National Library of Canada
Bibliothèque nationale du Canada

Acquisitions and Bibliographic Services Branch
Direction des acquisitions et des services bibliographiques

395 Wellington Street
Ottawa, Ontario
K1A 0N4

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Canada

NOVEL SCHEME FOR FAULT DIAGNOSIS IN DYNAMIC SYSTEMS USING MULTIPLE UNKNOWN INPUT OBSERVERS

by

Qing Wang

B.S.E.E. Beijing Institute of Technology, Beijing, China, 1985

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE

in the School
of
Engineering Science

© Qing Wang 1993
SIMON FRASER UNIVERSITY
June 1993

All rights reserved. This work may not be reproduced in whole or in part, by photocopy or other means, without the permission of the author.

National Library of Canada

Bibliothèque nationale du Canada

Acquisitions and Bibliographic Services Branch

Direction des acquisitions et des services bibliographiques

395 Wellington Street
Ottawa, Ontario
K1A 0N4

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-91319-3

Canada

APPROVAL

Name: Qing Wang

Degree: Master of Applied Science

Title of thesis: **Novel Scheme for Fault Diagnosis in Dynamic Systems Using Multiple Unknown Input Observers**

Examining Committee: Dr. John Jones, Chairman

Dr. Mehrdad Saif, P. Eng., Senior Supervisor

Dr. Shahram Payandeh, P. Eng., Supervisor

Dr. Bill Gruver, External Examiner

Date Approved: June 25, 1993

PARTIAL COPYRIGHT LICENSE

I hereby grant to Simon Fraser University the right to lend my thesis, project or extended essay (the title of which is shown below) to users of the Simon Fraser University Library, and to make partial or single copies only for such users or in response to a request from the library of any other university, or other educational institution, on its own behalf or for one of its users. I further agree that permission for multiple copying of this work for scholarly purposes may be granted be me or the Dean of Graduate Studies. It is understood that copying or publication of this work for financial gain shall not be allowed without my written permission.

Title of Thesis/Project/Extended Essay

"Novel Scheme for Fault Diagnosis in Dynamic Systems Using Multiple Unknown Input Observers"

Author:

(signature)

Qing Wang
(name)

June 25, 1993
(date)

ABSTRACT

In this thesis, we present a design method for the detection, isolation and identification of multiple actuator failures in linear dynamical systems. This scheme is based on the theory of Unknown Input Observer (UIO).

It is known that a UIO exists under certain necessary as well as sufficient conditions. One of the necessary conditions is that the number of unknown inputs be less than or equal to the number of outputs. Unfortunately, this necessary condition is not always satisfied. In such a case no UIO will exist. On the other hand, it has been shown in the past that variety of actuator failures in a linear system can be modelled as unknown inputs to the system. If this formulation of actuator failures is used, then state estimation using a UIO would be possible only if the number of actuators is less than or equal to the number of system outputs.

In this thesis, this problem has been attacked by special multiple unknown input observers, called MUIOs. It has been shown that by careful formulation of MUIOs not only state estimation is possible, but also identification of multiple actuator failures could be accomplished in certain systems with parameter uncertainties. In addition, the shape and magnitude of the failures can be estimated which is useful in fault accommodation.

Finally, the applicability of the proposed MUIO approach has been illustrated through a numerical example.

To

My Parents

My Wife

and
Margaret, My lovely Daughter

ACKNOWLEDGMENTS

My sincere thanks to my senior supervisor Dr. Mehrdad Saif for his invaluable advice and guidance and time that he spent with me in the preparation of this thesis. I would also like to thank him for his supervision during my studies.

I also would like to thank Dr. John Jones, Dr. Shahram Payandeh and Dr. Bill Gruver for serving on the thesis committee. I am grateful to Mr. Yuping Guan for his help and useful suggestions. Mrs. Susan Stevenson deserves special thanks for her guidance in writing this thesis. I also wish to thank Mrs. Brigitte Rabold for her support. I would also like to thank my fellow students and friends in the School of Engineering Science for their help.

Finally, I acknowledge with humble gratitude, the constant encouragement and support from my parents, my brother and sister, and especially my wife.

This research was supported in part by Natural Sciences and Engineering Research Council of Canada (NSERC), and in part by grants from the Center for Systems Science (CSS) and Simon Fraser University (SFU).

Contents

List of Figures

Chapter 1

Introduction

Fault detection, isolation and accommodation (FDIA) in dynamic systems remains to be of tremendous importance in modern industry and technology, and is receiving increasing theoretical and practical attention. FDIA provides reliability, safety and survivability which are fundamental features in the design of any complex engineering system. Dynamical systems are often subjected to unexpected changes, such as component failures and variations in operating conditions that tend to degrade overall system performance. In particular, failure of actuators or sensors used to provide a feedforward or feedback signal in a control system can cause serious deteriorations in the performance of the system. A fault is normally understood to be any kind of malfunction in the actual dynamic system that leads to an unacceptable system performance. Such malfunctions may occur either in the sensors or actuators (instruments), or in the components of the processes. Any of these failures may lead to unacceptable economic loss or hazards to personnel. In order to maintain a high level of performance, it is important that failures be promptly detected and isolated so that appropriate remedies can be applied.

Generally, a computer controlled system is composed of actuators, a main structure (or process) and sensors. For example, in an airplane flight-control system, the actuators are the servomechanisms which drive the control surfaces and engine which in turn provide the driving thrust. An autopilot controller provides input signals to the actuators. The core part of the plant – the main structure or the process – is the airframe with its cargo and appendages, along with the aerodynamic forces exerted on the control surfaces. The instrumentation consists of several sensors or transducers attached to the airframe, which provide signals proportional to the vital motions of the airframe, including airspeed, altitude, heading, acceleration attitude and rates of change of attitude, control surface deflections and engine thrust. Obviously, sensor signals provide feedback information to the autopilot, but they are also used in the fault monitoring subsystem. In the early 1970's when failure detection theory and its applications were first developed, detection schemes concentrated primarily on detecting sensor failures, which, once detected, could usually be corrected by electronic switching techniques not requiring the reconfiguration of mechanical parts. Compensating for faults in actuators is usually more difficult than redirecting electrical signals. The configuration malfunction in the main structure (process) is even less feasible.

Over the past two decades, fundamental research on failure detection and isolation (FDI) has gained increasing consideration world-wide. This interest is stimulated by the trend towards more complex and the corresponding growing demand for higher availability and security of control systems. Nevertheless, a strong impetus also comes from the area of modern control theory that has brought forth powerful technologies of mathematical modelling, state and parameter estimation that are made feasible by the spectacular progress of modern computer technology. During the same period, numerous approaches to the problem of failure detection and isolation (FDI) in dynamical systems have been developed (Willsky, 1976; Frank, 1987 and 1990; Clark, 1978 a and b; Clark and Setzer, 1980; Clark *et al.*, 1975; Chow and Willsky, 1984; Isermann, 1984; Saif and Villaseca, 1986, 1987; Saif and Guan, 1992; Guan an Saif, 1991; etc.). Among many methods, the detection filter, or observer (state estimator) based methods (e.g. Wang, S. H., 1975; Guan and Saif, 1991) and the generalized likelihood ratio (GLR) method (Willsky and Jones, 1976), and the multiple model method (Willsky *et al.*, 1980) are some examples. We also noticed in recent years the detection and isolation of sensor and actuator failure has received much attention in control theory and its application literature (Clark et al 1975; Willsky and Jones, 1976; Wang, S. H., 1975; Saif and Guan, 1992).

Fault detection and isolation has been widely discussed by many authors. But the remaining task of the monitoring system — the identification of the failure and, most importantly, reconfiguration of system signals in order to maintain satisfactory operation of the system, have not been dealt with extensively. Another issue that has concerned many researcher for many years is that, although sensor failure detection has been dealt with extensively and attention has been given to develop computationally attractive schemes, work on actuator failure detection has not progressed in parallel. Some approaches proposed are computationally tedious (Willsky and Jones, 1976). Actuator failure jeopardizes the whole control strategy, and our focus has been on this specific category. This thesis provides some thoughts on how to detect, isolate, identify and, most importantly, to accommodate failures in actuators and keep the system functioning smoothly.

Chapter 2 begins with a brief review of various steps that need to be taken in tackling FDI problems, followed by brief discussions of three basic types of approaches: the hardware redundancy, knowledge-based (expert system) and the analytical (functional) redundancy methods. The analytical redundancy method is the only approach used in this thesis and hence it attracts the most detailed discussions. Among many analytical approaches, the analytical redundancy method described in this chapter outlines the principles and the most important techniques of model-based residual generation using state estimation methods with attempts to achieve robustness with respect to modelling errors, and finally, the state estimator (observer) scheme which is the main approach to the FDIA problem presented in this thesis work is dealt with in this chapter.

Chapter 3 provides background on the historical development for the design of a single unknown input observer (UIO) to detect, isolate, and identify failures. Various schemes that use unknown input observers or estimators, full order or reduced order are also reviewed. The existence of a single observer in a dynamical system is conditional to the relationship between the total number of its output signals and the total number of its unknown input signals. These unknowns may be modelling errors, disturbances and parameter variations; sometimes other sources of unmeasurable information can be organized into so called "unknown inputs" of dynamical system. The theorem for the existence of a single observer is presented and its limitation with respect to the applications of such an observer to FDIA issues are also discussed in this chapter.

Chapter 4 is the core of this thesis. It presents a novel design scheme for multiple unknown input observers (MUIOs). Unlike the previous work of Guan (1990), the approach presented in this chapter can cope with certain situations where the total number of unknown inputs is greater than the total number of outputs. In this thesis, a robust approach for FDIA in linear systems with parameter variations or uncertain elements is presented. This approach is based on the fact that the model of a linear dynamical system under plant parameter variations and uncertainties can be transformed to make the application of the UIO theory for state estimation purposes possible. The design scheme of MUIOs for the linear dynamical system under such parameter variations and uncertainties are taken into account. The proposed fault-tolerant control system is based upon MUIOs that can estimate unavailable state variables of the system at the same time for the purpose of control. The limitation of the proposed scheme is also discussed in this chapter. This chapter also discusses how the above design method is used for failure detection and isolation for actuators and provides the corresponding results. It also describes detailed system accommodation techniques that enable us to keep the dynamic system, which is subject to system uncertainties or parameter variations, functioning smoothly subsequent to an actuator failure. Systematic detection methodology along with the system reconfiguration technology used in this chapter could be implemented in a real-time dynamical control system.

In Chapter 5, the MUIO design scheme and the FDIA techniques which are developed in Chapter 4 are applied to a linear, time-invariant, dynamic system which describes the longitudinal dynamics of the F18 High Alpha Research Vehicle (F18/HARV). The discussion of this application demonstrates the usefulness of MUIO design methodology and the FDIA scheme. The results of simulations indicate that the scheme for fault diagnosis in dynamical systems using multiple unknown input observers can detect, isolate, and accommodate multiple actuator failures under the existence condition of MUIO.

Conclusions and future work are presented in Chapter 6. The contributions of this thesis work are also summarized here.

Chapter 2

Approaches to FDIA Problem

Over the past few decades numerous approaches to the problem of failure detection, isolation and identification in dynamical systems have been reported. In general, there are three major approaches: the hardware redundancy method, the knowledge-based method, and the functional or analytical redundancy method. The following sections describe each of these schemes.

2.1 Hardware Redundancy Method

Traditionlly, fault diagnosis in dynamical systems is conducted through the use of *hardware redundancy.* Repeated hardware elements (actuators, measurement sensors, process components, etc.) are usually distributed spatially in the system to prevent localized damage. Such methods typically function in a set of three (triplex) or a set of four (quadruplex) redundancy configurations, and they compare redundant signals for consistency. Consider the case of sensing as an example. The idea is that three (or more) sensors measuring the same variable are installed where one would be sufficient if it were completely reliable. Signals from these sensors are monitored by a logic circuit which ignores small differences in signals due to electronic noise, manufacturing tolerance, and the monitoring error inherent in the instrument. This monitoring device declares that a sensor is faulty if its signal deviates too far from the average value of others (assuming that only one fails at a time and the difference among the others remains small). This fault-tolerance approach is simple and straightforward and is therefore widely utilized. It is essential in the control of airplanes, space vehicles and in certain process plants which are safety-critical such as nuclear power plants handling dangerous chemicals.

Major problems encountered in using the hardware redundancy approach are that first they require extra cost for the redundant hardwares and that second they take extra spaces and weights. In aircraft, the additional room could be used for more mission-oriented equipment. Another limitation of this approach is that it has been realized that since redundant components (sensors) tend to have similar life expectations, it is more likely that when one of a set of sensors malfunctions others will soon become faulty as well.

2.2 Knowledge-Based (Expert System) Method

Knowledge-based (expert system) methods complements existing analytical (see Section 2.3) and algorithmetical methods of fault detection; they open a new dimension of possible fault diagnosis for complex process by adding new algorithms to improve the process. In recent years, attempts have been made to apply artificial intelligence and knowledge-based techniques which combine numeric and symbolic methods for performing fault diagnosis. Research on computer-based automated diagnosis is receiving increasing attention and currently available numeric and non-numeric (symbolic) tools are already sufficient for developing practical systems for on-line and off-line automated diagnosis and supervision of electronic, mechanical, chemical, aerospace and other devices and processes. Knowledge-based technology has now reached the level of full-scale, efficient, productive utilization in industrial systems and other complex modern life systems.

In this section, the knowledge-based approach to fault diagnosis is briefly discussed and its advantages and drawbacks are also given. A knowledge-based expert system is designed using artificial intelligence (AI) techniques, emulating human performance and presenting a human-like action to the user. Expert systems are currently finding applications in an increasing repertory of human-life domains, at the center of which lies fault diagnosis and the repair or reconfiguration of technological processes.

Fault diagnosis and reconfiguration are knowledge-intensive, experiential tasks which in reality could sometimes go beyond the capabilities of skilled technicians, operators, or engineers. Expert systems can perform at least at the level of a highly experienced human trouble-shooting/repair expert whose knowledge greatly exceeds the contents of service manuals. This expert system provides the critically required assistance for prompt detection, location and repair of process faults and improves overall field service efficiency and performance. The field of system diagnosis/repair is presently at the heart of industrial automation and has all the required characteristics (closed domain, rich expertise available, good underlying models, heuristic methods, readily performed test/validation procedures) that make expert systems very likely to succeed in industrial environments.

The main advantages of using a knowledge-based (expert system) method to solve the problem of fault diagnosis are:

1. They provide a homogeneous representation of knowledge;
2. They allow incremental growth of knowledge about faults through addition of reasoning processes;
3. They allow unplanned but useful interactions.

On the other side of the coin, knowledge-based diagnosis methods have their own drawbacks:

1. A great deal of prior knowledge of the system is necessary; and sometimes we can obtain only limited knowledge;
2. The knowledge acquisition from the domain expert is time consuming and difficult;
3. All possibilities have to be explicitly enumerated and there is no capability for system generalization.

An example of available knowledge-based diagnosis (supervision) systems is called LATEST, which was developed by IBM (contracting with GHC Corp. and funded by NASA) for troubleshooting the Space Shuttle launch countdown (Wood *et al.*, 1989).

LATEST is a rule-based expert system written in Ada language that gives the reason for a hold or abort within about three seconds. Occasionally during the launch of the Space Shuttle, an abnormality demands a detailed explanation of how the general purpose computers and their programs reacted to particular inputs. Nevertheless, the software responses to a hold or abort command in a fraction of a second and may leave even the experts puzzled as to the exact sequence of events leading to the interpretation of the countdown. Manual analysis of the data often takes hours since experts must print out the data and find anomalies by comparing data blocks to computer program design documents. This is an expensive use of manpower. Automated diagnosis technologies of countdown failures, such as LATEST — which provides a cost-effective launch countdown anomaly tool using expert system technology — have become essential given the frequency of Shuttle missions returning to normal after the Challenger disaster with up to 10 flights scheduled each year.

Methods of diagnosing/detecting failures in industrial systems based on hardware redundancy and knowledge do not need any mathematical model of the plant (although some expert system approaches are more or less model based). The third major scheme applies to the problem of FDIA is the analytical redundancy method which will be presented in full detail in the next section. My thesis project focus is on this category of failure detection, isolation and identification and on the design method we developed that is based on state estimation (observer scheme) in dynamical systems.

2.3 Analytical (Functional) Redundancy Method

In this section we outline the general procedure of FDIA using analytical redundancy. The procedure for evaluating redundancy information given by a mathematical model of the system can be roughly divided into the following two steps: residual generation and decision making. We will describe these two steps in depth. The schematic of the FDIA procedure using analytical redundancy is illustrated in Fig. 2.1.

2.3.1 Analytical Redundancy Method

In the course of developing the basic research on FDI, a novel philosophy for the FDI methodology has emerged and is increasingly discussed in the research literature. It is based on the use of *analytical* (i.e. *functional*) rather than *physical* or *hardware* redundancy. This method implies that the inherent redundancy contained in the static and dynamic relationships among the system inputs and the measured outputs is exploited for FDIA. In other words, one makes use of a *mathematical model* of the system or of a part of it for generating redundant information for FDIA purpose. Contrary to the hardware redundancy approach, where redundancy measurements from each sensor are compared, values of estimated variables of sensor measurements are used as redundant information for fault detection purposes. As opposed to the previous approaches discussed, analytical redundancy approach requires advanced information processing technology such as state estimation, parameter estimation, adaptive filtering, variable threshold logic, and some more sophisticated approaches such as cost functions or statistical tests, and various logical operations, all of which can be implemented on a digital computer.

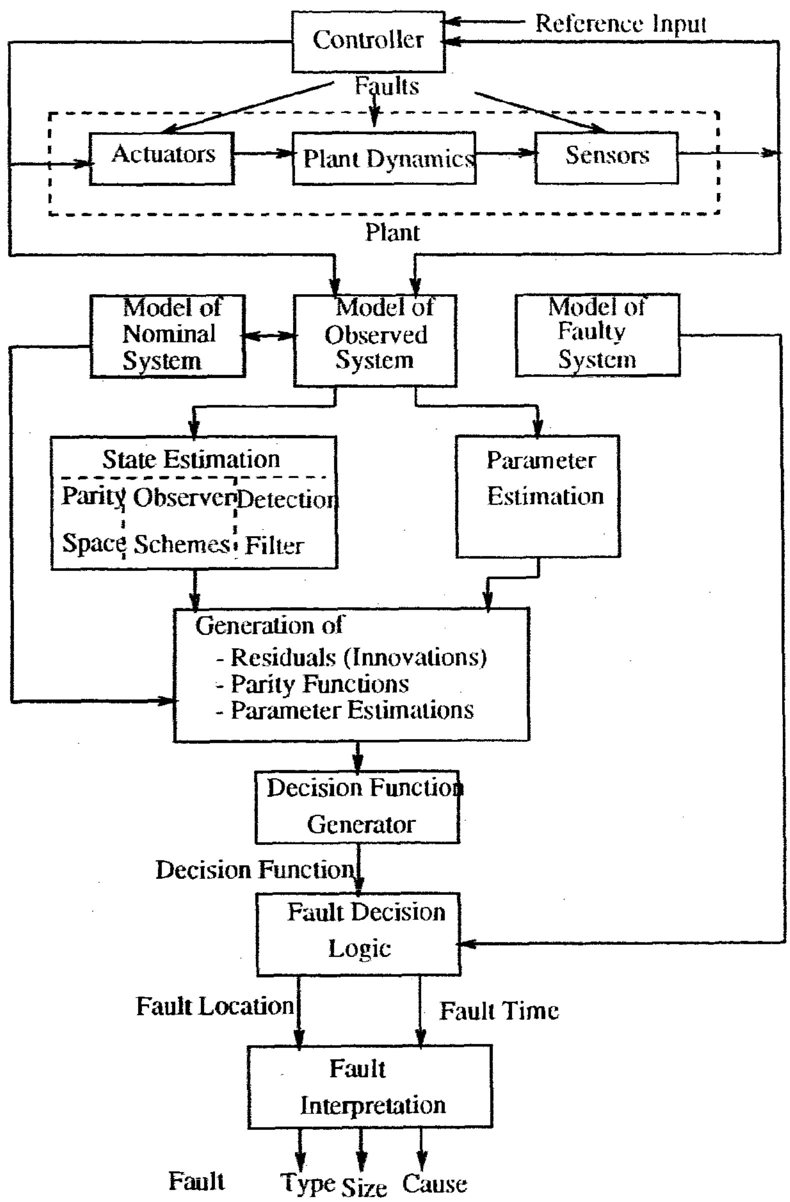

Figure 2.1: General architecture of FDI based on analytical redundancy

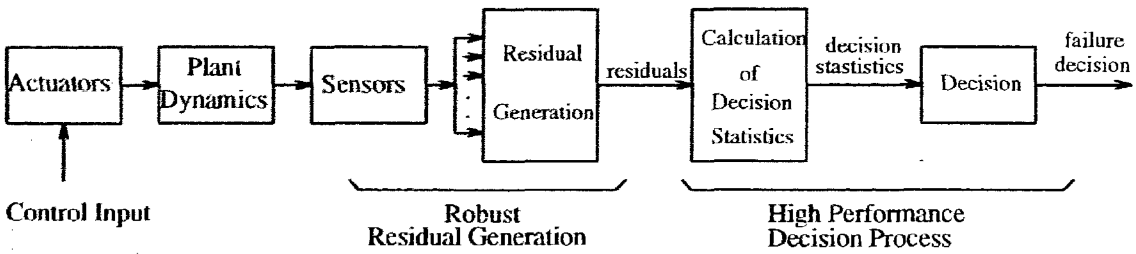

Figure 2.2: Two-stage structure of the FDI process

The first essential process of failure detection, isolation and identification is the generation of the so-called *residuals*. Often these residuals are generated by first estimating certain outputs or variables of the system, and obtaining the estimation error which is commonly referred to as the estimator's residual, or simply the residuals. For a particular set of hypothesized failures, an FDI system has the structure shown in Fig. 2.2.

Output from sensors are initially processed to enhance the effect of a failure (if present) so that it can be recognized. An enhanced failure effect on the residual is called the *signature* of the failure. Residuals should be unbiased in the absence of a failure, showing agreement between the observed and expected normal behavior of the system; a failure signature typically takes the form of residual bias which is a characteristic of the failure. Thus, residual generation is based on knowledge of the normal behavior of the system. The actual process of the residual generation varies in their complexity. For example, in some voting systems, residuals are simply the difference between outputs of various like sensors, whereas in a GLR test scheme (Willsky and Jones, 1976), residuals are innovation process of the Kalman filter.

In the second stage of an FDI algorithm, the decision-making process, residuals are examined for the presence of failure. Decision functions or statistics are calculated using residuals, and a decision rule is then applied to the decision statistics to determine if any failure has occurred. A decision process may consist of a simple threshold test on instantaneous values or the moving average of residuals, or it may be based directly on methods of statistical decision theory and the sequential probability ratio test (Willsky, 1976). More specifically, decision making consists of the following tasks:

1. *Failure detection;* i.e. an indication that something is going wrong in the system;
2. *Failure isolation/identification;* i.e. determination of a faulty component and determination of the size and/or the shape of the fault and its removal from the system;
3. *Failure accommodation;* i.e. reconfiguration of the system so that it can continue to function without interruption.

The idea to replace hardware redundancy by analytical redundancy was originated by Beard (Beard, 1971) and Meier *et. al.* (Meier, 1971). Beard developed methods of self-recognization to maintain closed-loop stability. Such issues as identifying failures and changes in system sensors were solved by comparing the outputs of observers. Meier *et. al.* investigated the usefulness of functional redundancy to detect aircraft control data instrument failures. The functional redundancy was obtained from functional relations that existed among different measurements. These were checked for consistency with the aid of two Kalman filters and several algebraic relations.

An innovation test using a single Kalman filter was proposed by Mehra and Peshon in 1971 (Mehra and Peshon, 1971). In their method, an innovation sequence was generated and subjected to statistical tests of whiteness, mean, and covariance. Knowing the time history of the output variables under normal conditions allows for the detection of any deviations by statistical decision theory. This approach is only capable of detecting failures. But the failures can not be isolated. To do this, more advanced techniques such as the M-ary hypothesis testing, etc. were developed.

More software expenditure is needed for failure accommodation using Baysian decision theory as proposed by Montogomery and Caglayan in 1976 (Montogomery and Caglayan, 1976). They provide a bank of parallel Kalman filters designed for a set of *M-1* (*M* is the number of outputs) possible failure modes and for normal operation. The erroneous instrument is detected with the aid of M-array hypothesis testing. A moving window of the innovation of each Kalman filter drives a detector that calculates the likelihood ratio for each hypothesis corresponding to a possible failure mode.

To relax the computation complexity of techniques such as the M-ary hypothesis testing and GLR test discussed above, Deskert *et. al.* (1977) presented a functional redundancy scheme combined with dual sensor redundancy of the process. The identification of the failure is achieved on the basis of functional relationships among outputs of dissimilar instruments by performing sequential probability ratio tests of differences among outputs. Similar work had been done by Onken and Stuckenberg (Onken and Stuckenberg, 1979). By using dual sensors and state estimators for the generation of analytical redundancy, they obtained the quality of a triplex system. Other schemes using the Kalman filters for the analytical redundancy include Cunnimgham and Poyneer, 1977; Montgomery and Tabak, 1979.

Several contributions to IFD with state estimation methods using either an observer or Kalman filter were made by Clark (Clark, 1977, 1978, 1980 and Clark *et.*

al., 1975). In 1977, Clark proposed the dedicated observer scheme (DOS) for IFD using a bank of Luenberger observers, each driven by one sensor output. If none of the sensors fails, all reconstructed state vectors converge to the actual state vector. However, if one of the sensor fails, then a difference occurs in the output vector of the corresponding observer. The difference can be used to identify the faulty sensor. A simplified IFD scheme was also introduced by Clark in 1978 by using a single observer driven by one of the measured variables. If all sensors work perfectly, no difference will be seen between reconstructed outputs and actual instrument outputs. If one of sensors that does not drive the observer fails, there will be a reconstruction error in corresponding channels. But, if a sensor failure occurs in the channel that drive observers, all reconstructed outputs will be erroneous. Therefore, a unique means of detecting and isolating of the faulty sensor is possible. Since the introduction of DOS by Clark, other more sophiscated aproaches based on it have been proposed (Frank, 1987, 1990; Saif and Villaseca, 1986, 1987 a, b).

In 1979, Shapiro and Decarli developed an analytical redundancy scheme for the flight control sensors of the Lockheed L-1011 aircraft. They used Luenberger observers to reconstruct signals of failed sensors from associated unfailed sensors. Instead of using a set of observers for each failure configuration, they used an observer that is driven by the airframe input and the output of the sensor with the highest reliability.

In order to deal with the IFD problem in the presence of random disturbances as well as to increase the robustness of observer schemes, several authors have proposed some schemes using a Kalman filte. Clark and Setzer (1980) proposed to modify the simplified IFD by using a Kalman filter with modified detection logic.

In the instrument fault detection (IFD) and actuator fault detection (AFD) schemes so far described, errors of reconstructed states that are used for IFD and AFD are affected by sensor malfunction, actuator malfunction, and variations of the process parameters. Frank and Keller (1980) developed an observer design scheme in which insensitivity to parameter variations as a design specification was first included. They extended dedicated observer (DOS) schemes by duplicating observers to allow distinction between the parameter variation and instrument malfunctions. One observer in each pair is designed to be insensitive to parameter variations but sensitive to instrument malfunctions, and the other is insensitive to both. This method is robust with respect to parameter variations and applies to single output systems as well as multiple output systems.

A general approach to generating robustness in failure detection and isolation systems has been pursued by Willsky and others (Deckert *et. al.*, 1977; Chow, 1980; Leininger, 1981; Lou, 1982; Chow *et. al.* , 1984 and Lou *et. al.* , 1986), and by Watanabe (Watanabe *et. al.* , 1981, 1982). They researched the problem of robust residual generation from the viewpoint of analytical redundancy relations and have introduced the concept of general parity equation checks. They then considered innovations of an observer or a Kalman filter as the most general residual containing the complete set of redundancy relations. The underlying idea of robustness generation is now to utilize only those redundancy relations for FDI that are most reliable.

Another important approach to increasing the robustness of observer schemes by using a "robust" or so-called "unknown input" observer was recently dealt with by Wünnenberg and Frank (1986), Guan and Saif (1991) and Saif and Guan (1992). Saif and Guan proposed a novel scheme of robust estimation with application to failure detection and identification in dynamical systems (Saif and Guan, 1993). The system being dealt with is subject to plant parameter variations or uncertainties in the system, and this fault tolerant control system is based on a single robust estimator that can simultaneously estimate unmeasurable state variables of the system for the purpose of feedback control. The available results provide the necessary information for a detection logic device capable of detecting and isolating actuator and sensor failure. Additionally, Saif and Guan's scheme is also able to identify the exact shape and magnitude of the failure. The essence of this method was the robust observer design scheme along with the necessary and sufficient condition for the existence of a single observer.

While most of the work on FDI is concerned with instrument (sensor) failures, some attention has been given to component failures and actuator failures in dynamical systems. One of crucial issues of component failure detections (CFD) is the problem of failure *isolation* which is much more complex than the isolation of failed instruments. The actuator failure in the control system jeopardizes the entire control strategy and recent work has been done on restructurable control strategies for maintaining stability and performance in the presence of these failures (Athans, 1982).

The advantage of the analytical redundancy approach lies in the fact that the existing redundancy can simply be evaluated by information processing under well-featured operating conditions (i.e. at the operation center) without the need of additive physical instrumentation in the plant. Although a price, which arises from the need for the mathematical model, has to be paid for this benefit, considerably less computational expenditure is required for on-line modelling of the process with the assistance of modern computer technology. Our focus is on this category of actuator failure detection and isolation, and also on the design method we developed that is based on state estimation (an observer scheme).

2.4 Summary

In this chapter, we have discussed the various approaches for FDIA in dynamic systems. It has been pointed out that there are many techniques and very elaborate procedures ready for application. Simulation studies and experimental results have shown that the FDIA schemes using analytical redundancy have reached a certain degree of maturity. There are, in particular, a number of encouraging results in the application to mechanical systems such as aircraft or advanced transport systems. It should be noted, however, that in cases where only poor or imprecise analytical models are available, such as in chemical plants, the model-based FDIA approach is still problematical. In such cases the support by knowledge-based methods may be unavoidable.

Finally, one can see that the question of application of any model-based FDIA scheme is primarily a question of the quality of the available mathematical model of the system. Additionally, the reachable quality of fault isolation decisively depends on the number of available measurements.

Chapter 3

Unknown Input Observer: Theory and Design

This chapter describes a unified method for the design of a robust observer scheme for sensor, actuator, and component fault detection, isolation and identification in dynamic systems. This method focuses on the problem of residual generation with the goal of providing effective discrimination between different faults in the presence of unknown inputs such as system disturbances, modelling uncertainties, process parameter variations and measurement noises. The approach is based on the theory of the unknown input observer (UIO) which provides complete fault decoupling and the modes of faults and disturbances. In this chapter we will focus our attention on the first stage of FDIA, i.e., the process of residual generation using state estimation techniques with emphasis on robustness with respect to unknown input. As mentioned earlier, a few algorithms have been proposed for the design of full order or reduced order unknown input observers to achieve better results of an observer-based FDIA scheme by providing increased robustness with respect to unknown inputs (Yang and Wilde, 1988; Viswanadham and Ramakarishna, 1980; Gopinath, 1971; Viswanadham and Srichander, 1987; Kudva, Viswanadham and Ramakarishna, 1980; Wang, Davison and Dorato, 1975; Kurek, 1983).

The crucial point in any model-based FDIA scheme is the influence of unmodelled disturbances such as system uncertainties, changes in the system parameters, and system and measurement noises. These influences can be summarized as unknown inputs acting on the system. The effects of unknown inputs hinder the performance of fault detection, isolation and identification scheme and act as a source of false alarms. Therefore, in order to minimize the false alarm rate, one should design the observer such that it becomes robust with respect to unknown inputs. The first essential step in the development of an observer-based FDIA scheme is a realistic representation of the physical process under consideration, which includes system dynamics, faults and all kinds of possible unknown inputs. The resulting mathematical equation, the state space equation, then serves as a basis for the mathematical derivation of the FDIA procedure described here. Residual generation using an observer-based (full-order or reduced-order) method is presented in this chapter, followed by the design of unknown input observer. This chapter also describes how a UIO can be built with the presence of unknown inputs. A major result of the derivation is a necessary and sufficient condition for the existence of an unknown input fault detection observer.

3.1 System Specification and Problem Formulation

Since as described in Chapter 2 the achievable quality of the FDIA scheme mainly depends upon the quality of the system's model, it is important to start with a thorough and realistic specification of the given process. Such a specification will be the basis for the later fundamental solution of the FDIA problem. We consider the linear, time-invariant, dynamical system (i.e. the plant in a feedback control system), as shown in Fig. 3.1. In general, the system consists of actuators, the plant dynamics (components) and sensors. For a realistic and thorough representation with respect to later use in the FDIA task, it is important to model all effects that can lead to alarms and false alarms. Such effects are:

(a) Faults in the actuators, the components and the sensors of the plant dynamics;

(b) Modeling errors between the actual system and its mathematical model;

(c) System noises and measurement noises.

Fig. 3.2 shows the simplified block representation where all faults are represented by a fault vector $\mathbf{f}$ and all the other effects such as modelling errors, system noises and measurement noises that obscure the fault detection are represented by the so-called vector of unknown inputs, $\mathbf{v}$.

As discussed in Chapter 2, actuator failures in dynamic systems impede the proper function of the system. Therefore, solving this problem requires special effort. We will also pay attention to the issues of uncertainties and parameter variations in the system. In this thesis, we consider the time-invariant, linear, dynamic system, assuming that all sensors in the system are free of failures. Such a system can be

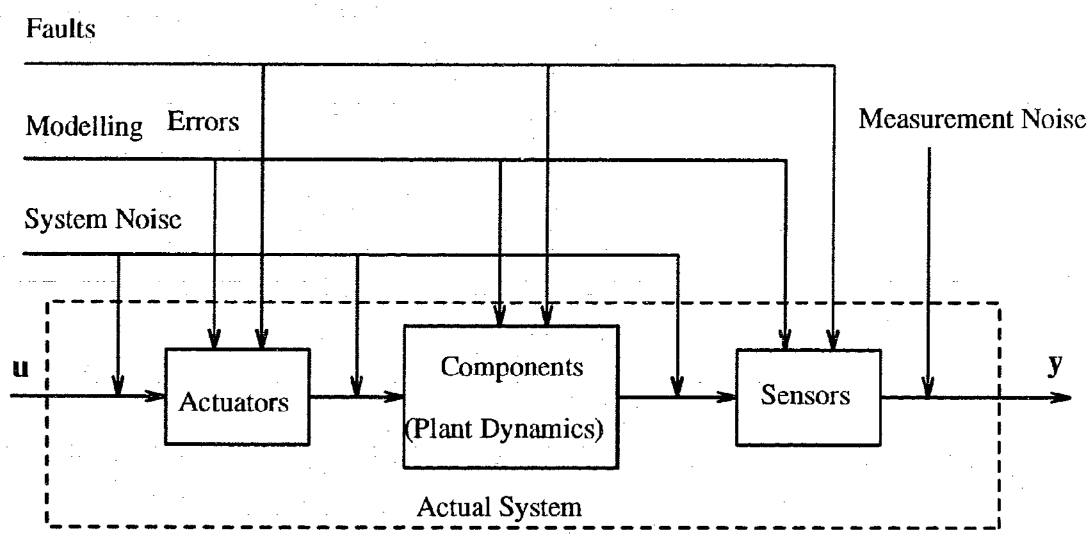

Figure 3.1: System Representation

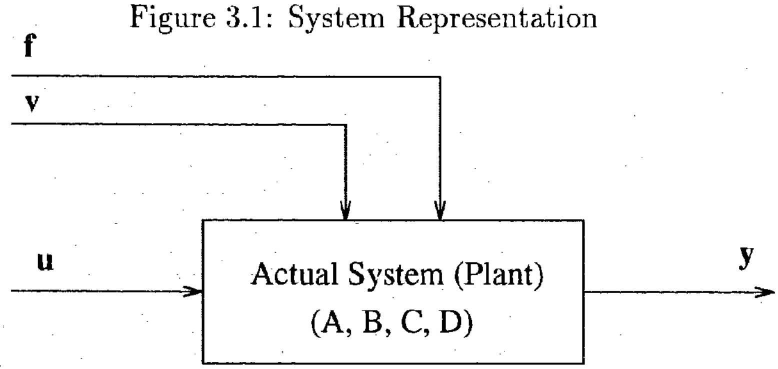

Figure 3.2: Simplified Block Representation of the System

expressed in the following state-space form

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{D}\mathbf{v} \tag{3.1}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} \tag{3.2}$$

where $\mathbf{x} \in \Re^n$ is the state vector, $\mathbf{u} \in \Re^q$ is the known input vector, $\mathbf{v} \in \Re^m$ is the unknown input, which can be treated as the effect of actuator failures, system uncertainties and parameter variations and higher order terms in case of linearizing a nonlinear system, etc, and $\mathbf{y} \in \Re^p$ is the measurable output vector. **A, B, C** are the known matirces of appropriate dimensions. Notice that **A, B, C** are the nominal matrices of the system. Faults that are principally reflected in the changes of **A, B, C** and modelling errors are considered by **v** associated with the proper chioce of **D**. While theses matrices **A, B, C, D** are usually given, the modes (i.e. the evolution) of **v** must generally be considered as unknown.

The fault modes of the system might be classified as:

(1) Abrupt (sudden) faults, for example, step-like changes;

(2) Incipient (slowly developing) faults, for example, bias or drift.

Typically, an abrupt fault plays a role in safety-related systems where hardware failures have to be detected early enough so that catastrophic consequences can be avoided by prompt system reconfiguration. It also keeps the system functioning smoothly. On the other hand, incipient faults are of major relevance in connection with maintenance problems where early detection of worn equipment is required. In this category, faults are typically small and not easy to detect. In this thesis, we will deal with the FDIA issue of both types of faults in actuators. The uncertainties and parameter variations of the system will also be taken into account. The following section describes the design of an unknown input observer (UIO).

3.2 Unknown Input Observer(UIO) Design

In this section, we will focus on the issue of the state estimation for the system with unknown inputs. First, the definition of unknown input observer is given, then the condition for the existence of a single unknown input observer will be described, and finally, the detailed procedure for designing a UIO will be described.

3.2.1 Definition of UIO

The unknown input observer is defined as follows.

Definition: A dynamic system:

$$\dot{\mathbf{w}} = \mathbf{F}\mathbf{w} + \mathbf{E}\mathbf{y} + \mathbf{L}\mathbf{u} \tag{3.3}$$

$$\hat{\mathbf{x}} = \mathbf{w} + \mathbf{N}\mathbf{y} \tag{3.4}$$

is called an *unknown input observer (UIO)* of the system described in equations (Eq. 3.1 - 3.2), if $\|\mathbf{x}(t) - \hat{\mathbf{x}}\| \longrightarrow 0$ as $t \rightarrow \infty$. Matrices **F, E, L** and **N** have appropriate dimensions and **w** is the *(n-p)*-dimensional state vector of the estimator and $\hat{\mathbf{x}}$ is the estimate of the state **x**. Detailed procedures for finding suitable matrices **F**, **E**, **L** and **N** and problems associated with the procedure is given in (Saif and Guan, 1992).

3.2.2 Necessary Condition for the Existence of a Single UIO

Consider the system given in (Eq. 3.1 - 3.2). We make the following assumptions: (1) the **C** matrix has a special form which is given by $C = [\mathbf{0} \quad \mathbf{I}]$; [1](2) **D** is of full rank; (3) the sensors are all healthy.

As mentioned earlier, a few algorithms have been proposed for the design of a single full-order or reduced-order UIO (Yang and Wilde, 1988; Viswanadham and Ramakarishna, 1980; Gopinath, 1971; Viswanadham and Srichander, 1987; Kudva, Viswanadham and Ramakarishna, 1980; Wang, Davison and Dorato, 1975; Kurek, 1983; Guan and Saif 1991). Although the design of UIOs in the above literature varies, the condition for the existence of a single UIO in their schemes is essentially the same. The following theorem presents the necessary condition for the existence of a single UIO.

THEOREM A: A necessary condition for the existence of any order observer for the system (Eq. 3.1 - 3.2) is that:

1. The total number of unknown inputs is less than or equal to the total number of outputs, that is, $m \leq p$; which implies
2. Rank(**CD**) = m, which in our formulation implies
3. Rank($\mathbf{D_2}$) = m.

[1]Note: This is not a restrictive assumption because as long as C is of full rank, there will always exist a similarity transformation that if performed on the system will bring the matrix C of the transformed system in this special structure.(Chen, 1984)

where

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}_1 \\ \mathbf{D}_2 \end{bmatrix} \tag{3.5}$$

$\mathbf{D}_1$ is a (n-p)×m, and $\mathbf{D}_2$ is p×m matrices. The proof of this theorem is given in (Saif and Guan, 1992).

The remainder of this section, describes in detail the procedures for designing a single UIO, but only the final results of the design scheme proposed by Guan and Saif (1991) are presented. It is assumed that existence conditions for the system (Eq. 3.1 - 3.2) are satisfied. There are two cases discussed in their method:

1. p>m, the eigenvalues of the observer can be freely chosen;
2. p=m, the eigenvalues of the observer are fixed.

3.2.3 Case 1: p>m

In this case, the system that the UIO is based on is as follows:

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \mathbf{A}_3 \end{bmatrix} \mathbf{x} + \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \\ \mathbf{B}_3 \end{bmatrix} \mathbf{u} + \begin{bmatrix} \mathbf{D}_1 \\ \mathbf{D}_2 \\ \mathbf{D}_3 \end{bmatrix} \mathbf{v} \tag{3.6}$$

$$\mathbf{y} = [\mathbf{0} \quad \mathbf{I}]\mathbf{x} \tag{3.7}$$

The Observer is given by:

$$\dot{\mathbf{w}} = \mathbf{F}\mathbf{w} + \mathbf{E}\mathbf{y} + \mathbf{L}\mathbf{u} \tag{3.8}$$

where

$$\mathbf{G} = \bar{\mathbf{A}}_{12} - \mathbf{M}\bar{\mathbf{A}}_{22} - \bar{\mathbf{A}}_{11}\mathbf{M} - \mathbf{M}\bar{\mathbf{A}}_{21}\mathbf{M} \tag{3.9}$$

$$\mathbf{H} = \bar{\mathbf{A}}_{13} - \mathbf{M}\bar{\mathbf{A}}_{23} + (\bar{\mathbf{A}}_{11} - \mathbf{M}\bar{\mathbf{A}}_{21})(\mathbf{D}_1 - \mathbf{M}\mathbf{D}_2)\mathbf{D}_3^{-1} \tag{3.10}$$

$$\mathbf{F} = \bar{\mathbf{A}}_{11} - \mathbf{M}\bar{\mathbf{A}}_{21} \tag{3.11}$$

$$\mathbf{E} = [\mathbf{G} \qquad \mathbf{H}] \tag{3.12}$$

$$\mathbf{L} = \bar{\mathbf{B}}_1 - \mathbf{M}\bar{\mathbf{B}}_2 \tag{3.13}$$

here $\mathbf{M}$ is the observer's gain and

$$\bar{\mathbf{A}}_i = \mathbf{A}_i - \mathbf{D}_i\mathbf{D}_3^{-1}\mathbf{A}_3 \tag{3.14}$$

$$\bar{\mathbf{B}}_i = \mathbf{B}_i - \mathbf{D}_i\mathbf{D}_3^{-1}\mathbf{B}_3 \tag{3.15}$$

$$\bar{\mathbf{A}}_i = [\bar{\mathbf{A}}_{i1} \quad \bar{\mathbf{A}}_{i2} \quad \bar{\mathbf{A}}_{i3}] \tag{3.16}$$

From (Eq. 3.8) and (Eq. 3.11), it is easy to see that the necessary and sufficient condition for a stable observer to exist is given in the following theorem.

THEOREM B: The necessary and sufficient condition for the existence of an observer capable of estimating the states of the dynamical system given in (Eq. 3.1 - 3.2) is that the pair $\{\bar{\mathbf{A}}_{11}, \bar{\mathbf{A}}_{21}\}$ given in (Eq. 3.11) must be completely observerable. In addition, if the above condition is satisfied, then the eigenspectrum of the closed loop observer can be assigned arbitrarily as long as complex conjugate eigenvalue appear in pairs. The proof of this theorem is given in (Saif and Guan, 1992). Then the estimate of the state is:

$$\hat{\mathbf{x}} = \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} \mathbf{w} + \begin{bmatrix} \mathbf{N} \\ \mathbf{I} \end{bmatrix} \mathbf{y} \tag{3.17}$$

where

$$\mathbf{N} = \begin{bmatrix} \mathbf{M} & (\mathbf{D}_1 - \mathbf{M}\mathbf{D}_2)\mathbf{D}_3^{-1} \end{bmatrix} \tag{3.18}$$

3.2.4 Case 2: $\mathbf{m} = \mathbf{p}$

In this case, the system which the UIO based on is as follows:

$$\dot{\mathbf{x}}_1 = \mathbf{A}_{11}\mathbf{x}_1 + \mathbf{A}_{12}\mathbf{y} + \mathbf{B}_1\mathbf{u} + \mathbf{D}_1\mathbf{v} \tag{3.19}$$

$$\dot{\mathbf{y}} = \mathbf{A}_{21}\mathbf{x}_1 + \mathbf{A}_{22}\mathbf{y} + \mathbf{B}_2\mathbf{u} + \mathbf{D}_2\mathbf{v} \tag{3.20}$$

The observer is given by:

$$\dot{\mathbf{w}} = \mathbf{G}\mathbf{w} + \mathbf{H}\mathbf{u} + \mathbf{L}\mathbf{y} \tag{3.21}$$

The estimate of the state is:

$$\hat{\mathbf{x}}_1 = \mathbf{w} + \mathbf{N}\mathbf{y} \tag{3.22}$$

where

$$\mathbf{N} = \mathbf{D}_1\mathbf{D}_2^{-1} \tag{3.23}$$

$$\mathbf{G} = \mathbf{A}_{11} - \mathbf{N}\mathbf{A}_{21} \tag{3.24}$$

$$\mathbf{L} = \mathbf{A}_{12} - \mathbf{N}\mathbf{A}_{22} + \mathbf{G}\mathbf{N} \tag{3.25}$$

$$\mathbf{H} = \mathbf{B}_1 - \mathbf{N}\mathbf{B}_2 \tag{3.26}$$

Given above equations, it can be seen that the eigenspectrum of the observer can not be arbitrarily assigned and it is fixed by $\mathbf{G}$ (See Saif and Guan (1992) for more details).

3.3 Residual Generation Using State Estimation

It is well known from observer theory that a linear or nonlinear, full-order or reduced-order state observers (in deterministic cases) or Kalman filters (when noise is considered)can be used for state estimation. In either case a mathematical model of the process is involved. The standard observer-based residual generation configuration for the case of a full-order observer is given in Fig. 3.3. In this figure, the observer is driven by the input and output signals of the system. The estimate of state variables is $\hat{\mathbf{x}} \in \Re^n$ and the estimate of the measurable output vector is $\hat{\mathbf{y}} \in \Re^p$.

The key problem of the observer-based fault detection, isolation and identification procedure is the generation and evaluation of a set of residuals, which permit not only the detection but also a unique distinction (the location, and most importantly, the size and the shape of the fault) between different faults in the face of an unknown input. Generally, this goal can be achieved by a bank of observers or an observer scheme (e.g. a UIO scheme), where each observer is made sensitive to a different fault or a group of faults and insensitive to unmodelled disturbances, noises, modelling uncertainties and process parameter variations. The idea of residual generation via state estimation is to reconstruct the state variables and outputs of the process and to use the estimation error or innovation, or some functions of them as residuals. Residual generation using a full-order observer is briefly described as follows.

Consider a system in the form of state equations (Eq. 3.1 - 3.2). The state estimate $\hat{\mathbf{x}}$ and the output estimate $\hat{\mathbf{y}}$ of a full-order observer obey the equations:

$$\dot{\hat{\mathbf{x}}} = (\mathbf{A} - \mathbf{MC})\hat{\mathbf{x}} + \mathbf{Bu} + \mathbf{My}, \quad \hat{\mathbf{x}}_0 = \hat{\mathbf{x}}(0) \tag{3.27}$$

$$\hat{\mathbf{y}} = \mathbf{C}\hat{\mathbf{x}} \tag{3.28}$$

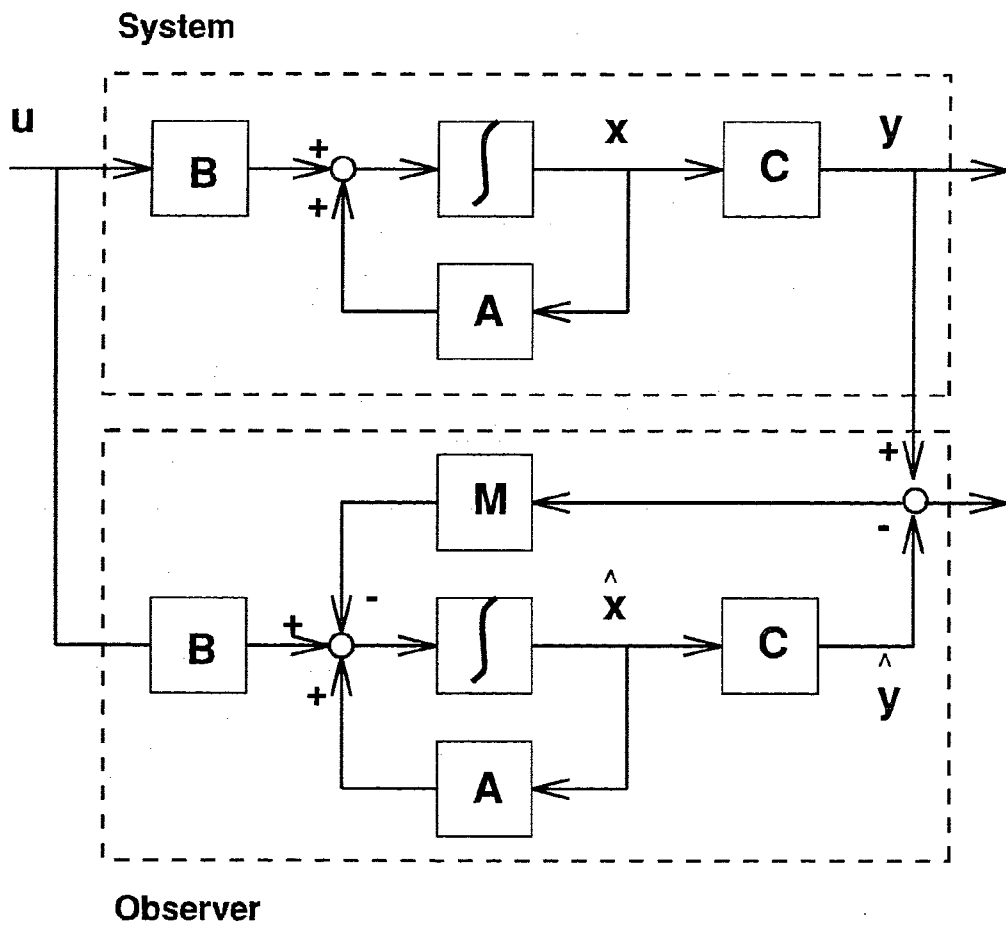

Figure 3.3: Residual Generation with full-order observer

where $\mathbf{M}$ denotes the observer feedback gain. With equations (Eq. 3.1 - 3.2) and (Eq. 3.27 -3.28), the equations of the state estimation error and the output estimation error, $\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}}$, become:

$$\epsilon = \mathbf{x} - \hat{\mathbf{x}} \tag{3.29}$$

$$\dot{\epsilon} = (\mathbf{A} - \mathbf{MC})\epsilon + \mathbf{Dv} \tag{3.30}$$

$$\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}} = \mathbf{C}\epsilon \tag{3.31}$$

When $\mathbf{e}$ is taken as the residual $\mathbf{r}$, one can see from equations (Eq. 3.30 - 3.31) that $\mathbf{r}$ is a function of ϵ and $\mathbf{v}$. In a similar way, one can find the residuals for reduced-order observers due to the well established state estimation theory. Residual generations based on the state estimator (observer) scheme lay the foundation of FDIA using analytical redundancy in the dynamical system. This scheme has been adopted for this thesis project.

3.4 Summary

In this chapter, we discussed the reduced-order unknown input observer design approach. This approach is computationally simple and attractive. In Theorem A, the conditional existence of the UIO was given. It was shown that the eigenvalues of the estimator can be freely chosen if and only if the total number of outputs is greater than the total number of unknown inputs, and certain observability condition is satisfied. It is also shown (Saif and Guan, 1992) that, if the total number of unknown inputs is equal to that of the total number of outputs, the eigenvalues of the estimator can not be freely chosen; however, an observer with fixed eigenspectrum may exist. Another important issue about the UIO is that a single UIO design is simply not possible if the total number of unknown inputs is greater than the total number of outputs. Residual generations based on the state estimator (observer) scheme are also discussed in this chapter.

It should be pointed out that the review of the UIO theory presented in this chapter was restricted to deterministic, continuous, linear, time-invariant systems. However, recent works (Saif, 1993 a, b) have extended the theory to discrete systems, systems with stochastic noise, as well as a special class of nonlinear systems, namely bilinear systems.

The next chapter shows that failure detection, isolation, identification and accommodation in dynamical systems is still possible using multiple unknown input observers (MUIOs) under the condition that the total number of unknown inputs is greater than the total number of outputs. The UIO design scheme presented in this chapter is still used in our MUIO design scheme. Based on our MUIO design scheme, a method for detecting, isolating, identifying actuator failure in an uncertain dynamical system will be possible. The accommodation or the reconfiguration of dynamical systems will also discussed in the next chapter.

Chapter 4

Robust Estimation and Actuator FDIA

In this chapter, we will develop a scheme for robust estimation, actuator failure detection, identification and isolation and system reconfiguration.

4.1 Model Formulation

Developing the UIO design scheme presented in Chapter 3 relied on the assumptions that the dynamic system in (Eq. 3.1 - 3.2) was known perfectly and that no parameter variations would occur. In practice, however, there are many disturbances affecting plant state and output trajectories such as system uncertainties, plant parameter variations, and sensor and actuator failures. In addition, some of the plant parameters might be unknown or time-varying. In this section, we will consider these effects and build a system's model after we arrive at a model of a practical dynamical system that accounts for all of these effects.

4.1.1 Uncertainty Effects in the System

Consider the following linear, time-invariant, dynamical system described in state space formulation

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{4.1}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} \tag{4.2}$$

where $\mathbf{x} \in \Re^n$ is the state vector, $\mathbf{u} \in \Re^q$ is the input and $\mathbf{y} \in \Re^p$ is the measurable output vector. We assume that matrices $\mathbf{A}$ and/or $\mathbf{B}$ in (Eq. 4.1)contain uncertainty effects; and only the nominal value of matrices $\mathbf{A}$ and $\mathbf{B}$ are known. In the rest of this section, we will reformulate the systems described in (Eq. 4.1 - 4.2) in the form of a known system with unknown inputs. In order to achieve this reformulation, the following definition is given.

DEFINITION: The n by l_a *uncertainty indicator matrix* of any n by k matrix $\mathbf{A}$ is defined as $\mathbf{I}_A(a_1, a_2, ..., a_{l_a})$, where l_a is the number of rows in $\mathbf{A}$ that contain uncertain elements. The jth column of this matrix has zero entries except for the a_jth entry which has a value of one.

We illustrate the above definition using the following example: If $\mathbf{A}$ is a 4 by 4 matrix and there are uncertain elements in the first and the fourth rows ($l_a = 2$), then $a_1 = 1$ and $a_2 = 4$. In addition, it is assumed that

$$\mathbf{A} = \mathbf{A}_o + \Delta\mathbf{A} \tag{4.3}$$

where $\mathbf{A}_o$ is the known nominal value, and

$$\Delta\mathbf{A} = \begin{bmatrix} \Delta\mathbf{A}_1 \\ \Delta\mathbf{A}_2 \\ \vdots \\ \Delta\mathbf{A}_n \end{bmatrix} \tag{4.4}$$

is the uncertainty matrix associated with $\mathbf{A}$.

Assuming that only l_a number of rows of $\mathbf{A}$ have uncertain elements associated with them ($l_a < n$), (Eq. 4.4) can then be rewritten as

$$\Delta\mathbf{A} = \mathbf{I}_A(a_1, a_2, ..., a_{l_a}) \begin{bmatrix} \Delta\mathbf{A}_{a_1} \\ \Delta\mathbf{A}_{a_2} \\ \vdots \\ \Delta\mathbf{A}_{a_{l_a}} \end{bmatrix} \tag{4.5}$$

where $\Delta\mathbf{A}_{a_i}$, $i = 1, 2, ..., l_a$, is the a_ith row of the matrix $\Delta\mathbf{A}$. Therefore, we denote the $\Delta\mathbf{A}$ as

$$\Delta\mathbf{A} = \mathbf{I}_A \Delta\mathbf{A}_a \tag{4.6}$$

Similar definitions will apply to the matrix $\mathbf{B}$ in (Eq. 4.1), and the uncertain system can be written as

$$\dot{\mathbf{x}} = \mathbf{A}_O\mathbf{x} + \mathbf{B}_O\mathbf{u} + \mathbf{I}_A\Delta\mathbf{A}_a\mathbf{x} + \mathbf{I}_B\Delta\mathbf{B}_b\mathbf{u} \tag{4.7}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} \tag{4.8}$$

where $\mathbf{I}_B(b_1, b_2, ..., b_{l_b})$ has the similar definition and l_b is the number of rows in $\mathbf{B}$ that contain uncertain elements.

Therefore, we have the equivalent

$$\dot{\mathbf{x}} = \mathbf{A}_O\mathbf{x} + \mathbf{B}_O\mathbf{u} + \mathbf{D}\mathbf{v} \tag{4.9}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} \tag{4.10}$$

where

$$\mathbf{D} = [\mathbf{I}_A \quad \mathbf{I}_B] \tag{4.11}$$

$$\mathbf{v} = \begin{bmatrix} \Delta\mathbf{A}_a\mathbf{x} \\ \Delta\mathbf{B}_b\mathbf{u} \end{bmatrix} \tag{4.12}$$

We can see that an uncertain dynamic system described in (Eq. 4.1 - 4.2) can be transformed into (Eq. 4.9 - 4.12) — the form of a known system with uncertainty effects as its unknown inputs.

4.1.2 Actuator Failure Models

Let $\mathbf{u}$ represent the output of a healthy actuator. Let $\bar{\mathbf{u}}$ be the actual output of the actuator where the possibility of a failure has been taken into account (See Fig. 4.1). Then we have

$$\bar{\mathbf{u}} = \mathbf{u} + \mathbf{v} \tag{4.13}$$

where $\mathbf{v}$ is a time-varying vector with elements v_i. By the appropriate choice of v_i we can capture various failure modes of the ith actuator. For example, if the ith actuator freezes at its zero position providing no output at all, then $v_i = -u_i$; if there is a bias h_i appearing on the actuator for some reasons, then $v_i = h_i$; if the ith actuator is stuck at a constant value k_i, then $v_i = k_i - u_i$. Multiple failures can be captured in the above setting by specifying elements of v_i corresponding to the unfailed actuator

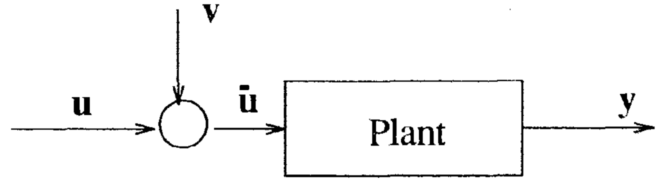

Figure 4.1: System with actuator failures (Sensors are all good)

to be zero. The model presented in (Eq. 4.9 - 4.12) does not take into account the actuator failure effects. Now, taking into account the effects of actuator failures, we will arrive at the following model

$$\dot{\mathbf{x}} = \mathbf{A}_O\mathbf{x} + \mathbf{B}_O\mathbf{u} + \mathbf{D}^*\mathbf{v}^* \tag{4.14}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} \tag{4.15}$$

where

$$\mathbf{D}^* = [\mathbf{D} \quad \mathbf{I}_A \quad \mathbf{I}_B] \tag{4.16}$$

$$\mathbf{v}^* = \begin{bmatrix} \mathbf{v} \\ \Delta\mathbf{A}_a\mathbf{x} \\ \Delta\mathbf{B}_b\mathbf{u} \end{bmatrix} \tag{4.17}$$

We will, in the rest of this thesis, adopt this system represented in equations (Eq. 4.14 - 4.17) — the form of a known system with uncertainty effects and actuator failure effects as its unknown inputs — to deal with the actuator FDIA problem using the MUIO scheme.

4.2 Necessary Condition for UIO based FDIA

Failure detection, isolation and identification in dynamic control systems using the state estimation concept and a single UIO scheme is conditional to the relationship between the total number of unknown inputs and the number of output signals. Guan and Saif proposed a scheme for the FDIA in the dynamic system (Guan and Saif, 1991). In their scheme, they stated that the FDIA task would be possible by using a single UIO scheme if the total number of uncertain rows in the plant matrices **A** and **B** plus the total number of actuator failures minus the number of common rows in **A** and **B** for which there exists uncertain parameters is less than the total number of output signals. It should be pointed out that the FDIA task would not be possible by using a single UIO scheme if the total number of unknown inputs is greater than the total number of output signals.

In order to deal with the FDIA problem in certain cases where the above condition is not satisfied, here we propose an approach that would somewhat relax the condition described in Guan (1990). We will introduces a design scheme for state estimation which makes it possible to deal with actuator FDIA problem and to implement system reconfiguration for a linear dynamic system subject to plant parameter variations or uncertainties if certain conditions are met. Assume that the total number of uncertain rows in the plant matrices **A** and **B** minus the number of common rows with uncertain parameters is m_1 and the total number of actuators is q. In the remainder of this thesis, we assume that $m_1 + 2$ is less than the number of output signals in the dynamic system, i.e., $m_1 + 2 < p$.

4.3 Unknown Input Sub-system Formulation

Note that the minimum number of UIO's needed to accomplish the FDIA task is dependent on the existence condition of the sub-UIO's. In the rest of this chapter, we have assumed that at most one actuator can be accommodated with each sub-UIO. This is the worst scenario which results in the number of sub-UIO's to be equal to the number of the actuators. It is clear that depending on the problem on hand, the FDIA could be accomplished using fewer UIOs (see Case 1 in the numerical example of Chapter 5).

For FDIA purpose, we will have "q" sub-systems which are called ***unknown input sub-systems***. The unknown input to each sub-system will consist of mode of one of the actuator failures and the uncertainty effects of A and B matrices. For the ith sub-system the output consists of all of the outputs of the system except the ith one. In the rest of this thesis, superscript notation is used for referring to variables corresponding to different sub-systems. Therefore, the ith unknown input sub-system will be

$$\dot{\mathbf{x}}^i = \mathbf{A}_O \mathbf{x}^i + \mathbf{B}_O \mathbf{u} + \mathbf{D}^{*^i} \mathbf{v}^{*^i}, \quad \forall \quad i = 1, 2, ..., q \tag{4.18}$$

$$\mathbf{y}^i = \mathbf{C}^i \mathbf{x}^i \tag{4.19}$$

where

$$\mathbf{D}^{*^i} = [\mathbf{d}_i \quad \mathbf{I}_A \quad \mathbf{I}_B] \tag{4.20}$$

$$\mathbf{v}^{*^i} = \begin{bmatrix} v_i \\ \Delta \mathbf{A}_a \mathbf{x}^i \\ \Delta \mathbf{B}_b \mathbf{u} \end{bmatrix} \in \Re^{m_1+1} \tag{4.21}$$

where $\mathbf{y}^i$ is the output signal of unknown input sub-system i and is obtained by deleting the ith output signal, y_i, from the system's output, $\mathbf{d}_i$ is the ith column of the matrix $\mathbf{D}$ described in (Eq. 4.16) and v_i is the ith actuator failure signal to the system described in (Eq. 4.14). In (Eq. 4.22) the $\mathbf{y}^i$ and the $\mathbf{u}$ are the measurable signals required to obtain $\hat{y}_i^i$, an estimate of y_i, where $\hat{y}_i^i$ is defined as the estimate of the ith element of the system's output vector, i.e., y_i using the ith UIO. Therefore, we can see that the ith UIO is driven by all of the inputs and all of the outputs except the ith output. By the nature of the construction of unknown input observer (see Chapter 3), the value of $\mathbf{v}^{*^i}$ does not affect $\hat{y}_i^i$. Thus, for the above system, assuming that the following conditions are satisfied:

1. Rank($\mathbf{C}^i\mathbf{D}^{*^i}$)=Rank($\mathbf{D}^{*^i}$) $= m_1 + 1$; where $i = 1, 2, ..., q$;
2. Observability condition in THEOREM B in Chapter 3 is satisfied.

We will build q UIOs for which the dynamics of the ith one is given by

$$\dot{\mathbf{w}}^i = \mathbf{F}^i\mathbf{w}^i + \mathbf{E}^i\mathbf{y}^i + \mathbf{L}^i\mathbf{u} \tag{4.22}$$

The estimation of state variables is:

$$\hat{\mathbf{x}}^i = \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} \mathbf{w}^i + \begin{bmatrix} \mathbf{N}^i \\ \mathbf{I} \end{bmatrix} \mathbf{y}^i \tag{4.23}$$

where $\mathbf{F}^i$, $\mathbf{E}^i$, $\mathbf{L}^i$, $\mathbf{N}^i$, can be obtained as described in Chapter 3.

4.4 Estimation Error Dynamics Under Actuator Failures

Consider the dynamic system representations (Eq. 4.18 - 4.19) and the ith UIO dynamics (Eq. 4.22 - 4.23).

We define the error

$$\epsilon^i = \mathbf{x}_1^i - \hat{\mathbf{x}}_1^i, \quad \forall \quad i = 1, 2, ..., q \tag{4.24}$$

where $\mathbf{x}_1^i \in \Re^{n-p+1}$ is the vector in the ith unknown input sub-system that need to be estimated, and $\hat{\mathbf{x}}_1^i$ is its estimate obtained from the ith UIO (given in Eq. 4.23). We can obtain $\hat{y}_i^i$, the estimate of y_i, by using the relation,

$$\hat{y}_i^i = c^i \hat{\mathbf{x}}_1^i, \quad \forall \quad i = 1, 2, ..., q \tag{4.25}$$

Therefore the observation error, or the residual can be obtained:

$$e^i = y_i - \hat{y}_i^i = y_i - c^i \hat{\mathbf{x}}_1^i, \quad \forall \quad i = 1, 2, ..., q \tag{4.26}$$

where c^i is the ith row of matrix $\mathbf{C}$.

It can be shown easily that

$$\dot{\epsilon}^i = (\bar{\mathbf{A}}_{11}^i - \mathbf{M}^i \bar{\mathbf{A}}_{21}^i)\epsilon^i, \quad \forall \quad i = 1, 2, ..., q \tag{4.27}$$

Since $(\bar{\mathbf{A}}_{11}^i - \mathbf{M}^i \bar{\mathbf{A}}_{21}^i)$ is stable, $\epsilon^i \longrightarrow 0$, $\hat{\mathbf{x}}_1^i$ estimates $\mathbf{x}_1^i$ asymptotically, therefore the observation error $e^i = y_i - \hat{y}_i^i = y_i - c^i \hat{\mathbf{x}}_1^i \longrightarrow 0$ as $t \rightarrow \infty$, where $\bar{\mathbf{A}}_{ij}$'s are as defined in (Eq. 3.16).

The error dynamics of the ith UIO has two characteristics:

1. **ith Actuator has failed:**

   In this case, we have proven that $\epsilon^i = \mathbf{x}_1^i - \hat{\mathbf{x}}_1^i \longrightarrow 0$, and observation error $e^i = y_i - \hat{y}_i^i = y_i - c^i\hat{\mathbf{x}}_1^i \longrightarrow 0$. Thus the error equation of the ith UIO is not affected by the failure of the ith actuator.

2. **jth Actuator, $j \neq i$ has failed:**

   In this case, e^i will have a steady state error in its dynamics.

4.5 Actuator FDIA Using MUIOs

In order to conduct actuator FDIA for dynamical system using MUIOs scheme, there are some conditions have to be met in this thesis work:

1. Only one actuator fails at any instant of time;

2. The failures occur only after the estimator's transients have died out;

3. The subsequent failure in a certain actuator occurs only after the transient effects from previous failure has died out.

In order to detect and identify actuator failures in dynamical systems, we will first introduce the following result:

We consider the following linear, time-invariant system

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{D}\mathbf{v} \tag{4.28}$$

For the above system we can prove the following theorem.

THEOREM B: Let the value of $\mathbf{x}$ and $\mathbf{v}$ at time kT by $\mathbf{x}$(k) and $\mathbf{v}$(k), where T is the sampling time of the system. If T is small enough, then for system (Eq. 4.28), given $\mathbf{x}$(k), the input $\mathbf{v}$(k) can be calculated (or retrieved) as follows:

$$\mathbf{v}(k) = (\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T\mathbf{S}(k) \tag{4.29}$$

where

$$\mathbf{S}(k) = \mathbf{A}(e^{\mathbf{A}T} - \mathbf{I})^{-1}(\mathbf{x}(k+1) - e^{\mathbf{A}T}\mathbf{x}(k)) \tag{4.30}$$

The proof of this theorem can be found in (Guan, 1990). Equation (Eq. 4.29) shows that once we obtain system's states $\mathbf{x}$, we can calculate the value of $\mathbf{v}$ using this equation. However, since the entire state $\mathbf{x}$ in (Eq. 4.30) is not available from the measurements, we can use the state estimate $\hat{\mathbf{x}}$ of $\mathbf{x}$ in (Eq. 4.30) and (Eq. 4.29) to get the estimate of $\mathbf{v}$, i.e., $\hat{\mathbf{v}}$. Note also that by simple modification of the above we can account for additional terms such as known inputs in (Eq. 4.28).

4.5.1 On-Line Detection and Accommodation of Actuator Failures

Assume that the dynamical system is in actual operation. If there are no actuator failures, all failure estimates, i.e., $\hat{v}^i$, $i = 1, 2, ..., q$ should be zero and as well all the estimation residuals e^i should also be zero. Without loss of generality, assume that after sometime the first actuator fails. When the first actuator fails, all the MUIOs except MUIO1 will give wrong estimate of state variable $\mathbf{x}$. With the knowledge of $\hat{\mathbf{x}}^i$, we can use (Eq. 4.29) to obtain the estimate of v_i^i, i.e., $\hat{v}_i^i$, i= 1, 2, ..., q, where $\hat{v}_i^i$ is the estimate of ith actuator failure element obtained by using the ith MUIO. At this point we should observer that all $\hat{v}_i^i \neq 0$, which would indicate the presence of a failure. To detect the source of the failure, we would furthermore check the residuals e^i. We would declare the failure of actuator #1 if the following two conditions are met:

1. $\hat{v}_1^1 \neq 0$;

2. $e^1 = 0$.

It should be noted that if actuator #1 has failed, then the error residuals obtained from all MUIOs other than the first would be non-zero.

Here we present a possible approach for accommodating some class of incipient failures. Once we have detected and identified the failure of the first actuator we can account for it by compensating for this failure from the corresponding input signal and keep the system functioning smoothly. In this way, the dynamic system we are considering with actuator failure v_1 can be offset to a non-faulty system. In other words, the first actuator failure has been accommodated. Once the first actuator failure has been detected and identified, the failure becomes a *known input* to the system. When other actuators fail, the FDIA task will be undertaken in the similar way as the first one and will be dealt with similarly.

In practical applications, the residuals could be nonzero due to the presence of disturbances and unmodelled dynamics. Hence a threshold (δ) is fixed by conducting simulation studies and failure is signalled if the residual cross the thresholds.

In summary, using the MUIO design scheme and applying the actuator FDIA method discussed in this chapter, multiple actuator failures can be detected, isolated

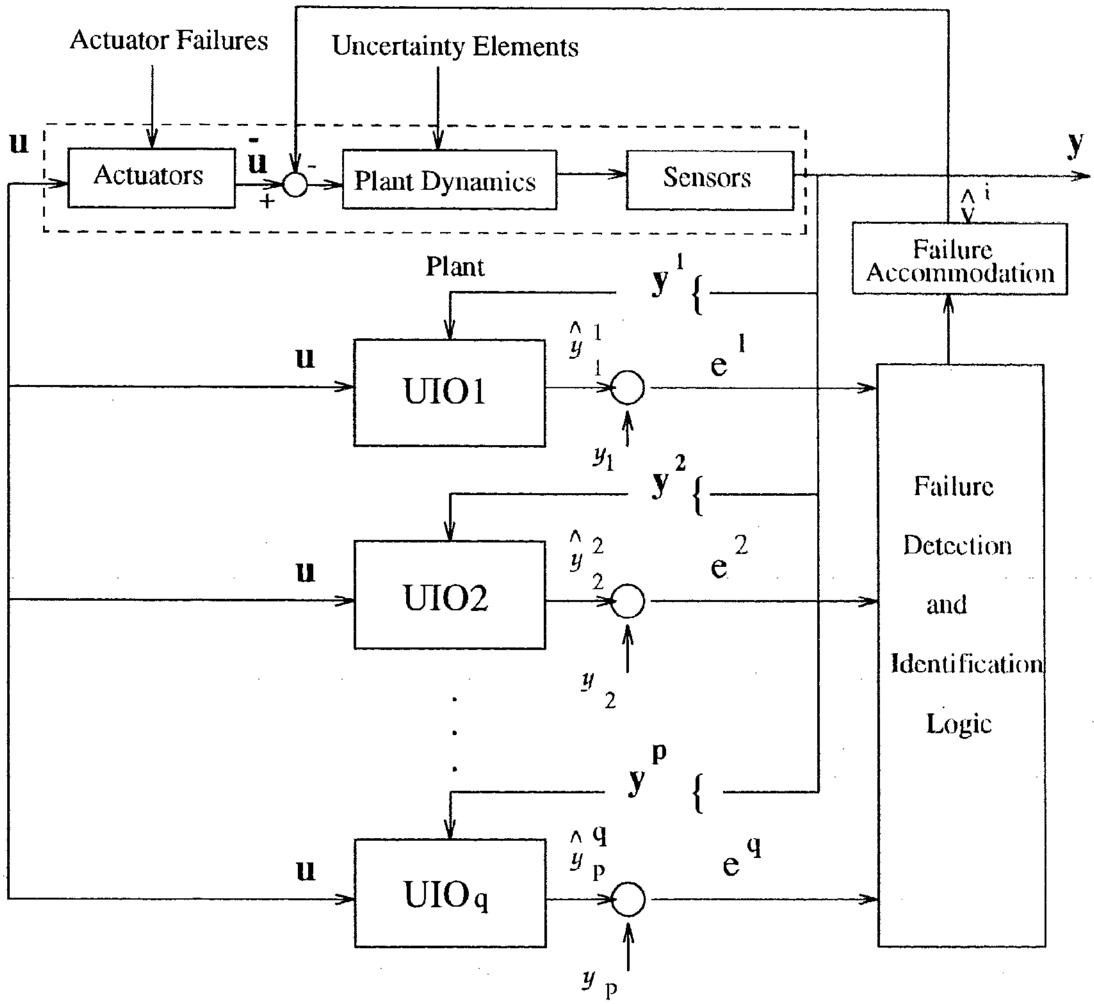

Figure 4.2: Multiple Actuator FDIA Scheme Using MUIOs

and identified in certain cases where the total number of unknown inputs is greater than the total number of outputs in dynamical systems. The accommodation of actuator failures would be possible by using the techniques described above.

Fig. 4.2 summarizes the architecture of the multiple actuator FDIA scheme in a dynamical system subject to system uncertainties.

Chapter 5

Example and Simulation Results

To show the applicability of the proposed actuator FDIA scheme using MUIOs, in this chapter, we will consider designing a bank of MUIOs for a linear, time-invariant system based on the theoretical results of the previous chapters. The system under consideration is a fourth-order dynamical model describing the longitudinal dynamics of the F18 High Alpha Research Vehicle (F18/HARV) (Voulgaris and Valavani, 1991). We will use the scheme we have developed to detect , isolate and identify actuator failures when the total number of unknown input is greater than the total number of output. To demonstrate the actuator FDIA scheme by using MUIOs, we will consider the following two cases:

1. Assume that all actuators fail at the different time and the plant dynamics are known perfectly. This is to demonstrate that multiple actuator failures can be detected, isolated and identified;

2. All actuators fail, and there are uncertain elements in the fourth row of the plant matrix **A**. In practice, however, the states of the system are not always available for constructing feedback control law. So although the states are available for use, the estimates of the system's states will be used to build feedback control law, i.e., $\mathbf{u} = -\mathbf{k}\hat{\mathbf{x}} + \mathbf{u_r}$, where $\mathbf{u}$ is the control input in the system, $\mathbf{k}$ is the feedback gain, $\hat{\mathbf{x}}$ is the estimates of the system's states and $\mathbf{u_r}$ is the reference input to the system. This is to demonstrate that multiple actuator failures can be detected, isolated and identified in an uncertain system while the estimates of the system are used in feedback control law.

5.1 Case 1: FDIA Using MUIOs in a Certain System

In this section, we will implement FDIA scheme when all actuators in the system fail at the different time.

The dynamics of the F18/HARV is given as follows:

$$\dot{\mathbf{x}} = \begin{bmatrix} -0.0750 & -24.0500 & 0 & -32.1600 \\ -0.0009 & -0.1959 & 0.9896 & 0 \\ -0.0002 & -0.1454 & -0.1677 & 0 \\ 0 & 0 & 1.0000 & 0 \end{bmatrix} \mathbf{x}$$

$$+\begin{bmatrix} -0.0230 & 0 & -0.0729 & 0.0393 & -0.0411 & 0.1600 \\ -0.0002 & -0.0001 & -0.0004 & -0.0000 & -0.0003 & -0.0003 \\ -0.0067 & -0.0007 & -0.0120 & -0.0006 & -0.0007 & 0.0005 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}\mathbf{u} \quad (5.1)$$

where

$$\mathbf{x} = \begin{bmatrix} \text{perturbation in true airspeed (ft/s)} \\ \text{perturbation in angle attack (rad)} \\ \text{perturbation in pitch rate (rad/s)} \\ \text{perturbation in pitch angle (rad)} \end{bmatrix}$$

and

$$\mathbf{u} = \begin{bmatrix} \delta_{tvs} \\ \delta_{as} \\ \delta_{ss} \\ \delta_{les} \\ \delta_{tes} \\ \delta_T \end{bmatrix} = \begin{bmatrix} \text{perturbation in symmetric thrust vectoring vane deflectio, (deg)} \\ \text{perturbation in symmetric aileron deflection (deg)} \\ \text{perturbation in symmetric stabilator deflection (deg)} \\ \text{perturbation in symmetric leading edge flap deflection (deg)} \\ \text{perturbation in symmetric trailing edge flap deflection (deg)} \\ \text{perturbation in throttle position (deg)} \end{bmatrix}$$

Let us denote

$$\mathbf{A}_0 = \begin{bmatrix} -0.0750 & -24.0500 & 0 & -32.1600 \\ -0.0009 & -0.1959 & 0.9896 & 0 \\ -0.0002 & -0.1454 & -0.1677 & 0 \\ 0 & 0 & 1.0000 & 0 \end{bmatrix} \quad (5.2)$$

and

$$\mathbf{B}_0 = \begin{bmatrix} -0.0230 & 0 & -0.0729 & 0.0393 & -0.0411 & 0.1600 \\ -0.0002 & -0.0001 & -0.0004 & -0.0000 & -0.0003 & -0.0003 \\ -0.0067 & -0.0007 & -0.0120 & -0.0006 & -0.0007 & 0.0005 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (5.3)$$

The parameters (i.e. $\mathbf{A}_0$ and $\mathbf{B}_0$) in the above dynamic system are nominal values.

It is assumed that the output equation is given by:

$$\mathbf{y} = \mathbf{C}\mathbf{x} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \mathbf{x} \quad (5.4)$$

In order to perform FDIA task for actuators in the system, here we transform the above system into the following system by using a linear transformation$\tilde{\mathbf{x}} = \mathbf{P}\mathbf{x}$. The $\mathbf{P}$ matrix is given by:

$$\mathbf{P} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad (5.5)$$

Therefore, the transformed system becomes:

$$\dot{\tilde{\mathbf{x}}} = \begin{bmatrix} 0 & 0 & 1.0000 & 0 \\ 0 & -0.1959 & 0.9896 & -0.0009 \\ 0 & -0.1454 & -0.1677 & -0.0002 \\ -32.1600 & -24.0500 & 0 & -0.0750 \end{bmatrix} \tilde{\mathbf{x}}$$

$$+\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -0.0002 & -0.0001 & -0.0004 & -0.0000 & -0.0003 & -0.0003 \\ -0.0067 & -0.0007 & -0.0120 & -0.0006 & -0.0007 & 0.0005 \\ -0.0230 & 0 & -0.0729 & 0.0393 & -0.0411 & 0.1600 \end{bmatrix}\mathbf{u} \quad (5.6)$$

The output equation becomes:

$$\mathbf{y} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\tilde{\mathbf{x}} \quad (5.7)$$

Denote

$$\tilde{\mathbf{A}}_0 = \begin{bmatrix} 0 & 0 & 1.0000 & 0 \\ 0 & -0.1959 & 0.9896 & -0.0009 \\ 0 & -0.1454 & -0.1677 & -0.0002 \\ -32.1600 & -24.0500 & 0 & -0.0750 \end{bmatrix} \quad (5.8)$$

$$\tilde{\mathbf{B}}_0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -0.0002 & -0.0001 & -0.0004 & -0.0000 & -0.0003 & -0.0003 \\ -0.0067 & -0.0007 & -0.0120 & -0.0006 & -0.0007 & 0.0005 \\ -0.0230 & 0 & -0.0729 & 0.0393 & -0.0411 & 0.1600 \end{bmatrix} \quad (5.9)$$

and

$$\tilde{\mathbf{C}}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(5.10)

where

$$\tilde{\mathbf{A}}_0 = \mathbf{P}\mathbf{A}_0\mathbf{P}^{-1}$$

$$\tilde{\mathbf{B}}_0 = \mathbf{P}\mathbf{B}_0$$

$$\tilde{\mathbf{C}}_0 = \mathbf{C}\mathbf{P}^{-1}$$

Because the eigenvalues of matrix $\tilde{\mathbf{A}}_0$, i.e., open-loop poles of the system are located at [-0.2433±j0,3619, 0.0240±j0.1222], the system is unstable and the feedback control is needed to stablize this system. We use state feedback to stablize the system. We arbitrarily put closed-loop poles of the system at [-2, -3, -4, -5] by using the following feedback control law:

$$\mathbf{u} = -\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -747.5279 & 664.6876 & -395.3252 & 1.4865 \\ 0 & 0 & 0 & 0 \\ 1223.9055 & -13567.3417 & -2061.8171 & -14.2739 \\ -227.2017 & -3332.5751 & -709.7493 & 15.2919 \end{bmatrix} \tilde{\mathbf{x}} + \mathbf{u_r}$$

where $\mathbf{u_r}$ is the reference input of the system.

Therefore, the closed-loop system is then modeled as:

$$\dot{\tilde{\mathbf{x}}} = \begin{bmatrix} 0 & 0 & 1.0000 & 0 \\ 0 & -5.0000 & 0 & 0 \\ -8.0000 & 0 & -6.0000 & 0 \\ 0 & 0 & 0 & -3.0000 \end{bmatrix} \tilde{\mathbf{x}}$$

$$+\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -0.0002 & -0.0001 & -0.0004 & -0.0000 & -0.0003 & -0.0003 \\ -0.0067 & -0.0007 & -0.0120 & -0.0006 & -0.0007 & 0.0005 \\ -0.0230 & 0 & -0.0729 & 0.0393 & -0.0411 & 0.1600 \end{bmatrix} \mathbf{u_r} \quad (5.11)$$

and the output equation is given by (Eq. 5.7).

It is assumed that the reference input (unit step with amplitude of 20) is only connected to the first actuator and no other reference inputs will be connected to the system. The followings describe the time and the failure parameters of the actuators:

v_1 = 5sin(t) deg, for $t \geq 20$ second

v_2 = -10 deg, for $t \geq 40$ second

v_3 = 6 deg, for $t \geq 60$ second

v_4 = -8cos(t) deg, for $t \geq 80$ second

v_5 = 10 deg, for $t \geq 100$ second

v_6 = 6 deg, for $t \geq 120$ second

By using (Eq. 4.14 - 4.15), the system can be modeled as:

$$\dot{\tilde{\mathbf{x}}} = \begin{bmatrix} 0 & 0 & 1.0000 & 0 \\ 0 & -5.0000 & 0 & 0 \\ -8.0000 & 0 & -6.0000 & 0 \\ 0 & 0 & 0 & -3.0000 \end{bmatrix} \tilde{\mathbf{x}}$$

$$+\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -0.0002 & -0.0001 & -0.0004 & -0.0000 & -0.0003 & -0.0003 \\ -0.0067 & -0.0007 & -0.0120 & -0.0006 & -0.0007 & 0.0005 \\ -0.0230 & 0 & -0.0729 & 0.0393 & -0.0411 & 0.1600 \end{bmatrix} \mathbf{u_r}$$

$$+\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -0.0002 & -0.0001 & -0.0004 & -0.0000 & -0.0003 & -0.0003 \\ -0.0067 & -0.0007 & -0.0120 & -0.0006 & -0.0007 & 0.0005 \\ -0.0230 & 0 & -0.0729 & 0.0393 & -0.0411 & 0.1600 \end{bmatrix} \mathbf{v} \quad (5.12)$$

Again, the output equation is given by (Eq. 5.7). Recall the necessary condition for UIO based FDIA described in Section 4.2. With the above formulation of the system's model, we know that the number of output signals in unknown input sub-system is 3 and the number of actuator failures allowed in unknown input sub-system is therefore 2. Since we have 6 actuators, the number of MUIOs we need is therefore 3. We then construct three unknown input sub-systems by using the equations of (Eq. 4.18 - 4.19) and build three MUIOs based on unknown input sub-systems we just obtained. The first unknown input sub-system is formulated as follows:

$$\begin{aligned} \dot{\tilde{\mathbf{x}}}^1 = & \begin{bmatrix} 0 & 0 & 1.0000 & 0 \\ 0 & -5.0000 & 0 & 0 \\ -8.0000 & 0 & -6.0000 & 0 \\ 0 & 0 & 0 & -3.0000 \end{bmatrix} \tilde{\mathbf{x}}^1 \\ & + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -0.0002 & -0.0001 & -0.0004 & -0.0000 & -0.0003 & -0.0003 \\ -0.0067 & -0.0007 & -0.0120 & -0.0006 & -0.0007 & 0.0005 \\ -0.0230 & 0 & -0.0729 & 0.0393 & -0.0411 & 0.1600 \end{bmatrix} \mathbf{u_r} \\ & + \begin{bmatrix} 0 & 0 \\ -0.0002 & -0.0001 \\ -0.0067 & -0.0007 \\ -0.0230 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \end{aligned} \quad (5.13)$$

$$\mathbf{y}^1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tilde{\mathbf{x}}^1 \tag{5.14}$$

where $\mathbf{y}^1$ is the output signal of the first unknown input sub-system and is obtained by deleting the first output signal, y_1, from the system's output signal $\mathbf{y}$. It can be seen obviously that the output equation (Eq. 5.14) is in the form of $[\mathbf{0} \quad \mathbf{I}]$, no similarity transformation is needed. Alos note the number of outputs of sub-system #1 is three and the system we are considering is of fourth order, the minimal order observer required to estimate the unmeasurable state variable of the original system is of dimension one. We will place the pole of the observer at -6. By applying the design algorithm of a single UIO presented in Chapter 3, the first MUIO is obtained as:

$$\begin{aligned} \dot{\mathbf{w}}^1 &= -6\mathbf{w}^1 + [-5.2500 \quad 1.0000 \quad -0.5185]\mathbf{y}^1 \\ &+ [0 \quad 0 \quad 0.0057 \quad -0.0072 \quad 0.0082 \quad -0.0257]\mathbf{u_r} \end{aligned} \tag{5.15}$$

The estimate of the state variables of the transformed system is:

$$\hat{\tilde{\mathbf{x}}}^1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \mathbf{w}^1 + \begin{bmatrix} 5.2500 & -0.7500 & 0.1728 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{y}^1 \tag{5.16}$$

Therefore, the estimate of the state variables of the original system is given by:

$$\hat{\mathbf{x}}^1 = \mathbf{P}^{-1}\hat{\tilde{\mathbf{x}}}^1$$

The estimate of y_1, $\hat{y}_1^1$, can be obtained using the relation,

$$\hat{y}_1^1 = c^1\hat{\mathbf{x}}^1 = [0 \quad 0 \quad 0 \quad 1]\hat{\mathbf{x}}^1 \tag{5.17}$$

where c^1 is the first row of matrix $\mathbf{C}$ in (Eq. 5.4), and the residual e^1 is obtained by $e^1 = y_1 - \hat{y}_1^1$.

The remaining two MUIOs design results are given as

$\mathbf{F}_2 = \mathbf{F}_3 = -6$,

$\mathbf{E}_2 = [-24.5868 \quad 1.0000 \quad 0.0344], \mathbf{N}_2 = [24.5868 \quad -0.7500 \quad -0.0115]$

$\mathbf{L}_2 = [-0.0004 \quad 0.0019 \quad 0 \quad 0.0000 \quad 0.0064 \quad 0.0096]$

$\mathbf{E}_3 = [-1.1369 \quad 1.0000 \quad -0.0134], \mathbf{N}_3 = [1.1369 \quad -0.7500 \quad 0.0045]$

$\mathbf{L}_3 = [-0.0047 \quad -0.0004 \quad -0.0082 \quad -0.0006 \quad 0.0000 \quad -0.0000]$

The residauls e^j, j=2, 3 are obtained as:

$$e^2 = y_1 - \hat{y}_1^2, \hat{y}_1^2 = c^1\hat{\mathbf{x}}^2$$

$$e^3 = y_1 - \hat{y}_1^3, \hat{y}_1^3 = c^1\hat{\mathbf{x}}^3$$

Note, the c^1 is used in all residual generation relations. It should be mentioned here that, in this particular case, the residual generation for the purpose of faulty actuator identification is not needed. The actuator failure FDI is done in the following way: By using **THEOREM B** in Chapter 4 and (Eq. 4.21), it is possible to obtain the the estimates of actuator failures from MUIOs. The failure of the first actuator can be identified by observing that $\hat{v}_1$ is non-zero and $\hat{v}_2$ is zero (Note that $\hat{v}_1$ and $\hat{v}_2$ are obtained from MUIO1) and $\hat{v}_3$, $\hat{v}_4$, $\hat{v}_5$ and $\hat{v}_6$ are all non-zero ($\hat{v}_3$ and $\hat{v}_4$ are obtained from MUIO2 and $\hat{v}_5$ and $\hat{v}_6$ are obtained from MUIO3). Since practically at any instant of time only one actuator can fail, this indicates that the first actuator has failed and its shape can be identified easily. Once we have detected and identified the failure of the first actuator we can account for it by compensating for this failure from the corresponding input signal and keep the system functioning smoothly. When other actuators fail, the FDIA task will be undertaken in the similar way as the first one and will be dealt with similarly.

Simulation studies are conducted by using MATLAB Software Toolbox (Control Systems Toolbox) on SUN Sparc Station and results are shown in Fig. 5.1 - 5.4. Fig. 5.1 (a and b) shows actuator failure detection, i.e., the estimate of actuator failure obtained from MUIO1. Fig. 5.1 (c) shows the residual obtained from MUIO1. Fig. 5.1 (d) shows the compensation for the first actuator failure. From Fig. 5.1 we can declare that the first actuator has failed this failure is a sinusoidal signal with an amplitude of 5 degree and the second actuator failed at 40 second with constant amplitude of -10 degree. Fig. 5.2 (a and b) shows actuator failure detection, i.e., the estimate of actuator failure obtained from MUIO2. Fig. 5.2 (c) shows the residual generation obtasined from MUIO2. From Fig. 5.2 we can declare that the third actuator failed at 60 second with constant amplitude of 6 degree and the fourth actuator failed and this failure is a cosine signal with the amplitude of -8 degree. Fig. 5.3 (a and b) shows actuator failure detection, i.e., the estimate of actuator failure obtained from MUIO3. Fig. 5.3 (c) shows the residual generation obtained from MUIO3. From Fig. 5.3 we can declare that the fifth actuator failed at 100 second with constant amplitude of 10 degree and the sixth actuator failed at 120 second with constant amplitude of 6 degree. Fig. 5.4 shows the actual and estimated state trajectories of the system. We can see from 4 plots in Fig. 5.4 the estimated state trajectories converge to the actual states. Therefore, it can be seen from Fig. 5.1 through Fig. 5.4 that multiple actuator failures can be detected, isolated, identified and accommodated when the total number of unknown inputs is greater than the total number of outputs in dynamic systems.

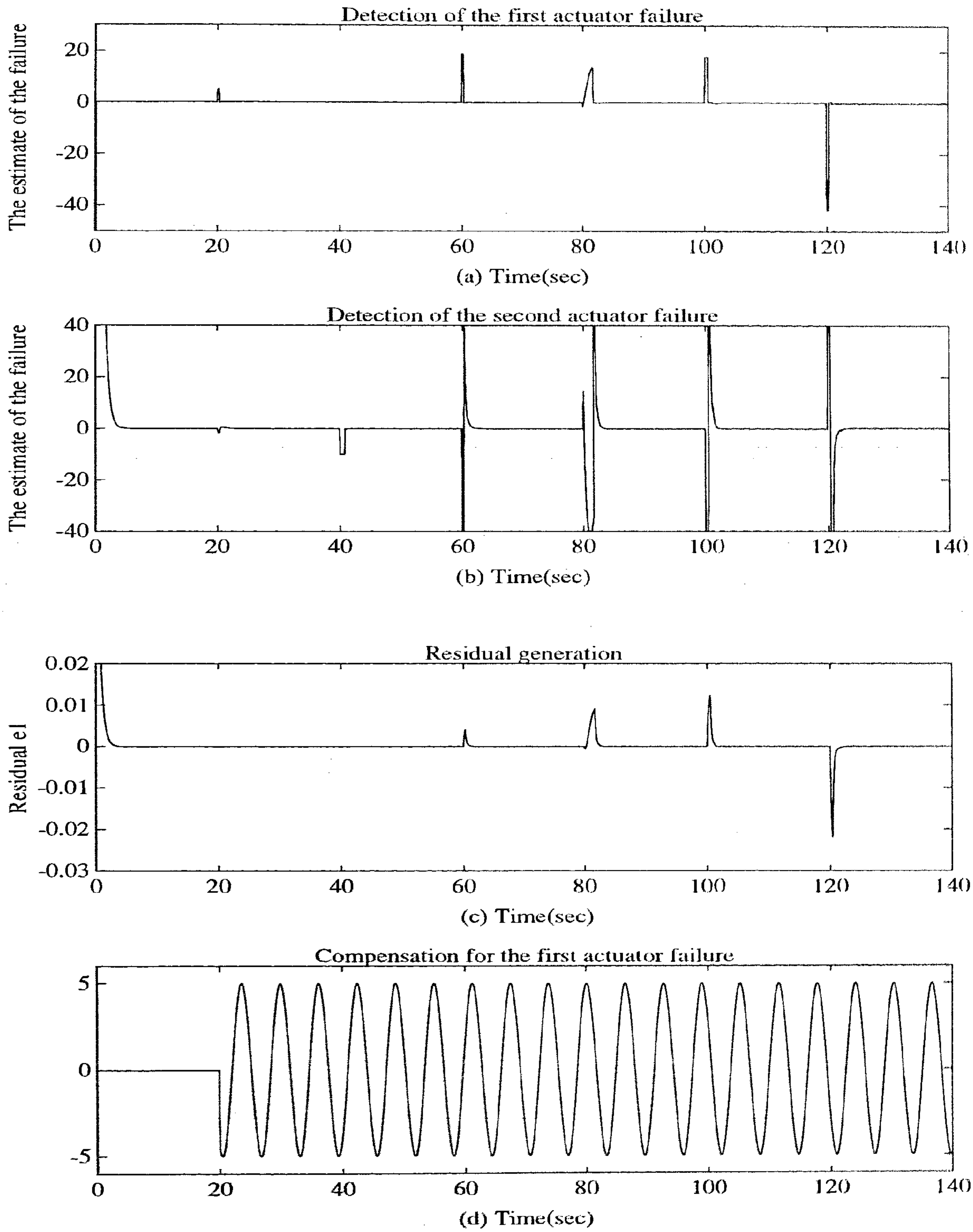

Figure 5.1: Actuator failure detection, isolation and accommodation (FDIA): (a) The estimate of failure – $\hat{v}_1$ – obtained from UIO1; (b) The estimate of failure – $\hat{v}_2$ – obtained from UIO2; (c) Residual Generation: e^1; (d) Compensation for the first actuator failure

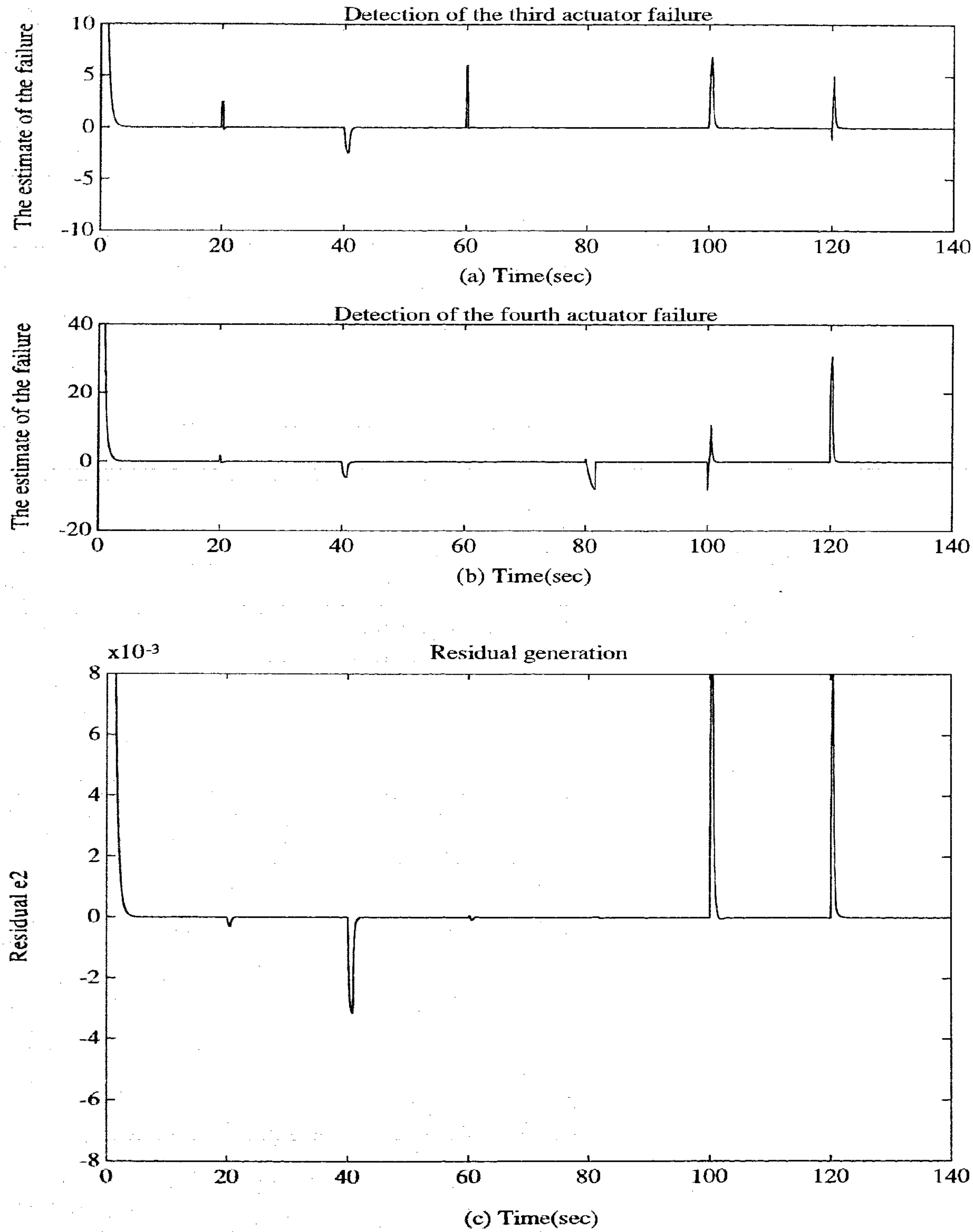

Figure 5.2: Actuator FDIA: (a) The estimate of failure – $\hat{v}_3$ – obtained from UIO3; (b) The estimate of failure – $\hat{v}_4$ – obtained from UIO4; (c) Residual Generation: e^2

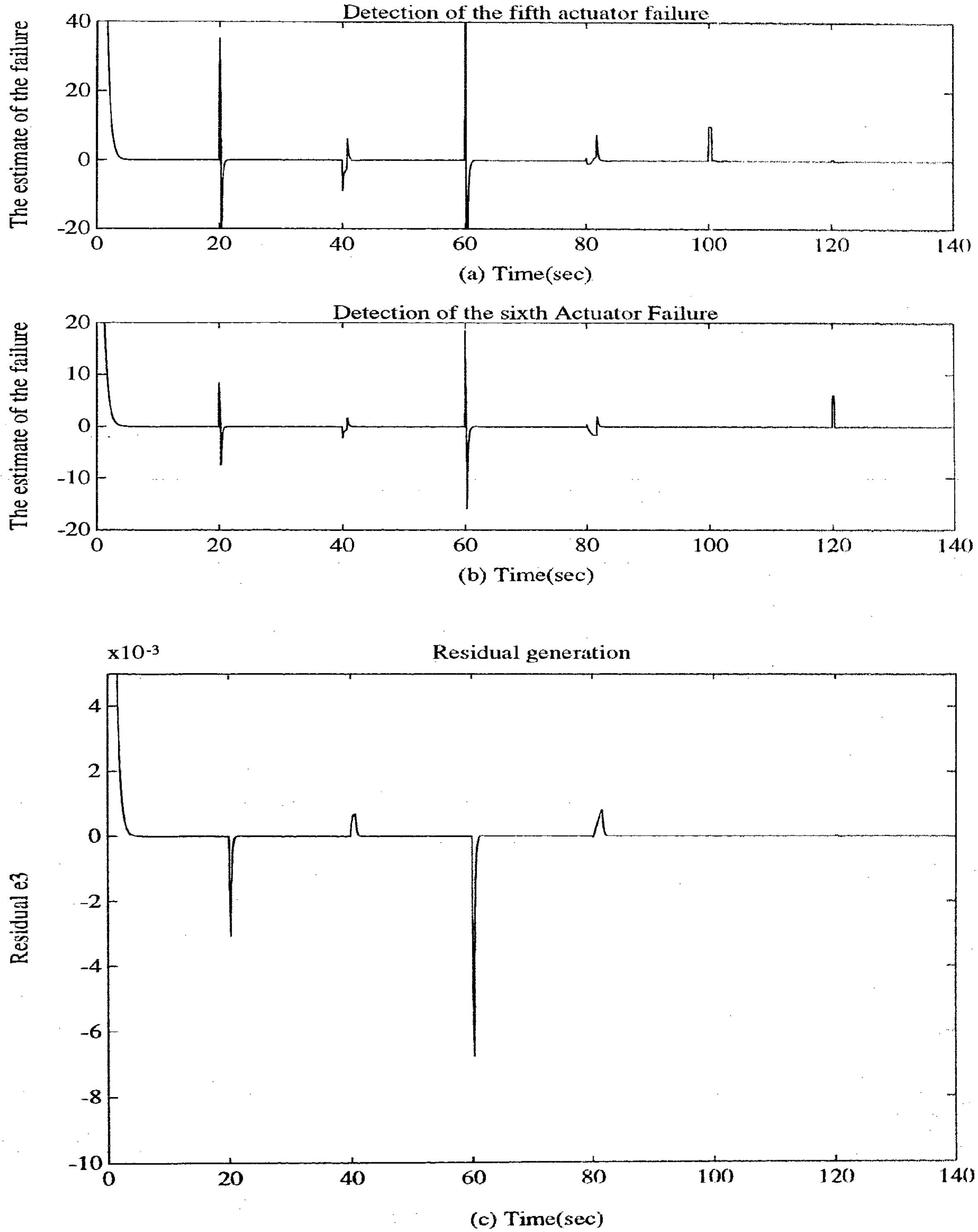

Figure 5.3: Actuator FDIA: (a) The estimate of failure – $\hat{v}_5$ – obtained from UIO5; (b) The estimate of failure – $\hat{v}_6$ – obtained from UIO6; (c) Residual Generation: e^3

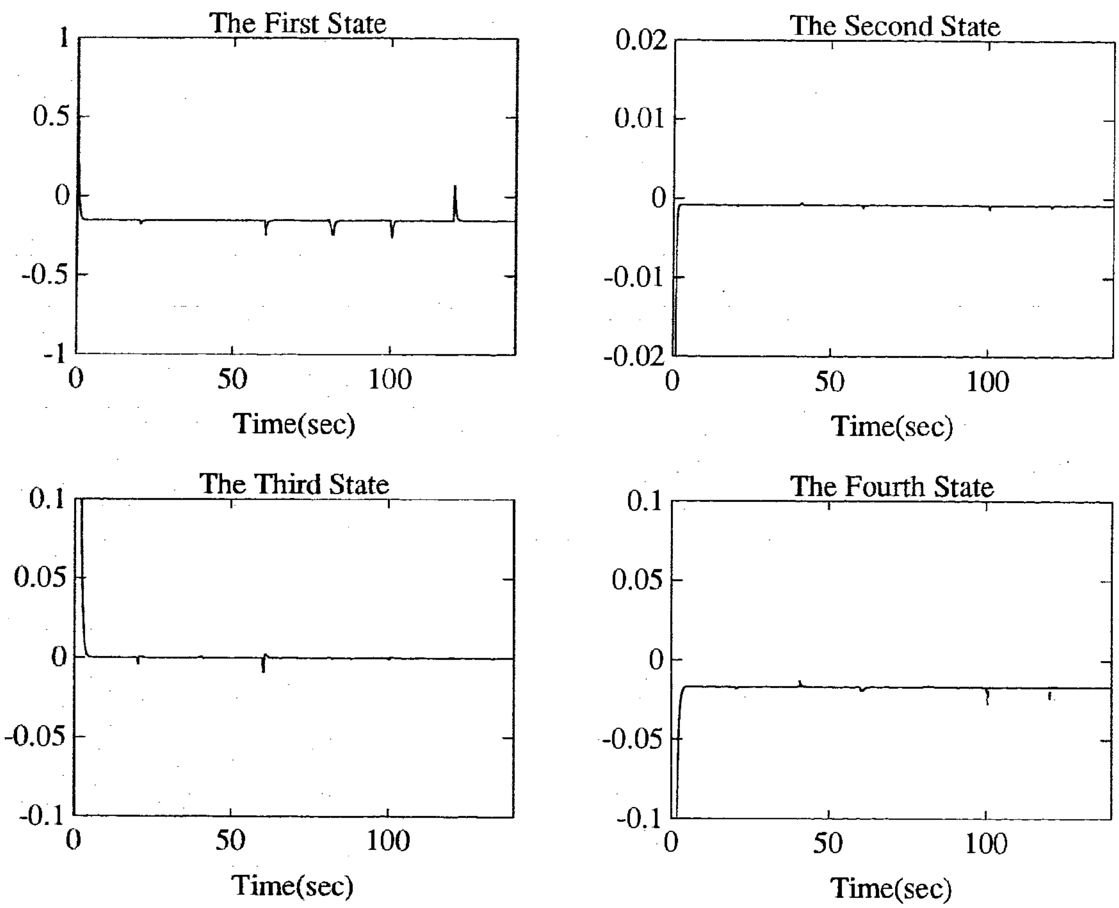

Figure 5.4: Estimation of system's states: actual states (solid lines) and estimated states (dashed lines)

5.2 Case 2: FDIA Using MUIOs in an Uncertain System – Some Practical Considerations

In this section, we will perform the same tasks for the system described in (Eq. 5.1) when parameter variations or uncertainties exist in the system in the form of (Eq. 4.6). We will give some practical considerations about MUIO design and FDIA scheme. Assume that the fourth row of **A** contains parameter variations. The numerical values of these parameter changes are:

$$\Delta \mathbf{a}_{42} = 0.3, \Delta \mathbf{a}_{43} = -0.05, \Delta \mathbf{a}_{44} = 0.8 \tag{5.18}$$

Thus

$$\Delta \mathbf{A}_a = [0 \quad 0.3 \quad -0.05 \quad 0.8] \tag{5.19}$$

$$\mathbf{I}_A = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{5.20}$$

We can see that parameter changes in the fourth row of **A** is relatively large($\mathbf{a}_{42}$ changes from 0 to 0.3, $\mathbf{a}_{43}$ changes from 1 to 0.95 and $\mathbf{a}_{44}$ changes from 0 to 0.8). Also assume that actuator failure scenarios, closed-loop poles of the system in this case are the same as in the Case 1. In practice, however, not all of the state variables of the system are available for establishing feedback control law, it is necessary to use the estimates of the system's states to build feedback control law, i.e., $\mathbf{u} = -\mathbf{k}\hat{\mathbf{x}} + \mathbf{u_r}$. In a MUIO-based actuator FDIA scheme, the estimates of the system's state variables, i.e., $\hat{x}$ must be used to construct the feedback control law. This is to demonstrate that multiple actuator failures can be detected, isolated, identified and accommodated in an uncertain system while the estimates of the system are used in feedback control law.

The dynamic system described in (Eq. 5.1) under the above consideration can then be rewritten as

$$\dot{\mathbf{x}} = \mathbf{A}_O\mathbf{x} + \mathbf{B}_O\mathbf{u} + \mathbf{D}^*\mathbf{v}^* \tag{5.21}$$

where

$$\mathbf{D}^* = [\mathbf{D} \quad \mathbf{I}_A] \tag{5.22}$$

$$\mathbf{v}^* = \begin{bmatrix} \mathbf{v} \\ \Delta\mathbf{A}_a\mathbf{x} \end{bmatrix} \tag{5.23}$$

where $\mathbf{A}_O$, $\mathbf{B}_O$, $\Delta\mathbf{A}_a$ and $\mathbf{I}_A$ are given in (Eq. 5.2 - 5.3) and (Eq. 5.19 - 5.20) and $\mathbf{D} = \mathbf{B}_O$.

It is assumed that the ouput equation is given by:

$$\mathbf{y} = \mathbf{C}\mathbf{x} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x} \tag{5.24}$$

The first unknown input sub-system is given as:

$$\dot{\mathbf{x}}^1 = \mathbf{A}_O\mathbf{x}^1 + \mathbf{B}_O\mathbf{u} + \mathbf{D}^{*^1}\mathbf{v}^{*^1} \tag{5.25}$$

$$\mathbf{y}^1 = \mathbf{C}^1\mathbf{x}^1 \tag{5.26}$$

where

$$\mathbf{D}^{*^1} = [\mathbf{d}_1 \quad \mathbf{I}_A] \tag{5.27}$$

$$\mathbf{v}^{*^1} = \begin{bmatrix} v_1 \\ \Delta\mathbf{A}_a\mathbf{x}^1 \end{bmatrix} \tag{5.28}$$

and $\mathbf{D}^{*^1}$ is the first unknown input matrix and its first column, $\mathbf{d}_1$, is the first column of the matrix $\mathbf{D}$ and the remaining column is the uncertainty indicator matirx of $\mathbf{A}$ ($\mathbf{A} = \mathbf{A}_O + \Delta\mathbf{A}$).

Because part of the system's states is not available for constructing feedback control law, the closed-loop poles of the system will be placed at the same locations as in the Case 1 by using the estimates of system's states:

$$\mathbf{u} = -\mathbf{k}\hat{\mathbf{x}}^1 + \mathbf{u}_r \tag{5.29}$$

The first MUIO is given as

$$\dot{\mathbf{w}}^1 = \mathbf{F}^1\mathbf{w}^1 + \mathbf{E}^1\mathbf{y}^1 + \mathbf{L}^1\mathbf{u} \tag{5.30}$$

where

$$\hat{\mathbf{x}}^1 = \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix}\mathbf{w}^1 + \begin{bmatrix} \mathbf{N}^1 \\ \mathbf{I} \end{bmatrix}\mathbf{y}^1 \tag{5.31}$$

Let

$$\mathbf{R}^1 = \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix}$$

and

$$\mathbf{P}^1 = \begin{bmatrix} \mathbf{N}^1 \\ \mathbf{I} \end{bmatrix}$$

In order to obtain the estimates of the system's state variables, the equations (Eq. 5.25) and (Eq. 5.30) are augmented by adding (Eq. 5.29) and (Eq. 5.31). The result is as follows:

$$\begin{bmatrix} \dot{\mathbf{x}}^1 \\ \dot{\mathbf{w}}^1 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_O + \Delta\mathbf{A} - \mathbf{B}_O\mathbf{k}\mathbf{P}^1\mathbf{C}^1 & \mathbf{B}_O\mathbf{k}\mathbf{R}^1 \\ \mathbf{E}^1\mathbf{C}^1 - \mathbf{L}^1\mathbf{k}\mathbf{P}^1\mathbf{C}^1 & \mathbf{F}^1 - \mathbf{L}^1\mathbf{k}\mathbf{R}^1 \end{bmatrix} \begin{bmatrix} \mathbf{x}^1 \\ \mathbf{w}^1 \end{bmatrix} + \begin{bmatrix} \mathbf{B}_O & \mathbf{d}_1 \\ \mathbf{L}^1 & \underline{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}_r \\ v_1 \end{bmatrix} \tag{5.32}$$

Note that the $\mathbf{C}^1$ matrix in (Eq. 5.26) is in the form of $[\mathbf{0} \quad \mathbf{I}]$. Hence the estimates of the system's state variables are:

$$\hat{\mathbf{x}}^1 = \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} \mathbf{w}^1 + \begin{bmatrix} \mathbf{N}^1 \\ \mathbf{I} \end{bmatrix} \mathbf{y}^1 \tag{5.33}$$

When putting numerical values of the parameters into the above equations, we have

$$\begin{bmatrix} \dot{\mathbf{x}}^1 \\ \dot{\mathbf{w}}^1 \end{bmatrix} = \begin{bmatrix} -0.0750 & 43576.0839 & -1314.2525 & 0 & 3.9250 \\ -0.0009 & -14.9920 & 0.30136 & 0 & -0.0009 \\ -0.0002 & -2.2204 & -4.9330 & -6.0000 & -0.0002 \\ 0 & 0.3000 & 0.9500 & 0.8000 & 0 \\ 0 & -11102.1870 & 334.8414 & 2009.0484 & 4.0000 \end{bmatrix} \begin{bmatrix} \mathbf{x}^1 \\ \mathbf{w}^1 \end{bmatrix}$$

$$+ \begin{bmatrix} -0.0230 & 0 & -0.0729 & 0.0393 & -0.0411 & 0.1600 & -0.0230 \\ -0.0002 & -0.0001 & -0.0004 & 0 & -0.0003 & -0.0003 & -0.0002 \\ -0.0067 & -0.0007 & -0.0120 & -0.0006 & -0.0007 & 0.0005 & -0.0067 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.8758 & -0.4957 & 0.2402 & -3.1374 & -3.3381 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_r \\ v_1 \end{bmatrix}$$

(5.34)

and the estimatese of the system's state variables can be calculated as:

$$\hat{\mathbf{x}}^1 = \begin{bmatrix} 1.0000 \\ 0.0000 \\ 0.0000 \\ 0.0000 \end{bmatrix} \mathbf{w}^1 + \begin{bmatrix} -11102.1870 & 334.8414 & 0 \\ 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & 1.0000 & 0.0000 \\ 0.0000 & 0.0000 & 1.0000 \end{bmatrix} \mathbf{y}^1 \tag{5.35}$$

The residual e_1 is obtained by $e^1 = y_1 - \hat{y}_1^1$ and the estimate of the first actutator failure, $\hat{v}_1$, can also be obtained from Eq. 4.29.

The remaining five MUIO design results are given below and the residauls e^j, j=2, 3, 4, 5, 6, are obtained simialrly.

$\mathbf{F}^2 = \mathbf{F}^3 = \mathbf{F}^4 = \mathbf{F}^5 = \mathbf{F}^6 = -10,$

$\mathbf{E}^2 = [111874.7930 \quad -4726.7406 \quad -32.1600], \mathbf{N}^2 = [-11389.3443 \quad 1627.0492 \quad 0],$

$\mathbf{L}^2 = [8.6004 \quad 0 \quad 14.8960 \quad 1.0155 \quad -2.3190 \quad -4.0703],$

$\mathbf{E}^3 = [108968.3037 \quad 7294.4459 \quad -32.1600], \mathbf{N}^3 = [-11111.4347 \quad 376.4562 \quad 0],$

$\mathbf{L}^3 = [0.2770 \quad -0.8476 \quad 0 \quad 0.2652 \quad -3.1110 \quad -3.3617],$

$\mathbf{E}^4 = [107941.1583 \quad 11542.7004 \quad -32.1600], \mathbf{N}^4 = [-11013.2222 \quad -65.5000 \quad 0],$

$\mathbf{L}^4 = [-2.6645 \quad -1.1472 \quad -5.2642 \quad 0 \quad -3.3909 \quad -3.1112],$

$\mathbf{E}^5 = [120384.5137 \quad -39922.7886 \quad -32.1600], \mathbf{N}^5 = [-12203.0175 \quad 5288.5789 \quad 0],$

$\mathbf{L}^5 = [32.9700 \quad 2.4817 \quad 58.5088 \quad 3.2124 \quad 0 \quad -6.1452],$

$\mathbf{E}^6 = [951810.2167 \quad 643183.9069 \quad 32.1600], \mathbf{N}^6 = [-9793.1373 \quad -5555.8824 \quad 0],$

$\mathbf{L}^6 = [-39.2060 \quad -4.8684 \quad -70.6607 \quad -3.2942 \quad -6.8682 \quad 0].$

Different from the Case 1, the residual generation functions are used for actuator FDIA in this case. The actuator failure FDIA is done in the following way: By using **THEOREM B** in Chapter 4 and (Eq. 4.21), it is possible to obtain the the estimates of actuator failures from MUIOs. When the first actuator has failed, we can observe that $\hat{v}_1$ is non-zero and e^1 is zero (Note that $\hat{v}_1$ and e^1 are obtained from MUIO1). We also observe that $\hat{v}_2$, $\hat{v}_3$, $\hat{v}_4$, $\hat{v}_5$ and $\hat{v}_6$ are all non-zero, but the residuals e^2, e^3, e^4, e^5 and e^6 are all non-zero. This indicates that the first actuator has failed and its shape can be identified easily. Once we have detected and identified the failure of the first actuator we can account for it by compensating for this failure from the corresponding input signal and keep the system functioning smoothly. When other actuators fail, the FDIA task will be undertaken in the similar way as the first one and will be dealt with similarly.

Simulation results are shown in Fig. 5.5 - 5.7. Fig. 5.5 (a) shows actuator failure detection, i.e., the estimate of actuator failure obtained from MUIO1. Fig. 5.5 (b) shows the residual generation obtained from MUIO1. Similarly, Fig. 5.5 (c), Fig. 5.6 (a), Fig. 5.6 (c), Fig. 5.7 (a) and Fig. 5.7 (c) show other actuator failure detection, i.e., the estimates of other actuator failures obtained from MUIO2, MUIO3, MUIO4, MUIO5 and MUIO6, respectively. Fig. 5.5 (d), Fig. 5.6 (b), Fig. 5.6 (d), Fig. 5.7 (b) and Fig. 5.7 (d) show residual generation obtained from MUIO2, MUIO3, MUIO4, MUIO5 and MUIO6, respectively. The declaration of the multiple actuator FDIA results shown in the following three figures are the same as described in Case 1. We conclude that multiple actuator failures can be detected, isolated, identified and accommodated when the total number of unknown inputs is greater than the total number of outputs in dynamical systems with parameter variations.

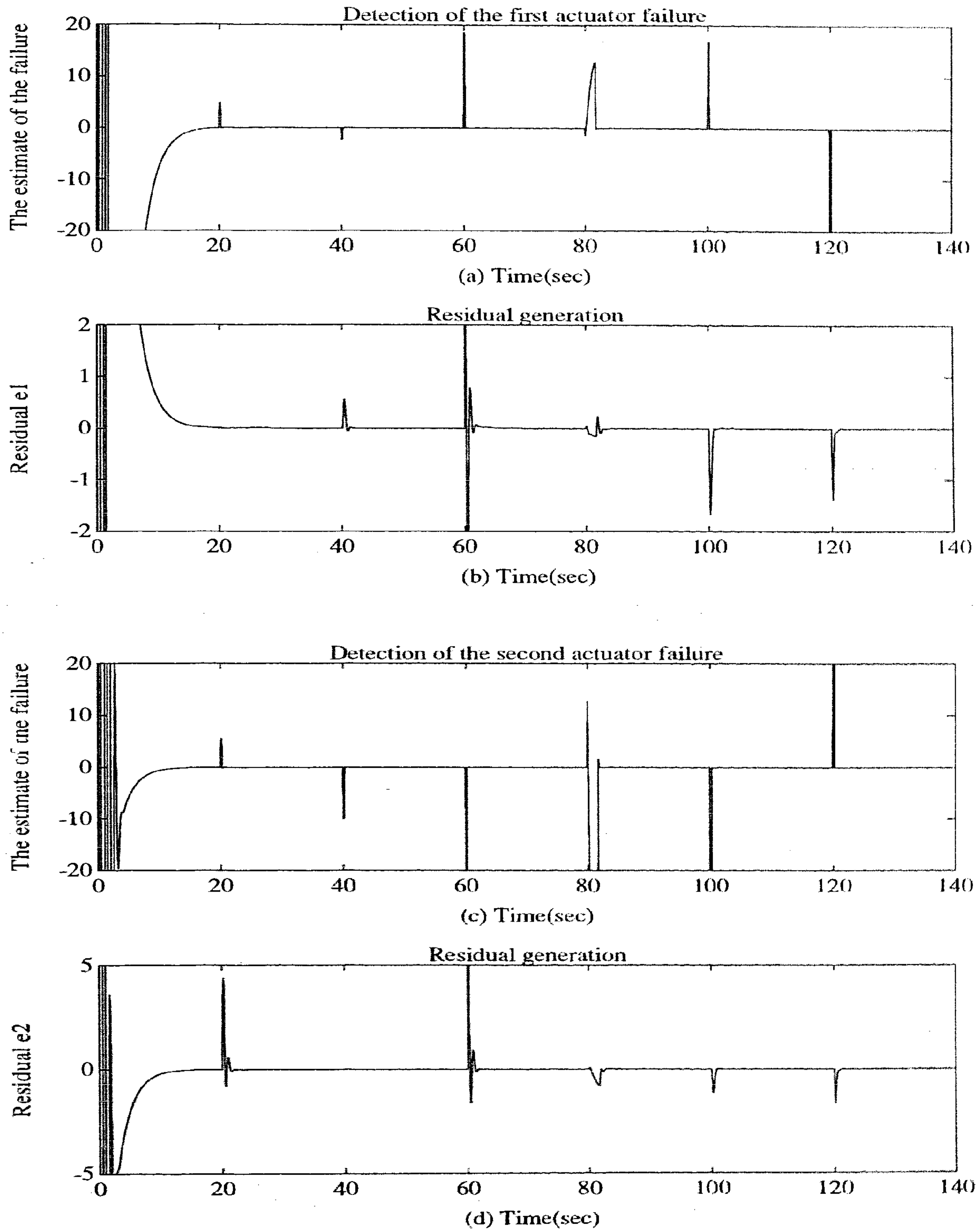

Figure 5.5: Actuator failure detection, isolation and accommodation (FDIA): (a) The estimate of failure – $\hat{v}_1$ – obtained from UIO1; (b) Residual Generation: e^1; (c) The estimate of failure – $\hat{v}_2$ – obtained from UIO2; (d) Residual Generation: e^2

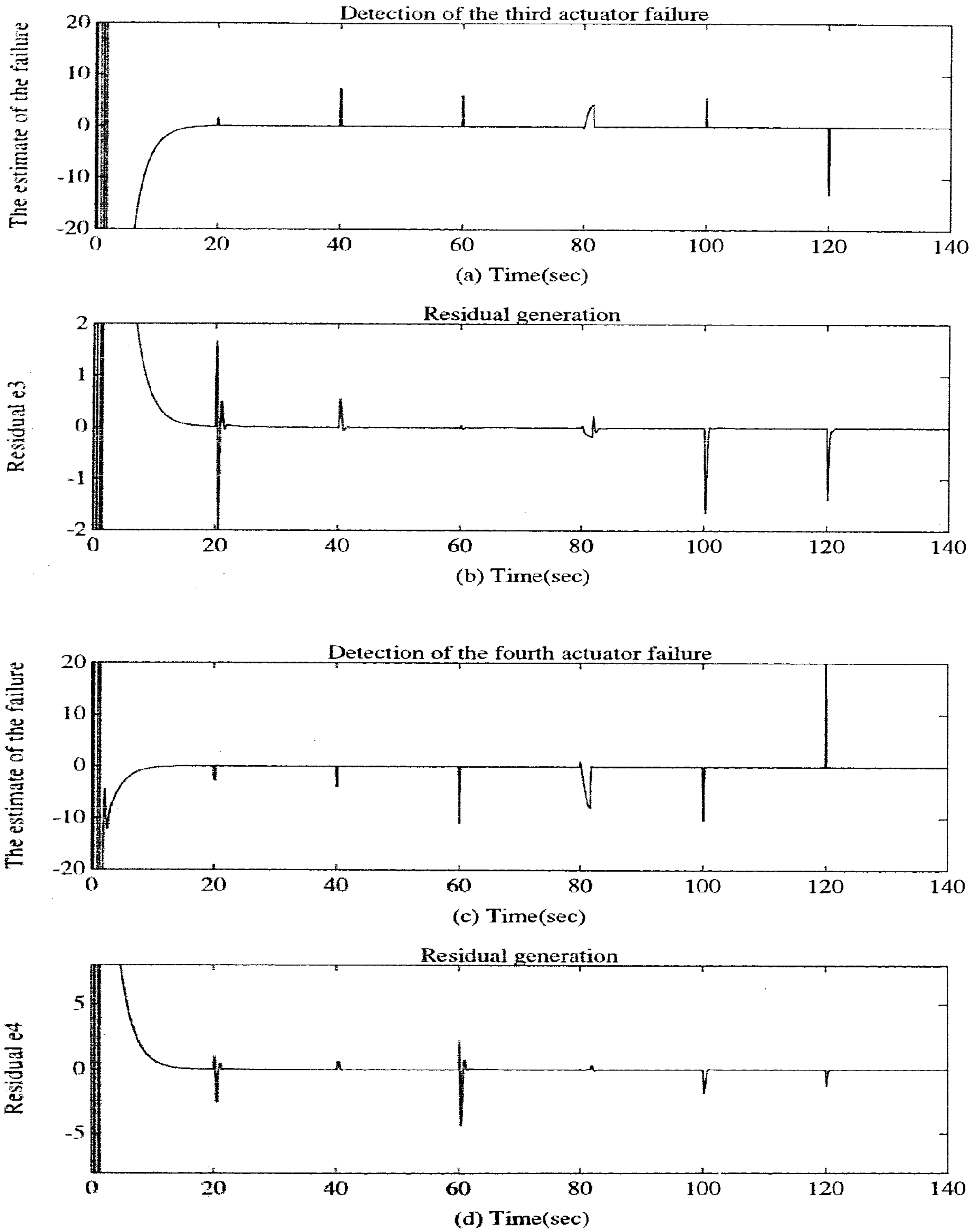

Figure 5.6: Actuator failure detection. isolation and accommodation (FDIA): (a) The estimate of failure – $\hat{v}_3$ – obtained from UIO3; (b) Residual Generation: ϵ^3; (c) The estimate of failure – $\hat{v}_4$ – obtained from UIO4; (d) Residual Generation: ϵ^4

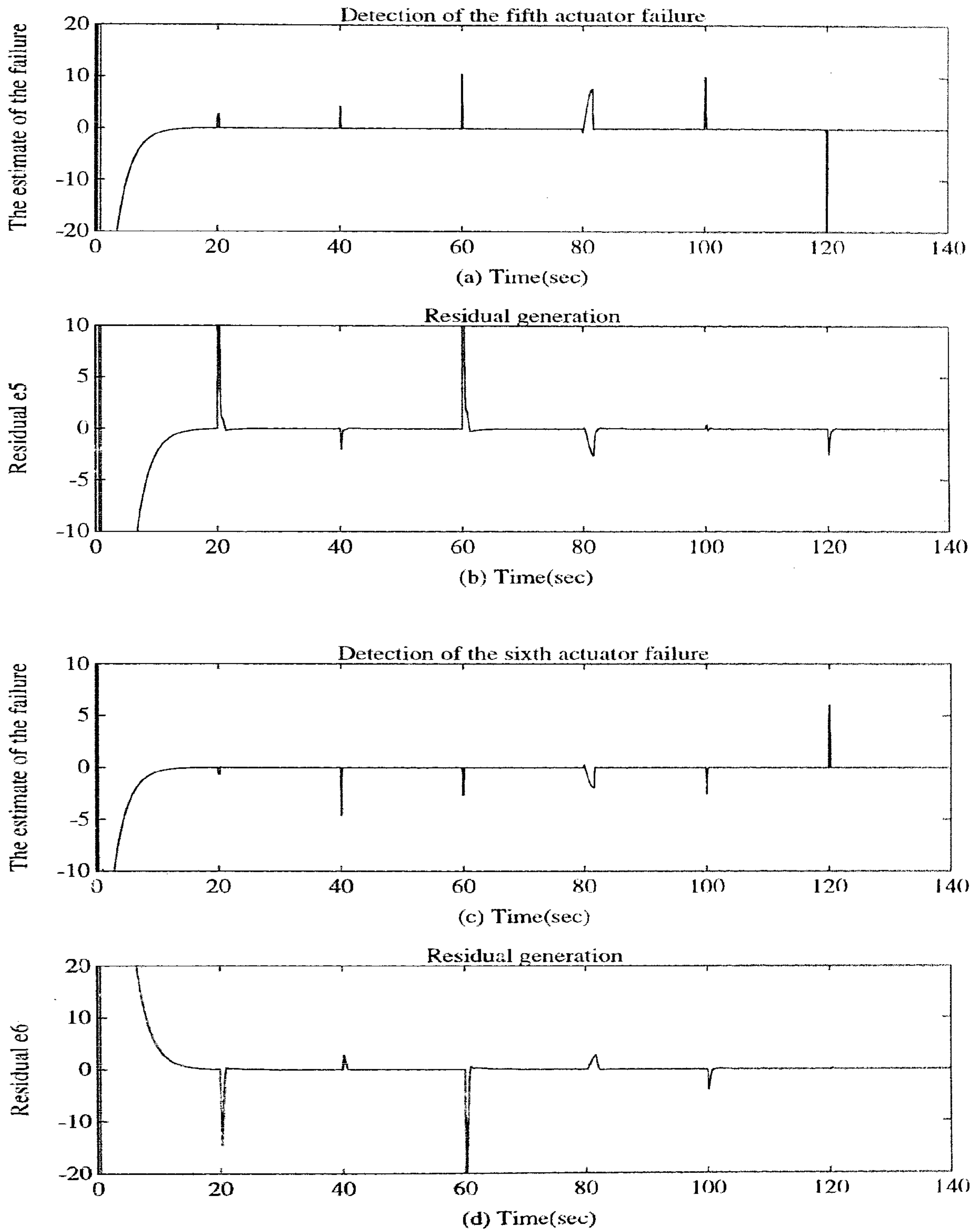

Figure 5.7: Actuator failure detection, isolation and accommodation (FDIA): (a) The estimate of failure – $\hat{v}_5$ – obtained from UIO5; (b) Residual Generation: e^5; (c) The estimate of failure – $\hat{v}_6$ – obtained from UIO6; (d) Residual Generation: e^6

Chapter 6

Conclusions and Future Work

6.1 Conclusions

The major contributions of this thesis project are:

1. We have developed a scheme of designing multiple unknown input observers in order to detect, isolate, identify and accommodate actuator failures when the total number of uncertain rows in the plant matrices **A** and **B** plus the total number of actuator failures is greater than the total number of output signals in dynamic systems. We assume that the total number of uncertain rows in the plant matrices **A** and **B** plus two is less than the number of output signals in the dynamic system. The algorithm of designing MUIOs and FDIA scheme for multiple actuators in a dynamic system are presented in full detail. It is well proven that a single unknown input observer will not exist if the total number of unknown inputs is greater than the total number of output signals. However,

by increasing the number of multiple unknown input observer, the shortcomings of using a single UIO for FDIA purpose under the above conditions will be overcome and multiple actuator failure detection, isolation, identification and accommodation becomes possible.

2. A fault detection, isolation, identification and accommodation scheme based on the MUIO design algorithm is presented in this thesis. The proposed FDIA scheme keeps dynamic systems functioning smoothly. The scheme enables to accommodate the actuator failures by promptly compensating for the actuator failures from the corresponding input signals. At this point we should point out that the simple accommodation approach presented here for soft actuator failure may not be very desirable in practice. Recall that the accommodation is accomplished by simply compensating for the actuator failure once its shape has been correctly estimated. However, it is probable in practical situations that an incipient soft failure may get worsen and the device may actually fail completely after some time. In such situations the accommodation strategy proposed here is not recommended and control reconfiguration should be investigated. This is one topic for future research.

3. In order to fulfill the task of FDIA, an appropriate mathematical model which consider system's uncertainties, parameter variations and actuator failure effects is proposed. The final stage of model building is characterized by transforming an uncertain system into a known system with unknown inputs. This enables us to utilize the model building and FDIA technologies suited for known systems to design feedback controller and estimator for an uncertain system.

4. The MUIO design algorithm is very straightforward and computationally attractive, and system model transformation is simple in essence. The scheme makes it possible to choose the eigenspectrum of the observer at arbitrary locations as long as the complex conjugate eigenvalues appear in pairs. Moreover, unlike the other previous approaches that need calculating a set of matrix equations, the proposed scheme just requires simple matrix calculations (addition or multiplication).

5. The approach can not only detect and isolate multiple actuator failures, actually identify the exact shape of the failure, but most importantly, can accommodate the actuator failures in dynamic systems and maintain the system functioning smoothly.

Simulation tests are conducted for a linear aircraft longitudinal dynamical system to give clear illustrations of the effectiveness of the MUIO design scheme and the FDIA method.

6.2 The Areas of Further Investigations

The main contributions and the advantages of this thesis work have been demonstrated. On the other hand, there are some future work need to be done. Some of these issues are:

1. In this thesis, we only detect, isolate and identify actuator failures by using MUIO design scheme under the condition that the total number of outputs is

less than the total number of the unknown inputs in dynamical system. Sensor failures, however, have not been dealt with and are the subject of future work.

2. In practice, many systems possess nonlinear properties, therefore, future investigations are needed to extend the applications to nonlinear, and bilinear systems.

3. A truly integrated fault-tolerant system would be some hybrid combinations of analytical redundancy and knowledge-based schemes. This is a major topic that need be investigated.

Appendix A

MATLAB Programs

```
% This is the main program which simulates Case 1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This is the main program
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Simulation time set up
t=140;
tt0=t*10;
ttp=(0:0.1:t)';
ttp1=ttp(1:1400,:);
% Reference input
u11=ttp*0+20;
u21=ttp*0-0;
u31=ttp*0+0;
u41=ttp*0+0;
```

```
u51=ttp*0-0;

u61=ttp*0+0;

% The time and the failure parameters of the actuators

vt1=20;

vt2=40;

vt3=60;

vt4=80;

vt5=100;

vt6=120;

tt11=(0:0.1:19.9);

tt12=(20:0.1:t);

V1=[tt11'*0;(5*sin(tt12'))];

v3=[0:0.1:vt2-0.1]'*0+0;

v4=[vt2:0.1:t]'*0-10;

V2=[v3;v4];

v5=[0:0.1:vt3-0.1]'*0+0;

v6=[vt3:0.1:t]'*0+6;

V3=[v5;v6];

tt41=(0:0.1:79.9);

tt42=(80:0.1:t);

V4=[tt41'*0;(-8*cos(tt42'))];

v9=[0:0.1:vt5-0.1]'*0+0;

v10=[vt5:0.1:t]'*0+10;

V5=[v9;v10];

v11=[0:0.1:vt6-0.1]'*0+0;
```

```
v12=[vt6:0.1:t]'*0+6;
V6=[v11;v12];
% Initial condition of the variables in the main loop
x=x0 ;
x_out(:,1)=x0;
y_out(:,1)=c*x0;
w1=w10;
w2=w20;
w3=w30;
w1_out(:,1)=w10;
w2_out(:,1)=w20;
w3_out(:,1)=w30;
xh1_out(:,1)=xh0;
xh2_out(:,1)=xh0;
xh3_out(:,1)=xh0;
% The main loop
for i=1:tt0,
U=[u11(i) u21(i) u31(i) u41(i) u51(i) u61(i)]';
V=[V1(i) V2(i) V3(i) V4(i) V5(i) V6(i)]';
[x,y]=sys(Ph1,Ga1,Gc1,U,V,x);
x_out(:,i+1)=x;
y_out(:,i+1)=y;
y234=y(2:4,:);
% This is the UIO parameter calculation
% Model:  w= F*w + [E L]*|y|
```

```
%                          |u|
%            yw= cw*w + dw *|y|
%                          |u|
[w1,z1,xh1]=esti(Phu1,Gau1,R1,P1,y234,U,T1,w1);
w1_out(:,i+1)=w1;
xh1_out(:,i+1)=xh1;
r1=y_out-c*xh1_out;
r11=r1(1,:);
[w2,z2,xh2]=esti(Phu2,Gau2,R2,P2,y234,U,T1,w2);
w2_out(:,i+1)=w2;
xh2_out(:,i+1)=xh2;
r2=y_out-c*xh2_out;
r21=r2(1,:);
[w3,z3,xh3]=esti(Phu3,Gau3,R3,P3,y234,U,T1,w3);
w3_out(:,i+1)=w3;
xh3_out(:,i+1)=xh3;
r3=y_out-c*xh3_out;
r31=r3(1,:);
% This is failure detection and identification for actuator 1 to actuator 6
Ac1=[a0];
Bc1=[b0 d(:,1) d(:,2)];
[Ph1v,Ga1v]=c2d(Ac1,Bc1,0.1);
dstar1=Ga1v(:,7:8);
xhh1=xh1_out;
s1(:,i)=xhh1(:,i+1)-Ph1v*xhh1(:,i)-Ga1v(:,1:6)*U;
```

```
v1h(:,i)=inv(dstar1'*dstar1)*dstar1'*s1(:,i);
get1=v1h(1,:);
get2=v1h(2,:);
Ac2=[a0];
Bc2=[b0 d(:,3) d(:,4)];
[Ph2v,Ga2v]=c2d(Ac2,Bc2,0.1);
dstar2=Ga2v(:,7:8);
xhh2=xh2_out;
s2(:,i)=xhh2(:,i+1)-Ph2v*xhh2(:,i)-Ga2v(:,1:6)*U;
v2h(:,i)=inv(dstar2'*dstar2)*dstar2'*s2(:,i);
get3=v2h(1,:);
get4=v2h(2,:);
Ac3=[a0];
Bc3=[b0 d(:,5) d(:,6)];
[Ph3v,Ga3v]=c2d(Ac3,Bc3,0.1);
dstar3=Ga3v(:,7:8);
xhh3=xh3_out;
s3(:,i)=xhh3(:,i+1)-Ph3v*xhh3(:,i)-Ga3v(:,1:6)*U;
v3h(:,i)=inv(dstar3'*dstar3)*dstar3'*s3(:,i);
get5=v3h(1,:);
get6=v3h(2,:);
% Failure isolation and accommodation
if  i >= (vt1-0.1)*10 & abs(get1(:,i)) >= 0.1 &
    abs(r11(:,i)) <= 1e-2 & i < vt2*10
vvp11=V1(1:(i),:);
```

```
vvp12=[((i)/10):0.1:(t)]'*0+V1(i-1)-get1(:,i);
V1=[vvp11;vvp12];
end
if  i >= (vt2-0.1)*10 & abs(get2(:,i)) >= 0.1 &
    abs(get2(:,i)-get2(:,i-1)) <= 1e-2 & abs(r21(:,i)) <= 0.01
    & i < vt3*10
vvp21=V2(1:(i),:);
vvp22=[((i)/10):0.1:(t)]'*0+V2(i-1)-get2(:,i);
V2=[vvp21;vvp22];
end
if  i >= (vt3-0.1)*10 & abs(get3(:,i)) >= 0.1 &
    abs(get3(:,i)-get3(:,i-1)) <= 1e-2 & abs(r31(:,i)) <= 0.01
    & i < vt4*10
vvp31=V3(1:(i),:);
vvp32=[((i)/10):0.1:(t)]'*0+V3(i-1)-get3(:,i);
V3=[vvp31;vvp32];
end
if  i >= (vt4-0.1)*10 & abs(get4(:,i)) >= 0.1 &
    abs(r41(:,i)) <= 1e-3 & i < vt5*10
vvp41=V4(1:(i),:);
vvp42=[((i)/10):0.1:(t)]'*0+V4(i-1)-get4(:,i);
V4=[vvp41;vvp42];
end
if  i >= (vt5-0.1)*10 & abs(get5(:,i)) >= 0.1 &
    abs(get5(:,i)-get5(:,i-1)) <= 1e-2 & abs(r51(:,i)) <= 0.01
```

```
    & i < vt6*10
vvp51=V5(1:(i),:);
vvp52=[((i)/10):0.1:(t)]'*0+V5(i-1)-get5(:,i);
V5=[vvp51;vvp52];
end
if  i >= (vt6-0.1)*10 & abs(get6(:,i)) >= 0.1 &
    abs(get6(:,i)-get6(:,i-1)) <= 1e-2 & abs(r61(:,i)) <= 0.01
vvp61=V6(1:(i),:);
vvp62=[((i)/10):0.1:(t)]'*0+V6(i-1)-get6(:,i);
V6=[vvp61;vvp62];
end
end % End of main loop
% Transformation to the original system
x_out=inv(P)*x_out;
xh1_out=inv(P)*xh1_out;
xh2_out=inv(P)*xh2_out;
xh3_out=inv(P)*xh3_out;
xh4_out=inv(P)*xh4_out;
xh5_out=inv(P)*xh5_out;
xh6_out=inv(P)*xh6_out;
% Plots of failure detection and residuals
clg
subplot(211)
axis([ 0 140 -50 30])
plot(ttp1,get1), title('Detection of the first actuator failure')
```

```
xlabel('(a) Time(sec)'), ylabel('The estimate of the failure')
subplot(212)
axis([ 0 140 -40 40])
plot(ttp1,get2), title('Detection of the second actuator failure')
xlabel('(b) Time(sec)'), ylabel('The estimate of the failure')
pause
clg
subplot(211)
axis([ 0 140 -0.03 0.02])
plot(ttp,r11), title('Residual generation')
xlabel(' (c) Time(sec)'), ylabel('Residual e1')
pause
clg
subplot(211)
axis([ 0 140 -10 10])
plot(ttp1,get3),title('Detection of the third actuator failure')
xlabel('(a) Time(sec)'), ylabel('The estimate of the failure')
subplot(212)
axis([ 0 140 -20 40])
plot(ttp1,get4),title('Detection of the fourth actuator failure')
xlabel('(b) Time(sec)'), ylabel('The estimate of the failure')
pause
clg
axis([ 0 140 -0.008 0.008])
plot(ttp,r21),title('Residual generation')
```

```
xlabel(' (c) Time(sec)'), ylabel('Residual e2')
pause
clg
subplot(211)
axis([ 0 140 -20 40])
plot(ttp1,get5),title('Detection of the fifth actuator failure')
xlabel('(a) Time(sec)'), ylabel('The estimate of the failure')
subplot(212)
axis([ 0 140 -20 20])
plot(ttp1,get6),title('Detection of the sixth Actuator Failure')
xlabel('(b) Time(sec)'), ylabel('The estimate of the failure')
pause
clg
axis([ 0 140 -0.01 0.005])
plot(ttp,r31),title('Residual generation')
xlabel(' (c) Time(sec)'), ylabel('Residual e3')
pause
% Plots of the system's states and their estimates
clg
subplot(221)
axis([0 140 -1 1])
plot(ttp,x_out(1,:),ttp,xh2_out(1,:)),title('The First State')
xlabel('Time(sec)')
subplot(222)
axis([0 140 -0.02 0.02])
```

```
plot(ttp,x_out(2,:),ttp,xh2_out(2,:)),title('The Second State')
xlabel('Time(sec)')
subplot(223)
axis([0 140 -0.1 0.1])
plot(ttp,x_out(3,:),ttp,xh2_out(3,:)),title('The Third State')
xlabel('Time(sec)')
subplot(224)
axis([0 140 -0.1 0.1])
plot(ttp,x_out(4,:),ttp,xh2_out(4,:)),title('The Fourth State')
xlabel('Time(sec)')
pause

% This subroutine calculates the system parameters and linear
% transformation in Case 1
% System parameters
a0=[-0.0750 -24.0500 0 -32.1600;-0.0009 -0.1959 0.9896 0;
    -0.0002 -0.1454 -0.1677 0;0 0 1.0000 0];
b0=[-0.0230 0 -0.0729 0.0393 -0.0411 0.1600;
    -0.0002 -0.0001 -0.0004 -0.0000 -0.0003 -0.0003;
    -0.0067 -0.0007 -0.0120 -0.0006 -0.0007 0.0005;
     0 0 0 0 0 0];
c=[0 0 0 1;0 1 0 0;0 0 1 0;1 0 0 0];
d=b0;
% Linear transformation
P=[0 0 0 1;0 1 0 0;0 0 1 0;1 0 0 0];
```

```
a0=P*a0*inv(P);

b0=P*b0;

c=c*inv(P);

d=P*d;

% Closed-loop poles placement

pa=[-2 -3 -4 -5]';

% Feedback control gain

ka=place(a0,b0,pa);

% Closed-loop system

a0=a0-b0*ka;

[n,nl]=size(a0);

% Get equivalent zero-order hold discrete system

b=[b0 d];

[Ph1,Ga1]=c2d(a0,b,0.1);

[phr,phc]=size(Ph1);

[gar,gac]=size(Ga1);

Gc1=c;

[ch,cl]=size(c);


% This subroutine calculates three UIO parameters in Case 1

% Design of the first UIO

A=a0;B=b0;C=c(2:4,:);D=[d(:,1) d(:,2)];

% Linear transformation when designing UIO

c1=[1 0 0 0];

% Observer's gain
```

```
f1=-6;

% Unknown input observer design algorithm

thesis_uio

% The Results of the first UIO

F1=F;E1=E;L1=L;N1=N;

% Design and results of the second UIO

D=[d(:,3) d(:,4)];

thesis_uio

F2=F;E2=E;L2=L;N2=N;

% Design and results of the third UIO

D=[d(:,5) d(:,6)];

thesis_uio

F3=F;E3=E;L3=L;N3=N;

% Get equivalent zero-order hold discrete system

[Phu1,Gau1]=c2d(F1,[E1 L1],0.1);

[Phu2,Gau2]=c2d(F2,[E2 L2],0.1);

[Phu3,Gau3]=c2d(F3,[E3 L3],0.1);


% This subroutine is the UIO design algorithm for both Case 1 and Case 2

% The above is for p>m.

C1=input('Insert C1=')

T1=[C1;C]

RANKT1=rank(T1)

pause

[n,n]=size(A)
```

```
[n,k]=size(B)
[n,m]=size(D)
[p,n]=size(C)
AH=T1*A*inv(T1)
BH=T1*B
CH=C*inv(T1)
DH=T1*D
pause
if p>m;
A1=AH(1:n-p,:)
A2=AH(n-p+1:n-m,:)
pause
A3=AH(n-m+1:n,:)
B1=BH(1:n-p,:)
B2=BH(n-p+1:n-m,:)
B3=BH(n-m+1:n,:)
D1=DH(1:n-p,:)
D2=DH(n-p+1:n-m,:)
D3=DH(n-m+1:n,:)
AH1=A1-D1*inv(D3)*A3
AH2=A2-D2*inv(D3)*A3
BH1=B1-D1*inv(D3)*B3
BH2=B2-D2*inv(D3)*B3
A11=AH1(:,1:n-p)
A12=AH1(:,n-p+1:n-m)
```

```
A13=AH1(:,n-m+1:n)
A21=AH2(:,1:n-p)
A22=AH2(:,n-p+1:n-m)
A23=AH2(:,n-m+1:n)
[k1,k2]=size(A11)
[l1,l2]=size(A21)
OB=obsv(A11,A21)
RANKOB=rank(OB)
pause
Fsize=k1
F=input('Insert F=')
M=place(A11',A21',F)'
F=A11-M*A21
eig(F)
G=A12-M*A22+(A11-M*A21)*M
H=A13-M*A23+(A11-M*A21)*(D1-M*D2)*inv(D3)
L=BH1-M*BH2
% Observer:
E=[G H]
N=[M  (D1-M*D2)*inv(D3)]
else;
A11=AH(1:n-p,1:n-p);
A12=AH(1:n-p,n-p+1:n);
A21=AH(n-p+1:n,1:n-p);
A22=AH(n-p+1:n,n-p+1:n);
```

```
B1=BH(1:n-p,:)
B2=BH(n-p+1:n,:);
D1=DH(1:n-p,:);
D2=DH(n-p+1:n,:);
N=D1*inv(D2);
G=A11-N*A21;
EIGG=eig(G)
pause
H=B1-N*B2
L=A12-N*A22+G*N;
[ts,s] = size(N);
M = eye(ts);
end

% This subroutine calculates the initial conditions of the variables and
% the variables needed to obtain the estimate of state variables in Case 1
x0=[1 -1 2 -2]';
x0=P*x0;
[fr,fc]=size(F);
[elc,ell]=size([E L]);
[nc,nl]=size(N);
R1=[eye(fc);zeros(phr-fr,fc)];
R2=R1;
R3=R1;
P1=[N1;eye(nl)];
```

```
P2=[N2;eye(nl)];
P3=[N3;eye(nl)];
xh0=[-4 -1 2 -2]';
xh0=P*xh0;
z10=T1*xh0;
z20=T1*xh0;
z30=T1*xh0;
w10=inv(R1'*R1)*R1'*(z10-P1*c(2:4,:)*x0);
w20=inv(R2'*R2)*R2'*(z20-P2*c(2:4,:)*x0);
w30=inv(R3'*R3)*R3'*(z30-P3*c(2:4,:)*x0);

% This subroutine is the function which calculates the state variables
% and output varaibles in Case 1
function [x,y]=sys(Ph1,Ga1,Gc1,U,V,x);
x=Ph1*x+Ga1*[U;V];
y=Gc1*x;

% This subroutine calculates the estimates of the state variable in Case 1
function [w,z,xh]=esti(Phu,Gau,R,P.y,U,T1,w);
w=Phu*w+Gau*[y;U];
z=R*w+P*y;
xh=inv(T1)*z;


% This is the main program which simulates Case 2
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This is the main program
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Simulation time set up
t=140;
tt0=t*10;
ttp=(0:0.1:t)';
ttp1=ttp(1:1400,:);
% Reference input
u11=ttp*0+20;
u21=ttp*0;
u31=ttp*0;
u41=ttp*0;
u51=ttp*0;
u61=ttp*0;
% The time and the failure parameters of the actuators
vt1=20;
vt2=40;
vt3=60;
vt4=80;
vt5=100;
vt6=120;
tt11=(0:0.1:19.9);
tt12=(20:0.1:t);
V1=[tt11'*0;(5*sin(tt12'))];
```

```
v3=[0:0.1:vt2-0.1]'*0+0;

v4=[vt2:0.1:t]'*0-10;

V2=[v3;v4];

v5=[0:0.1:vt3-0.1]'*0+0;

v6=[vt3:0.1:t]'*0+6;

V3=[v5;v6];

tt41=(0:0.1:79.9);

tt42=(80:0.1:t);

V4=[tt41'*0;(-8*cos(tt42'))];

v9=[0:0.1:vt5-0.1]'*0+0;

v10=[vt5:0.1:t]'*0+10;

V5=[v9;v10];

v11=[0:0.1:vt6-0.1]'*0+0;

v12=[vt6:0.1:t]'*0+6;

V6=[v11;v12];

% Initial condition of the variables in the main loop'

x=x0 ;

xh1=xh0;

xh2=xh0;

xh3=xh0;

xh4=xh0;

xh5=xh0;

xh6=xh0;

x_out(:,1)=x0;

y_out(:,1)=c*x0;
```

```
w1=w10;
w2=w20;
w3=w30;
w4=w40;
w5=w50;
w6=w60;
w1_out(:,1)=w10;
w2_out(:,1)=w20;
w3_out(:,1)=w30;
w4_out(:,1)=w40;
w5_out(:,1)=w50;
w6_out(:,1)=w60;
xh1_out(:,1)=xh0;
xh2_out(:,1)=xh0;
xh3_out(:,1)=xh0;
xh4_out(:,1)=xh0;
xh5_out(:,1)=xh0;
xh6_out(:,1)=xh0;
% The main loop
for i=1:tt0,
U=[u11(i) u21(i) u31(i) u41(i) u51(i) u61(i)]';
V=[V1(i) V2(i) V3(i) V4(i) V5(i) V6(i)]';
xx=xh1_out(:,i);
[x,y]=syskxh(Ph1,Ga1,Gc1,xx,U,V,x);
x_out(:,i+1)=x;
```

```
y_out(:,i+1)=y;
y234=y(2:4,:);
% This is the UIO parameter calculation
[w1,z1,xh1]=estikxh1(Phu1,Gau1,R1,P1,y234,xx,U,T1,w1);
w1_out(:,i+1)=w1;
xh1_out(:,i+1)=xh1;
r1=y_out-c*xh1_out;
r11=r1(1,:);
[w2,z2,xh2]=estikxh1(Phu2,Gau2,R2,P2,y234,xx,U,T1,w2);
w2_out(:,i+1)=w2;
xh2_out(:,i+1)=xh2;
r2=y_out-c*xh2_out;
r21=r2(1,:);
[w3,z3,xh3]=estikxh1(Phu3,Gau3,R3,P3,y234,xx,U,T1,w3);
w3_out(:,i+1)=w3;
xh3_out(:,i+1)=xh3;
r3=y_out-c*xh3_out;
r31=r3(1,:);
[w4,z4,xh4]=estikxh1(Phu4,Gau4,R4,P4,y234,xx,U,T1,w4);
w4_out(:,i+1)=w4;
xh4_out(:,i+1)=xh4;
r4=y_out-c*xh4_out;
r41=r4(1,:);
[w5,z5,xh5]=estikxh1(Phu5,Gau5,R5,P5,y234,xx,U,T1,w5);
w5_out(:,i+1)=w5;
```

```
xh5_out(:,i+1)=xh5;
r5=y_out-c*xh5_out;
r51=r5(1,:);
[w6,z6,xh6]=estikxh1(Phu6,Gau6,R6,P6,y234,xx,U,T1,w6);
w6_out(:,i+1)=w6;
xh6_out(:,i+1)=xh6;
r6=y_out-c*xh6_out;
r61=r6(1,:);
% This is failure detection and identification for actuator 1 to actuator 6
Ac1=[A0];
Bc1=[-b1*ka b1 d(:,1)];
[Ph1v,Ga1v]=c2d(Ac1,Bc1,0.1);
dstar1=Ga1v(:,11);
xvh=xh1_out;
s1(:,i)=xvh(:,i+1)-Ph1v*xvh(:,i)-Ga1v(:,1:10)*[xx;U];
v1h(:,i)=inv(dstar1'*dstar1)*dstar1'*s1(:,i);
get1=v1h(1,:);
Ac2=[A0];
Bc2=[-b1*ka b1 d(:,2)];
[Ph2v,Ga2v]=c2d(Ac2,Bc2,0.1);
dstar2=Ga2v(:,11);
s2(:,i)=xvh(:,i+1)-Ph2v*xvh(:,i)-Ga2v(:,1:10)*[xx;U];
v2h(:,i)=inv(dstar2'*dstar2)*dstar2'*s2(:,i);
get2=v2h(1,:);
Ac3=[A0];
```

```
Bc3=[-b1*ka b1 d(:,3)];
[Ph3v,Ga3v]=c2d(Ac3,Bc3,0.1);
dstar3=Ga3v(:,11);
s3(:,i)=xvh(:,i+1)-Ph3v*xvh(:,i)-Ga3v(:,1:10)*[xx;U];
v3h(:,i)=inv(dstar3'*dstar3)*dstar3'*s3(:,i);
get3=v3h(1,:);
Ac4=[A0];
Bc4=[-b1*ka b1 d(:,4)];
[Ph4v,Ga4v]=c2d(Ac4,Bc4,0.1);
dstar4=Ga4v(:,11);
s4(:,i)=xvh(:,i+1)-Ph4v*xvh(:,i)-Ga4v(:,1:10)*[xx;U];
v4h(:,i)=inv(dstar4'*dstar4)*dstar4'*s4(:,i);
get4=v4h(1,:);
Ac5=[A0];
Bc5=[-b1*ka b1 d(:,5)];
[Ph5v,Ga5v]=c2d(Ac5,Bc5,0.1);
dstar5=Ga5v(:,11);
s5(:,i)=xvh(:,i+1)-Ph5v*xvh(:,i)-Ga5v(:,1:10)*[xx;U];
v5h(:,i)=inv(dstar5'*dstar5)*dstar5'*s5(:,i);
get5=v5h(1,:);
Ac6=[A0];
Bc6=[-b1*ka b1 d(:,6)];
[Ph6v,Ga6v]=c2d(Ac6,Bc6,0.1);
dstar6=Ga6v(:,11);
s6(:,i)=xvh(:,i+1)-Ph6v*xvh(:,i)-Ga6v(:,1:10)*[xx;U];
```

```
v6h(:,i)=inv(dstar6'*dstar6)*dstar6'*s6(:,i);
get6=v6h(1,:);
% Failure isolation and accommodation
if  i >= (vt1-0.1)*10 & abs(get1(:,i)) >= 0.1 &
    abs(r11(:,i)) <= 0.1 & i < vt2*10
vvp11=V1(1:(i),:);
vvp12=[((i)/10):0.1:(t)]'*0+V1(i-1)-get1(:,i);
V1=[vvp11;vvp12];
end
if  i >= (vt2-0.1)*10 & abs(get2(:,i)) >= 0.1 &
    abs(get2(:,i)-get2(:,i-1)) <= 1e-3 & abs(r21(:,i)) <= 0.1
    & i < vt3*10
vvp21=V2(1:(i),:);
vvp22=[((i)/10):0.1:(t)]'*0+V2(i-1)-get2(:,i);
V2=[vvp21;vvp22];
end
if  i >= (vt3-0.1)*10 & abs(get3(:,i)) >= 0.1 &
    abs(get3(:,i)-get3(:,i-1)) <= 1e-3 & abs(r31(:,i)) <= 0.1
    & i < vt4*10
vvp31=V3(1:(i),:);
vvp32=[((i)/10):0.1:(t)]'*0+V3(i-1)-get3(:,i);
V3=[vvp31;vvp32];
end
if  i >= (vt4-0.1)*10 & abs(get4(:,i)) >= 0.1 &
    abs(r41(:,i)) <= 0.1 & i < vt5*10
```

```
vvp41=V4(1:(i),:);
vvp42=[((i)/10):0.1:(t)]'*0+V4(i-1)-get4(:,i);
V4=[vvp41;vvp42];
end
if  i >= (vt5-0.1)*10 & abs(get5(:,i)) >= 0.1 &
    abs(get5(:,i)-get5(:,i-1)) <= 1e-3 & abs(r51(:,i)) <= 0.1
    & i < vt6*10
vvp51=V5(1:(i),:);
vvp52=[((i)/10):0.1:(t)]'*0+V5(i-1)-get5(:,i);
V5=[vvp51;vvp52];
end
if  i >= (vt6-0.1)*10 & abs(get6(:,i)) >= 0.1 &
    abs(get6(:,i)-get6(:,i-1)) <= 1e-3 & abs(r61(:,i)) <= 0.1
vvp61=V6(1:(i),:);
vvp62=[((i)/10):0.1:(t)]'*0+V6(i-1)-get6(:,i);
V6=[vvp61;vvp62];
end
end % End of main loop
% Plots of failure detection and residuals
clg
subplot(211)
axis([ 0 140 -20 20])
plot(ttp1,get1), title('Detection of the first actuator failure')
xlabel('(a) Time(sec)'), ylabel('The estimate of the failure')
subplot(212)
```

```
axis([ 0 140 -2 2])
plot(ttp,r11), title('Residual generation')
xlabel(' (b) Time(sec)'), ylabel('Residual e1')
pause
clg
subplot(211)
axis([ 0 140 -20 20])
plot(ttp1,get2), title('Detection of the second actuator failure')
xlabel('(c) Time(sec)'), ylabel('The estimate of the failure')
subplot(212)
axis([ 0 140 -5 5])
plot(ttp,r21), title('Residual generation')
xlabel(' (d) Time(sec)'), ylabel('Residual e2')'
pause
clg
subplot(211)
axis([ 0 140 -20 20])
plot(ttp1,get3), title('Detection of the third actuator failure')
xlabel('(a) Time(sec)'), ylabel('The estimate of the failure')
subplot(212)
axis([ 0 140 -2 2])
plot(ttp,r31), title('Residual generation')
xlabel(' (b) Time(sec)'), ylabel('Residual e3')
pause
clg
```

```
subplot(211)
axis([ 0 140 -20 20])
plot(ttp1,get4), title('Detection of the fourth actuator failure')
xlabel('(c) Time(sec)'), ylabel('The estimate of the failure')
subplot(212)
axis([ 0 140 -8 8])
plot(ttp,r41), title('Residual generation')
xlabel(' (d) Time(sec)'), ylabel('Residual e4')
pause
clg
subplot(211)
axis([ 0 140 -20 20])
plot(ttp1,get5), title('Detection of the fifth actuator failure')
xlabel('(a) Time(sec)'), ylabel('The estimate of the failure')
subplot(212)
axis([ 0 140 -10 10])
plot(ttp,r51), title('Residual generation')
xlabel(' (b) Time(sec)'), ylabel('Residual e5')
pause
clg
subplot(211)
axis([ 0 140 -10 10])
plot(ttp1,get6), title('Detection of the sixth actuator failure')
xlabel('(c) Time(sec)'), ylabel('The estimate of the failure')
subplot(212)
```

```
axis([ 0 140 -20 20])
plot(ttp,r61), title('Residual generation')
xlabel(' (d) Time(sec)'), ylabel('Residual e6')
pause
% Plots of the system's states and their estimates
clg
subplot(221)
axis([0 140 -20 20])
plot(ttp,x_out(1,:),ttp,xh1_out(1,:)),title('The First State')
xlabel('Time(sec)')
subplot(222)
axis([0 140 -0.1 0.1])
plot(ttp,x_out(2,:),ttp,xh1_out(2,:)),title('The Second State')
xlabel('Time(sec)')
subplot(223)
axis([0 140 -1 1])
plot(ttp,x_out(3,:),ttp,xh1_out(3,:)),title('The Third State')
xlabel('Time(sec)')
subplot(224)
axis([0 140 -1 1])
plot(ttp,x_out(4,:),ttp,xh1_out(4,:)),title('The Fourth State')
xlabel('Time(sec)')
pause
clg
subplot(221)
```

```
axis([0 140 -20 20])
plot(ttp,x_out(1,:),ttp,xh2_out(1,:)),title('The First State')
xlabel('Time(sec)')
subplot(222)
axis([0 140 -0.1 0.1])
plot(ttp,x_out(2,:),ttp,xh2_out(2,:)),title('The Second State')
xlabel('Time(sec)')
subplot(223)
axis([0 140 -1 1])
plot(ttp,x_out(3,:),ttp,xh2_out(3,:)),title('The Third State')
xlabel('Time(sec)')
subplot(224)
axis([0 140 -1 1])
plot(ttp,x_out(4,:),ttp,xh2_out(4,:)),title('The Fourth State')
xlabel('Time(sec)')
pause
clg
subplot(221)
axis([0 140 -20 20])
plot(ttp,x_out(1,:),ttp,xh3_out(1,:)),title('The First State')
xlabel('Time(sec)')
subplot(222)
axis([0 140 -0.1 0.1])
plot(ttp,x_out(2,:),ttp,xh3_out(2,:)),title('The Second State')
xlabel('Time(sec)')
```

```
subplot(223)
axis([0 140 -1 1])
plot(ttp,x_out(3,:),ttp,xh3_out(3,:)),title('The Third State')
xlabel('Time(sec)')
subplot(224)
axis([0 140 -1 1])
plot(ttp,x_out(4,:),ttp,xh3_out(4,:)),title('The Fourth State')
xlabel('Time(sec)')
pause
clg
subplot(221)
axis([0 140 -20 20])
plot(ttp,x_out(1,:),ttp,xh4_out(1,:)),title('The First State')
xlabel('Time(sec)')
subplot(222)
axis([0 140 -0.1 0.1])
plot(ttp,x_out(2,:),ttp,xh4_out(2,:)),title('The Second State')
xlabel('Time(sec)')
subplot(223)
axis([0 140 -1 1])
plot(ttp,x_out(3,:),ttp,xh4_out(3,:)),title('The Third State')
xlabel('Time(sec)')
subplot(224)
axis([0 140 -1 1])
plot(ttp,x_out(4,:),ttp,xh4_out(4,:)),title('The Fourth State')
```

```
xlabel('Time(sec)')
pause
clg
subplot(221)
axis([0 140 -20 20])
plot(ttp,x_out(1,:),ttp,xh5_out(1,:)),title('The First State')
xlabel('Time(sec)')
subplot(222)
axis([0 140 -0.1 0.1])
plot(ttp,x_out(2,:),ttp,xh5_out(2,:)),title('The Second State')
xlabel('Time(sec)')
subplot(223)
axis([0 140 -1 1])
plot(ttp,x_out(3,:),ttp,xh5_out(3,:)),title('The Third State')
xlabel('Time(sec)')
subplot(224)
axis([0 140 -1 1])
plot(ttp,x_out(4,:),ttp,xh5_out(4,:)),title('The Fourth State')
xlabel('Time(sec)')
pause
clg
subplot(221)
axis([0 140 -20 20])
plot(ttp,x_out(1,:),ttp,xh6_out(1,:)),title('The First State')
xlabel('Time(sec)')
```

```
subplot(222)
axis([0 140 -0.1 0.1])
plot(ttp,x_out(2,:),ttp,xh6_out(2,:)),title('The Second State')
xlabel('Time(sec)')
subplot(223)
axis([0 140 -1 1])
plot(ttp,x_out(3,:),ttp,xh6_out(3,:)),title('The Third State')
xlabel('Time(sec)')
subplot(224)
axis([0 140 -1 1])
plot(ttp,x_out(4,:),ttp,xh6_out(4,:)),title('The Fourth State')
xlabel('Time(sec)')
pause

% This subroutine calculates system parameters in Case 2
% System Parameters
a0=[-0.0750 -24.0500 0 -32.1600;-0.0009 -0.1959 0.9896 0;
    -0.0002 -0.1454 -0.1677 0;0 0 1.0000 0];
[n,nl]=size(a0);
b0=[-0.0230 0 -0.0729 0.0393 -0.0411 0.1600;
    -0.0002 -0.0001 -0.0004 -0.0000 -0.0003 -0.0003;
    -0.0067 -0.0007 -0.0120 -0.0006 -0.0007 0.0005;
     0 0 0 0 0 0];
c=[1 0 0 0;0 1 0 0;0 0 1 0;0 0 0 1];
d=b0;
```

```
% Uncertainty effects
Ia=[0 0 0 1]';
deltA=[0 0.3 -0.05 0.8];
unA=Ia*deltA;
% Closed-loop poles placement
pa=[-2 -3 -4 -5]';
% Feedback control gain
ka=place(a0,b0,pa);
% System with uncertainties
A0=a0+unA;
% Get equivalent zero-order hold discrete system
b=[-b0*ka b0 d];
[Ph1,Ga1]=c2d(A0,b,0.1);
[phr,phc]=size(Ph1);
[gar,gac]=size(Ga1);
Gc1=c;
[ch,cl]=size(c);


% This subroutine calculates three UIO parameters in Case 2
% Design of the first UIO
A=a0;B=b0;C=c(2:4,:);D=[d(:,1) Ia];
c1=[1 0 0 0];
f1=-10;
thesis_uio
F1=F;E1=E;L1=L;N1=N;
```

```
% Design and results of the second UIO
D=[d(:,2) Ia];
thesis_uio
F2=F;E2=E;L2=L;N2=N;
% Design and results of the third UIO
D=[d(:,3) Ia];
thesis_uio
F3=F;E3=E;L3=L;N3=N;
% Design and results of the fourth UIO
D=[d(:,4) Ia];
thesis_uio
F4=F;E4=E;L4=L;N4=N;
% Design and results of the fifth UIO
D=[d(:,5) Ia];
thesis_uio
F5=F;E5=E;L5=L;N5=N;
% Design and results of the sixth UIO
D=[d(:,6) Ia];
thesis_uio
F6=F;E6=E;L6=L;N6=N;
% Get equivalent zero-order hold discrete system
[Phu1,Gau1]=c2d(F1,[E1 -L1*ka L1],0.1);
[Phu2,Gau2]=c2d(F2,[E2 -L2*ka L2],0.1);
[Phu3,Gau3]=c2d(F3,[E3 -L3*ka L3],0.1);
[Phu4,Gau4]=c2d(F4,[E4 -L4*ka L4],0.1);
```

```
[Phu5,Gau5]=c2d(F5,[E5 -L5*ka L5],0.1);
[Phu6,Gau6]=c2d(F6,[E6 -L6*ka L6],0.1);

% This subroutine calculates the initial conditions of the variables and
% the variables needed to obtain the estimate of state variables in Case 2
x0=[1 -1 2 -2]';
[fr,fc]=size(F);
[elc,ell]=size([E L]);
[nc,nl]=size(N);
R1=[eye(fc);zeros(phr-fr,fc)];
R2=R1;
R3=R1;
R4=R1;
R5=R1;
R6=R1;
P1=[N1;eye(nl)];
P2=[N2;eye(nl)];
P3=[N3;eye(nl)];
P4=[N4;eye(nl)];
P5=[N5;eye(nl)];
P6=[N6;eye(nl)];
xh0=[0.9 -1 2 -2]';
z10=T1*xh0;
z20=T1*xh0;
z30=T1*xh0;
```

```
z40=T1*xh0;
z50=T1*xh0;
z60=T1*xh0;
w10=inv(R1'*R1)*R1'*(z10-P1*c(2:4,:)*x0);
w20=inv(R2'*R2)*R2'*(z20-P2*c(2:4,:)*x0);
w30=inv(R3'*R3)*R3'*(z30-P3*c(2:4,:)*x0);
w40=inv(R4'*R4)*R4'*(z40-P4*c(2:4,:)*x0);
w50=inv(R5'*R5)*R5'*(z50-P5*c(2:4,:)*x0);
w60=inv(R6'*R6)*R6'*(z60-P6*c(2:4,:)*x0);

% This subroutine is the function which calculates the state variables
% and output varaibles in Case2
function [x,y]=syskxh(Ph1,Ga1,Gc1,xh,U,V,x);
x=Ph1*x+Ga1*[xh;U;V];
y=Gc1*x;

% This subroutine calculates the estimates of the state variable in Case 2
function [w,z,xh]=estikxh1(Phu,Gau,R,P,y,xh,U,T1,w);
w=Phu*w+Gau*[y;xh;U];
z=R*w+P*y;
xh=inv(T1)*z;
```

REFERENCES

[1] Athens, M. (1982), *On Restructurable Control System Theory,* **NASA CP-2277,** Proc. of A Workshop Held at NASA Langley Research Center, Hampton, VA.

[2] Basile, G. and Marro, G., (1969), *On the Observability of Linear Time-invariant Systems with Unknown Inputs,* **Journal of Optimization Theory and Application,** Vol. 3, No. 6, pp. 410-415.

[3] Basile, G. and Marro, G., (1973), *A New Characterization of Some Structural Properties of Linear Systems: Unknown-Input Observability, Invertibility and Functional Controllability,* **Int. J. Control,** Vol. 17, No. 5, pp. 931-943.

[4] Chen, C. T., **Linear System Theory and Design,** Holt, Rinehart and Wilson Publishing, New York, 1984.

[5] Chow, E. Y. (1980), *A Failure Detection System Design Methodology,* **Sc. D. Thesis,** Lab. Inform. Decision Syst., Mass. Inst. Technol., Cambridge, MA.

[6] Chow, E. Y. and Willsky, A. S., (1984), *Analytical Redundancy and the Design of Robust Failure Detection Systems,* **IEEE Trans. on Auto. Cont.**, Vol. AC-29, pp. 603-614.

[7] Clark, R. N., (1978), *Instrument Fault Detection,* **IEEE Trans. on Aero. and Electronic Syst.,** Vol. AES-14, pp. 456-465.

[8] Clark, R. N., (1978), *A Simplified Instrument Failure Detection Scheme,* **IEEE Trans. on Aero. and Electronic Syst.,** Vol. AES-14, pp. 558-563.

[9] Clark, R. N. and Setzer, W., (1980), *Sensor Fault Detection in a System with Random Disturbances,* **IEEE Trans. on Aero. and Electronic Syst.**, Vol. AES-16, No. 14, pp. 468-473.

[10] Clark, R. N., Fosth, D. C., and Walton, V. W., (1975), *Detecting Instrument Malfunctions in Control Systems,* **IEEE Trans. on Aero. and Electronic Syst.,** Vol. AES-11, pp. 465-473.

[11] Clark, R. N. and Setzer, W. (1980), *Sensor Fault Detection in System with Random Disturbances,* **IEEE Trans. on Aero. and Electronic Syst.**, Vol. AES-16, No. 4, pp. 468-473.

[12] Deckert, J.C., Deyst, J. J., Desai, M. N. and Willsky, A. S. (1977), *F-8 DF BW Sensor Failure Identification Using Analytic Redundancy,* **IEEE Trans. on Auto. Cont.,** Vol. AC-22, No. 5, pp. 795-803.

[13] Frank, P. M., (1990), *Fault Diagnosis in Dynamic Systems Using Analytical and Knowledge-Based Redundancy - A Survey and Some New Results,* **Automatica,** Vol. 26, pp. 459-474.

[14] Frank, P. M., (1987), *Fault Diagnosis in Dynamic Systems Via State Estimation - A Survey,* **System Fault Diagnosis, Reliability and Related Knowledge Based Approaches,** S. Tzafesttats et. al. (Eds), Vol. 1, pp. 35-98, D. Reidel Publishing Co.

[15] Frank, P. M. and Keller, L., (1980), *Sensitivity Discriminating Observer Design for Instrument Failure Detection,* **IEEE Trans. on Aero. and Electronic Syst.,** Vol. AES-16, No. 14, pp. 460-467.

[16] Gopinath, B., (1971), *On the Control of Linear Multiple Input-Output Systems,* **Bell System Tech. Journal,** Vol. 50, No. 3, pp.1063-1081.

[17] Guan, Y., and Saif, M., (1991), *A New approach for Sensor Failure Detection in Dynamic Systems,* **Proc. of The Canadian Conference in Electrical Engineering,** Vol. II, pp. 1001-1004, Montreal, Canada, September, 1989.

[18] Guan, Y., and Saif, M., (1991), *A Novel approach to the Design of Unknown Input Observers,* **IEEE Trans. on Auto. Cont.,** Vol. 36, pp. 632-635.

[19] Guan, Y., (1990), *Robust Estimation with Application to Failure Detection and Identification (FDI) in Dynamical Systems,* M. A. Sc. Thesis, Simon Fraser University, Burnaby, B. C., Canada.

[20] Huang, B., and Saif, M. and Jamshidi, M., (1993), *A Neural Network Based Fault Detection and Identification (FDI) for a Presurized Water Reactor*, **Proc. of The 12th IFAC World Congress**, Sydney, Australia, July, 1993.

[21] Hostetter, G. H., **Digital Control System Design,** Holt, Rinehart and Wilson, Inc., The Dryden Press, 1987.

[22] Isermann, R., (1984), *Process Fault Detection Based on Modeling and Estimation Methods - A Survey,* **Automatica,** Vol. 20, pp. 384-404.

[23] Kailath, T., **Linear Systems,** Prentice-Hall International, 1980.

[24] Kudva, P., Viswanadham, N., and Ramakrishna, A., (1980), *Observers for Linear Systems with Unknown Inputs,* **IEEE Trans. on Auto. Cont.,** Vol. AC-25, pp. 113-115.

[25] Kurek, J. E., (1982), *Observation of the State Vector of Linear Multivariable Systems with Unknown Input,* **Int. J. Control,** Vol. 36, No. 3, pp. 511-515.

[26] Kurek, J. E., (1983), *The State Vector Reconstruction for Linear Systems with Unknown Inputs,* **IEEE Trans. on Auto. Cont.,** Vol. AC-28, pp. 1120-1122.

[27] Leininger, G. G. (1981), *Model degradation effects on Sensor Failure Detection,* **Proc. of 1981 Joint Automatic Control Conference.**

[28] Lou, X. C. (1982), *A System Failure Detection Method – Failure Projection Method,* **S. M. Thesis,** Lab. Inform. Decision Syst., Mass. Inst. Technol., Cambridge, MA.

[29] Lou, X-C , Willsky, A. S. and Verghese, G. C., (1986), *Optimally Robust Redundancy Relations for Failure Detection in Uncertain Systems,* **Automatica,** Vol. 22, No. 3, pp. 333-344.

[30] Mehra, R. K. and Peshon, I. (1971), *An innovations approach to fault detection and diagnosis in dynamic systems,* **Automatica,** 7, pp. 637-640.

[31] Montogomery, R. C. and Caglayan, A. K. (1976), *Failure Accommodation in Digital Flight Control Systems by Bayesian Decision Theory,* **J. Aircraft,** Vol. 13, No. 3.

[32] Onken, R. and Stuckenberg, N. (1979), *Failure Detection in Signal Processing and Sensing in Flight Control Systems,* **Proc. of the 1978 IEEE Conf. on Decision and Control,** San Diego, California.

[33] Patton, R., Frank, P. and Clark, R. (Editors), **Fault Diagnosis in Dynamic Systems – Theory and Application**, Prentice Hall, 1989.

[34] Pau, L. F., **Failure Diagnosis and Performance Monitoring**, Marcel Dekker Inc, 1981.

[35] Shapiro, E. Y. and Decarli, H. E. (1979), *Analytic Redundancy for Flight Control Sensors on the Lockheed L-1011 Aircraft,* **IEEE Conf. on Decision and Control,** San Diego, California.

[36] Saif, M., and Villaseca, F.E., (1986), *Sensor Failure Detection in Optimal Controlled Systems,* **Proceedings of ACC,** pp. 2104-2110, Seattle, WA., USA, June, 1986.

[37] Saif, M., and Villaseca, F.E., (1987a), *Optimal Control Under Distributed Sensor Failure Detectors,* **Control of Distributed Parameter Systems, H.E. Ranch (Editor), Pergamon Press, 1987.**

[38] Saif, M., and Villaseca, F.E., (1987b), *Simplified Sensor Failure Detection Scheme,* **Circuits and Systems, G. Glasford and K. Jabbour (Editors), North Holland, pp. 660-665.**

[39] Saif, M., and Guan, Y., (1991), *Sensor and Actuator Failure Detection Using UIO Theory,* **Invited Paper, Proceedings of The COMCON3,** Victoria, B.C., Canada, October, 1991.

[40] Saif, M., and Guan, Y., (1992), *Decentralized State Estimation in Large-Scale Interconnected Dynamical Systems,* **Automatica,** Vol. 28, No. 1, pp. 215-219.

[41] Saif, M., (1993), *Estimation and Fault Detection in Discrete-Time Systems with Noisy Measurements,* **Proceedings of The IASTED, International Conference on Modelling and Simulation,** Pittsburgh, PA., USA, May, 1993.

[42] Saif, M., (1993), *A Disturbance Accommodating Estimator For Bilinear System,* **Proceedings of ACC,** San Fransisco, CA., USA, June, 1993.

[43] Saif, M., and Guan, Y., (1993), *A New Approach to Robust Fault Detection and Identification,* **to appear in IEEE Trans. on Aero. and Electronic Syst.,** Vol. 29, No. 3, July, 1993.

[44] Viswanadham, N., and Srichander, R., (1987), *Fault Detection Using Unknown Input Observers,* **Control Theory and Advanced Technology (C-TAT),** Vol. 3, pp. 91-101.

[45] Voulgaris, P and Valavani, L. (1991), *High Performance Linear-Quadratic and H-Infinity Design for a "Supermaneuverable" Aircraft,* **Journal od Guidance, Control and Dynamics**, Vol. 14, No. 1, pp. 157-165.

[46] Wang, S. H., Davison, E. J., and Dorato, P., (1975), *Observing the States of Systems with Unmeasurable Disturbances,* **IEEE Trans. on Auto. Cont.,** Vol. AC-20, pp. 716-717.

[47] Watanabe, K. and Himmeblau, D. M., (1982), *Instrument Fault Detection in Systems with Uncertainties,* **Int. J. Systems Sci.,** Vol. 13, No. 2, pp.137-158.

[48] Watanabe, K., Yoshimura, T. and Sodea, T. (1981), *A Diagnosis Method for Linear Stochastic Systems with Parameter Failures,* **Trans. on the ASME,** Vol. 103, No. 2, pp.28-35.

[49] Willsky, A. S., (1976), *A Survey of Design Methods for Failure Detection in Dynamic System,* **Automatica,** Vol. 12, pp. 601-611.

[50] Wood, K., Hanley, L. G., Lawson, B., Randall, C., Turner, R., and Wang, C. W., (1989), *Shuttle Failure detection* **Aerospace America,** Vol. 27, No. 7, pp. 34-36.

[51] Wünnenberg, J. and Frank, P. M., (1988), *Model-Based Residual Generation for Dynamic Systems with Unknown Inputs,* **12th IMACS World Congress on Scientific Computation,** Paris, pp. 18-22.

[52] Yang, F., and Wilde, R. W., (1988), *Observers for Linear Systems with Unknown Inputs,* **IEEE Trans. on Auto. Cont.,** Vol. AC-33, pp. 677-681.