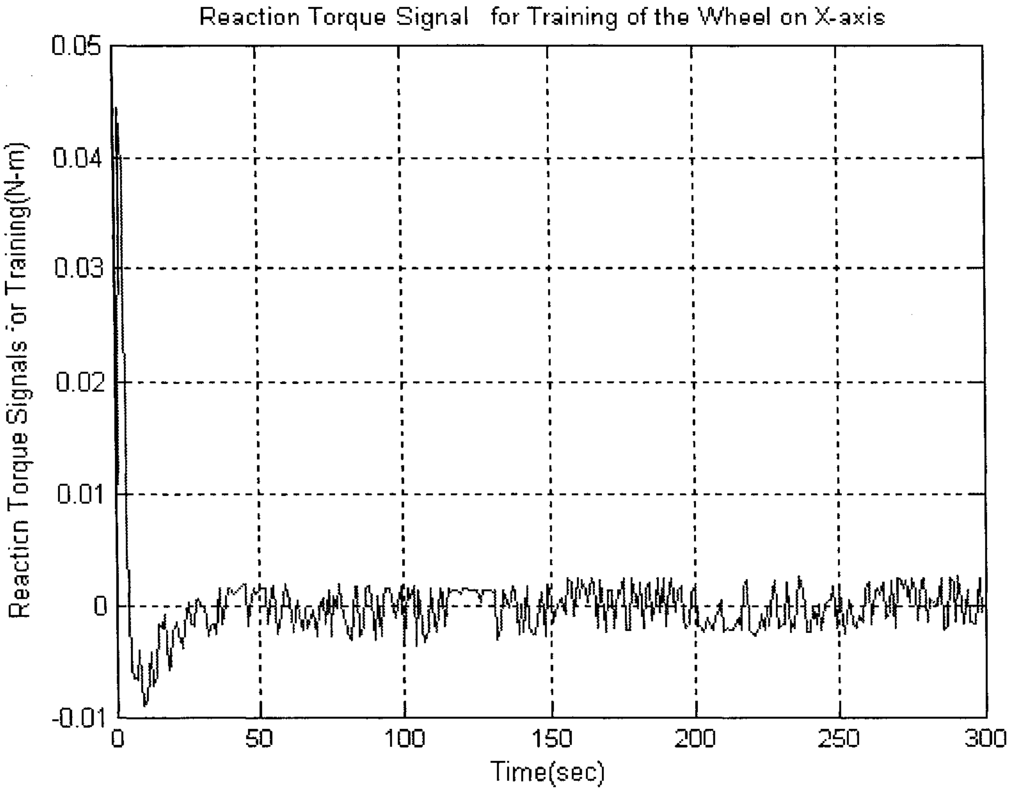


Figure 4.12 Network Training Original Data

2) Training data pre-processing

First, we sample the training data signals with 1 second as the sampling period. And since the $\tan sig$ activation function is used, from the Figure 4.11, we need to normalize these training data into $[-1,1]$ for better data representation.

3) Network Initialization

We create an Elman network $NN_{2\times25\times1}$ with the network parameters specified in section 4.3.2. Before we start the network training, we need to initialize all the weights and biases. First we randomize all the weights and biases with small values into$[-0.1,0.1]$. But in order to obscure this randomization effect of selecting initial weights and biases, we do as follows

I. Randomize all the weights and biases with small values into $[-0.1, 0.1]$;

II. Let the network training go through one epoch using these initial values and save the adjusted weights W_k and biases B_k.

III. Go through the steps I and II for 19 times more, and then average the 20 saved weights and biases correspondingly.

$$\overline{W} = \frac{1}{20}\sum_{k=1}^{20} W_k \;\; ; \;\; \overline{B} = \frac{1}{20}\sum_{k=1}^{20} B_k \, ;$$

IV. Using these $\overline{W}$ and $\overline{B}$ as the initial weights and biases for the network training in the following step.

4) Network Training

Using the initial values $\overline{W}$ and $\overline{B}$, and the gradient descent training algorithm with momentum and adaptive learning rate backpropagation as specified in section 4.3.2, we train the network for a maximum 5000 epochs to reach an error target with $mse = 1.0e^{-3}$. The network training process performance and the mean square error is shown in Figure 4.13.

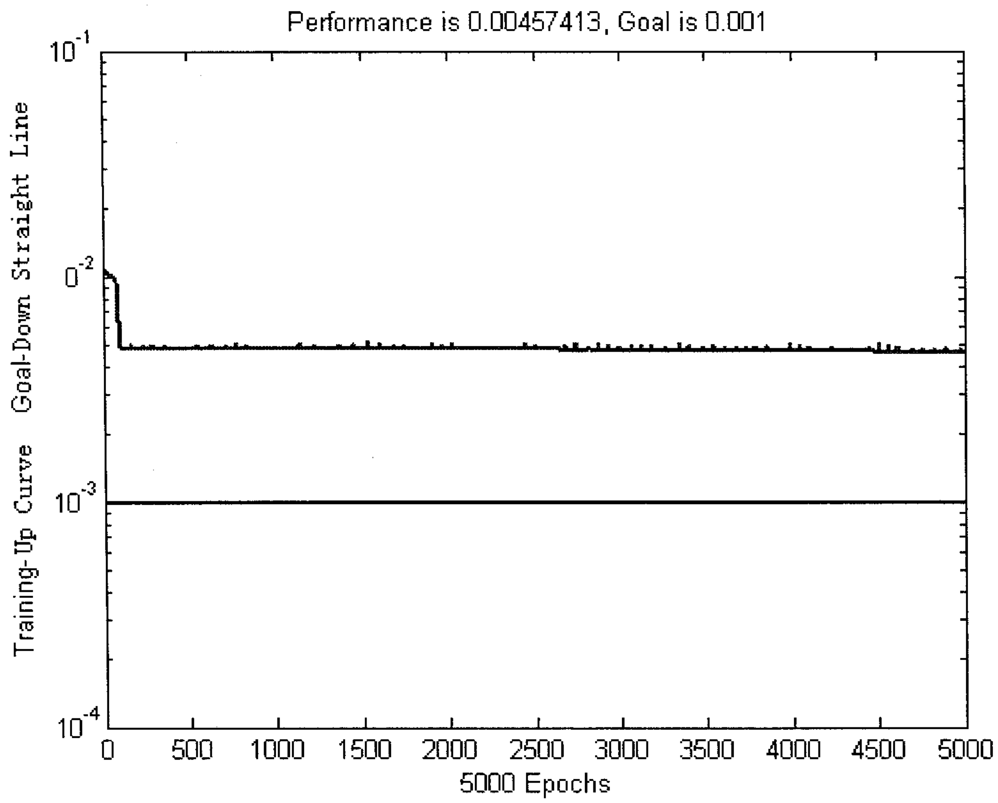


Figure 4.13 Learning Process and MSE Performance

5) Network Performance after Training

After the network has been trained, we can observe the output of the network and compare it with the desired output signal. The results are shown in Figure 4.14. As we see, the error between the actual and estimated output is around zero after a few seconds from the beginning, which demonstrates that the trained network can model the desired reaction wheel output quite well.

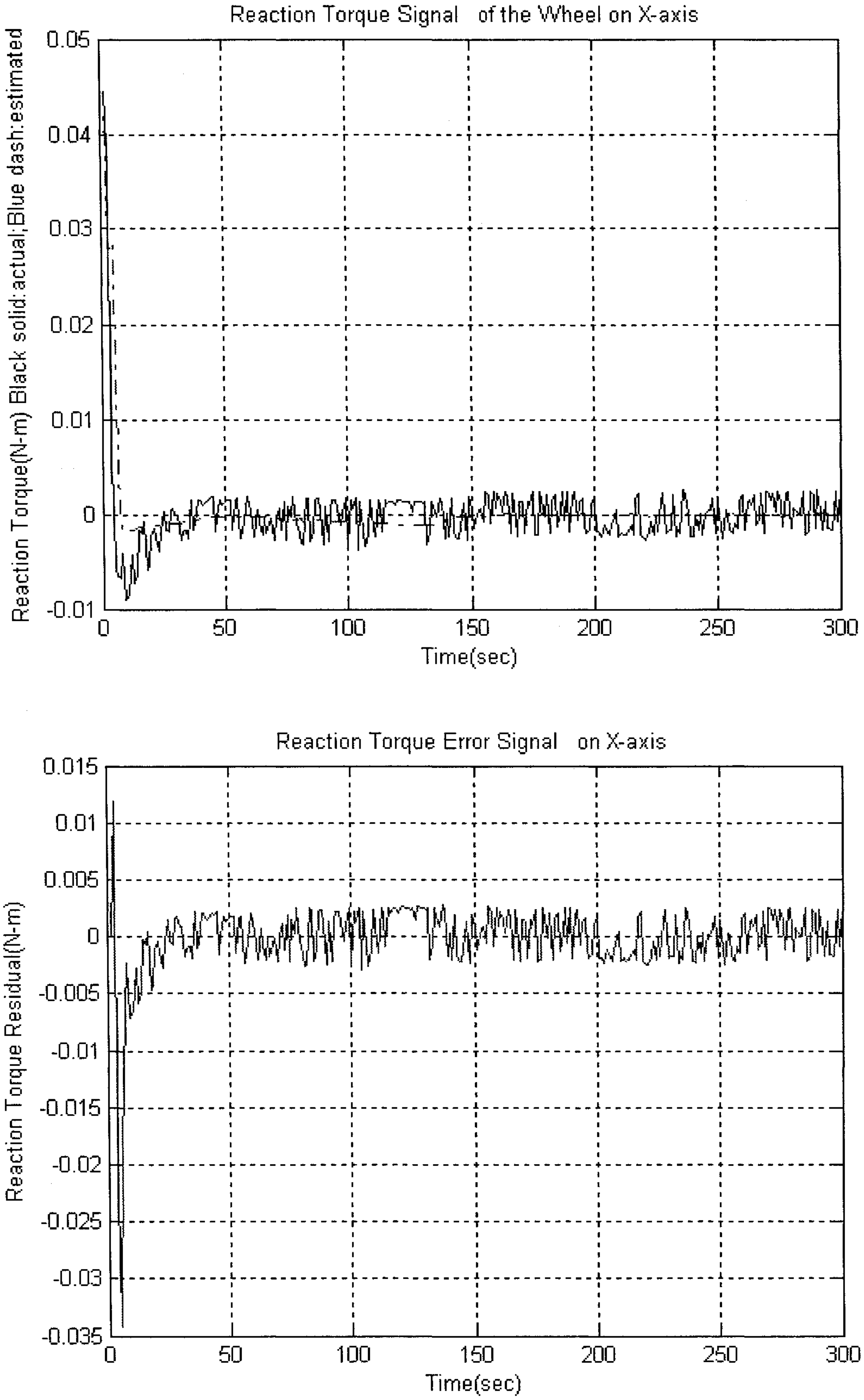


Figure 4.14 Trained Network Output Signal and Error Signal

2. Residual Generation

Based on the scheme block diagram in Figure 4.8, we used the trained Elman network to estimate the output of the reaction wheel. Then the difference or error between the actual and estimated values of reaction torque is put into a post-processing block to generate the residual for FDI purpose. First, we calculate the magnitude of the error. Second, we let it go through a moving average low pass filter to generate smooth signal. Finally, this moving average signal serves as the residual in our scheme. The window size of the moving average filter is the parameter we need to consider. The larger the window size, the smoother the reisidual signal will be, but more fault detection delay will be. So there is a compromise here. Through some experiements, we select 40 as the window size. The magnitude of error and the corresponding residual signal are shown in Figure 4.15 and Figure 4.16 for comparison.

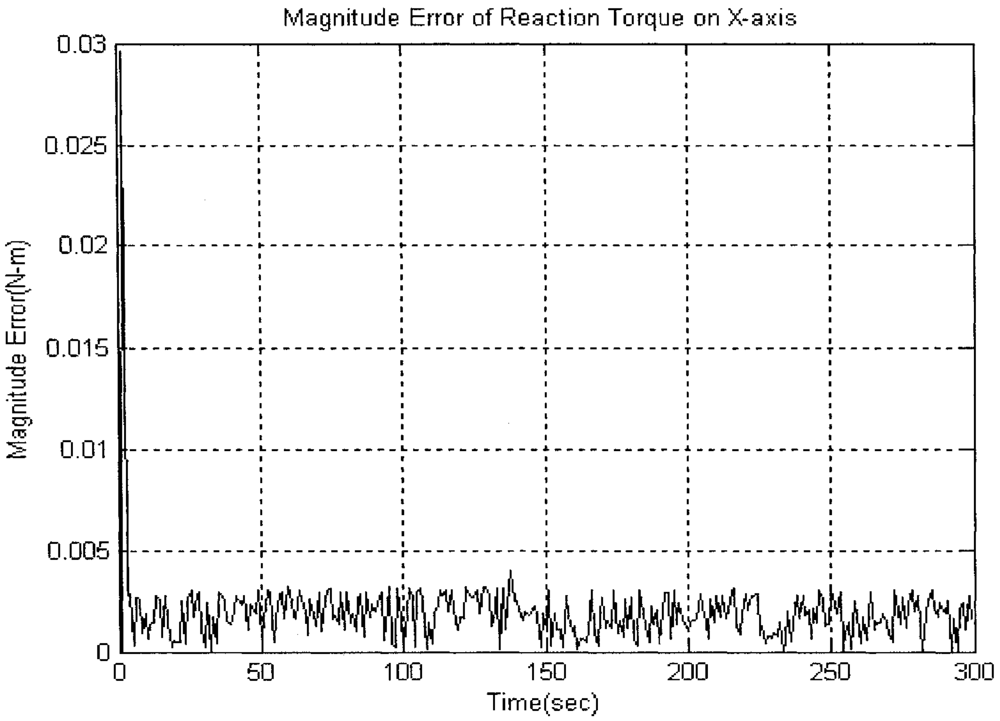


Figure 4.15 Maganitude Error of Reaction Torque on X-axis

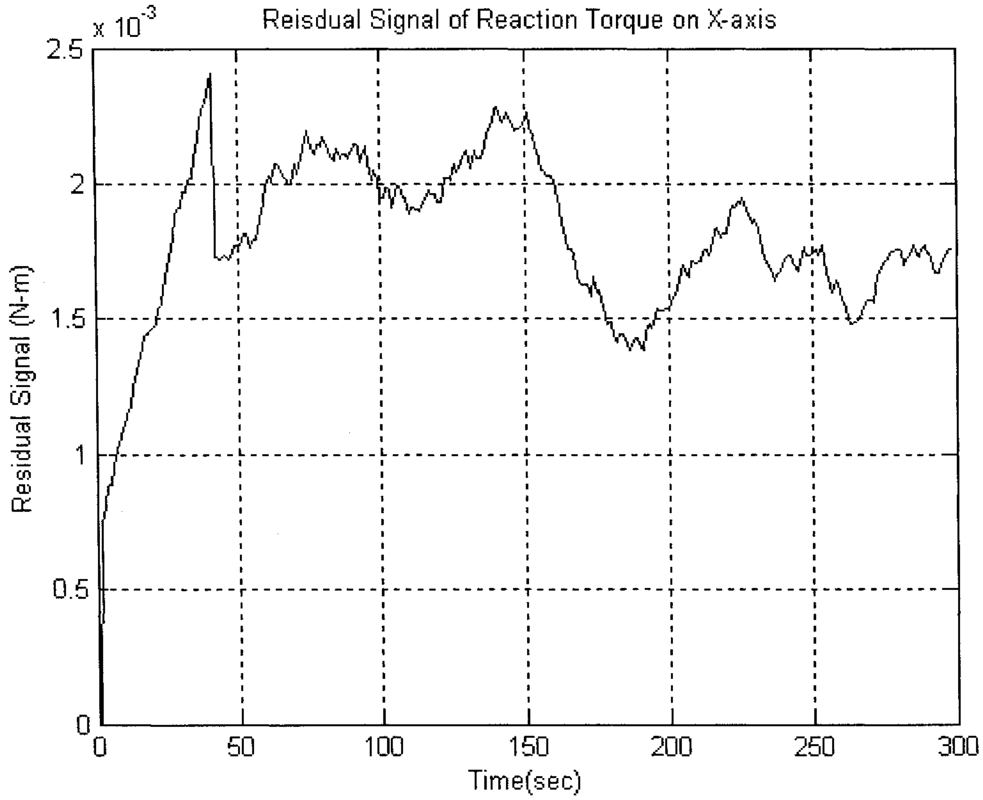


Figure 4.16 Residual Signal of Reaction Torque on X-axis

3. Threshold Curve Determination for FDI

For FDI purpose, we need threshold curve for threshold checking. This kind of threshold curve should represent the upper bound of the residual signal in all cases in normal (fault free) operating condition according to Table 2.3. To achieve this purpose, we collect 100 residual curves (an example is shown in Figure 4.16) generated in normal operation. Then we calculate the mean value $\overline{X}_{residual}$ and the standard deviation value σ of these 100 residual curves. Thus the threshold curve is defined as

$$X_{threshold} = \overline{X}_{residual} + 3\times\sigma \qquad (4.11)$$

The threshold curve for FDI on X-axis is shown in Figure 4.17.

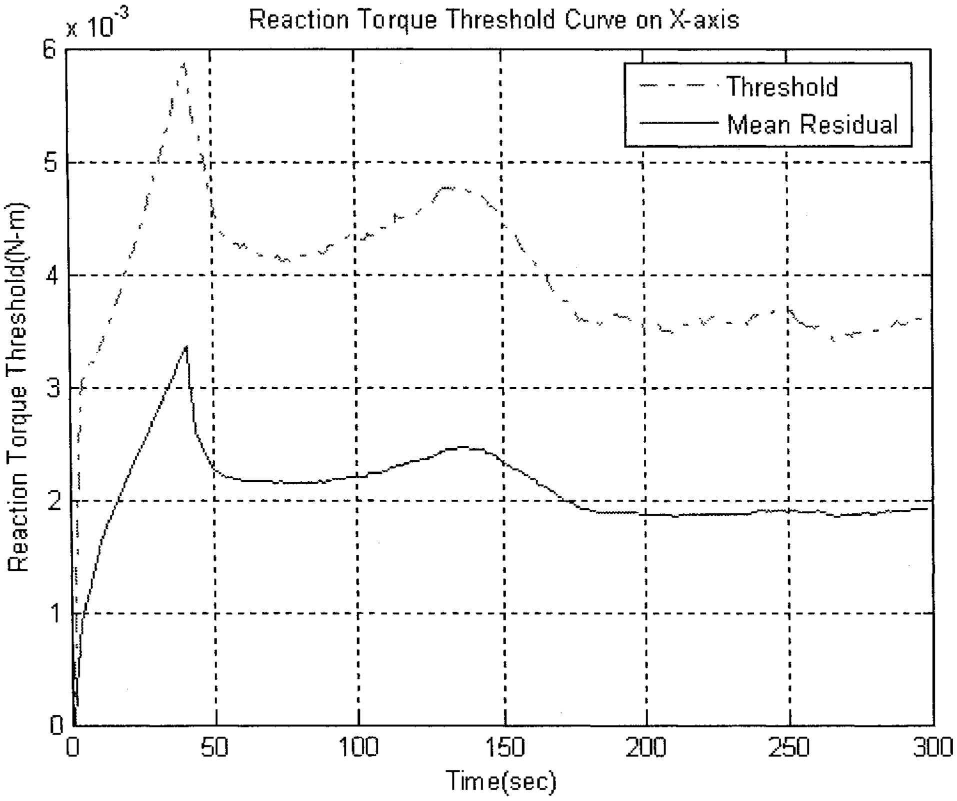


Figure 4.17 Threshold Curve for FDI on X-axis

In order to verify the reliability of this threshold curve, we check every curve of the original 100 residual curves with the threshold curve. We consider it correct only if all points on one residual curve are beneath the threshold curve. Results are shown in Table 4.1 for the threshold curves on all theree axes. From this table, we can see these threshold curves are quite reliable.

Threshold Curve	$X-Axis$	$Y-Axis$	$Z-Axis$
Reliability	97%	100%	96%

Table 4.1 Threshold Curves' Reliability Test in Normal Operation

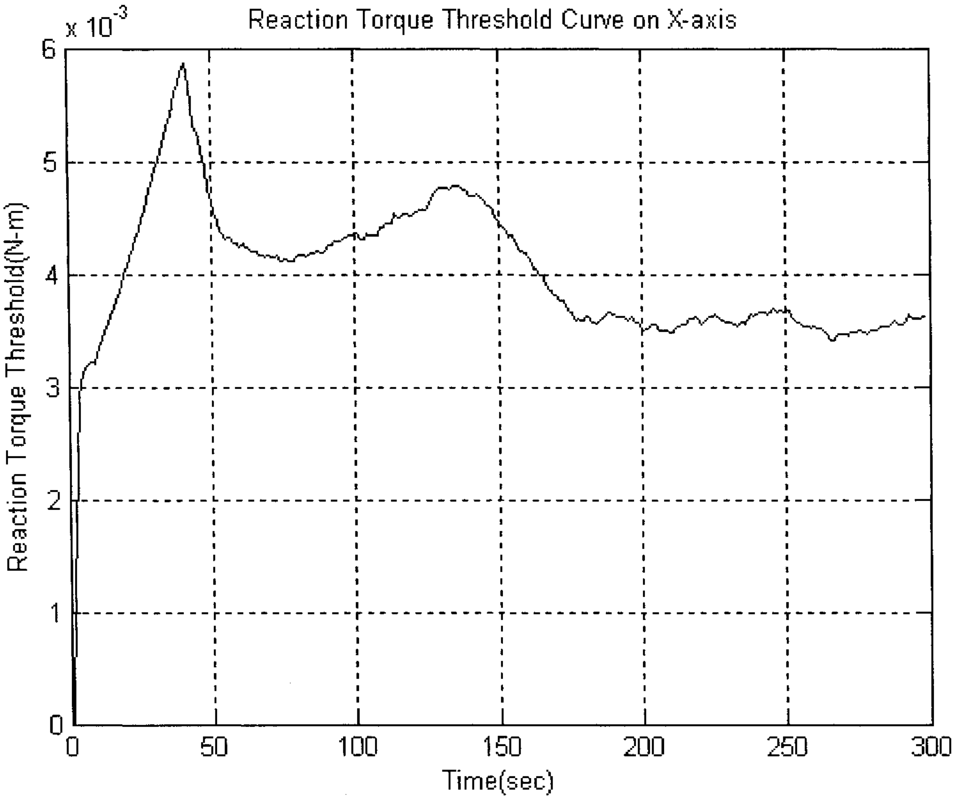
4. Fault Detection and Isolation by Using the Threshold Curves

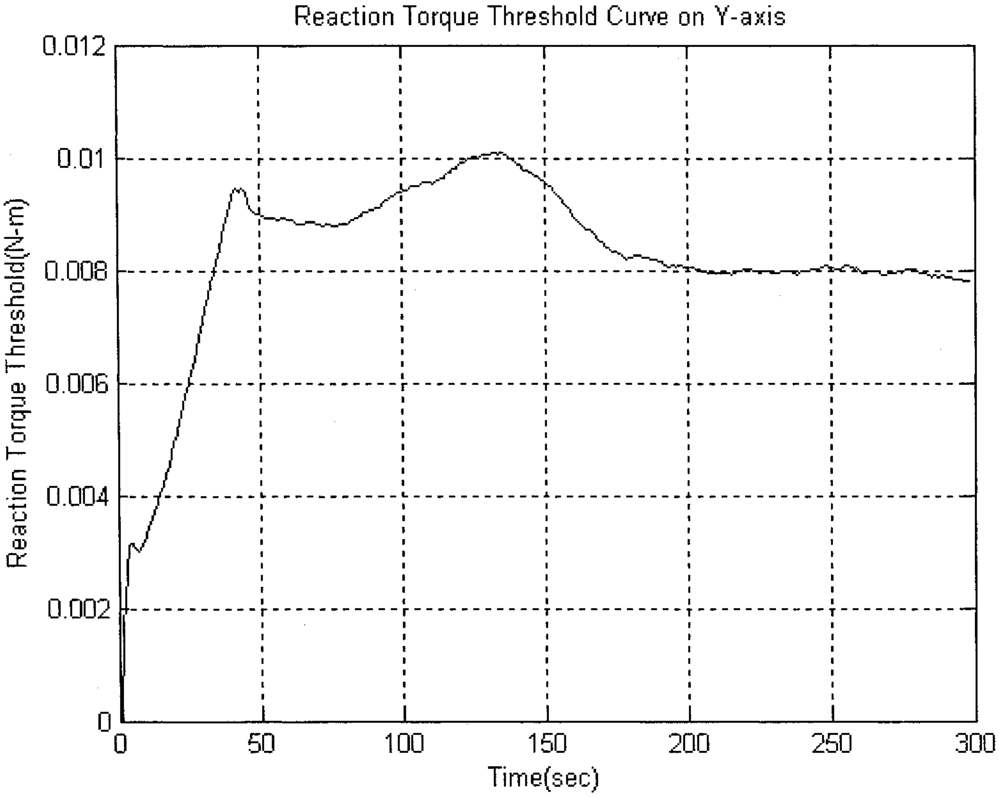
If any of the three residual curves exceeds its corresponding threshold curve, it indicates the reaction wheel on that axis is faulty. Since these three neural networks are designed separately and independenly, the fault happenning in one wheel should not affect the residual curves on the other two axes. That means we can detect and isolate the fault in the reaction wheels at the same time by the threshold checking.

4.4 Simulation Results of Neural Network Observer-based Scheme for Fault Detection and Isolation in Reaction Wheels

4.4.1 Threshold Curves Determination

Through the method specified in subsection 4.3.3, we can determine the three threshold curves as shown in Figure 4.18.


Reaction Torque Threshold Curve on X-axis
x 10^-3
6
5
4
3
2
1
0
Reaction Torque Threshold(N-m)
0
50
100
150
200
250
300
Time(sec)


Reaction Torque Threshold Curve on Y-axis
0.012
0.01
0.008
0.006
0.004
0.002
0
Reaction Torque Threshold(N-m)
0
50
100
150
200
250
300
Time(sec)

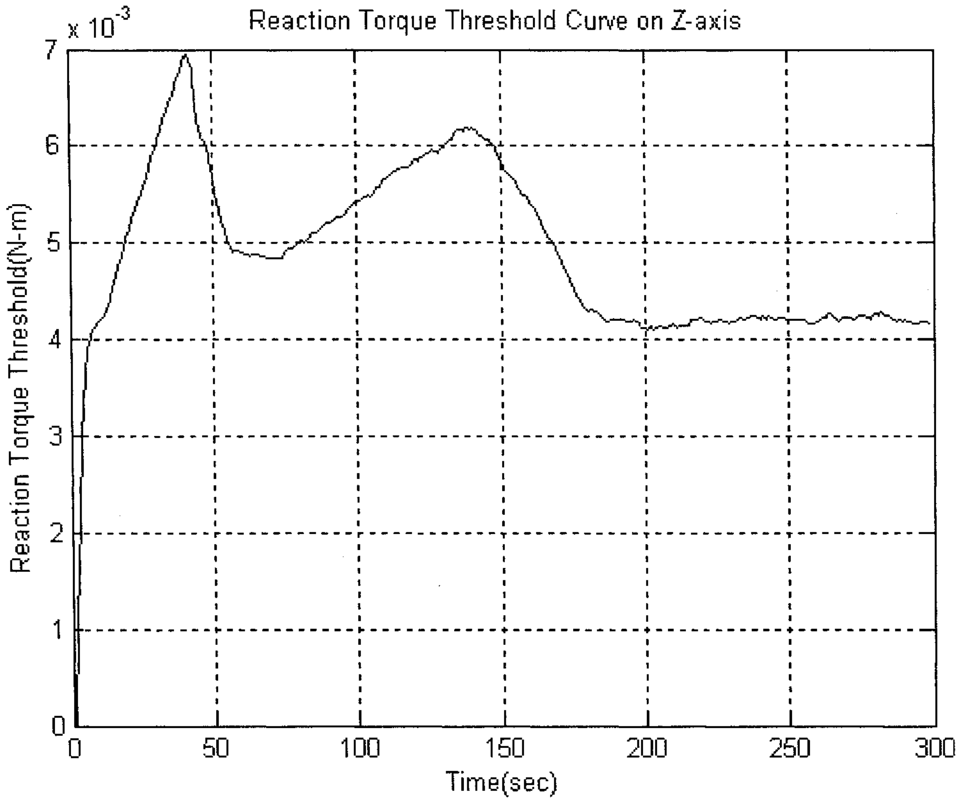
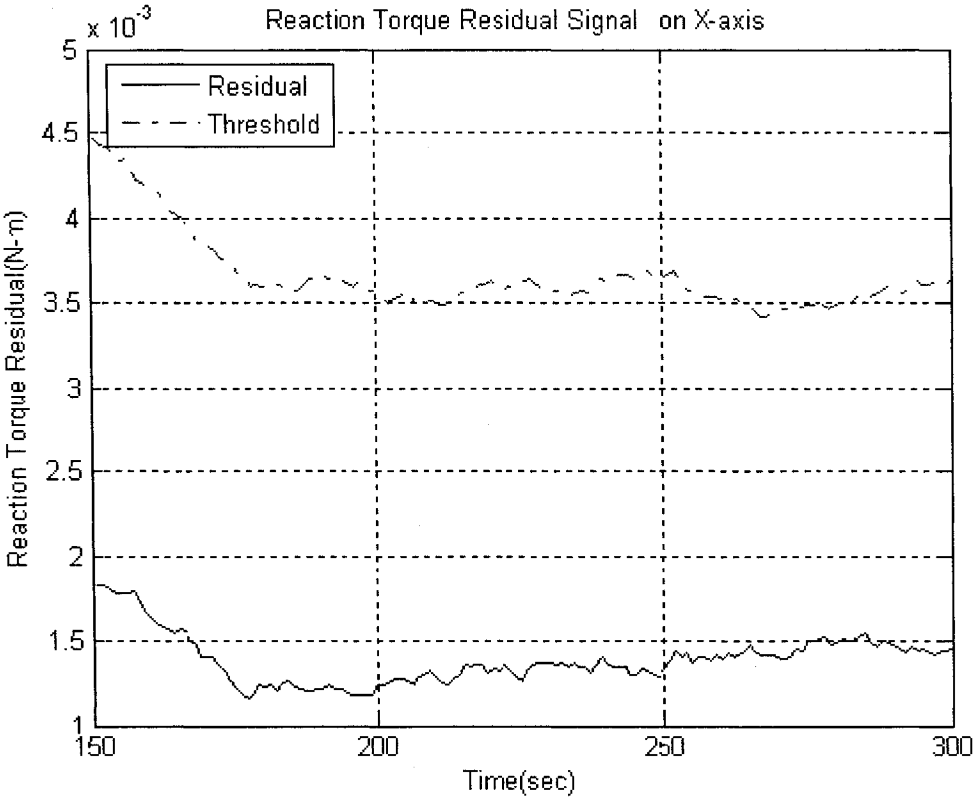
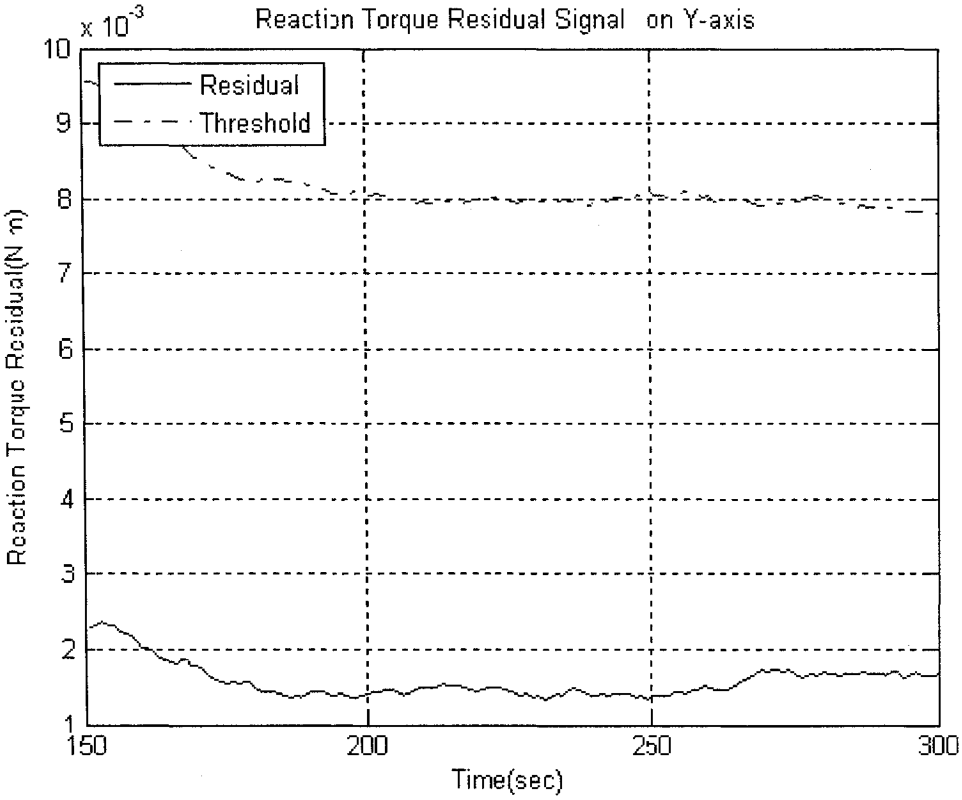


Figure 4.18 Three Threshold Curves of NN Observer-based FDI Scheme

4.4.2 Fault Detection and Isolation Performance in Fault Free Cases

For the cases in normal operation, the fault detection scheme can make the correct decision, which can be seen in following fault free case detection and isolation results. Every one of the three residual curves is beneath its corresponding threshold curve. We can conclude that no fault has happened in the wheels based on these results. No false detection has been obtained.


Reaction Torque Residual Signal on X-axis
x 10^-3
Residual
Threshold
Reaction Torque Residual(N-m)
5
4.5
4
3.5
3
2.5
2
1.5
1
150
200
250
300
Time(sec)


Reactiɔn Torque Residual Signal on Y-axis
x 10^-3
Residual
Threshold
Reaction Torque Residual(N-m)
10
9
8
7
6
5
4
3
2
1
150
200
250
300
Time(sec)

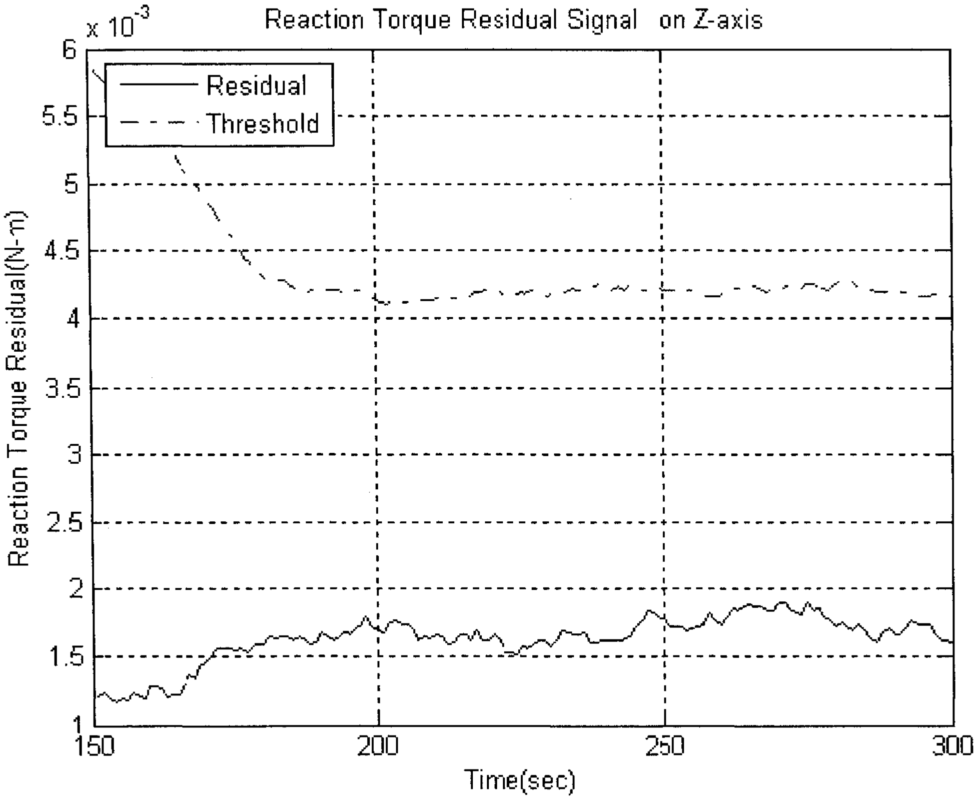
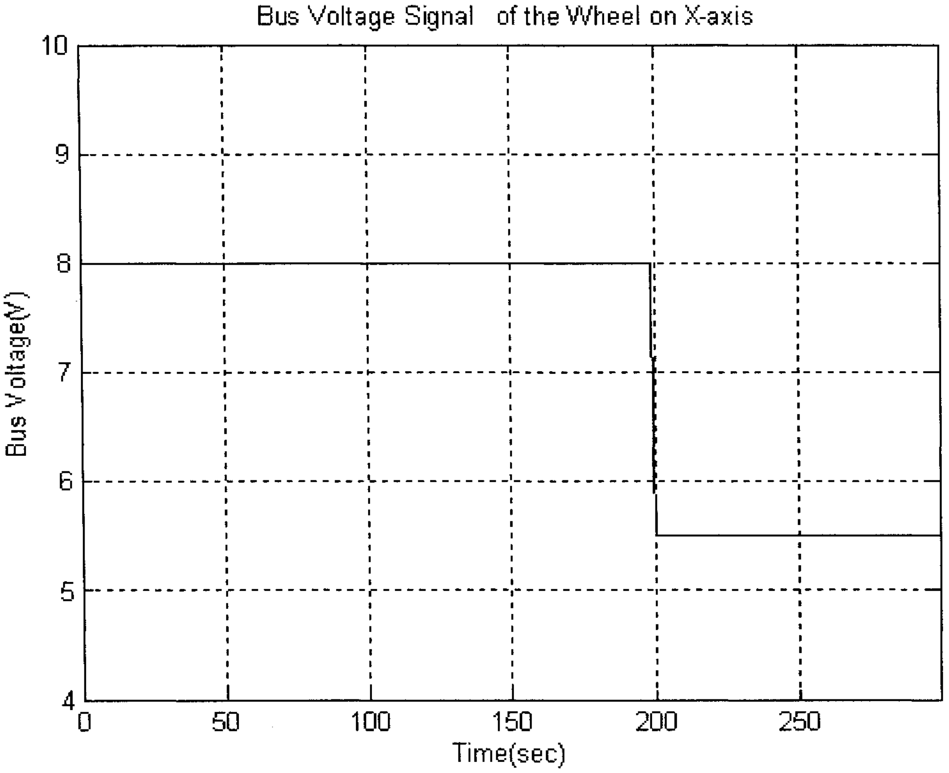


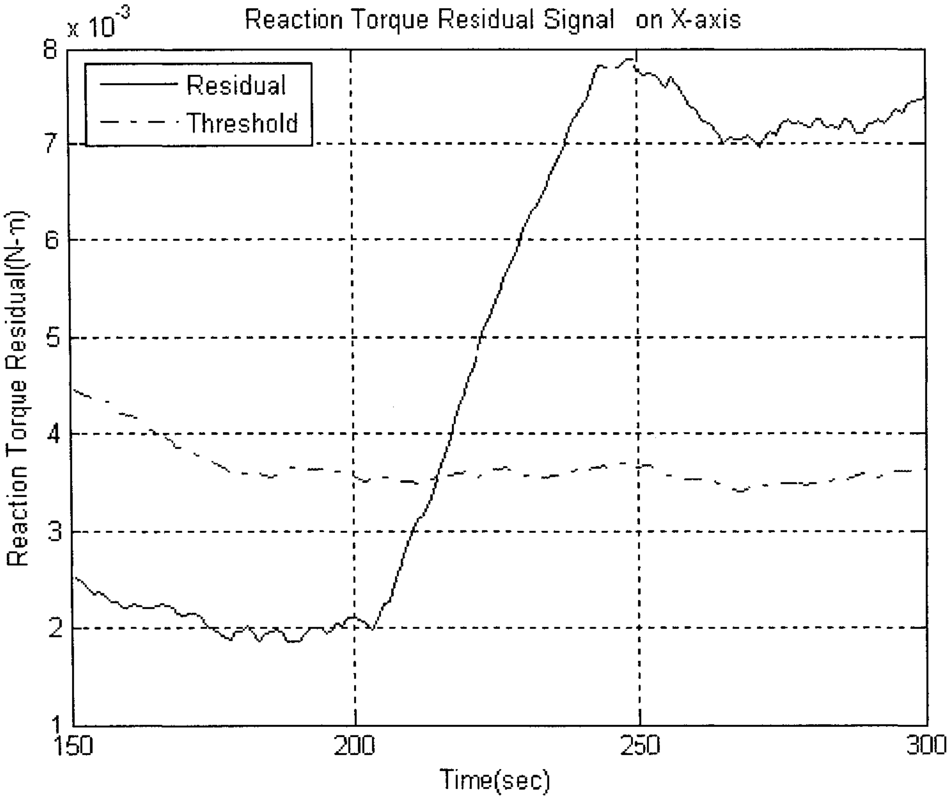
Figure 4.19 Fault Detection and Isolation Performance in Fault-free Case

4.4.3 Fault Detection and Isolation Performance in Faulty Cases

Bus Voltage Faults

Figure 4.20 shows a case study of bus voltage fault detection and isolation. As we see, the bus voltage of the reaction wheel aligned on X axis dropped from the normal value $8V$ to $5.5V$ at 200 seconds after a set point change command was applied. The residual curve of the wheel on X axis exceeded its corresponding threshold curve shortly after the fault happened. At the same time, the residual signals of the wheels on other two axes are still beneath the related threshold curves and are not affected by the fault in the wheel on X axis. So the bus voltage fault in the wheel of X axis is detected and isolated correctly.





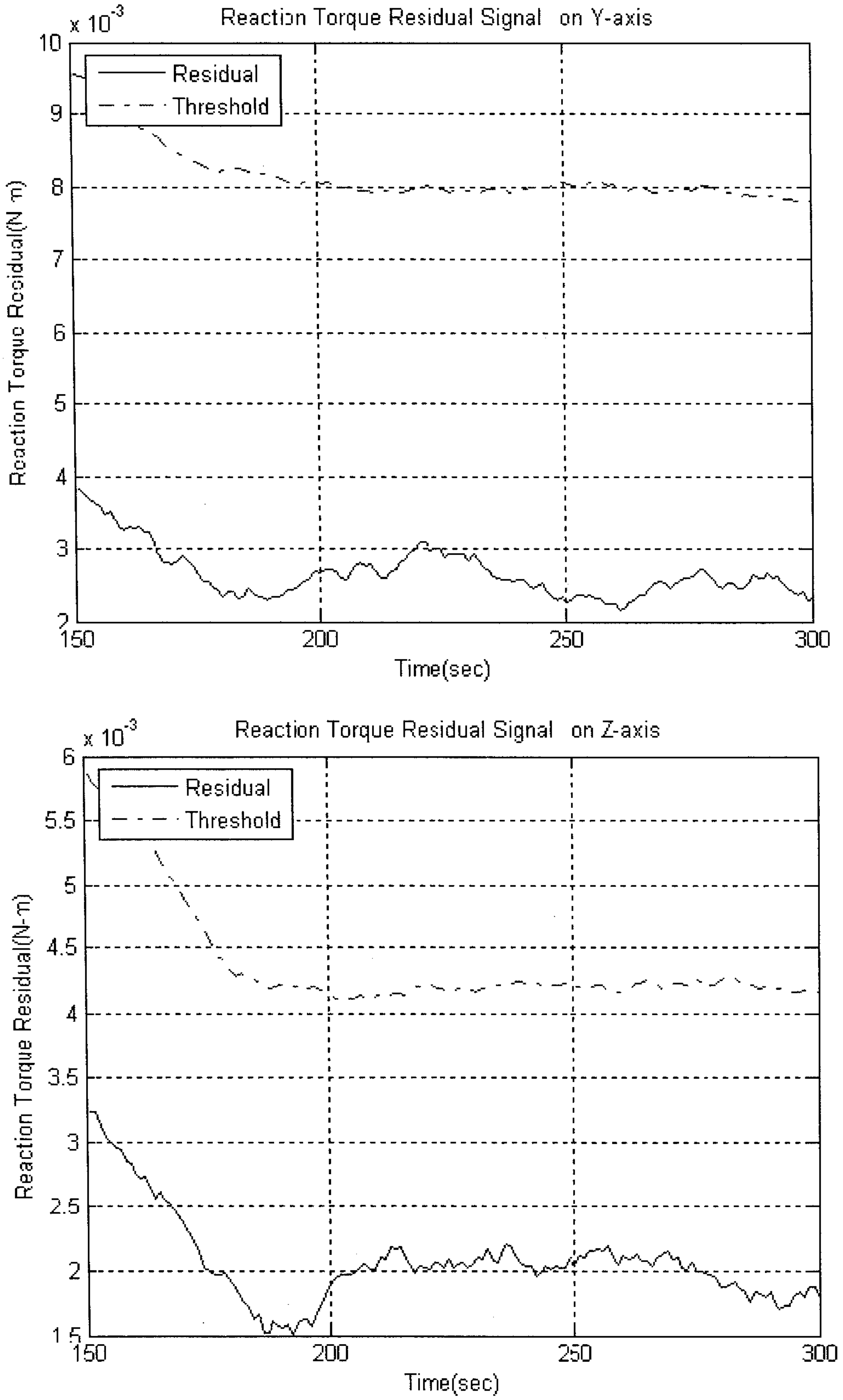
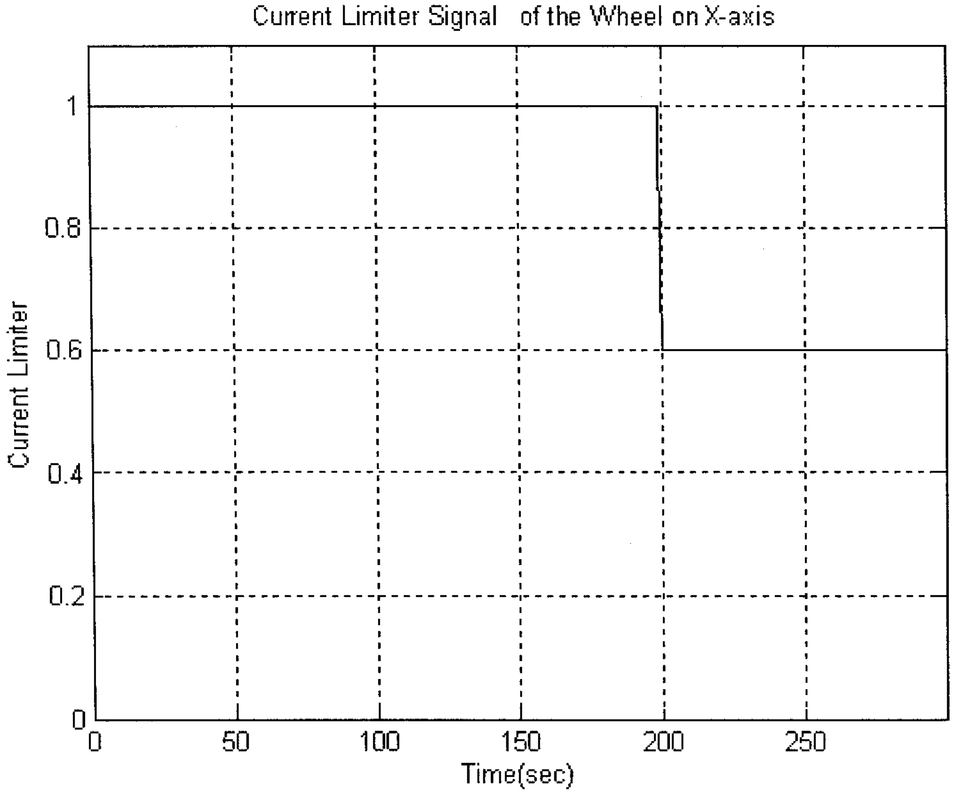
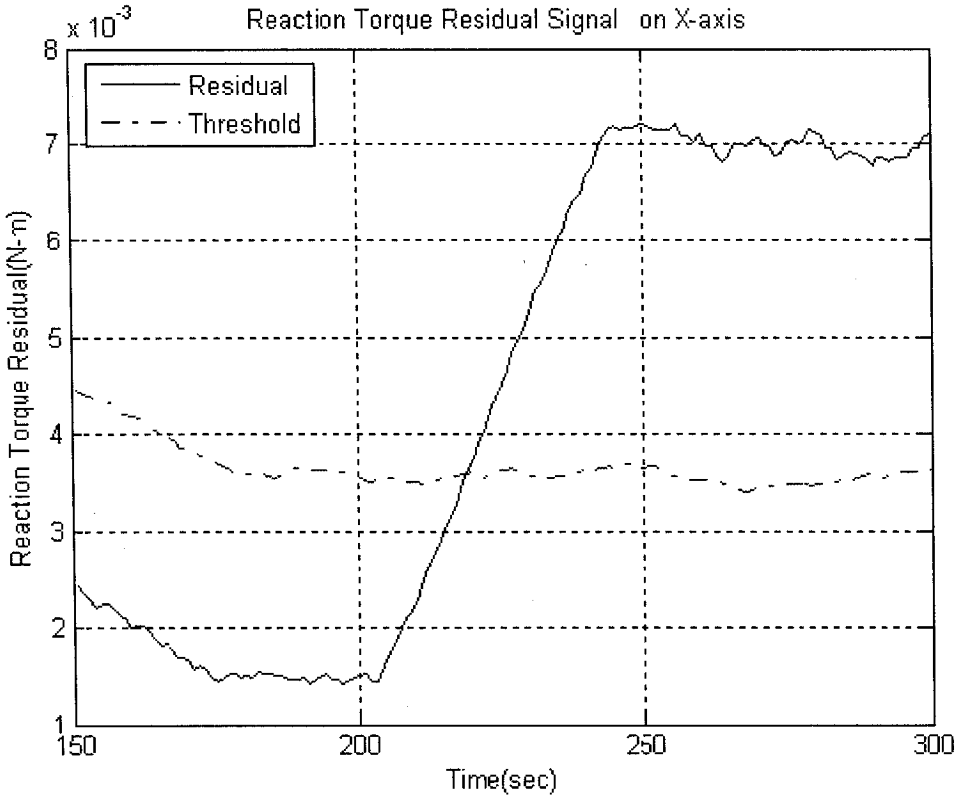


Figure 4.20 Bus Voltage Fault Detection and Isolation Case

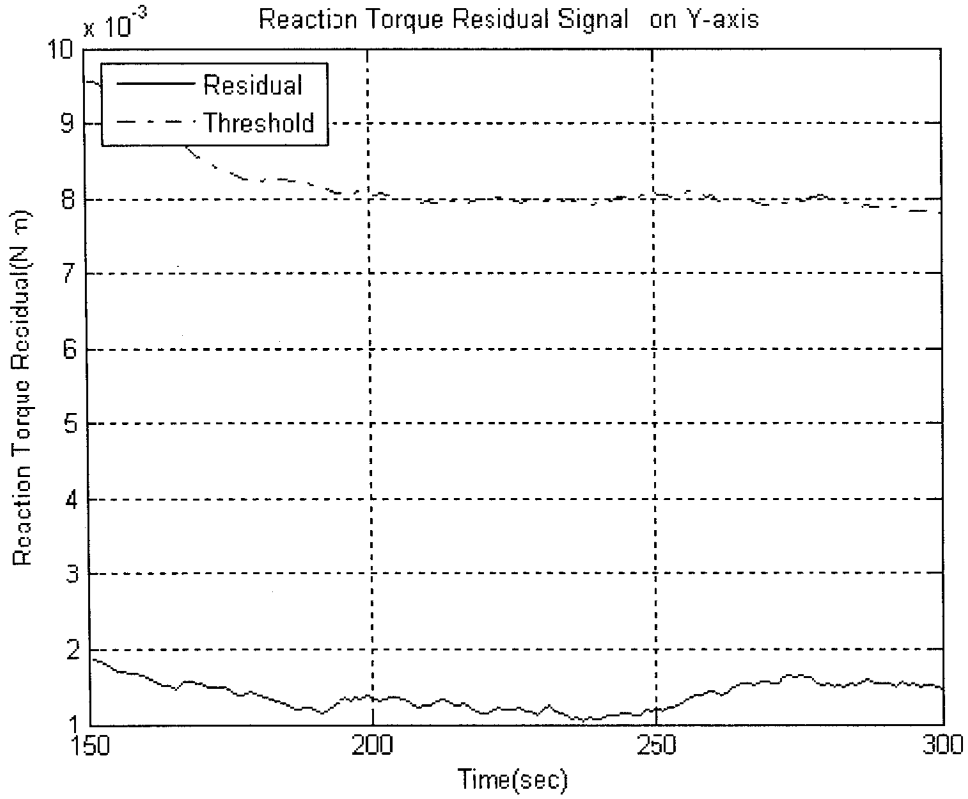
Current Loss Faults

Similarly, a current loss fault in the wheel of X axis is properly detected and isolated as shown in Figure 4.21. The attitude of spacecraft was changed according to a set point change command at zero second. And the current limiter signals of the wheel on X axis is dropped from 1 to 0.60 at 200 seconds, which means the 40% of motor current was lost. Accordingly, the residual signal of X axis increased above the threshold curve after some time delay and indicated that the wheel on X axis is faulty.





x 10^-3
Reaction Torque Residual Signal on X-axis
Residual
Threshold
Reaction Torque Residual(N-m)
8
7
6
5
4
3
2
1
150
200
250
300
Time(sec)



x 10^-3
Reaction Torque Residual Signal on Y-axis
Residual
Threshold
Reaction Torque Residual(N-m)
10
9
8
7
6
5
4
3
2
1
150
200
250
300
Time(sec)

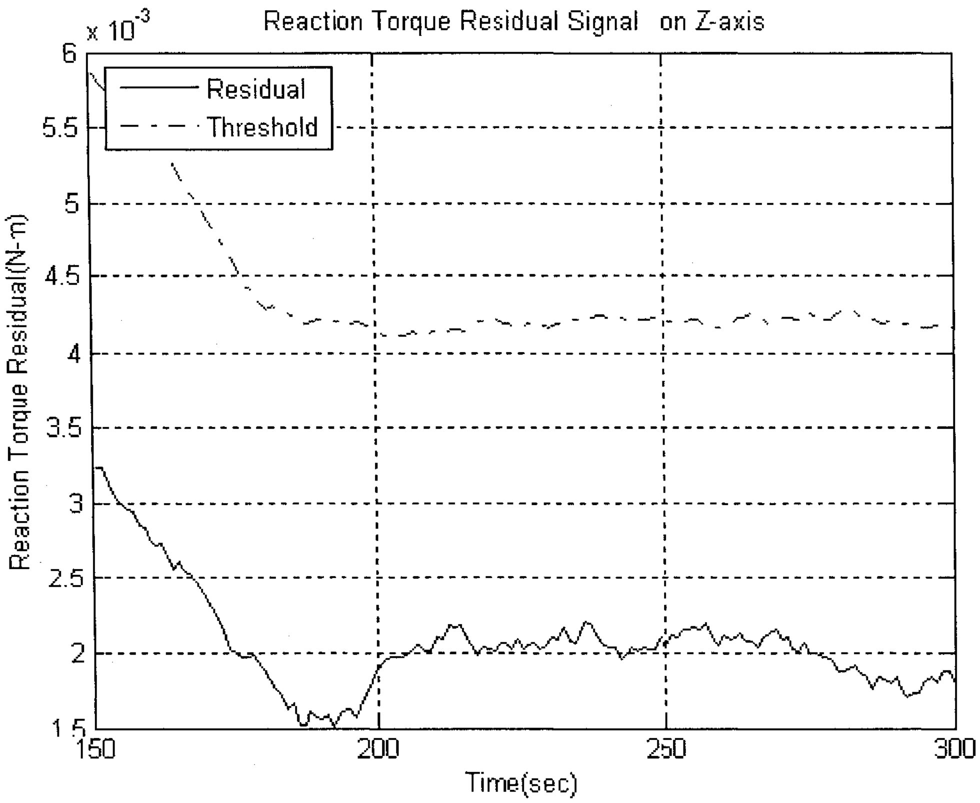
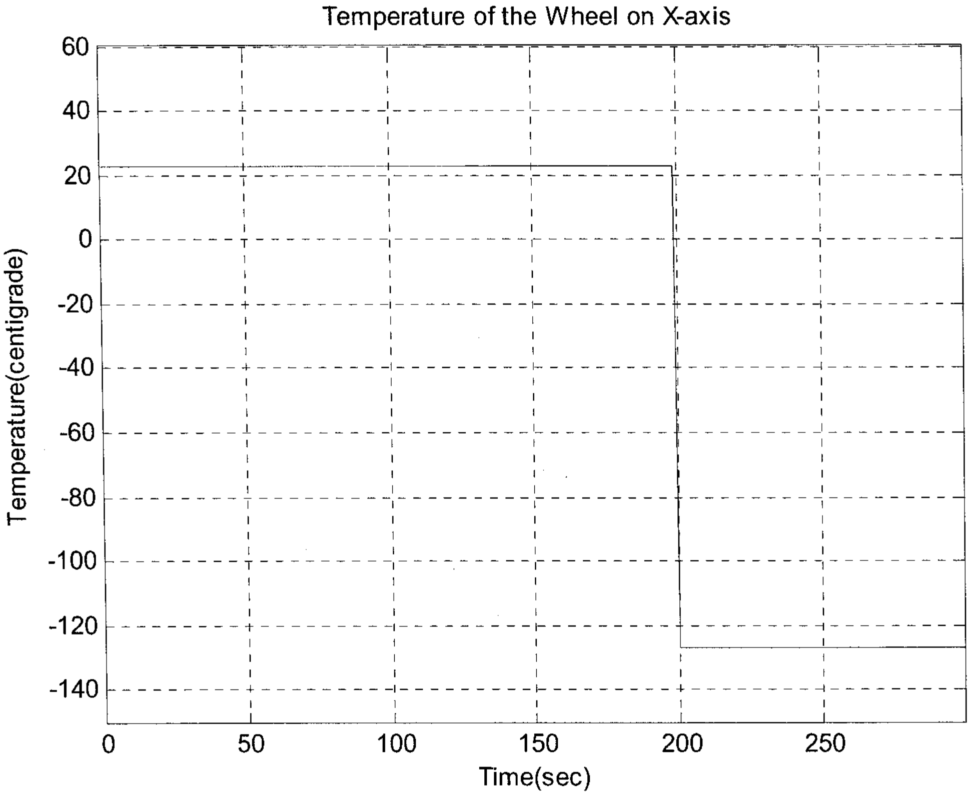
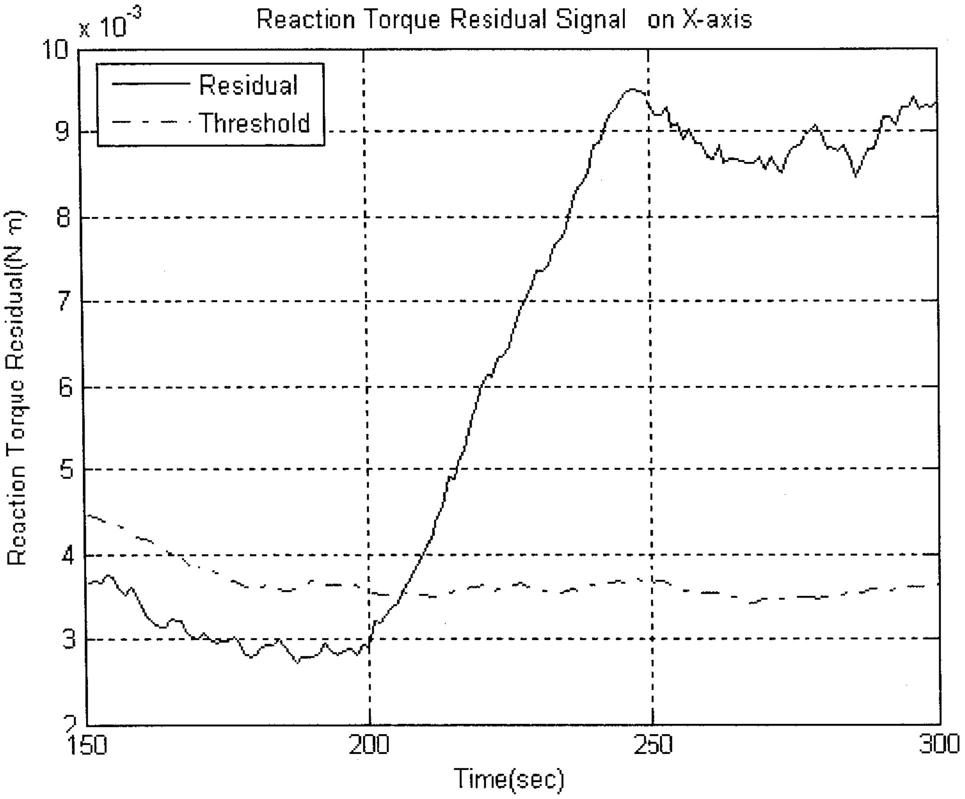


Figure 4.21 Current Loss Fault Detection and Isolation Case

Temperature Faults

The simulation results in Figure 4.22 prove that this neural network observer-based scheme is also effective in detecting and isolating a temperature fault happening in the wheel on X axis at 200 seconds.


Temperature of the Wheel on X-axis
60
40
20
0
-20
-40
-60
-80
-100
-120
-140
Temperature(centigrade)
0
50
100
150
200
250
Time(sec)


x 10^-3
Reaction Torque Residual Signal on X-axis
Residual
Threshold
10
9
8
7
6
5
4
3
2
Reaction Torque Residual(N m)
150
200
250
300
Time(sec)

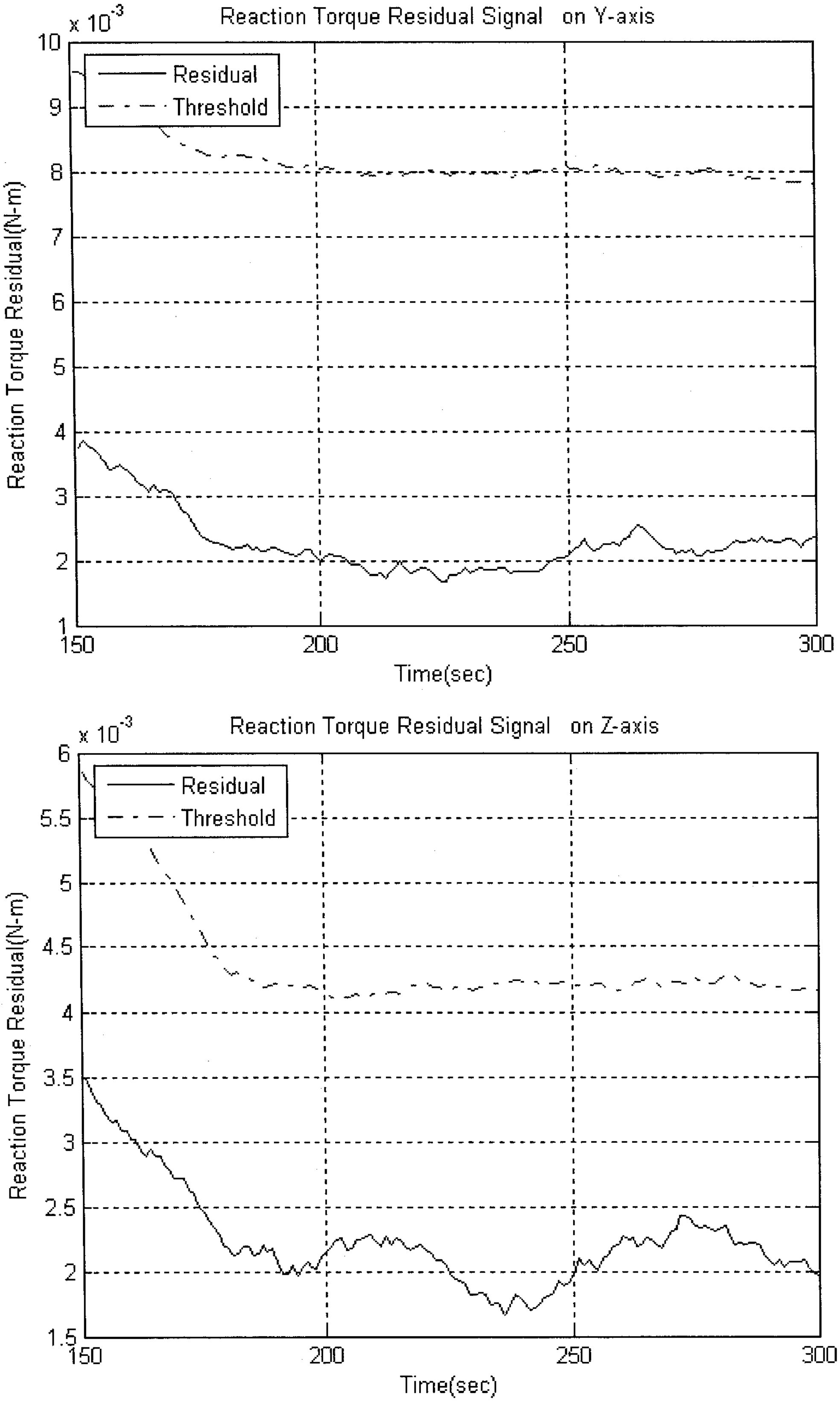
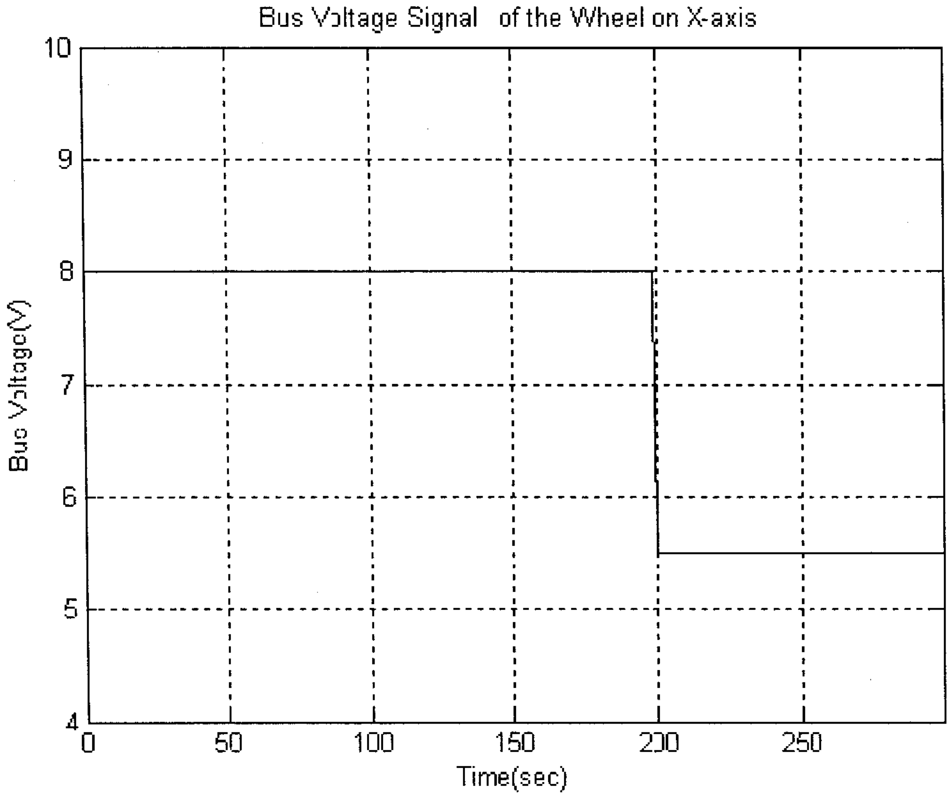
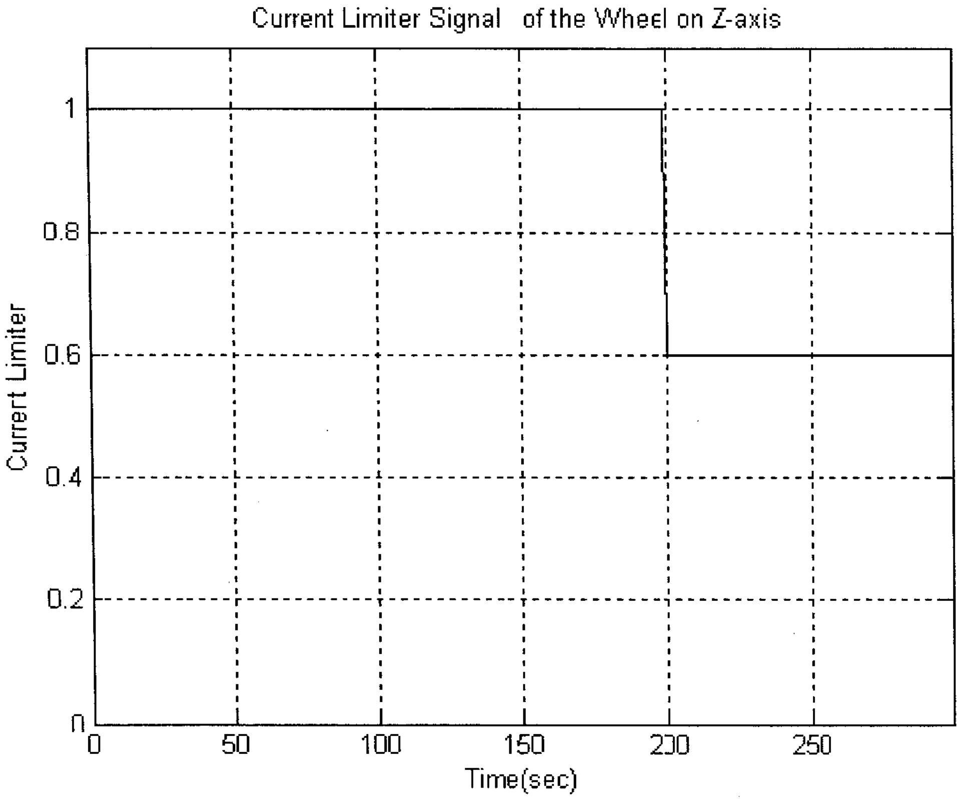


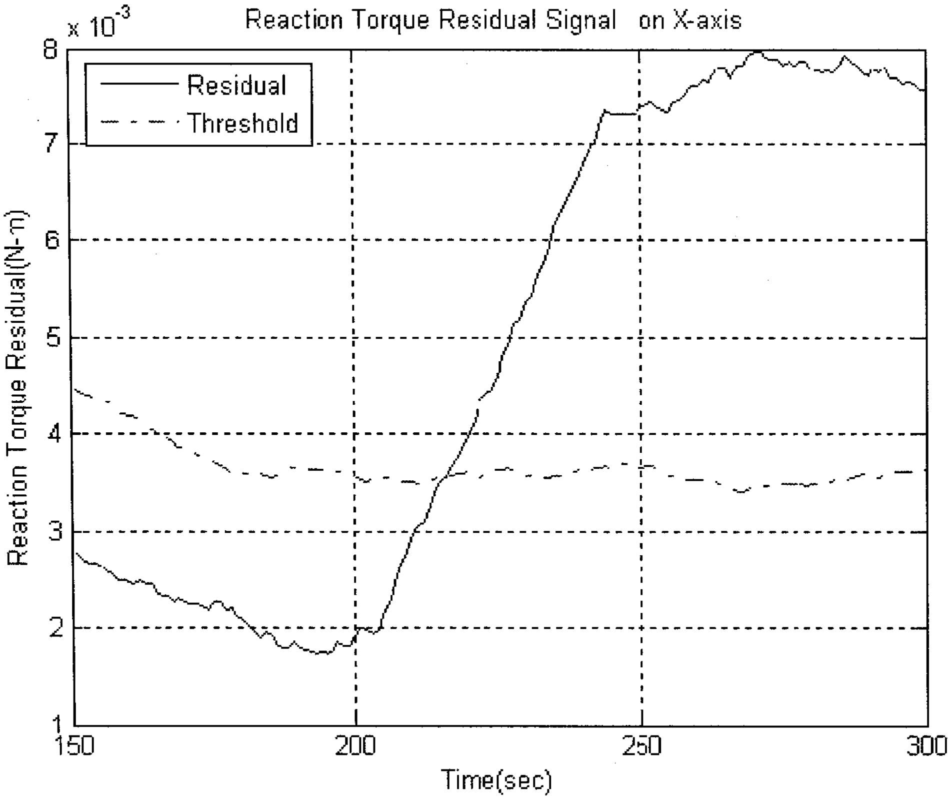
Figure 4.22 Temperature Fault Detection and Isolation Case

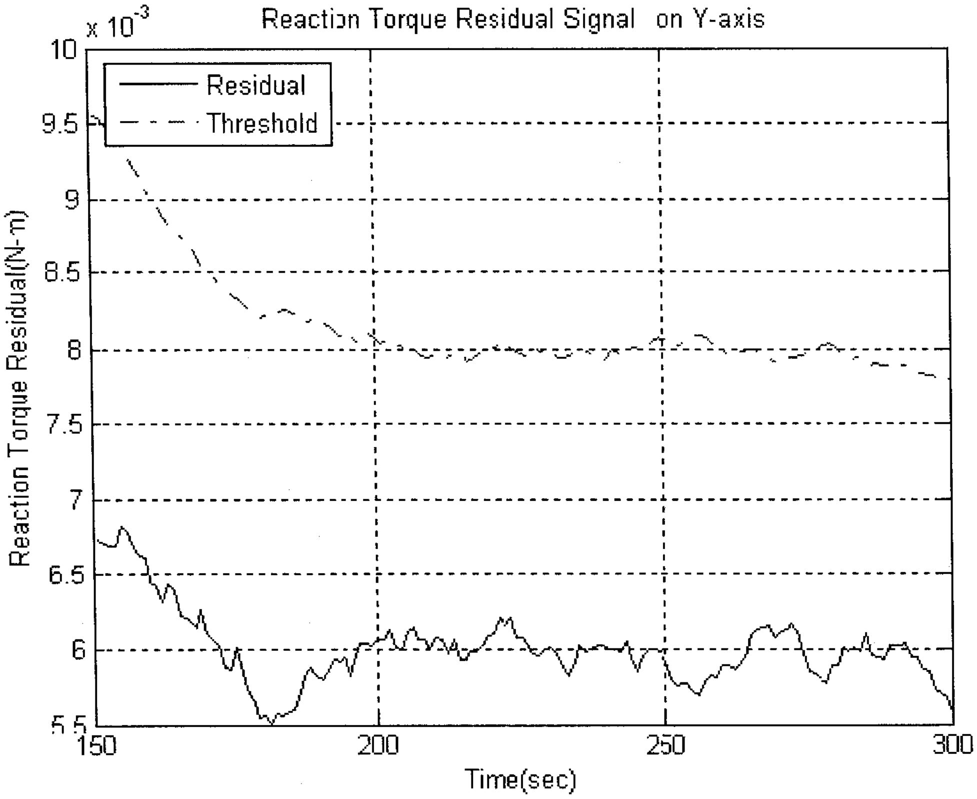
Multiple Faults (faults happening in more than one axis)

This case study is used to check the performance of this scheme under multiple faults, when more than one wheel was faulty. Like the previous case, a bus voltage fault happens in the wheel of X axis at 200 seconds and a current loss fault in the wheel of Z axis at 200 seconds. Obviously, from Figure 4.23, these two faults were both detected and isolated properly.




Current Limiter Signal of the Wheel on Z-axis
Current Limiter
1
0.8
0.6
0.4
0.2
0
0
50
100
150
200
250
Time(sec)


x 10^-3
Reaction Torque Residual Signal on X-axis
Residual
Threshold
Reaction Torque Residual(N-m)
8
7
6
5
4
3
2
1
150
200
250
300
Time(sec)



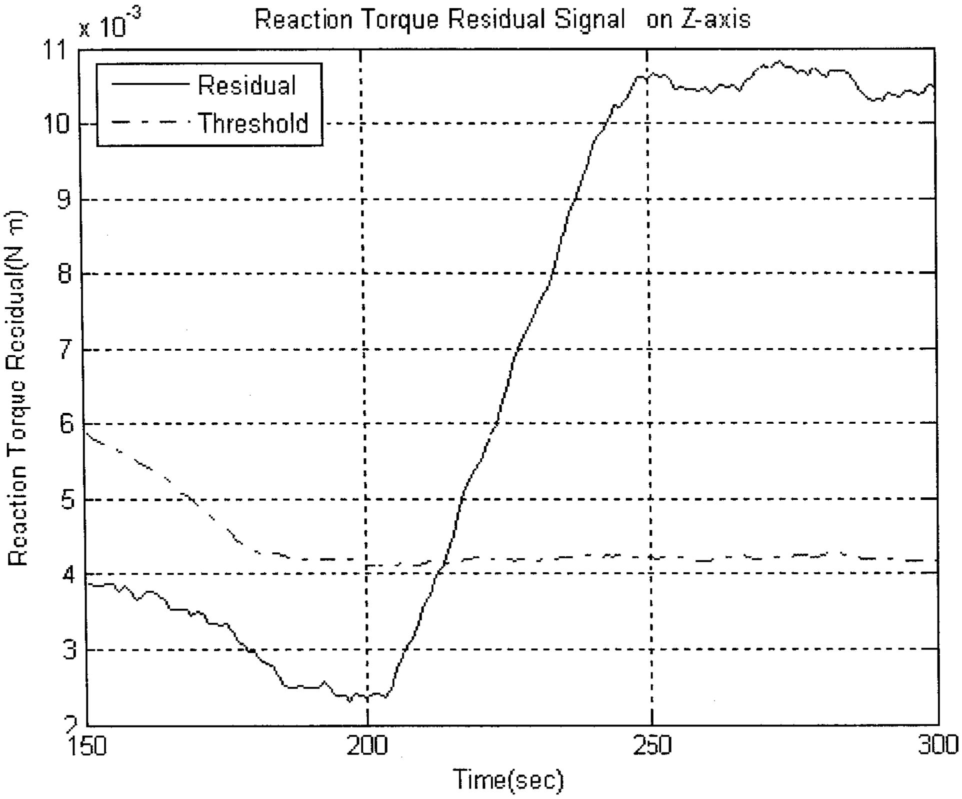


Figure 4.23 Multiple Faults Detection and Isolation Case

A summary of the correctness of this neural network observer-based FDI scheme with a number of (100 groups for each type of fault) simulation results is shown in Table 4.2. From this table, we can observe the neural network observer-based scheme is very effective in the FDI for the reaction wheels.

Fault Types	Bus Voltage Fault (2.5 V off normal value)	Current Loss Fault (40% current loss)	Temperature Fault (150 ^0C off normal value)
Correctness	95%	98%	98%

Table 4.2 Correctness Test in Fault Cases by Using Neural Network Observer-based Scheme

4.5 Conclusions

In this chapter, firstly, we gave a brief introduction to neural networks. We also developed a neural network observer-based scheme for FDI purpose in the reaction wheels. Three Elman neural networks are trained to model the dynamics of the reaction wheels separately. By using the three threshold curves generated in normal operating conditions, we can detect and isolate faults in the reaction wheels effectively. As the simulation results shown above, this scheme is quite reliable for FDI purpose in the wheels.

Chapter 5

Comparative Study of Neural Network Observer-based FDI Scheme and Linear Observer-based FDI Scheme

5.1 General Comparison of Linear Observer-based FDI Scheme and Neural Network Observer-based FDI Scheme

As we have discussed in Chapter 3 and Chapter 4, both of these schemes can be used to detect and isolate some faults in the reaction wheels. These two FDI schemes highly depends on the modeling accuracy of the reaction wheel dynamics. But as we showed in Figure 2.2, reaction wheel is a highly non-linear dynamic system. It contains many non-linear elements and disturbances, such as viscous friction, torque ripple disturbances and bearing friction and disturbances.

For linear observer-based FDI scheme, we ignore all these non-linear and disturbances effects and only use a linear observer to model the dynamics of the reaction wheel. This kind of representation is not quite accurate. Thus this FDI scheme will be insensitive to the small faults and not roubust to disturbances and nosies.

While the neural network observer-based FDI scheme developed in Chapter 4 uses a well trained recurrent neural network to represent the dynamics of the reaction wheel. According to [46], neural network processes the inherent nonlinearity property which is suitable for non-linear dynamic system modeling. Also neural networks show good adaptive capability to noises and disturbances. All these properties of neural networks make the neural network observer-based FDI scheme more suitable for faults detection and isolation in reaction wheels than the linear observer-based scheme, especially for small faults detection and robustness to disturbances and noises. These will be shown through the simulation results in the following subsections.

5.2 Comparative Study through Simulation Results

Two main aspects of comparative study will be conducted here through some simulation results: their sensitivity to small faults and their robustness to the disturbances and noises. We are going to use these two schemes to detect and isolate the same fault and indicate the advantages of the neural network observer-base scheme.

5.2.1 Sensitivity Comparative Study to Small Faults

1) This example shows a case study of bus voltage fault detection and isolation. As we see in Figure 5.1, the bus voltage of the reaction wheel aligned on X axis dropped from the normal value $8V$ to $5.5V$ at 200 seconds after a set point change command was applied. As we compare the simulation results in Figure 5.2 and Figure 5.3, the residual curve of X axis generated by the linear observer-based scheme exceeded its correspongding threshold curve shortly after the fault

happened, but it came down beneath the residual curve again in a short time. This is unsuitable for fault detection and isolation. While the residual curve generated by the neural network observer-based scheme exceeded its corresponding threshold curve and remained above the threshold curve thereafter. It supplied enough time for FDI purpose. Obviously, in this way, the neural network observer-based scheme is better than the linear counterpart.

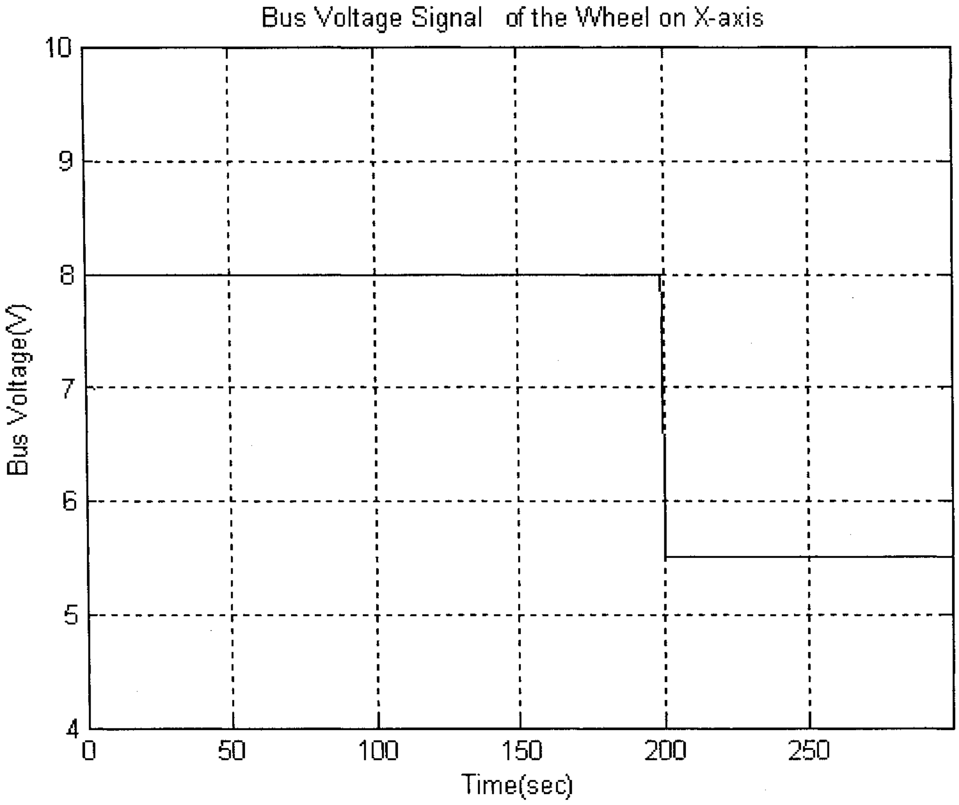
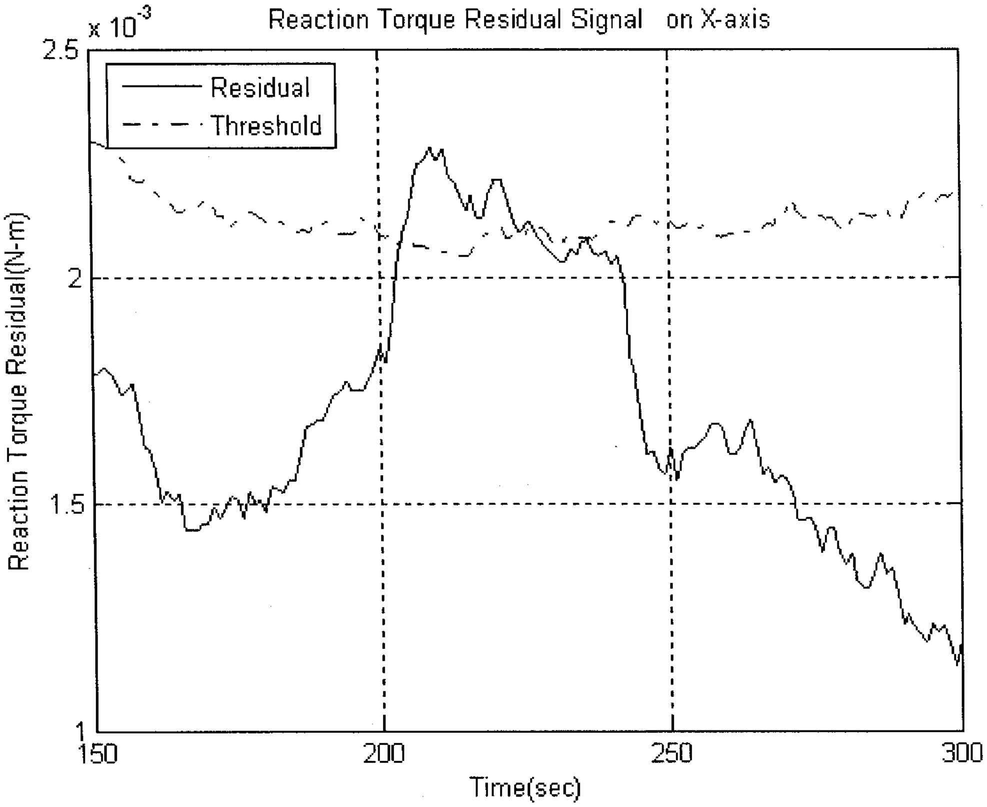
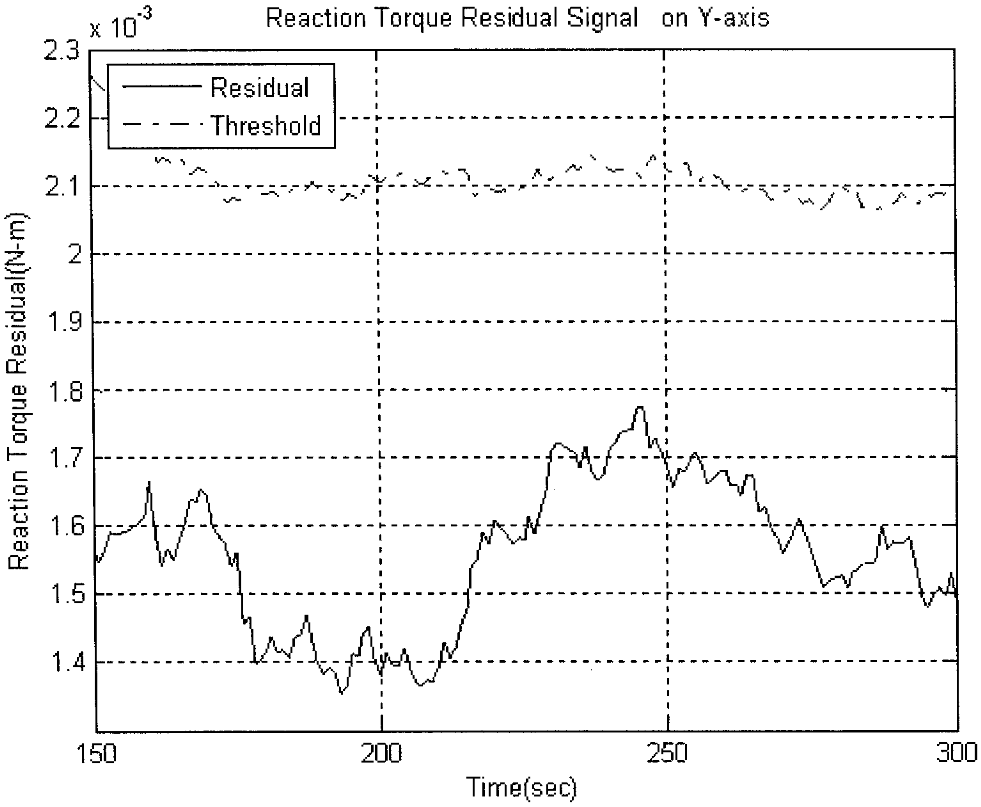


Figure 5.1 Small Bus Voltage Fault Signal

- Results by using the linear observer-based FDI scheme


Reaction Torque Residual Signal on X-axis
x 10^-3
2.5
2
1.5
1
150
200
250
300
Time(sec)
Reaction Torque Residual(N-m)
Residual
Threshold


Reaction Torque Residual Signal on Y-axis
x 10^-3
2.3
2.2
2.1
2
1.9
1.8
1.7
1.6
1.5
1.4
150
200
250
300
Time(sec)
Reaction Torque Residual(N-m)
Residual
Threshold

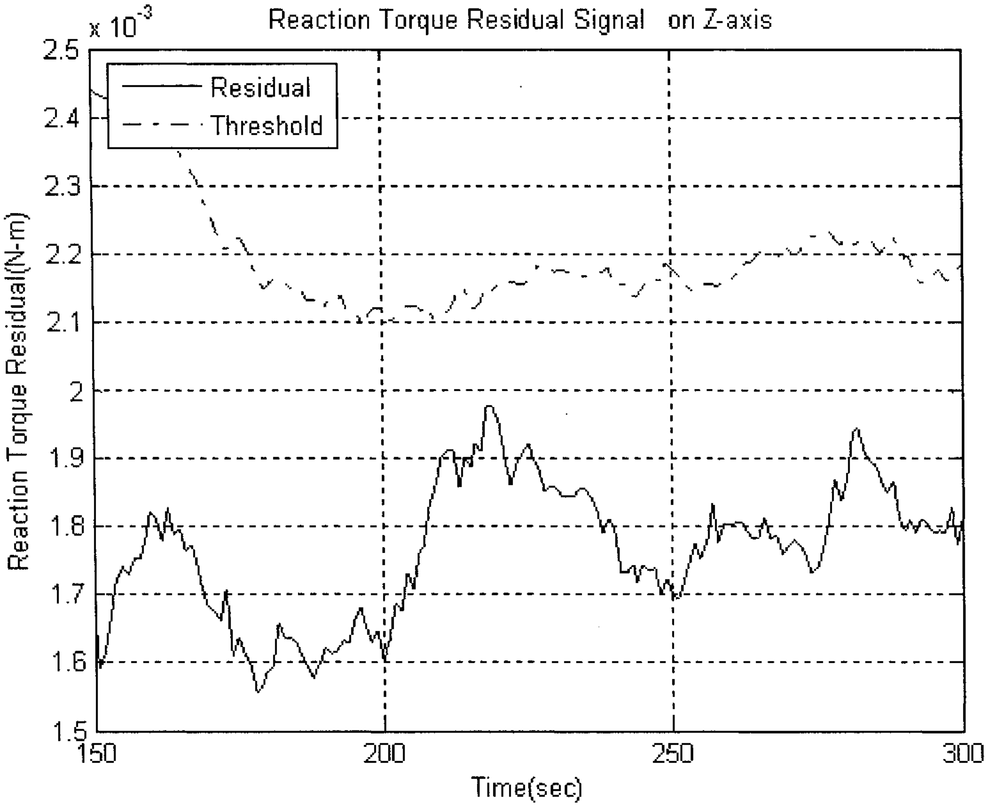
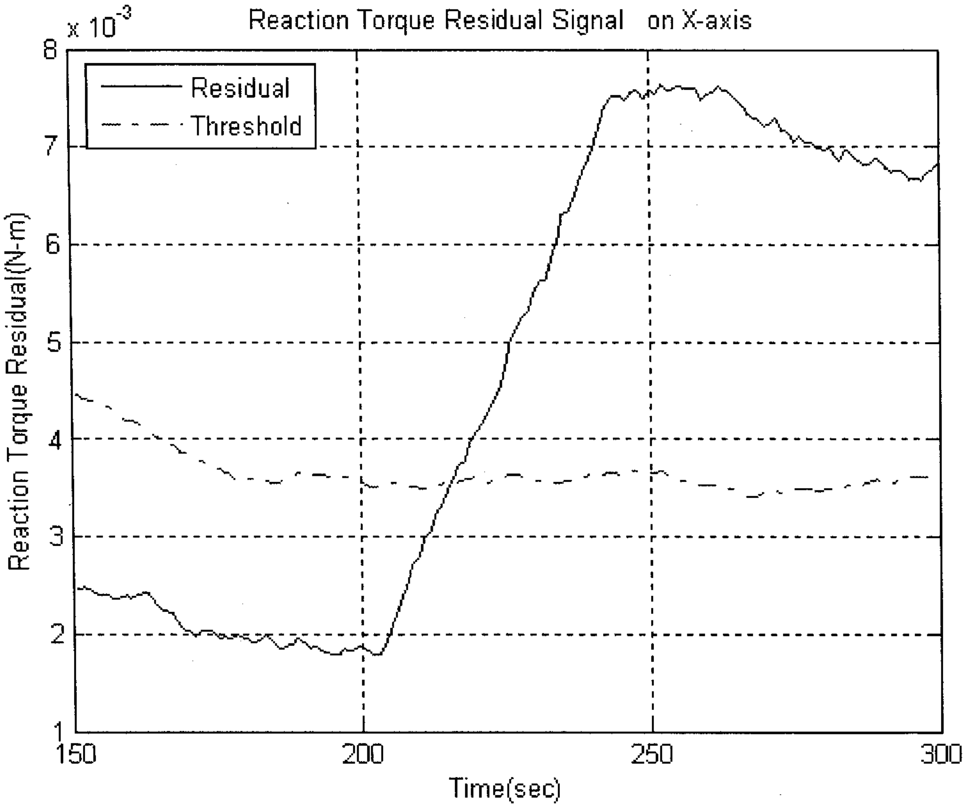
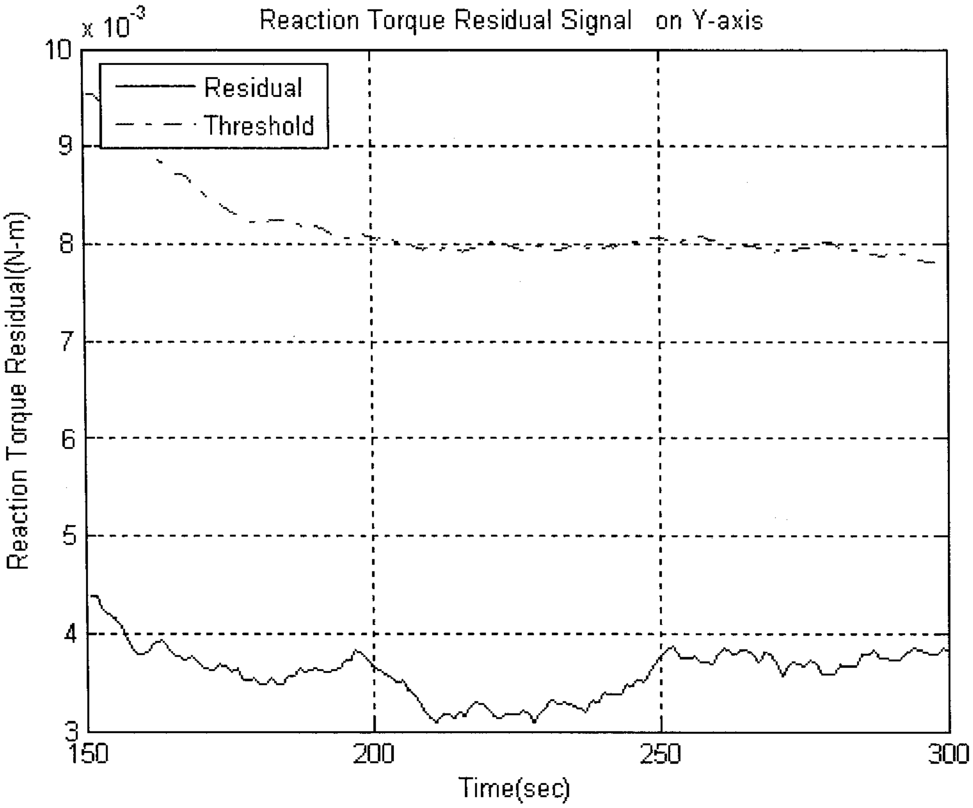


Figure 5.2 Linear Observer-based Scheme Performance for Small Bus Voltage Fault

- Results by using the neural network observer-based FDI scheme



Reaction Torque Residual Signal on X-axis
x 10^-3
Residual
Threshold
Reaction Torque Residual(N-m)
8
7
6
5
4
3
2
1
150
200
250
300
Time(sec)



Reaction Torque Residual Signal on Y-axis
x 10^-3
Residual
Threshold
Reaction Torque Residual(N-m)
10
9
8
7
6
5
4
3
150
200
250
300
Time(sec)

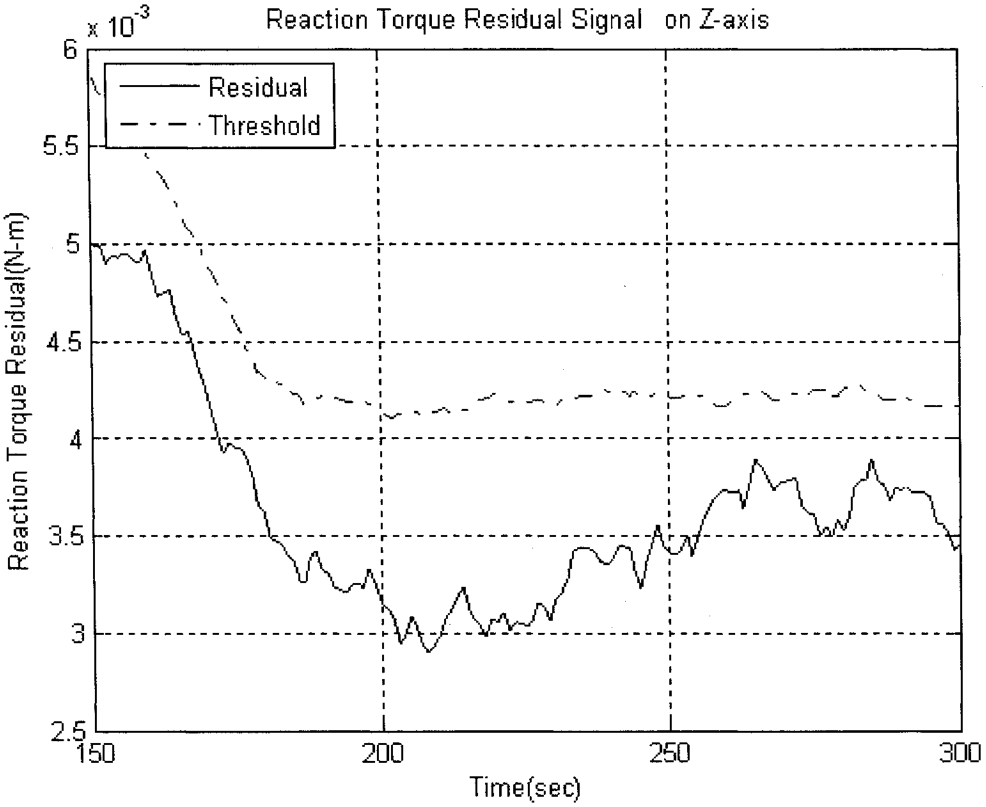


Figure 5.3 Neural Network Observer-based Scheme Performance for Small Bus Voltage Fault

2) This example shows another case study about current loss fault detection and isolation. As we see in Figure 5.4, 40% of motor current of the reaction wheel aligned on X axis was lost at 200 seconds after a set point change command was applied. Through comparing the simulation results between Figure 5.5 and Figure 5.6, we find that although the residual curve of X-axis generated by the linear observer-based scheme increased for a few seconds after the fault happened, it was always beneath the residual curve. Therefore, we cannot detect this fault with this method. While by using the neural network observer-based scheme, the residual curve exceeded its corresponding threshold curve and remained above the threshold curve thereafter. In this way, the current loss fault was correctly detected and isolated. Clearly, the neural network observer-based

scheme shows much better performance than the linear observer-based scheme in this case.

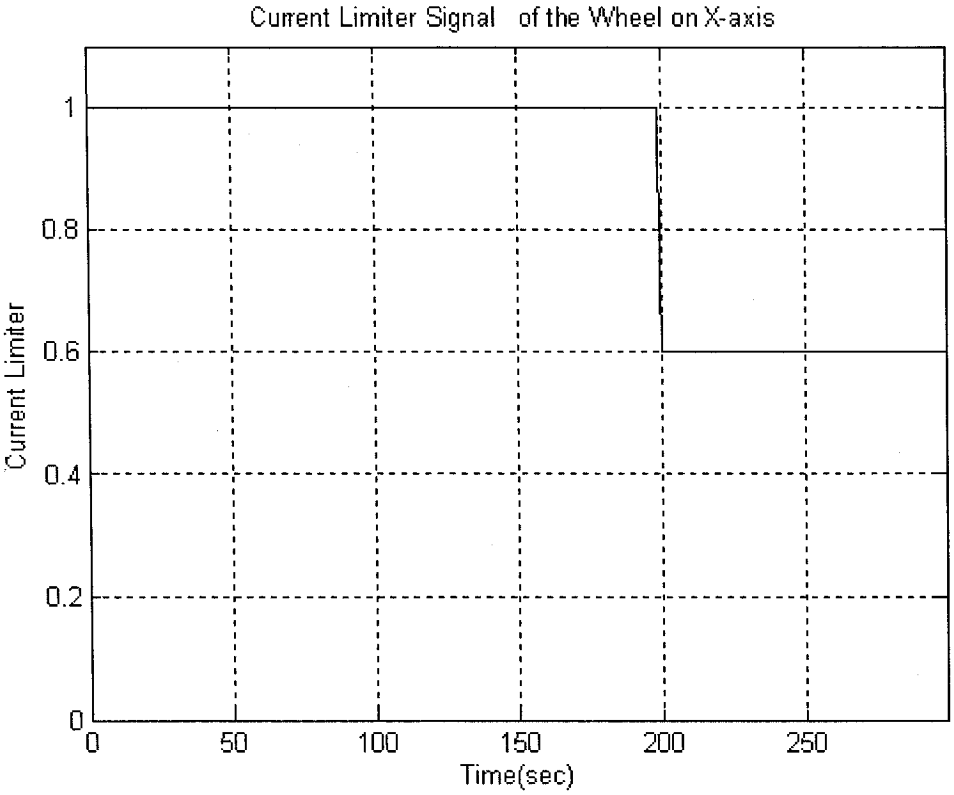
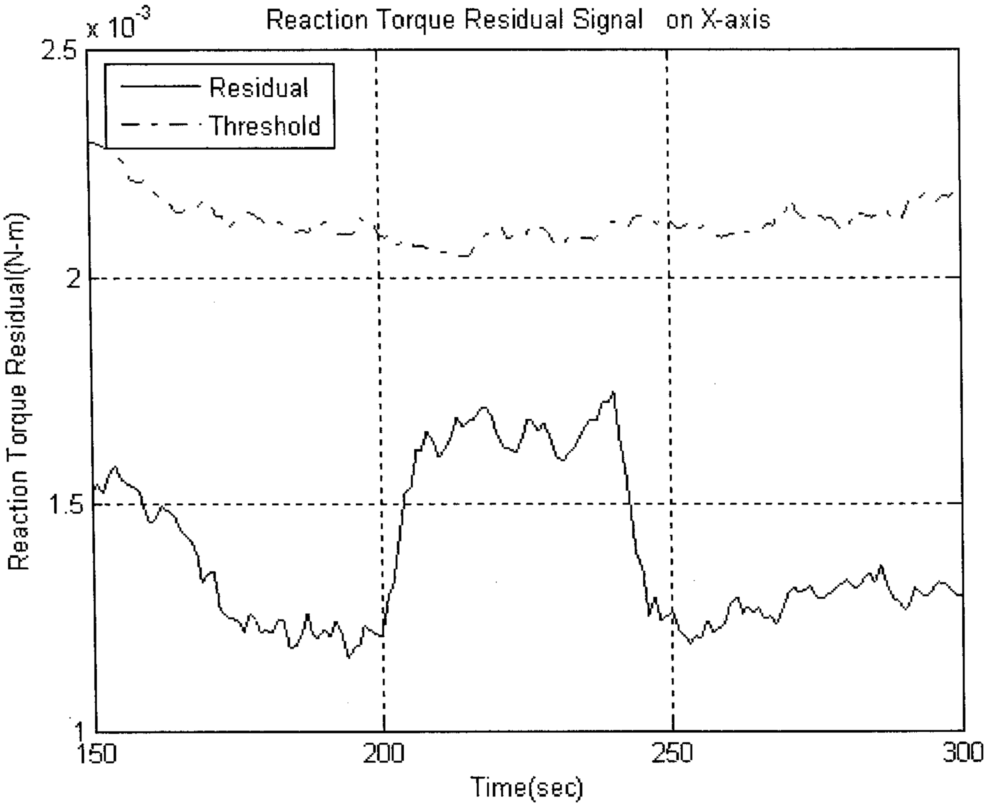
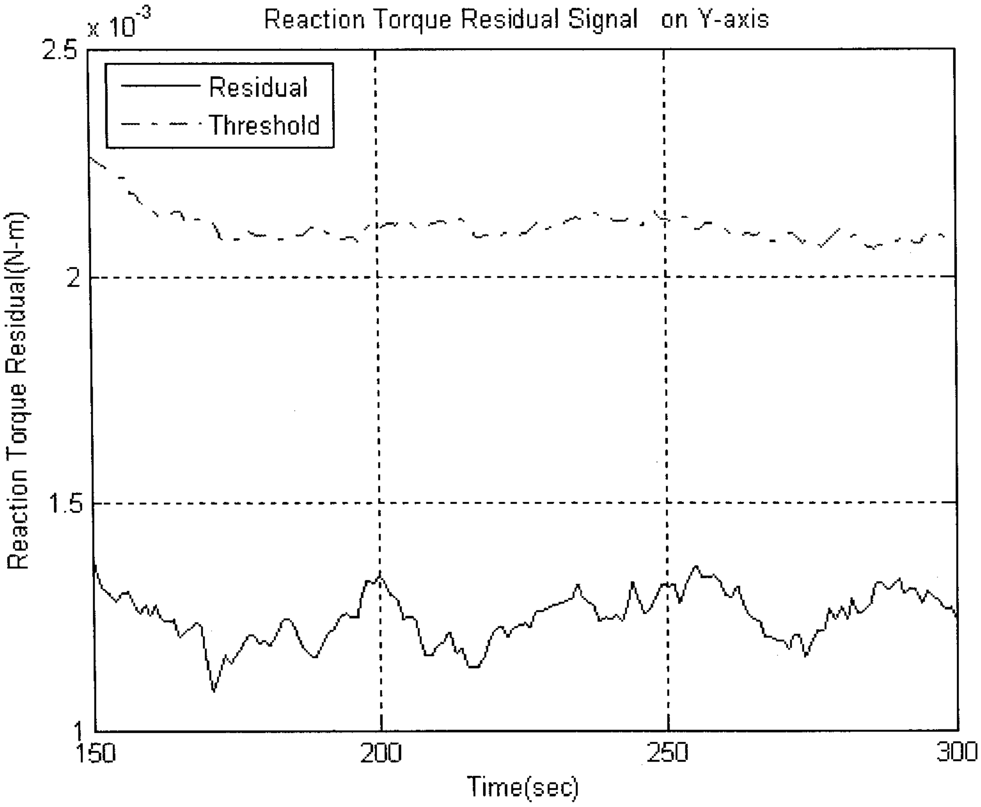


Figure 5.4 Small Current Loss Fault Signal

- Results by using the linear observer-based FDI scheme


Reaction Torque Residual Signal on X-axis
x 10^-3
2.5
2
1.5
1
Reaction Torque Residual(N-m)
150
200
250
300
Time(sec)
Residual
Threshold


Reaction Torque Residual Signal on Y-axis
x 10^-3
2.5
2
1.5
1
Reaction Torque Residual(N-m)
150
200
250
300
Time(sec)
Residual
Threshold

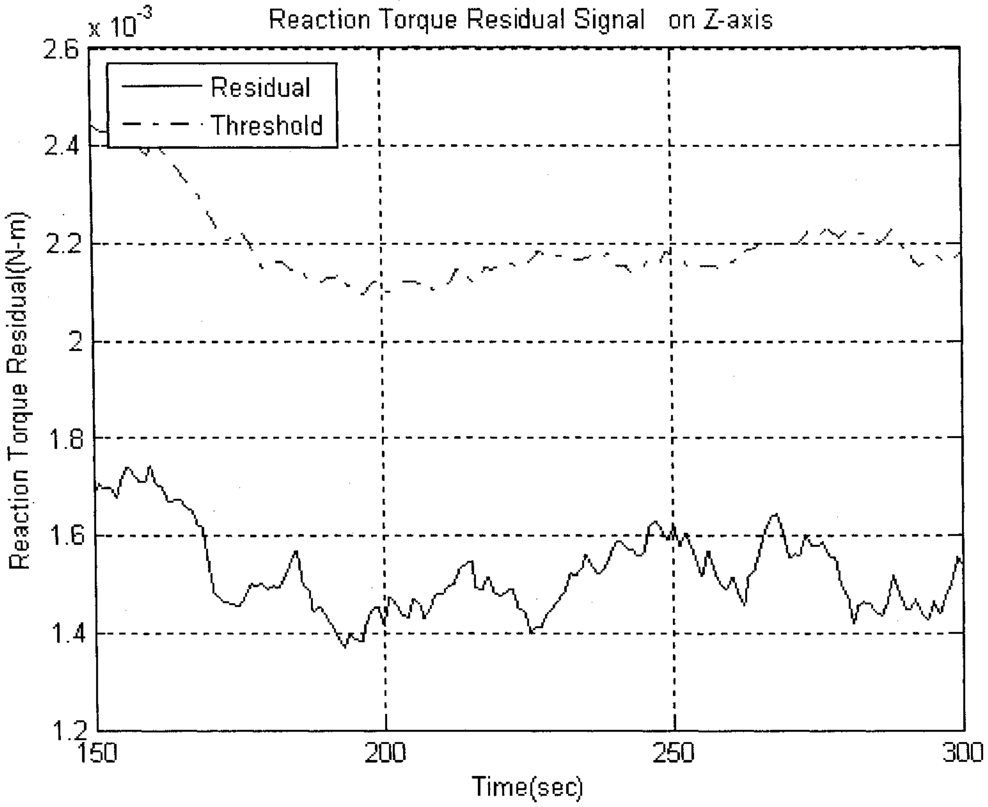
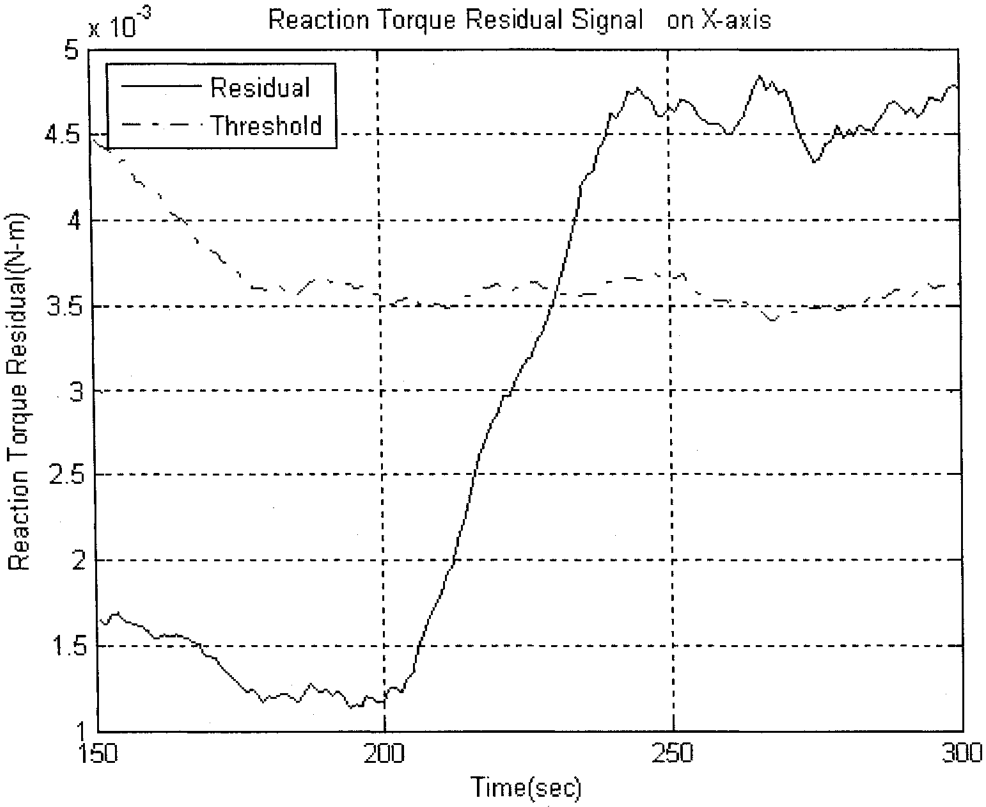
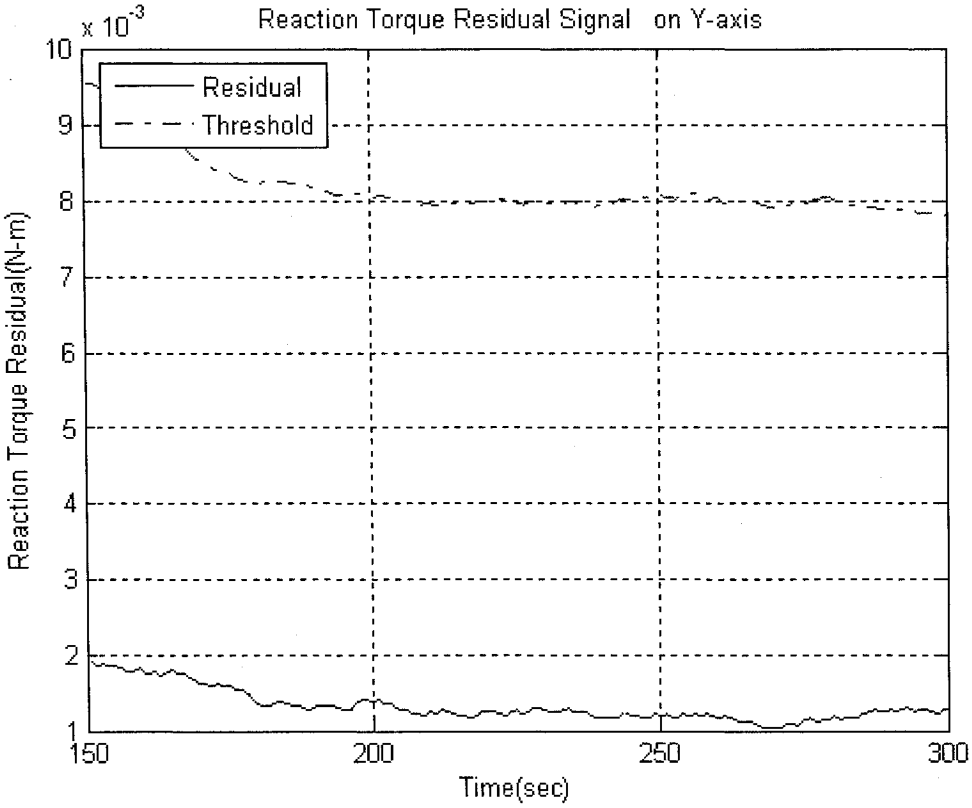


Figure 5.5 Linear Observer-based Scheme Performance for Small Current Loss Fault

- Results by using the neural network observer-based FDI scheme


Reaction Torque Residual Signal on X-axis
x 10^{-3}
5
4.5
4
3.5
3
2.5
2
1.5
1
150
200
250
300
Time(sec)
Reaction Torque Residual(N-m)
Residual
Threshold


Reaction Torque Residual Signal on Y-axis
x 10^{-3}
10
9
8
7
6
5
4
3
2
1
150
200
250
300
Time(sec)
Reaction Torque Residual(N-m)
Residual
Threshold

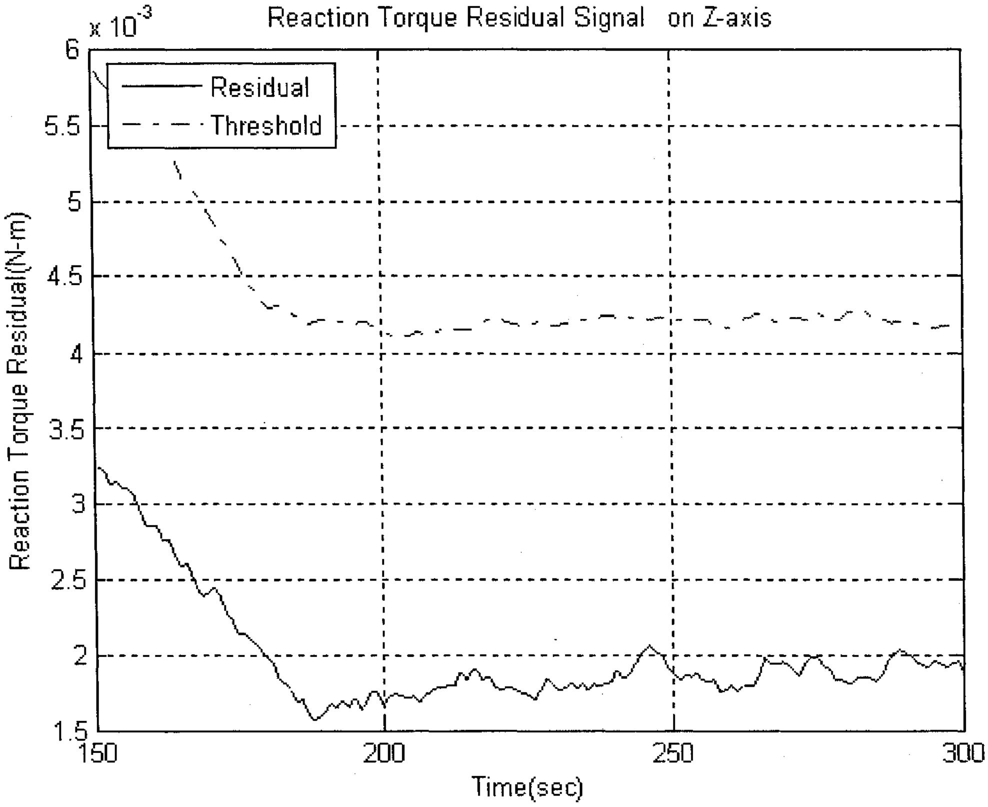


Figure 5.6 Neural Network Observer-based Scheme Performance for Small Current Loss Fault

A detailed comparison with simulation results of these two schemes is shown in Table 5.1 and Table 5.2. We assume that the initial conditions are the same for all cases and all these faults occur at 200 seconds after a set point change command was applied on the wheel aligned with X-axis. For every faulty case, these two schemes are used for FDI. Three parameters are used as guideline for comparison: detection time, detection delay and the gap between threshold and residual. (Vbus: Bus Voltage Fault; Curr: Current Loss Fault; Temp: Temperature Fault)

Fault Scenario	Detectable			
	Yes			No
	Detection Time (second)	Detection Delay (second)	Gap (10^{-3}N-m)	Gap (10^{-3}N-m)
VBus 10% off	/	/	/	0.35
VBus 20% off	/	/	/	0.31
VBus 30% off	/	/	/	0.20
VBus 40% off	20	10	0.10	/
Curr 20% off	/	/	/	0.34
Curr 30% off	/	/	/	0.25
Curr 40% off	/	/	/	0.15
Curr 50% off	40	5	0.30	/
Temp 60^oC off	/	/	/	0.00
Temp 90^oC off	70	20	0.20	/
Temp 120^oC off	>80	15	0.40	/
Temp 150^oC off	>90	10	0.50	/

Table 5.1 Fault Detection Using Linear Observer-based Scheme

Fault Scenario	Detectable			
	Yes			No
	Detection Time (second)	Detection Delay (second)	Gap (10^{-3} N-m)	Gap (10^{-3} N-m)
VBus 10% off	/	/	/	0.00
VBus 20% off	>80	20	1.20	/
VBus 30% off	>80	20	3.30	/
VBus 40% off	>85	15	4.90	/
Curr 20% off	>35	65	0.25	/
Curr 30% off	>65	35	1.00	/
Curr 40% off	>80	20	3.00	/
Curr 50% off	>85	15	6.50	/
Temp 60^{o}C off	>30	60	0.25	/
Temp 90^{o}C off	>75	25	1.20	/
Temp 120^{o}C off	>85	15	2.90	/
Temp 150^{o}C off	>80	20	3.60	/

Table 5.2 Fault Detection Using Neural Network Observer-based Scheme

5.2.2 Robustness Comparative Study to Disturbances and Noises

1) We want to check the performance of these two schemes when some disturbances change in the wheel, specifically, we want to see whether disturbance changes will result in false detection as a fault in the wheel. As shown in Figure 2.2, torque ripple disturbance is one main disturbances in the wheel. In this example let us suppose this torque ripple disturbance in the wheel on X-axis increases by 40% at 200 seconds as shown in Figure 5.7 after a set point change command was applied at zero second. From Figure 5.8, we find that the residual curve of X-axis generated by the linear observer-based scheme exceeded its corresponding threshold curve after this disturbance changed, thus this disturbance change will result in false detection. While by using the neural network observer-based scheme, the residual curve was almost not affected by this disturbance increasing and remained beneath the threshold curve. This time, no fault is indicated by the neural network observer-based scheme due to this disturbance change. Through this example, we can see that a neural network observer-based FDI scheme is more robust to the disturbances and noises.

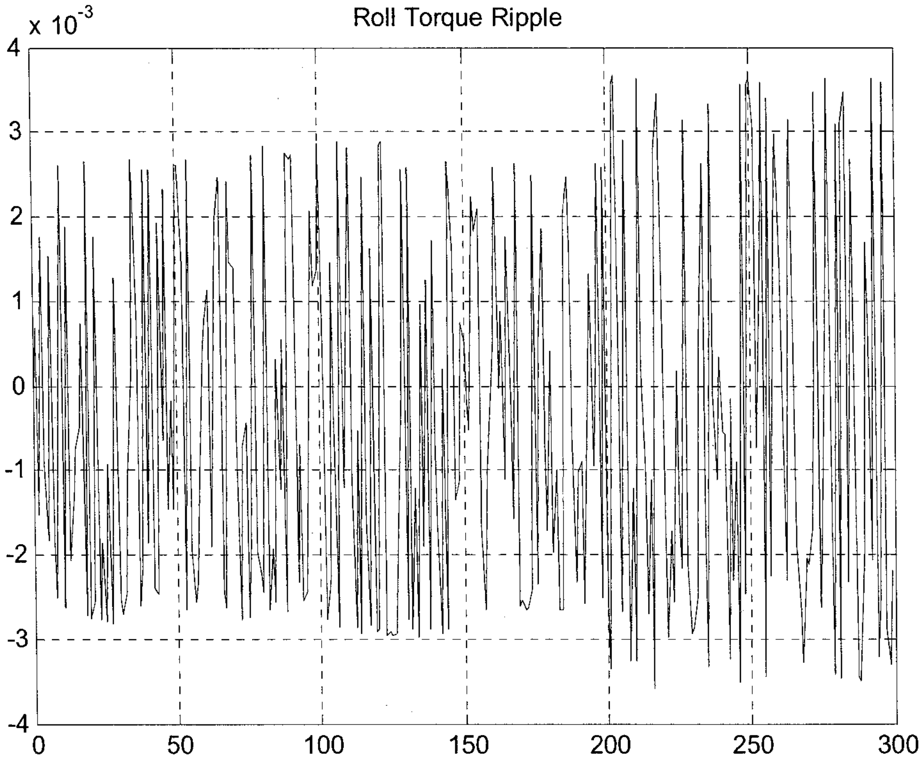
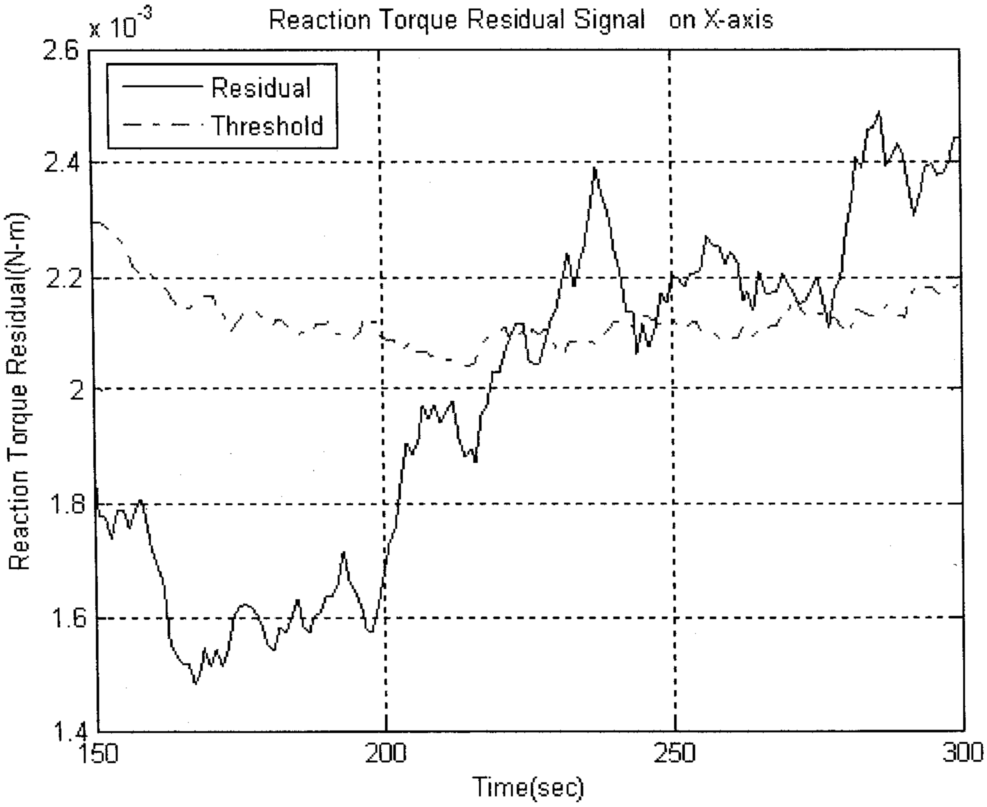
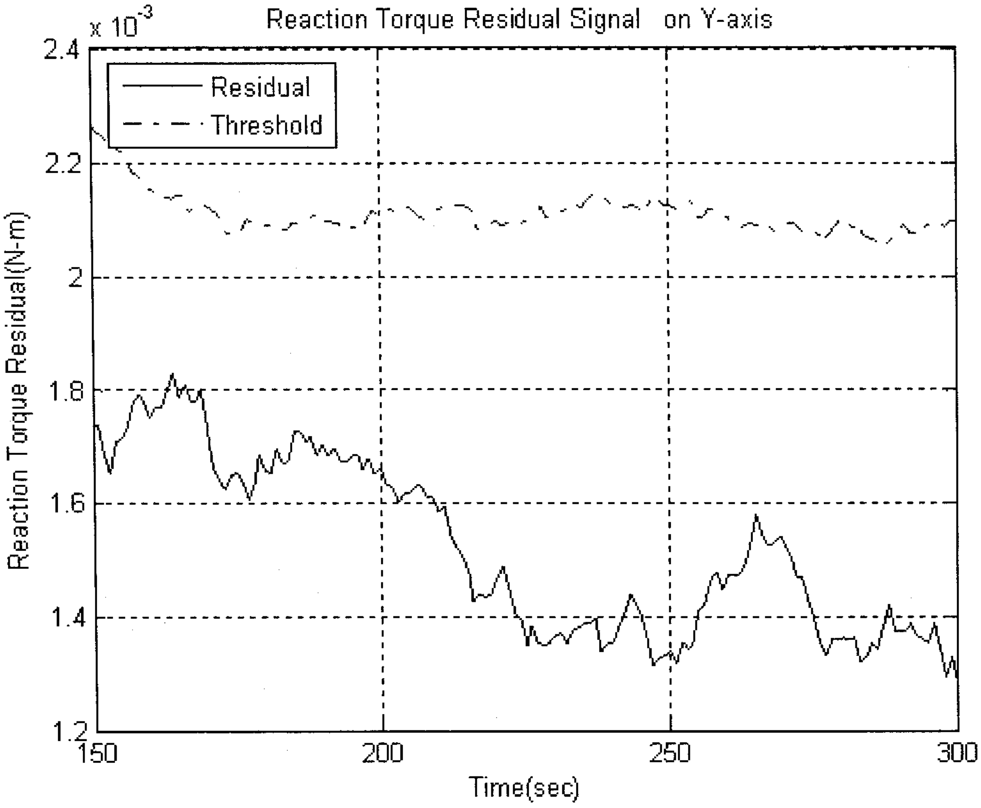


Figure 5.7 Torque Ripple Disturbance Signal in the Wheel on X-axis

- Results by using the linear observer-based FDI scheme


Reaction Torque Residual Signal on X-axis
x 10^-3
Residual
Threshold
Reaction Torque Residual(N-m)
2.6
2.4
2.2
2
1.8
1.6
1.4
150
200
250
300
Time(sec)


Reaction Torque Residual Signal on Y-axis
x 10^-3
Residual
Threshold
Reaction Torque Residual(N-m)
2.4
2.2
2
1.8
1.6
1.4
1.2
150
200
250
300
Time(sec)

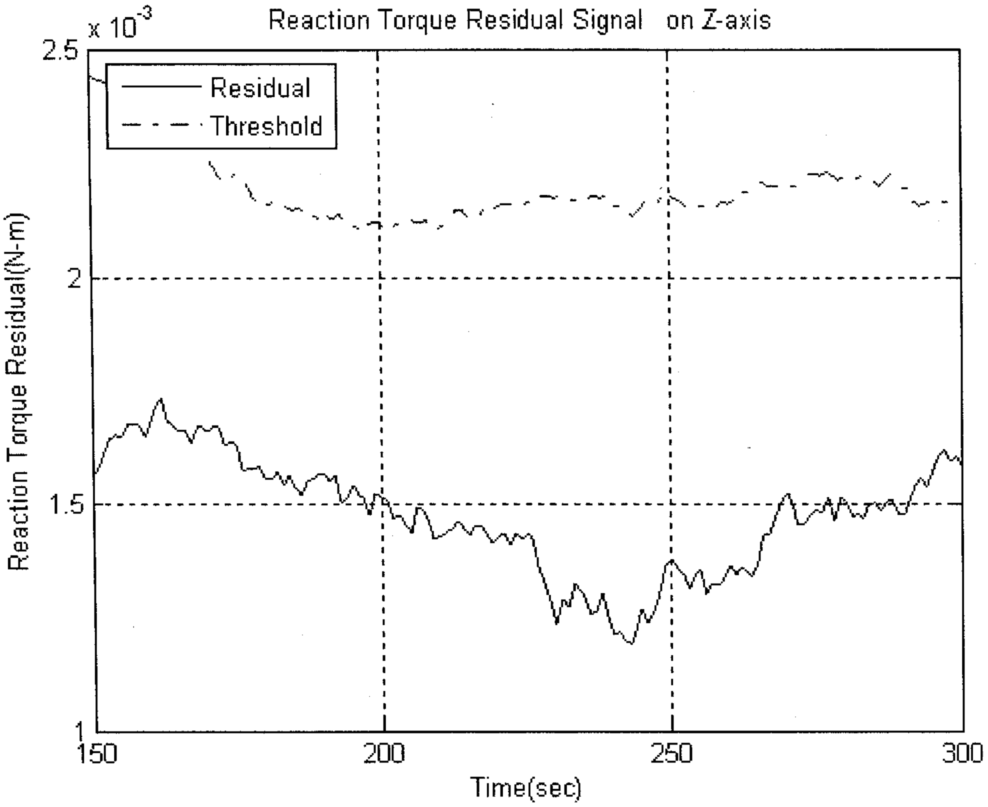
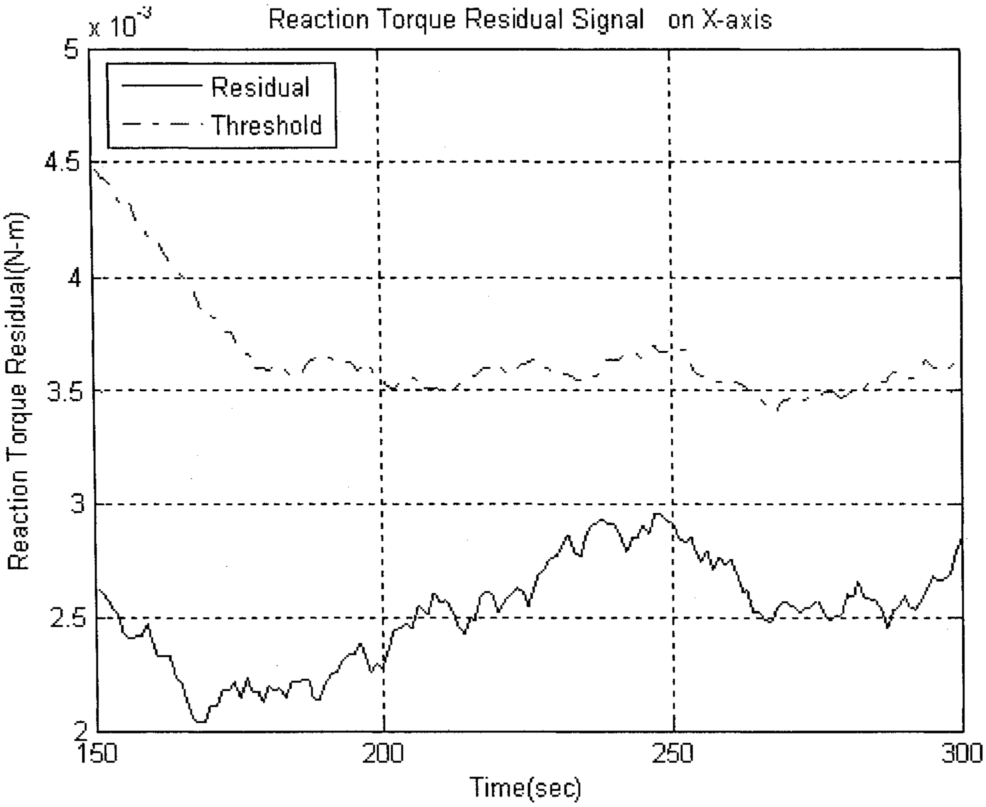
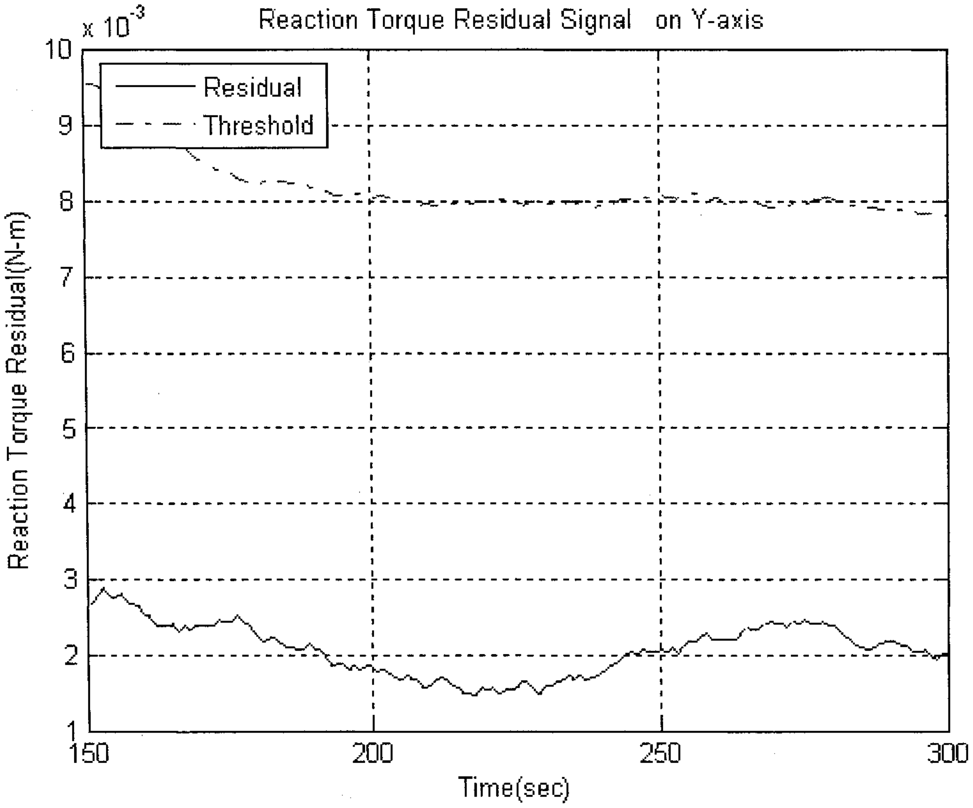


Figure 5.8 Linear Observer-based Scheme Performance when Disturbance Changes

- Results by using the neural network observer-based FDI scheme


Reaction Torque Residual Signal on X-axis
x 10^-3
Residual
Threshold
Reaction Torque Residual(N-m)
5
4.5
4
3.5
3
2.5
2
150
200
250
300
Time(sec)


Reaction Torque Residual Signal on Y-axis
x 10^-3
Residual
Threshold
Reaction Torque Residual(N-m)
10
9
8
7
6
5
4
3
2
1
150
200
250
300
Time(sec)

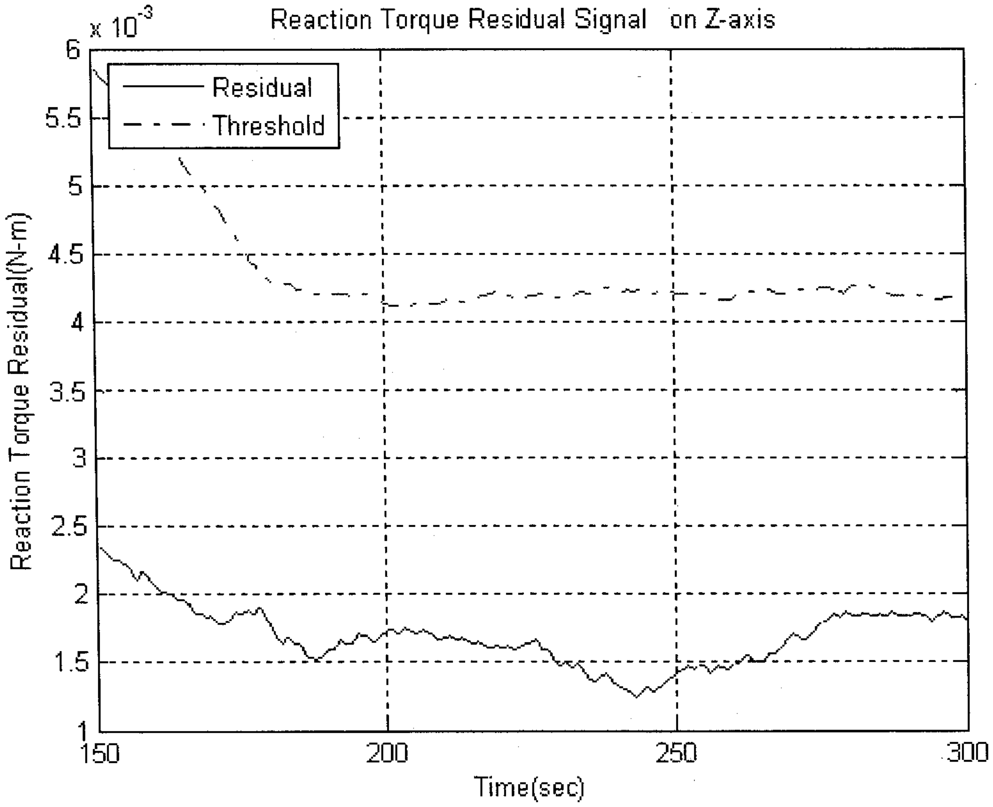


Figure 5.9 Neural Network Observer-based Scheme Performance when Disturbance Changes

From the simulation results above, we can conclude that the neural network observer-based FDI scheme is better than the linear observer-based method in the sense that it is more sensitive to small faults and robustness to disturbances.

5.3 Conclusions

In this chapter, we conducted thorough comparative studies between the two FDI schemes developed in Chapter 3 and Chapter 4. Based on many simulation results as shown above, the neural network observer-based scheme shows better performance in its sensitivity to the small faults and robustness to disturbances than its linear counterpart.

Chapter 6

Conclusions and Future work

6.1 Conclusions

This thesis has developed a neural network observer-based scheme for fault detection and isolation in the reaction wheels. Reaction wheels serves as actuators for the spacecraft attitude control. Any fault occuring in the reaction wheels will deteriorate the performance of the spacecraft according to the attitude set point change command, or if serious, will cause the spacecraft attitude out of control. The faults we consider in the thesis are bus voltage fault, motor current loss fault and temperature fault.

First, we design three classical PID controllers to achieve the three axes attitude control of the spacecraft. Each PID controller generates a desired torque command voltage and supplies it to the corresponding reaction wheel for control purpose. Then we investigate a linear observer-based scheme for fault detection and isolation in the reaction wheels. It does show kind of capability in fault detect and isolation in some cases, but it is not good enough for small fault detection and it is easily affected by the disturbances and noises in the wheels.

Alternatively, based on its non-linear mapping and adaptive properties, a neural network is introduced in our thesis to achieve better FDI performance. According to our neural network observer-based scheme, three neural networks are applied to estimate the outputs of the wheels in three axes separately and independently to simply perform the fault detection and isolation.

The network we introduced is one kind of recurrent network, Elman network, which is suitable for non-linear dynamic system modeling. The torque command voltage generated by the PID controller and one step delay of actual wheel output or estimated wheel output serve as the network inputs. Obviously, the network output is the estimated value of the wheel output, that is reaction torque. After the network has been trained approximately, it can model the dynamics of the reaction wheel and supplies a good estimated reaction torque for FDI purpose. After some post-processing of the error signal between the actual and estimated value of reaction torque, we generate the residual curve for threshold checking. Any threshold exceeding will indicate the corresponding wheel is faulty, thus the fault is detected and isolated.

Through a comparative study between the linear observer-based scheme and neural network observer-based scheme, the latter one is shown to be a better choice for fault detection and isolation in the reaction wheels. It is more sensitive to small faults and it has less probability of making false detection in noisy environments.

All the attitude control system construction, development of FDI schemes and cases studies are simulated in Matlab (Version 7.01) with Neural Network Toolbox.

6.2 Future Work

As a further work it would be useful to design another network (probably radial basis neural network) to analyse the residual signals for fault classification. Possibly some fuzzy logic methods can be considered too. In this way, we can distinguish the type of fault : bus voltage fault, motor current loss fault or temperature fault. Consequently, we can achieve better system reconfiguration in fault cases.

Another recommendation for future work is to conduct a comparative study between our proposed neural network observer-based schemes with some other non-linear observer-based schemes, such as the methods based on unknow input observer and eigenstructure assignment. These methods are more complex than the linear observer-based scheme in our thesis and show good performance in many problems in the literature. Their comparison with the neural network observer-based scheme is also helpful.

The last thing we suggest here is to investigate other types of neural network can be tried to use for the FDI purpose. The network we introduced is Elman network. The application of some other types of networks such as the network with tapped-delay line or more complex networks is worth further investigation.

References

[1] Hughes, P. C. Spacecraft Attitude Dynamics, John Wiley & Sons, 1986

[2] Kaplan, M. H. Modern Spacecraft Dynamics and Control, John Wiley & Sons, 1976

[3] Kane, T. R., Likins, P. W. and Levinson, D. A. Spacecraft Dynamics, McGraw-Hill, 1983

[4] Wertz, J. R. (editor), Spacecraft Attitude Determination and Control, Kluwer Academics Publishers, 1995

[5] Vadali, S. R., Krishnan, S. and Singh, T., Attitude Control of Spacecraft Using Neural Networks, Proceedings of the 1993 AAS/AIAA Spaceflight Mechanics Meeting, Paper no. AAS 93-192, pp.271-285, 1993

[6] Satyadas, A. and KrishnaKumar, K., EFM-based Controller for Space Station Attitude Control: Application and Analysis, Genetic Algorithms and Software Computing, pp.152-171, Physica-Verlag (Studies in Fuzziness, Vol. 8), 1996

[7] Lindblad, T., Lindsey, C. S., Minerskjold, M., Eide, A., Linden, T. and Shelton, Robert O., Attitude Control Systems for Spacecrafts using Neural Networks and Fuzzy Logic, Proceedings of AIHENP95, April 1995

[8] Bialke, Bill,, High Fidelity Mathematical Modeling of Reaction Wheel Performance, 1998

[9] Simani,Silvio, Fantuzzi, Cesare and Patton, Ron J. Model-Based Fault Diagnosis in Dynamic Systems Using Identification Techniques, Springer, Berlin, 2002

[10] Isermann, R., Supervison, Fault Detection and Fault Diagnosis Methods: An Introduction, Contr. Eng. Practice 5(5), 1997

[11] Patton, R. J., Frank, P. M., and Clark, R. N., editors Issues of Fault Diagnosis for Dynamic Systems, Springer-Verlag, London Limited, 2000

[12] Basseville, M. and Nikiforov, I. V. Detection of Abrupt Changes: Theory and Application, 1993

[13] Gertler, J. Fault Detection and Diagnosis in Engineering Systems, Marcel Dekker, New York, 1998

[14] Chow, E. Y. and Willsky, A. S., Issues in the development of a general algorithm for reliable failure detection, proc. of the 19th Conf. on Decision and Control, Albuquerque, NM, 1980

[15] Patton, R. J., Uppal, F. J., Lopez-Toribio, C. J., Soft Computing Approaches to Fault Diagnosis for Dynamic System: A Survey, Proc. 4th IFAC Symposium on Fault Detection Supervision and safety for Technical Process, June 2000

[16] Lapp, S. A. and Powers, G. A., Computer-aided synthesis of fault-trees, IEEE Trans. Reliability, 37, 2-13, 1977

[17] Antsaklis, P. J. and Passino, K. M., Eds., An Introduction to Intelligent and Autonomous Control, Norwell, MA: Kluwer, 1993

[18] Stengel, R. F., Toward Intelligent Flight Control, IEEE Trans. Syst., Man, Cybern., Vol. 23, No.6, pp. 1699-1717, 1993

[19] Sorsa, T., Koivo, H. N., and Koivisto, H., Neural Networks in Process Fault Diagnosis, IEEE Trans. Syst., Man, Cybern., Vol. 21, No. 4, pp. 815-825, 1991

[20] Maki, Y. and Loparo, K. A., A Neural Network Approach to Fault Detection and Diagnosis in Industry Processes, IEEE Trans. Syst. Technol., Vol. 5, No. 4, pp. 529-541, 1997

[21] Ayoubi, M. and Isermann, R., Neuro-fuzzy System for Diagnosis, Fuzzy Sets Syst., Vol. 89, No. 3,

pp. 289-307, 1997

[22] Passino, K. M. and Antsaklis, P. J., Fault Detection and Identification in an Intelligent Restructurable Controller, J. Intell. Robot. Syst., Vol. 1, pp. 145-161, 1988

[23] Laukonen, E. G., Passino, K. M., Krishnaswami, V., Luh, G. C., and Rizzoni, G., Fault Detection and Isolation for an Experimental Internal Combustion Engine via Fuzzy Identification, IEEE Trans. Contr. Syst. Technol., Vol. 3, No. 4, pp. 347-355, 1995

[24] Schneider, H. and Frank, P., Observer Based Supervision and Fault Detection in Robots Using Nonlinear and Fuzzy Logic Residual Evaluation, IEEE Trans. Contr. Syst. Technol., Vol. 4, No. 3, pp. 274-282, 1996

[25] Frank, P. M. and Koppen-Seliger, B., Fuzzy Logic and Neural Network Application to Fault Diagnosis, Int. J. Approximate Reasoning, Vol. 16, No. 1, pp. 67-88, 1997

[26] Isermann, R., One Fuzzy Logic Application for Automatic Control, Supervision, and Fault Diagnosis, IEEE Trans. Syst., Man, Cybern. Part A, Vol. 28, No. 2, pp. 221-235, 1998

[27] Dash, S. and Venkatasubramanian, V., Challeges in the industrial application of fault diagnostic systems, Proceedings of the conference on Process Systems Engineering 2000, Keystone, Colorado, July 2000, Comp. & Chem. Eng., 24 (2-7), pp. 785-791, 2000

[28] Frank, P. M., Enhancement of Robustness in Observer-Based Fault Detection, Preprints of IFAC/IMACS Sympo. SAFETYPROCESS'91, Baden-Baden, pp. 275-287 (Vol.1), 1991

[29] Frank, P. M. , Ding, X. , Survey of Robust Residual Generation and Evaluation Methods in Observer-Based Fault Detection System, Journal of Process Control, 7 (6): 403-424, 1997

[30] Chen, J. , Patton, R. J. , and Zhang, H. Y. , Design of Unknown Input Observer and Robust Fault

Detection Filters, Int. J. Control, 63 (1): 85-105, 1996

[31] Liu, G. P. and Patton, R. J. , Eigenstructure Assignment for Control System Design, John Wieley & Sons, England, 1998

[32] Jazwinski, A. H., Stochastic Process and Filtering Theory, Academic Press, New York, 1970

[33] Ray, A. and Luck, R., An Introduction to Sensor Signal Validation in Redundant Measurement Systems, IEEE Contr. Syst. Mag. 11 (2): 44-49, 1991

[34] Potter, I. E. and Suman, M. C., Thresholdless Redundancy Management with Arrays of Skewed Instruments, Technical Reports AGARDOGRAPH 224 (pp 15-11 to 15-25), AGARD. Integrity in Electronic Flight Control System, 1977

[35] Gai, E., Harrison, J. V. and Daly, K. C., Failure Detection and Isolation Performance of Two Redundancy Sensor Configurations, Proc. of Position Location and Navigation Symposium (PLANS)m San Diago, pp. 122-131, 1978

[36] Desai, M. and Ray, A., A Fault Detection and Isolation Methodology Theory and Application, Proc. 1984 Amer. Control Conf., pp. 262-270, 1984

[37] Delay, K. C., Gai, E. and Harrison, J. V., Generalized Likelihood Test for FDI in Redundancy Sensor Configurations, J. of Guidance, Contr. & Dynamics 2 (1): 9-17, 1979

[38] Ray, A. and Luck, R., An introduction to sensor signal validation in redundant measurement systems, IEEE Contr. Syst. Mag. 11(2): 44-49, 1991

[39] Chow, E. Y. and Willsky, A. S., Analytical Redundancy and the Design of Robust Detection Systems, IEEE Trans. Automa. Contr. AC-29 (7): 603-614, 1984

[40] Lou, X., Willsky, A. S., and Verghese, G. C., Optimally Robust Redundancy Relations for Failure

Detection in Uncertain Systems, Automatica 22 (3): 333-344, 1986

[41] Massoumnia, M. A. and Vander Velde, W. E., Generating Parity Relations for Detecting and Identifying Control System Component Failures, J. of Guidance, Contr. & Dynamics 11 (1): 60-65, 1988

[42] Gertler, J. and Singer, D., A New Structural Framework for Parity Equation-based Detection and Isolation, Automatica 26 (2): 381-388, 1990

[43] Patton, R. J. and Chen, J., Optimal Selection of Unknown Input Distribution Matrix in the Design of Robust Observers for Fault Diagnosis, Preprints of IFAC/IMACS Sympo.: SAFEPROCESS'91, Baden-Baden, pp. 221-226 (Vol. 1). Also published in revised form in Automatica, Vol. 29, No. 4, 837-841, 1991

[44] Isermann, R., Process Fault Detection Based on Modeling and Estimation Methods: A Survey, Automatica 20 (4): 387-404, 1984

[45] Isermann, R., Experiences with Process Fault Detection via Parameter Estimation, in S. G. Tzafestas, M. G. Singh and G. Schmidt (eds), System Fault Diagnostic, Reliability & Related Knowledge-based Approaches, D. Reidel Press, Dordrecht, pp. 3-33, 1987

[46] Isermann, R. and Freyermuth, B., Process Fault Diagnosis Based on Process Model Knowledge, Journal A 31 (4): 58-65, 1990

[47] Isermann, R., Fault Diagnosis of Machine via Parameter Estimation and Knowledge Processing-Tutorial Paper, Preprints of IFAC/IMACS Sympo.: SAFEPROCESS'91, Baden-Baden, pp. 121-133 (Vol. 1). A modified version also published in revised form in Automatica, Vol. 29, No. 4, 815-839, 1991

[48] Isermann, R., Supervision, Fault-detection and Fault-diagnosis Methods – An Introduction, Contr. Eng. Practice 5 (5): 639-652, 1997

[49] Doraiswami, R. and Stevenson, M., A Robust Influence Matrix Approach to Fault-diagnosis, IEEE Trans. Contr. Sys. Techno. 4 (1): 29-39, 1996

[50] Takagi, T. and Sugeno, M., Fuzzy Identification of Systems and Its Applications to Modeling and Control, IEEE Trans. Sys. Man & Cyber. 15 (1):116-132, 1985

[51] Chen,Jie and Patton, Ron J., Robust Model-Based Fault Diagnosis for Dynamic Systems. pp.272-286, 1999

[52] Chen, Jie and Patton, Ron J. Robust Model-Based Fault Diagnosis for Dynamic Systems, 1999

[53] Haykin, Simon Neural Networks: A comprehensive Foundation, Prentice Hall, 1999

[54] Frank, Paul M. and Koppen-seliger, Birgit Fuzzy Logic and Neural Network Applications to Fault Diagnosis, 1996

[55] Patton, R., Lopez-Toribio, C., and Uppal, F. Artificial Intelligence Approaches to Fault Diagnosis for Dynamic Systems. Journal of Applied Mathematics and Computer Science, 1999

[56] Elman, J. L., Finding structure in time, Cognitive Science, vol. 14, pp.179-211, 1990

[57] Gajic, Z. and Lelic, M. Modern Control System Engineering. Prentice Hall Europe, pp. 331-380, 1996

[58] Chen, Jie and Patton, Ron J., Robust Model-Based Fault Diagnosis for Dynamic Systems. pp.35-38, 1999

[59] Demuth, Howard and Beale, Mark, Neural Network Toolbox User's Guide, The MathWorks, Inc., 2004

[60] Elman, J. L., Finding Structure in Time, Cognitive Science, vol. 14, pp. 179-211,1990

[61] Li, J., A. N. Michel, and W. Porod, Analysis and Synthesis of a Class of Neural Networks: Linear Systems Operating on a Closed Hypercube, IEEE Transactions on Circuits and Systems, vol. 36, no. 11, pp. 1405-1422, 1989

[62] Hiroshi Ninomiya, Naoki Kinoshita, A New Learning Algorithm without Explicit Error Back-propagation, International Joint Conference on Neural Networks Proceedings, vol. 3, pp. 1389-1392, 1999

[63] Patton, R. J., Chen, J. and Siew, T. M., Fault Diagnosis in Nonlinear Dynamic Systems via Neural Networks, Proc. Of the IEE Int. Conf.: Control' 94, Peregrinus Press, Conf. Pub. No. 389, Warwick, UK, pp. 1346-1351, 1994

[64] Zhang, Ren, Chen, Jie, Tang, Xiaojing and Yan, Weisheng, A Combined Method Based on Neural Network for Control System Fault Detection and Diagnosis, Proceedings of the 2000 IEEE, International Conference on Control Applications, Anchorage, Alaska, USA, pp104-108, 2000

[65] Berniari, Andrea, Betta, Giovanni, Pietrocento, Antonio and Sansone, Carlo, A Neural Network Approach to Instrument Fault Detection and Isolation, IEEE Trans. On Instrument and Measurement, Vol 44, No. 3, pp747-750, 1995

[66] Simani, S., Fantuzzi, C. and Spina, P.R., Application of a Neural Network in Gas Turbine Control Sensor Fault Detection, Proceedings of the 1998 IEEE, International Conference on Control Applications, Trieste, Italy, pp182-186, 1998

[67] Wang, Hong, Fault Detection and Diagnosis for Unknown Nonlinear Systems: A Generalized Framework via Neural Network, 1997 IEEE International Conference on Intelligent Processing Systems, Beijing, China, pp1506-1510, 1997

[68] Raj K. Aggarwal, Q.Y.Xuan, A.T.Johns, A novel approach to fault diagnosis multicircuit transmission lines using fuzzy ARTmap neural network, IEEE Trans. on Neural Network, Vol.10, No.5, pp1214-1221, 1999

[69] Taniguchi, S., Akhmetov, D. F., Dote, Y. and Ovaska, S. J., Nonlinear Modeling and Fault Detection Using Fuzzy-neural Network, Proc 9th ISCA International Conference on Intelligent Systems, Louisville, KY, pp96-100, 2000

[70] Chow, M. Y., Sharpe, R. N. and Hung, J. C., On the application of artificial neural networks for motor fault detection, IEEE Trans. on Industrial Electronics, Vol.40, no.2, pp181-196, 1993